

## 8.3 Register summary

The register summary tables list the registers in CMN S3(AE).



When `LEGACY_TZ_EN` input pin is asserted:

- Root registers will be accessible as Secure registers.
- Realm registers will be accessible as Non-Secure registers.
- For CHI and AP Non-Secure transactions with the Non-Secure attribute, Root registers cannot be accessed. To allow access set the Root Control Register override for the register.
- For APB interface, tie PNSE pin to high to allow APB side access.

The Root-owned Root Control Register (RCR) and Secure-owned Secure Control Register (SCR) registers provide the ability to override the accessibility of register groups. Each group has a corresponding bit in both RCR and SCR. The following table shows the override behavior.

**Table 8-5: Root RAS register accessibility overrides**

RCR	SCR	Root RAS Register Set Accessibility
0	0	Default accessibility as described in Usage Constraints
0	1	A Secure-owned register can be accessed with Non-secure transactions
1	0	A Root-owned register can be accessed by Secure transactions
1	1	Root and Secure-owned registers can be accessed by Non-secure transactions

### 8.3.1 APB register summary

The following table describes the registers for the relevant component.

**Table 8-6: por\_apb\_cfg register summary**

Offset	Name	Type	Description
0x0	por_apb_node_info	RO	<a href="#">por_apb_node_info</a>
0x80	por_apb_child_info	RO	<a href="#">por_apb_child_info</a>
0x990	por_apb_rcr	RW	<a href="#">por_apb_rcr</a>
0x998	por_apb_scr	RW	<a href="#">por_apb_scr</a>
0x980	por_apb_only_access	RW	<a href="#">por_apb_only_access</a>
0x988	por_axu_control	RW	<a href="#">por_axu_control</a>

### 8.3.1.1 por\_apb\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

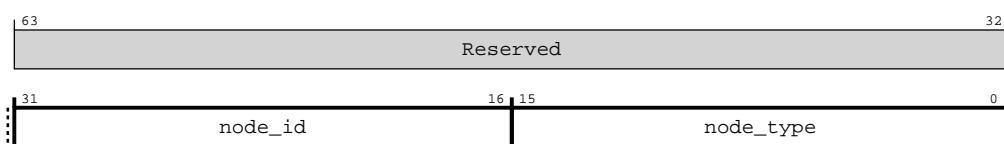
##### Usage constraints

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 8-9: por\_apb\_node\_info**



**Table 8-7: por\_apb\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x1000

### 8.3.1.2 por\_apb\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x80

### Type

RO

### Reset value

See individual bit resets

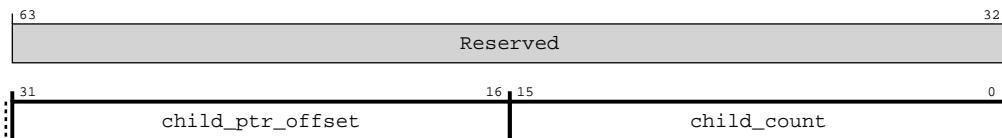
### Usage constraints

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 8-10: por\_apb\_child\_info**



**Table 8-8: por\_apb\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0000
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.1.3 por\_apb\_rcr

Root register access override

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x990

### Type

RW

### Reset value

See individual bit resets

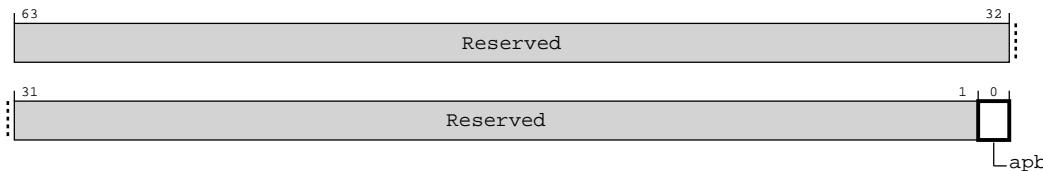
### Usage constraints

This register is owned in the Root space and is 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 8-11: por\_apb\_rcr**



**Table 8-9: por\_apb\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	apb	Allows access to root APB registers	RW	0b0

### 8.3.1.4 por\_apb\_scr

Secure register access override

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x998

#### Type

RW

#### Reset value

See individual bit resets

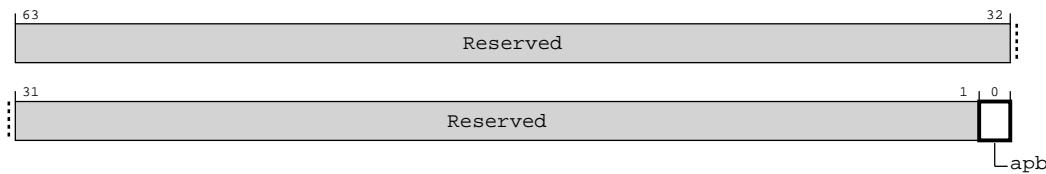
## Usage constraints

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 8-12: por\_apb\_scr**



**Table 8-10: por\_apb\_scr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	apb	Allows access to secure APB registers	RW	0b0

### 8.3.1.5 por\_apb\_only\_access

Functions as the CMN access control register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x980

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_apb\_rcr.apb

### Secure group override

por\_apb\_scr.apb

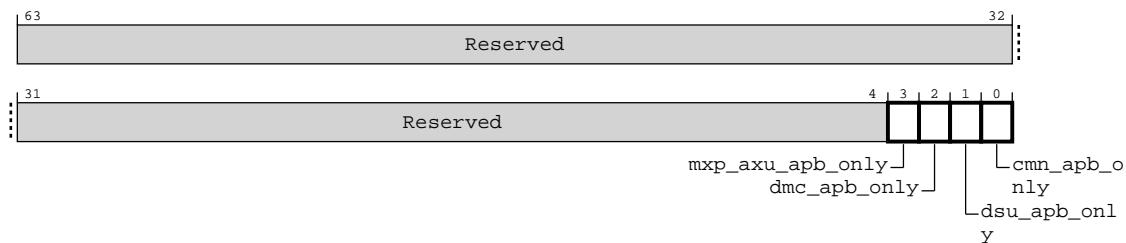
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_apb\_scr.apb bit is set, Secure accesses to this register are permitted. If both the por\_apb\_scr.apb bit and por\_apb\_rcr.apb 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 8-13: por\_apb\_only\_access**



**Table 8-11: por\_apb\_only\_access attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	mxp_axu_apb_only	MXP AXI utility bus access by APB only  <b>0b0</b> MXP AXU accessible via CHI/AXI in addition to APB  <b>0b1</b> MXP AXU accessible only via APB	RW	0b0
[2]	dmc_apb_only	DMC AXI utility bus access by APB only  <b>0b0</b> DMC AXU accessible via CHI/AXI in addition to APB  <b>0b1</b> DMC AXU accessible only via APB	RW	0b0
[1]	dsu_apb_only	DSU AXI utility bus access by APB only  <b>0b0</b> DSU AXU accessible via CHI/AXI in addition to APB  <b>0b1</b> DSU AXU accessible only via APB	RW	0b0
[0]	cmn_apb_only	CMN config access by APB only  <b>0b0</b> CMN config accessible via CHI/AXI in addition to APB  <b>0b1</b> CMN config accessible only via APB	RW	0b0

### 8.3.1.6 por\_axu\_control

Functions as the CMN AXU interface control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_apb\_rcr.apb

##### Secure group override

por\_apb\_scr.apb

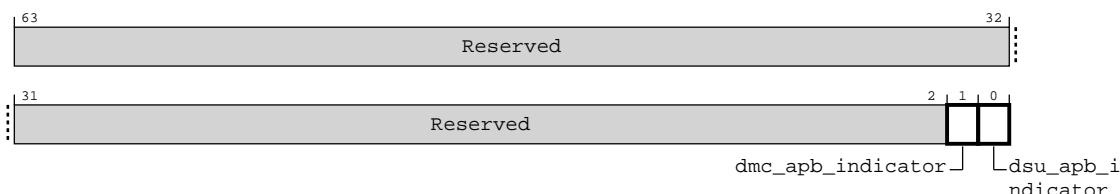
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_apb\_scr.apb bit is set, Secure accesses to this register are permitted. If both the por\_apb\_scr.apb bit and por\_apb\_rcr.apb 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 8-14: por\_axu\_control**



**Table 8-12: por\_axu\_control attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[1]	dmc_apb_indicator	<p>DMC AxPROT[0] access indicator enable</p> <p><b>0b1</b> AxPROT[0] will carry source indicator</p> <p><b>0</b> from CHI</p> <p><b>1</b> from APB</p> <p><b>0b0</b> AxPROT[0] behaves as default</p>	RW	0b0
[0]	dsu_apb_indicator	<p>DSU AxPROT[0] access indicator enable</p> <p><b>0b1</b> AxPROT[0] will carry source indicator</p> <p><b>0</b> from CHI</p> <p><b>1</b> from APB</p> <p><b>0b0</b> AxPROT[0] behaves as default</p>	RW	0b0

### 8.3.2 CCG HA register summary

The following table describes the registers for the relevant component.

Table 8-13: por\_ccg\_ha\_cfg register summary

Offset	Name	Type	Description
0x0	por_ccg_ha_node_info	RO	<a href="#">por_ccg_ha_node_info</a>
0x8	por_ccg_ha_id	RW	<a href="#">por_ccg_ha_id</a>
0x80	por_ccg_ha_child_info	RO	<a href="#">por_ccg_ha_child_info</a>
0xA00	por_ccg_ha_cfg_ctl	RW	<a href="#">por_ccg_ha_cfg_ctl</a>
0xA08	por_ccg_ha_aux_ctl	RW	<a href="#">por_ccg_ha_aux_ctl</a>
0xA10	por_ccg_ha_mpam_control_link0	RW	<a href="#">por_ccg_ha_mpam_control_link0</a>
0xA18	por_ccg_ha_mpam_control_link1	RW	<a href="#">por_ccg_ha_mpam_control_link1</a>
0xA20	por_ccg_ha_mpam_control_link2	RW	<a href="#">por_ccg_ha_mpam_control_link2</a>
0xA50 : 0xA60	por_ccg_ha_link0-2_cache_id_ctl	RW	<a href="#">por_ccg_ha_link0-2_cache_id_ctl</a>
0x980	por_ccg_ha_scr	RW	<a href="#">por_ccg_ha_scr</a>
0x988	por_ccg_ha_rcr	RW	<a href="#">por_ccg_ha_rcr</a>
0x900	por_ccg_ha_unit_info	RO	<a href="#">por_ccg_ha_unit_info</a>
0x908	por_ccg_ha_unit_info2	RO	<a href="#">por_ccg_ha_unit_info2</a>
0x910	por_ccg_ha_unit_info3	RO	<a href="#">por_ccg_ha_unit_info3</a>
0x1C00	por_ccg_ha_agentid_to_linkid_reg0	RW	<a href="#">por_ccg_ha_agentid_to_linkid_reg0</a>
0x1CF8	por_ccg_ha_agentid_to_linkid_val	RW	<a href="#">por_ccg_ha_agentid_to_linkid_val</a>

Offset	Name	Type	Description
0xC00 : 0xCF8	por_ccg_ha_rnf_exp_raid_to_lidid_reg_0-31	RW	<a href="#">por_ccg_ha_rnf_exp_raid_to_lidid_reg_0-31</a>
0xD00 : 0xDF8	por_ccg_ha_link1_rnf_exp_raid_to_lidid_reg_0-31	RW	<a href="#">por_ccg_ha_link1_rnf_exp_raid_to_lidid_reg_0-31</a>
0xE00 : 0xEF8	por_ccg_ha_link2_rnf_exp_raid_to_lidid_reg_0-31	RW	<a href="#">por_ccg_ha_link2_rnf_exp_raid_to_lidid_reg_0-31</a>
0xD900	por_ccg_ha_pmu_event_sel	RW	<a href="#">por_ccg_ha_pmu_event_sel</a>
0x1900	por_ccg_ha_cxprtl_link0_ctl	RW	<a href="#">por_ccg_ha_cxprtl_link0_ctl</a>
0x1908	por_ccg_ha_cxprtl_link0_status	RO	<a href="#">por_ccg_ha_cxprtl_link0_status</a>
0x1910	por_ccg_ha_cxprtl_link1_ctl	RW	<a href="#">por_ccg_ha_cxprtl_link1_ctl</a>
0x1918	por_ccg_ha_cxprtl_link1_status	RO	<a href="#">por_ccg_ha_cxprtl_link1_status</a>
0x1920	por_ccg_ha_cxprtl_link2_ctl	RW	<a href="#">por_ccg_ha_cxprtl_link2_ctl</a>
0x1928	por_ccg_ha_cxprtl_link2_status	RO	<a href="#">por_ccg_ha_cxprtl_link2_status</a>
0xA30	por_ccg_ha_datasource_ctl_link0	RW	<a href="#">por_ccg_ha_datasource_ctl_link0</a>
0xA38	por_ccg_ha_datasource_ctl_link1	RW	<a href="#">por_ccg_ha_datasource_ctl_link1</a>
0xA40	por_ccg_ha_datasource_ctl_link2	RW	<a href="#">por_ccg_ha_datasource_ctl_link2</a>

### 8.3.2.1 por\_ccg\_ha\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

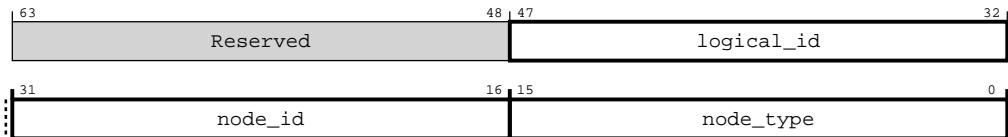
##### Usage constraints

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 8-15: por\_ccg\_ha\_node\_info**



**Table 8-14: por\_ccg\_ha\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0104

### 8.3.2.2 por\_ccg\_ha\_id

Contains the CCIX-assigned HAID.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x8

##### Type

RW

##### Reset value

See individual bit resets

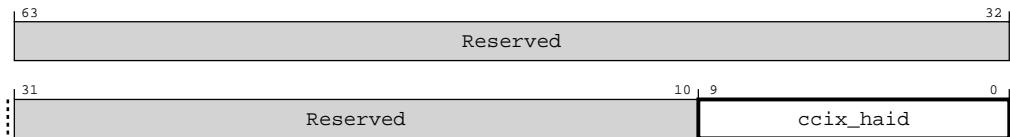
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-16: por\_ccg\_ha\_id**



**Table 8-15: por\_ccg\_ha\_id attributes**

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	
[9:0]	ccix_haid	CCIX HAID	RW	0x0

### 8.3.2.3 por\_ccg\_ha\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

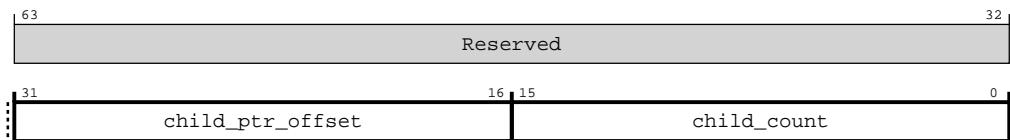
##### Usage constraints

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 8-17: por\_ccg\_ha\_child\_info**



**Table 8-16: por\_ccg\_ha\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0x0

### 8.3.2.4 por\_ccg\_ha\_cfg\_ctl

Functions as the configuration control register. Specifies the current mode.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ha\_scr.cfg\_ctl

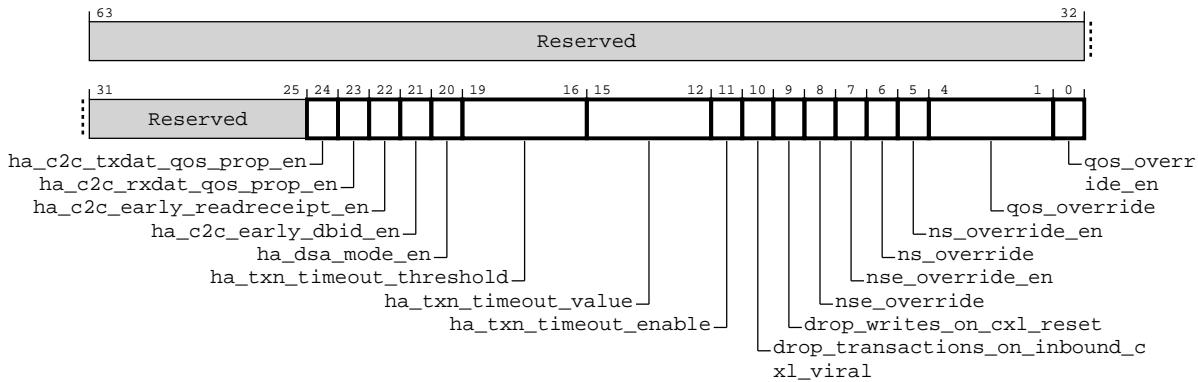
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.cfg\_ctl bit and por\_ccg\_ha\_rcr.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 8-18: por\_ccg\_ha\_cfg\_ctl**



**Table 8-17: por\_ccg\_ha\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	
[24]	ha_cc2c_txdat_qos_prop_en	For C2C, propagate QOS from CHI-C2C to CHI-DAT port. If disabled, reflect request value	RW	0b0
[23]	ha_cc2c_rxdat_qos_prop_en	For C2C, propagate QOS from CHI-DAT port to CHI-C2C. If disabled, QOS=0 on C2C	RW	0b0
[22]	ha_cc2c_early_readreceipt_en	C2C, for ordered read requests, HA enables the early-readreceipt flow	RW	0b0
[21]	ha_cc2c_early_dbid_en	C2C, for 3-hop flows Enables the early-dbld flow	RW	0b0
[20]	ha_dsa_mode_en	Enables DSA mode, ReadNoSnp & WriteNoSnp requests are sent to SN	RW	0b0
[19:16]	ha_txn_timeout_threshold	The number of transaction timeouts, which would trigger UE	RW	0x4
[15:12]	ha_txn_timeout_value	this field allows system software to modify the Transaction Timeout Value for CCG HA  <b>Transactions</b> 0b0000 50us to 10ms 0b0001 50us to 100us 0b0010 1ms to 10ms 0b0101 16ms to 55ms 0b0110 65ms to 210ms 0b1001 260ms to 900ms 0b1010 1s to 3.5s 0b1101 4s to 13s 0b1110 17s to 64s others Reserved	RW	0x0
[11]	ha_txn_timeout_enable	When Set this bit enables CCG HA Transaction Timeout mechanism.	RW	0b0
[10]	drop_transactions_on_inbound_cxl_viral	When set, write/read requests over CCL will be dropped during Inbound CXL Viral. Propagate by default	RW	0b0
[9]	drop_writes_on_cxl_reset	When set, write requests over CCL will be dropped during CXL Reset. Propagate by default	RW	0b0
[8]	nse_override	NSE override value	RW	0b0

Bits	Name	Description	Type	Reset
[7]	nse_override_en	NSE override en When set, NSE(Non-Secure) value on CHI side is driven from NSE override value in this register	RW	0b0
[6]	ns_override	NS override value	RW	0b1
[5]	ns_override_en	NS override en When set, NS(Non-Secure) value on CHI side is driven from NS override value in this register	RW	0b0
[4:1]	qos_override	QoS override value	RW	0b0
[0]	qos_override_en	QoS override en When set, QoS value on CHI side is driven from QoS override value in this register	RW	0b0

### 8.3.2.5 por\_ccg\_ha\_aux\_ctl

Functions as the auxiliary control register for CXHA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

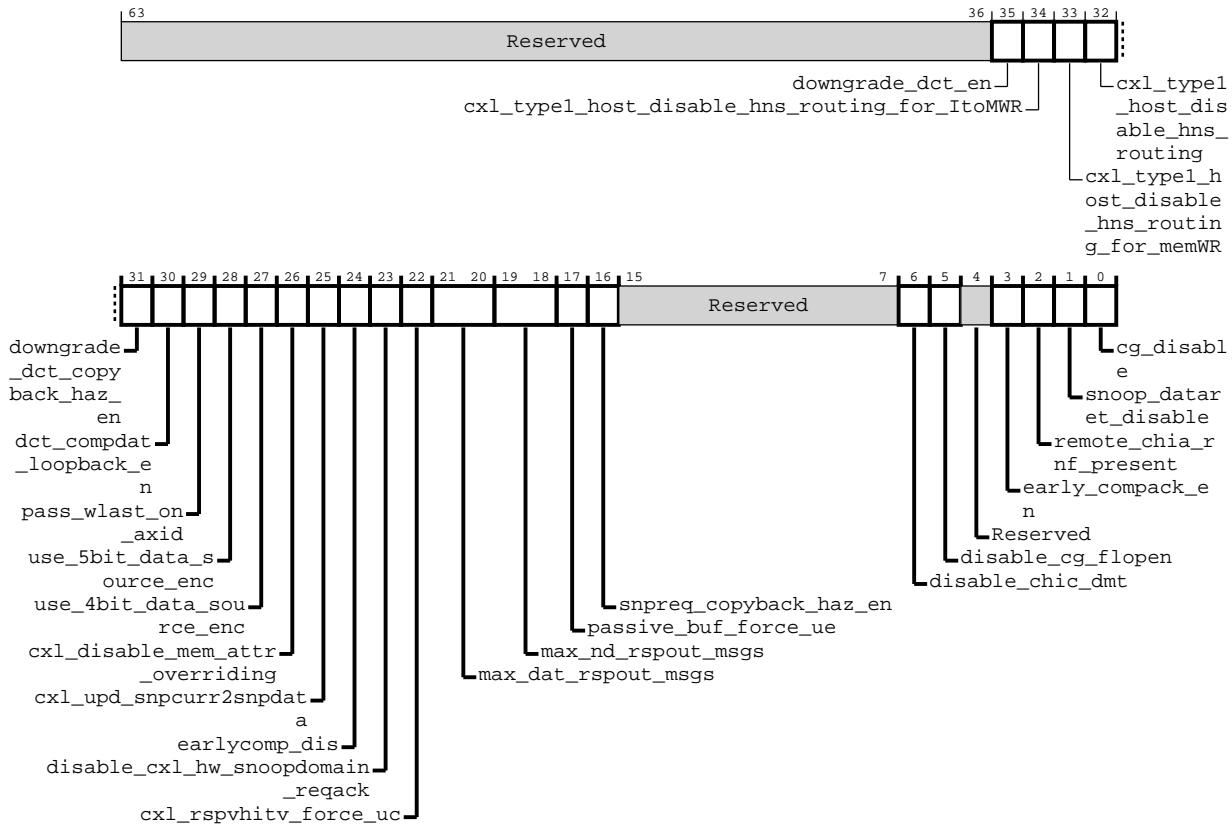
#### Usage constraints

This register is owned in the Root space and is 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 8-19: por\_ccg\_ha\_aux\_ctl**



**Table 8-18: por\_ccg\_ha\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35]	downgrade_dct_en	When set, Downgrades the Forward Snoop opcodes to the non-forward Snoops.	RW	0b0
[34]	cxl_type1_host_disable_hns_routing_for_ItoMWR	When set, HA disable sending CXL ItoMWR transactions(LBT bound) to the LCN. ItoMWR transactions (LBT bound) are steered to RA.	RW	0b0
[33]	cxl_type1_host_disable_hns_routing_for_memWR	When set, HA disable sending CXL memWR transactions(LBT bound) to the LCN. MemWR transactions (LBT bound) are steered to RA.	RW	0b1
[32]	cxl_type1_host_disable_hns_routing	When set, HA bypasses HNS for LBT bound transactions and directly steers LBT bound transactions to RA	RW	0b0
[31]	downgrade_dct_copyback_haz_en	When set, Downgrades the Forward Snoop opcodes to the non-forward Snoops during the Copyback hazard scenarios (data is sent to HNF only). When Reset, Data is sent to RN & HN depending on the RetToSrc and the Copyback opcode.	RW	0b0
[30]	dct_commdat_loopback_en	When set, enables COMPDAT loopback from HA TXDAT to HA RXDAT. This scenario arises when snoopee and requestor are both in the Remote (HA sends COMPDAT to itself)	RW	0b0

Bits	Name	Description	Type	Reset
[29]	pass_wlast_on_axid	When set, forces last txn indicator received from remote chip to be passed as the MSB of AWID for CCG_RNI instance (if present)	RW	0b1
[28]	use_5bit_data_source_enc	When set, forces use of 5-bit Data Source Encoding (CHI-F)	RW	0b0
[27]	use_4bit_data_source_enc	When set, forces use of 4-bit Data Source Encoding	RW	0b0
[26]	cxl_disable_mem_attr_overriding	When set, Disables the static memory attributes on the CHI reqs and derives from RNSAM configuration based on HTG/non-hashed regions	RW	0b0
[25]	cxl_upd_snpcurr2snpdata	When set, updates SnpCurr to SnpData on CXL.Cache	RW	0b0
[24]	earlycomp_dis	When set, disables sending early completions for requests. This is applicable to CXL.Cache (writebacks that come with Data) - D2H_DIRTYEVICT	RW	0b1
[23]	disable_cxl_hw_snoopdomain_reqack	When set, disables CXL Hardware based Snoop domain handshake	RW	0b0
[22]	cxl_rspvhitv_force_uc	When set, forces UC response state for Snoop response on CHI when D2H RspVHitV response is received on CXL	RW	0b0
[21:20]	max_dat_rspout_msgs	<p>Used to configure the maximum number of response data messages (Read Data) presented to CCLA's packing logic.</p> <p><b>0b00</b> one message</p> <p><b>0b01</b> two messages</p> <p><b>0b10</b> three messages</p> <p><b>0b11</b> four messages</p> <p><b>Note</b> The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)</p>	RW	0b11
[19:18]	max_nd_rspout_msgs	<p>Used to configure the maximum number of non-data response messages (Completions/GO/WritePulls) presented to CCLA's packing logic.</p> <p><b>0b00</b> one message</p> <p><b>0b01</b> two messages</p> <p><b>0b10</b> three messages</p> <p><b>0b11</b> four messages</p> <p><b>Note</b> The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)</p>	RW	0b11
[17]	passive_buf_force_ue	When set, forces all DE's originating in Passive Buffer to be reported as UEs	RW	0b0

Bits	Name	Description	Type	Reset
[16]	snpreq_copyback_haz_en	When set enables Snoop-CopyBack hazarding at incoming Snoop Request pipe	RW	0b0
[15:7]	Reserved	Reserved	RO	
[6]	disable_chic_dmt	When set disables CHI-C style DMT	RW	0b0
[5]	disable_cg_flopen	Disables enhanced flop enable control for dynamic power savings	RW	0b0
[4]	Reserved	Reserved	RO	
[3]	early_compack_en	Early CompAck enable; enables sending early CompAck on CCIX for requests that require CompAck	RW	0b1
[2]	remote_chia_rnf_present	Indicates existence of CHIA RN-F in system; HA uses this indication to send SnpToS or SnpToSC  <b>0b0</b> HA converts SnpShared, SnpClean, and SnpNotSharedDirty to SnpToSC  <b>0b1</b> HA converts SnpShared, SnpClean, and SnpNotSharedDirty to SnpToS	RW	0b0
[1]	snoop_dataret_disable	Disables setting data return for CCIX snoop requests for all CHI snoop opcodes	RW	0b0
[0]	cg_disable	Disables clock gating when set	RW	0b0

### 8.3.2.6 por\_ccg\_ha\_mpam\_control\_link0

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA10

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.mpam\_ctl

##### Secure group override

por\_ccg\_ha\_scr.mpam\_ctl

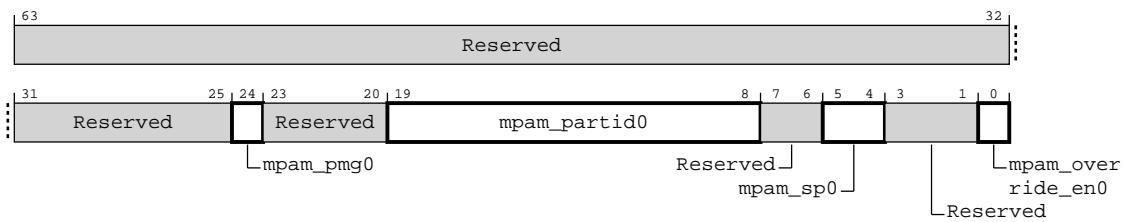
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.mpam\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.mpam\_ctl bit and por\_ccg\_ha\_rcr.mpam\_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 8-20: por\_ccg\_ha\_mpam\_control\_link0**



**Table 8-19: por\_ccg\_ha\_mpam\_control\_link0 attributes**

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	
[24]	mpam_pmg0	MPAM_PMG value	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	mpam_partid0	MPAM_PARTID value	RW	0b0
[7:6]	Reserved	Reserved	RO	
[5:4]	mpam_sp0	MPAM_SP value	RW	0b00
[3:1]	Reserved	Reserved	RO	
[0]	mpam_override_en0	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	0b0

### 8.3.2.7 por\_ccg\_ha\_mpam\_control\_link1

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link1

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

## Address offset

0xA18

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_ccg\_ha\_rcr.mpam\_ctl

## Secure group override

por\_ccg\_ha\_scr.mpam\_ctl

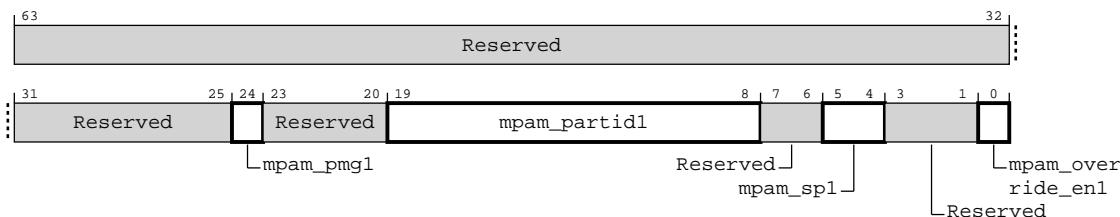
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.mpam\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.mpam\_ctl bit and por\_ccg\_ha\_rcr.mpam\_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 8-21: por\_ccg\_ha\_mpam\_control\_link1**



**Table 8-20: por\_ccg\_ha\_mpam\_control\_link1 attributes**

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	
[24]	mpam_pmg1	MPAM_PMG value	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	mpam_partid1	MPAM_PARTID value	RW	0b0
[7:6]	Reserved	Reserved	RO	
[5:4]	mpam_sp1	MPAM_SP value	RW	0b00
[3:1]	Reserved	Reserved	RO	
[0]	mpam_override_en1	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	0b0

### 8.3.2.8 por\_ccg\_ha\_mpam\_control\_link2

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link2

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA20

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.mpam\_ctl

##### Secure group override

por\_ccg\_ha\_scr.mpam\_ctl

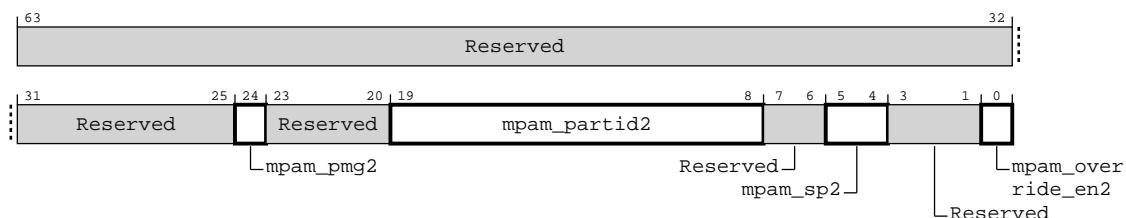
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.mpam\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.mpam\_ctl bit and por\_ccg\_ha\_rcr.mpam\_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 8-22: por\_ccg\_ha\_mpam\_control\_link2**



**Table 8-21: por\_ccg\_ha\_mpam\_control\_link2 attributes**

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	
[24]	mpam_pmg2	MPAM_PMG value	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	mpam_partid2	MPAM_PARTID value	RW	0b0
[7:6]	Reserved	Reserved	RO	
[5:4]	mpam_sp2	MPAM_SP value	RW	0b00
[3:1]	Reserved	Reserved	RO	
[0]	mpam_override_en2	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	0b0

### 8.3.2.9 por\_ccg\_ha\_link0-2\_cache\_id\_ctl

There are 3 iterations of this register. The index ranges from 0 to 2. Link-0 configuration to determine the LDID from Incoming RAID

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA50 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

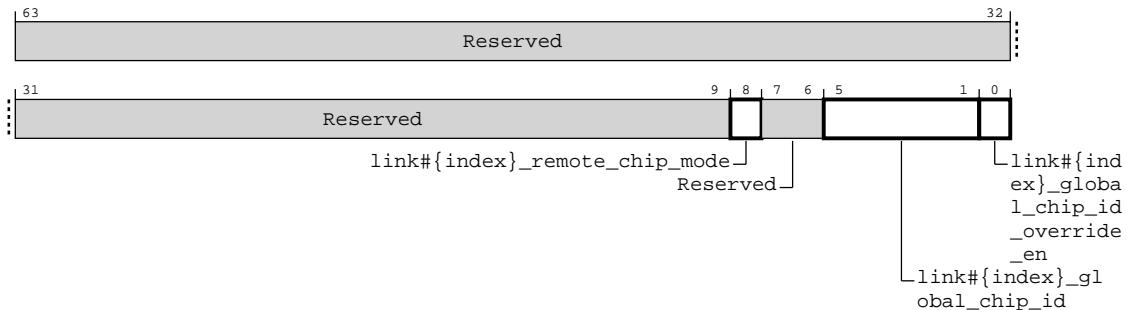
##### Usage constraints

This register is owned in the Root space and is 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 8-23: por\_ccg\_ha\_link0-2\_cache\_id\_ctl**



**Table 8-22: por\_ccg\_ha\_link0-2\_cache\_id\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8]	link#{index}_remote_chip_mode	Defines the remote chip as Hierarchical(HN-S) or Flat(HN-F) 0b0 - Hierarchical system 0b1 - Flat system	RW	0b0
[7:6]	Reserved	Reserved	RO	
[5:1]	link#{index}_global_chip_id	global chip id value	RW	0b0
[0]	link#{index}_global_chip_id_override_en	override global_chip_id on the incoming requests from non-caching agent	RW	0b0

### 8.3.2.10 por\_ccg\_ha\_scr

Secure Control Registers for predefined groups of root registers.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

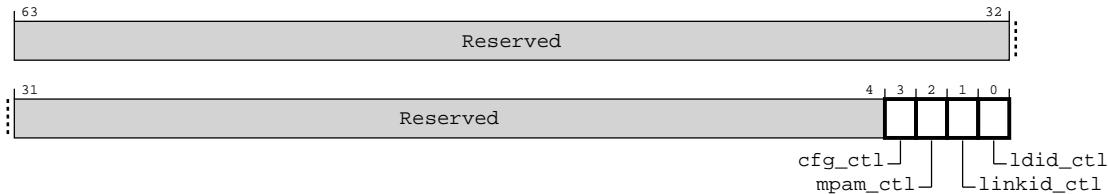
##### Usage constraints

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 8-24: por\_ccg\_ha\_scr**



**Table 8-23: por\_ccg\_ha\_scr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	cfg_ctl	Secure Control Register for root HA config control registers	RW	0b0
[2]	mpam_ctl	Secure Control Register for root HA MPAM override registers	RW	0b0
[1]	linkid_ctl	Secure Control Register for root HA Link ID registers	RW	0b0
[0]	ldid_ctl	Secure Control Register for root HA LDID registers	RW	0b0

### 8.3.2.11 por\_ccg\_ha\_rcr

Root Control Registers for predefined groups of root registers.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

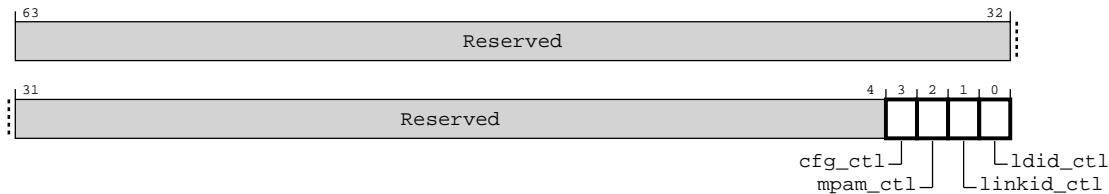
##### Usage constraints

This register is owned in the Root space and is 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 8-25: por\_ccg\_ha\_rccr**



**Table 8-24: por\_ccg\_ha\_rccr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	cfg_ctl	Root Control Register for root HA config control registers	RW	0b0
[2]	mpam_ctl	Root Control Register for root HA MPAM override registers	RW	0b0
[1]	linkid_ctl	Root Control Register for root HA Link ID registers	RW	0b0
[0]	ldid_ctl	Root Control Register for root HA LDID registers	RW	0b0

### 8.3.2.12 por\_ccg\_ha\_unit\_info

Provides component identification information for CXHA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

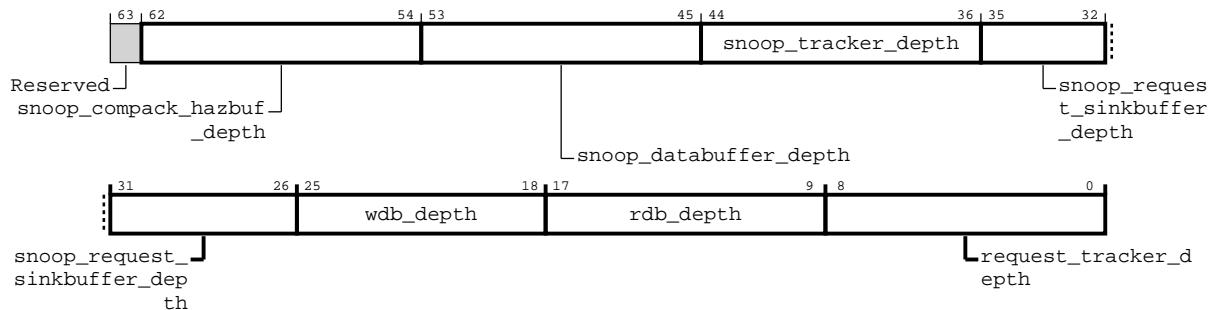
##### Usage constraints

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 8-26: por\_ccg\_ha\_unit\_info**



**Table 8-25: por\_ccg\_ha\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:54]	snoop_compack_hazbuf_depth	Depth of CompAck snoop hazard buffer	RO	0x10
[53:45]	snoop_databuffer_depth	Depth of snoop data buffer	RO	0x20
[44:36]	snoop_tracker_depth	Depth of snoop tracker; number of outstanding SNP requests on CCIX	RO	0x100
[35:26]	snoop_request_sinkbuffer_depth	Depth of snoop request sink buffer; number of CHI SNP requests that can be sunk by CXHA	RO	Configuration dependent
[25:18]	wdb_depth	Depth of write data buffer	RO	Configuration dependent
[17:9]	rdb_depth	Depth of read data buffer	RO	Configuration dependent
[8:0]	request_tracker_depth	Depth of request tracker	RO	Configuration dependent

### 8.3.2.13 por\_ccg\_ha\_unit\_info2

Provides additional component identification information for CXHA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x908

##### Type

RO

### Reset value

See individual bit resets

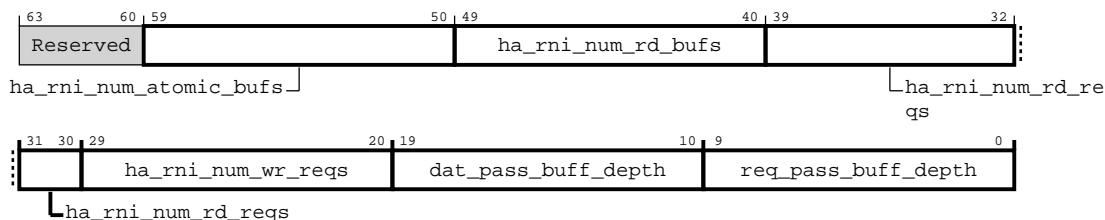
### Usage constraints

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 8-27: por\_ccg\_ha\_unit\_info2**



**Table 8-26: por\_ccg\_ha\_unit\_info2 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:50]	ha_rni_num_atomic_bufs	Number of HA_RNI atomic data buffers	RO	Configuration dependent
[49:40]	ha_rni_num_rd_bufs	Number of HA_RNI read data buffers	RO	Configuration dependent
[39:30]	ha_rni_num_rd_reqs	Number of HA_RNI outstanding read requests	RO	Configuration dependent
[29:20]	ha_rni_num_wr_reqs	Number of HA_RNI outstanding write requests	RO	Configuration dependent
[19:10]	dat_pass_buff_depth	Depth of DAT Passive Buffer	RO	Configuration dependent
[9:0]	req_pass_buff_depth	Depth of REQ Passive Buffer	RO	Configuration dependent

### 8.3.2.14 por\_ccg\_ha\_unit\_info3

Provides additional component identification information for CXHA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x910

##### Type

RO

## Reset value

See individual bit resets

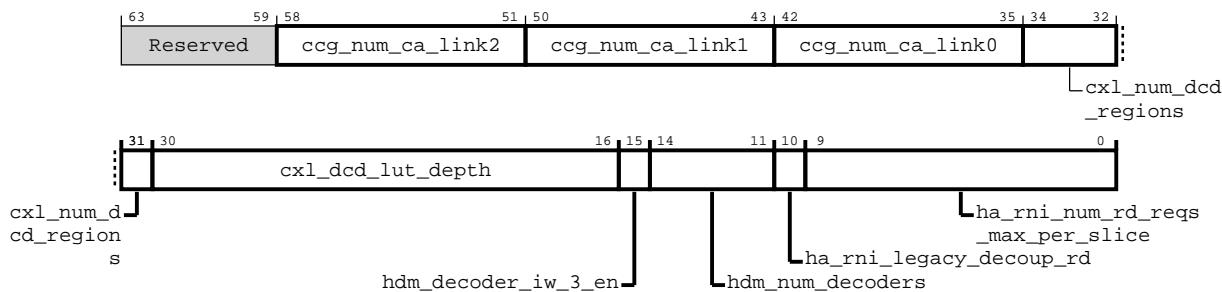
## Usage constraints

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 8-28: por\_ccg\_ha\_unit\_info3**



**Table 8-27: por\_ccg\_ha\_unit\_info3 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:51]	ccg_num_ca_link2	Number of Caching agents on the remote chip connected through link2	RO	0x64
[50:43]	ccg_num_ca_link1	Number of Caching agents on the remote chip connected through link1	RO	0x64
[42:35]	ccg_num_ca_link0	Number of Caching agents on the remote chip connected through link0	RO	0x64
[34:31]	cxl_num_dcd_regions	Number of CXL Dynamic Capacity Device Regions	RO	0x0
[30:16]	cxl_dcd_lut_depth	Depth of CXL Dynamic Capacity Device LUT	RO	0x1024
[15]	hdm_decoder_iw_3_en	HDM decoder support for {3, 6, 12} interleave way enabled	RO	0x1
[14:11]	hdm_num_decoders	Number of HDM decoders	RO	0x1
[10]	ha_rni_legacy_decoup_rd	Legacy decoupled read mode, no read burst propagation	RO	0x1
[9:0]	ha_rni_num_rd_reqs_max_per_slice	Number of HA_RNI read request entries per slice	RO	0x40

### 8.3.2.15 por\_ccg\_ha\_agentid\_to\_linkid\_reg0

Specifies the mapping of Baseid to Link ID for Agent IDs 0 to 7.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x1C00

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ha\_rcr.linkid\_ctl

### Secure group override

por\_ccg\_ha\_scr.linkid\_ctl

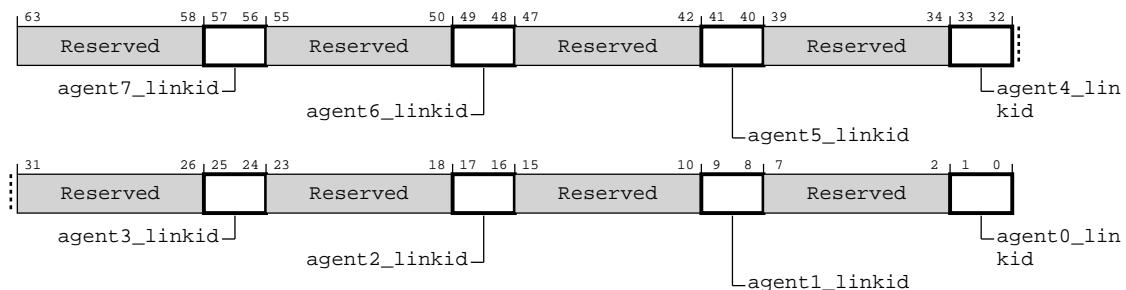
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.linkid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.linkid\_ctl bit and por\_ccg\_ha\_rcr.linkid\_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 8-29: por\_ccg\_ha\_agentid\_to\_linkid\_reg0**



**Table 8-28: por\_ccg\_ha\_agentid\_to\_linkid\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	
[57:56]	agent7_linkid	Specifies Link ID for baseid-7	RW	0x0
[55:50]	Reserved	Reserved	RO	
[49:48]	agent6_linkid	Specifies Link ID for baseid-6	RW	0x0

Bits	Name	Description	Type	Reset
[47:42]	Reserved	Reserved	RO	
[41:40]	agent5_linkid	Specifies Link ID for baseid-5	RW	0x0
[39:34]	Reserved	Reserved	RO	
[33:32]	agent4_linkid	Specifies Link ID for baseid-4	RW	0x0
[31:26]	Reserved	Reserved	RO	
[25:24]	agent3_linkid	Specifies Link ID for baseid-3	RW	0x0
[23:18]	Reserved	Reserved	RO	
[17:16]	agent2_linkid	Specifies Link ID for baseid-2	RW	0x0
[15:10]	Reserved	Reserved	RO	
[9:8]	agent1_linkid	Specifies Link ID for baseid-1	RW	0x0
[7:2]	Reserved	Reserved	RO	
[1:0]	agent0_linkid	Specifies Link ID for baseid-0	RW	0x0

### 8.3.2.16 por\_ccg\_ha\_agentid\_to\_linkid\_val

Specifies which Agent ID to Link ID mappings are valid.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1CF8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.linkid\_ctl

##### Secure group override

por\_ccg\_ha\_scr.linkid\_ctl

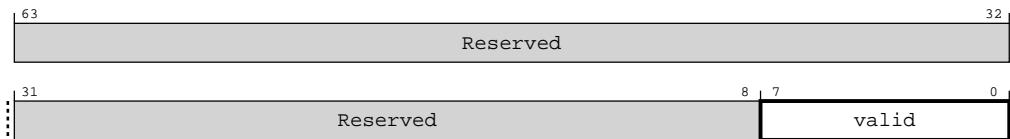
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.linkid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.linkid\_ctl bit and por\_ccg\_ha\_rcr.linkid\_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 8-30: por\_ccg\_ha\_agentid\_to\_linkid\_val**



**Table 8-29: por\_ccg\_ha\_agentid\_to\_linkid\_val attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:0]	valid	Specifies whether the Link ID is valid; bit number corresponds to logical Agent ID number (from 0 to 7)	RW	0x0

### 8.3.2.17 por\_ccg\_ha\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of Link0 Expanded RAID to RN-F LDID for Expanded RAIDs #{{index}4} to #{{index}4+3}

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xC00 + #{{8\*index}}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ha\_rcr.ldid\_ctl

### Secure group override

por\_ccg\_ha\_scr.ldid\_ctl

### Usage constraints

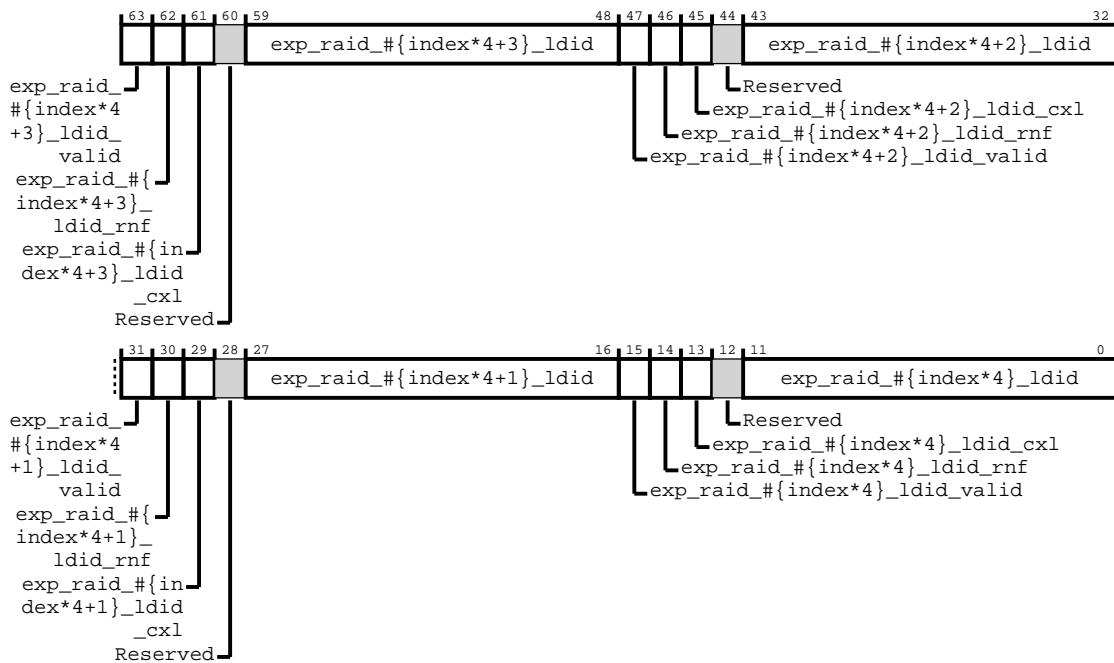
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.ldid\_ctl bit and por\_ccg\_ha\_rcr.ldid\_ctl

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 8-31: por\_ccg\_ha\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31**



**Table 8-30: por\_ccg\_ha\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	exp_raid_{index*4+3}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+3} is valid	RW	0b0
[62]	exp_raid_{index*4+3}_ldid_rnf	Specifies if Expanded RAID #{index*4+3} is RN-F	RW	0b0
[61]	exp_raid_{index*4+3}_ldid_cxl	Specifies if Expanded RAID #{index*4+3} is CXL agent	RW	0b0
[60]	Reserved	Reserved	RO	
[59:48]	exp_raid_{index*4+3}_ldid	Specifies the LDID for Expanded RAID #{index*4+3}	RW	0x0
[47]	exp_raid_{index*4+2}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+2} is valid	RW	0b0
[46]	exp_raid_{index*4+2}_ldid_rnf	Specifies if Expanded RAID #{index*4+2} is RN-F	RW	0b0
[45]	exp_raid_{index*4+2}_ldid_cxl	Specifies if Expanded RAID #{index*4+2} is CXL agent	RW	0b0
[44]	Reserved	Reserved	RO	
[43:32]	exp_raid_{index*4+2}_ldid	Specifies the LDID for Expanded RAID #{index*4+2}	RW	0x0
[31]	exp_raid_{index*4+1}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+1} is valid	RW	0b0
[30]	exp_raid_{index*4+1}_ldid_rnf	Specifies if Expanded RAID #{index*4+1} is RN-F	RW	0b0
[29]	exp_raid_{index*4+1}_ldid_cxl	Specifies if Expanded RAID #{index*4+1} is CXL agent	RW	0b0
[28]	Reserved	Reserved	RO	

<b>Bits</b>	<b>Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
[27:16]	exp_raid_#{index*4+1}_ldid	Specifies the LDID for Expanded RAID #{index*4+1}	RW	0x0
[15]	exp_raid_#{index*4}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4} is valid	RW	0b0
[14]	exp_raid_#{index*4}_ldid_rnf	Specifies if Expanded RAID #{index*4} is RN-F	RW	0b0
[13]	exp_raid_#{index*4}_ldid_cxl	Specifies if Expanded RAID #{index*4} is CXL agent	RW	0b0
[12]	Reserved	Reserved	RO	
[11:0]	exp_raid_#{index*4}_ldid	Specifies the LDID for Expanded RAID #{index*4}	RW	0x0

### 8.3.2.18 por\_ccg\_ha\_link1\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of Link0 Expanded RAID to RN-F LDID for Expanded RAIDs #{index4} to #{index4+3}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD00 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.ldid\_ctl

##### Secure group override

por\_ccg\_ha\_scr.ldid\_ctl

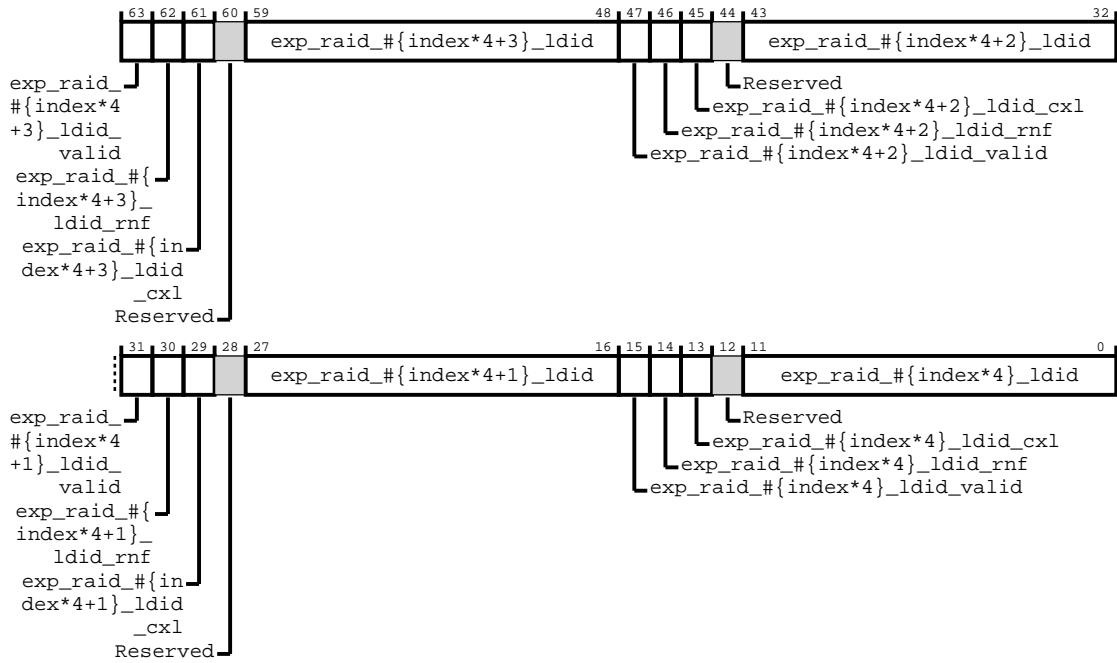
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.ldid\_ctl bit and por\_ccg\_ha\_rcr.ldid\_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 8-32: por\_ccg\_ha\_link1\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31**



**Table 8-31: por\_ccg\_ha\_link1\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	exp_raid_#{index*4+3}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+3} is valid	RW	0b0
[62]	exp_raid_#{index*4+3}_ldid_rnf	Specifies if Expanded RAID #{index*4+3} is RN-F	RW	0b0
[61]	exp_raid_#{index*4+3}_ldid_cxl	Specifies if Expanded RAID #{index*4+3} is CXL agent	RW	0b0
[60]	Reserved	Reserved	RO	
[59:48]	exp_raid_#{index*4+3}_ldid	Specifies the LDID for Expanded RAID #{index*4+3}	RW	0x0
[47]	exp_raid_#{index*4+2}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+2} is valid	RW	0b0
[46]	exp_raid_#{index*4+2}_ldid_rnf	Specifies if Expanded RAID #{index*4+2} is RN-F	RW	0b0
[45]	exp_raid_#{index*4+2}_ldid_cxl	Specifies if Expanded RAID #{index*4+2} is CXL agent	RW	0b0
[44]	Reserved	Reserved	RO	
[43:32]	exp_raid_#{index*4+2}_ldid	Specifies the LDID for Expanded RAID #{index*4+2}	RW	0x0
[31]	exp_raid_#{index*4+1}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+1} is valid	RW	0b0
[30]	exp_raid_#{index*4+1}_ldid_rnf	Specifies if Expanded RAID #{index*4+1} is RN-F	RW	0b0
[29]	exp_raid_#{index*4+1}_ldid_cxl	Specifies if Expanded RAID #{index*4+1} is CXL agent	RW	0b0
[28]	Reserved	Reserved	RO	
[27:16]	exp_raid_#{index*4+1}_ldid	Specifies the LDID for Expanded RAID #{index*4+1}	RW	0x0
[15]	exp_raid_#{index*4}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4} is valid	RW	0b0
[14]	exp_raid_#{index*4}_ldid_rnf	Specifies if Expanded RAID #{index*4} is RN-F	RW	0b0
[13]	exp_raid_#{index*4}_ldid_cxl	Specifies if Expanded RAID #{index*4} is CXL agent	RW	0b0
[12]	Reserved	Reserved	RO	
[11:0]	exp_raid_#{index*4}_ldid	Specifies the LDID for Expanded RAID #{index*4}	RW	0x0

### 8.3.2.19 por\_ccg\_ha\_link2\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of Link0 Expanded RAID to RN-F LDID for Expanded RAIDs #*{index4}* to #*{index4+3}*

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE00 + #*{8\*index}*

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ha\_rcr.ldid\_ctl

##### Secure group override

por\_ccg\_ha\_scr.ldid\_ctl

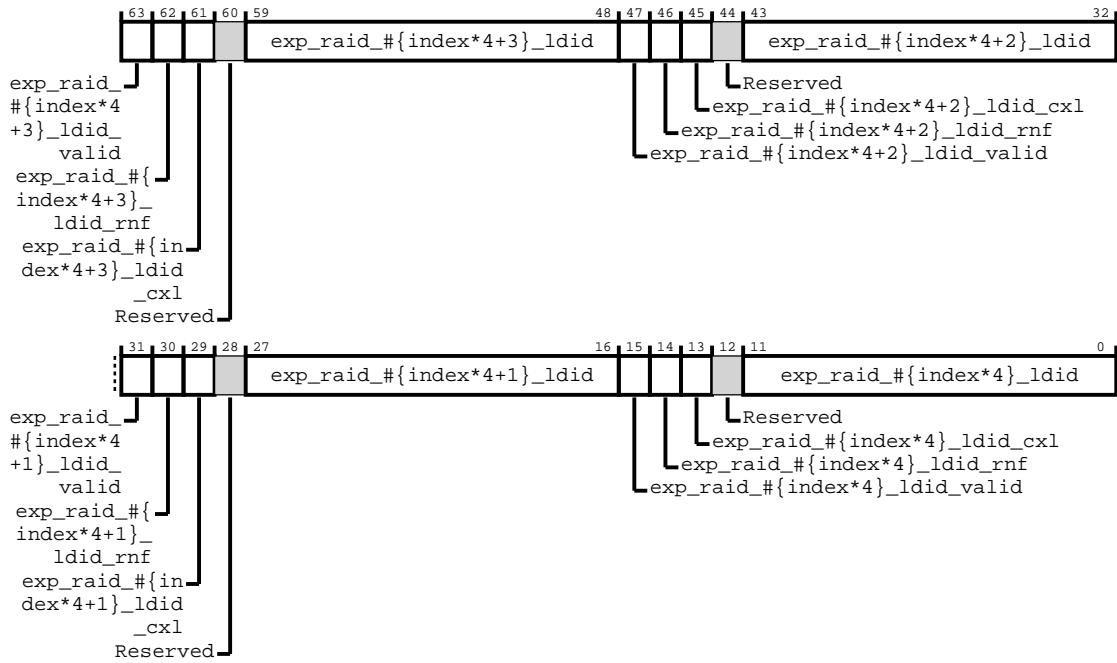
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ha\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ha\_scr.ldid\_ctl bit and por\_ccg\_ha\_rcr.ldid\_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 8-33: por\_ccg\_ha\_link2\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31**



**Table 8-32: por\_ccg\_ha\_link2\_rnf\_exp\_raid\_to\_ldid\_reg\_0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	<code>exp_raid_#{index*4+3}_ldid_valid</code>	Specifies if LDID for Expanded RAID #{index*4+3} is valid	RW	0b0
[62]	<code>exp_raid_#{index*4+3}_ldid_rnf</code>	Specifies if Expanded RAID #{index*4+3} is RN-F	RW	0b0
[61]	<code>exp_raid_#{index*4+3}_ldid_cxl</code>	Specifies if Expanded RAID #{index*4+3} is CXL agent	RW	0b0
[60]	Reserved	Reserved	RO	
[59:48]	<code>exp_raid_#{index*4+3}_ldid</code>	Specifies the LDID for Expanded RAID #{index*4+3}	RW	0x0
[47]	<code>exp_raid_#{index*4+2}_ldid_valid</code>	Specifies if LDID for Expanded RAID #{index*4+2} is valid	RW	0b0
[46]	<code>exp_raid_#{index*4+2}_ldid_rnf</code>	Specifies if Expanded RAID #{index*4+2} is RN-F	RW	0b0
[45]	<code>exp_raid_#{index*4+2}_ldid_cxl</code>	Specifies if Expanded RAID #{index*4+2} is CXL agent	RW	0b0
[44]	Reserved	Reserved	RO	
[43:32]	<code>exp_raid_#{index*4+2}_ldid</code>	Specifies the LDID for Expanded RAID #{index*4+2}	RW	0x0
[31]	<code>exp_raid_#{index*4+1}_ldid_valid</code>	Specifies if LDID for Expanded RAID #{index*4+1} is valid	RW	0b0
[30]	<code>exp_raid_#{index*4+1}_ldid_rnf</code>	Specifies if Expanded RAID #{index*4+1} is RN-F	RW	0b0
[29]	<code>exp_raid_#{index*4+1}_ldid_cxl</code>	Specifies if Expanded RAID #{index*4+1} is CXL agent	RW	0b0
[28]	Reserved	Reserved	RO	
[27:16]	<code>exp_raid_#{index*4+1}_ldid</code>	Specifies the LDID for Expanded RAID #{index*4+1}	RW	0x0
[15]	<code>exp_raid_#{index*4}_ldid_valid</code>	Specifies if LDID for Expanded RAID #{index*4} is valid	RW	0b0
[14]	<code>exp_raid_#{index*4}_ldid_rnf</code>	Specifies if Expanded RAID #{index*4} is RN-F	RW	0b0
[13]	<code>exp_raid_#{index*4}_ldid_cxl</code>	Specifies if Expanded RAID #{index*4} is CXL agent	RW	0b0
[12]	Reserved	Reserved	RO	
[11:0]	<code>exp_raid_#{index*4}_ldid</code>	Specifies the LDID for Expanded RAID #{index*4}	RW	0x0

### 8.3.2.20 por\_ccg\_ha\_pmu\_event\_sel

Specifies the PMU event to be counted as a 8-bit ID with the following encodings:

**0x00**

CXHA\_PMU\_EVENT\_NULL

**0x61**

CXHA\_PMU\_EVENT\_RDDATBYP

**0x62**

CXHA\_PMU\_EVENT\_CHIRSP0\_UP\_STALL

**0x63**

CXHA\_PMU\_EVENT\_CHIDAT\_UP\_STALL

**0x64**

CXHA\_PMU\_EVENT\_SNPPCRD\_LNK0\_STALL

**0x65**

CXHA\_PMU\_EVENT\_SNPPCRD\_LNK1\_STALL

**0x66**

CXHA\_PMU\_EVENT\_SNPPCRD\_LNK2\_STALL

**0x67**

CXHA\_PMU\_EVENT\_REQTRK\_OCC

**0x68**

CXHA\_PMU\_EVENT\_RDB\_OCC

**0x69**

CXHA\_PMU\_EVENT\_RDBBYP\_OCC

**0x6A**

CXHA\_PMU\_EVENT\_WDB\_OCC

**0x6B**

CXHA\_PMU\_EVENT\_SNPTRK\_OCC

**0x6C**

CXHA\_PMU\_EVENT\_SDB\_OCC

**0x6D**

CXHA\_PMU\_EVENT\_SNPHAZ\_OCC

**0x6E**

CXHA\_PMU\_EVENT\_REQTRK\_ALLOC

**0x6F**

CXHA\_PMU\_EVENT\_RDB\_ALLOC

**0x70**

CXHA\_PMU\_EVENT\_RDBBYP\_ALLOC

**0x71**

CXHA\_PMU\_EVENT\_WDB\_ALLOC

**0x72**

CXHA\_PMU\_EVENT\_SNPTRK\_ALLOC

**0x73**

CXHA\_PMU\_EVENT\_SDB\_ALLOC

**0x74**

CXHA\_PMU\_EVENT\_SNPHAZ\_ALLOC

**0x75**

CCHA\_PMU\_EVENT\_PB\_RHU\_REQ\_OCC

**0x76**

CCHA\_PMU\_EVENT\_PB\_RHU\_REQ\_ALLOC

**0x77**

CCHA\_PMU\_EVENT\_PB\_RHU\_PCIE\_REQ\_OCC

**0x78**

CCHA\_PMU\_EVENT\_PB\_RHU\_PCIE\_REQ\_ALLOC

**0x79**

CCHA\_PMU\_EVENT\_PB\_PCIE\_WR\_REQ\_OCC

**0x7A**

CCHA\_PMU\_EVENT\_PB\_PCIE\_WR\_REQ\_ALLOC

**0x7B**

CCHA\_PMU\_EVENT\_PB\_PCIE\_REG\_REQ\_OCC

**0x7C**

CCHA\_PMU\_EVENT\_PB\_PCIE\_REG\_REQ\_ALLOC

**0x7D**

CCHA\_PMU\_EVENT\_PB\_PCIE\_RSVD\_REQ\_OCC

**0x7E**

CCHA\_PMU\_EVENT\_PB\_PCIE\_RSVD\_REQ\_ALLOC

**0x7F**

CCHA\_PMU\_EVENT\_PB\_RHU\_DAT\_OCC

**0x80**

CCHA\_PMU\_EVENT\_PB\_RHU\_DAT\_ALLOC

**0x81**

CCHA\_PMU\_EVENT\_PB\_RHU\_PCIE\_DAT\_OCC

**0x82**

CCHA\_PMU\_EVENT\_PB\_RHU\_PCIE\_DAT\_ALLOC

**0x83**

CCHA\_PMU\_EVENT\_PB\_PCIE\_WR\_DAT\_OCC

**0x84**

CCHA\_PMU\_EVENT\_PB\_PCIE\_WR\_DAT\_ALLOC

**0x85**

CXHA\_PMU\_EVENT\_CHIRSP1\_UP\_STALL

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD900

### Type

RW

### Reset value

See individual bit resets

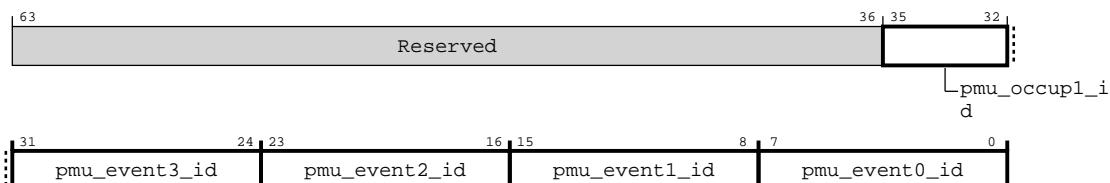
### Usage constraints

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 8-34: por\_ccg\_ha\_pmu\_event\_sel**



**Table 8-33: por\_ccg\_ha\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35:32]	pmu_occup1_id	CXHA PMU occupancy event selector ID	RW	0b0
[31:24]	pmu_event3_id	CXHA PMU Event 3 ID	RW	0b0
[23:16]	pmu_event2_id	CXHA PMU Event 2 ID	RW	0b0
[15:8]	pmu_event1_id	CXHA PMU Event 1 ID	RW	0b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	CXHA PMU Event 0 ID	RW	0b0

### 8.3.2.21 por\_ccg\_ha\_cxprtcl\_link0\_ctl

Functions as the CXHA CCIX Protocol Link 0 control register. Works with por\_ccg\_ha\_cxprtcl\_link0\_status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1900

##### Type

RW

##### Reset value

See individual bit resets

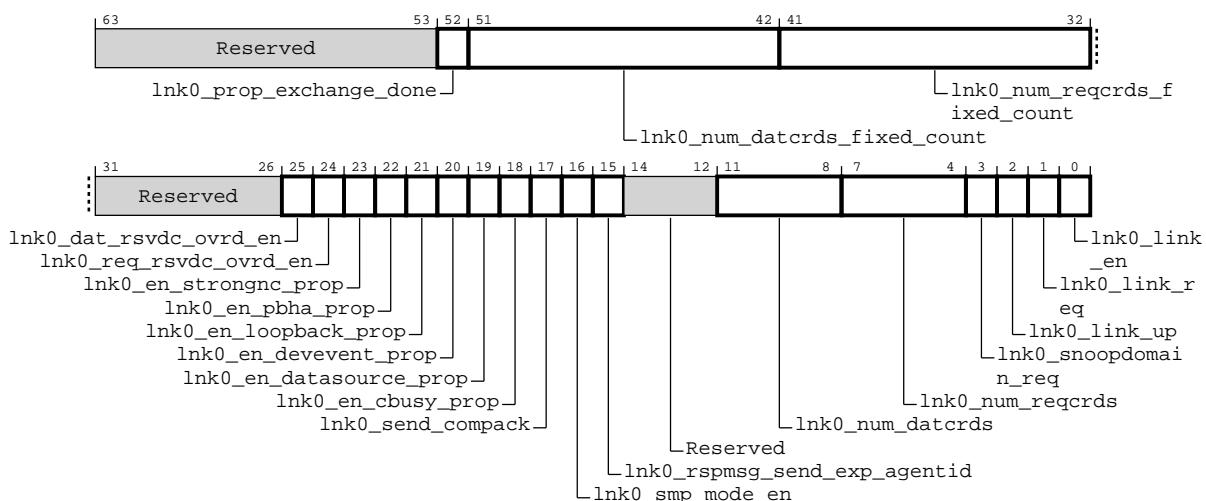
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-35: por\_ccg\_ha\_cxprtcl\_link0\_ctl**



**Table 8-34: por\_ccg\_ha\_cxprtcl\_link0\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52]	lnk0_prop_exchange_done	Status bbit to indicate property exchange is done	RW	0b0
[51:42]	lnk0_num_datcrds_fixed_count	Controls the number of CCIX data credits assigned to Link 0 If this field along with link1, link2 fixed credit count is zero then the hardware uses lnk0_num_datcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_datcrds_fixed_count should not exceed dat_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds dat_pass_buff_depth then it will cause fatal errors	RW	0x00
[41:32]	lnk0_num_reqcrds_fixed_count	Controls the number of CCIX request credits assigned to Link 0 If this field along with link1, link2 fixed credit count is zero then the hardware uses lnk0_num_reqcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_reqcrds_fixed_count should not exceed req_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds req_pass_buff_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-
[25]	lnk0_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 0.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[24]	lnk0_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 0.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[23]	lnk0_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 0.	RW	0b0
[22]	lnk0_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 0.	RW	0b0
[21]	lnk0_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 0.	RW	0b0
[20]	lnk0_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 0.	RW	0b1
[19]	lnk0_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 0.	RW	0b1
[18]	lnk0_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 0.	RW	0b1
[17]	lnk0_send_compack	When set, sends CompAck for CCIX Link 0.	RW	0b0
[16]	lnk0_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 0.	RW	0x0
[15]	lnk0_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 0	RW	0b0
[14:12]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[11:8]	lnk0_num_datcrds	<p>Controls the number of CCIX data credits assigned to Link 0</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[7:4]	lnk0_num_reqcrds	<p>Controls the number of CCIX request credits assigned to Link 0</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk0_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 0	RW	0b0
[2]	lnk0_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0

Bits	Name	Description	Type	Reset
[1]	lnk0_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk0_link_en	<p>Enables CCIX Link 0 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.2.22 por\_ccg\_ha\_cxprtcl\_link0\_status

Functions as the CXHA CCIX Protocol Link 0 status register. Works with por\_ccg\_ha\_cxprtcl\_link0\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1908

##### Type

RO

##### Reset value

See individual bit resets

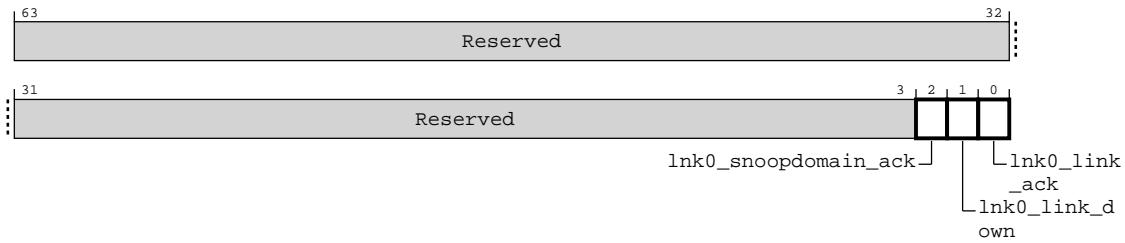
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-36: por\_ccg\_ha\_cxprtcl\_link0\_status**



**Table 8-35: por\_ccg\_ha\_cxprtcl\_link0\_status attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	lnk0_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 0	RO	0b0
[1]	lnk0_link_down	<p>Link Down status; hardware updates this register bit to indicate Link Down status</p> <p><b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request</p> <p><b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear</p>	RO	0b1
[0]	lnk0_link_ack	<p>Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request</p> <p><b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear</p> <p><b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent</p> <p><b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.</p>	RO	0b0

### 8.3.2.23 por\_ccg\_ha\_cxprtcl\_link1\_ctl

Functions as the CXHA CCIX Protocol Link 1 control register. Works with por\_ccg\_ha\_cxprtcl\_link1\_status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

### Address offset

0x1910

### Type

RW

### Reset value

See individual bit resets

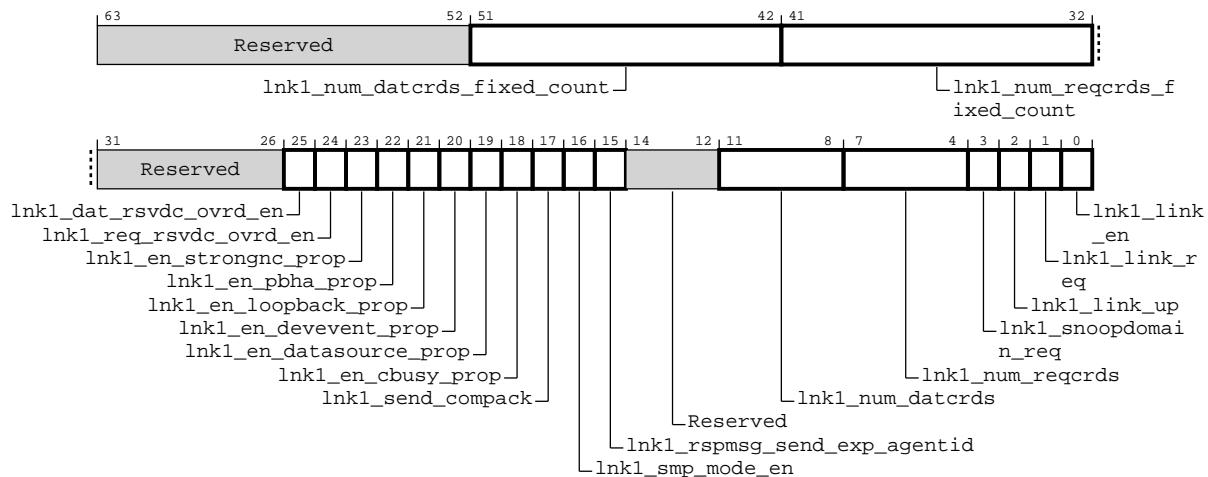
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-37: por\_ccg\_ha\_cxpctl\_link1\_ctl**



**Table 8-36: por\_ccg\_ha\_cxpctl\_link1\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:42]	lnk1_num_datcrds_fixed_count	Controls the number of CCIX data credits assigned to Link 1 If this field along with link0, link2 fixed credit count is zero then the hardware uses lnk1_num_datcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_datcrds_fixed_count should not exceed dat_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds dat_pass_buff_depth then it will cause fatal errors	RW	0x00
[41:32]	lnk1_num_reqcrds_fixed_count	Controls the number of CCIX request credits assigned to Link 1 If this field along with link0, link2 fixed credit count is zero then the hardware uses lnk1_num_reqcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_reqcrds_fixed_count should not exceed req_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds req_pass_buff_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25]	lnk1_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 1.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[24]	lnk1_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 1.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[23]	lnk1_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 1.	RW	0b0
[22]	lnk1_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 1.	RW	0b0
[21]	lnk1_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 1.	RW	0b0
[20]	lnk1_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 1.	RW	0b1
[19]	lnk1_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 1.	RW	0b1
[18]	lnk1_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 1.	RW	0b1
[17]	lnk1_send_compack	When set, sends CompAck for CCIX Link 1.	RW	0b0
[16]	lnk1_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 1.	RW	0x0
[15]	lnk1_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 1	RW	0b0
[14:12]	Reserved	Reserved	RO	-
[11:8]	lnk1_num_datcrds	Controls the number of CCIX data credits assigned to Link 1  <b>0x0</b> Total credits equally divided across all links <b>0x1</b> 25% of credits assigned <b>0x2</b> 50% of credits assigned <b>0x3</b> 75% of credits assigned <b>0x4</b> 100% of credits assigned <b>0xF</b> 0% of credits assigned	RW	0b0

Bits	Name	Description	Type	Reset
[7:4]	lnk1_num_reqcrds	<p>Controls the number of CCIX request credits assigned to Link 1</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk1_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 1	RW	0b0
[2]	lnk1_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0
[1]	lnk1_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk1_link_en	<p>Enables CCIX Link 1 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.2.24 por\_ccg\_ha\_cxprtcl\_link1\_status

Functions as the CXHA CCIX Protocol Link 1 status register. Works with por\_ccg\_ha\_cxprtcl\_link1\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1918

##### Type

RO

##### Reset value

See individual bit resets

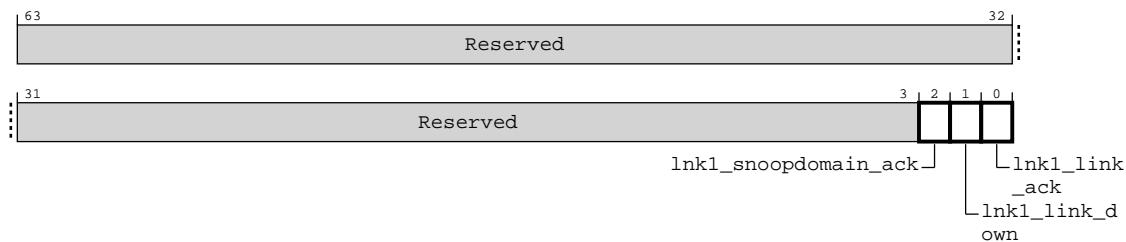
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-38: por\_ccg\_ha\_cxprtcl\_link1\_status**



**Table 8-37: por\_ccg\_ha\_cxprtcl\_link1\_status attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	lnk1_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 1	RO	0b0

Bits	Name	Description	Type	Reset
[1]	lnk1_link_down	<p>Link Down status; hardware updates this register bit to indicate Link Down status</p> <p><b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request</p> <p><b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear</p>	RO	0b1
[0]	lnk1_link_ack	<p>Link Up/Down Acknowledge; hardware updates this register bit to acknowledge the software link request</p> <p><b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear</p> <p><b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent</p> <p><b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.</p>	RO	0b0

### 8.3.2.25 por\_ccg\_ha\_cxprtcl\_link2\_ctl

Functions as the CXHA CCIX Protocol Link 2 control register. Works with por\_ccg\_ha\_cxprtcl\_link2\_status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1920

##### Type

RW

##### Reset value

See individual bit resets

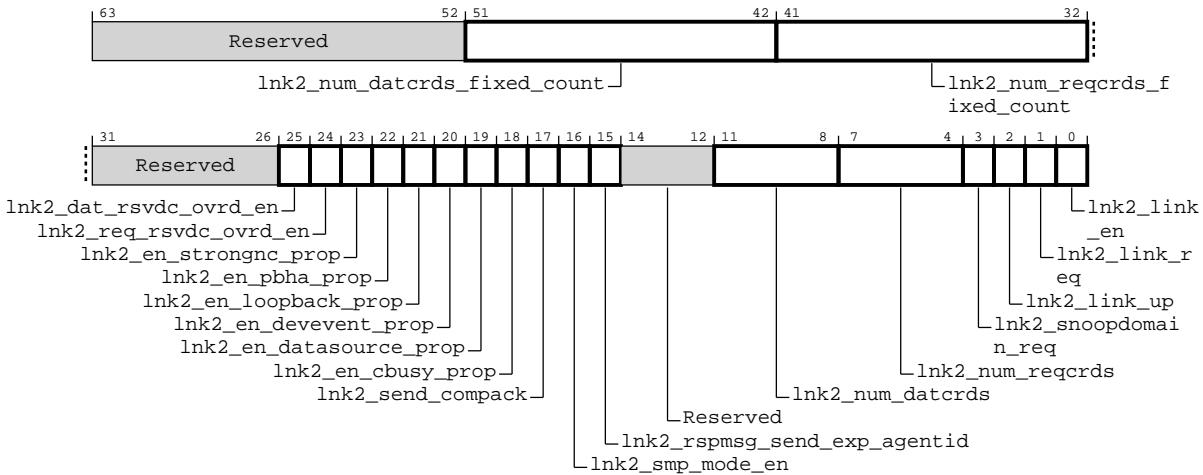
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-39: por\_ccg\_ha\_cxprtcl\_link2\_ctl**



**Table 8-38: por\_ccg\_ha\_cxprtcl\_link2\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:42]	lnk2_num_datcrds_fixed_count	Controls the number of CCIX data credits assigned to Link 2 If this field along with link0, link1 fixed credit count is zero then the hardware uses lnk2_num_datcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_datcrds_fixed_count should not exceed dat_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds dat_pass_buff_depth then it will cause fatal errors	RW	0x00
[41:32]	lnk2_num_reqcrds_fixed_count	Controls the number of CCIX request credits assigned to Link 2 If this field along with link0, link1 fixed credit count is zero then the hardware uses lnk2_num_reqcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_reqcrds_fixed_count should not exceed req_pass_buff_depth field which is in por_ccg_ha_unit_info2. if the sum exceeds req_pass_buff_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-
[25]	lnk2_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 2.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[24]	lnk2_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 2.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	0b0
[23]	lnk2_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 2.	RW	0b0
[22]	lnk2_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 2.	RW	0b0
[21]	lnk2_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 2.	RW	0b0
[20]	lnk2_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 2.	RW	0b1
[19]	lnk2_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 2.	RW	0b1
[18]	lnk2_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 2.	RW	0b1

Bits	Name	Description	Type	Reset
[17]	lnk2_send_compack	When set, sends CompAck for CCIX Link 2.	RW	0b0
[16]	lnk2_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 2.	RW	0x0
[15]	lnk2_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 2	RW	0b0
[14:12]	Reserved	Reserved	RO	-
[11:8]	lnk2_num_datcrds	<p>Controls the number of CCIX data credits assigned to Link 2</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[7:4]	lnk2_num_reccrds	<p>Controls the number of CCIX request credits assigned to Link 2</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk2_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 2	RW	0b0
[2]	lnk2_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0

Bits	Name	Description	Type	Reset
[1]	lnk2_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk2_link_en	<p>Enables CCIX Link 2 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.2.26 por\_ccg\_ha\_cxprtcl\_link2\_status

Functions as the CXHA CCIX Protocol Link 2 status register. Works with por\_ccg\_ha\_cxprtcl\_link2\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1928

##### Type

RO

##### Reset value

See individual bit resets

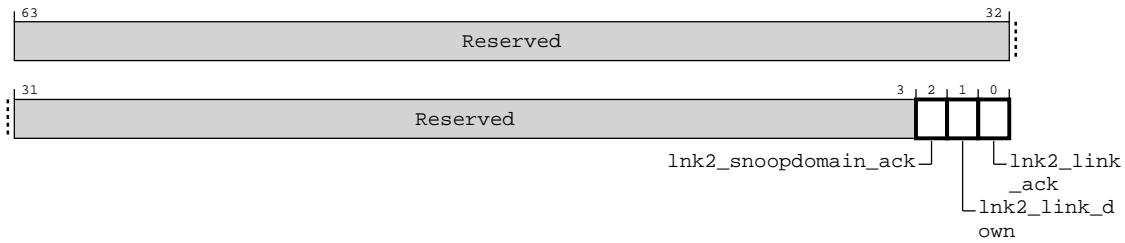
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-40: por\_ccg\_ha\_cxprtcl\_link2\_status**



**Table 8-39: por\_ccg\_ha\_cxprtcl\_link2\_status attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	Lnk2_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 2	RO	0b0
[1]	Lnk2_link_down	<p>Link Down status; hardware updates this register bit to indicate Link Down status</p> <p><b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request</p> <p><b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear</p>	RO	0b1
[0]	Lnk2_link_ack	<p>Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request</p> <p><b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear</p> <p><b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent</p> <p><b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.</p>	RO	0b0

### 8.3.2.27 por\_ccg\_ha\_datasource\_ctl\_link0

Control to determine the chi-g datasource on SnpRespData (Link-0)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xA30

## Type

RW

## Reset value

See individual bit resets

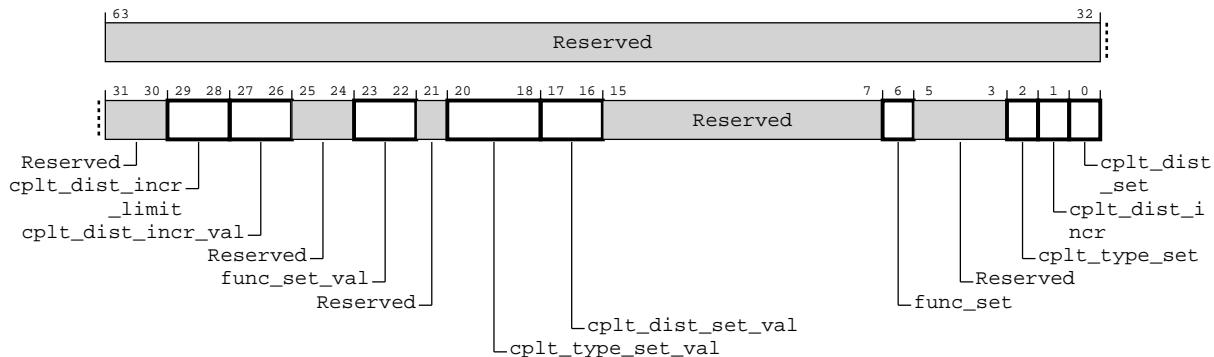
## Usage constraints

This register is owned in the Root space and is 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 8-41: por\_ccg\_ha\_datasource\_ctl\_link0**



**Table 8-40: por\_ccg\_ha\_datasource\_ctl\_link0 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCLIMIT	RW	0b01
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b000
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0

Bits	Name	Description	Type	Reset
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.2.28 por\_ccg\_ha\_datasource\_ctl\_link1

Control to determine the chi-g datasource on SnpRespData (Link-1)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA38

##### Type

RW

##### Reset value

See individual bit resets

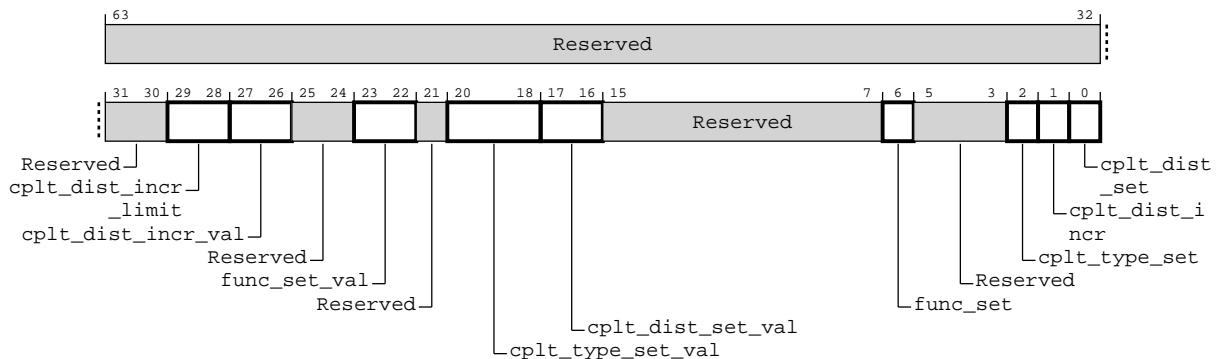
##### Usage constraints

This register is owned in the Root space and is 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 8-42: por\_ccg\_ha\_datasource\_ctl\_link1**



**Table 8-41: por\_ccg\_ha\_datasource\_ctl\_link1 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b01
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b000
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.2.29 por\_ccg\_ha\_datasource\_ctl\_link2

Control to determine the chi-g datasource on SnpRespData (Link-2)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA40

##### Type

RW

##### Reset value

See individual bit resets

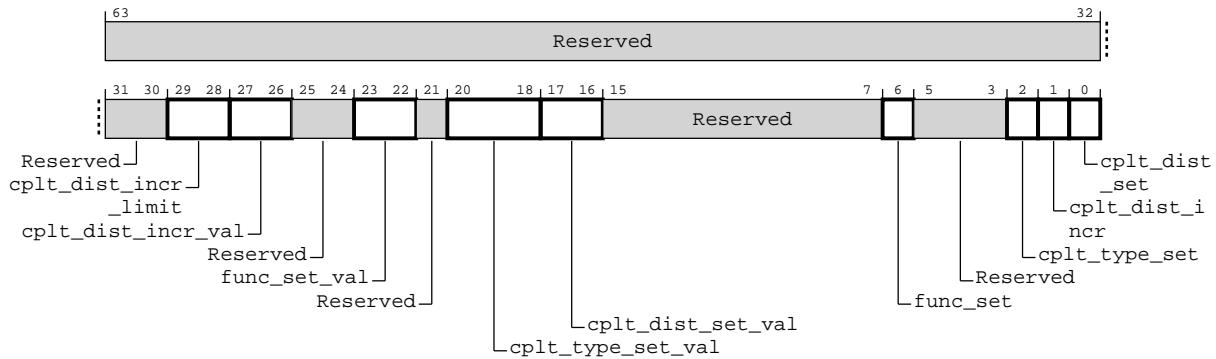
##### Usage constraints

This register is owned in the Root space and is 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 8-43: por\_ccg\_ha\_datasource\_ctl\_link2**



**Table 8-42: por\_ccg\_ha\_datasource\_ctl\_link2 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b01
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b000
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.3 CCG RA register summary

The following table describes the registers for the relevant component.

**Table 8-43: por\_ccg\_ra\_cfg register summary**

Offset	Name	Type	Description
0x0	por_ccg_ra_node_info	RO	<a href="#">por_ccg_ra_node_info</a>
0x80	por_ccg_ra_child_info	RO	<a href="#">por_ccg_ra_child_info</a>
0x980	por_ccg_ra_scr	RW	<a href="#">por_ccg_ra_scr</a>
0x988	por_ccg_ra_rcr	RW	<a href="#">por_ccg_ra_rcr</a>

Offset	Name	Type	Description
0x900	por_ccg_ra_unit_info	RO	por_ccg_ra_unit_info
0x908	por_ccg_ra_unit_info_1	RO	por_ccg_ra_unit_info_1
0xA00	por_ccg_ra_cfg_ctl	RW	por_ccg_ra_cfg_ctl
0xA08	por_ccg_ra_aux_ctl	RW	por_ccg_ra_aux_ctl
0xA10	por_ccg_ra_aux_ctl2	RW	por_ccg_ra_aux_ctl2
0xA18	por_ccg_ra_cbusy_limit_ctl	RW	por_ccg_ra_cbusy_limit_ctl
0xC00 : 0xC38	por_ccg_ra_sam_addr_region_reg0-7	RW	por_ccg_ra_sam_addr_region_reg0-7
0xD00	por_ccg_ra_agentid_to_linkid_val	RW	por_ccg_ra_agentid_to_linkid_val
0xD10 : 0xD10	por_ccg_ra_agentid_to_linkid_reg0-0	RW	por_ccg_ra_agentid_to_linkid_reg0-0
0xE00 : 0xEF8	por_ccg_ra_rni_ldid_to_exp_raid_reg0-31	RW	por_ccg_ra_rni_ldid_to_exp_raid_reg0-31
0xF00 : 0xFF8	por_ccg_ra_rnd_ldid_to_exp_raid_reg0-31	RW	por_ccg_ra_rnd_ldid_to_exp_raid_reg0-31
0x1000 : 0x13F8	por_ccg_ra_rnf_ldid_to_exp_raid_reg0-127	RW	por_ccg_ra_rnf_ldid_to_exp_raid_reg0-127
0x1400 : 0x15F8	por_ccg_ra_ha_ldid_to_exp_raid_reg0-63	RW	por_ccg_ra_ha_ldid_to_exp_raid_reg0-63
0x1600 : 0x1678	por_ccg_ra_hns_ldid_to_exp_raid_reg0-15	RW	por_ccg_ra_hns_ldid_to_exp_raid_reg0-15
0x1680 : 0x1A78	por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg0-127	RW	por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg0-127
0x1A80 : 0x1B78	por_ccg_ra_cml_port_aggr_grp0-31_addr_mask	RW	por_ccg_ra_cml_port_aggr_grp0-31_addr_mask
0x1B80 : 0x1BE0	por_ccg_ra_cml_port_aggr_grp_reg0-12	RW	por_ccg_ra_cml_port_aggr_grp_reg0-12
0x1BE8 : 0x1C18	por_ccg_ra_cml_port_aggr_ctrl_reg0-6	RW	por_ccg_ra_cml_port_aggr_ctrl_reg0-6
0x1C20 : 0x1E18	por_ccg_ra_ha_ldid_to_ovrd_ldid_reg0-63	RW	por_ccg_ra_ha_ldid_to_ovrd_ldid_reg0-63
0x1E38 : 0x2030	por_ccg_ra_ha_ldid_to_exp_raid_cpagen_reg0-63	RW	por_ccg_ra_ha_ldid_to_exp_raid_cpagen_reg0-63
0xD900	por_ccg_ra_pmu_event_sel	RW	por_ccg_ra_pmu_event_sel
0x4000	por_ccg_ra_ccprtcl_link0_ctl	RW	por_ccg_ra_ccprtcl_link0_ctl
0x4008	por_ccg_ra_ccprtcl_link0_status	RO	por_ccg_ra_ccprtcl_link0_status
0x4010	por_ccg_ra_ccprtcl_link1_ctl	RW	por_ccg_ra_ccprtcl_link1_ctl
0x4018	por_ccg_ra_ccprtcl_link1_status	RO	por_ccg_ra_ccprtcl_link1_status
0x4020	por_ccg_ra_ccprtcl_link2_ctl	RW	por_ccg_ra_ccprtcl_link2_ctl
0x4028	por_ccg_ra_ccprtcl_link2_status	RO	por_ccg_ra_ccprtcl_link2_status
0x4030	ccg_ra_c2c_non_caching_agent_id	RW	ccg_ra_c2c_non_caching_agent_id
0xA20	por_ccg_ra_datasource_ctl_link0	RW	por_ccg_ra_datasource_ctl_link0
0xA28	por_ccg_ra_datasource_ctl_link1	RW	por_ccg_ra_datasource_ctl_link1
0xA30	por_ccg_ra_datasource_ctl_link2	RW	por_ccg_ra_datasource_ctl_link2
{0-31} 0x3000 : 0x30F8	ra_rnsam_hashed_tgt_grp_cfg1_region0-31	RW	ra_rnsam_hashed_tgt_grp_cfg1_region0-31
{0-31} 0x3100 : 0x31F8	ra_rnsam_hashed_tgt_grp_cfg2_region0-31	RW	ra_rnsam_hashed_tgt_grp_cfg2_region0-31
{0-31} 0x3200 : 0x32F8	ra_rnsam_hashed_target_grp_secondary_cfg1_reg0-31	RW	ra_rnsam_hashed_target_grp_secondary_cfg1_reg0-31
{0-31} 0x3300 : 0x33F8	ra_rnsam_hashed_target_grp_secondary_cfg2_reg0-31	RW	ra_rnsam_hashed_target_grp_secondary_cfg2_reg0-31
{0-31} 0x3400 : 0x34F8	ra_rnsam_hashed_target_grp_hash_cntl_reg0-31	RW	ra_rnsam_hashed_target_grp_hash_cntl_reg0-31

Offset	Name	Type	Description
{0-3} 0x3700 : 0x3718	ra_rnsam_hashed_target_group_hn_count_reg0-3	RW	<a href="#">ra_rnsam_hashed_target_group_hn_count_reg0-3</a>
{0-31} 0x3500 : 0x35F8	ra_rnsam_hashed_target_grp_hnf_nodeid_reg0-31	RW	<a href="#">ra_rnsam_hashed_target_grp_hnf_nodeid_reg0-31</a>
{0-7} 0x3780 : 0x37B8	ra_rnsam_hashed_target_grp_cal_mode_reg0-7	RW	<a href="#">ra_rnsam_hashed_target_grp_cal_mode_reg0-7</a>
{0-31} 0x3840 : 0x3938	ra_rnsam_hashed_target_grp_compact_hash_ctrl0-31	RW	<a href="#">ra_rnsam_hashed_target_grp_compact_hash_ctrl0-31</a>
{0-7} 0x3A00 : 0x3A38	ra_rnsam_hashed_target_grp_misc_nodeid_reg0-7	RW	<a href="#">ra_rnsam_hashed_target_grp_misc_nodeid_reg0-7</a>
{0-7} 0x3B00 : 0x3B38	ra_rnsam_hashed_tgt_override_cam_reg0-7	RW	<a href="#">ra_rnsam_hashed_tgt_override_cam_reg0-7</a>
0x37C0	ra_rnsam_hash_addr_mask_reg	RW	<a href="#">ra_rnsam_hash_addr_mask_reg</a>
0x37C8	ra_rnsam_region_cmp_addr_mask_reg	RW	<a href="#">ra_rnsam_region_cmp_addr_mask_reg</a>
0x37D0	ra_rnsam_status	RW	<a href="#">ra_rnsam_status</a>
{0-7} 0x3800 : 0x3838	ra_rnsam_sam_generic_regs0-7	RW	<a href="#">ra_rnsam_sam_generic_regs0-7</a>

### 8.3.3.1 por\_ccg\_ra\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

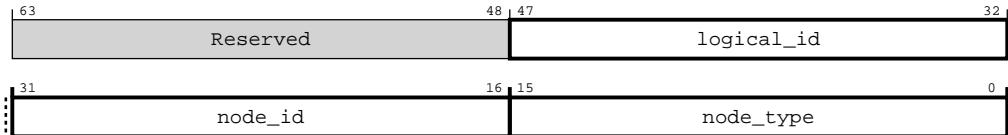
##### Usage constraints

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 8-44: por\_ccg\_ra\_node\_info**



**Table 8-44: por\_ccg\_ra\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0103

### 8.3.3.2 por\_ccg\_ra\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

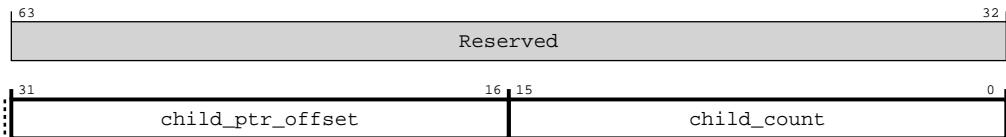
#### Usage constraints

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 8-45: por\_ccg\_ra\_child\_info**



**Table 8-45: por\_ccg\_ra\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0x0

### 8.3.3.3 por\_ccg\_ra\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

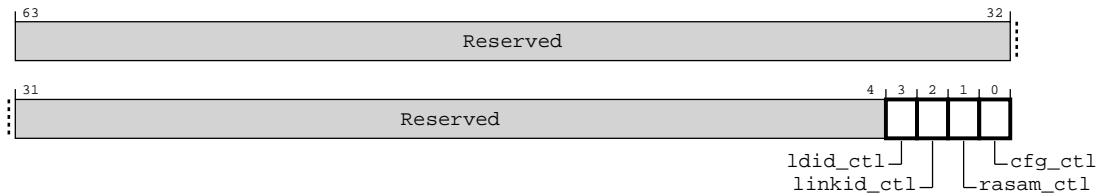
#### Usage constraints

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 8-46: por\_ccg\_ra\_scr**



**Table 8-46: por\_ccg\_ra\_scr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	ldid_ctl	Allows Secure override of the RA LDID registers	RW	0b0
[2]	linkid_ctl	Allows Secure override of the RA Link ID registers	RW	0b0
[1]	rasam_ctl	Allows Secure override of the RA SAM control registers	RW	0b0
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.3.4 por\_ccg\_ra\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

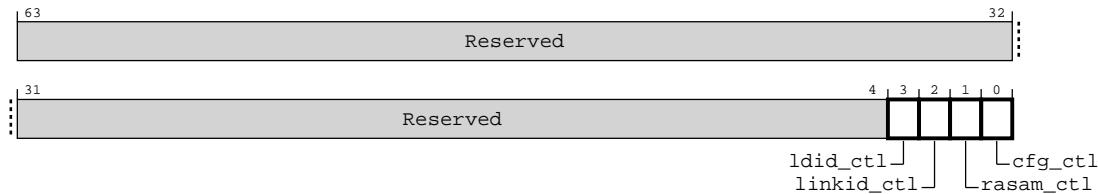
##### Usage constraints

This register is owned in the Root space and is 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 8-47: por\_ccg\_ra\_rcr**



**Table 8-47: por\_ccg\_ra\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	ldid_ctl	Allows Root override of the RA LDID registers	RW	0b0
[2]	linkid_ctl	Allows Root override of the RA Link ID registers	RW	0b0
[1]	rasam_ctl	Allows Root override of the RA SAM control registers	RW	0b0
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.3.5 por\_ccg\_ra\_unit\_info

Provides component identification information for CXRA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

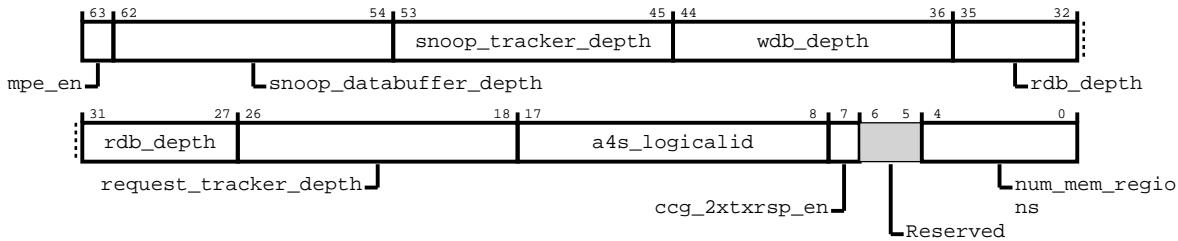
##### Usage constraints

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 8-48: por\_ccg\_ra\_unit\_info**



**Table 8-48: por\_ccg\_ra\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63]	mpe_en	Enable MPE in CCG. MPE is not supported in CML SMP and needs to be set to 0b0	RO	0x1
[62:54]	snoop_databuffer_depth	Depth of Snoop Data Buffer - number of outstanding SNP requests on CHI	RO	Configuration dependent
[53:45]	snoop_tracker_depth	Depth of Snoop Tracker - number of outstanding SNP requests on CCIX	RO	Configuration dependent
[44:36]	wdb_depth	Depth of Write Data Buffer	RO	Configuration dependent
[35:27]	rdb_depth	Depth of Read Data Buffer	RO	Configuration dependent
[26:18]	request_tracker_depth	Depth of Request Tracker - number of outstanding Memory requests on CCIX	RO	Configuration dependent
[17:8]	a4s_logicalid	AXI4Stream interfaces logical ID	RO	0x0
[7]	ccg_2txrsp_en	Enable more than 1 TX CHI ports. Total number of TX CHI response ports are 2.	RO	Configuration dependent
[6:5]	Reserved	Reserved	RO	-
[4:0]	num_mem_regions	Number of memory regions supported	RO	0x6

### 8.3.3.6 por\_ccg\_ra\_unit\_info\_1

Provides component identification information for CXRA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x908

##### Type

RO

### Reset value

See individual bit resets

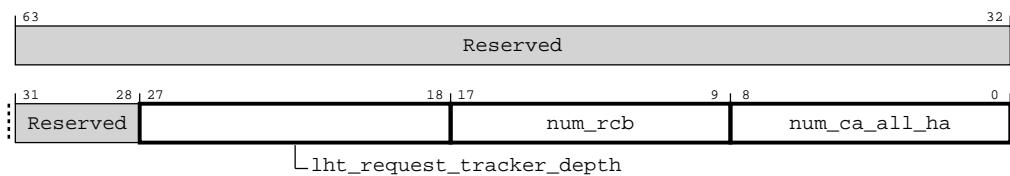
### Usage constraints

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 8-49: por\_ccg\_ra\_unit\_info\_1**



**Table 8-49: por\_ccg\_ra\_unit\_info\_1 attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:18]	lht_request_tracker_depth	Depth of LHT Request Tracker - number of outstanding Memory requests on C2C	RO	0x200
[17:9]	num_rcb	Number rcb in RA	RO	0x80
[8:0]	num_ca_all_ha	Number of Caching agents behind a HA on CC at IOC-RA.	RO	0x40

### 8.3.3.7 por\_ccg\_ra\_cfg\_ctl

Functions as the configuration control register. Specifies the current mode.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA00

#### Type

RW

#### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

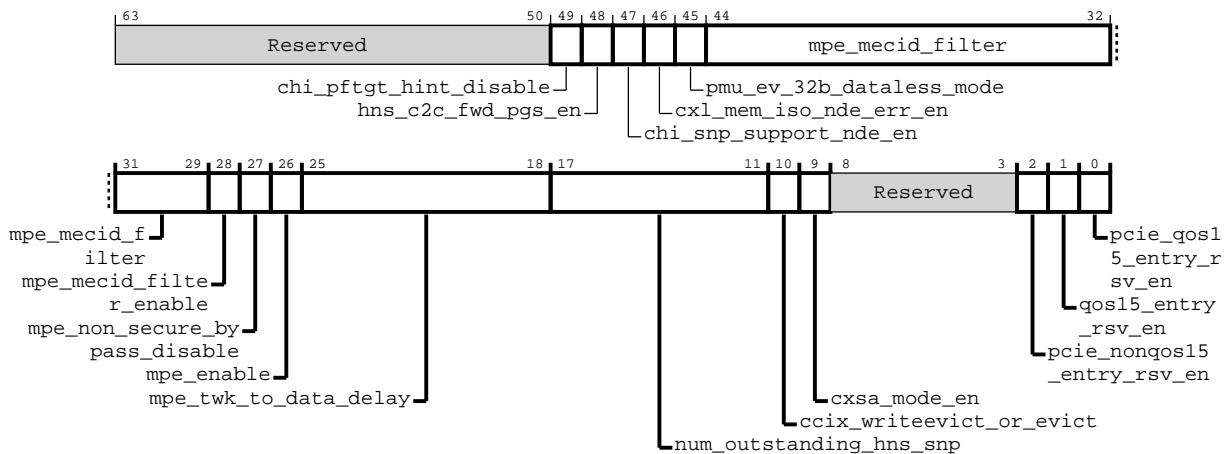
### Usage constraints

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 8-50: por\_ccg\_ra\_cfg\_ctl**



**Table 8-50: por\_ccg\_ra\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49]	chi_pftgt_hint_disable	<b>0b1</b> CHI Prefetch target hint is not propagated with the CHI request. <b>0b0</b> CHI Prefetch target hint is propagated with the CHI request.	RW	0x0
[48]	hns_c2c_fwd_pgs_en	<b>0b1</b> hns traffic to ra over chi c2c is considered forward progress gaurantee <b>0b0</b> hns traffic to ra over chi c2c is not considered forward progress gaurantee	RW	0x1
[47]	chi_snp_support_nde_en	Sends NDE when RA downgrades an incoming request for C2C SnoopSupport=false property	RW	0x1
[46]	cxl_mem_iso_nde_err_en	Switch to NDE for the completion response for CHI Write requests, data with all 1's and NDE for CHI Read requests	RW	0x0
[45]	pmu_ev_32b_dataless_mode	Select 32B mode for dataless Rd/Wr request for PMU events	RW	0x0
[44:29]	mpe_mecid_filter	Specifies mecid that is used to filter out incoming request to not to send to MPE	RW	0x0

Bits	Name	Description	Type	Reset
[28]	mpe_mecid_filter_enable	Specifies if mecid based filtering to send incoming request to MPE is enabled	RW	0b0
[27]	mpe_non_secure_bypass_disable	When clear, non-secure data bypasses the MPE. When set, all data (including non-secure) is sent to the MPE. This register bit polarity allows non-secure data to be bypassed by default	RW	0b0
[26]	mpe_enable	Specifies if MPE is enable	RW	0b0
[25:18]	mpe_twk_to_data_delay	Specifies the number of clocks between mpe tweak and data	RW	0x1f
[17:11]	num_outstanding_hns_snp	Specifies the Max number of outstanding snoops from the given RA to each HNS to guarantee snoop sink and deadlock prevention. Must be set to (HNS_NUM_ENTRIES_SNPQ_PARAM)/(NUM_NON_CXSA_RA).	RW	0x2
[10]	ccix_writeevict_or_evict	When set, downgrades WriteEvictOrEvict to Evict  <b>0b1</b> Evict is sent  <b>0b0</b> WriteEvict is sent	RW	0b0
[9]	cxsa_mode_en	When set, enables the CCIX Subordinate Agent mode.In this mode RA functions as a CCIX Subordinate Agent  <b>0b1</b> CCIX Subordinate Agent  <b>0b0</b> CCIX Requesting Agent	RW	0b0
[8:3]	Reserved	Reserved	RO	
[2]	pcie_nonqos15_entry_rsv_en	Enables entry reservation for non QoS15 traffic from PCIe RN-I/RN-D  <b>0b1</b> Reserves tracker entry for non QoS15 requests from PCIe RN-I/RN-D  <b>0b0</b> Does not reserve tracker entry for non QoS15 requests from PCIe RN-I/RN-D	RW	0b1
[1]	qos15_entry_rsv_en	Enables entry reservation for QoS15 traffic  <b>0b1</b> Reserves tracker entry for QoS15 requests  <b>0b0</b> Does not reserve tracker entry for QoS15 requests	RW	0b1
[0]	pcie_qos15_entry_rsv_en	Enables entry reservation for QoS15 traffic from PCIe RN-I/RN-D  <b>0b1</b> Reserves tracker entry for QoS15 requests from PCIe RN-I/RN-D  <b>0b0</b> Does not reserve tracker entry for QoS15 requests from PCIe RN-I/RN-D	RW	0b1

### 8.3.3.8 por\_ccg\_ra\_aux\_ctl

Functions as the auxiliary control register for CXRA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

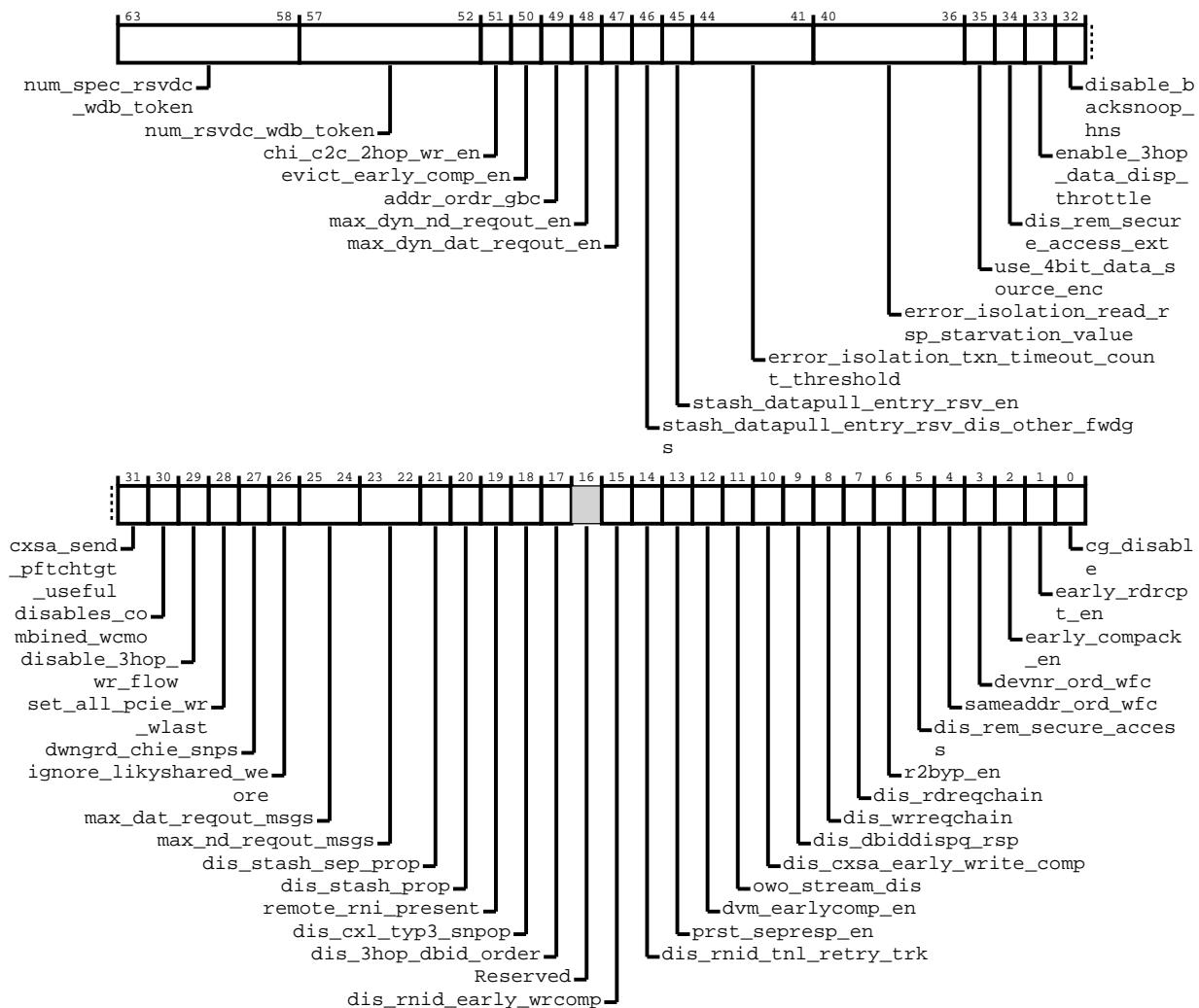
#### Usage constraints

This register is owned in the Root space and is 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 8-51: por\_ccg\_ra\_aux\_ctl**



**Table 8-51: por\_ccg\_ra\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:58]	num_spec_rsvdc_wdb_token	Number of SPEC RSVD WDB token for write pull. Cannot be changed below 2	RW	0x2
[57:52]	num_rsvdc_wdb_token	Number of RSVD WDB token for write pull. Cannot be changed below 2	RW	0x2
[51]	chi_c2c_2hop_wr_en	Enables Write Push for CHI C2C in RA	RW	0b1
[50]	evict_early_comp_en	When set, allows Evict send from HNS to receive early comp	RW	0b1
[49]	addr_ordr_gbc	When set, enables waiting for completion (COMP) before dispatching next same Addr dependent transaction (TXN). This includes the request group of CMO, WCMO, Stash and Atomics	RW	0b0
[48]	max_dyn_nd_reqout_en	When set, enables RA to dynamically pack 3 or 4 ND request based on number of extensions	RW	0b1

Bits	Name	Description	Type	Reset
[47]	max_dyn_dat_reqout_en	When set, enables RA to dynamically pack 3 or 4 DATHDR request based on number of extensions	RW	0b1
[46]	stash_datapull_entry_rsv_dis_other_fwdgs	When set, only entry 3 in bank0 can be used for read request due to stash snoop data pull.	RW	0b0
[45]	stash_datapull_entry_rsv_en	When set, reserves at least entry 3 in bank0 of the RA RHT for read request due to stash snoop data pull	RW	0b1
[44:41]	error_isolation_txn_timeout_count_threshold	The number of transaction timeouts, which would trigger the cxl.mem isolation.	RW	0x1
[40:36]	error_isolation_read_rsp_starvation_value	The number of cycles for which incoming data resp will take priority over synthesized read resp in case of error isolation triggered.	RW	0b11111
[35]	use_4bit_data_source_enc	This field is deprecated from CHIG. When set, forces use of 4-bit Data Source Encoding. Used for DMT	RW	0b0
[34]	dis_rem_secure_access_ext	When set, forces the NSE bit to 1 and the address space can be selected between Root and Realm with NS bit for the incoming snoops	RW	0b0
[33]	enable_3hop_data_disp_throttle	When set, enables throttling of the 3hop write data throttling if dathdr_rspout_fifo is out of tokens	RW	0b1
[32]	disable_backsnoop_hns	When set, disables back snooping on an HNS/LCN initiated WriteClean request which is due to LLC eviction	RW	0b1
[31]	cxsa_send_pftchtgt_useful	When set, sends PreFetch Usefull indication on CHI datasource field in CXSA mode	RW	0b0
[30]	disables_combined_wcmo	When set, disables sending combined W+CMO to remote HA. Applicable only in SMP mode	RW	0b1
[29]	disable_3hop_wr_flow	When set, disables 3-hop write flow to remote HA. Applicable only in SMP mode	RW	0b0
[28]	set_all_pcie_wr_wlast	When set, sets WLAST indication for all PCIe writes going to HA RNI	RW	0b0
[27]	dwngrd_chie_snps	When set, downgrades CHIE SnpPreferUnique to SnpNotSharedDirty	RW	0b0
[26]	ignore_likyshared_weore	<p>When set, disables the use of LikelyShared(LS) bit to make a decision for WriteEvictOrEvict</p> <p><b>0b0</b> Send WriteEvict when LS= 0 and send Evict when LS=1</p> <p><b>0b1</b> Ignore LS bit. WriteEvict is sent. Further static decision can be made using ccix_writeevict_or_evict in cfg_ctl register</p>	RW	0b0

Bits	Name	Description	Type	Reset
[25:24]	max_dat_reqout_msgs	<p>Used to configure the maximum number of data requests messages (writes, atomics etc.) presented to CCLA's packing logic.</p> <p><b>0b00</b> one message</p> <p><b>0b01</b> two messages</p> <p><b>0b10</b> three messages</p> <p><b>0b11</b> four messages</p> <p><b>Note</b> The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)</p>	RW	0b11
[23:22]	max_nd_reqout_msgs	<p>Used to configure the maximum number of non-data requests messages (reads, dataless) presented to CCLA's packing logic.</p> <p><b>0b00</b> one message</p> <p><b>0b01</b> two messages</p> <p><b>0b10</b> three messages</p> <p><b>0b11</b> four messages</p> <p><b>Note</b> The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)</p>	RW	0b11
[21]	dis_stash_sep_prop	When set, disables propagation of StashSep opcodes on CCIX. StashSep opcodes are sent as Stash opcodes when set. Applicable only in SMP mode	RW	0b0
[20]	dis_stash_prop	When set, disables propagation of stash opcodes on CCIX. Applicable only in SMP mode. This is not applicable to C2C and should not be changed.	RW	0b0
[19]	remote_rni_present	When set, Enables TXNID coloring to enable traffic to remote RNI	RW	0b1
[18]	dis_cxl_typ3_snpop	When set, drives SnpType= <b>NOP</b> on CXL Type3 M2S Req and RwD messages	RW	0b1
[17]	dis_3hop_dbid_order	When set, disables ordered dispatch of DBIDs for 3-hop writes. By default, 3-hop DBIDs are dispatched in order. Applicable only if 3-hop write flow is enabled	RW	0b0
[16]	Reserved	Reserved	RO	-
[15]	dis_rnid_early_wrcomp	When set, disables early write completions for tunneled writes from RNI.	RW	0b0
[14]	dis_rnid_tnl_retry_trk	When set, disables RNID write request tunneling retry tracker.	RW	0b0

Bits	Name	Description	Type	Reset
[13]	prst_sepresp_en	When set, enables separate persist response on CCIX for persistent cache maintenance (PCMO2) operation  <b>Note</b> this bit is applicable only in SMP mode.	RW	0b1
[12]	dvm_earlycomp_en	When set, enables early DVM Op completion responses from RA.	RW	0b1
[11]	owo_stream_dis	When set, disables CompAck dependency to dispatch an ordered PCIe write.	RW	0b1
[10]	dis_cxsa_early_write_comp	When set, disables early write completions in CCIX Subordinate Agent mode.	RW	0b0
[9]	dis_dbiddispq_rsp	When set, disables the dispatch of DBID responses from a separate DispatchQ.	RW	0b0
[8]	dis_wrreqchain	When set, disables chaining of write requests.	RW	0b0
[7]	dis_rdreqchain	When set, disables chaining of read and dataless requests.	RW	0b0
[6]	r2byp_en	When set, enables request bypass. Applies to read and dataless requests only.  <b>Note</b> When set will affect the capability to chain a request on the TX side	RW	0b1
[5]	dis_rem_secure_access	When set, treats all the incoming snoops as non-secure and forces the NS bit to 1	RW	0b0
[4]	sameaddr_ord_wfc	When set, enables waiting for completion (COMP) before dispatching next same Addr dependent transaction (TXN)	RW	0b0
[3]	devnr_ord_wfc	When set, enables waiting for completion (COMP) before dispatching next Device-nR dependent transaction (TXN)	RW	0b0
[2]	early_compack_en	Early CompAck enable; enables sending early CompAck on CCIX for requests that require CompAck	RW	0b1
[1]	early_rdrcpt_en	Early ReadReceipt enable; enables sending early ReadReceipt for ordered read requests	RW	0b1
[0]	cg_disable	Disables clock gating when set	RW	0b0

### 8.3.3.9 por\_ccg\_ra\_aux\_ctl2

Extension to the first aux ctl reg which functions as the auxiliary control register for CXRA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA10

## Type

RW

## Reset value

See individual bit resets

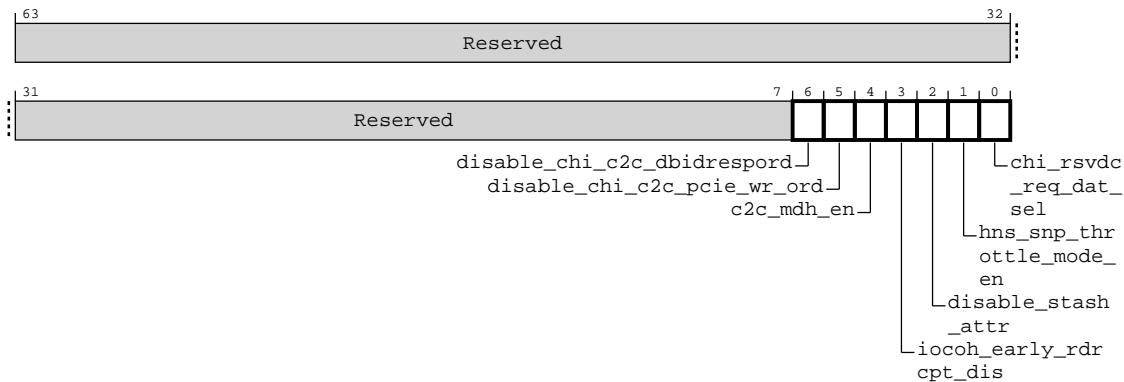
## Usage constraints

This register is owned in the Root space and is 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 8-52: por\_ccg\_ra\_aux\_ctl2**



**Table 8-52: por\_ccg\_ra\_aux\_ctl2 attributes**

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	disable_chi_c2c_dbidrespord	Disables DBIDRESPORD for non-copy-back writes when chi c2c mode is enabled	RW	0x1
[5]	disable_chi_c2c_pcie_wr_ord	PCIE Write Tunneling and Write Streaming features are disabled and no ordering is enforced. RNI is expected to serialize all PCIE Writes. Only applicable during CHI C2C mode	RW	0x1
[4]	c2c_mdh_en	CCL DATHDR RSPOUT and REQOUT MDH Mode Enable during C2C	RW	0x0
[3]	iocoh_early_rdrcpt_dis	early read receipt disable for io coherent read request in c2c mode. This should be set if DataSepResp and RespSepData are supported responses for io coherent reads from c2c downstream	RW	0x1
[2]	disable_stash_attr	stashnid, stashnidv, stashlpid and stashlpidv are not pass through in chi c2c mode	RW	0x1
[1]	hns_snp_throttle_mode_en	To enable Snoop throttling for HNS	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[0]	chi_rsvdc_req_dat_sel	Selects RSVDC fields to be used from Request or Data for Write Push for C2C  <b>0b1</b> Req RSVDC is selected  <b>0b0</b> Data RSVDC is selected	RW	0x1

### 8.3.3.10 por\_ccg\_ra\_cbusy\_limit\_ctl

Cbusy threshold limits for RHT entries.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA18

##### Type

RW

##### Reset value

See individual bit resets

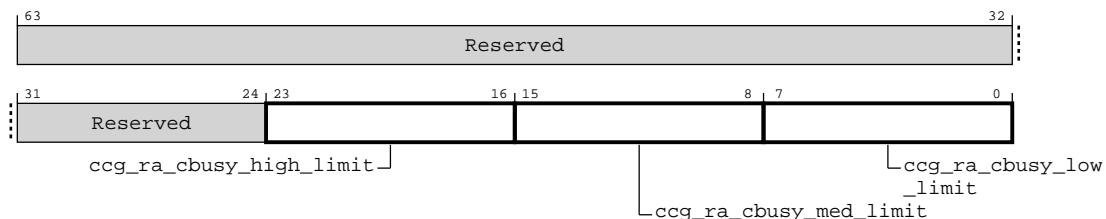
##### Usage constraints

This register is owned in the Root space and is 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 8-53: por\_ccg\_ra\_cbusy\_limit\_ctl**



**Table 8-53: por\_ccg\_ra\_cbusy\_limit\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	ccg_ra_cbusy_high_limit	RHT limit for CBusy High	RW	0x18
[15:8]	ccg_ra_cbusy_med_limit	RHT limit for CBusy Med	RW	0x10
[7:0]	ccg_ra_cbusy_low_limit	RHT limit for CBusy Low	RW	0x8

### 8.3.3.11 por\_ccg\_ra\_sam\_addr\_region\_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures Address Region #{index} for RA SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.rasam\_ctl

##### Secure group override

por\_ccg\_ra\_scr.rasam\_ctl

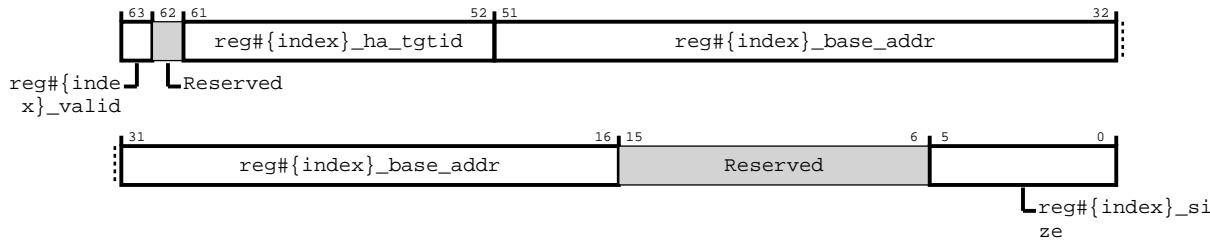
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.rasam\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.rasam\_ctl bit and por\_ccg\_ra\_rcr.rasam\_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 8-54: por\_ccg\_ra\_sam\_addr\_region\_reg0-7**



**Table 8-54: por\_ccg\_ra\_sam\_addr\_region\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63]	reg#{index}_valid	Specifies if the memory region is valid	RW	0b0
[62]	Reserved	Reserved	RO	
[61:52]	reg#{index}_ha_tgtid	Specifies the target HAID	RW	0b0
[51:16]	reg#{index}_base_addr	Specifies the 2^n-aligned base address for the memory region	RW	0x0
[15:6]	Reserved	Reserved	RO	
[5:0]	reg#{index}_size	Specifies the size of the memory region	RW	0b0

### 8.3.3.12 por\_ccg\_ra\_agentid\_to\_linkid\_val

Specifies which Agent ID to Link ID mappings are valid.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.linkid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.linkid\_ctl

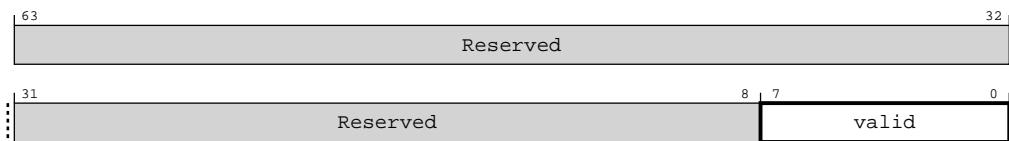
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.linkid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.linkid\_ctl bit and por\_ccg\_ra\_rcr.linkid\_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 8-55: por\_ccg\_ra\_agentid\_to\_linkid\_val**



**Table 8-55: por\_ccg\_ra\_agentid\_to\_linkid\_val attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:0]	valid	Specifies whether the Link ID is valid; bit number corresponds to logical Agent ID number (from 0 to 63)	RW	0x0

### 8.3.3.13 por\_ccg\_ra\_agentid\_to\_linkid\_reg0-0

There are 1 iterations of this register. The index ranges from 0 to 0. Specifies the mapping of Agent ID to Link ID for Agent IDs #{{index}8} to #{{index}8+7}.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD10 + #{{index}\*8}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.linkid\_ctl

## Secure group override

por\_ccg\_ra\_scr.linkid\_ctl

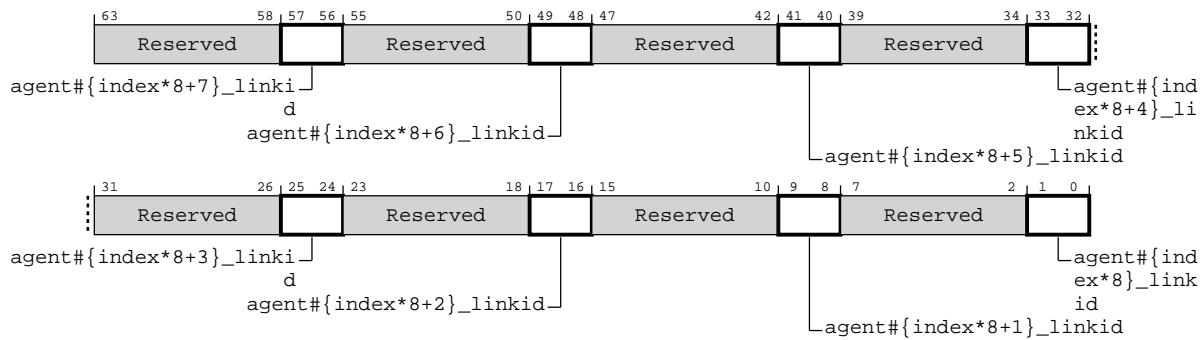
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.linkid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.linkid\_ctl bit and por\_ccg\_ra\_rcr.linkid\_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 8-56: por\_ccg\_ra\_agentid\_to\_linkid\_reg0-0**



**Table 8-56: por\_ccg\_ra\_agentid\_to\_linkid\_reg0-0 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	
[57:56]	agent#{index*8+7}_linkid	Specifies the Link ID for Agent ID #{index*8+7}	RW	0x0
[55:50]	Reserved	Reserved	RO	
[49:48]	agent#{index*8+6}_linkid	Specifies the Link ID for Agent ID #{index*8+6}	RW	0x0
[47:42]	Reserved	Reserved	RO	
[41:40]	agent#{index*8+5}_linkid	Specifies the Link ID for Agent ID #{index*8+5}	RW	0x0
[39:34]	Reserved	Reserved	RO	
[33:32]	agent#{index*8+4}_linkid	Specifies the Link ID for Agent ID #{index*8+4}	RW	0x0
[31:26]	Reserved	Reserved	RO	
[25:24]	agent#{index*8+3}_linkid	Specifies the Link ID for Agent ID #{index*8+3}	RW	0x0
[23:18]	Reserved	Reserved	RO	
[17:16]	agent#{index*8+2}_linkid	Specifies the Link ID for Agent ID #{index*8+2}	RW	0x0
[15:10]	Reserved	Reserved	RO	
[9:8]	agent#{index*8+1}_linkid	Specifies the Link ID for Agent ID #{index*8+1}	RW	0x0
[7:2]	Reserved	Reserved	RO	
[1:0]	agent#{index*8}_linkid	Specifies the Link ID for Agent ID #{index*8}	RW	0x0

### 8.3.3.14 por\_ccg\_ra\_rni\_ldid\_to\_exp\_raid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of RN-I's LDID to Expanded RAID for LDIDs # $\{index4\}$  to # $\{index4+3\}$ . Valid only if CCG\_RA\_RNI\_PRESENT\_PARAM is 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE00 + # $\{index^*8\}$

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

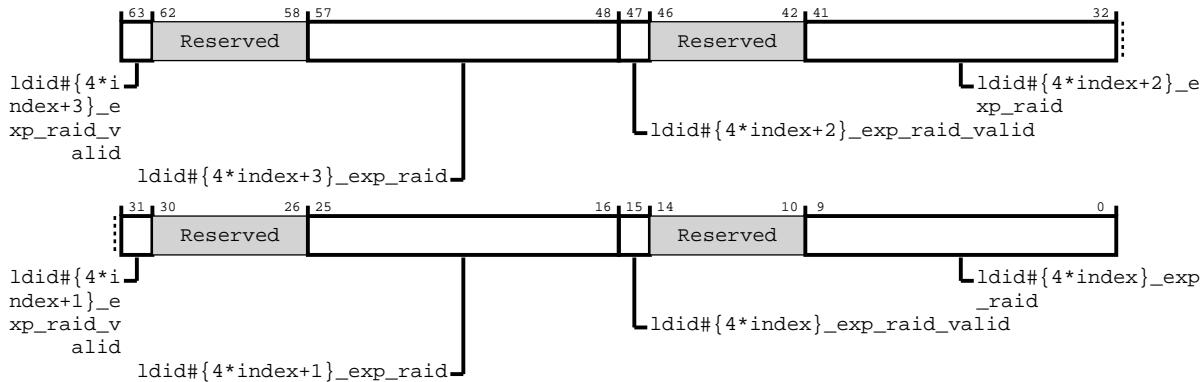
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-57: por\_ccg\_ra\_rni\_ldid\_to\_exp\_raid\_reg0-31**



**Table 8-57: por\_ccg\_ra\_rni\_ldid\_to\_exp\_raid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	ldid#{4*index+3}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+3} is valid;	RW	0x0
[62:58]	Reserved	Reserved	RO	
[57:48]	ldid#{4*index+3}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+3}	RW	0x0
[47]	ldid#{4*index+2}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+2} is valid;	RW	0x0
[46:42]	Reserved	Reserved	RO	
[41:32]	ldid#{4*index+2}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+2}	RW	0x0
[31]	ldid#{4*index+1}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+1} is valid;	RW	0x0
[30:26]	Reserved	Reserved	RO	
[25:16]	ldid#{4*index+1}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+1}	RW	0x0
[15]	ldid#{4*index}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index} is valid;	RW	0x0
[14:10]	Reserved	Reserved	RO	
[9:0]	ldid#{4*index}_exp_raid	Specifies the Expanded RAID for LDID #{4*index}	RW	0x0

### 8.3.3.15 por\_ccg\_ra\_rnd\_ldid\_to\_exp\_raid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of RN-D's LDID to Expanded RAID for LDIDs #{{index}4} to #{{index}4+3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF00 + #{{index}\*8}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

## Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

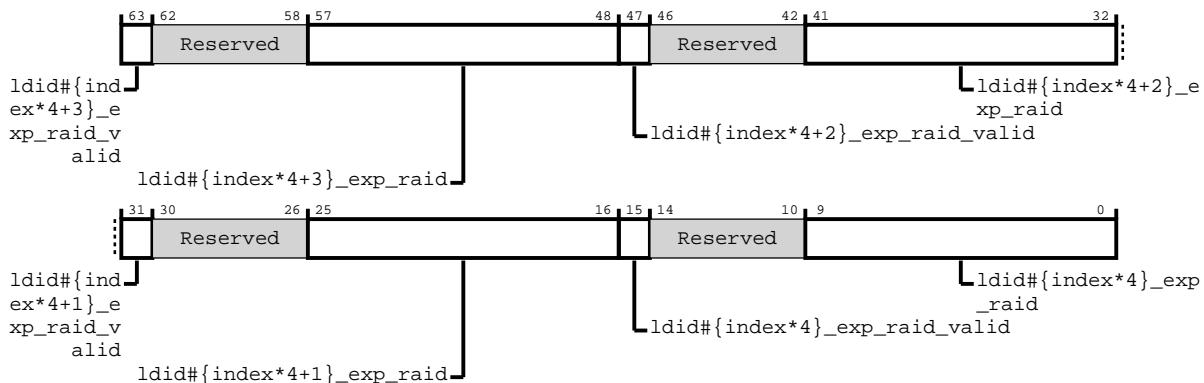
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-58: por\_ccg\_ra\_rnd\_ldid\_to\_exp\_raid\_reg0-31**



**Table 8-58: por\_ccg\_ra\_rnd\_ldid\_to\_exp\_raid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	ldid#[index*4+3]_exp_raid_valid	Specifies whether the Expanded RAID for LDID#[index*4+3] is valid;	RW	0x0
[62:58]	Reserved	Reserved	RO	
[57:48]	ldid#[index*4+3]_exp_raid	Specifies the Expanded RAID for LDID #[index*4+3]	RW	0x0
[47]	ldid#[index*4+2]_exp_raid_valid	Specifies whether the Expanded RAID for LDID#[index*4+2] is valid;	RW	0x0
[46:42]	Reserved	Reserved	RO	
[41:32]	ldid#[index*4+2]_exp_raid	Specifies the Expanded RAID for LDID #[index*4+2]	RW	0x0
[31]	ldid#[index*4+1]_exp_raid_valid	Specifies whether the Expanded RAID for LDID#[index*4+1] is valid;	RW	0x0
[30:26]	Reserved	Reserved	RO	
[25:16]	ldid#[index*4+1]_exp_raid	Specifies the Expanded RAID for LDID #[index*4+1]	RW	0x0

Bits	Name	Description	Type	Reset
[15]	lDid#[index*4]_exp_raid_valid	Specifies whether the Expanded RAID for LDID#[index*4] is valid;	RW	0x0
[14:10]	Reserved	Reserved	RO	
[9:0]	lDid#[index*4]_exp_raid	Specifies the Expanded RAID for LDID #[index*4]	RW	0x0

### 8.3.3.16 por\_ccg\_ra\_rnf\_lDid\_to\_exp\_raid\_reg0-127

There are 128 iterations of this register. The index ranges from 0 to 127. Specifies the mapping of RN-F's LDID to Expanded RAID for LDIDs #[index4] to #[index4+3].

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1000 + #{index\*8}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.lDid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.lDid\_ctl

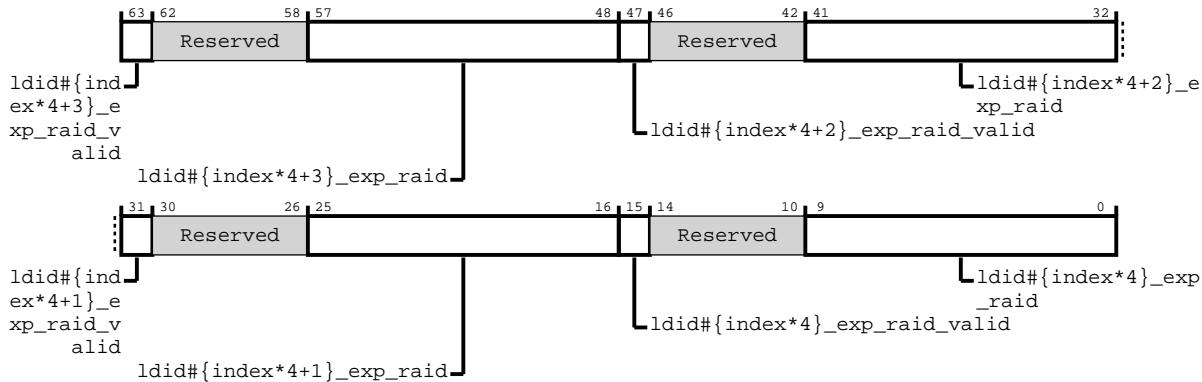
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.lDid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.lDid\_ctl bit and por\_ccg\_ra\_rcr.lDid\_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 8-59: por\_ccg\_ra\_rnf\_ldid\_to\_exp\_raid\_reg0-127**



**Table 8-59: por\_ccg\_ra\_rnf\_ldid\_to\_exp\_raid\_reg0-127 attributes**

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+3} is valid;	RW	0x0
[62:58]	Reserved	Reserved	RO	
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	0x0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+2} is valid;	RW	0x0
[46:42]	Reserved	Reserved	RO	
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	0x0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+1} is valid;	RW	0x0
[30:26]	Reserved	Reserved	RO	
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	0x0
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	0x0
[14:10]	Reserved	Reserved	RO	
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	0x0

### 8.3.3.17 por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_reg0-63

There are 64 iterations of this register. The index ranges from 0 to 63. Specifies the mapping of HA's LDID to Expanded RAID for LDIDs #{index\*4} to #{index\*4+3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1400 + #{index\*8}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

## Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

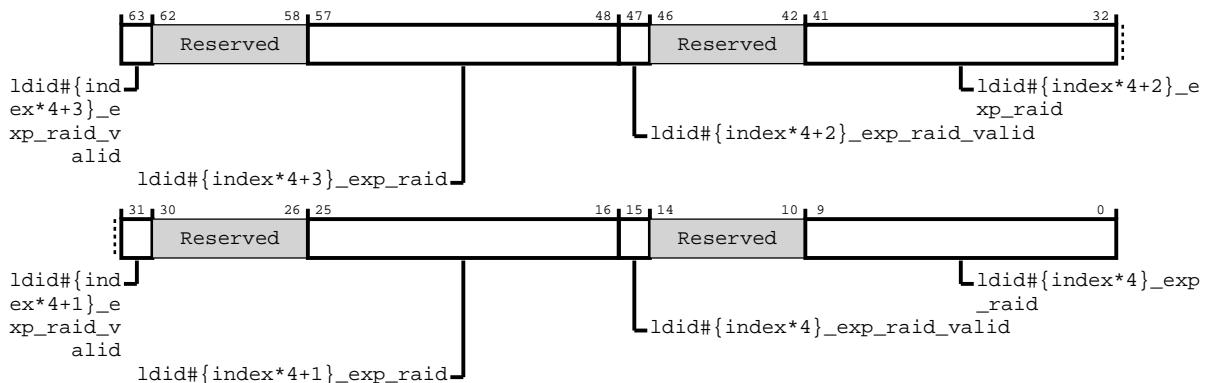
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-60: por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_reg0-63**



**Table 8-60: por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_reg0-63 attributes**

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+3} is valid;	RW	0x0
[62:58]	Reserved	Reserved	RO	
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	0x0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+2} is valid;	RW	0x0
[46:42]	Reserved	Reserved	RO	
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	0x0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+1} is valid;	RW	0x0
[30:26]	Reserved	Reserved	RO	
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	0x0

Bits	Name	Description	Type	Reset
[15]	lDid#[index*4]_exp_raid_valid	Specifies whether the look table entry for default LDID#[index*4] is valid;	RW	0x0
[14:10]	Reserved	Reserved	RO	
[9:0]	lDid#[index*4]_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	0x0

### 8.3.3.18 por\_ccg\_ra\_hns\_ldid\_to\_exp\_raid\_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Specifies the mapping of HNS's LDID to Expanded RAID for LDIDs #{index4} to #{index4+3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1600 + #{index\*8}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.lDid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.lDid\_ctl

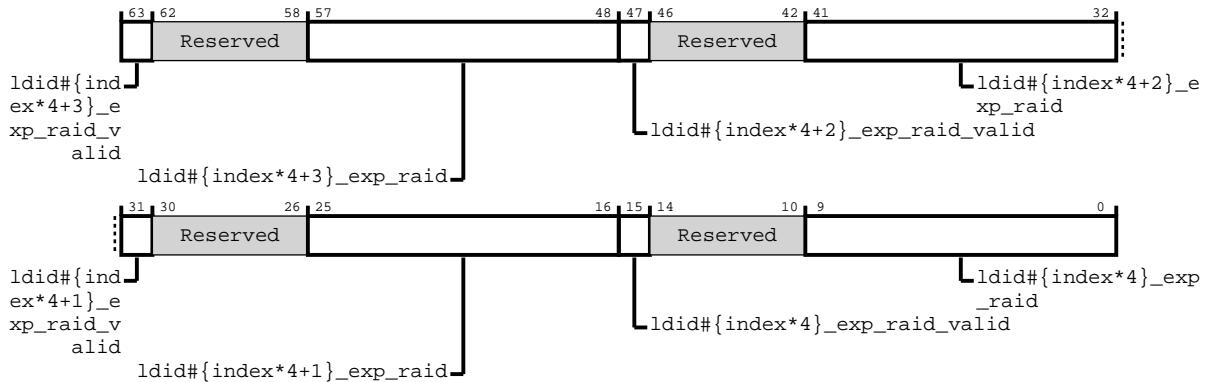
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.lDid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.lDid\_ctl bit and por\_ccg\_ra\_rcr.lDid\_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 8-61: por\_ccg\_ra\_hns\_ldid\_to\_exp\_raid\_reg0-15**



**Table 8-61: por\_ccg\_ra\_hns\_ldid\_to\_exp\_raid\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+3} is valid;	RW	0x0
[62:58]	Reserved	Reserved	RO	
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	0x0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+2} is valid;	RW	0x0
[46:42]	Reserved	Reserved	RO	
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	0x0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+1} is valid;	RW	0x0
[30:26]	Reserved	Reserved	RO	
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	0x0
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	0x0
[14:10]	Reserved	Reserved	RO	
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	0x0

### 8.3.3.19 por\_ccg\_ra\_rnf\_ldid\_to\_ovrd\_ldid\_reg0-127

There are 128 iterations of this register. The index ranges from 0 to 127. Specifies the mapping of RN-F's overridden LDID for default LDIDs #{index4} to #{index4+3}.Valid only if POR\_MXP\_RNF\_CLUSTER\_EN\_PARAM is 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

$0x1680 + \{\text{index}\} * 8$

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_ccg_ra_rcr.ldid_ctl`

### Secure group override

`por_ccg_ra_scr.ldid_ctl`

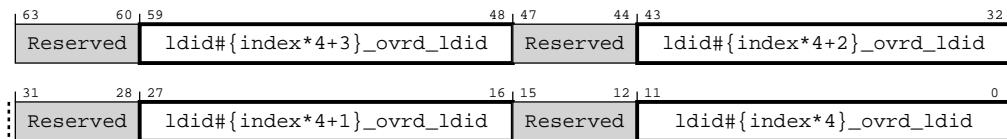
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.ldid_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.ldid_ctl` bit and `por_ccg_ra_rcr.ldid_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 8-62: por\_ccg\_ra\_rnf\_ldid\_to\_ovrd\_ldid\_reg0-127**



**Table 8-62: por\_ccg\_ra\_rnf\_ldid\_to\_ovrd\_ldid\_reg0-127 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	ldid#[index*4+3]_ovrd_ldid	Specifies the Overridden LDID for Default LDID #[index*4+3]	RW	Configuration dependent
[47:44]	Reserved	Reserved	RO	-
[43:32]	ldid#[index*4+2]_ovrd_ldid	Specifies the Overridden LDID for Default LDID #[index*4+2]	RW	Configuration dependent
[31:28]	Reserved	Reserved	RO	-
[27:16]	ldid#[index*4+1]_ovrd_ldid	Specifies the Overridden LDID for Default LDID #[index*4+1]	RW	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:0]	ldid#[index*4]_ovrd_ldid	Specifies the Overridden LDID for Default LDID #[index*4]	RW	Configuration dependent

### 8.3.3.20 por\_ccg\_ra\_cml\_port\_aggr\_grp0-31\_addr\_mask

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CCIX port aggregation address mask for group 0.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1A80 + #8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

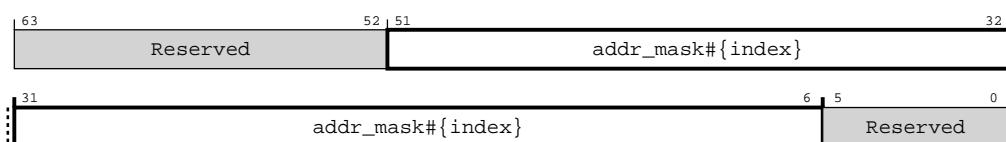
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-63: por\_ccg\_ra\_cml\_port\_aggr\_grp0-31\_addr\_mask**



**Table 8-63: por\_ccg\_ra\_cml\_port\_aggr\_grp0-31\_addr\_mask attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:6]	addr_mask#[index]	Address mask to be applied before hashing	RW	0xFFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

### 8.3.3.21 por\_ccg\_ra\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1B80 + #{index\*8}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

##### Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

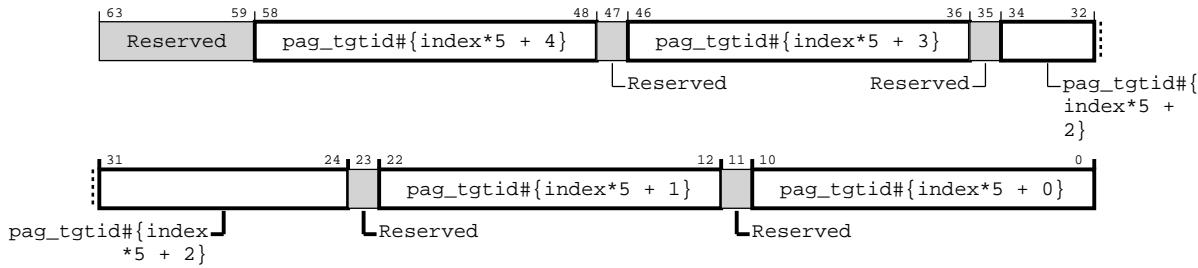
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-64: por\_ccg\_ra\_cml\_port\_aggr\_grp\_reg0-12**



**Table 8-64: por\_ccg\_ra\_cml\_port\_aggr\_grp\_reg0-12 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	
[58:48]	pag_tgtid#{index*5 + 4}	Specifies the target ID #{index*5 + 4} for CPAG	RW	0b0
[47]	Reserved	Reserved	RO	
[46:36]	pag_tgtid#{index*5 + 3}	Specifies the target ID #{index*5 + 3} for CPAG	RW	0b0
[35]	Reserved	Reserved	RO	
[34:24]	pag_tgtid#{index*5 + 2}	Specifies the target ID {index*5 + 2} for CPAG	RW	0b0
[23]	Reserved	Reserved	RO	
[22:12]	pag_tgtid#{index*5 + 1}	Specifies the target ID {index*5 + 1} for CPAG	RW	0b0
[11]	Reserved	Reserved	RO	
[10:0]	pag_tgtid#{index*5 + 0}	Specifies the target ID {index*5 + 0} for CPAG	RW	0b0

### 8.3.3.22 por\_ccg\_ra\_cml\_port\_aggr\_ctrl\_reg0-6

There are 7 iterations of this register. The index ranges from 0 to 6. Configures the CCIX port aggregation port groups

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1BE8 + #{index\*8}

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

`por_ccg_ra_rcr.ldid_ctl`

### Secure group override

`por_ccg_ra_scr.ldid_ctl`

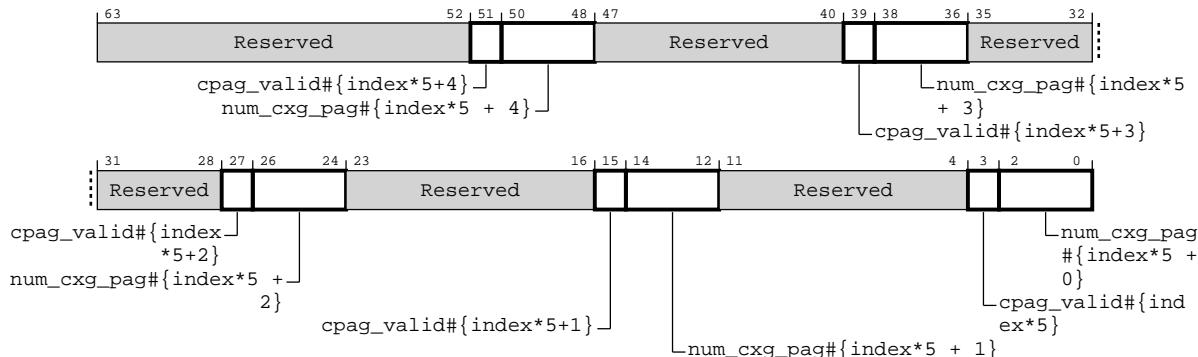
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.ldid_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.ldid_ctl` bit and `por_ccg_ra_rcr.ldid_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 8-65: por\_ccg\_ra\_cml\_port\_aggr\_ctrl\_reg0-6**



**Table 8-65: por\_ccg\_ra\_cml\_port\_aggr\_ctrl\_reg0-6 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	cpag_valid#{index*5+4}	Valid programming for CPAG #{index*5 + 4}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag#{index*5 + 4}	<p>Specifies the number of CXRAs in CPAG4#{index*5 + 4}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid8 through pag_tgtid9 of por_ccg_ra_cml_port_aggr_ctrl_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{index*5 + 3}, Enabled by default (backward compatible)	RW	0b1
[38:36]	num_cxg_pag#{index*5 + 3}	<p>Specifies the number of CXRAs in CPAG3#{index*5 + 3}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid6 through pag_tgtid7 of por_ccg_ra_cml_port_aggr_ctrl_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[35:28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{index*5 + 2}, Enabled by default (backward compatible)	RW	0b1
[26:24]	num_cxg_pag#{index*5 + 2}	<p>Specifies the number of CXRAs in CPAG2#{index*5 + 2}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid4 through pag_tgtid7 of por_ccg_ra_cml_port_aggr_ctrl_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{index*5 + 1}, Enabled by default (backward compatible)	RW	0b1
[14:12]	num_cxg_pag#{index*5 + 1}	<p>Specifies the number of CXRAs in CPAG1#{index*5 + 1}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid2 through pag_tgtid3 of por_ccg_ra_cml_port_aggr_ctrl_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

Bits	Name	Description	Type	Reset
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	0b1
[2:0]	num_cxg_pag#{index*5 + 0}	<p>Specifies the number of CXRAs in CPAG#{index*5 + 0}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid0 through pag_tgtid7 of por_ccg_ra_cml_port_aggr_ctrl_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

### 8.3.3.23 por\_ccg\_ra\_ha\_ldid\_to\_ovrd\_ldid\_reg0-63

There are 64 iterations of this register. The index ranges from 0 to 63. Specifies the mapping of HA's RN-F overridden LDID for default LDIDs #{index4} to #{index4+3}. Valid only if POR\_MXP\_RNF\_CLUSTER\_EN\_PARAM is 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1C20 + #{index\*8}

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

### Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

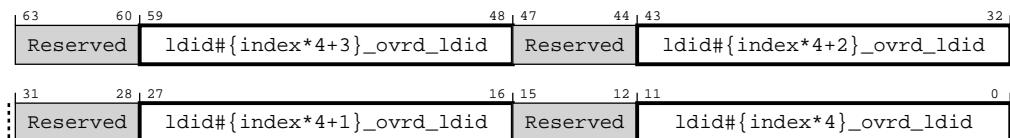
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-66: por\_ccg\_ra\_ha\_ldid\_to\_ovrd\_ldid\_reg0-63**



**Table 8-66: por\_ccg\_ra\_ha\_ldid\_to\_ovrd\_ldid\_reg0-63 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	ldid#{index*4+3}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+3}	RW	Configuration dependent
[47:44]	Reserved	Reserved	RO	-
[43:32]	ldid#{index*4+2}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+2}	RW	Configuration dependent
[31:28]	Reserved	Reserved	RO	-
[27:16]	ldid#{index*4+1}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+1}	RW	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:0]	ldid#{index*4}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4}	RW	Configuration dependent

### 8.3.3.24 por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_cpagen\_reg0-63

There are 64 iterations of this register. The index ranges from 0 to 63. Specifies for a given raid to ldid cpag is enable, if not then it provides physical nodeid

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1E38 + #{index\*8}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.ldid\_ctl

### Secure group override

por\_ccg\_ra\_scr.ldid\_ctl

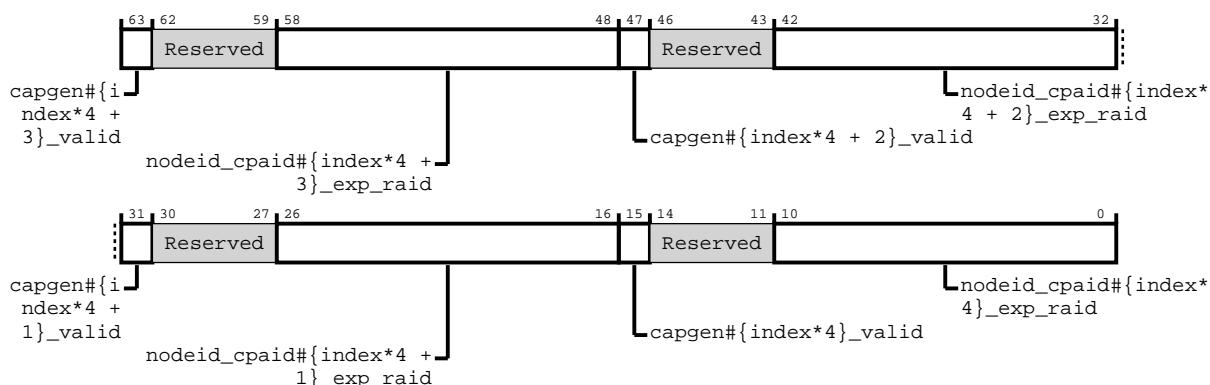
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.ldid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.ldid\_ctl bit and por\_ccg\_ra\_rcr.ldid\_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 8-67: por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_cpagen\_reg0-63**



**Table 8-67: por\_ccg\_ra\_ha\_ldid\_to\_exp\_raid\_cpagen\_reg0-63 attributes**

Bits	Name	Description	Type	Reset
[63]	capgen#[index*4 + 3]_valid	Specifies whether the look table entry for default LDID#[index*4] is valid;	RW	0x0
[62:59]	Reserved	Reserved	RO	
[58:48]	nodeid_cpaid#{index*4 + 3}_exp_raid	Specifies the CPAGID if capgen#[index*4]_valid is 0b1 else specifies nodeid	RW	0x0

Bits	Name	Description	Type	Reset
[47]	capgen#{index*4 + 2}_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	0x0
[46:43]	Reserved	Reserved	RO	
[42:32]	nodeid_cpaid#{index*4 + 2}_exp_raid	Specifies the CPAGID if capgen#{index*4}_valid is 0b1 else specifies nodeid	RW	0x0
[31]	capgen#{index*4 + 1}_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	0x0
[30:27]	Reserved	Reserved	RO	
[26:16]	nodeid_cpaid#{index*4 + 1}_exp_raid	Specifies the CPAGID if capgen#{index*4}_valid is 0b1 else specifies nodeid	RW	0x0
[15]	capgen#{index*4}_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	0x0
[14:11]	Reserved	Reserved	RO	
[10:0]	nodeid_cpaid#{index*4}_exp_raid	Specifies the CPAGID if capgen#{index*4}_valid is 0b1 else specifies nodeid	RW	0x0

### 8.3.3.25 por\_ccg\_ra\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

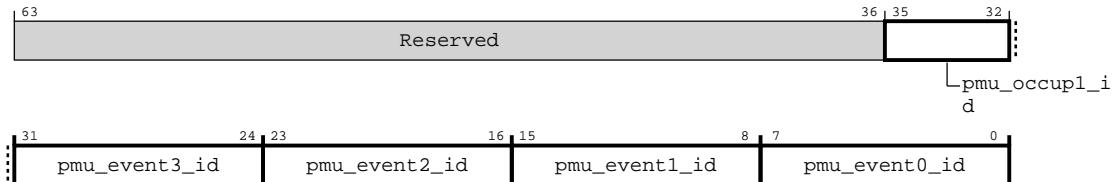
##### Usage constraints

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 8-68: por\_ccg\_ra\_pmu\_event\_sel**



**Table 8-68: por\_ccg\_ra\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35:32]	pmu_occup1_id	PMU occupancy event selector ID	RW	0b0
[31:24]	pmu_event3_id	CXRA PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[23:16]	pmu_event2_id	CXRA PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[15:8]	pmu_event1_id	CXRA PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	<p>CXRA PMU Event 0 ID</p> <p><b>0x00</b> No event</p> <p><b>0x41</b> Request Tracker (RHT) occupancy count overflow</p> <p><b>0x42</b> Snoop Tracker (SHT) occupancy count overflow</p> <p><b>0x43</b> Read Data Buffer (RDB) occupancy count overflow</p> <p><b>0x44</b> Write Data Buffer (WDB) occupancy count overflow</p> <p><b>0x45</b> Snoop Sink Buffer (SSB) occupancy count overflow</p> <p><b>0x46</b> CCIX RX broadcast snoops</p> <p><b>0x47</b> CCIX TX request chain</p> <p><b>0x48</b> CCIX TX request chain average length</p> <p><b>0x49</b> CHI internal RSP stall on Port0</p> <p><b>0x4A</b> CHI internal DAT stall</p> <p><b>0x4B</b> CCIX REQ Protocol credit Link 0 stall</p> <p><b>0x4C</b> CCIX REQ Protocol credit Link 1 stall</p> <p><b>0x4D</b> CCIX REQ Protocol credit Link 2 stall</p> <p><b>0x4E</b> CCIX DAT Protocol credit Link 0 stall</p> <p><b>0x4F</b> CCIX DAT Protocol credit Link 1 stall</p> <p><b>0x50</b> CCIX DAT Protocol credit Link 2 stall</p> <p><b>0x51</b> CHI external RSP stall on Port0</p> <p><b>0x52</b> CHI external DAT stall</p> <p><b>0x53</b> CCIX MISC Protocol credit Link 0 stall</p>	RW	0b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	<p>CXRA PMU Event 0 ID</p> <p><b>0x54</b> CCIX MISC Protocol credit Link 1 stall</p> <p><b>0x55</b> CCIX MISC Protocol credit Link 2 stall</p> <p><b>0x56</b> Request Tracker (RHT) allocations</p> <p><b>0x57</b> Snoop Tracker (SHT) allocations</p> <p><b>0x58</b> Read Data Buffer (RDB) allocations</p> <p><b>0x59</b> Write Data Buffer (WDB) allocations</p> <p><b>0x5A</b> Snoop Sink Buffer (SSB) allocations</p> <p><b>0x5B</b> CHI internal RSP stall on Port1</p> <p><b>0x5C</b> CHI external RSP stall on Port1</p> <p><b>0x5D</b> CHI Write requests with all data bits set to 0.</p> <p><b>0x5E</b> CHI Read requests with all data bits set to 0.</p>	RW	0b0

### 8.3.3.26 por\_ccg\_ra\_ccprtcl\_link0\_ctl

Functions as the CXRA CCIX Protocol Link 0 control register. Works with por\_ccg\_ra\_ccprtcl\_link0\_status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x4000

##### Type

RW

##### Reset value

See individual bit resets

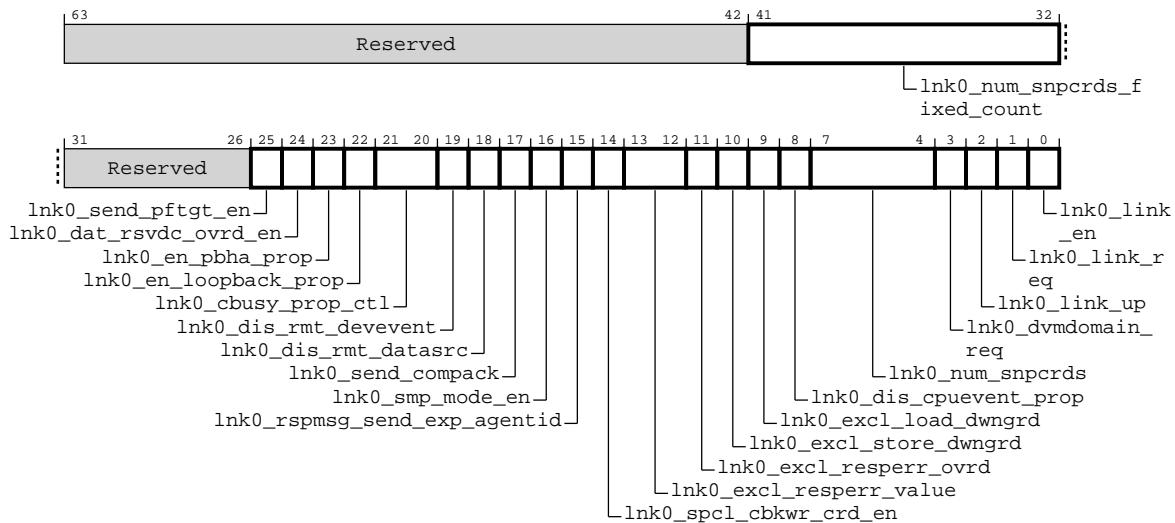
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-69: por\_ccg\_ra\_ccprtcl\_link0\_ctl**



**Table 8-69: por\_ccg\_ra\_ccprtcl\_link0\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41:32]	lnk0_num_snpcrds_fixed_count	Controls the number of CCIX snoop credits assigned to Link 0 If this field along with link1, link2 fixed credit count is zero then the hardware uses lnk0_num_snpcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_snpcrds_fixed_count should not exceed snoop_tracker_depth field which is in por_ccg_ra_unit_info. If the sum exceeds snoop_tracker_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-
[25]	lnk0_send_pftgt_en	When set, enables sending Prefetch Target (CHI) or MemSpecRd (CXL) over link 0.	RW	0b1
[24]	lnk0_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 0.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode)	RW	0b0
[23]	lnk0_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 0.	RW	0b1
[22]	lnk0_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 0.	RW	0b1

Bits	Name	Description	Type	Reset
[21:20]	lnk0_cbusy_prop_ctl	<p>Controls the propagation of Cbusy field for CCIX Link 0.</p> <p><b>0b00</b> Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl</p> <p><b>0b01</b> Pass through remote CBusy on late completion responses (CompData, Comp)</p> <p><b>0b10</b> Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent</p> <p><b>NOTE</b> This field is applicable SMP and CXL modes</p>	RW	0b0
[19]	lnk0_dis_rmt_devevent	<p>When set, disables propagation of remote Dev Event field for CCIX Link 0.</p> <p><b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)</p>	RW	0b0
[18]	lnk0_dis_rmt_datasrc	<p>This field is deprecated from CHIG. When set, disables propagation of remote data source for CCIX Link 0.</p> <p><b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)</p>	RW	0b0
[17]	lnk0_send_compack	When set, sends CompAck for CCIX Link 0.	RW	0b0
[16]	lnk0_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 0.	RW	0x0
[15]	lnk0_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 0	RW	0b0
[14]	lnk0_spcl_cbkwr_crd_en	<p>When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 0</p> <p><b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)</p>	RW	0b0
[13:12]	lnk0_excl_resperr_value	<p>Two bit value to override RespErr field of an exclusive response. Applicable only if lnk0_excl_resperr_ovrd bit is set.</p> <p><b>NOTE</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[11]	lnk0_excl_resperr_ovrd	<p>When set, overrides the RespErr field of exclusive response with the lnk0_excl_resperr_value field</p> <p><b>NOTE</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0

Bits	Name	Description	Type	Reset
[10]	lnk0_excl_store_dwgnd	<p>When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 0</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[9]	lnk0_excl_load_dwgnd	<p>When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 0</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[8]	lnk0_dis_cpuevent_prop	<p>When set, disables the propagation of CPU Events on CCIX Link 0</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable is set.</p>	RW	0b0
[7:4]	lnk0_num_snpcrds	<p>Controls the number of CCIX snoop credits assigned to Link 0</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk0_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 0	RW	0b0
[2]	lnk0_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0

Bits	Name	Description	Type	Reset
[1]	lnk0_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk0_link_en	<p>Enables CCIX Link 0 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.3.27 por\_ccg\_ra\_ccprtcl\_link0\_status

Functions as the CXRA CCIX Protocol Link 0 status register. Works with por\_ccg\_ra\_ccprtcl\_link0\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x4008

##### Type

RO

##### Reset value

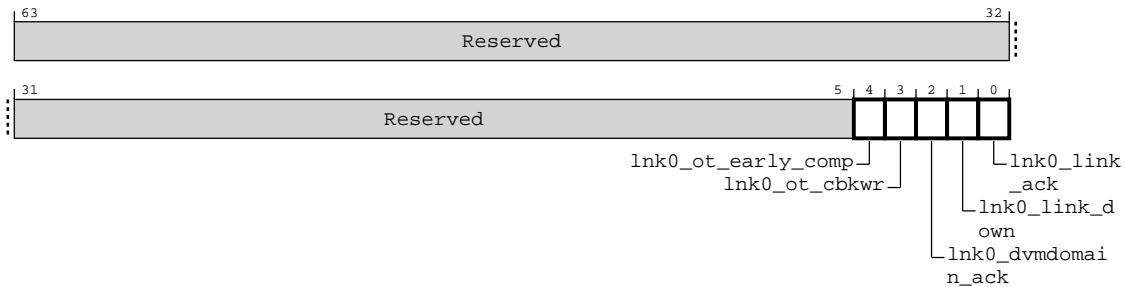
See individual bit resets

#### Usage constraints

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-70: por\_ccg\_ra\_ccprtcl\_link0\_status**



**Table 8-70: por\_ccg\_ra\_ccprtcl\_link0\_status attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	
[4]	lnk0_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link0	RO	0b0
[3]	lnk0_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link0	RO	0b0
[2]	lnk0_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 0	RO	0b0
[1]	lnk0_link_down	Link Down status; hardware updates this register bit to indicate Link Down status <b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request <b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	0b1
[0]	lnk0_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request <b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops granting protocol credits and starts returning protocol credits to the remote agent when Link_ACK is clear <b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent <b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.	RO	0b0

### 8.3.3.28 por\_ccg\_ra\_ccprtcl\_link1\_ctl

Functions as the CXRA CCIX Protocol Link 1 control register. Works with por\_ccg\_ra\_ccprtcl\_link1\_status.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x4010

### Type

RW

### Reset value

See individual bit resets

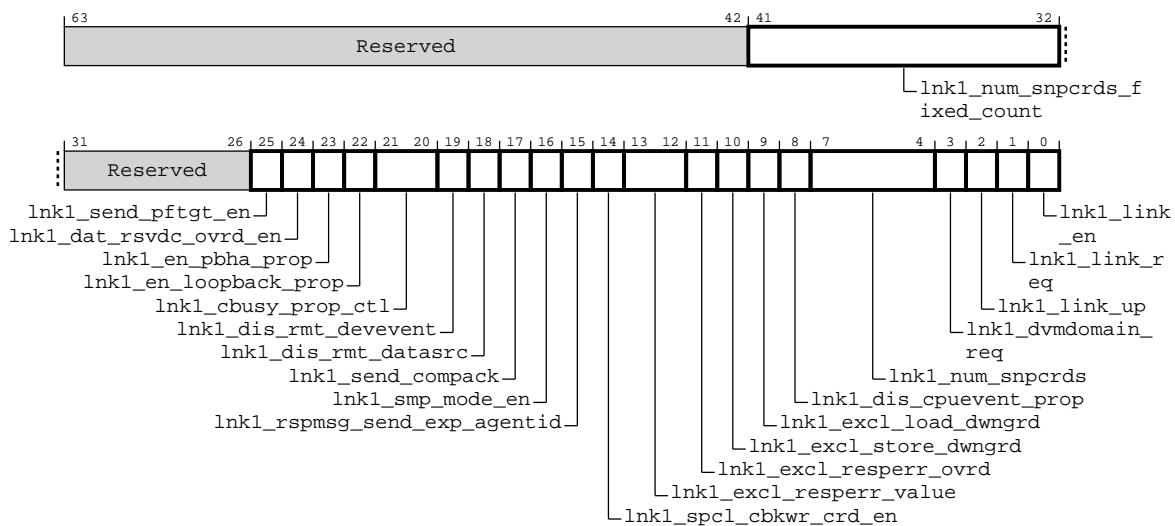
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-71: por\_ccg\_ra\_ccprtcl\_link1\_ctl**



**Table 8-71: por\_ccg\_ra\_ccprtcl\_link1\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41:32]	lnk1_num_snpcrds_fixed_count	Controls the number of CCIX snoop credits assigned to Link 1. If this field along with link0, link2 fixed credit count is zero then the hardware uses lnk1_num_snpcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_snpcrds_fixed_count should not exceed snoop_tracker_depth field which is in por_ccg_ra_unit_info. if the sum exceeds snoop_tracker_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25]	lnk1_send_pftgt_en	When set, enables sending Prefetch Target (CHI) over link 1.  <b>Note</b> This field is not-applicable in CXL mode for link1	RW	0b1
[24]	lnk1_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 1.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode)	RW	0b0
[23]	lnk1_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 1.	RW	0b1
[22]	lnk1_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 1.	RW	0b1
[21:20]	lnk1_cbusy_prop_ctl	Controls the propagation of Cbusy field for CCIX Link 1.  <b>0b00</b> Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl  <b>0b01</b> Pass through remote CBusy on late completion responses (CompData, Comp)  <b>0b10</b> Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[19]	lnk1_dis_rmt_deevent	When set, disables propagation of remote Dev Event field for CCIX Link 1.  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[18]	lnk1_dis_rmt_datasrc	When set, disables propagation of remote data source for CCIX Link 1.  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[17]	lnk1_send_compack	When set, sends CompAck for CCIX Link 1.	RW	0b0
[16]	lnk1_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 1.	RW	0x0
[15]	lnk1_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 1	RW	0b0
[14]	lnk1_spcl_cbkwr_crd_en	When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 1  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[13:12]	lnk1_excl_resperr_value	Two bit value to override RespErr field of an exclusive response. Applicable only if lnk1_excl_resperr_ovrd bit is set.  <b>NOTE</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	0b0

Bits	Name	Description	Type	Reset
[11]	lnk1_excl_resperr_ovrd	<p>When set, overrides the RespErr field of exclusive response with the lnk1_excl_resperr_value field</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[10]	lnk1_excl_store_dwngrd	<p>When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 1</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[9]	lnk1_excl_load_dwngrd	<p>When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 1</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[8]	lnk1_dis_cpuevent_prop	<p>When set, disables the propagation of CPU Events on CCIX Link 1</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable parameter is set.</p>	RW	0b0
[7:4]	lnk1_num_snpcrds	<p>Controls the number of CCIX snoop credits assigned to Link 1</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk1_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 1	RW	0b0
[2]	lnk1_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0

Bits	Name	Description	Type	Reset
[1]	lnk1_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk1_link_en	<p>Enables CCIX Link 1 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.3.29 por\_ccg\_ra\_ccprtcl\_link1\_status

Functions as the CXRA CCIX Protocol Link 1 status register. Works with por\_ccg\_ra\_ccprtcl\_link1\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x4018

##### Type

RO

##### Reset value

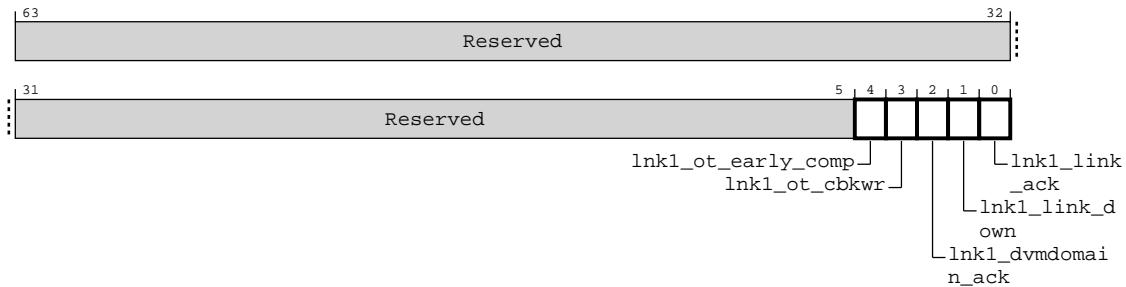
See individual bit resets

#### Usage constraints

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-72: por\_ccg\_ra\_ccprtcl\_link1\_status**



**Table 8-72: por\_ccg\_ra\_ccprtcl\_link1\_status attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	
[4]	lnk1_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link1	RO	0b0
[3]	lnk1_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link1	RO	0b0
[2]	lnk1_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 1	RO	0b0
[1]	lnk1_link_down	<p>Link Down status; hardware updates this register bit to indicate Link Down status</p> <p><b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request</p> <p><b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear</p>	RO	0b1
[0]	lnk1_link_ack	<p>Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request</p> <p><b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear</p> <p><b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent</p> <p><b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.</p>	RO	0b0

### 8.3.3.30 por\_ccg\_ra\_ccprtcl\_link2\_ctl

Functions as the CXRA CCIX Protocol Link 2 control register. Works with por\_ccg\_ra\_ccprtcl\_link2\_status.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x4020

### Type

RW

### Reset value

See individual bit resets

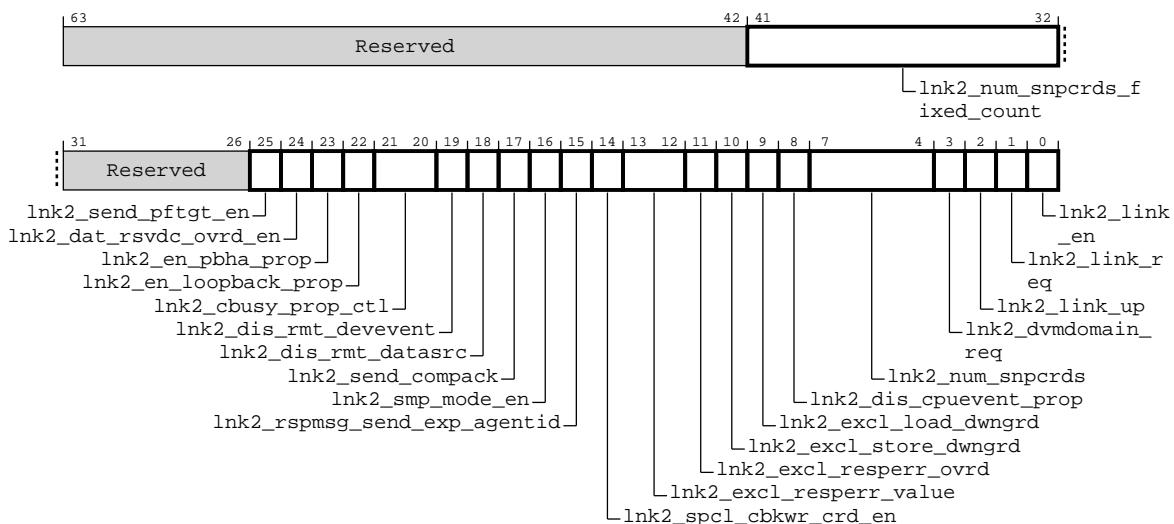
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-73: por\_ccg\_ra\_ccprtcl\_link2\_ctl**



**Table 8-73: por\_ccg\_ra\_ccprtcl\_link2\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41:32]	lnk2_num_snpcrds_fixed_count	Controls the number of CCIX snoop credits assigned to Link 2 If this field along with link0, link1 fixed credit count is zero then the hardware uses lnk0_num_snpcrds field to assign percentage based credits. lnk0+lnk1+lnk2 num_snpcrds_fixed_count should not exceed snoop_tracker_depth field which is in por_ccg_ra_unit_info. if the sum exceeds snoop_tracker_depth then it will cause fatal errors	RW	0x00
[31:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25]	lnk2_send_pftgt_en	When set, enables sending Prefetch Target (CHI) over link 2.  <b>Note</b> This field is not-applicable in CXL mode for link2	RW	0b1
[24]	lnk2_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 2.  <b>Note</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode)	RW	0b0
[23]	lnk2_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 2.	RW	0b1
[22]	lnk2_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 2.	RW	0b1
[21:20]	lnk2_cbusy_prop_ctl	Controls the propagation of Cbusy field for CCIX Link 2.  <b>0b00</b> Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl  <b>0b01</b> Pass through remote CBusy on late completion responses (CompData, Comp)  <b>0b10</b> Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[19]	lnk2_dis_rmt_deevent	When set, disables propagation of remote Dev Event field for CCIX Link 2.  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[18]	lnk2_dis_rmt_datasrc	When set, disables propagation of remote data source for CCIX Link 2.  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[17]	lnk2_send_compack	When set, sends CompAck for CCIX Link 2.	RW	0b0
[16]	lnk2_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 2.	RW	0x0
[15]	lnk2_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 2	RW	0b0
[14]	lnk2_spcl_cbkwr_crd_en	When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 2  <b>NOTE</b> This field is applicable only if the link programmed for SMP mode (index.e. SMP Mode enable bit is set)	RW	0b0
[13:12]	lnk2_excl_resperr_value	Two bit value to override RespErr field of an exclusive response. Applicable only if lnk2_excl_resperr_ovrd bit is set.  <b>NOTE</b> This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	0b0

Bits	Name	Description	Type	Reset
[11]	lnk2_excl_resperr_ovrd	<p>When set, overrides the RespErr field of exclusive response with the lnk2_excl_resperr_value field</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[10]	lnk2_excl_store_dwngrd	<p>When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 2</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[9]	lnk2_excl_load_dwngrd	<p>When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 2</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable bit is clear (index.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring</p>	RW	0b0
[8]	lnk2_dis_cpuevent_prop	<p>When set, disables the propagation of CPU Events on CCIX Link 2</p> <p><b>NOTE</b></p> <p>This field is applicable only when SMP Mode enable parameter is set.</p>	RW	0b0
[7:4]	lnk2_num_snpcrds	<p>Controls the number of CCIX snoop credits assigned to Link 2</p> <p><b>0x0</b> Total credits are equally divided across all links</p> <p><b>0x1</b> 25% of credits assigned</p> <p><b>0x2</b> 50% of credits assigned</p> <p><b>0x3</b> 75% of credits assigned</p> <p><b>0x4</b> 100% of credits assigned</p> <p><b>0xF</b> 0% of credits assigned</p>	RW	0b0
[3]	lnk2_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 2	RW	0b0
[2]	lnk2_link_up	<p>Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent</p> <p><b>0b0</b> Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear</p> <p><b>0b1</b> Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent</p>	RW	0b0

Bits	Name	Description	Type	Reset
[1]	lnk2_link_req	<p>Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent</p> <p><b>0b0</b> Link Down request</p> <p><b>NOTE</b> The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.</p> <p><b>0b1</b> Link Up request</p>	RW	0b0
[0]	lnk2_link_en	<p>Enables CCIX Link 2 when set</p> <p><b>0b0</b> Link is disabled</p> <p><b>0b1</b> Link is enabled</p>	RW	0b0

### 8.3.3.31 por\_ccg\_ra\_ccprtcl\_link2\_status

Functions as the CXRA CCIX Protocol Link 2 status register. Works with por\_ccg\_ra\_ccprtcl\_link2\_ctl.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x4028

##### Type

RO

##### Reset value

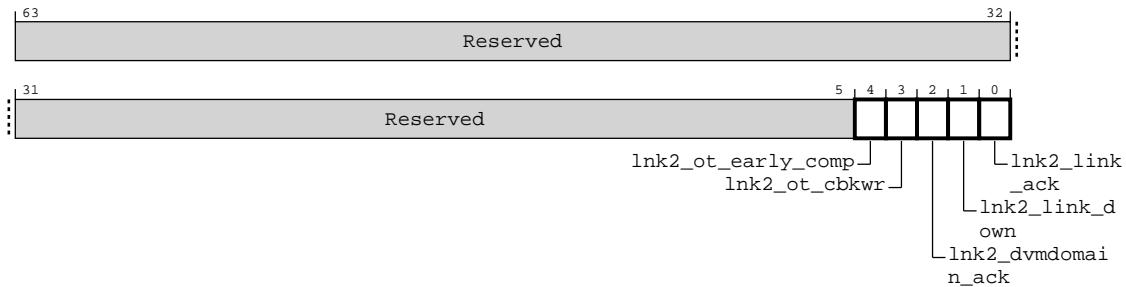
See individual bit resets

#### Usage constraints

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-74: por\_ccg\_ra\_ccprtcl\_link2\_status**



**Table 8-74: por\_ccg\_ra\_ccprtcl\_link2\_status attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	
[4]	lnk2_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link2	RO	0b0
[3]	lnk2_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link2	RO	0b0
[2]	lnk2_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 2	RO	0b0
[1]	lnk2_link_down	Link Down status; hardware updates this register bit to indicate Link Down status <b>0b0</b> Link is not Down; hardware clears Link_DN when it receives a Link Up request <b>0b1</b> Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	0b1
[0]	lnk2_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request <b>0b0</b> Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear <b>0b1</b> Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent <b>NOTE</b> The local agent must clear Link_DN before setting Link_ACK.	RO	0b0

### 8.3.3.32 ccg\_ra\_c2c\_non\_caching\_agent\_id

It specifies the RAID for non Caching Agents. The max limit is 128. This ID needs to be different than RA Caching AgentID

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x4030

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

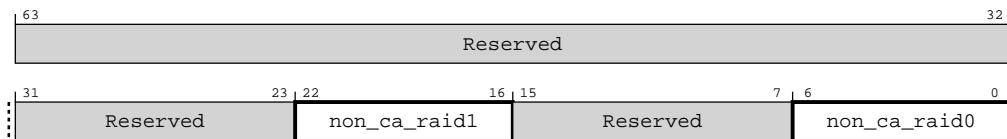
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-75: ccg\_ra\_c2c\_non\_caching\_agent\_id**



**Table 8-75: ccg\_ra\_c2c\_non\_caching\_agent\_id attributes**

Bits	Name	Description	Type	Reset
[63:23]	Reserved	Reserved	RO	-
[22:16]	non_ca_raid1	RAID for non Caching Agents	RW	Configuration dependent
[15:7]	Reserved	Reserved	RO	-
[6:0]	non_ca_raid0	RAID for non Caching Agents	RW	Configuration dependent

### 8.3.3.33 por\_ccg\_ra\_datasource\_ctl\_link0

Control to determine the chi-g datasource on SnpRespData (Link-0)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA20

##### Type

RW

##### Reset value

See individual bit resets

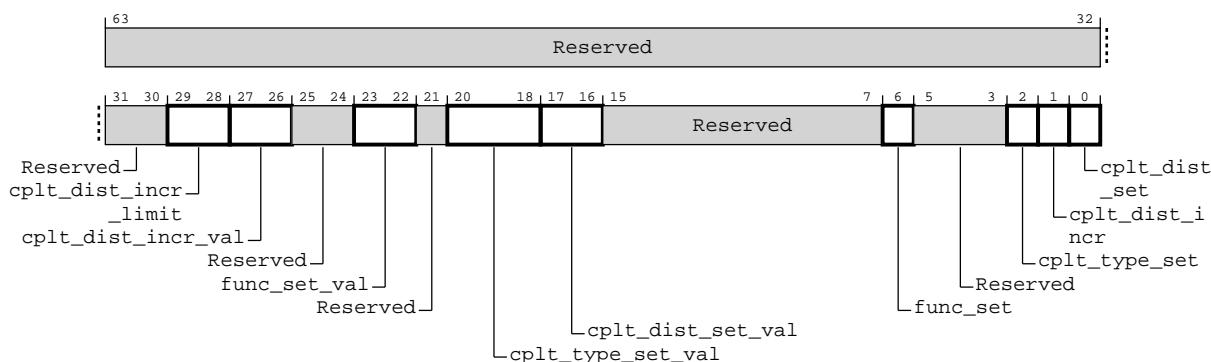
##### Usage constraints

This register is owned in the Root space and is 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 8-76: por\_ccg\_ra\_datasource\_ctl\_link0**



**Table 8-76: por\_ccg\_ra\_datasource\_ctl\_link0 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b01

Bits	Name	Description	Type	Reset
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b010
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.3.34 por\_ccg\_ra\_datasource\_ctl\_link1

Control to determine the chi-g datasource on SnpRespData (Link-1)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA28

##### Type

RW

##### Reset value

See individual bit resets

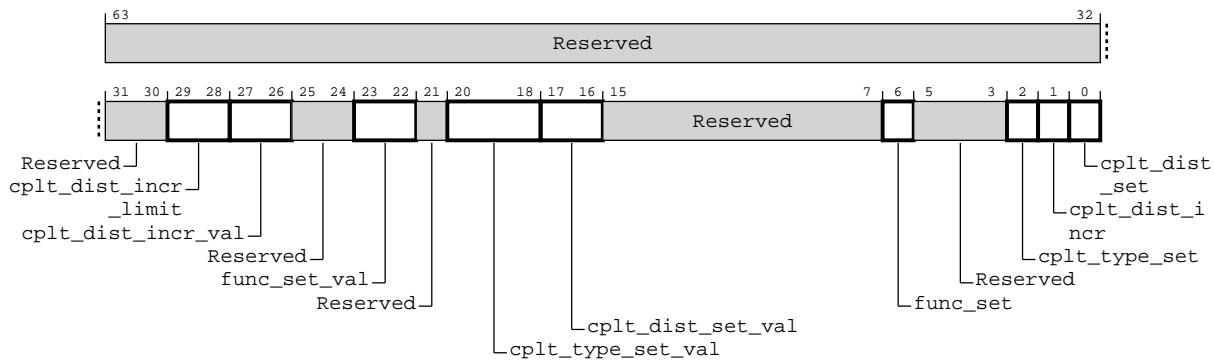
##### Usage constraints

This register is owned in the Root space and is 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 8-77: por\_ccg\_ra\_datasource\_ctl\_link1**



**Table 8-77: por\_ccg\_ra\_datasource\_ctl\_link1 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCRLIMIT	RW	0b01
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b010
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.3.35 por\_ccg\_ra\_datasource\_ctl\_link2

Control to determine the chi-g datasource on SnpRespData (Link-2)

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xA30

## Type

RW

## Reset value

See individual bit resets

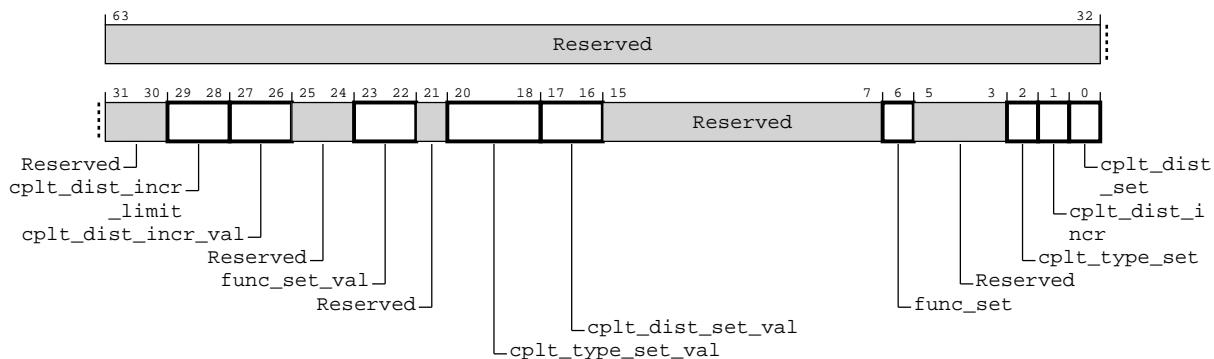
## Usage constraints

This register is owned in the Root space and is 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 8-78: por\_ccg\_ra\_datasource\_ctl\_link2**



**Table 8-78: por\_ccg\_ra\_datasource\_ctl\_link2 attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCLIMIT	RW	0b11
[27:26]	cplt_dist_incr_val	If CPLT_DIST_INCR is set, increment the incoming Completer distance by programmed CPLT_DIST_INCRVALUE up to CPLT_DIST_INCLIMIT	RW	0b01
[25:24]	Reserved	Reserved	RO	-
[23:22]	func_set_val	If FUNCTIONAL_SET is set, then DataSource [7:6] = FUNCTIONAL_SET_VALUE	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	If CPLT_TYPE_SET is set, then DataSource [4:2] = CPLT_TYPE_SETVALUE	RW	0b010
[17:16]	cplt_dist_set_val	If CPLT_DIST_SET is set, then DataSource [1:0] = CPLT_DIST_SETVALUE	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	func_set	FUNCTIONAL_SET is applicable at units where information is not available, either within or from downstream to drive on DataSource[7:6]	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	CPLT_TYPE_SET is applicable at the source of data.	RW	0b0

Bits	Name	Description	Type	Reset
[1]	cplt_dist_incr	CPLT_DIST_INCR is applicable at the below mentioned non-data sources.	RW	0b0
[0]	cplt_dist_set	CPLT_DIST_SET is applicable at the source of data.	RW	0b0

### 8.3.3.36 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3000 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

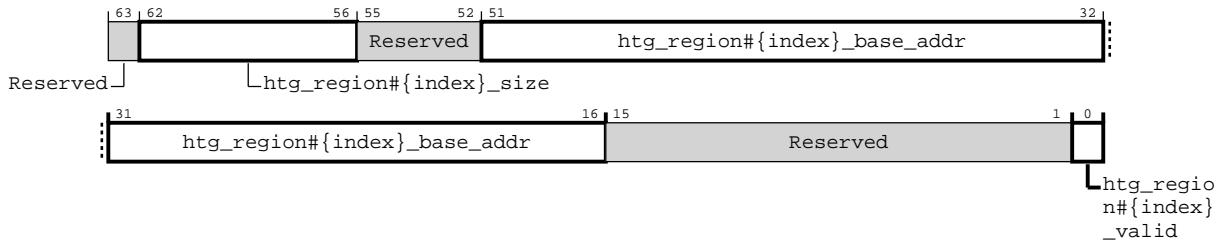
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-79: ra\_rnsam\_hashed\_tgt\_grp\_cfg1\_region0-31**



**Table 8-79: ra\_rnsam\_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 <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b0000000
[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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15:1]	Reserved	Reserved	RO	-
[0]	htg_region#{index}_valid	Memory region #{index} valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.3.37 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3100 + # {8 \* index}

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

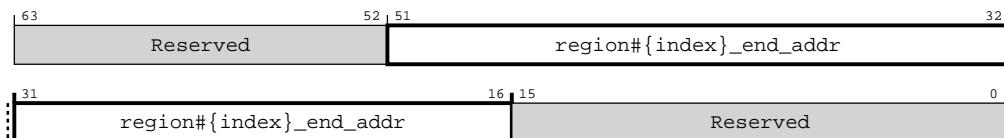
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-80: ra\_rnsam\_hashed\_tgt\_grp\_cfg2\_region0-31**



**Table 8-80: ra\_rnsam\_hashed\_tgt\_grp\_cfg2\_region0-31 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	0x0
[15:0]	Reserved	Reserved	RO	

### 8.3.3.38 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

## Address offset

`index(0-31) : 0x3200 + #{8 * index}`

## Type

RW

## Reset value

See individual bit resets

## Root group override

`por_ccg_ra_rcr.cfg_ctl`

## Secure group override

`por_ccg_ra_scr.cfg_ctl`

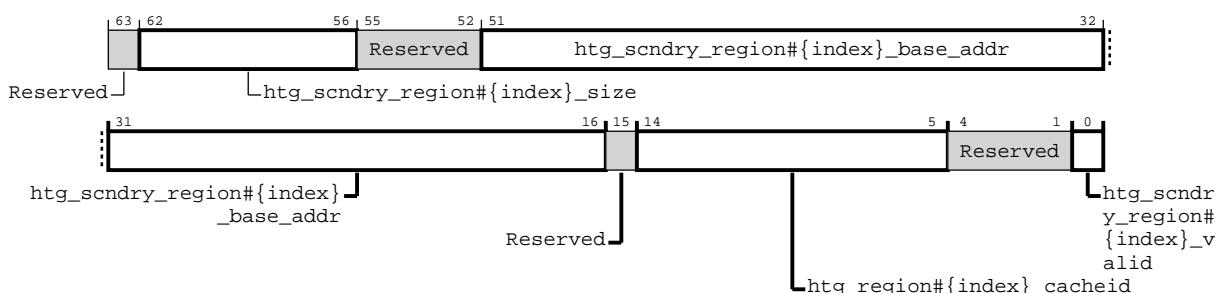
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.cfg_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.cfg_ctl` bit and `por_ccg_ra_rcr.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 8-81: ra\_rnsam\_hashed\_target\_grp\_secondary\_cfg1\_reg0-31**



**Table 8-81: ra\_rnsam\_hashed\_target\_grp\_secondary\_cfg1\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	<code>htg_scndry_region#{index}_size</code>	Secondary memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	<code>htg_scndry_region#{index}_base_addr</code>	Bits [51:16] of base address of the range, LSB bit is defined by the parameter <code>POR_RNSAM_HTG_RCOMP_LSB_PARAM</code>	RW	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:5]	htg_region#{index}_cacheid	Specifies the coherency domain associated to the HTG#{index}	RW	0x0
[4:1]	Reserved	Reserved	RO	-
[0]	htg_scndry_region#{index}_valid	Secondary memory region #{index} valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.3.39 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3300 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

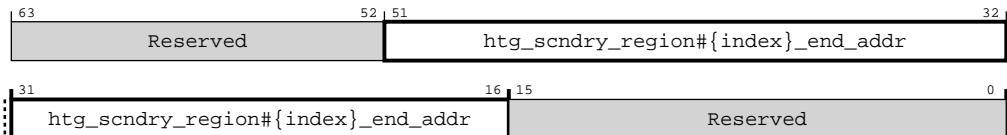
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-82: ra\_rnsam\_hashed\_target\_grp\_secondary\_cfg2\_reg0-31**



**Table 8-82: ra\_rnsam\_hashed\_target\_grp\_secondary\_cfg2\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAMHTG_RCOMP_LSB_PARAM	RW	0b00000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

### 8.3.3.40 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3400 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

##### Usage constraints

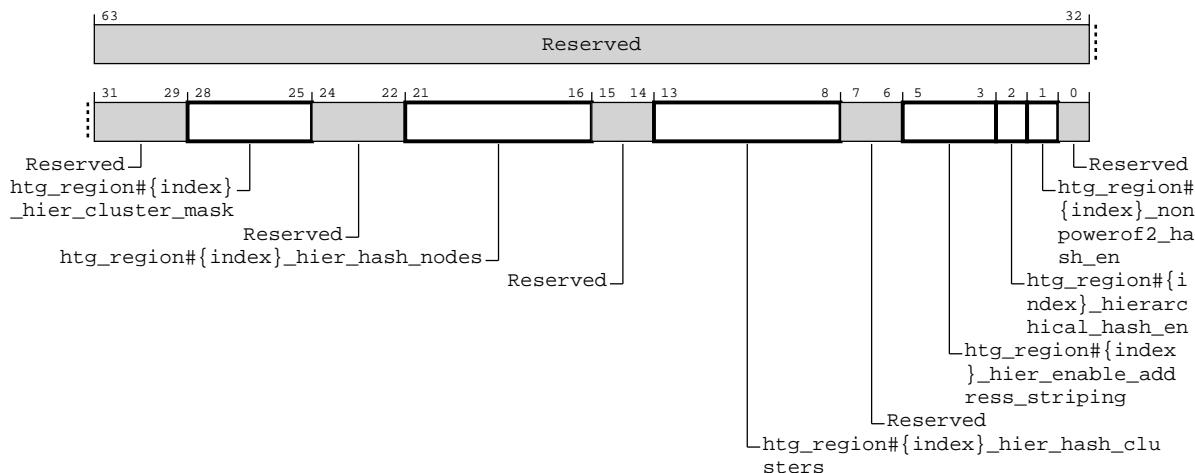
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.cfg\_ctl bit

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 8-83: ra\_rnsam\_hashed\_target\_grp\_hash\_cntl\_reg0-31**



**Table 8-83: ra\_rnsam\_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	<b>Hierarchical hashing</b> Enable cluster masking to achieve different interleave granularity across clusters. <b>0b0000</b> 64 byte interleave granularity across clusters <b>0b0110</b> 4096 byte interleave granularity across clusters <b>0b1111</b> Cluster interleaving disabled <b>others</b> Reserved	RW	0b1111
[24:22]	Reserved	Reserved	RO	-
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	0x0
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	0x0
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:3]	htg_region#{index}_hier_enable_address_striping	<p><b>Hierarchical hashing</b></p> <p>configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask).</p> <p><b>0b000</b> no address shuttering</p> <p><b>0b001</b> one addr bit shuttered (2 clusters)</p> <p><b>0b010</b> two addr bit shuttered (4 clusters)</p> <p><b>0b011</b> three addr bit shuttered (8 clusters)</p> <p><b>0b100</b> four addr bit shuttered (16 clusters)</p> <p><b>0b101</b> five addr bit shuttered (32 clusters)</p> <p><b>0b110</b> six addr bit shuttered (These are configured when the cachelines are &gt; 64B interleaved and clusters are enabled)</p> <p><b>0b111</b> seven addr bit shuttered</p> <p><b>others</b> Reserved</p>	RW	0b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	0b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	0b0
[0]	Reserved	Reserved	RO	-

### 8.3.3.41 ra\_rnsam\_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 in hashed target groups #{index8} to #{index8 + 7}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-3) : 0x3700 + #{8 \* index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

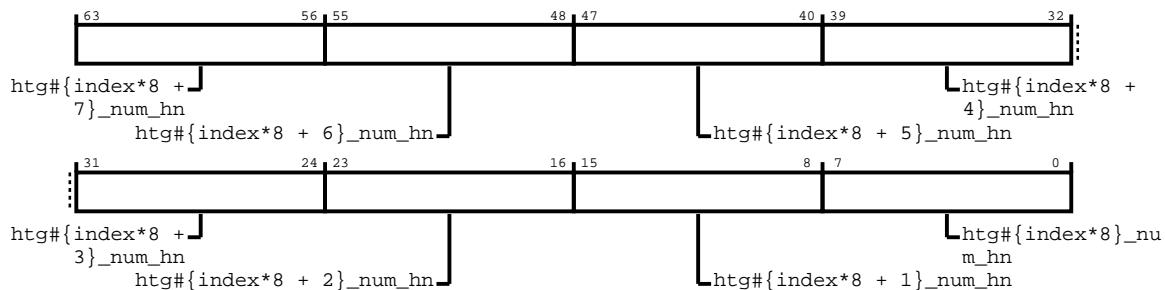
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-84: ra\_rnsam\_hashed\_target\_group\_hn\_count\_reg0-3**



**Table 8-84: ra\_rnsam\_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	0x00
[55:48]	htg#[index*8 + 6]_num_hn	HN count for hashed target group 6	RW	0x00
[47:40]	htg#[index*8 + 5]_num_hn	HN count for hashed target group 5	RW	0x00
[39:32]	htg#[index*8 + 4]_num_hn	HN count for hashed target group 4	RW	0x00
[31:24]	htg#[index*8 + 3]_num_hn	HN count for hashed target group 3	RW	0x00
[23:16]	htg#[index*8 + 2]_num_hn	HN count for hashed target group 2	RW	0x00
[15:8]	htg#[index*8 + 1]_num_hn	HN count for hashed target group 1	RW	0x00
[7:0]	htg#[index*8]_num_hn	HN count for hashed target group 0	RW	0x00

### 8.3.3.42 ra\_rnsam\_hashed\_target\_grp\_hnf\_nodeid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures HNF node IDs for hashed target groups. Controls target HNF node IDs  $\#\{index*4\}$  to  $\#\{index*4 + 3\}$ .

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$index(0-31) : 0x3500 + \#8 * index$

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

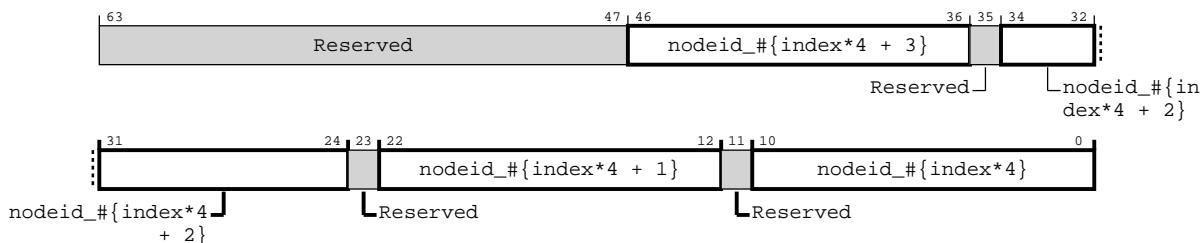
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-85: ra\_rnsam\_hashed\_target\_grp\_hnf\_nodeid\_reg0-31**



**Table 8-85: ra\_rnsam\_hashed\_target\_grp\_hnf\_nodeid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{index*4 + 3}	HNF target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{index*4 + 2}	HNF target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{index*4 + 1}	HNF target node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	HNF target node ID #{index*4}	RW	0b000000000000

### 8.3.3.43 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7) : 0x3780 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

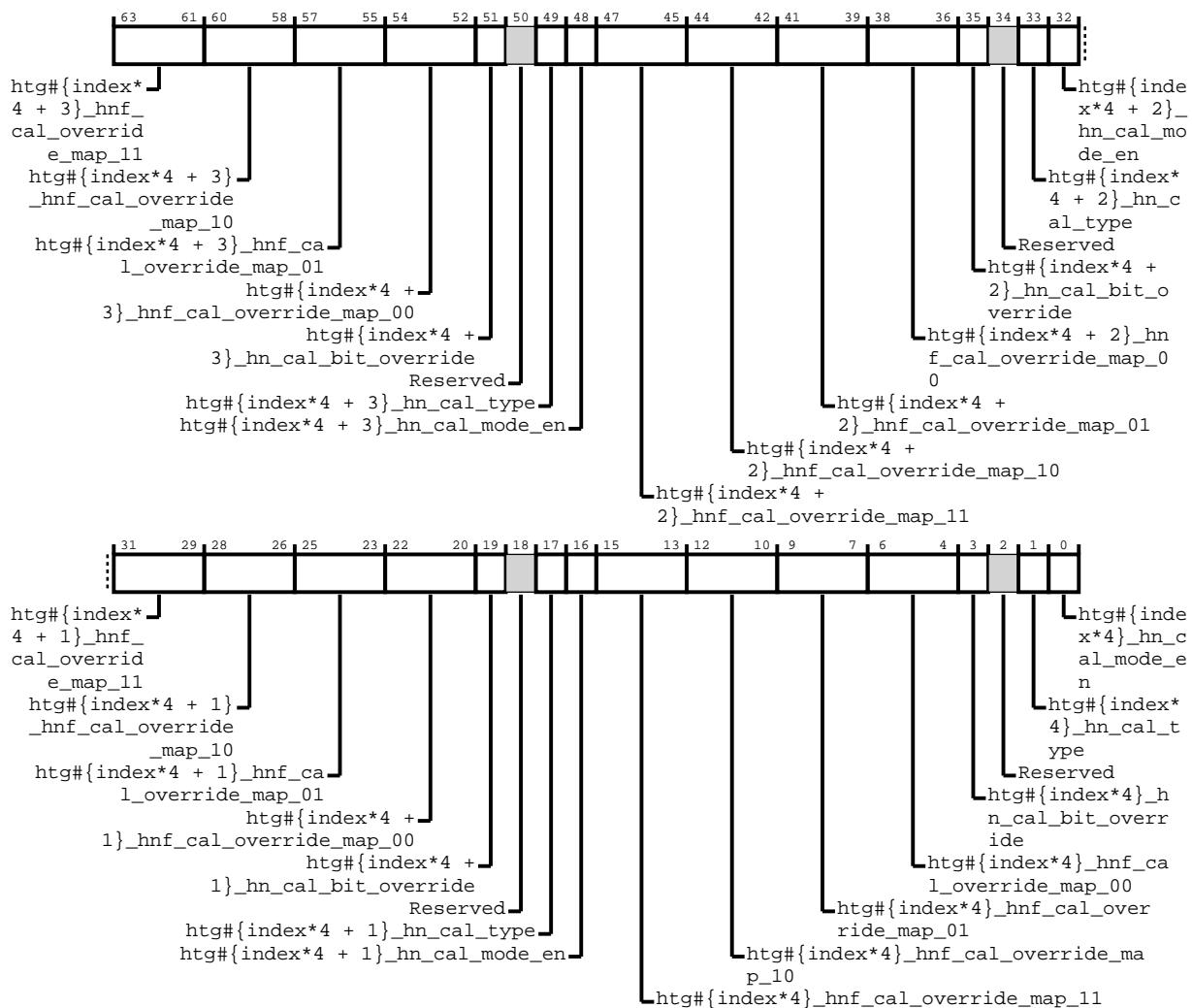
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-86: ra\_rnsam\_hashed\_target\_grp\_cal\_mode\_reg0-7**



**Table 8-86: ra\_rnsam\_hashed\_target\_grp\_cal\_mode\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:61]	htg#[index*4 + 3]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[60:58]	htg#[index*4 + 3]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[57:55]	htg#[index*4 + 3]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[54:52]	htg#[index*4 + 3]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000

Bits	Name	Description	Type	Reset
[51]	htg#[index*4 + 3]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 3}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[50]	Reserved	Reserved	RO	-
[49]	htg#[index*4 + 3]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 3}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[48]	htg#[index*4 + 3]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 3}	RW	0b0
[47:45]	htg#[index*4 + 2]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[44:42]	htg#[index*4 + 2]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[41:39]	htg#[index*4 + 2]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[38:36]	htg#[index*4 + 2]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[35]	htg#[index*4 + 2]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 2}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[34]	Reserved	Reserved	RO	-
[33]	htg#[index*4 + 2]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 2}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[32]	htg#[index*4 + 2]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 2}	RW	0b0
[31:29]	htg#[index*4 + 1]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[28:26]	htg#[index*4 + 1]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[25:23]	htg#[index*4 + 1]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[22:20]	htg#[index*4 + 1]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000

Bits	Name	Description	Type	Reset
[19]	htg#[index*4 + 1]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 1}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[18]	Reserved	Reserved	RO	-
[17]	htg#[index*4 + 1]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 1}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[16]	htg#[index*4 + 1]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 1}	RW	0b0
[15:13]	htg#[index*4]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[12:10]	htg#[index*4]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[9:7]	htg#[index*4]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[6:4]	htg#[index*4]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[3]	htg#[index*4]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[2]	Reserved	Reserved	RO	-
[1]	htg#[index*4]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[0]	htg#[index*4]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4}	RW	0b0

### 8.3.3.44 ra\_rnsam\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

$\text{index}(0\text{-}31) : 0x3840 + \#\{8 * \text{index}\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_ccg_ra_rcr.cfg_ctl`

### Secure group override

`por_ccg_ra_scr.cfg_ctl`

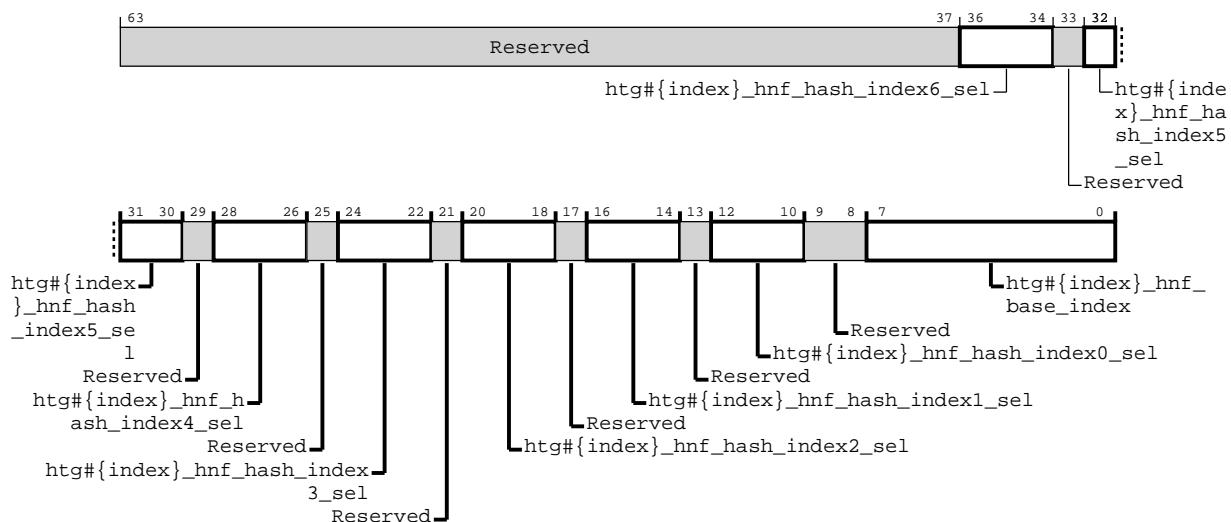
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.cfg_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.cfg_ctl` bit and `por_ccg_ra_rcr.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 8-87: ra\_rnsam\_hashed\_target\_grp\_compact\_hash\_ctrl0-31**



**Table 8-87: ra\_rnsam\_hashed\_target\_grp\_compact\_hash\_ctrl0-31 attributes**

Bits	Name	Description	Type	Reset
[63:37]	Reserved	Reserved	RO	-
[36:34]	htg#[index].hnf_hash_index6_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. <b>0b000</b> pass through from the SMP hnf_hash_index6. <b>0b001</b> SMP hash index6 + 1. <b>0b010</b> SMP hash index6 + 2. <b>0b011</b> SMP hash index6 + 3. <b>0b100</b> SMP hash index6 + 4. <b>0b101</b> SMP hash index6 + 5. <b>0b110</b> SMP hash index6 + 6. <b>0b111</b> Hardcoded value 0b0	RW	0b0
[33]	Reserved	Reserved	RO	-
[32:30]	htg#[index].hnf_hash_index5_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. <b>0b000</b> pass through from the SMP hnf_hash_index5. <b>0b001</b> SMP hash index5 + 1. <b>0b010</b> SMP hash index5 + 2. <b>0b011</b> SMP hash index5 + 3. <b>0b100</b> SMP hash index5 + 4. <b>0b101</b> SMP hash index5 + 5. <b>0b110</b> SMP hash index5 + 6. <b>0b111</b> Hardcoded value 0b0	RW	0b0
[29]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[28:26]	htg#[index]_hnf_hash_index4_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index4.</p> <p><b>0b001</b> SMP hash index4 + 1.</p> <p><b>0b010</b> SMP hash index4 + 2.</p> <p><b>0b011</b> SMP hash index4 + 3.</p> <p><b>0b100</b> SMP hash index4 + 4.</p> <p><b>0b101</b> SMP hash index4 + 5.</p> <p><b>0b110</b> SMP hash index4 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[25]	Reserved	Reserved	RO	-
[24:22]	htg#[index]_hnf_hash_index3_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index3.</p> <p><b>0b001</b> SMP hash index3 + 1.</p> <p><b>0b010</b> SMP hash index3 + 2.</p> <p><b>0b011</b> SMP hash index3 + 3.</p> <p><b>0b100</b> SMP hash index3 + 4.</p> <p><b>0b101</b> SMP hash index3 + 5.</p> <p><b>0b110</b> SMP hash index3 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[21]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[20:18]	htg#[index]_hnf_hash_index2_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index2.</p> <p><b>0b001</b> SMP hash index2 + 1.</p> <p><b>0b010</b> SMP hash index2 + 2.</p> <p><b>0b011</b> SMP hash index2 + 3.</p> <p><b>0b100</b> SMP hash index2 + 4.</p> <p><b>0b101</b> SMP hash index2 + 5.</p> <p><b>0b110</b> SMP hash index2 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[17]	Reserved	Reserved	RO	-
[16:14]	htg#[index]_hnf_hash_index1_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index1.</p> <p><b>0b001</b> SMP hash index1 + 1.</p> <p><b>0b010</b> SMP hash index1 + 2.</p> <p><b>0b011</b> SMP hash index1 + 3.</p> <p><b>0b100</b> SMP hash index1 + 4.</p> <p><b>0b101</b> SMP hash index1 + 5.</p> <p><b>0b110</b> SMP hash index1 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[13]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[12:10]	htg#[index]_hnf_hash_index0_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index0.</p> <p><b>0b001</b> SMP hash index0 + 1.</p> <p><b>0b010</b> SMP hash index0 + 2.</p> <p><b>0b011</b> SMP hash index0 + 3.</p> <p><b>0b100</b> SMP hash index0 + 4.</p> <p><b>0b101</b> SMP hash index0 + 5.</p> <p><b>0b110</b> SMP hash index0 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[9:8]	Reserved	Reserved	RO	-
[7:0]	htg#[index]_hnf_base_index	base index to the HNF target ID table	RW	Configuration dependent

### 8.3.3.45 ra\_rnsam\_hashed\_target\_grp\_misc\_nodeid\_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures Misc node IDs for hashed target groups. This registers is used either for HNF/HNP/CCGs. Controls misc node IDs #{{index}4} to #{{index}4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7) : 0x3A00 + #8 \* index

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

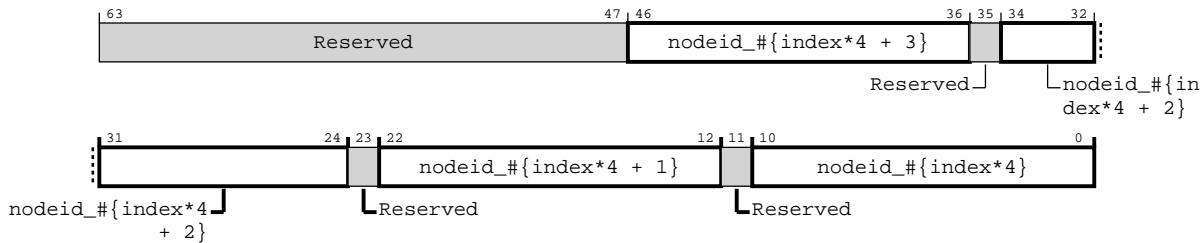
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-88: ra\_rnsam\_hashed\_target\_grp\_misc\_nodeid\_reg0-7**



**Table 8-88: ra\_rnsam\_hashed\_target\_grp\_misc\_nodeid\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{index*4 + 3}	Misc target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{index*4 + 2}	Misc target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{index*4 + 1}	Misc target node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	Misc target node ID #{index*4}	RW	0b000000000000

### 8.3.3.46 ra\_rnsam\_hashed\_tgt\_override\_cam\_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures hashed target override CAM structure register. Each entry contains a logical-id and physical-id of the override HN

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

$\text{index}(0\text{-}7) : 0x3B00 + \#\{8 * \text{index}\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_ccg_ra_rcc.cfg_ctl`

### Secure group override

`por_ccg_ra_scr.cfg_ctl`

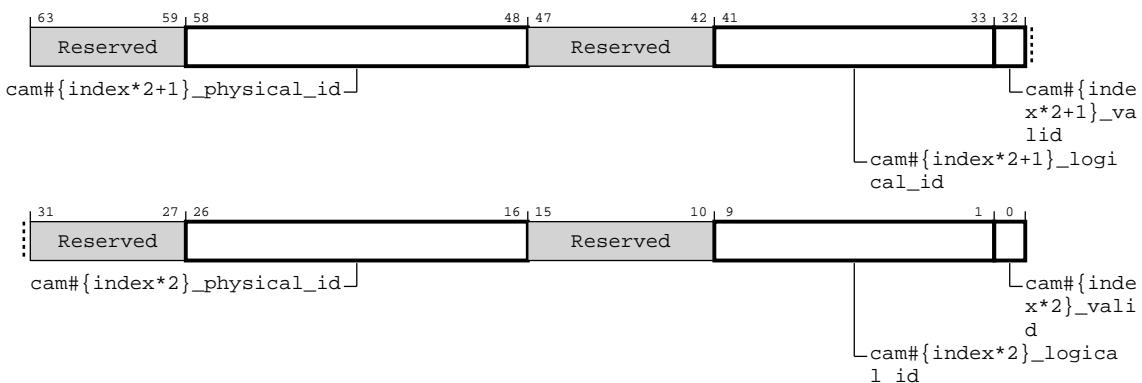
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.cfg_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.cfg_ctl` bit and `por_ccg_ra_rcc.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 8-89: ra\_rnsam\_hashed\_tgt\_override\_cam\_reg0-7**



**Table 8-89: ra\_rnsam\_hashed\_tgt\_override\_cam\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	cam#{index*2+1}_physical_id	CAM structure HN target-id #{index*4+1}	RW	0b000000000000
[47:42]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[41:33]	cam#{index*2+1}_logical_id	CAM structure HN logical-id #{index*4+1}	RW	0b0000000000
[32]	cam#{index*2+1}_valid	CAM structure valid #{index*4+1}	RW	0b0
[31:27]	Reserved	Reserved	RO	-
[26:16]	cam#{index*2}_physical_id	CAM structure HN target-id #{index*4}	RW	0b000000000000
[15:10]	Reserved	Reserved	RO	-
[9:1]	cam#{index*2}_logical_id	CAM structure HN logical-id #{index*4}	RW	0b0000000000
[0]	cam#{index*2}_valid	CAM structure valid #{index*4}	RW	0b0

### 8.3.3.47 ra\_rnsam\_hash\_addr\_mask\_reg

Configures the address mask that is applied before hashing the address bits.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x37C0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

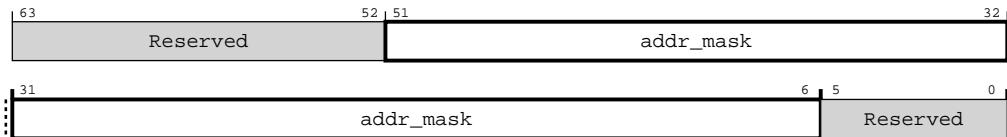
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-90: ra\_rnsam\_hash\_addr\_mask\_reg**



**Table 8-90: ra\_rnsam\_hash\_addr\_mask\_reg attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask applied before hashing	RW	0x3FFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

### 8.3.3.48 ra\_rnsam\_region\_cmp\_addr\_mask\_reg

Configures the address mask that is applied before region compare.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x37C8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

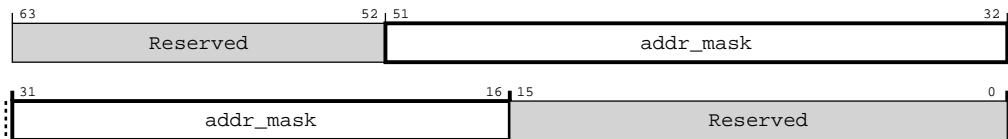
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.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 8-91: ra\_rnsam\_region\_cmp\_addr\_mask\_reg**



**Table 8-91: ra\_rnsam\_region\_cmp\_addr\_mask\_reg attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region compare	RW	0xFFFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

### 8.3.3.49 ra\_rnsam\_status

Functions as the default and programming mode status register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x37D0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccg\_ra\_rcr.cfg\_ctl

##### Secure group override

por\_ccg\_ra\_scr.cfg\_ctl

#### Usage constraints

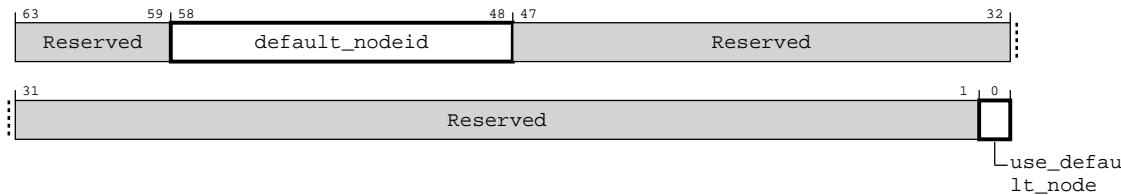
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccg\_ra\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccg\_ra\_scr.cfg\_ctl bit and por\_ccg\_ra\_rcr.cfg\_ctl bit

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 8-92: ra\_rnsam\_status**



**Table 8-92: ra\_rnsam\_status attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	default_nodeid	Default Node ID	RW	0x000
[47:1]	Reserved	Reserved	RO	-
[0]	use_default_node	Indicates target ID selection mode  <b>0b0</b> Enables RN SAM to hash address bits and generate target ID  <b>0b1</b> Uses default target ID	RW	0b1

### 8.3.3.50 ra\_rnsam\_sam\_generic\_regs0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configuration register for the custom logic

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-7) : 0x3800 + #{8 \* index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_ccg_ra_rcr.cfg_ctl`

### Secure group override

`por_ccg_ra_scr.cfg_ctl`

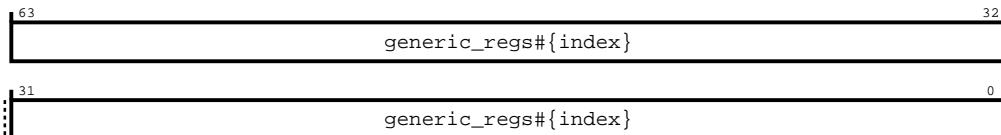
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccg_ra_scr.cfg_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_ccg_ra_scr.cfg_ctl` bit and `por_ccg_ra_rcr.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 8-93: ra\_rnsam\_sam\_generic\_regs0-7**



**Table 8-93: ra\_rnsam\_sam\_generic\_regs0-7 attributes**

Bits	Name	Description	Type	Reset
[63:0]	generic_regs#{index}	Configuration register for the custom logic	RW	0x0

### 8.3.4 CCLA register summary

The following table describes the registers for the relevant component.

**Table 8-94: por\_ccla\_cfg register summary**

Offset	Name	Type	Description
0x0	<code>por_ccla_node_info</code>	RO	<code>por_ccla_node_info</code>
0x80	<code>por_ccla_child_info</code>	RO	<code>por_ccla_child_info</code>
0x980	<code>por_ccla_scr</code>	RW	<code>por_ccla_scr</code>
0x988	<code>por_ccla_rcr</code>	RW	<code>por_ccla_rcr</code>
0x910	<code>por_ccla_unit_info</code>	RO	<code>por_ccla_unit_info</code>
0xB00	<code>por_ccla_cfg_ctl</code>	RW	<code>por_ccla_cfg_ctl</code>
0xB08	<code>por_ccla_aux_ctl</code>	RW	<code>por_ccla_aux_ctl</code>
0xB10	<code>por_ccla_c2c_packing_ctl</code>	RW	<code>por_ccla_c2c_packing_ctl</code>

Offset	Name	Type	Description
0xC00	por_ccla_ccix_prop_capabilities	RO	por_ccla_ccix_prop_capabilities
0xC08	por_ccla_cxs_attr_capabilities	RO	por_ccla_cxs_attr_capabilities
0xD28	por_ccla_err_agent_id	RW	por_ccla_err_agent_id
0xD30	por_ccla_agentid_to_portid_reg0	RW	por_ccla_agentid_to_portid_reg0
0xD70	por_ccla_agentid_to_portid_val	RW	por_ccla_agentid_to_portid_val
0xD78	por_ccla_portfwd_en	RW	por_ccla_portfwd_en
0x1500	por_ccla_ide_freq_factor	RW	por_ccla_ide_freq_factor
0x1458	por_ccla_ull_idle_counter	RW	por_ccla_ull_idle_counter
0xD80	por_ccla_portfwd_status	RO	por_ccla_portfwd_status
0xD88	por_ccla_portfwd_req	RW	por_ccla_portfwd_req
0xD90	por_ccla_linkid_to_hops	RW	por_ccla_linkid_to_hops
0xE00	por_ccla_cxl_link_rx_credit_ctl	RW	por_ccla_cxl_link_rx_credit_ctl
0xE08	por_ccla_cxl_link_rx_credit_return_stat	RO	por_ccla_cxl_link_rx_credit_return_stat
0xE10	por_ccla_cxl_link_tx_credit_stat	RO	por_ccla_cxl_link_tx_credit_stat
0xE50	por_ccla_cxl_security_policy	RW	por_ccla_cxl_security_policy
0xE78	por_ccla_cxl_hdm_decoder_capability	RO	por_ccla_cxl_hdm_decoder_capability
0xE80	por_ccla_cxl_hdm_decoder_global_control	RW	por_ccla_cxl_hdm_decoder_global_control
0x1040 : 0x1078	por_ccla_cxl_hdm_decoder_0-7_base_low	RWL	por_ccla_cxl_hdm_decoder_0-7_base_low
0x1080 : 0x10B8	por_ccla_cxl_hdm_decoder_0-7_base_high	RWL	por_ccla_cxl_hdm_decoder_0-7_base_high
0x10C0 : 0x10F8	por_ccla_cxl_hdm_decoder_0-7_size_low	RWL	por_ccla_cxl_hdm_decoder_0-7_size_low
0x1100 : 0x1138	por_ccla_cxl_hdm_decoder_0-7_size_high	RWL	por_ccla_cxl_hdm_decoder_0-7_size_high
0x1140 : 0x1178	por_ccla_cxl_hdm_decoder_0-7_control	RWL	por_ccla_cxl_hdm_decoder_0-7_control
0x1180 : 0x11B8	por_ccla_cxl_hdm_decoder_0-7_dpa_skip_low	RWL	por_ccla_cxl_hdm_decoder_0-7_dpa_skip_low
0x11C0 : 0x11F8	por_ccla_cxl_hdm_decoder_0-7_dpa_skip_high	RWL	por_ccla_cxl_hdm_decoder_0-7_dpa_skip_high
0x1200 : 0x1238	por_ccla_cxl_dcd_region_base_address_low0-7	RW	por_ccla_cxl_dcd_region_base_address_low0-7
0x1240 : 0x1278	por_ccla_cxl_dcd_region_base_address_high0-7	RW	por_ccla_cxl_dcd_region_base_address_high0-7

Offset	Name	Type	Description
0x1280 : 0x12B8	por_ccla_cxl_dcd_region_max_address_low0-7	RW	por_ccla_cxl_dcd_region_max_address_low0-7
0x12C0 : 0x12F8	por_ccla_cxl_dcd_region_max_address_high0-7	RW	por_ccla_cxl_dcd_region_max_address_high0-7
0x1300 : 0x1338	por_ccla_cxl_dcd_region_lut_control0-7	RW	por_ccla_cxl_dcd_region_lut_control0-7
0x1340	por_ccla_cxl_dcd_write_intf	WO	por_ccla_cxl_dcd_write_intf
0x1348	por_ccla_cxl_dcd_read_intf	RW	por_ccla_cxl_dcd_read_intf
0xED0	por_ccla_snoop_filter_group_id	RW	por_ccla_snoop_filter_group_id
0xED8	por_ccla_snoop_filter_effective_size	RW	por_ccla_snoop_filter_effective_size
0xEE0	por_ccla_dvsec_cxl_range_1_base_high	RWL	por_ccla_dvsec_cxl_range_1_base_high
0xEE8	por_ccla_dvsec_cxl_range_1_base_low	RWL	por_ccla_dvsec_cxl_range_1_base_low
0xEF0	por_ccla_dvsec_cxl_range_2_base_high	RWL	por_ccla_dvsec_cxl_range_2_base_high
0xEF8	por_ccla_dvsec_cxl_range_2_base_low	RWL	por_ccla_dvsec_cxl_range_2_base_low
0x1000	por_ccla_dvsec_cxl_range_1_size_high	RW	por_ccla_dvsec_cxl_range_1_size_high
0x1008	por_ccla_dvsec_cxl_range_1_size_low	RW	por_ccla_dvsec_cxl_range_1_size_low
0x1010	por_ccla_dvsec_cxl_range_2_size_high	RW	por_ccla_dvsec_cxl_range_2_size_high
0x1018	por_ccla_dvsec_cxl_range_2_size_low	RW	por_ccla_dvsec_cxl_range_2_size_low
0xF00	por_ccla_dvsec_cxl_control	RWL	por_ccla_dvsec_cxl_control
0xF08	por_ccla_dvsec_cxl_control2	RW	por_ccla_dvsec_cxl_control2
0xF10	por_ccla_dvsec_cxl_lock	RW	por_ccla_dvsec_cxl_lock
0x1F90	por_ccla_ras_err_en	RW	por_ccla_ras_err_en
0x1F98	por_ccla_ras_viral_en	RW	por_ccla_ras_viral_en
0xF18	por_ccla_dvsec_flex_bus_port_control	RW	por_ccla_dvsec_flex_bus_port_control
0xF40	por_ccla_err_capabilities_control	RW	por_ccla_err_capabilities_control
0xF58	por_ccla_IDE_key_refresh_time_control	RW	por_ccla_IDE_key_refresh_time_control
0xF60	por_ccla_IDE_truncation_transmit_delay_control	RW	por_ccla_IDE_truncation_transmit_delay_control
0xF90	por_ccla_IDE_outbound_state	RO	por_ccla_IDE_outbound_state
0xF98	por_ccla_IDE_inbound_state	RO	por_ccla_IDE_inbound_state
0xF70	por_ccla_ll_to_ull_msg	RW	por_ccla_ll_to_ull_msg
0xF80	por_ccla_cxl_timeout_iso_ctl	RW	por_ccla_cxl_timeout_iso_ctl
0xF88	por_ccla_cxl_timeout_iso_capability	RW	por_ccla_cxl_timeout_iso_capability
0xF78	por_ccla_cxl_timeout_iso_status	RW	por_ccla_cxl_timeout_iso_status
0x1020	por_ccla_timeout_base_value_cycles	RW	por_ccla_timeout_base_value_cycles
0xE58	por_ccla_CXL_Cache_ID_Decoder_Capability	RW	por_ccla_CXL_Cache_ID_Decoder_Capability
0xE60	por_ccla_CXL_Cache_ID_Decoder_Control	RW	por_ccla_CXL_Cache_ID_Decoder_Control
0xE68	por_ccla_CXL_Cache_ID_Decoder_Status	RW	por_ccla_CXL_Cache_ID_Decoder_Status
0xF28	por_ccla_root_port_n_security_policy	RW	por_ccla_root_port_n_security_policy
0xF30	por_ccla_root_port_n_id	RW	por_ccla_root_port_n_id

Offset	Name	Type	Description
0xE18	por_ccla_cxl_link_layer_defeature	RW	por_ccla_cxl_link_layer_defeature
0xE20	por_ccla_ull_ctl	RW	por_ccla_ull_ctl
0xE28	por_ccla_ull_status	RO	por_ccla_ull_status
0xE30	por_ccla_cxl_ll_errinject_ctl	RW	por_ccla_cxl_ll_errinject_ctl
0xE38	por_ccla_cxl_ll_errinject_stat	RO	por_ccla_cxl_ll_errinject_stat
0xE40	por_ccla_cxl_viral_prop_en	RW	por_ccla_cxl_viral_prop_en
0xD908	por_ccla_pmu_event_sel	RW	por_ccla_pmu_event_sel
0xF700	por_ccla_c2c_port_register_offset_header	RO	por_ccla_c2c_port_register_offset_header
0xF708	por_ccla_c2c_port_register_offset_table_0	RO	por_ccla_c2c_port_register_offset_table_0
0xF710	por_ccla_c2c_port_register_offset_table_1	RO	por_ccla_c2c_port_register_offset_table_1
0xF718	por_ccla_c2c_port_capabilities_and_control_header	RO	por_ccla_c2c_port_capabilities_and_control_header
0xF720	por_ccla_c2c_port_capabilities_1	RO	por_ccla_c2c_port_capabilities_1
0xF728	por_ccla_c2c_port_capabilities_2	RO	por_ccla_c2c_port_capabilities_2
0xF730	por_ccla_c2c_port_control_and_status	RW	por_ccla_c2c_port_control_and_status
0xF738	por_ccla_c2c_port_ingressid_route_table_header	RO	por_ccla_c2c_port_ingressid_route_table_header
0xF828	por_ccla_c2c_port_ingressid_route_table_capabilities	RO	por_ccla_c2c_port_ingressid_route_table_capabilities
0xF740	por_ccla_c2c_port_ingressid_route_table_control_and_status	RW	por_ccla_c2c_port_ingressid_route_table_control_and_status
0xF748	por_ccla_c2c_port_ingressid_route_entry_0	RW	por_ccla_c2c_port_ingressid_route_entry_0
0xF750	por_ccla_c2c_port_egressid_route_table_header	RO	por_ccla_c2c_port_egressid_route_table_header
0xF820	por_ccla_c2c_port_egressid_route_table_capabilities	RO	por_ccla_c2c_port_egressid_route_table_capabilities
0xF758	por_ccla_c2c_port_egressid_route_table_control_and_status	RW	por_ccla_c2c_port_egressid_route_table_control_and_status
0xF760	por_ccla_c2c_port_egressid_route_entry_0	RW	por_ccla_c2c_port_egressid_route_entry_0
0xF768	por_ccla_c2c_logical_linkend_common_header	RO	por_ccla_c2c_logical_linkend_common_header
0xF770	por_ccla_c2c_logical_linkend_offset_register	RO	por_ccla_c2c_logical_linkend_offset_register
0xF778	por_ccla_c2c_logical_linkend_0-0_header	RO	por_ccla_c2c_logical_linkend_0-0_header
0xF780	por_ccla_c2c_logical_linkend_0-0_control_and_status	RW	por_ccla_c2c_logical_linkend_0-0_control_and_status
0xF788	por_ccla_c2c_logical_linkend_0-0_map_table	RW	por_ccla_c2c_logical_linkend_0-0_map_table
0xF790	por_ccla_c2c_linkproperties_supported_uniform1_0-0	RO	por_ccla_c2c_linkproperties_supported_uniform1_0-0
0xF798	por_ccla_c2c_linkproperties_advertised_uniform1_0-0	RW	por_ccla_c2c_linkproperties_advertised_uniform1_0-0
0xF7A0	por_ccla_c2c_linkproperties_informed_uniform1_0-0	RO	por_ccla_c2c_linkproperties_informed_uniform1_0-0

Offset	Name	Type	Description
0xF7A8 : 0xF7A8	por_ccla_c2c_linkproperties_negotiated_uniform1_0-0	RO	por_ccla_c2c_linkproperties_negotiated_uniform1_0-0
0xF7B0 : 0xF7B0	por_ccla_c2c_linkproperties_supported_rx1_0-0	RO	por_ccla_c2c_linkproperties_supported_rx1_0-0
0xF7B8 : 0xF7B8	por_ccla_c2c_linkproperties_advertised_rx1_0-0	RW	por_ccla_c2c_linkproperties_advertised_rx1_0-0
0xF7C0 : 0xF7C0	por_ccla_c2c_linkproperties_informed_rx1_0-0	RO	por_ccla_c2c_linkproperties_informed_rx1_0-0
0xF7C8 : 0xF7C8	por_ccla_c2c_linkproperties_negotiated_rx1_0-0	RO	por_ccla_c2c_linkproperties_negotiated_rx1_0-0
0xF7D0 : 0xF7D0	por_ccla_c2c_linkproperties_supported_tx1_0-0	RO	por_ccla_c2c_linkproperties_supported_tx1_0-0
0xF7D8 : 0xF7D8	por_ccla_c2c_linkproperties_advertised_tx1_0-0	RW	por_ccla_c2c_linkproperties_advertised_tx1_0-0
0xF7E0 : 0xF7E0	por_ccla_c2c_linkproperties_informed_tx1_0-0	RO	por_ccla_c2c_linkproperties_informed_tx1_0-0
0xF7E8 : 0xF7E8	por_ccla_c2c_linkproperties_negotiated_tx1_0-0	RO	por_ccla_c2c_linkproperties_negotiated_tx1_0-0
0xF7F8 : 0xF7F8	por_ccla_c2c_logical_linkend_0-0_mc_credit1	RW	por_ccla_c2c_logical_linkend_0-0_mc_credit1
0xF800 : 0xF800	por_ccla_c2c_logical_linkend_0-0_mc_credit2	RW	por_ccla_c2c_logical_linkend_0-0_mc_credit2
0xF808 : 0xF808	por_ccla_c2c_logical_linkend_0-0_mc_credit3	RW	por_ccla_c2c_logical_linkend_0-0_mc_credit3
0xF810 : 0xF810	por_ccla_c2c_logical_linkend_0-0_mc_credit4	RW	por_ccla_c2c_logical_linkend_0-0_mc_credit4
0xF818 : 0xF818	por_ccla_c2c_logical_linkend_0-0_mc_credit5	RW	por_ccla_c2c_logical_linkend_0-0_mc_credit5
0xE000	por_ccla_errfr	RO	por_ccla_errfr
0xE008	por_ccla_errctlr	RW	por_ccla_errctlr
0xE010	por_ccla_errstatus	W1C	por_ccla_errstatus
0xE018	por_ccla_erraddr	RW	por_ccla_erraddr
0xE028	por_ccla_errmisc1	RW	por_ccla_errmisc1
0xE800	por_ccla_errpfgf	RO	por_ccla_errpfgf
0xE808	por_ccla_errpfgctl	RW	por_ccla_errpfgctl

Offset	Name	Type	Description
0xE810	por_ccla_errpfgcdn	RW	por_ccla_errpfgcdn
0xE040	por_ccla_errfr_NS	RO	por_ccla_errfr_NS
0xE048	por_ccla_errctlr_NS	RW	por_ccla_errctlr_NS
0xE050	por_ccla_errstatus_NS	W1C	por_ccla_errstatus_NS
0xE058	por_ccla_erraddr_NS	RW	por_ccla_erraddr_NS
0xE068	por_ccla_errmisc1_NS	RW	por_ccla_errmisc1_NS
0xE840	por_ccla_errpfgf_NS	RO	por_ccla_errpfgf_NS
0xE848	por_ccla_errpfgctl_NS	RW	por_ccla_errpfgctl_NS
0xE850	por_ccla_errpfgcdn_NS	RW	por_ccla_errpfgcdn_NS
0xED00	por_ccla_errcapctl	RW	por_ccla_errcapctl
0xEE00	por_ccla_errgsr	RO	por_ccla_errgsr
0xEE10	por_ccla_errildr	RO	por_ccla_errildr
0xEFA8	por_ccla_errdevaff	RO	por_ccla_errdevaff
0xEF88	por_ccla_errdevarch	RO	por_ccla_errdevarch
0xEFC8	por_ccla_errdevid	RO	por_ccla_errdevid
0xEF00	por_ccla_errpidr45	RO	por_ccla_errpidr45
0xEF00	por_ccla_errpidr01	RO	por_ccla_errpidr01
0xEF08	por_ccla_errpidr23	RO	por_ccla_errpidr23
0xEFF0	por_ccla_errcidr01	RO	por_ccla_errcidr01
0xEFF8	por_ccla_errcidr23	RO	por_ccla_errcidr23
0x1BF8	por_ccg_multi_mesh_chn_ctrl	RW	por_ccg_multi_mesh_chn_ctrl
0x1C08 : 0x1C80	por_ccg_multi_mesh_chn_sel_0-15	RW	por_ccg_multi_mesh_chn_sel_0-15

### 8.3.4.1 por\_ccla\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

### Reset value

See individual bit resets

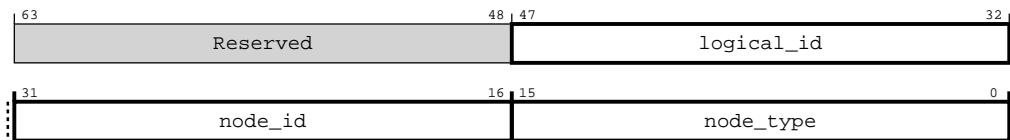
### Usage constraints

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 8-94: por\_ccla\_node\_info**



**Table 8-95: por\_ccla\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0105

### 8.3.4.2 por\_ccla\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

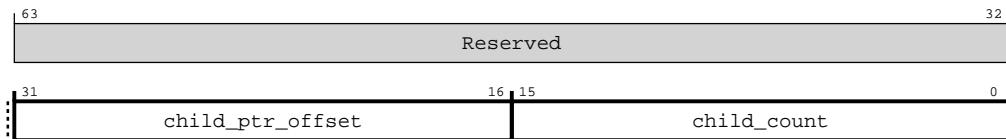
## Usage constraints

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 8-95: por\_ccla\_child\_info**



**Table 8-96: por\_ccla\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.4.3 por\_ccla\_scr

Secure register access override.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x980

### Type

RW

### Reset value

See individual bit resets

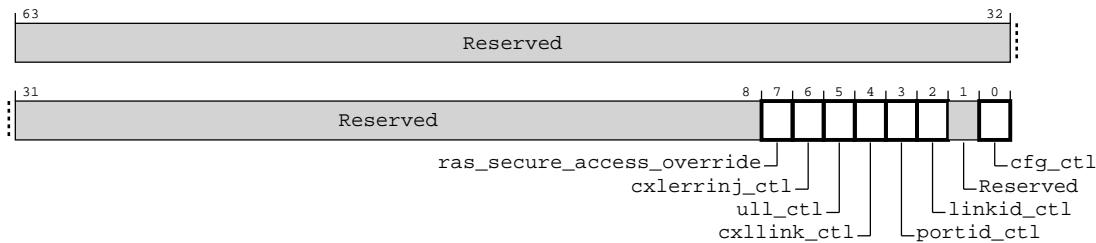
## Usage constraints

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 8-96: por\_ccla\_scr**



**Table 8-97: por\_ccla\_scr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras_secure_access_override	Allows Secure override of the RAS registers	RW	0b0
[6]	cxlerrinj_ctl	Allows Secure override of the CXL error injection registers	RW	0b0
[5]	ull_ctl	Allows Secure override of the upper link layer control registers	RW	0b0
[4]	cxllink_ctl	Allows Secure override of the CXL link layer registers	RW	0b0
[3]	portid_ctl	Allows Secure override of the LA Port ID registers	RW	0b0
[2]	linkid_ctl	Allows Secure override of the LA Link ID registers	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.4.4 por\_ccla\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

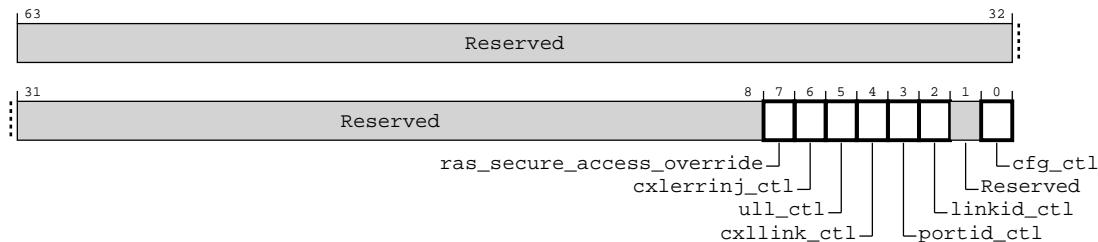
## Usage constraints

This register is owned in the Root space and is 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 8-97: por\_ccla\_rcr**



**Table 8-98: por\_ccla\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras_secure_access_override	Allows Root override of the RAS registers	RW	0b0
[6]	cxlerrinj_ctl	Allows Root override of the CXL error injection registers	RW	0b0
[5]	ull_ctl	Allows Root override of the upper link layer control registers	RW	0b0
[4]	cxllink_ctl	Allows Root override of the CXL link layer registers	RW	0b0
[3]	portid_ctl	Allows Root override of the LA Port ID registers	RW	0b0
[2]	linkid_ctl	Allows Root override of the LA Link ID registers	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.4.5 por\_ccla\_unit\_info

Provides component identification information for CCLA.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x910

## Type

RO

## Reset value

See individual bit resets

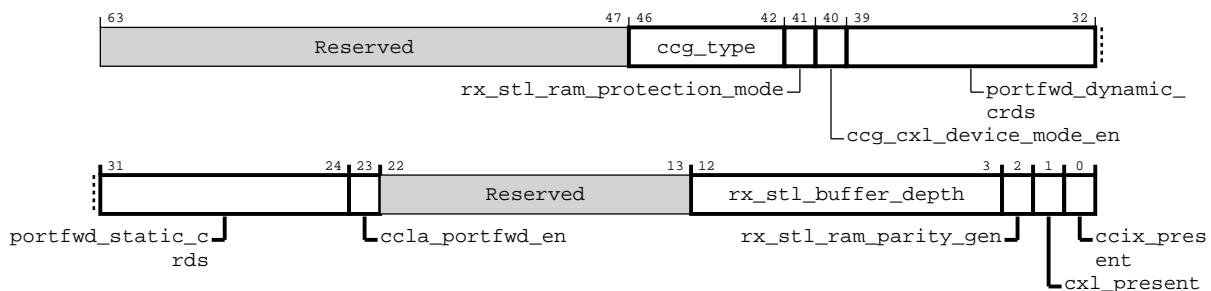
## Usage constraints

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 8-98: por\_ccla\_unit\_info**



**Table 8-99: por\_ccla\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:42]	ccg_type	CCG type	RO	0x0
[41]	rx_stl_ram_protection_mode	Protection mode on the RX_STL RAM	RO	0x0
	0	parity		
	1	ECC		
[40]	ccg_cxl_device_mode_en	ccg_cxl_device_mode_en	RO	Configuration dependent
[39:32]	portfwd_dynamic_crds	Number of dynamic credits granted by this CCLA port for port forwarded traffic	RO	Configuration dependent
[31:24]	portfwd_static_crds	Number of static credits granted by this CCLA port for port forwarded traffic	RO	Configuration dependent
[23]	ccla_portfwd_en	Port forwarding is enabled at this CCLA port	RO	0x0
[22:13]	Reserved	Reserved	RO	-
[12:3]	rx_stl_buffer_depth	Depth of CCL stalling channel RX buffer for CXS RSP with data messages	RO	0x0
[2]	rx_stl_ram_parity_gen	Option to generate parity bits for the RX STL buffer	RO	0x0
[1]	cxl_present	Option to generate CXL support over CXS	RO	0x0

Bits	Name	Description	Type	Reset
[0]	ccix_present	Option to generate CCIX support over CXS	RO	0x0

### 8.3.4.6 por\_ccla\_cfg\_ctl

Functions as the configuration control register for CCLA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xB00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cfg\_ctl

##### Secure group override

por\_ccla\_scr.cfg\_ctl

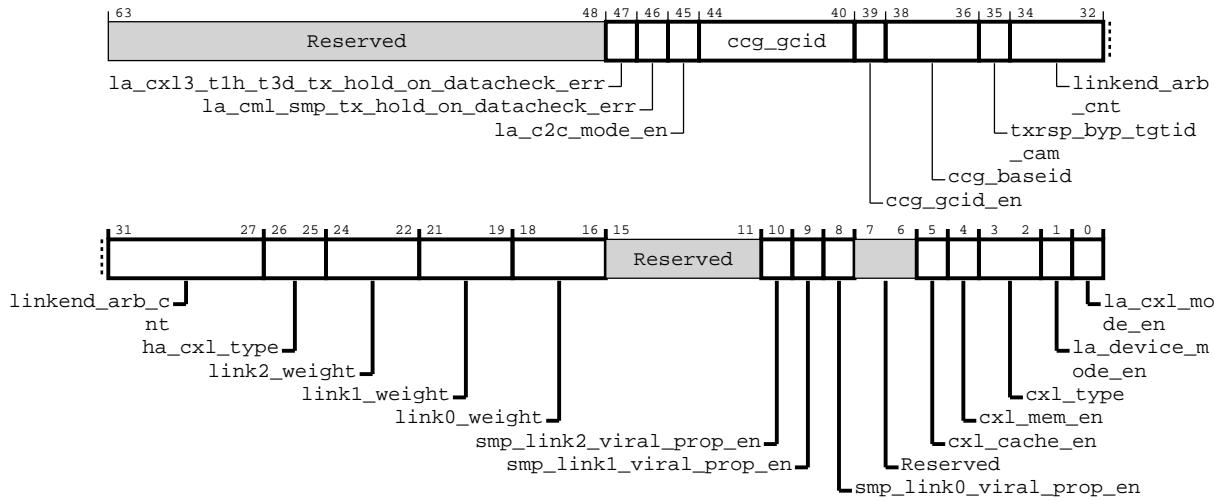
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cfg\_ctl bit and por\_ccla\_rcr.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 8-99: por\_ccla\_cfg\_ctl**



**Table 8-100: por\_ccla\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47]	la_cxl3_t1h_t3d_tx_hold_on_datacheck_err	When set holds either CXL3 Type1 host or Type 3 device TX from sending CXS flits when datacheck has a parity error. When set, will halt TX (not restartable) so that the flit with datacheck error does not get sent on CXS. When clear this allows flits with datcheck error to be sent over TX	RW	0b0
[46]	la_cml_smp_tx_hold_on_datacheck_err	When set holds CML_SMP TX from sending CXS flits when datacheck has a parity error. When set, will halt TX (not restartable) so that the flit with datacheck error does not get sent on CXS. When clear this allows flits with datcheck error to be sent over TX	RW	0b0
[45]	la_c2c_mode_en	When set enables CHI C2C mode	RW	0b0
[44:40]	ccg_gcid	Global Chip ID	RW	0b00000
[39]	ccg_gcid_en	enable indicating if CCG-RA needs to send global chip id (gcid) for non-caching agent request	RW	0b0
[38:36]	ccg_baseid	3-bit id that is unique within the connected physical ccix link. this will be send as LSB (3-bit) of SRCID while sending a packet accross ccix link	RW	0b000
[35]	txrsp_byp_tgtid_cam	Bypass the Tgtid CAM for finding the tx rsp port to be used.	RW	0x0
[34:27]	linkend_arb_cnt	The count for how long each linkend is selected during linkend arbitration. If linkend_arb_cnt=8, each linkend is active for 8 cycles before switching to the next linkend	RW	0x10

Bits	Name	Description	Type	Reset
[26:25]	ha_cxl_type	Used to program CXL Type for HA  <b>0b00</b> Reserved <b>0b01</b> Type1 <b>0b10</b> Type2 <b>0b11</b> Type3	RW	0b01
[24:22]	link2_weight	Determines weight of link2 in CCL linkend arbitration	RW	0b001
[21:19]	link1_weight	Determines weight of link1 in CCL linkend arbitration	RW	0b001
[18:16]	link0_weight	Determines weight of link0 in CCL linkend arbitration	RW	0b001
[15:11]	Reserved	Reserved	RO	-
[10]	smp_link2_viral_prop_en	When set, enables viral propagation on SMP link2	RW	0b0
[9]	smp_link1_viral_prop_en	When set, enables viral propagation on SMP link1	RW	0b0
[8]	smp_link0_viral_prop_en	When set, enables viral propagation on SMP link0	RW	0b0
[7:6]	Reserved	Reserved	RO	-
[5]	cxl_cache_en	Enable CXL .cache mode, by default is disabled	RW	0b0
[4]	cxl_mem_en	Enable CXL .mem mode, by default is enabled	RW	0b1
[3:2]	cxl_type	Used to program CXL Type for RA  <b>0b00</b> Reserved <b>0b01</b> Type1 <b>0b10</b> Type2 <b>0b11</b> Type3	RW	0b11
[1]	la_device_mode_en	Enable the Device mode, by default set to Host mode	RW	0b0
[0]	la_cxl_mode_en	When set enables CXL mode	RW	0b0

### 8.3.4.7 por\_ccla\_aux\_ctl

Functions as the auxiliary control register for CCLA.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xB08

## Type

RW

## Reset value

See individual bit resets

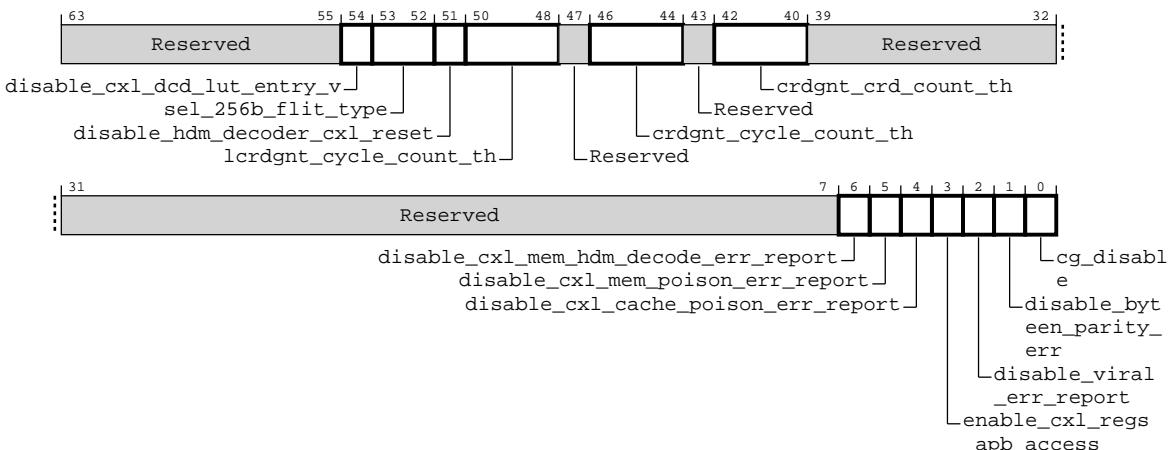
## Usage constraints

This register is owned in the Root space and is 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 8-100: por\_ccla\_aux\_ctl**



**Table 8-101: por\_ccla\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54]	disable_cxl_dcd_lut_entry_v	When set, ignores the entry valid information read from the DCD LUT. Applicable only in CXL Type3 Device mode with Dynamic Capacity Device support enabled	RW	0b1
[53:52]	sel_256b_flit_type	selects the Flit Type[1:0] in 256B flit header for CXL 3 and SMP <b>0b00</b> 00b Physical Layer <b>NOP</b> or Idle flit or CXL.io <b>NOP</b> <b>0b01</b> 01b CXL.io Payload flit <b>0b10</b> 10b CXL.cachemem Payload flit or CXL.cachemen empty flit <b>0b11</b> 11b ALMP arb/mux link anagement packet	RW	0b10

Bits	Name	Description	Type	Reset
[51]	disable_hdm_decoder_cxl_reset	When set, disables resetting HDM decoder registers on CXL reset	RW	0b0
[50:48]	lcrdgnt_cycle_count_th	<p>Maximum number of cycles that need to be elapsed since previous piggyback credits were sent, to send a link credit grant message</p> <p><b>0b000</b> 8 cycles</p> <p><b>0b001</b> 16 cycles</p> <p><b>0b010</b> 32 cycles</p> <p><b>0b011</b> 64 cycles</p>	RW	0b001
[47]	Reserved	Reserved	RO	-
[46:44]	crdgnt_cycle_count_th	<p>Maximum number of cycles that need to be elapsed since previous piggyback credits were sent, to send a protocol (Req, Dat, Snp..) credit grant message</p> <p><b>0b000</b> 8 cycles</p> <p><b>0b001</b> 16 cycles</p> <p><b>0b010</b> 32 cycles</p> <p><b>0b011</b> 64 cycles</p> <p><b>0b100</b> 128 cycles</p> <p><b>0b101</b> 256 cycles</p>	RW	0b100
[43]	Reserved	Reserved	RO	-
[42:40]	crdgnt_crd_count_th	<p>Maximum number of protocol credits (i.e. Req, Dat, Snp..) that need to be accumulated to send a credit grant message</p> <p><b>0b000</b> 4 cycles</p> <p><b>0b001</b> 8 cycles</p> <p><b>0b010</b> 16 credits</p> <p><b>0b011</b> 32 credits</p> <p><b>0b100</b> 64 credits</p> <p><b>0b101</b> 128 credits</p>	RW	0b100
[39:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6]	disable_cxl_mem_hdm_decode_err_report	When set, disables cxl HDM decode errors reporting through CMN's error reporting mechanism	RW	0b0
[5]	disable_cxl_mem_poison_err_report	When set, disables cxl mem poison err reporting through CMN's error reporting mechanism	RW	0b0
[4]	disable_cxl_cache_poison_err_report	When set, disables cxl cache poison err reporting through CMN's error reporting mechanism	RW	0b0
[3]	enable_cxl_regs_apb_access	When set, Enables the APB access to the CXL registers and Disables the CMN access	RW	0b0
[2]	disable_viral_err_report	When set, disables viral error reporting through CMN's error reporting mechanism	RW	0b1
[1]	disable_byteen_parity_err	Disables CCLA RX RAM byte enable parity errors	RW	0b0
[0]	cg_disable	Disables CCLA architectural clock gates	RW	0b0

### 8.3.4.8 por\_ccla\_c2c\_packing\_ctl

Controls CCLA C2C Packing Logic Arbiter Control

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xB10

##### Type

RW

##### Reset value

See individual bit resets

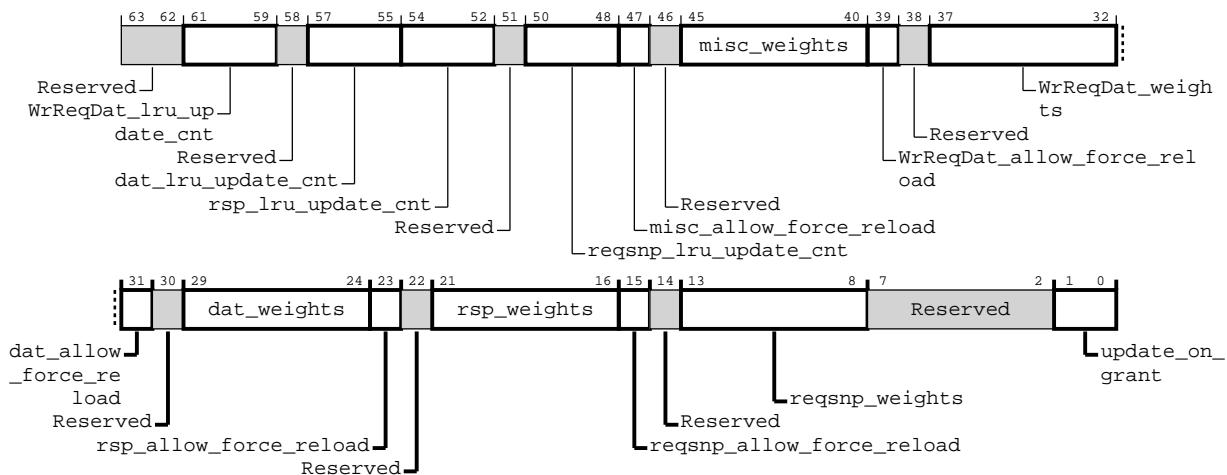
##### Usage constraints

This register is owned in the Root space and is 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 8-101: por\_ccla\_c2c\_packing\_ctl**



**Table 8-102: por\_ccla\_c2c\_packing\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	
[61:59]	WrReqDat_lru_update_cnt	Max consecutive WrReqDat Message class requests to pack before updating LRU order  <b>0b000</b> Update LRU order after 1 WrReqDat Message packed  <b>0b001</b> Update LRU order after 2 WrReqDat Message packed  <b>0b010</b> Update LRU order after 3 WrReqDat Message packed  <b>0b011</b> Update LRU order after 4 WrReqDat Message packed  <b>0b100</b> Update LRU order after 5 WrReqDat Message packed  <b>0b101</b> Update LRU order after 5 WrReqDat Message packed  <b>0b110</b> Update LRU order after 7 WrReqDat Message packed  <b>0b111</b> Update LRU order after 8 WrReqDat Message packed	RW	0x3
[58]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[57:55]	dat_lru_update_cnt	<p>Max consecutive Dat Message class requests to pack before updating LRU order</p> <p><b>0b000</b> Update LRU order after 1 Dat Message packed</p> <p><b>0b001</b> Update LRU order after 2 Dat Message packed</p> <p><b>0b010</b> Update LRU order after 3 Dat Message packed</p> <p><b>0b011</b> Update LRU order after 4 Dat Message packed</p> <p><b>0b100</b> Update LRU order after 5 Dat Message packed</p> <p><b>0b101</b> Update LRU order after 5 Dat Message packed</p> <p><b>0b110</b> Update LRU order after 7 Dat Message packed</p> <p><b>0b111</b> Update LRU order after 8 Dat Message packed</p>	RW	0x3
[54:52]	rsp_lru_update_cnt	<p>Max consecutive Response Message class requests to pack before updating LRU order</p> <p><b>0b000</b> Update LRU order after 1 Response Message packed</p> <p><b>0b001</b> Update LRU order after 2 Response Message packed</p> <p><b>0b010</b> Update LRU order after 3 Response Message packed</p> <p><b>0b011</b> Update LRU order after 4 Response Message packed</p> <p><b>0b100</b> Update LRU order after 5 Response Message packed</p> <p><b>0b101</b> Update LRU order after 5 Response Message packed</p> <p><b>0b110</b> Update LRU order after 7 Response Message packed</p> <p><b>0b111</b> Update LRU order after 8 Response Message packed</p>	RW	0x3
[51]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[50:48]	reqsnp_lru_update_cnt	<p>Max consecutive Request/Snoop Message class requests to pack before updating LRU order</p> <p><b>0b000</b> Update LRU order after 1 Request/Snoop Message packed</p> <p><b>0b001</b> Update LRU order after 2 Request/Snoop Message packed</p> <p><b>0b010</b> Update LRU order after 3 Request/Snoop Message packed</p> <p><b>0b011</b> Update LRU order after 4 Request/Snoop Message packed</p> <p><b>0b100</b> Update LRU order after 5 Request/Snoop Message packed</p> <p><b>0b101</b> Update LRU order after 5 Request/Snoop Message packed</p> <p><b>0b110</b> Update LRU order after 7 Request/Snoop Message packed</p> <p><b>0b111</b> Update LRU order after 8 Request/Snoop Message packed</p>	RW	0x3
[47]	misc_allow_force_reload	<p>Allow reloading Misc Message class weights even when non-zero</p> <p><b>0</b> Reload Misc Message class weights only when exhausted</p> <p><b>1</b> Allow reloading Misc Message class weights even when non-zero in case all other message types have exhausted their weights</p>	RW	0b1
[46]	Reserved	Reserved	RO	
[45:40]	misc_weights	Number of Misc Message class messages allowed	RW	0x8
[39]	WrReqDat_allow_force_reload	<p>Allow reloading WrReqDat Message class weights even when non-zero</p> <p><b>0</b> Reload WrReqDat Message class weights only when exhausted</p> <p><b>1</b> Allow reloading WrReqDat Message class weights even when non-zero in case all other message types have exhausted their weights</p>	RW	0b0
[38]	Reserved	Reserved	RO	
[37:32]	WrReqDat_weights	Number of WrReqDat Message class messages allowed	RW	0x32
[31]	dat_allow_force_reload	<p>Allow reloading Dat Message class weights even when non-zero</p> <p><b>0</b> Reload Dat Message class weights only when exhausted</p> <p><b>1</b> Allow reloading Dat Message class weights even when non-zero in case all other message types have exhausted their weights</p>	RW	0b0
[30]	Reserved	Reserved	RO	
[29:24]	dat_weights	Number of Dat Message class messages allowed	RW	0x32

Bits	Name	Description	Type	Reset
[23]	rsp_allow_force_reload	Allow reloading Response Message class weights even when non-zero <b>0</b> Reload Response Message class weights only when exhausted <b>1</b> Allow reloading Response Message class weights even when non-zero in case all other message types have exhausted their weights	RW	0b0
[22]	Reserved	Reserved	RO	
[21:16]	rsp_weights	Number of Response Message class messages allowed	RW	0x32
[15]	reqsnp_allow_force_reload	Allow reloading Request/Snoop Message class weights even when non-zero <b>0</b> Reload Request/Snoop Message class weights only when exhausted <b>1</b> Allow reloading Request/Snoop Message class weights even when non-zero in case all other message types have exhausted their weights	RW	0b0
[14]	Reserved	Reserved	RO	
[13:8]	reqsnp_weights	Number of Request/Snoop Message class messages allowed	RW	0x32
[7:2]	Reserved	Reserved	RO	
[1:0]	update_on_grant	Controls when LRU arbitration order is updated <b>0b00</b> Update LRU order after Message Class weight exhausted <b>0b01</b> Update LRU order after every grant <b>0b10</b> Update LRU order after packing max_message limit <b>0b11</b> Reserved	RW	0b01

### 8.3.4.9 por\_ccla\_ccix\_prop\_capabilities

Contains CCIX-supported properties.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC00

##### Type

RO

### Reset value

See individual bit resets

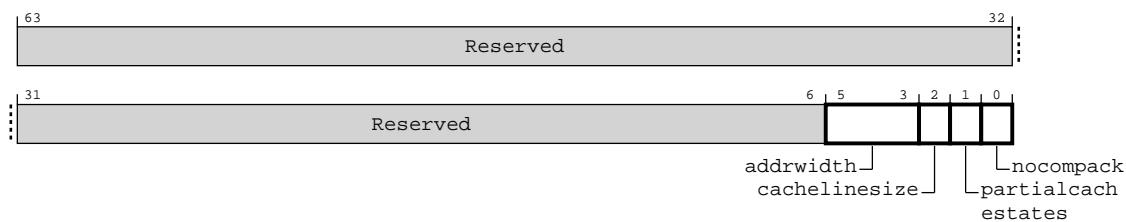
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-102: por\_ccla\_ccix\_prop\_capabilities**



**Table 8-103: por\_ccla\_ccix\_prop\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:3]	addrwidth	Address width supported <b>0b000</b> 48b <b>0b001</b> 52b <b>0b010</b> 56b <b>0b011</b> 60b <b>0b100</b> 64b	RO	0b001
[2]	cachelinesize	Cacheline size supported <b>0b0</b> 64B <b>0b1</b> 128B	RO	0b0
[1]	partialcachestates	Partial cache states supported <b>0b0</b> False <b>0b1</b> True	RO	0b0

Bits	Name	Description	Type	Reset
[0]	nocompack	No CompAck supported  <b>0b0</b> False  <b>0b1</b> True	RO	0b1

### 8.3.4.10 por\_ccla\_cxs\_attr\_capabilities

Contains CXS supported attributes.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC08

##### Type

RO

##### Reset value

See individual bit resets

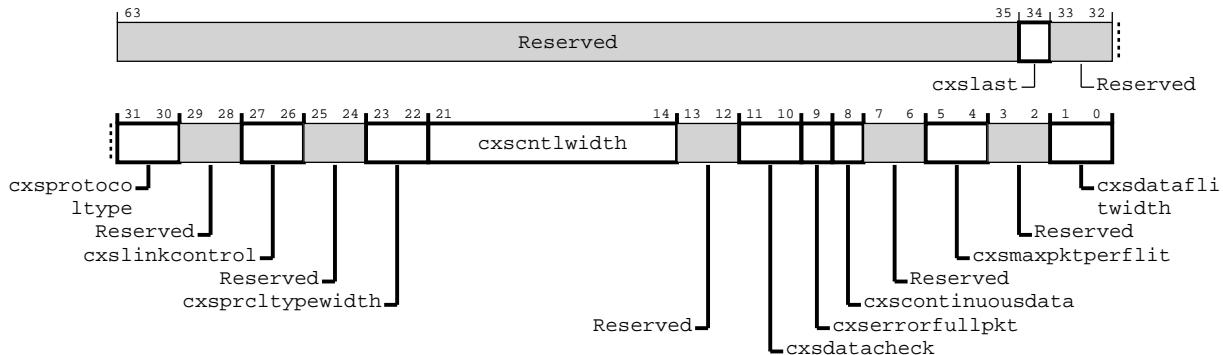
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-103: por\_ccla\_cxs\_attr\_capabilities**



**Table 8-104: por\_ccla\_cxs\_attr\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34]	cxslast	CXS LAST signal is supported	RO	0x0
[33:32]	Reserved	Reserved	RO	-
[31:30]	cxsprotocoltype	CXS Protocol type signal is supported	RO	0x0
[29:28]	Reserved	Reserved	RO	-
[27:26]	cxslinkcontrol	Set to Explicit Credit Return.	RO	0x00
[25:24]	Reserved	Reserved	RO	-
[23:22]	cxsprltypewidth	Width of CXS TX/RX control	RO	0x0
[21:14]	cxsctlwidth	Width of CXS TX/RX control	RO	Configuration dependent
[13:12]	Reserved	Reserved	RO	-
[11:10]	cxsdatacheck	CXS datacheck supported <b>0b00</b> None <b>0b01</b> Parity <b>0b10</b> SECDED	RO	0x0
[9]	cxserrorfullpkt	CXS error full packet supported <b>0b0</b> False <b>0b1</b> True	RO	0x0
[8]	cxscontinuousdata	CXS continuous data supported <b>0b0</b> False <b>0b1</b> True	RO	0x0
[7:6]	Reserved	Reserved	RO	-
[5:4]	cxsmaxpktperflit	CXS maximum packets per flit supported <b>0b00</b> 2 <b>0b01</b> 3 <b>0b10</b> 4	RO	0x0
[3:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1:0]	cxsdataflitwidth	CXS data flit width supported <b>0b00</b> 256b <b>0b01</b> 512b <b>0b10</b> 1024b	RO	Configuration dependent

### 8.3.4.11 por\_ccla\_err\_agent\_id

Contains Error Agent ID. Must be programmed by CCIX discovery s/w. Used as TargetID on CCIX Protocol Error (PER) Message.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD28

##### Type

RW

##### Reset value

See individual bit resets

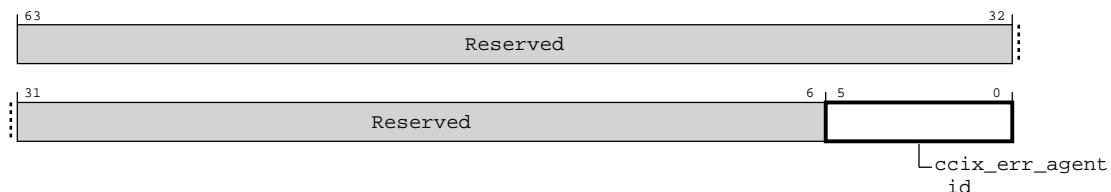
##### Usage constraints

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 8-104: por\_ccla\_err\_agent\_id**



**Table 8-105: por\_ccla\_err\_agent\_id attributes**

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	
[5:0]	ccix_err_agent_id	CCIX Error AgentID	RW	0b0

### 8.3.4.12 por\_ccla\_agentid\_to\_portid\_reg0

Specifies the mapping of Agent ID to Port ID for Agent IDs 0 to 7.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD30

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.portid\_ctl

##### Secure group override

por\_ccla\_scr.portid\_ctl

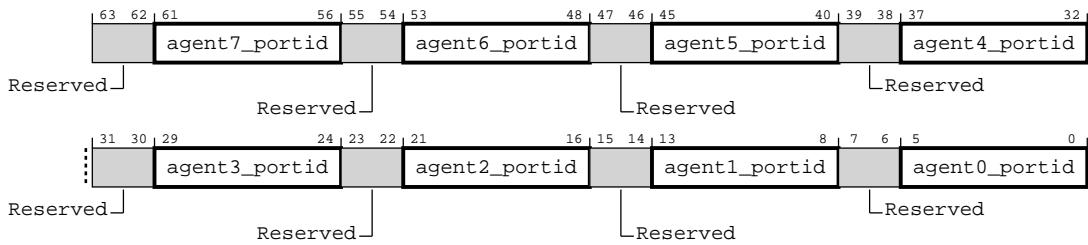
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.portid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.portid\_ctl bit and por\_ccla\_rcr.portid\_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 8-105: por\_ccla\_agentid\_to\_portid\_reg0**



**Table 8-106: por\_ccla\_agentid\_to\_portid\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	
[61:56]	agent7_portid	Specifies the Port ID for Agent ID 7	RW	0x0
[55:54]	Reserved	Reserved	RO	
[53:48]	agent6_portid	Specifies the Port ID for Agent ID 6	RW	0x0
[47:46]	Reserved	Reserved	RO	
[45:40]	agent5_portid	Specifies the Port ID for Agent ID 5	RW	0x0
[39:38]	Reserved	Reserved	RO	
[37:32]	agent4_portid	Specifies the Port ID for Agent ID 4	RW	0x0
[31:30]	Reserved	Reserved	RO	
[29:24]	agent3_portid	Specifies the Port ID for Agent ID 3	RW	0x0
[23:22]	Reserved	Reserved	RO	
[21:16]	agent2_portid	Specifies the Port ID for Agent ID 2	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	agent1_portid	Specifies the Port ID for Agent ID 1	RW	0x0
[7:6]	Reserved	Reserved	RO	
[5:0]	agent0_portid	Specifies the Port ID for Agent ID 0	RW	0x0

### 8.3.4.13 por\_ccla\_agentid\_to\_portid\_val

Specifies which Agent ID to Port ID mappings are valid.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD70

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.portid\_ctl

### Secure group override

por\_ccla\_scr.portid\_ctl

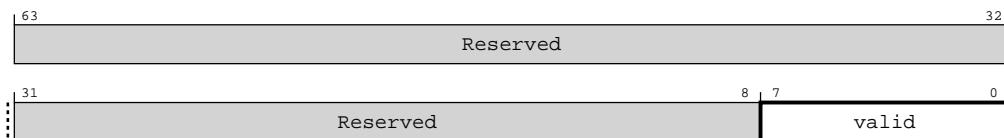
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.portid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.portid\_ctl bit and por\_ccla\_rcr.portid\_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 8-106: por\_ccla\_agentid\_to\_portid\_val**



**Table 8-107: por\_ccla\_agentid\_to\_portid\_val attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:0]	valid	Specifies whether the Port ID is valid; bit number corresponds to logical Agent ID number (from 0 to 63)	RW	0x0

### 8.3.4.14 por\_ccla\_portfwd\_en

Functions as the Port-to-Port forwarding control register. Works with por\_ccla\_portfwd\_status.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

0xD78

### Type

RW

### Reset value

See individual bit resets

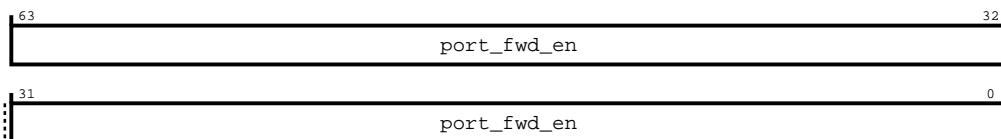
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-107: por\_ccla\_portfwd\_en**



**Table 8-108: por\_ccla\_portfwd\_en attributes**

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_en	Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit when set, enables Port-to-Port forwarding with the corresponding port.  <b>0b0</b> Port-to-Port forwarding is disabled  <b>0b1</b> Port-to-Port forwarding is enabled	RW	0b0

### 8.3.4.15 por\_ccla\_ide\_freq\_factor

Register contents are used as ratio of freq between Host and device.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x1500

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

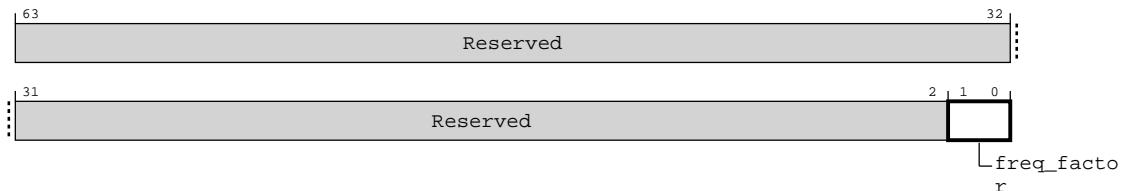
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-108: por\_ccla\_ide\_freq\_factor**



**Table 8-109: por\_ccla\_ide\_freq\_factor attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	freq_factor	<p>Freq factor encodings. This field is used by ULL to determine the number of cycles it has to wait during the IDE key programming stage or after sending IDE TMAC to start a new MAC epoch. For example, If CMN(ULL) is running at 2GHz and LLL is running at 1GHZ, select freq factor 0b01 which gives a 1:2 ratio. This freq factor determines the number of cycles based on the number of flits programmed either in key_refresh or truncation transmit delay registers.</p> <p><b>0b00</b> 1:1</p> <p><b>0b01</b> 1:2</p> <p><b>0b10</b> 1:3</p> <p><b>0b11</b> 1:4</p>	RW	0x0

### 8.3.4.16 por\_ccla ull\_idle\_counter

The number of cycles ull needs to be down for it to be counted as idle

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1458

##### Type

RW

##### Reset value

See individual bit resets

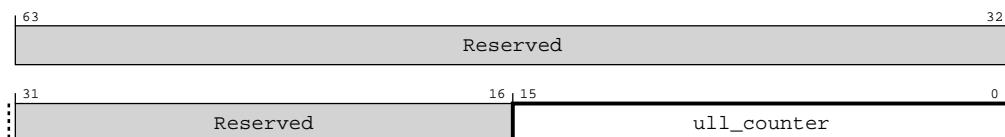
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-109: por\_ccla ull\_idle\_counter**



**Table 8-110: por\_ccla ull\_idle\_counter attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	ull_counter	ull counter	RW	0x20

### 8.3.4.17 por\_ccla\_portfwd\_status

Functions as the Port-to-Port forwarding status register. Works with por\_ccla\_portfwd\_ctl.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xD80

### Type

RO

### Reset value

See individual bit resets

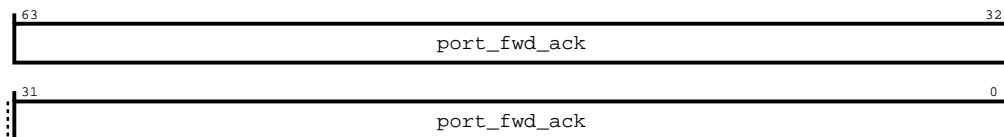
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-110: por\_ccla\_portfwd\_status**



**Table 8-111: por\_ccla\_portfwd\_status attributes**

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_ack	Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit represents the status of the port-to-port control request sent to the corresponding port.  0b0 Port-to-Port forwarding channel is de-active.  0b1 Port-to-Port forwarding channel is active	RO	0b0

### 8.3.4.18 por\_ccla\_portfwd\_req

Functions as the Port-to-Port forwarding control register. Works with por\_ccla\_portfwd\_status.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xD88

### Type

RW

### Reset value

See individual bit resets

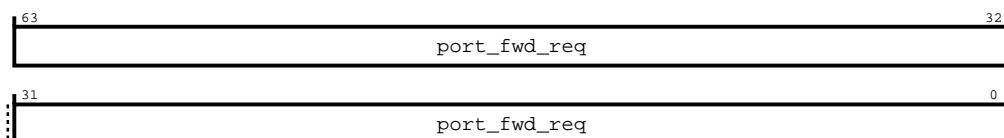
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-111: por\_ccla\_portfwd\_req**



**Table 8-112: por\_ccla\_portfwd\_req attributes**

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_req	<p>Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit is used to control the communication channel with the corresponding port.</p> <p><b>0b0</b> Port-to-Port forwarding channel de-activate request</p> <p><b>0b1</b> Port-to-Port forwarding channel activate request</p>	RW	0b0

### 8.3.4.19 por\_ccla\_linkid\_to\_hops

Specifies number of portforward hops for the linkid.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD90

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.portid\_ctl

### Secure group override

por\_ccla\_scr.portid\_ctl

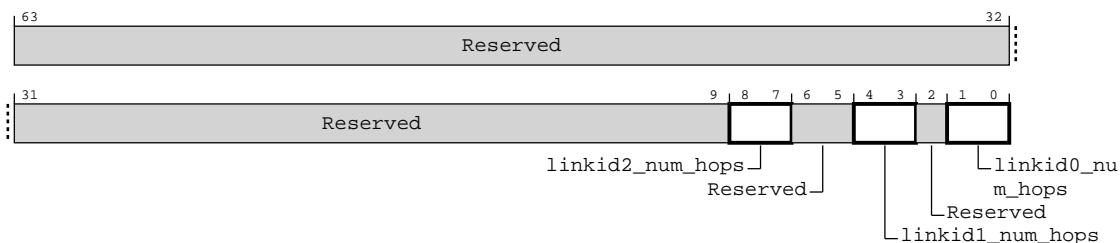
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.portid\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.portid\_ctl bit and por\_ccla\_rcr.portid\_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 8-112: por\_ccla\_linkid\_to\_hops**



**Table 8-113: por\_ccla\_linkid\_to\_hops attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8:7]	linkid2_num_hops	Specifies the number of portforward hops for linkid2	RW	0b00
[6:5]	Reserved	Reserved	RO	
[4:3]	linkid1_num_hops	Specifies the number of portforward hops for linkid1	RW	0b00
[2]	Reserved	Reserved	RO	
[1:0]	linkid0_num_hops	Specifies the number of portforward hops for linkid0	RW	0b00

### 8.3.4.20 por\_ccla\_cxl\_link\_rx\_credit\_ctl

CXL Link Rx Credit Control Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

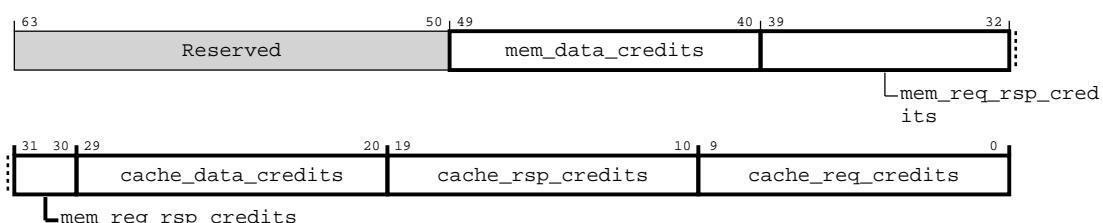
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-113: por\_ccla\_cxl\_link\_rx\_credit\_ctl**



**Table 8-114: por\_ccla\_cxl\_link\_rx\_credit\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Credits to advertise for Mem Data channel at init	RW	0x0

Bits	Name	Description	Type	Reset
[39:30]	mem_req_rsp_credits	Credits to advertise for Mem Request or Response channel at init	RW	0x0
[29:20]	cache_data_credits	Credits to advertise for Cache Data channel at init	RW	Configuration dependent
[19:10]	cache_rsp_credits	Credits to advertise for Cache Response channel at init	RW	0x0
[9:0]	cache_req_credits	Credits to advertise for Cache Request channel at init	RW	0x100

### 8.3.4.21 por\_ccla\_cxl\_link\_rx\_credit\_return\_stat

CXL Link Rx Credit Return Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE08

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

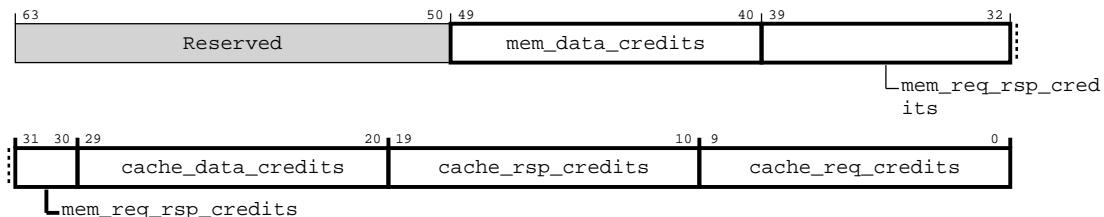
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-114: por\_ccla\_cxl\_link\_rx\_credit\_return\_stat**



**Table 8-115: por\_ccla\_cxl\_link\_rx\_credit\_return\_stat attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits to be returned	RO	0b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits to be returned	RO	0b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits to be returned	RO	0b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits to be returned	RO	0b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits to be returned	RO	0b0

### 8.3.4.22 por\_ccla\_cxl\_link\_tx\_credit\_stat

CXL Link Tx Credit Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE10

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

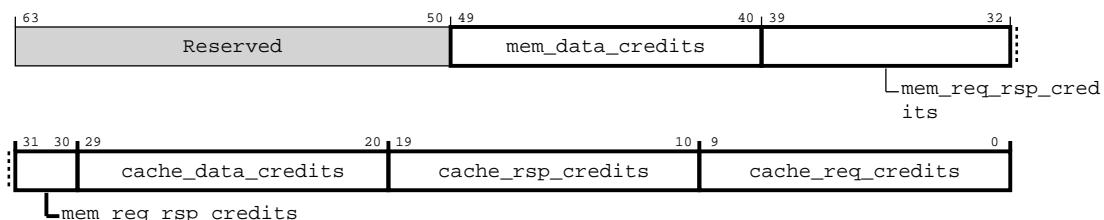
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-115: por\_ccla\_cxl\_link\_tx\_credit\_stat**



**Table 8-116: por\_ccla\_cxl\_link\_tx\_credit\_stat attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits for TX	RO	0b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits for TX	RO	0b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits for TX	RO	0b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits for TX	RO	0b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits for TX	RO	0b0

### 8.3.4.23 por\_ccla\_cxl\_security\_policy

Contains CXL Security Policy

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE50

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

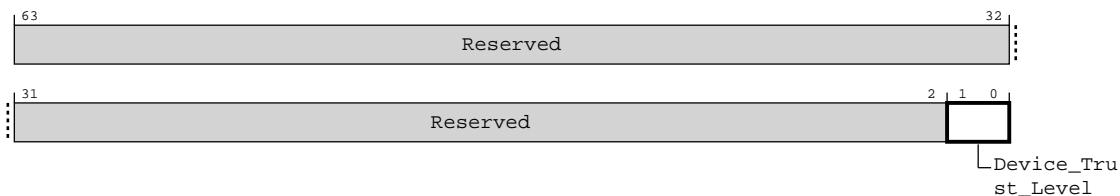
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-116: por\_ccla\_cxl\_security\_policy**



**Table 8-117: por\_ccla\_cxl\_security\_policy attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	Device_Trust_Level	0 -> Trusted CXL Device. At this setting a CXL Device will be able to get access on CXL.cache for both host-attached and device attached memory ranges. The Host can still protect security sensitive memory regions.  '1 -> Trusted for Device Attached Memory Range Only. At this setting a CXL Device will be able to get access on CXL.cache for device attached memory ranges only. Requests on CXL.cache for host-attached memory ranges will be aborted by the Host.  '2 -> Untrusted CXL Device. At this setting all requests on CXL.cache will be aborted by the Host. Please note that these settings only apply to requests on CXL.cache. The device can still source requests on CXL.io regardless of these settings. Protection on CXL.io will be implemented using IOMMU based page tables. Default value of this field is 2.	RW	0x2

### 8.3.4.24 por\_ccla\_cxl\_hdm\_decoder\_capability

Contains CXL\_HDM\_Decoder\_Capability\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xE78

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

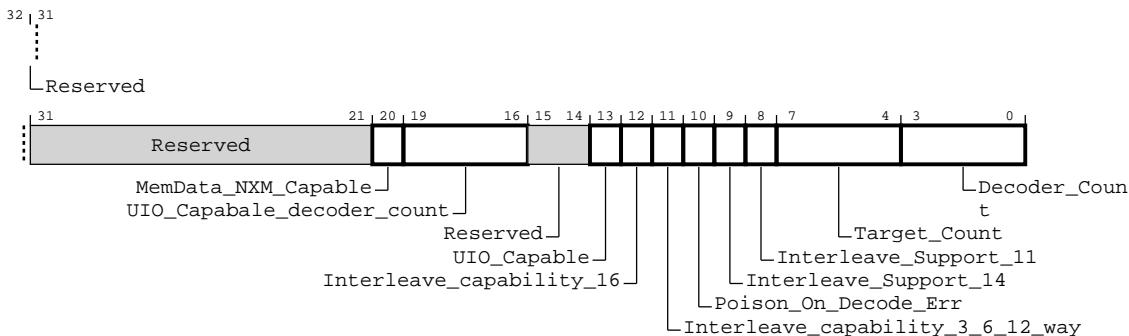
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-117: por\_ccla\_cxl\_hdm\_decoder\_capability**



**Table 8-118: por\_ccla\_cxl\_hdm\_decoder\_capability attributes**

Bits	Name	Description	Type	Reset
[31:21]	Reserved	Reserved	RO	
[20]	MemData_NXM_Capable	If set, the component supports MemData-NXM opcode. If cleared, the component does not support MemData-NXM opcode. All 256B Flit mode-capable components shall set this bit to 1.	RO	0b1
[19:16]	UIO_Capabale_decoder_count	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x0
[15:14]	Reserved	Reserved	RO	
[13]	UIO_Capable	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x0
[12]	Interleave_capability_16	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1
[11]	Interleave_capability_3_6_12_way	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1
[10]	Poison_On_Decode_Err	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1
[9]	Interleave_Support_14	If set the component supports interleaving based on Address bit 14 Address bit 13 and Address bit 12. Root ports and switches shall always set this bit indicating support for interleaving based on Address bits 14-12.	RO	0x0
[8]	Interleave_Support_11	If set the component supports interleaving based on Address bit 11 Address bit 10 Address bit 9 and Address bit 8. Root Ports and Upstream Switch Ports shall always set this bit indicating support for interleaving based on Address bit 11-8.	RO	0x0
[7:4]	Target_Count	The number of target ports each decoder supports (applicable to Upstream Switch Port and Root Port only). Maximum of 8. 1 1 target port 2 2 target ports 4 4 target ports 8 8 target ports All other values are reserved	RO	0x1
[3:0]	Decoder_Count	Reports the number of memory address decoders implemented by the component. 0 1 Decoder 1 2 Decoders 2 4 Decoders 36 Decoders 48 Decoders 510 Decoders All other values are reserved	RO	0x4

### 8.3.4.25 por\_ccla\_cxl\_hdm\_decoder\_global\_control

Contains CXL\_HDM\_Decoder\_Global\_Control\_Register . Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

## Address offset

0xE80

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_ccla\_rcr.cxllink\_ctl

## Secure group override

por\_ccla\_scr.cxllink\_ctl

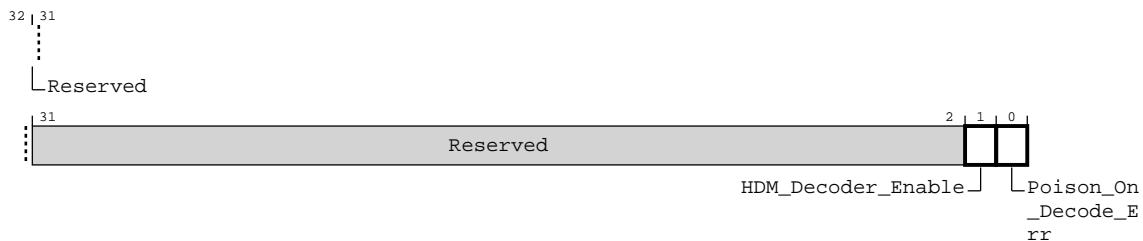
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-118: por\_ccla\_cxl\_hdm\_decoder\_global\_control**



**Table 8-119: por\_ccla\_cxl\_hdm\_decoder\_global\_control attributes**

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Reserved	RO	
[1]	HDM_Decoder_Enable	This bit is only applicable to CXL.mem devices and shall return 0 on Root Ports and Upstream Switch Ports. When this bit is set device shall use HDM decoders to decode CXL.mem transactions and not use HDM Base registers in DVSEC ID 0. Root Ports and Upstream Switch Ports always use HDM Decoders to decode CXL.mem transactions.	RW	0x0
[0]	Poison_On_Decode_Error	This bit is RO and is hard-wired to 0 if Poison On Decode Error Capability=0. If set the component returns poison on read access to addresses that are not positively decoded by the component. If clear the component returns all 1s data without a poison under such scenarios.	RW	0x0

### 8.3.4.26 por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_0\_Base\_High\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1040 + #{8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

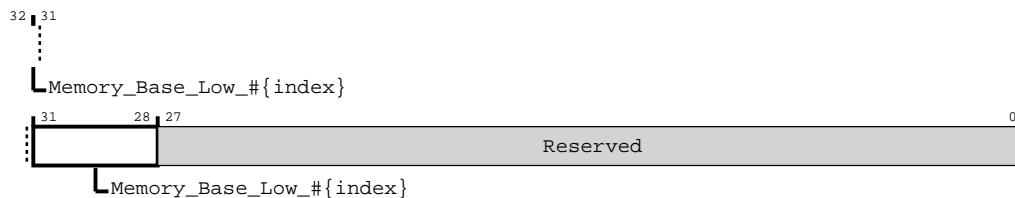
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-119: por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_low**



**Table 8-120: por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low_{index}	Corresponds to bits 31:28 of the base of the address range managed by decoder 0	RWL	0x0

Bits	Name	Description	Type	Reset
[27:0]	Reserved	Reserved	RO	

### 8.3.4.27 por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Base\_Low\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1080 + #{8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

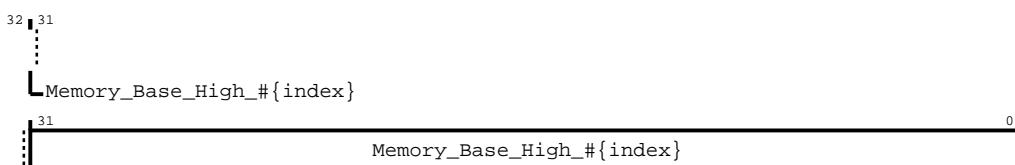
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-120: por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_high**



**Table 8-121: por\_ccla\_cxl\_hdm\_decoder\_0-7\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High_{index}	Corresponds to bits 63:32 of the base of the address range managed by decoder 0.	RWL	0x0

### 8.3.4.28 por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Size\_Low\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x10C0 + #8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

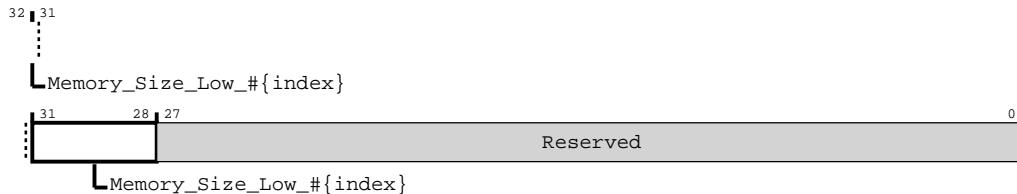
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-121: por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_low**



**Table 8-122: por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low_{index}	Corresponds to bits 31:28 of the size of the address range managed by decoder 0	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.4.29 por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Size\_High\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1100 + #{8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

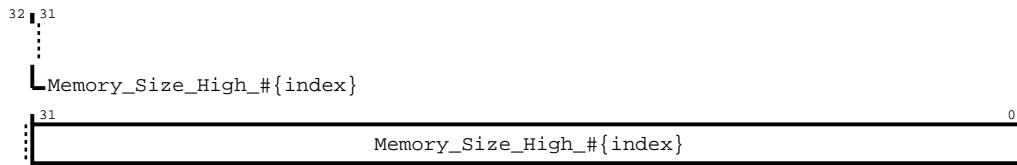
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-122: por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_high**



**Table 8-123: por\_ccla\_cxl\_hdm\_decoder\_0-7\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High_{index}	Corresponds to bits 63:32 of the size of address range managed by decoder 0.	RWL	0x0

### 8.3.4.30 por\_ccla\_cxl\_hdm\_decoder\_0-7\_control

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Control\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1140 + #{8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

#### Usage constraints

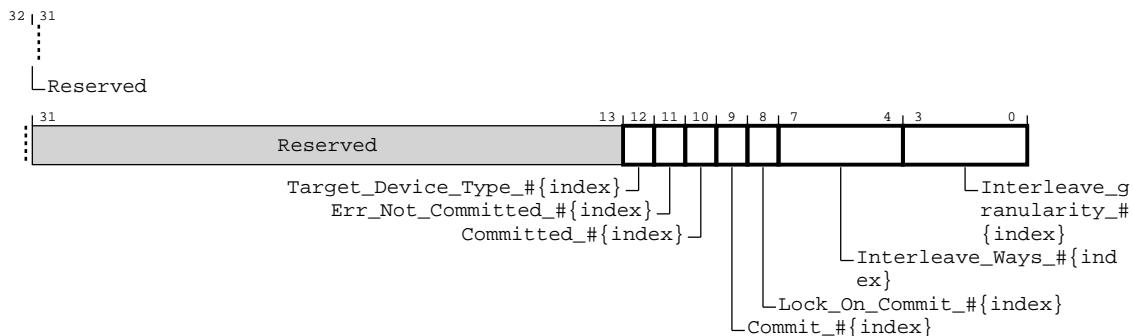
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to

this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-123: por\_ccla\_cxl\_hdm\_decoder\_0-7\_control**



**Table 8-124: por\_ccla\_cxl\_hdm\_decoder\_0-7\_control attributes**

Bits	Name	Description	Type	Reset
[31:13]	Reserved	Reserved	RO	
[12]	Target_Device_Type_{index}	<p><b>0</b> Target is a CXL Type 2 Device</p> <p><b>1</b> Target is a CXL Type 3 Device</p>	RWL	0x0
[11]	Err_Not_Committed_{index}	Indicates the decode programming had an error and decoder is not active.	RWL	0x0
[10]	Committed_{index}	Indicates the decoder is active	RWL	0x0
[9]	Commit_{index}	Software sets this to 1 to commit this decoder	RWL	0x0
[8]	Lock_On_Commit_{index}	If set all RWL fields in Decoder 0 registers will become read only when Committed changes to 1.	RWL	0x0
[7:4]	Interleave_Ways_{index}	<p>The number of targets across which this memory range is interleaved.</p> <p><b>0</b> 1 way</p> <p><b>1</b> 2 way</p> <p><b>2</b> 4 way</p> <p><b>3</b> 8 way</p> <p><b>Others</b> All reserved</p>	RWL	0x0

Bits	Name	Description	Type	Reset
[3:0]	Interleave_granularity_{index}	The number of consecutive bytes that are assigned to each target in the Target List.	RWL	0x0
0		256 Bytes		
1		512 Bytes		
2		1024 Bytes (1KB)		
3		2048 Bytes (2KB)		
4		4096 Bytes (4KB)		
5		8192 Bytes (8KB)		
4		16384 Bytes (16KB)		

### 8.3.4.31 por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_DPA\_Skip\_Low\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1180 + #{8\*index}

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

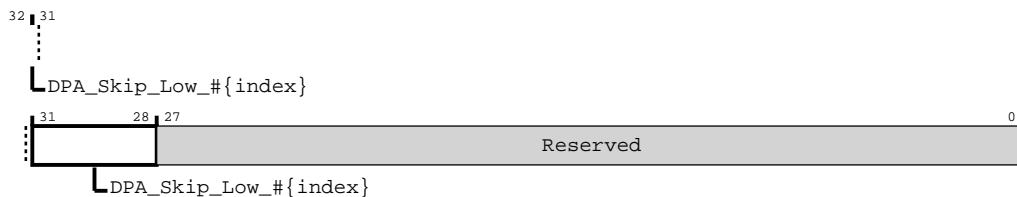
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-124: por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low**



**Table 8-125: por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	DPA_Skip_Low_{index}	Corresponds to bits 31:28 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.4.32 por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_DPA\_Skip\_High\_Register. Only applicable for Device. Host does not use this register

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x11C0 + #{8\*index}

### Type

RWL

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

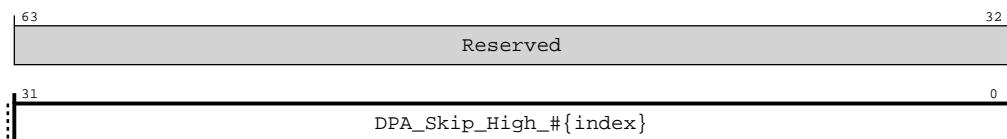
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-125: por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high**



**Table 8-126: por\_ccla\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	DPA_Skip_High_{index}	Corresponds to bits 63:32 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	0x0

### 8.3.4.33 por\_ccla\_cxl\_dcd\_region\_base\_address\_low0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL Dynamic Capacity Device (DCD) Base Address Low\_{index}. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x1200 + #8\*index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

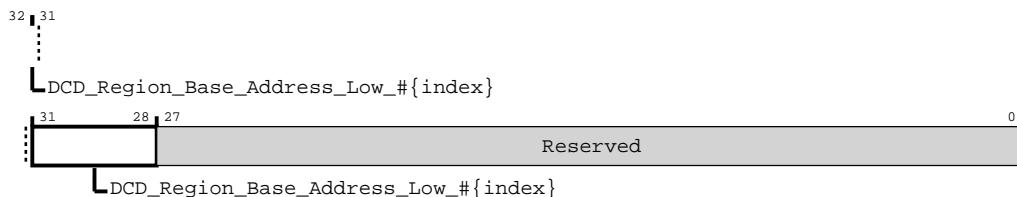
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-126: por\_ccla\_cxl\_dcd\_region\_base\_address\_low0-7**



**Table 8-127: por\_ccla\_cxl\_dcd\_region\_base\_address\_low0-7 attributes**

Bits	Name	Description	Type	Reset
[31:28]	DCD_Region_Base_Address_Low_{index}	Corresponds to bits 31:28 of the DCD Region base address	RW	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.4.34 por\_ccla\_cxl\_dcd\_region\_base\_address\_high0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL Dynamic Capacity Device (DCD) Base Address High\_{index}. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

$0x1240 + \{8 * \text{index}\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-127: por\_ccla\_cxl\_dcd\_region\_base\_address\_high0-7**



**Table 8-128: por\_ccla\_cxl\_dcd\_region\_base\_address\_high0-7 attributes**

Bits	Name	Description	Type	Reset
[31:0]	DCD_Region_Base_Address_High_{index}	Corresponds to bits 63:32 of the DCD Region base address	RW	0x0

### 8.3.4.35 por\_ccla\_cxl\_dcd\_region\_max\_address\_low0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL Dynamic Capacity Device (DCD) Max Address Low\_{index}. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1280 + #8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

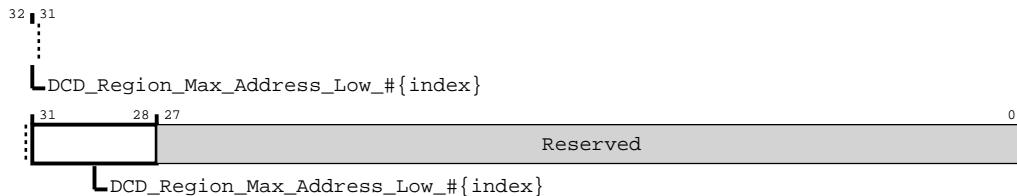
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-128: por\_ccla\_cxl\_dcd\_region\_max\_address\_low0-7**



**Table 8-129: por\_ccla\_cxl\_dcd\_region\_max\_address\_low0-7 attributes**

Bits	Name	Description	Type	Reset
[31:28]	DCD_Region_Max_Address_Low_{index}	Corresponds to bits 31:28 of the DCD Region Max address	RW	0x0

Bits	Name	Description	Type	Reset
[27:0]	Reserved	Reserved	RO	

### 8.3.4.36 por\_ccla\_cxl\_dcd\_region\_max\_address\_high0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL Dynamic Capacity Device (DCD) Max Address High\_{index}. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x12C0 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

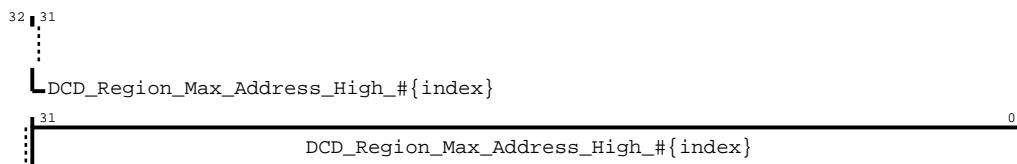
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-129: por\_ccla\_cxl\_dcd\_region\_max\_address\_high0-7**



**Table 8-130: por\_ccla\_cxl\_dcd\_region\_max\_address\_high0-7 attributes**

Bits	Name	Description	Type	Reset
[31:0]	DCDRegion_MaxAddressHigh_{index}	Corresponds to bits 63:32 of the DCD Region Max address	RW	0x0

### 8.3.4.37 por\_ccla\_cxl\_dcd\_region\_lut\_control0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL Dynamic Capacity Device (DCD) LUT Control\_{index}. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1300 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

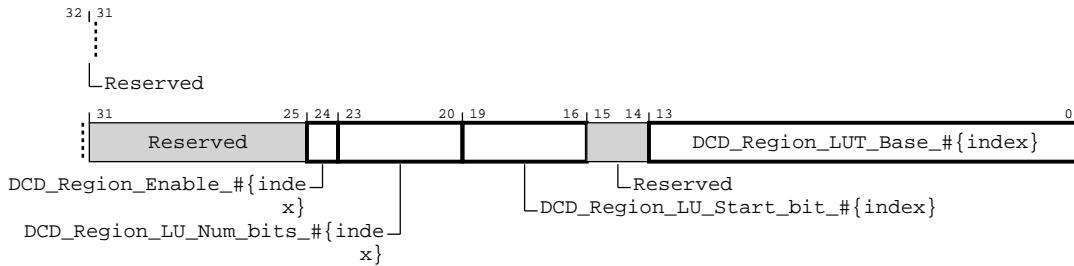
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-130: por\_ccla\_cxl\_dcd\_region\_lut\_control0-7**



**Table 8-131: por\_ccla\_cxl\_dcd\_region\_lut\_control0-7 attributes**

Bits	Name	Description	Type	Reset
[31:25]	Reserved	Reserved	RO	
[24]	DCD_Region_Enable_{index}	When set, enables DCD region	RW	0x0
[23:20]	DCD_Region_LU_Num_bits_{index}	Corresponds to number of DPA address bits to use for look-up in LUT for Region #{index} Minimum value for number of address LU bits is 8-bits which corresponds to 256 chunks per region Maximum value for number of address LU bits is 13-bits which corresponds to 8K chunks per region. All other values reserved	RW	0x8
[19:16]	DCD_Region_LU_Start_bit_{index}	Corresponds to DPA address bit where address to look-up in LUT starts for Region #{index}  <b>0b0000</b> DPA bit 26, corresponds to 64MB chunk size for region  <b>0b0001</b> DPA bit 27, corresponds to 128MB chunk size for region  <b>0b0010</b> DPA bit 28, corresponds to 256MB chunk size for region  <b>0b0011</b> DPA bit 29, corresponds to 512MB chunk size for region  <b>0b0100</b> DPA bit 30, corresponds to 1GB chunk size for region  <b>0b0101</b> DPA bit 31, corresponds to 2GB chunk size for region  <b>0b0110</b> DPA bit 32, corresponds to 4GB chunk size for region  <b>0b0111</b> DPA bit 33, corresponds to 8GB chunk size for region  All other values reserved	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:0]	DCD_Region_LUT_Base_{index}	Corresponds to base address for DCD Region #{index} in the LUT	RW	0x0

### 8.3.4.38 por\_ccla\_cxl\_dcd\_write\_intf

Write Interface to Dynamic Capacity LUT. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1340

##### Type

WO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

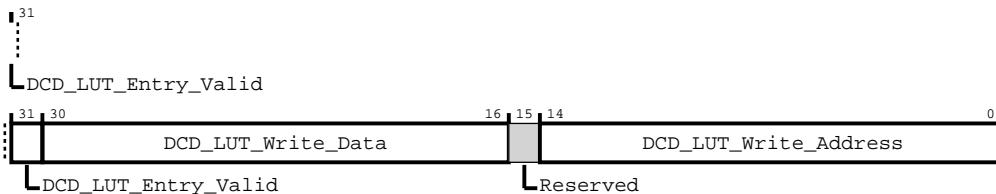
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-131: por\_ccla\_cxl\_dcd\_write\_intf**



**Table 8-132: por\_ccla\_cxl\_dcd\_write\_intf attributes**

Bits	Name	Description	Type	Reset
[31]	DCD_LUT_Entry_Valid	LUT Entry valid. Set this bit to indicate that the LUT entry is now ready to use. Clear this bit to invalidate LUT entry	WO	0x0

Bits	Name	Description	Type	Reset
[30:16]	DCD_LUT_Write_Data	Data that will be written to the LUT. Based on number of entries in the region being accessed, any unused most significant bits for this field must be set to zero	WO	0x0
[15]	Reserved	Reserved	RO	
[14:0]	DCD_LUT_Write_Address	Corresponds to LUT entry to write. Number of entries is limited by CCG_HA_DCD_LUT_DEPTH_PARAM	WO	0x0

### 8.3.4.39 por\_ccla\_cxl\_dcd\_read\_intf

Read Interface to Dynamic Capacity LUT. Only applicable for Device. Host does not use this register. To read LUT entry, first write to this register to program the read address. A subsequent read to this register will then return the data read from the LUT

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1348

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

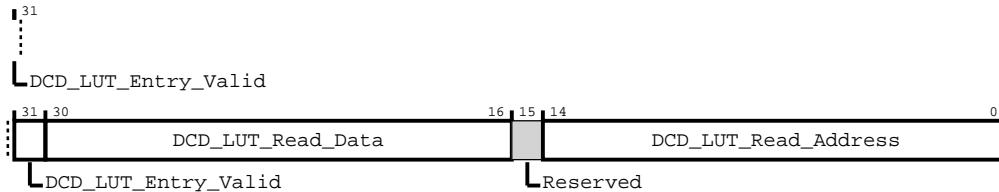
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-132: por\_ccla\_cxl\_dcd\_read\_intf**



**Table 8-133: por\_ccla\_cxl\_dcd\_read\_intf attributes**

Bits	Name	Description	Type	Reset
[31]	DCD_LUT_Entry_Valid	LUT Entry status	RW	0x0
[30:16]	DCD_LUT_Read_Data	Data read from the LUT. Data is loaded from LUT and will be valid after a write is performed to program the read address. Returns zero in Host mode	RW	0x0
[15]	Reserved	Reserved	RO	
[14:0]	DCD_LUT_Read_Address	Corresponds to LUT entry to read. Number of entries is limited by CCG_HA_DCD_LUT_DEPTH_PARAM. On reads, returns the last address programmed in this field. Returns zero in Host mode	RW	0x0

### 8.3.4.40 por\_ccla\_snoop\_filter\_group\_id

Contains Snoop\_Filter\_Group\_ID

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xED0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

##### Usage constraints

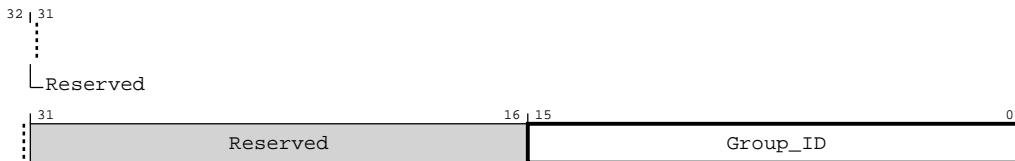
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to

this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-133: por\_ccla\_snoop\_filter\_group\_id**



**Table 8-134: por\_ccla\_snoop\_filter\_group\_id attributes**

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RO	
[15:0]	Group_ID	Uniquely identifies a snoop filter instance that is used to track CXL.cache devices below this Port. All Ports that share a single Snoop Filter instance shall set this field to the same value.	RW	0x0

### 8.3.4.41 por\_ccla\_snoop\_filter\_effective\_size

Contains Snoop\_Filter\_Effective\_Size

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xED8

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

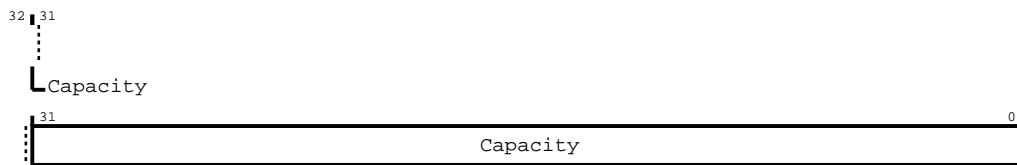
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-134: por\_ccla\_snoop\_filter\_effective\_size**



**Table 8-135: por\_ccla\_snoop\_filter\_effective\_size attributes**

Bits	Name	Description	Type	Reset
[31:0]	Capacity	Effective Snoop Filter Capacity representing the size of cache that can be effectively tracked by the Snoop Filter with this Group ID in multiples of 64K.	RW	0x0

### 8.3.4.42 por\_ccla\_dvsec\_cxl\_range\_1\_base\_high

Contains DVSEC\_CXL\_Range\_1\_Base\_High. Only applicable for Device. Host does not use this register

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xEE0

### Type

RWL

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

## Secure group override

por\_ccla\_scr.cxllink\_ctl

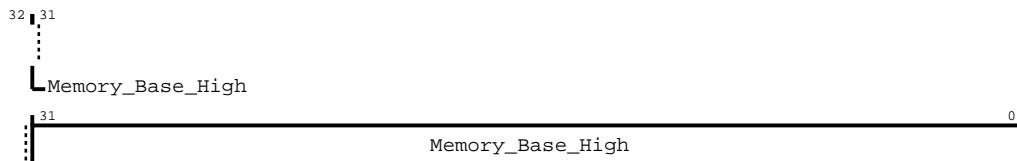
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-135: por\_ccla\_dvsec\_cxl\_range\_1\_base\_high**



**Table 8-136: por\_ccla\_dvsec\_cxl\_range\_1\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder O Base High Register for backward compatibility reasons.	RWL	0x0

### 8.3.4.43 por\_ccla\_dvsec\_cxl\_range\_1\_base\_low

Contains DVSEC\_CXL\_Range\_1\_Base\_Low. Only applicable for Device. Host does not use this register

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xEE8

### Type

RWL

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

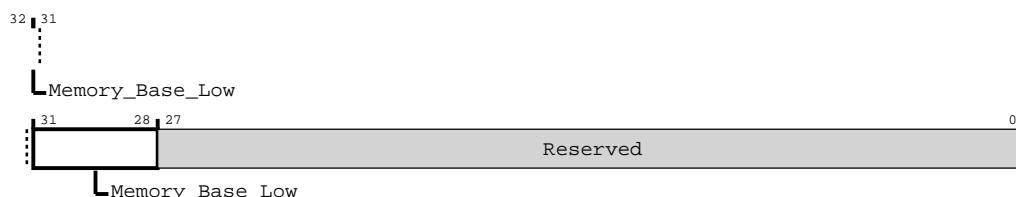
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-136: por\_ccla\_dvsec\_cxl\_range\_1\_base\_low**



**Table 8-137: por\_ccla\_dvsec\_cxl\_range\_1\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base Low Register for backward compatibility reasons.	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.4.44 por\_ccla\_dvsec\_cxl\_range\_2\_base\_high

Contains DVSEC\_CXL\_Range\_2\_Base\_High. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xEF0

### Type

RWL

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

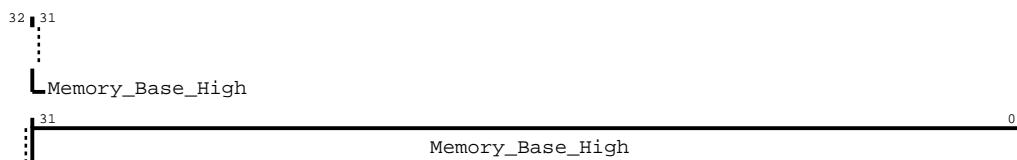
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-137: por\_ccla\_dvsec\_cxl\_range\_2\_base\_high**



**Table 8-138: por\_ccla\_dvsec\_cxl\_range\_2\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match the corresponding CXL HDM Decoder Base High Register for backward compatibility reasons.	RWL	0x0

### 8.3.4.45 por\_ccla\_dvsec\_cxl\_range\_2\_base\_low

Contains DVSEC\_CXL\_Range\_2\_Base\_Low. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xEF8

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

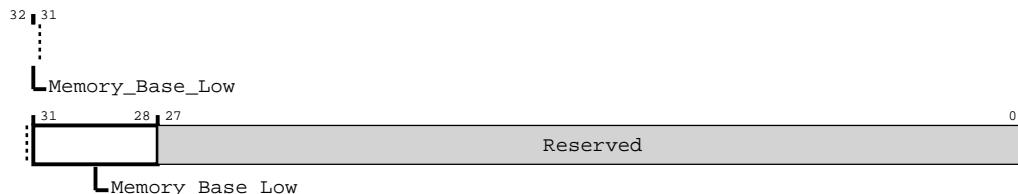
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-138: por\_ccla\_dvsec\_cxl\_range\_2\_base\_low**



**Table 8-139: por\_ccla\_dvsec\_cxl\_range\_2\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK	RWL	0x0

Bits	Name	Description	Type	Reset
[27:0]	Reserved	Reserved	RO	

### 8.3.4.46 por\_ccla\_dvsec\_cxl\_range\_1\_size\_high

Contains DVSEC\_CXL\_Range\_1\_Base\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1000

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

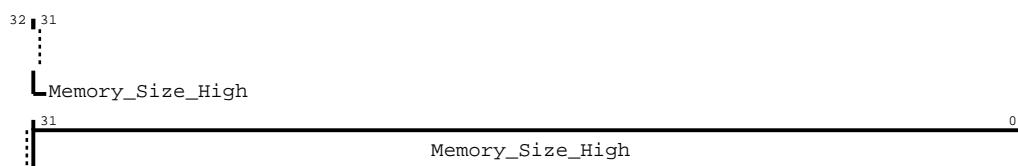
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-139: por\_ccla\_dvsec\_cxl\_range\_1\_size\_high**



**Table 8-140: por\_ccla\_dvsec\_cxl\_range\_1\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of CXL Range 1 memory size.	RW	0x0

### 8.3.4.47 por\_ccla\_dvsec\_cxl\_range\_1\_size\_low

Contains DVSEC\_CXL\_Range\_1\_Base\_Low. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1008

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

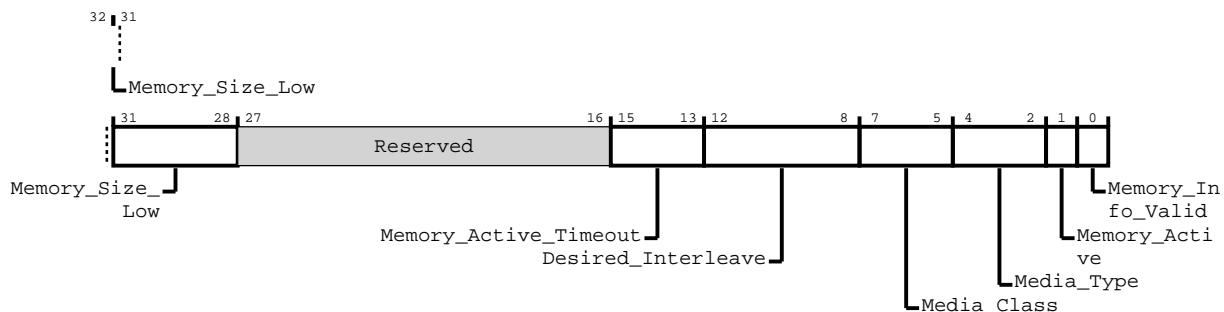
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-140: por\_ccla\_dvsec\_cxl\_range\_1\_size\_low**



**Table 8-141: por\_ccla\_dvsec\_cxl\_range\_1\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of CXL Range 1 memory size.	RW	0x0
[27:16]	Reserved	Reserved	RO	
[15:13]	Memory_Active_Timeout	All other - Reserved	RW	0x1
[12:8]	Desired_Interleave	all other settings are reserved.	RW	0x3
[7:5]	Media_Class	CXL 2.0 and future CXL.mem devices shall set this field to 010.	RW	0x2
[4:2]	Media_Type	other encodings are reserved. CXL 2.0 and future CXL.mem devices shall set this field to 010.	RW	0x2
[1]	Memory_Active	When set, indicates that the CXL Range 1 memory is fully initialized and available for software use. Must be set within Range 1. Memory_Active_Timeout of deassertion of reset to CXL device if CXL.mem HwInit Mode=1.	RW	0x0
[0]	Memory_Info_Valid	When set, indicates that the CXL Range 1 Size high and Size Low registers are valid. Must be set within 1 second of deassertion of reset to CXL device.	RW	0x0

### 8.3.4.48 por\_ccla\_dvsec\_cxl\_range\_2\_size\_high

Contains DVSEC\_CXL\_Range\_2\_Base\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1010

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

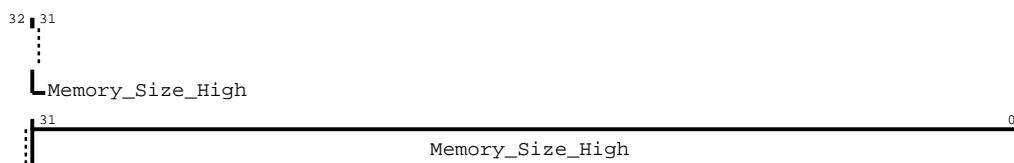
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-141: por\_ccla\_dvsec\_cxl\_range\_2\_size\_high**



**Table 8-142: por\_ccla\_dvsec\_cxl\_range\_2\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of CXL Range 2 memory size.	RW	0x0

### 8.3.4.49 por\_ccla\_dvsec\_cxl\_range\_2\_size\_low

Contains DVSEC\_CXL\_Range\_2\_Base\_Low. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x1018

#### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

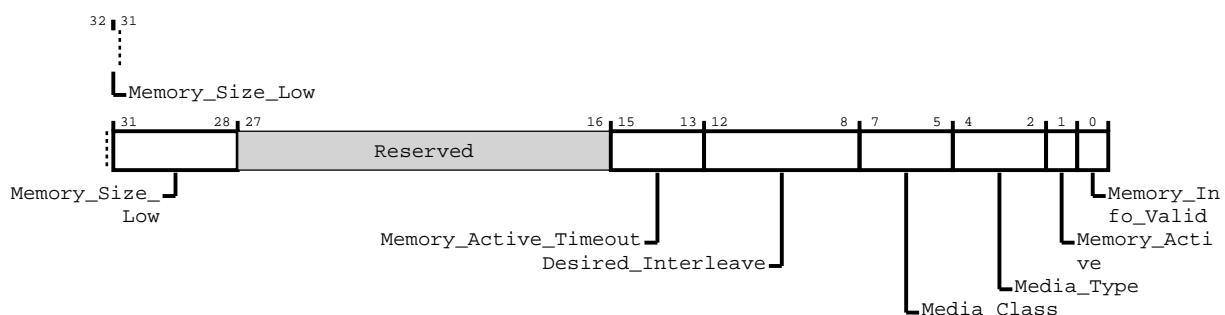
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-142: por\_ccla\_dvsec\_cxl\_range\_2\_size\_low**



**Table 8-143: por\_ccla\_dvsec\_cxl\_range\_2\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of CXL range 2 memory size.	RW	0x0
[27:16]	Reserved	Reserved	RO	
[15:13]	Memory_Active_Timeout	All other - Reserved	RW	0x1
[12:8]	Desired_Interleave	all other settings are reserved.	RW	0x3
[7:5]	Media_Class	CXL 2.0 and future CXL.mem devices shall set this field to 010.	RW	0x2
[4:2]	Media_Type	Other encodings are reserved. CXL 2.0 and future CXL.mem devices shall set this field to 010.	RW	0x2
[1]	Memory_Active	When set, indicates that the CXL range 2 memory is fully initialized and available for software use. Must be set within range 2. Memory_Active_Timeout of deassertion of reset to CXL device if CXL.mem HwInit Mode=1.	RW	0x0
[0]	Memory_Info_Valid	When set, indicates that the CXL range 2 Size high and Size Low registers are valid. Must be set within 1 second of deassertion of reset to CXL device.	RW	0x0

### 8.3.4.50 por\_ccla\_dvsec\_cxl\_control

Contains DVSEC\_CXL\_Control. Only applicable for Device. Host does not use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0xF00

##### Type

RWL

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

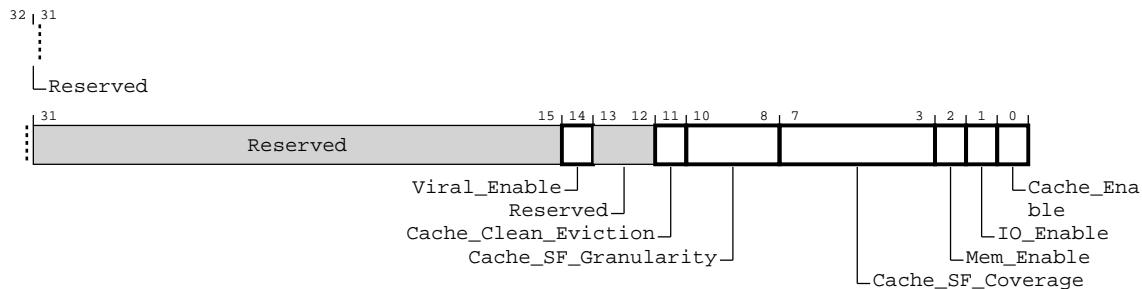
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-143: por\_ccla\_dvsec\_cxl\_control**



**Table 8-144: por\_ccla\_dvsec\_cxl\_control attributes**

Bits	Name	Description	Type	Reset
[31:15]	Reserved	Reserved	RO	
[14]	Viral_Enable	When set enables Viral handling in the CXL device. Locked by CONFIG_LOCK. If 0 the CXL device may ignore the viral that it receives	RWL	0x0
[13:12]	Reserved	Reserved	RO	
[11]	Cache_Clean_Eviction	<p>Performance hint to the device. Locked by CONFIG_LOCK.</p> <p><b>0</b> Indicates clean evictions from device caches are needed for best performance</p> <p><b>1</b> Indicates clean evictions from device caches are NOT needed for best performance</p>	RWL	0x0
[10:8]	Cache_SF_Granularity	<p>Performance hint to the device. Locked by CONFIG_LOCK.</p> <p><b>000</b> Indicates 64B granular tracking on the Host</p> <p><b>001</b> Indicates 128B granular tracking on the Host</p> <p><b>010</b> Indicates 256B granular tracking on the Host</p> <p><b>011</b> Indicates 512B granular tracking on the Host</p> <p><b>100</b> Indicates 1KB granular tracking on the Host</p> <p><b>101</b> Indicates 2KB granular tracking on the Host</p> <p><b>110</b> Indicates 4KB granular tracking on the Host</p> <p><b>111</b> Reserved</p>	RWL	0x0
[7:3]	Cache_SF_Coverage	<p>Performance hint to the device. Locked by CONFIG_LOCK.</p> <p><b>0x00</b> Indicates no Snoop Filter coverage on the Host</p> <p><b>For all other values of N</b> Indicates Snoop Filter coverage on the Host of <math>2^{(N+15d)}</math> Bytes.</p>	RWL	0x0
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	0x0
[1]	IO_Enable	When set enables CXL.io protocol operation when in Flex Bus.CXL mode.	RWL	0x1
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	0x0

### 8.3.4.51 por\_ccla\_dvsec\_cxl\_control2

Contains DVSEC\_CXL\_Control2. Only applicable for Device. Host does not use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0xF08

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

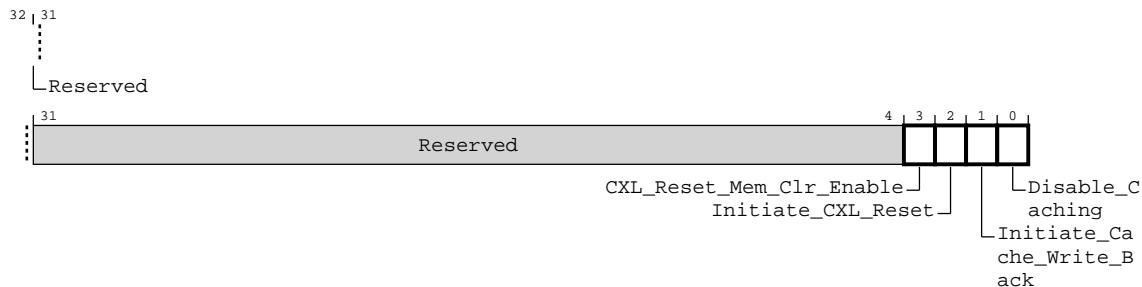
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-144: por\_ccla\_dvsec\_cxl\_control2**



**Table 8-145: por\_ccla\_dvsec\_cxl\_control2 attributes**

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	RO	
[3]	CXL_Reset_Mem_Clr_Enable	When set and CXL Reset Mem Clr Capable returns 1 Device shall clear or randomize volatile HDM ranges as part of the CXL Reset operation. When CXL Reset Mem Clr Capable is clear this bit is ignored and volatile HDM ranges may or may not be cleared or randomized during CXL Reset.	RW	0x0
[2]	Initiate_CXL_Reset	When set to 1 device shall initiate CXL Reset as defined in Section 9.7. This bit always returns the value of 0 when read by the software. A write of 0 is ignored.	RW	0x0
[1]	Initiate_Cache_Write_Back	When set to 1 device shall write back all modified lines in the local cache and invalidate all lines. The device shall send CacheFlushed message to host as required by CXL.Cache protocol to indicate it does not hold any modified lines.	RW	0x0
[0]	Disable_Caching	When set to 1 device shall no longer cache new modified lines in its local cache. Device shall continue to correctly respond to CXL.cache transactions.	RW	0x0

### 8.3.4.52 por\_ccla\_dvsec\_cxl\_lock

Contains DVSEC\_CXL\_Lock. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xF10

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

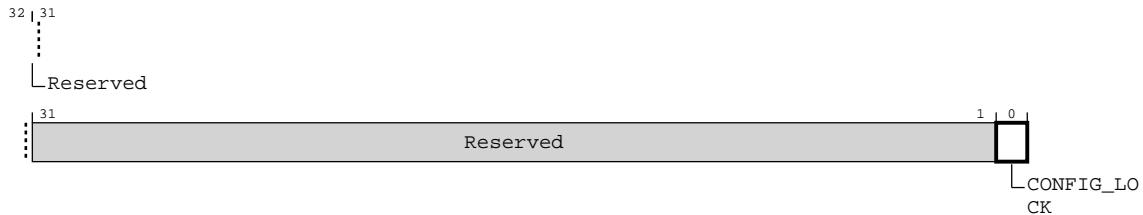
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-145: por\_ccla\_dvsec\_cxl\_lock**



**Table 8-146: por\_ccla\_dvsec\_cxl\_lock attributes**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	
[0]	CONFIG_LOCK	When set all register fields in the PCIe DVSEC for CXL Devices Capability with RWL attribute become read only. Consult individual register fields for details. This bit is cleared upon device Conventional Reset. This bit and all the fields that are locked by this bit are not affected by CXL Reset.	RW	0x0

### 8.3.4.53 por\_ccla\_ras\_err\_en

When enabled, RAS errors can be sent over CXL. Only applicable to device. Host doesn't use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0x1F90

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

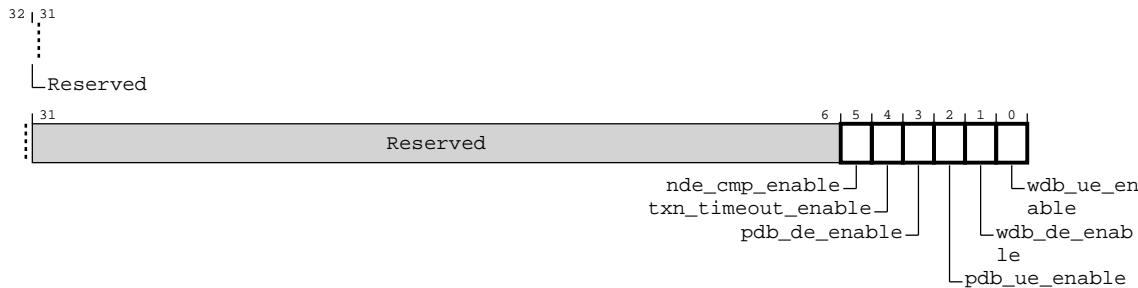
Non-Confidential

this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-146: por\_ccla\_ras\_err\_en**



**Table 8-147: por\_ccla\_ras\_err\_en attributes**

Bits	Name	Description	Type	Reset
[31:6]	Reserved	Reserved	RO	
[5]	nde_cmp_enable	If set, report non data error over CXL	RW	0x0
[4]	txn_timeout_enable	If set, report transaction timeout error over CXL	RW	0x0
[3]	pdb_de_enable	If set, report passive data buffer deferred error over CXL	RW	0x0
[2]	pdb_ue_enable	If set, report passive data buffer uncorrectable error over CXL	RW	0x0
[1]	wdb_de_enable	If set, report write data buffer deferred error over CXL	RW	0x0
[0]	wdb_ue_enable	If set, report write data buffer uncorrectable error over CXL	RW	0x0

### 8.3.4.54 por\_ccla\_ras\_viral\_en

When enabled, viral is sent over cxl

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0x1F98

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

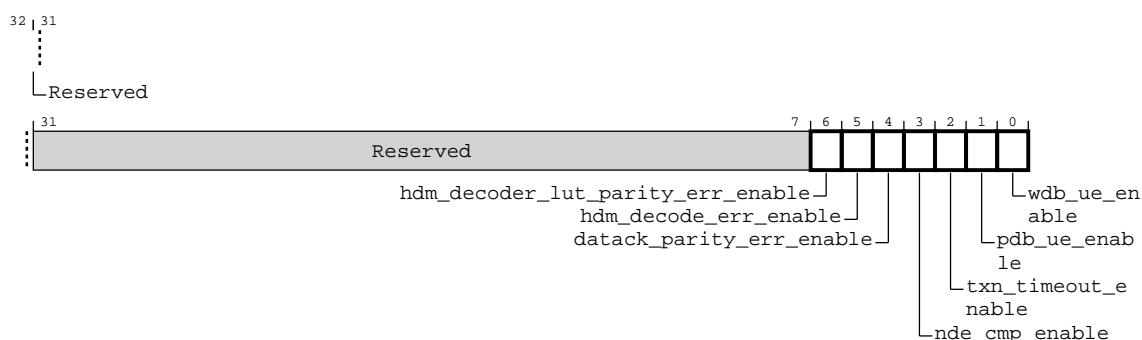
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-147: por\_ccla\_ras\_viral\_en**



**Table 8-148: por\_ccla\_ras\_viral\_en attributes**

Bits	Name	Description	Type	Reset
[31:7]	Reserved	Reserved	RO	
[6]	hdm_decoder_lut_parity_err_enable	If set, report hdm decoder lut parity error over CXL	RW	0x0
[5]	hdm_decode_err_enable	If set, report hdm decode error over CXL	RW	0x0
[4]	datack_parity_err_enable	If set, report data check parity error over CXL	RW	0x0
[3]	nde_cmp_enable	If set, report non data error over CXL	RW	0x0
[2]	txn_timeout_enable	If set, report transaction timeout error over CXL	RW	0x0
[1]	pdb_ue_enable	If set, report passive data buffer uncorrectable error over CXL	RW	0x0
[0]	wdb_ue_enable	If set, report write data buffer uncorrectable error over CXL	RW	0x0

#### 8.3.4.55 por\_ccla\_dvsec\_flex\_bus\_port\_control

Contains DVSEC\_Flex\_Bus\_Port\_Control

# Configurations

This register is available in all configurations.

## Attributes

## Width

32

## Address offset

0xF18

## Type

RW

## Reset value

See individual bit resets

## Root group override

por ccla rcr,cxllink ctl

## Secure group override

por ccla scr.cxllink ctl

## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-148:** por\_ccla\_dvsec\_flex\_bus\_port\_control

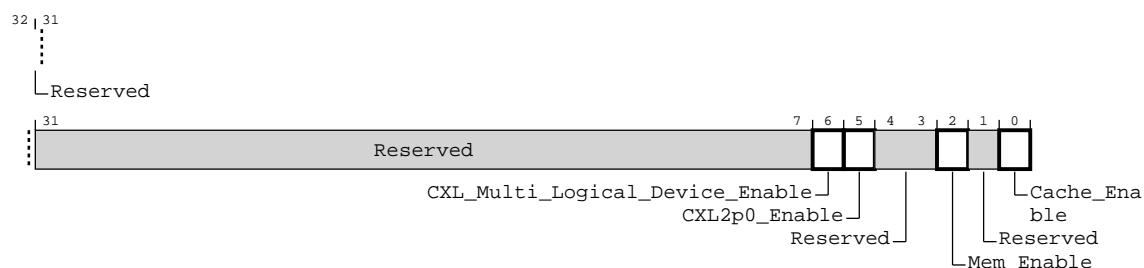


Table 8-149: por\_ccla\_dvsec\_flex\_bus\_port\_control attributes

Bits	Name	Description	Type	Reset
[31:7]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[6]	CXL_Multi_Logical_Device_Enable	When set enable Multi-Logical Device operation when in Flex Bus.CXL mode	RW	0x0
[5]	CXL2p0_Enable	When set enable CXL2.0 protocol operation when in Flex Bus.CXL mode.	RW	0x0
[4:3]	Reserved	Reserved	RO	
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode.	RW	0x0
[1]	Reserved	Reserved	RO	
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode.	RW	0x0

### 8.3.4.56 por\_ccla\_err\_capabilities\_control

Contains err\_capabilities\_control. Only applicable to device. Host doesn't use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0xF40

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

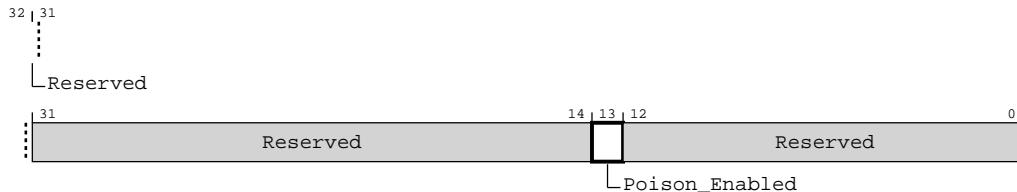
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-149: por\_ccla\_err\_capabilities\_control**



**Table 8-150: por\_ccla\_err\_capabilities\_control attributes**

Bits	Name	Description	Type	Reset
[31:14]	Reserved	Reserved	RO	
[13]	Poison_Enabled	If this bit is 0 CXL 1.1 Upstream Ports CXL 1.1 Downstream Ports and CXL 2.0 Root Port shall treat poison received on CXL.cache or CXL.mem as uncorrectable error and log the error in Uncorrectable Error Status Register. If this bit is 1 these ports shall treat poison received on CXL.cache or CXL.mem as correctable error and log the error in Correctable Error Status Register. This bit defaults to 1. This bit is hardwired to 1 in CXL 2.0 Upstream Switch Port CXL 2.0 Downstream Switch Port and CXL 2.0 device.	RW	0x0
[12:0]	Reserved	Reserved	RO	

### 8.3.4.57 por\_ccla\_IDE\_key\_refresh\_time\_control

Contains IDE\_key\_refresh\_time\_control. Not applicable to CMN 700

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xF58

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

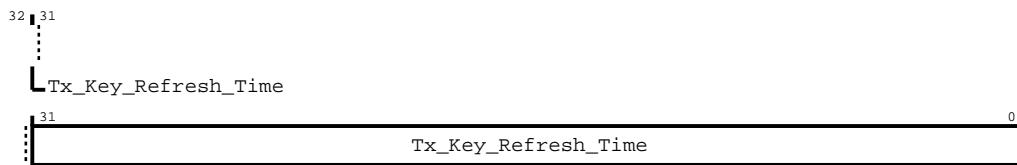
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-150: por\_ccla\_IDE\_key\_refresh\_time\_control**



**Table 8-151: por\_ccla\_IDE\_key\_refresh\_time\_control attributes**

Bits	Name	Description	Type	Reset
[31:0]	Tx_Key_Refesh_Time	Minimum number of flits transmitter needs to block transmission of protocol flits after IDE.Start has sent. Used when switching keys.	RW	0x0

### 8.3.4.58 por\_ccla\_IDE\_truncation\_transmit\_delay\_control

Contains IDE\_truncation\_transmit\_delay\_control. Not applicable to CMN 700

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xF60

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

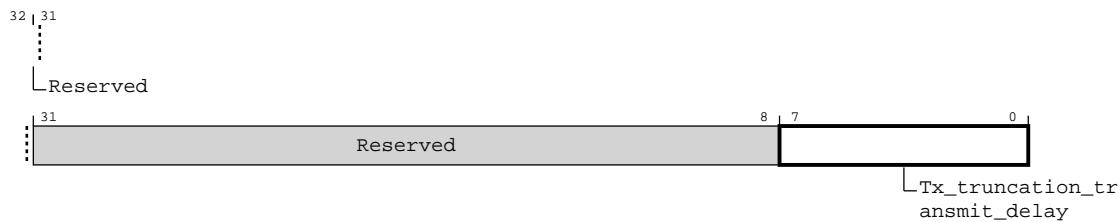
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-151: por\_ccla\_IDE\_truncation\_transmit\_delay\_control**



**Table 8-152: por\_ccla\_IDE\_truncation\_transmit\_delay\_control attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:0]	Tx_truncation_transmit_delay	This parameter feeds into the computation of minimum number of IDE idle flits Transmitter needs send after sending a truncated MAC flit.	RW	0x0

## 8.3.4.59 por\_ccla\_IDE\_outbound\_state

Contains IDE outbound state machine status

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xF90

#### Type

RO

#### Reset value

See individual bit resets

#### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

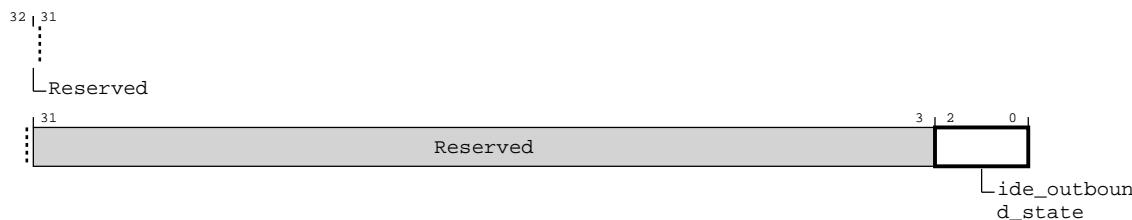
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-152: por\_ccla\_IDE\_outbound\_state**



**Table 8-153: por\_ccla\_IDE\_outbound\_state attributes**

Bits	Name	Description	Type	Reset
[31:3]	Reserved	Reserved	RO	
[2:0]	ide_outbound_state	Represents status of the outbound state machine in ULL.	RO	0x0

### 8.3.4.60 por\_ccla\_IDE\_inbound\_state

Contains IDE inbound state machine status

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xF98

#### Type

RO

#### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

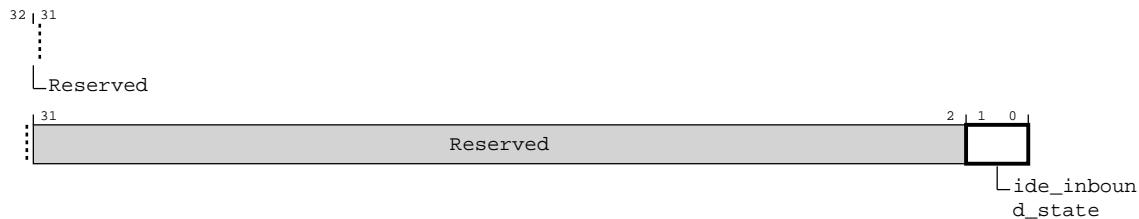
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-153: por\_ccla\_IDE\_inbound\_state**



**Table 8-154: por\_ccla\_IDE\_inbound\_state attributes**

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Reserved	RO	
[1:0]	ide_inbound_state	Represents status of the inbound state machine in ULL.	RO	0x0

### 8.3.4.61 por\_ccla\_ll\_to\_ull\_msg

Contains ll\_to\_ull\_message

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xF70

#### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

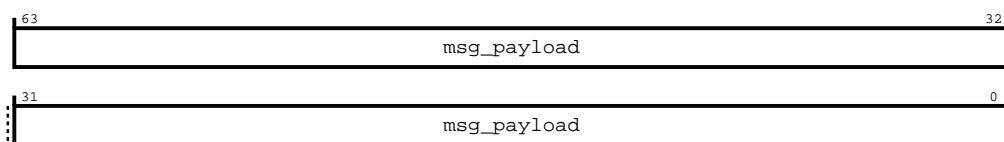
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-154: por\_ccla\_ll\_to\_ull\_msg**



**Table 8-155: por\_ccla\_ll\_to\_ull\_msg attributes**

Bits	Name	Description	Type	Reset
[63:0]	msg_payload	Contains 64 bits of message sent from ll to ull	RW	0x0

### 8.3.4.62 por\_ccla\_cxl\_timeout\_iso\_ctl

Contains cxl\_timeout\_iso\_ctl.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xF80

#### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

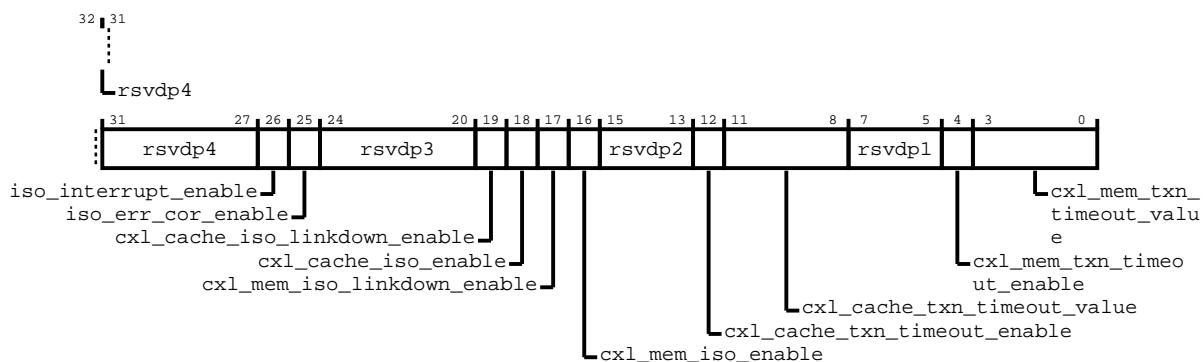
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-155: por\_ccla\_cxl\_timeout\_iso\_ctl**



**Table 8-156: por\_ccla\_cxl\_timeout\_iso\_ctl attributes**

Bits	Name	Description	Type	Reset
[31:27]	rsvdp4	Reserved	RW	0x0
[26]	iso_interrupt_enable	When Set this bit enables the generation of an interrupt to indicate that isolation has been triggered.	RW	0x0
[25]	iso_err_cor_enable	When Set this bit enables the sending of an ERR_COR Message to indicate isolation has been triggered.	RW	0x0
[24:20]	rsvdp3	Reserved	RW	0x0
[19]	cxl_cache_iso_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.cache enters isolation mode.	RW	0x0
[18]	cxl_cache_iso_enable	This field allows System Software to enable CXL.cache isolation actions.	RW	0x0
[17]	cxl_mem_iso_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.mem enters isolation mode.	RW	0x0
[16]	cxl_mem_iso_enable	This field allows System Software to enable CXL.mem isolation actions. If this field is set isolation actions will be triggered if either a CXL.mem Transaction Timeout is detected or if the CXL link went down.	RW	0x0

Bits	Name	Description	Type	Reset
[15:13]	rsvdp2	Reserved	RW	0x0
[12]	cxl_cache_txn_timeout_enable	-	RW	0x0
[11:8]	cxl_cache_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.cache.	RW	0x0
[7:5]	rsvdp1	Reserved	RW	0x0
[4]	cxl_mem_txn_timeout_enable	When Set this bit enables CXL.mem Transaction Timeout mechanism.	RW	0x0
[3:0]	cxl_mem_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.mem.	RW	0x0

### 8.3.4.63 por\_ccla\_cxl\_timeout\_iso\_capability

Contains cxl\_timeout\_iso\_capability.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xF88

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

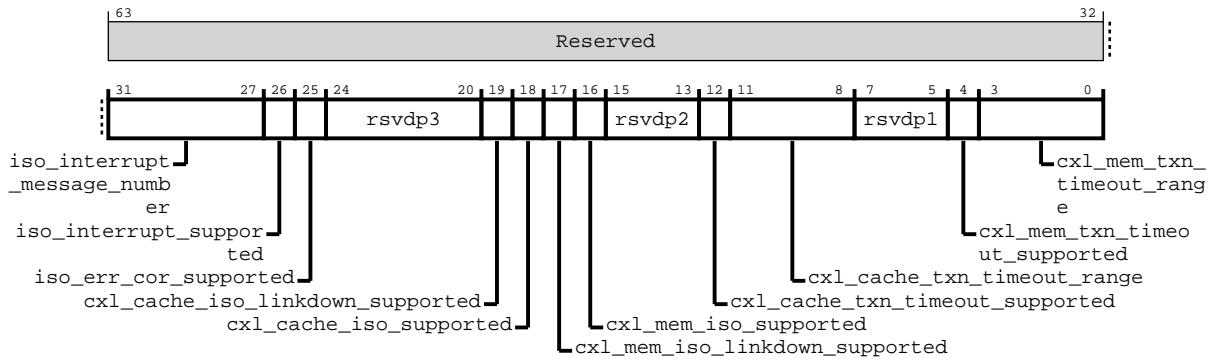
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-156: por\_ccla\_cxl\_timeout\_iso\_capability**



**Table 8-157: por\_ccla\_cxl\_timeout\_iso\_capability attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:27]	iso_interrupt_message_number	This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with the CXL Timeout and isolation Capability structure.	RW	0x0
[26]	iso_interrupt_supported	This field indicates support for signaling interrupt when isolation is triggered.	RW	0x1
[25]	iso_err_cor_supported	If Set, this bit indicates that the Root Port supports the ability to signal with ERR_COR when isolation is triggered.	RW	0x1
[24:20]	rsvdp3	Reserved	RW	0x0
[19]	cxl_cache_iso_linkdown_supported	This field indicates support for triggering Link-Down on the CXL Root Port if CXL.cache enters isolation mode.	RW	0x0
[18]	cxl_cache_iso_supported	This field indicates support for isolation on CXL.cache.	RW	0x0
[17]	cxl_mem_iso_linkdown_supported	This field indicates support for triggering Link-Down on the CXL port if CXL.mem enters isolation mode.	RW	0x1
[16]	cxl_mem_iso_supported	This field indicates support for isolation on CXL.mem.	RW	0x1
[15:13]	rsvdp2	Reserved	RW	0x0
[12]	cxl_cache_txn_timeout_supported	-	RW	0x0
[11:8]	cxl_cache_txn_timeout_range	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Range for CXL.cache.	RW	0x1
[7:5]	rsvdp1	Reserved	RW	0x0
[4]	cxl_mem_txn_timeout_supported	When Set indicates support for CXL.mem Transaction Timeout mechanism.	RW	0x1
[3:0]	cxl_mem_txn_timeout_range	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout ranges for CXL.mem.	RW	0x1

### 8.3.4.64 por\_ccla\_cxl\_timeout\_iso\_status

Contains cxl\_timeout\_iso\_status

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xF78

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

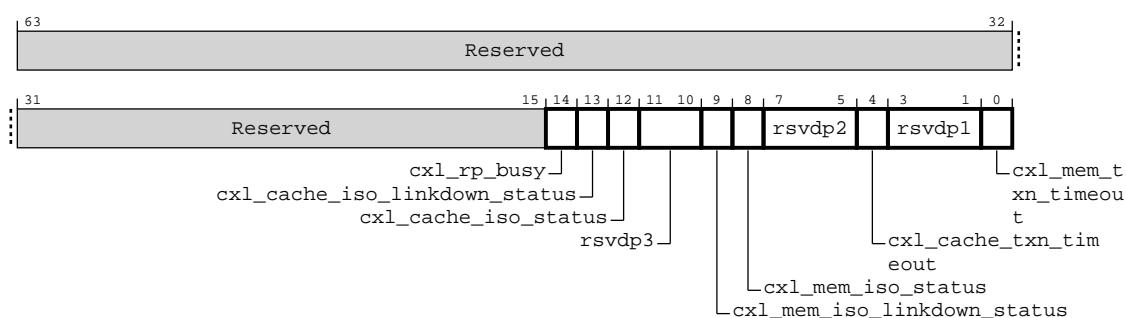
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-157: por\_ccla\_cxl\_timeout\_iso\_status**



**Table 8-158: por\_ccla\_cxl\_timeout\_iso\_status attributes**

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	
[14]	cxl_rp_busy	When either the CXL.mem isolation Status bit or the CXL.cache isolation Status bit is Set and this bit is Set, the Root Port is busy with internal activity that must complete before software is permitted to Clear the CXL.mem isolation Status bit and the CXL.cache isolation Status bit.	RW	0x0
[13]	cxl_cache_iso_linkdown_status	This field indicates that isolation Mode for CXL.cache was triggered due to link down.	RW	0x0
[12]	cxl_cache_iso_status	This field indicates that isolation mode for CXL.cache was triggered.	RW	0x0
[11:10]	rsvdp3	Reserved	RW	0x0
[9]	cxl_mem_iso_linkdown_status	This field indicates that isolation Mode for CXL.mem was triggered due to link down	RW	0x0
[8]	cxl_mem_iso_status	This field indicates that isolation mode for CXL.mem was triggered.	RW	0x0
[7:5]	rsvdp2	Reserved	RW	0x0
[4]	cxl_cache_txn_timeout	When Set, this indicates that a CXL.cache transaction timed out.	RW	0x0
[3:1]	rsvdp1	Reserved	RW	0x0
[0]	cxl_mem_txn_timeout	When Set, this indicates that a CXL.mem transaction timed out.	RW	0x0

### 8.3.4.65 por\_ccla\_timeout\_base\_value\_cycles

Contains number of cycles that represent 25us at a given design frequency

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1020

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

#### Usage constraints

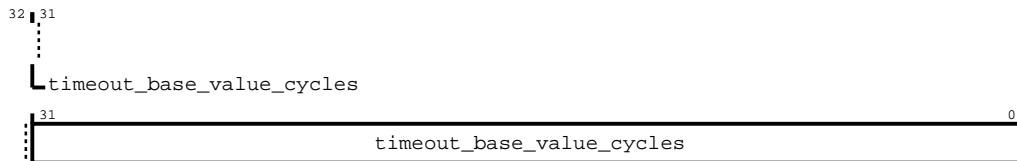
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to

this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-158: por\_ccla\_timeout\_base\_value\_cycles**



**Table 8-159: por\_ccla\_timeout\_base\_value\_cycles attributes**

Bits	Name	Description	Type	Reset
[31:0]	timeout_base_value_cycles	Contains number of cycles that represent 25us at a given design frequency.	RW	0xC350

### 8.3.4.66 por\_ccla\_CXL\_Cache\_ID\_Decoder\_Capability

Contains CXL\_Cache\_ID\_Decoder\_Capability

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xE58

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

### Usage constraints

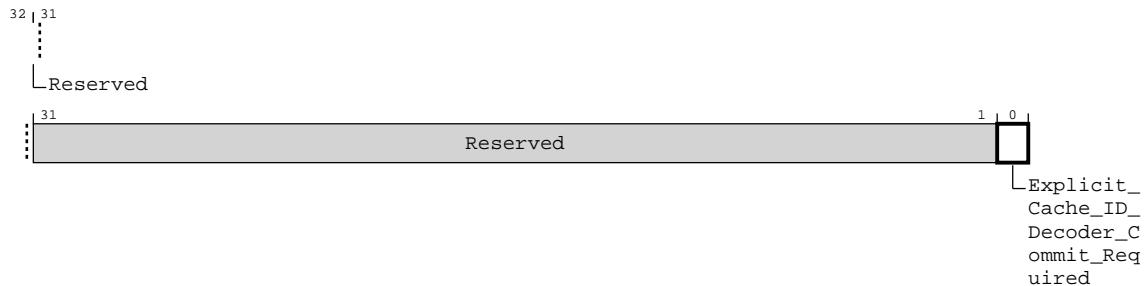
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to

this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-159: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Capability**



**Table 8-160: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Capability attributes**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	
[0]	Explicit_Cache_ID_Decoder_Commit_Required	If 1, indicates that the software must set Cache ID Decoder Commit bit anytime a new CXL.cache device is enabled anywhere below this port. If 1, the Cache ID Decoder Commit bit, the Cache ID Decoder Committed bit, the Cache ID Decoder Commit timeout Scale field, the Cache ID Decoder Commit Timeout Base field, and Cache ID Decoder Error Not Committed bit are implemented. Cache ID Decoder Commit operation may be used by a component to update its internal structures or perform consistency checks.	RW	0x0

### 8.3.4.67 por\_ccla\_CXL\_Cache\_ID\_Decoder\_Control

Contains CXL\_Cache\_ID\_Decoder\_Control

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xE60

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxllink\_ctl

### Secure group override

por\_ccla\_scr.cxllink\_ctl

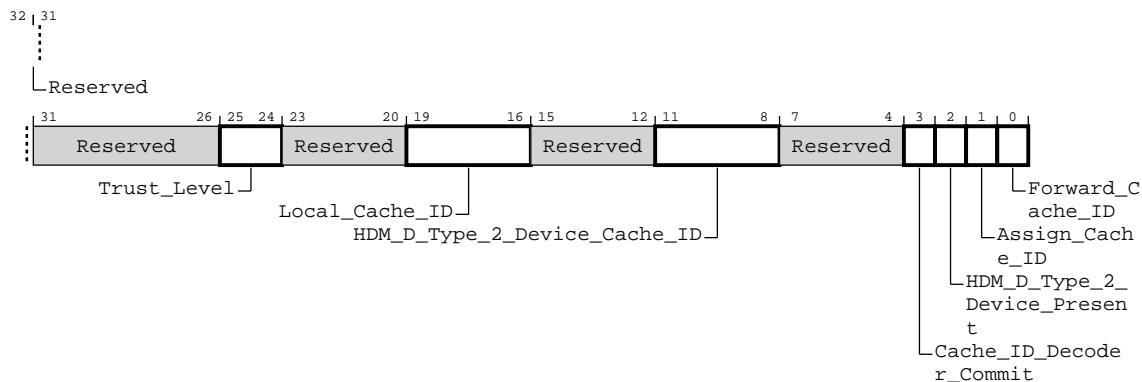
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-160: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Control**



**Table 8-161: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Control attributes**

Bits	Name	Description	Type	Reset
[31:26]	Reserved	Reserved	RO	
[25:24]	Trust_Level	Trust Level assigned to the directly connected device when Assign Cache ID=1. 00b = See Table 8-26 01b = Reserved 10b = See Table 8-26 11b = Reserved The reset default is 10b.	RW	0x0
[23:20]	Reserved	Reserved	RO	
[19:16]	Local_Cache_ID	If Assign Cache ID Enable=1, the Port assigns this Cache ID to the directly connected CXL.cache device regardless of whether it is using HDM-D flows or HDM-DB flows. The reset default is 0h.	RW	0x0
[15:12]	Reserved	Reserved	RO	
[11:8]	HDM_D_Type_2_Device_Cache_ID	If HDM-D Type 2 Device Present=1, this field represents the Cache ID that has been assigned to the Type 2 device below this Downstream Port that is using HDM-D flows. This field may be used by the port to identify a Type 2 device that is using HDM-D flows and must not be used for assigning Cache ID. The reset default is 0h.	RW	0x0

Bits	Name	Description	Type	Reset
[7:4]	Reserved	Reserved	RO	
[3]	Cache_ID_Decoder_Commit	If Explicit Cache ID Decoder Commit Required=1. software must cause this bit to transition from 0 to 1 to commit the Cache ID assignment change to this Cache ID Decoder instance. The default value of this field is 0. This bit must be RW if the Explicit Cache ID Decoder Commit Required bit is set; otherwise, it is permitted to be hardwired to 0 and the Cache ID Decoder update does not require an explicit commit. Software must not set this bit unless the Explicit Cache ID Decoder Commit Required bit is set.	RW	0x0
[2]	HDM_D_Type_2_Device_Present	1 indicates there is a Type 2 Device below this Downstream Port that is using HDM-D flows. The reset default is 0.	RW	0x0
[1]	Assign_Cache_ID	1 indicates this Downstream Port is connected directly to a CXL.cache Device and assigns a cache ID=Local Cache ID to it. The reset default is 0.	RW	0x0
[0]	Forward_Cache_ID	1 indicates the Port forwards CXL.cache messages in both directions. The reset default is 0.	RW	0x0

### 8.3.4.68 por\_ccla\_CXL\_Cache\_ID\_Decoder\_Status

Contains CXL\_Cache\_ID\_Decoder\_Status

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xE68

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

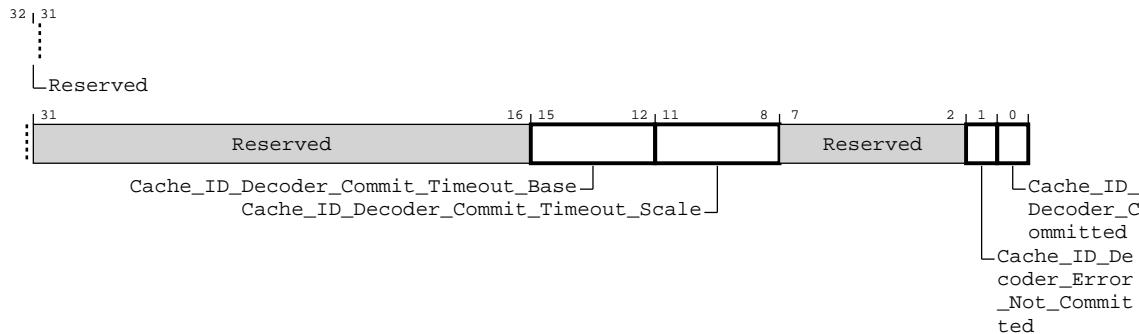
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-161: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Status**



**Table 8-162: por\_ccla\_CXL\_Cache\_ID\_Decoder\_Status attributes**

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RO	
[15:12]	Cache_ID_Decoder_Commit_Timeout_Base	This field determines the Cache ID Decoder Commit timeout. The timeout duration is calculated by multiplying the Timeout Base with the Timeout Scale. Failure to set either the Cache ID Decoder Committed bit or the Cache ID Decoder Error Not Committed bit within the timeout value is treated as equivalent to commit error. In case of a timeout, the software must clear the Cache ID Decoder Commit bit to 0 prior to setting it to 1 again. This field is reserved if Explicit Cache ID Decoder Commit Required=0.	RW	0x0
[11:8]	Cache_ID_Decoder_Commit_Timeout_Scale	This field specifies the time scale associated with Cache ID Decoder Commit Timeout. 0000b = 1 us 0001b = 10 us 0010b = 100 us 0011b = 1 ms 0100b = 10 ms 0101b = 100 ms 0110b = 1 second 0111b = 10 seconds All other encodings are reserved This field is reserved if Explicit Cache ID Decoder Commit Required=0.	RW	0x0
[7:2]	Reserved	Reserved	RO	
[1]	Cache_ID_Decoder_Error_Not_Committed	When set to 1, it indicates that the last write that caused the Cache ID Decoder Commit bit to transition from 0 to 1 was processed by the component, but resulted in an error. This bit is cleared when the software causes the Cache ID Decoder Commit bit to transition from 1 to 0. This bit is reserved if Explicit Cache ID Decoder Commit Required=0.	RW	0x0
[0]	Cache_ID_Decoder_Committed	When set to 1, it indicates that the last write that caused the Cache ID Decoder Commit bit to transition from 0 to 1 was successfully processed by the component. This bit is cleared when the software causes the Cache ID Decoder Commit bit to transition from 1 to 0. This bit is reserved if Explicit Cache ID Decoder Commit Required=0.	RW	0x0

### 8.3.4.69 por\_ccla\_root\_port\_n\_security\_policy

Contains Root\_Port\_n\_Security\_Policy\_Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xF28

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

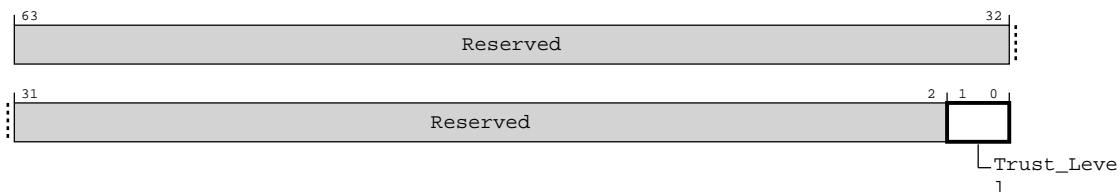
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-162: por\_ccla\_root\_port\_n\_security\_policy**



**Table 8-163: por\_ccla\_root\_port\_n\_security\_policy attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	Trust_Level	Trust Level for the CXL.cache Device below Root Port n	RW	0x2

### 8.3.4.70 por\_ccla\_root\_port\_n\_id

Contains Root\_Port\_n\_ID\_Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xF30

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

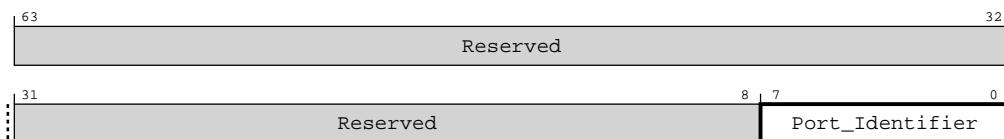
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-163: por\_ccla\_root\_port\_n\_id**



**Table 8-164: por\_ccla\_root\_port\_n\_id attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:0]	Port_Identifier	Port Identifier of the Root Port n	RW	0x0

### 8.3.4.71 por\_ccla\_cxl\_link\_layer\_defeature

CXL Link Layer Defeature Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE18

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxllink\_ctl

##### Secure group override

por\_ccla\_scr.cxllink\_ctl

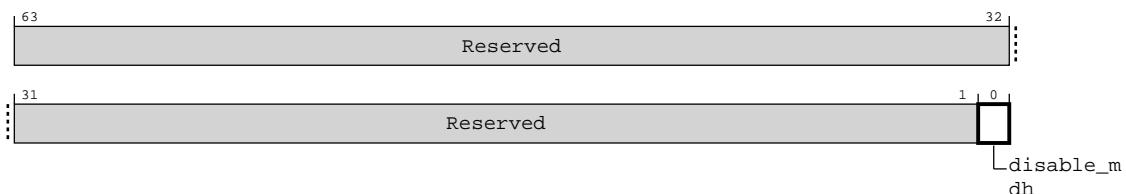
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxllink\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxllink\_ctl bit and por\_ccla\_rcr.cxllink\_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.

**Figure 8-164: por\_ccla\_cxl\_link\_layer\_defeature**



**Table 8-165: por\_ccla\_cxl\_link\_layer\_defeature attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	disable_mdh	Write 1 to disable MDH. Software needs to ensure it programs this value consistently on the UP & DP. After programming, a warm reset is required for the disable to take effect.	RW	0b0

### 8.3.4.72 por\_ccla ull\_ctl

Upper Link Layer Control Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE20

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ull\_ctl

##### Secure group override

por\_ccla\_scr.ull\_ctl

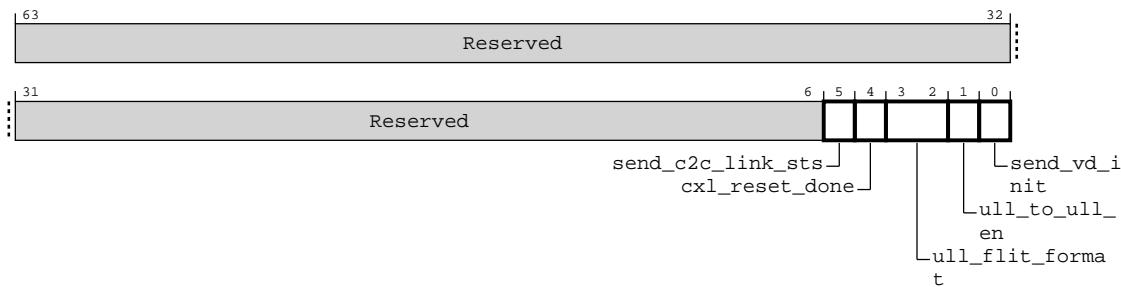
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ull\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ull\_ctl bit and por\_ccla\_rcr.ull\_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.

**Figure 8-165: por\_ccla\_ull\_ctl**



**Table 8-166: por\_ccla\_ull\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	
[5]	send_c2c_link_sts	send c2c link status linit	RW	0b0
[4]	cxl_reset_done	System software indication to communicate CXL reset done. HW de-asserts the CXL reset only after the de-assertion of rh_busy status and software cxl_reset_done handshake Read to this registers is always zero	RW	0b0
[3:2]	ull_flit_format	Used to enable 256 byte flits when in ULL-to_ULL mode. Format be set on both sides of the link on the same write as 'send_vd_init'  <b>0b00</b> 68B Legacy Flit format  <b>0b01</b> Reserved  <b>0b10</b> 128B Latency Optimized Flit format  <b>0b11</b> 256B Flit format	RW	0b00
[1]	ull_to_ull_en	Used to enable ULL-to_ULL mode. Must be set on both sides of the link before 'send_vd_init' is set on either side  <b>0b0</b> When clear, ull-to-ull mode is disabled.  <b>0b1</b> When set, ull-to-ull mode is enabled.	RW	0b0
[0]	send_vd_init	Used to send VD Init message. Must only be used for direct ULL to ULL connection. Must be used along with Tx ULL state (tx_ull_state) status bit  <b>0b0</b> When clear and tx_ull_state is in run_state, sends VD.De-activate flit.  <b>0b1</b> When set and tx_ull_state is in stop state, sends VD.Activate flit.	RW	0b0

### 8.3.4.73 por\_ccla\_ull\_status

Upper Link Layer Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE28

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ull\_ctl

##### Secure group override

por\_ccla\_scr.ull\_ctl

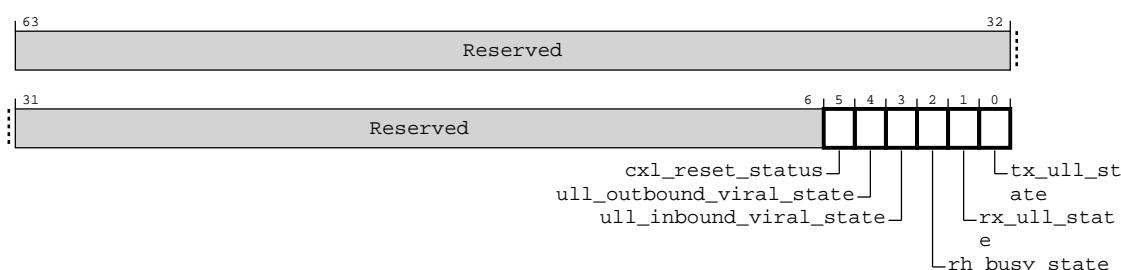
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ull\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ull\_ctl bit and por\_ccla\_rcr.ull\_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.

**Figure 8-166: por\_ccla\_ull\_status**



**Table 8-167: por\_ccla\_ull\_status attributes**

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5]	cxl_reset_status	Reflects CXL reset status.  <b>0b0</b> CXL reset de-asserted  <b>0b1</b> CXL reset asserted	RO	0b0
[4]	ull_outbound_viral_state	Indicates ULL is in Viral state originated in the ULL.  <b>0b0</b> ULL is not in Viral state  <b>0b1</b> ULL is in Viral state	RO	0b0
[3]	ull_inbound_viral_state	Indicates ULL is in Viral state from the Inbound CXS Flit.  <b>0b0</b> ULL is not in Viral state  <b>0b1</b> ULL is in Viral state	RO	0b0
[2]	rh_busy_state	Reflects the CCRH transaction pending state .  <b>0b0</b> RH (HA/RA) are in IDLE state  <b>0b1</b> RH (HA/RA) are in Busy state	RO	0b0
[1]	rx_ull_state	Reflects the Rx ULL state .  <b>0b0</b> Rx ULL is in Stop state  <b>0b1</b> Rx ULL is in Run state	RO	0b0
[0]	tx_ull_state	Reflects the Tx ULL state .  <b>0b0</b> Tx ULL is in Stop state  <b>0b1</b> Tx ULL is in Run state	RO	0b0

### 8.3.4.74 por\_ccla\_cxl\_ll\_errinject\_ctl

CXL .cache .mem Link Layer Error Injection Control Register. Not used in Host mode

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

### Address offset

0xE30

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cxlerrinj\_ctl

### Secure group override

por\_ccla\_scr.cxlerrinj\_ctl

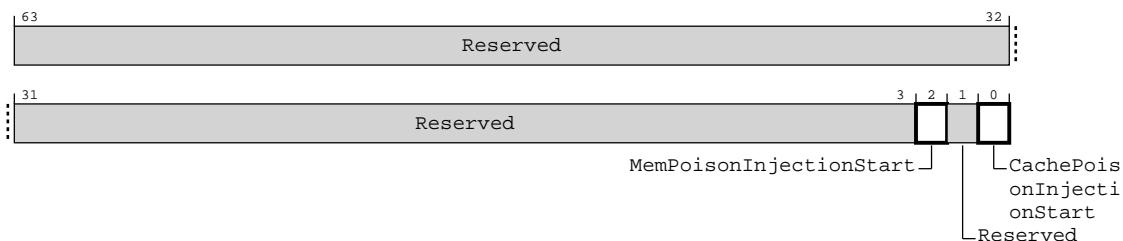
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxlerrinj\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxlerrinj\_ctl bit and por\_ccla\_rcr.cxlerrinj\_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.

**Figure 8-167: por\_ccla\_cxl\_ll\_errinject\_ctl**



**Table 8-168: por\_ccla\_cxl\_ll\_errinject\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	MemPoisonInjectionStart	Software writes 0x1 to this bit to trigger a single poison injection on a CXL.mem message in the Tx direction. Hardware must override the poison field in the data header slot of the corresponding message (DRS if device, RWD if Host). This bit is required only if CXL.mem protocol is supported.	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	CachePoisonInjectionStart	Software writes 0x1 to this bit to trigger a single poison injection on a CXL.cache message in the Tx direction. Hardware must override the poison field in the data header slot of the corresponding message (D2H if device, H2D if Host). This bit is required only if CXL.cache protocol is supported.	RW	0b0

### 8.3.4.75 por\_ccla\_cxl\_ll\_errinject\_stat

CXL .cache .mem Link Layer Error Injection Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE38

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cxlerrinj\_ctl

##### Secure group override

por\_ccla\_scr.cxlerrinj\_ctl

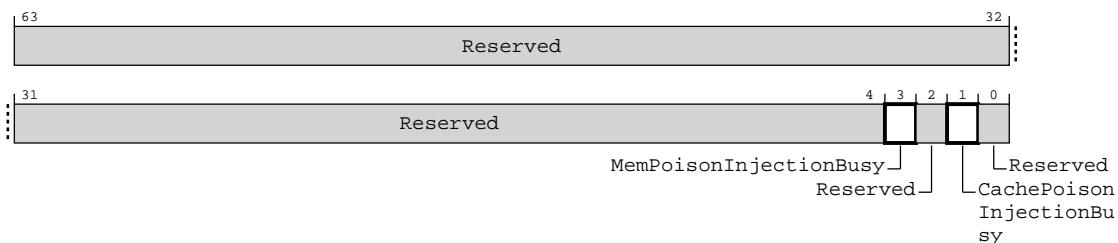
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cxlerrinj\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cxlerrinj\_ctl bit and por\_ccla\_rcr.cxlerrinj\_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.

**Figure 8-168: por\_ccla\_cxl\_ll\_errinject\_stat**



**Table 8-169: por\_ccla\_cxl\_ll\_errinject\_stat attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	MemPoisonInjectionBusy	Hardware loads 1 to this bit when the Start bit is written. Hardware must clear this bit to indicate that it has indeed finished poisoning a packet. Software is permitted to poll on this bit to find out when hardware has finished poison injection. This bit is required only if CXL.mem protocol is supported.	RO	0b0
[2]	Reserved	Reserved	RO	
[1]	CachePoisonInjectionBusy	Hardware loads 1 to this bit when the Start bit is written. Hardware must clear this bit to indicate that it has indeed finished poisoning a packet. Software is permitted to poll on this bit to find out when hardware has finished poison injection. This bit is required only if CXL.cache protocol is supported.	RO	0b0
[0]	Reserved	Reserved	RO	

### 8.3.4.76 por\_ccla\_cxl\_viral\_prop\_en

Bit Vector which controls viral propagation. Each bit represents the logical ID of corresponding CML gateway block.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE40

##### Type

RW

##### Reset value

See individual bit resets

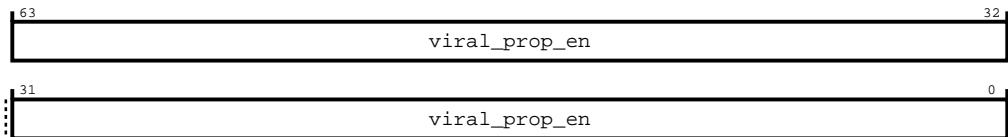
#### Usage constraints

This register is owned in the Root space and is 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 8-169: por\_ccla\_cxl\_viral\_prop\_en**



**Table 8-170: por\_ccla\_cxl\_viral\_prop\_en attributes**

Bits	Name	Description	Type	Reset
[63:0]	viral_prop_en	<p>Bit vector, where each bit represents logical ID of a CML gateway block present on CMN. Each bit when set, enables propagation of CXL Viral to that gateway block.</p> <p><b>0b0</b> Viral propagation is disabled</p> <p><b>0b1</b> Viral propagation is enabled</p>	RW	0b0

### 8.3.4.77 por\_ccla\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD908

##### Type

RW

##### Reset value

See individual bit resets

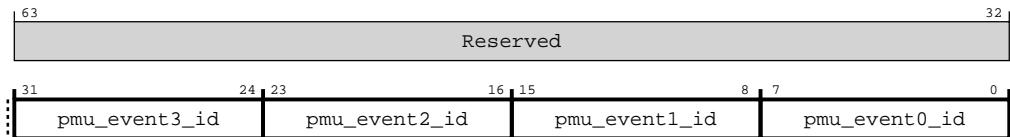
#### Usage constraints

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 8-170: por\_ccla\_pmu\_event\_sel**



**Table 8-171: por\_ccla\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:24]	pmu_event3_id	CCLA PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[23:16]	pmu_event2_id	CCLA PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[15:8]	pmu_event1_id	CCLA PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	<p>CCLA PMU Event 0 ID</p> <p><b>0x00</b> No event</p> <p><b>0x21</b> LA_RX_CXS: number of RX CXS beats</p> <p><b>0x22</b> LA_TX_CXS: number of TX CXS beats</p> <p><b>0x23</b> LA_RX_CXS_AVG_SIZE: average size of RX CXS beats</p> <p><b>0x24</b> LA_TX_CXS_AVG_SIZE: average size of TX CXS beats</p> <p><b>0x25</b> LA_TX_CXS_LCRD_BACKPRESSURE: CXS backpressure due to lack of CXS credits</p> <p><b>0x26</b> LA_LINK_CRDBUF_OCC: CCLA RX RAM buffer occupancy</p> <p><b>0x27</b> LA_LINK_CRDBUF_ALLOC: CCLA RX RAM buffer allocation</p> <p><b>0x28</b> PFWD RCVR CXS beats</p> <p><b>0x29</b> PFWD SNDR NUM FLITS</p> <p><b>0x2A</b> PFWD SNDR Number of message stalls due to static credits</p> <p><b>0x2B</b> PFWD SNDR Number of message stalls due to dynamic credits</p>	RW	0b0

### 8.3.4.78 por\_ccla\_c2c\_port\_register\_offset\_header

C2C Port Register Offset Header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xF700

### Type

RO

### Reset value

See individual bit resets

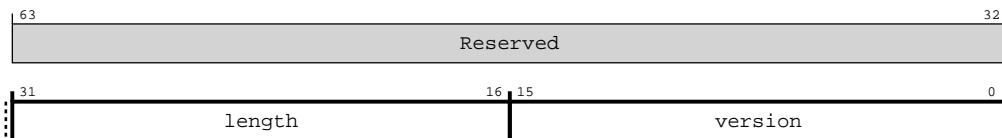
### Usage constraints

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 8-171: por\_ccla\_c2c\_port\_register\_offset\_header**



**Table 8-172: por\_ccla\_c2c\_port\_register\_offset\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	length	Length in bytes of this register block.	RO	0x20
[15:0]	version	Version of the Port Register Offset register block.	RO	0x0

## 8.3.4.79 por\_ccla\_c2c\_port\_register\_offset\_table\_0

### C2C Port Register Offset Table 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xF708

### Type

RO

### Reset value

See individual bit resets

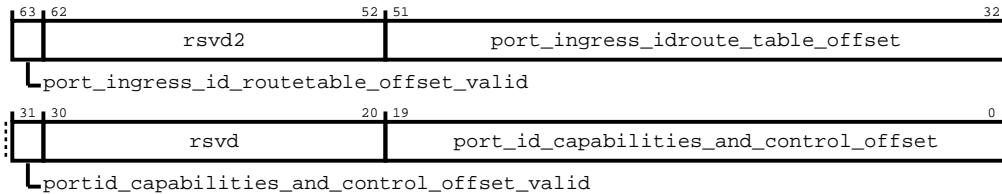
### Usage constraints

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 8-172: por\_ccla\_c2c\_port\_register\_offset\_table\_0**



**Table 8-173: por\_ccla\_c2c\_port\_register\_offset\_table\_0 attributes**

Bits	Name	Description	Type	Reset
[63]	port_ingress_id_routetable_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[62:52]	rsvd2	rsvd2	RO	0x0
[51:32]	port_ingress_idroutetable_offset	Offset from the Port Registers Base for this register block.	RO	0x2060
[31]	portid_capabilities_and_control_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[30:20]	rsvd	rsvd	RO	0x0
[19:0]	port_id_capabilities_and_control_offset	Offset from the Port Capabilities Registers Base for this register block.	RO	0x2020

### 8.3.4.80 por\_ccla\_c2c\_port\_register\_offset\_table\_1

#### C2C Port Register Offset Table 1

##### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

##### Attributes

###### Width

64

### Address offset

0xF710

### Type

RO

### Reset value

See individual bit resets

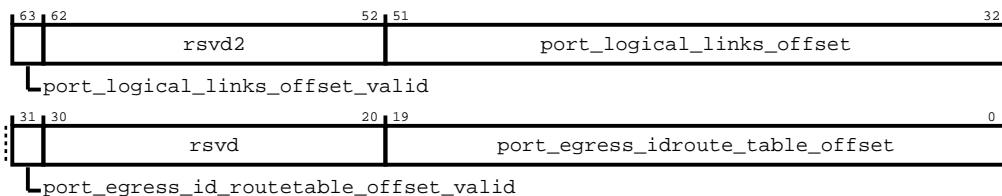
### Usage constraints

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 8-173: por\_ccla\_c2c\_port\_register\_offset\_table\_1**



**Table 8-174: por\_ccla\_c2c\_port\_register\_offset\_table\_1 attributes**

Bits	Name	Description	Type	Reset
[63]	port_logical_links_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[62:52]	rsvd2	rsvd	RO	0x0
[51:32]	port_logical_links_offset	Offset from the Port Registers Base for this register block.	RO	0x2200
[31]	port_egress_id_routetable_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[30:20]	rsvd	rsvd	RO	0x0
[19:0]	port_egress_idroute_table_offset	Offset from the Port Registers Base for this register block.	RO	0x20B0

### 8.3.4.81 por\_ccla\_c2c\_port\_capabilities\_and\_control\_header

C2C Port Register capabilities and control header

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

0xF718

### Type

RO

### Reset value

See individual bit resets

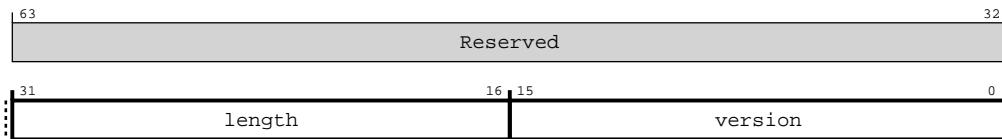
### Usage constraints

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 8-174: por\_ccla\_c2c\_port\_capabilities\_and\_control\_header**



**Table 8-175: por\_ccla\_c2c\_port\_capabilities\_and\_control\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	length	Length in bytes of this register block.	RO	0x40
[15:0]	version	Version value is 0h for this version of the spec.	RO	0x0

### 8.3.4.82 por\_ccla\_c2c\_port\_capabilities\_1

C2C Port Register capabilities and control register 1

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF720

#### Type

RO

### Reset value

See individual bit resets

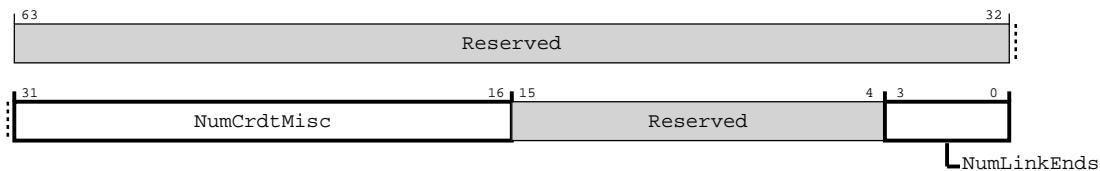
### Usage constraints

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 8-175: por\_ccla\_c2c\_port\_capabilities\_1**



**Table 8-176: por\_ccla\_c2c\_port\_capabilities\_1 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	NumCrdtMisc	Number of Misc Credits for this Port.	RO	0x40
[15:4]	Reserved	Reserved	RO	-
[3:0]	NumLinkEnds	Number of Link Ends for this Port. Value is 0 when the Port is disconnected.	RO	0x1

### 8.3.4.83 por\_ccla\_c2c\_port\_capabilities\_2

C2C Port Register capabilities and control register 1

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF728

#### Type

RO

#### Reset value

See individual bit resets

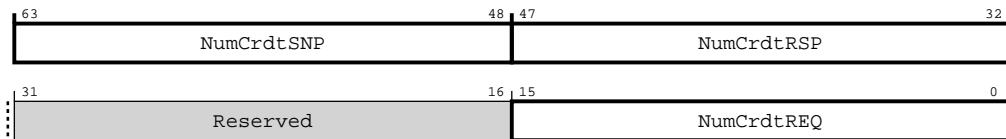
## Usage constraints

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 8-176: por\_ccla\_c2c\_port\_capabilities\_2**



**Table 8-177: por\_ccla\_c2c\_port\_capabilities\_2 attributes**

Bits	Name	Description	Type	Reset
[63:48]	NumCrdtSNP	Number is Snoop Credits capable.	RO	0x40
[47:32]	NumCrdtRSP	Number is Rsp Credits capable.	RO	0x40
[31:16]	Reserved	Reserved	RO	-
[15:0]	NumCrdtREQ	Number is Request Credits capable.	RO	0x0

## 8.3.4.84 por\_ccla\_c2c\_port\_control\_and\_status

C2C Port Register capabilities and control and status

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF730

#### Type

RW

#### Reset value

See individual bit resets

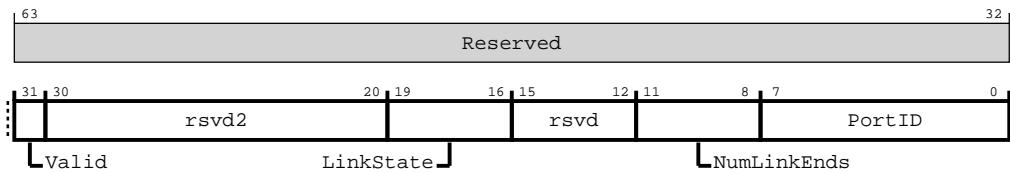
### Usage constraints

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 8-177: por\_ccla\_c2c\_port\_control\_and\_status**



**Table 8-178: por\_ccla\_c2c\_port\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	Valid	Indicates this register has been programmed and is valid.	RW	0x0
[30:20]	rsvd2	Rsvd	RW	0x0
[19:16]	LinkState	Link State for the Physical Link.  Bit definitions  0 :Active 1      Disconnected 7:2     Reserved	RW	0x2
[15:12]	rsvd	Rsvd	RW	0x0
[11:8]	NumLinkEnds	Number of Link Ends in this Port	RW	0x1
[7:0]	PortID	Port ID for this port	RW	0x0

### 8.3.4.85 por\_ccla\_c2c\_port\_ingressid\_route\_table\_header

C2C Port ID route table header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF738

### Type

RO

### Reset value

See individual bit resets

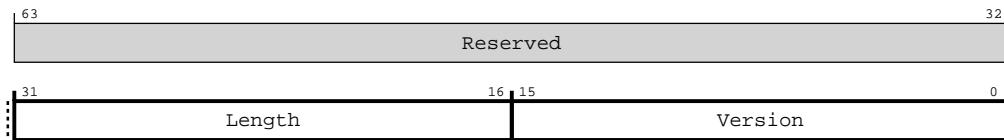
### Usage constraints

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 8-178: por\_ccla\_c2c\_port\_ingressid\_route\_table\_header**



**Table 8-179: por\_ccla\_c2c\_port\_ingressid\_route\_table\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x48
[15:0]	Version	Version of the Port ID Route Table register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.4.86 por\_ccla\_c2c\_port\_ingressid\_route\_table\_capabilities

C2C Port ID route table capabilities

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF828

#### Type

RO

#### Reset value

See individual bit resets

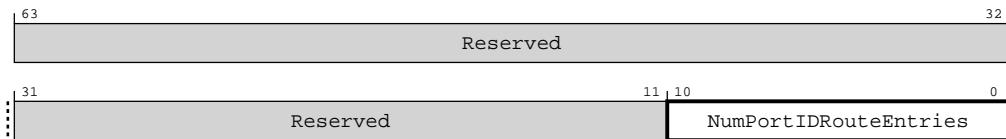
## Usage constraints

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 8-179: por\_ccla\_c2c\_port\_ingressid\_route\_table\_capabilities**



**Table 8-180: por\_ccla\_c2c\_port\_ingressid\_route\_table\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10:0]	NumPortIDRouteEntries	NumPortIdRouteEntries - Number of Route Id Entries supported. Valid values are 0 - 2^NumChipIDBits.	RO	0x0

## 8.3.4.87 por\_ccla\_c2c\_port\_ingressid\_route\_table\_control\_and\_status

C2C Port ID route table control and status register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF740

#### Type

RW

#### Reset value

See individual bit resets

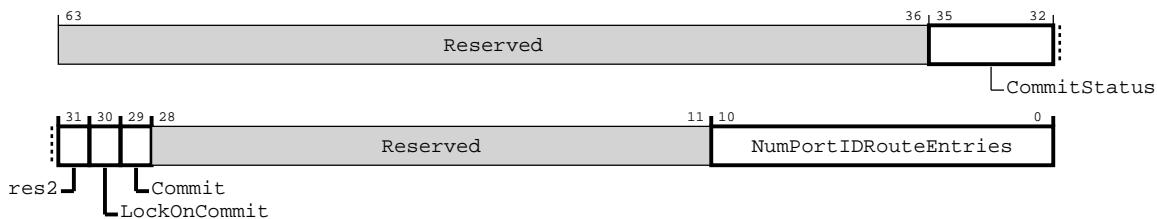
#### Usage constraints

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 8-180: por\_ccla\_c2c\_port\_ingressid\_route\_table\_control\_and\_status**



**Table 8-181: por\_ccla\_c2c\_port\_ingressid\_route\_table\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35:32]	CommitStatus	Status of address decoder commit. Bit definitions:  0 Committed - Value of 1 indicated commit is complete 1 Error - commit failed to complete 2-3 Reserved After a commit, the Committed and/or Error bit remain set until the Commit bit is cleared. Clearing the Commit bit will cause all bits in this field to be cleared.	RW	0x0
[31]	res2	Reserved	RW	0x0
[30]	LockOnCommit	Lock the programming of the route table when committed Default Values:  0 - No Lock on Commit 1 - Lock On Commit When Lock On Commit is enabled, once the committed,all the route tablesare read only until reset	RW	0x0
[29]	Commit	Commit the updates to the address decoder. Commit bit will remain set until cleared by software. This bit must be cleared to initiate a subsequent commit.	RW	0x0
[28:11]	Reserved	Reserved	RW	0x0
[10:0]	NumPortIDRouteEntries	Number of Route ID Entries enabled. Valid values are 0 - 2^NumChipIDBits.	RW	0x0

### 8.3.4.88 por\_ccla\_c2c\_port\_ingressid\_route\_entry\_0

C2C Port ID route table entry 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF748

### Type

RW

### Reset value

See individual bit resets

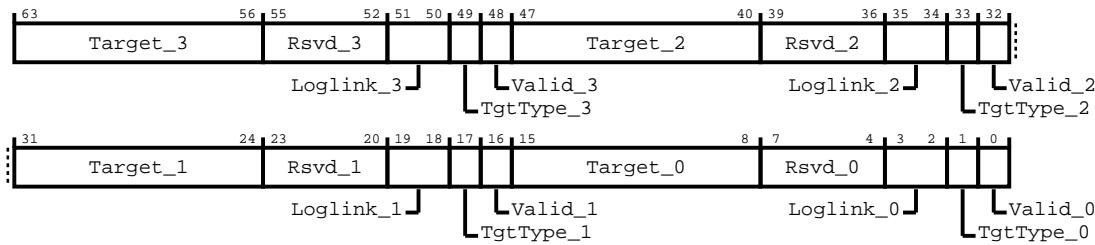
### Usage constraints

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 8-181: por\_ccla\_c2c\_port\_ingressid\_route\_entry\_0**



**Table 8-182: por\_ccla\_c2c\_port\_ingressid\_route\_entry\_0 attributes**

Bits	Name	Description	Type	Reset
[63:56]	Target_3	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[55:52]	Rsvd_3	Reserved	RW	0x0
[51:50]	Loglink_3	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[49]	TgtType_3	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[48]	Valid_3	Indicates this register has been programmed and is valid.	RW	0x0
[47:40]	Target_2	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[39:36]	Rsvd_2	Reserved	RW	0x0
[35:34]	Loglink_2	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[33]	TgtType_2	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[32]	Valid_2	Indicates this register has been programmed and is valid.	RW	0x0
[31:24]	Target_1	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[23:20]	Rsvd_1	Reserved	RW	0x0
[19:18]	Loglink_1	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[17]	TgtType_1	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[16]	Valid_1	Indicates this register has been programmed and is valid.	RW	0x0

Bits	Name	Description	Type	Reset
[15:8]	Target_0	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[7:4]	Rsvd_0	Reserved	RW	0x0
[3:2]	Loglink_0	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[1]	TgtType_0	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[0]	Valid_0	Indicates this register has been programmed and is valid.	RW	0x0

### 8.3.4.89 por\_ccla\_c2c\_port\_egressid\_route\_table\_header

C2C Port ID route table header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF750

##### Type

RO

##### Reset value

See individual bit resets

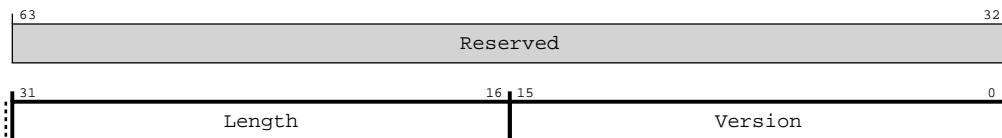
##### Usage constraints

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 8-182: por\_ccla\_c2c\_port\_egressid\_route\_table\_header**



**Table 8-183: por\_ccla\_c2c\_port\_egressid\_route\_table\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x48
[15:0]	Version	Version of the Port ID Route Table register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.4.90 por\_ccla\_c2c\_port\_egressid\_route\_table\_capabilities

C2C Port ID route table capabilities

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF820

##### Type

RO

##### Reset value

See individual bit resets

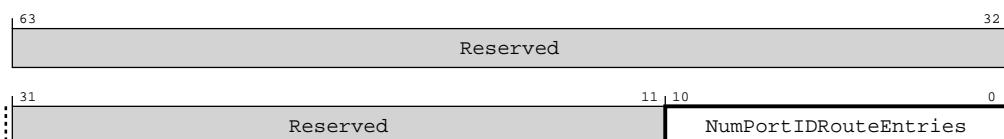
##### Usage constraints

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 8-183: por\_ccla\_c2c\_port\_egressid\_route\_table\_capabilities**



**Table 8-184: por\_ccla\_c2c\_port\_egressid\_route\_table\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[10:0]	NumPortIDRouteEntries	NumPortIdRouteEntries - Number of Route Id Entries supported. Valid values are 0 - 2^NumChipIDBits.	RO	0x0

### 8.3.4.91 por\_ccla\_c2c\_port\_egressid\_route\_table\_control\_and\_status

C2C Port ID route table control and status register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF758

##### Type

RW

##### Reset value

See individual bit resets

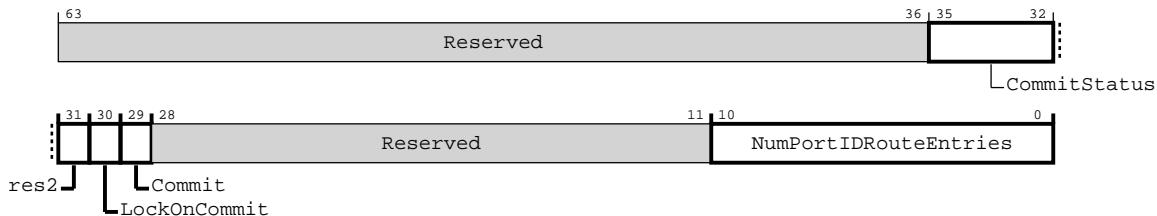
##### Usage constraints

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 8-184: por\_ccla\_c2c\_port\_egressid\_route\_table\_control\_and\_status**



**Table 8-185: por\_ccla\_c2c\_port\_egressid\_route\_table\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[35:32]	CommitStatus	Status of Route Table Commit Bit Definitions:  0 - Committed - Status of Commit 1 - Error - Commit failed 2-3 - ReservedOnce a commit is completed, to complete a subsequent Commit, the committed status must be cleared with a write of 1 to the Committed bit	RW	0x0
[31]	res2	res2	RW	0x0
[30]	LockOnCommit	Lock the programming of the route table when committed Default Values:  0 - No Lock on Commit 1 - Lock On Commit When Lock On Commit is enabled, once the committed, all the route tables are read only until reset	RW	0x0
[29]	Commit	Commit the updates to the decoder. Commit bit will remain set until Commit status -> Committed bit is cleared	RW	0x0
[28:11]	Reserved	Reserved	RW	0x0
[10:0]	NumPortIDRouteEntries	Number of Route ID Entries enabled. Valid values are 0 - 2^NumChipIDBits.	RW	0x0

### 8.3.4.92 por\_ccla\_c2c\_port\_egressid\_route\_entry\_0

C2C Port ID route table entry 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF760

##### Type

RW

##### Reset value

See individual bit resets

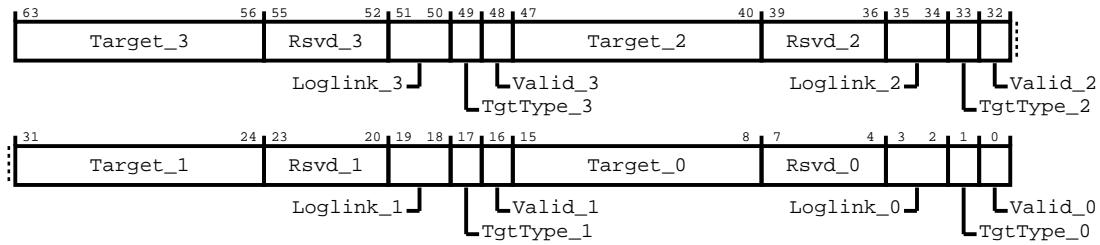
##### Usage constraints

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 8-185: por\_ccla\_c2c\_port\_egressid\_route\_entry\_0**



**Table 8-186: por\_ccla\_c2c\_port\_egressid\_route\_entry\_0 attributes**

Bits	Name	Description	Type	Reset
[63:56]	Target_3	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[55:52]	Rsvd_3	Reserved	RW	0x0
[51:50]	Loglink_3	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[49]	TgtType_3	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[48]	Valid_3	Indicates this register has been programmed and is valid.	RW	0x0
[47:40]	Target_2	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[39:36]	Rsvd_2	Reserved	RW	0x0
[35:34]	Loglink_2	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[33]	TgtType_2	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[32]	Valid_2	Indicates this register has been programmed and is valid.	RW	0x0
[31:24]	Target_1	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[23:20]	Rsvd_1	Reserved	RW	0x0
[19:18]	Loglink_1	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[17]	TgtType_1	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[16]	Valid_1	Indicates this register has been programmed and is valid.	RW	0x0
[15:8]	Target_0	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[7:4]	Rsvd_0	Reserved	RW	0x0
[3:2]	Loglink_0	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[1]	TgtType_0	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[0]	Valid_0	Indicates this register has been programmed and is valid.	RW	0x0

### 8.3.4.93 por\_ccla\_c2c\_logical\_linkend\_common\_header

C2C logical link end common header register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF768

##### Type

RO

##### Reset value

See individual bit resets

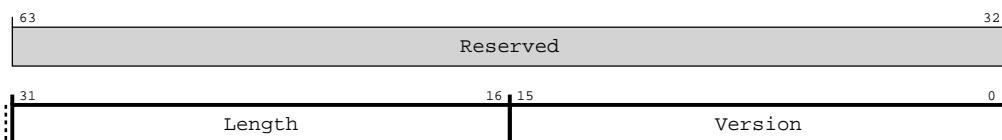
##### Usage constraints

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 8-186: por\_ccla\_c2c\_logical\_linkend\_common\_header**



**Table 8-187: por\_ccla\_c2c\_logical\_linkend\_common\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x40
[15:0]	Version	Version of the Logical Link End register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.4.94 por\_ccla\_c2c\_logical\_linkend\_offset\_register

C2C logical link end offset register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF770

##### Type

RO

##### Reset value

See individual bit resets

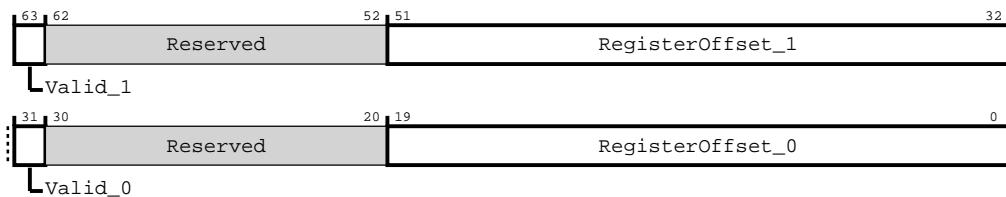
##### Usage constraints

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 8-187: por\_ccla\_c2c\_logical\_linkend\_offset\_register**



**Table 8-188: por\_ccla\_c2c\_logical\_linkend\_offset\_register attributes**

Bits	Name	Description	Type	Reset
[63]	Valid_1	Indicates this register has been programmed and is valid.	RO	0x0
[62:52]	Reserved	Reserved	RO	-
[51:32]	RegisterOffset_1	Offset from the Port Registers Base for this register block.	RO	0x0
[31]	Valid_0	Indicates this register has been programmed and is valid.	RO	0x1
[30:20]	Reserved	Reserved	RO	-
[19:0]	RegisterOffset_0	Offset from the Port Registers Base for this register block.	RO	0x2240

### 8.3.4.95 por\_ccla\_c2c\_logical\_linkend\_0-0\_header

There are 1 iterations of this register. The index ranges from 0 to 0. C2C logical link end header register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$0xE000 + 0x1778 + \{8 * \text{index}\}$

##### Type

RO

##### Reset value

See individual bit resets

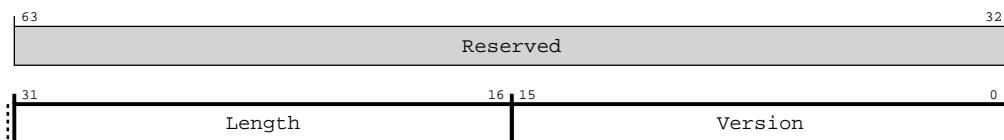
##### Usage constraints

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 8-188: por\_ccla\_c2c\_logical\_linkend\_0-0\_header**



**Table 8-189: por\_ccla\_c2c\_logical\_linkend\_0-0\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x180
[15:0]	Version	Version of the Logical Link End register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.4.96 por\_ccla\_c2c\_logical\_linkend\_0-0\_control\_and\_status

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Port ID route table control and status register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$0xE000 + 0x1780 + \{8 * \text{index}\}$

##### Type

RW

##### Reset value

See individual bit resets

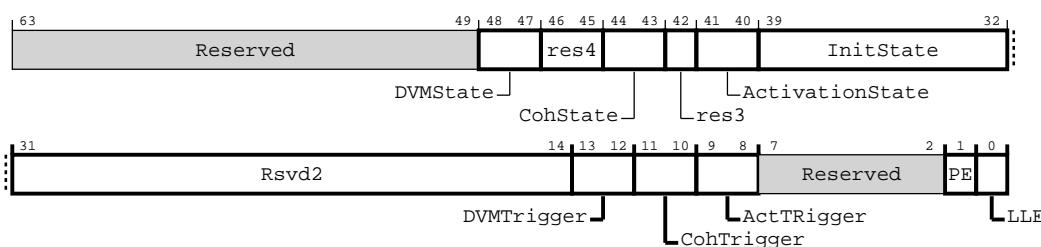
##### Usage constraints

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 8-189: por\_ccla\_c2c\_logical\_linkend\_0-0\_control\_and\_status**



**Table 8-190: por\_ccla\_c2c\_logical\_linkend\_0-0\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:48]	DVMState	DVM State for the Logical Link. Field values: 0 DVMDisabled 1 DVMConnect 2 DVMEnabled 3 DVMDDisconnect	RW	0x0
[47:46]	res4	Reserved	RW	0x0

Bits	Name	Description	Type	Reset
[45:44]	CohState	Coherency State for the Logical Link. Field values: 0 CohDisabled 1 CohConnect 2 CohEnabled 3 CohDisconnect	RW	0x0
[43:42]	res3	Reserved	RW	0x0
[41:40]	ActivationState	• Activity State for the Logical Link. Field values: 0 STOP 1 ACTIVATE 2 RUN 3 DEACTIVATE	RW	0x0
[39:32]	InitState	Initialization State for the Logical Link. Bit definitions: 0 LinkStatusRcvd 1 RemoteActive 2 PropNegComplete 7:3 Reserved	RW	0x0
[31:18]	Rsvd2	Reserved	RW	0x0
[17:16]	DVMTrigger	DVM Trigger	RW	0x0
[15:14]	Reserved6	Reserved	RW	0x0
[13:12]	CohTrigger	Coherency Trigger	RW	0x0
[11:10]	Reserved5	Reserved	RW	0x0
[9:8]	ActTRigger	Activation Trigger	RW	0x0
[7:2]	Reserved	Reserved	RW	0x0
[1]	PE	Property Exchange Enable Bit Values: 0 LogicalLinkEndEnd is disabled or not valid. 1 LogicalLinkEndEnd is enabled.	RW	0x1
[0]	LLE	LLE - LogicalLinkEndEnabled Bit Values: 0 LogicalLinkEndEnd is disabled or not valid. 1 LogicalLinkEndEnd is enabled.	RW	0x0

### 8.3.4.97 por\_ccla\_c2c\_logical\_linkend\_0-0\_map\_table

There are 1 iterations of this register. The index ranges from 0 to 0. C2C logical link end map table register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x1788 + #8\*index}

##### Type

RW

##### Reset value

See individual bit resets

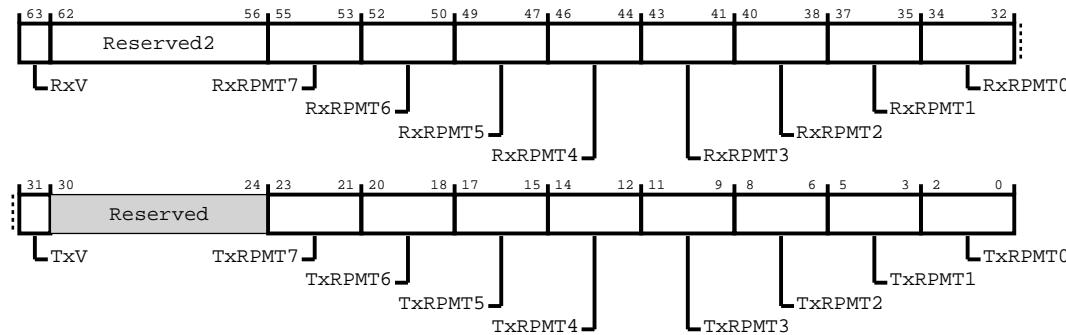
## Usage constraints

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 8-190: por\_ccla\_c2c\_logical\_linkend\_0-0\_map\_table**



**Table 8-191: por\_ccla\_c2c\_logical\_linkend\_0-0\_map\_table attributes**

Bits	Name	Description	Type	Reset
[63]	RxV	Rx RP Map Valid. Bit Values:  0: Rx RP Map is not valid  1: Rx RP Map is valid	RW	0x0
[62:56]	Reserved2	Reserved	RW	0x0
[55:53]	RxRPMT7	Value indicates on-chip RP mapped to C2C RP 7	RW	0x0
[52:50]	RxRPMT6	Value indicates on-chip RP mapped to C2C RP 6	RW	0x0
[49:47]	RxRPMT5	Value indicates on-chip RP mapped to C2C RP 5	RW	0x0
[46:44]	RxRPMT4	Value indicates on-chip RP mapped to C2C RP 4	RW	0x0
[43:41]	RxRPMT3	Value indicates on-chip RP mapped to C2C RP 3	RW	0x0
[40:38]	RxRPMT2	Value indicates on-chip RP mapped to C2C RP 2	RW	0x0
[37:35]	RxRPMT1	Value indicates on-chip RP mapped to C2C RP 1	RW	0x0
[34:32]	RxRPMT0	Value indicates on-chip RP mapped to C2C RP 0	RW	0x0
[31]	TxV	Tx RP Map Valid. Bit Values:  0: Tx RP Map is not valid  1: Tx RP Map is valid	RW	0x0
[30:24]	Reserved	Reserved	RW	0x0
[23:21]	TxRPMT7	Value indicates C2C RP mapped to on-chip RP 7	RW	0x0
[20:18]	TxRPMT6	Value indicates C2C RP mapped to on-chip RP 6	RW	0x0
[17:15]	TxRPMT5	Value indicates C2C RP mapped to on-chip RP 5	RW	0x0

Bits	Name	Description	Type	Reset
[14:12]	TxRPMT4	Value indicates C2C RP mapped to on-chip RP 4	RW	0x0
[11:9]	TxRPMT3	Value indicates C2C RP mapped to on-chip RP 3	RW	0x0
[8:6]	TxRPMT2	Value indicates C2C RP mapped to on-chip RP 2	RW	0x0
[5:3]	TxRPMT1	Value indicates C2C RP mapped to on-chip RP 1	RW	0x0
[2:0]	TxRPMT0	Value indicates C2C RP mapped to on-chip RP 0	RW	0x0

### 8.3.4.98 por\_ccla\_c2c\_linkproperties\_supported\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF790 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

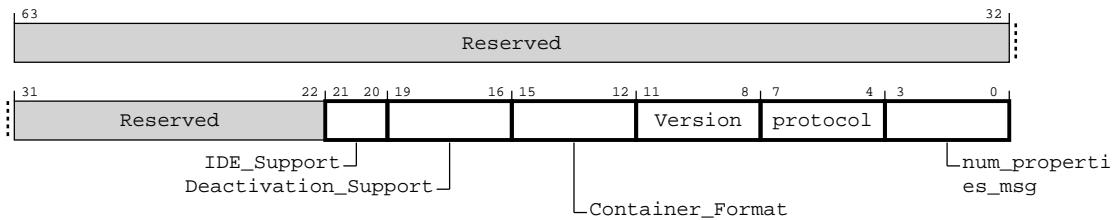
##### Usage constraints

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 8-191: por\_ccla\_c2c\_linkproperties\_supported\_uniform1\_0-0**



**Table 8-192: por\_ccla\_c2c\_linkproperties\_supported\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0: FormatX 0 Not supported 1 Supported  Bit 1: FormatY 0 Not supported 1 Supported  Bits 2: & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.4.99 por\_ccla\_c2c\_linkproperties\_advertised\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF798 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

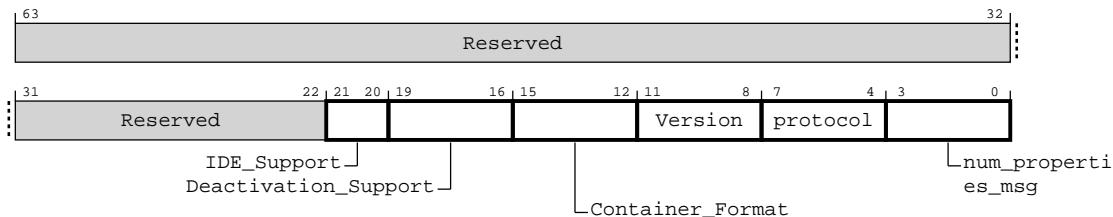
##### Usage constraints

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 8-192: por\_ccla\_c2c\_linkproperties\_advertised\_uniform1\_0-0**



**Table 8-193: por\_ccla\_c2c\_linkproperties\_advertised\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RW	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface  Bit 0 Protocol Agnostic 0 Not supported 1 Supported Bit 1 Protocol Aware 0 Not supported 1 Supported	RW	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions  Bit 0 FormatX 0 Not supported 1 Supported Bit 1 FormatY 0 Not supported 1 Supported Bits 2 & 3 Reserved	RW	0b0001
[11:8]	Version	0b0000 A Others Reserved	RW	0b0000
[7:4]	protocol	Linked to the Protocol property.	RW	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RW	0b0000

### 8.3.4.100 por\_ccla\_c2c\_linkproperties\_informed\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xF7A0 + #8\*index}

## Type

RO

## Reset value

See individual bit resets

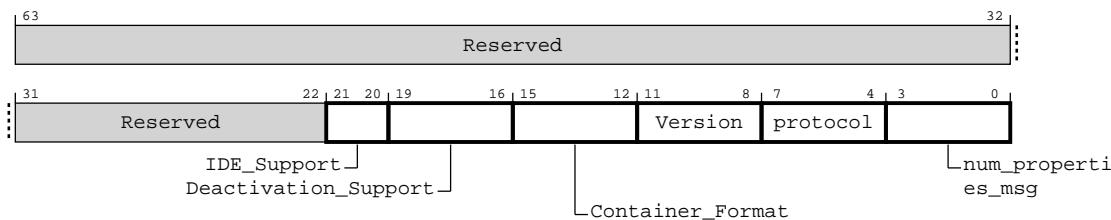
## Usage constraints

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 8-193: por\_ccla\_c2c\_linkproperties\_informed\_uniform1\_0-0**



**Table 8-194: por\_ccla\_c2c\_linkproperties\_informed\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0 FormatX 0 Not supported 1 Supported  Bit 1 FormatY 0 Not supported 1 Supported  Bits 2 & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.4.101 por\_ccla\_c2c\_linkproperties\_negotiated\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7A8 + #8\*index}

##### Type

RO

##### Reset value

See individual bit resets

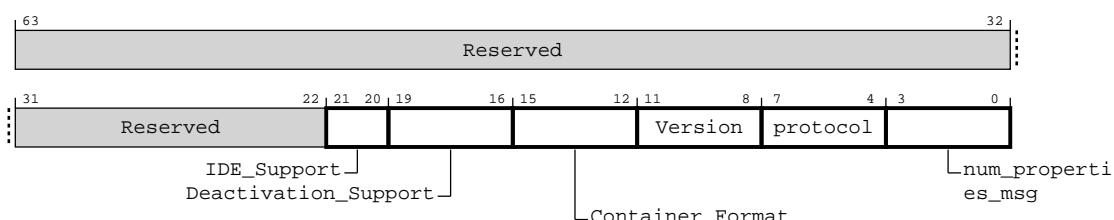
##### Usage constraints

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 8-194: por\_ccla\_c2c\_linkproperties\_negotiated\_uniform1\_0-0**



**Table 8-195: por\_ccla\_c2c\_linkproperties\_negotiated\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010

Bits	Name	Description	Type	Reset
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0 FormatX 0 Not supported 1 Supported  Bit 1 FormatY 0 Not supported 1 Supported  Bits 2 & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.4.102 por\_ccla\_c2c\_linkproperties\_supported\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7B0 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

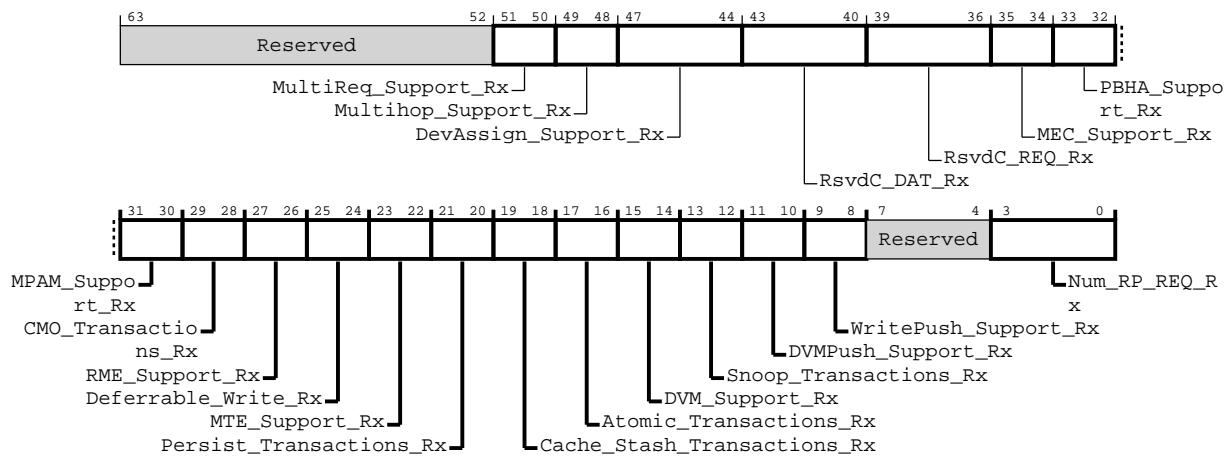
##### Usage constraints

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 8-195: por\_ccla\_c2c\_linkproperties\_supported\_rx1\_0-0**



**Table 8-196: por\_ccla\_c2c\_linkproperties\_supported\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RO	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RO	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00: False MTE not supported. 0b01: Reduced MTE transaction support. 0b10: Full MTE transaction support. 0b11: Reserved	RO	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00: False Persist CMO transactions not supported. 0b01: True Persist CMO transactions supported. Others: Reserved	RO	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RO	0b01
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0001

### 8.3.4.103 por\_ccla\_c2c\_linkproperties\_advertised\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7B8 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

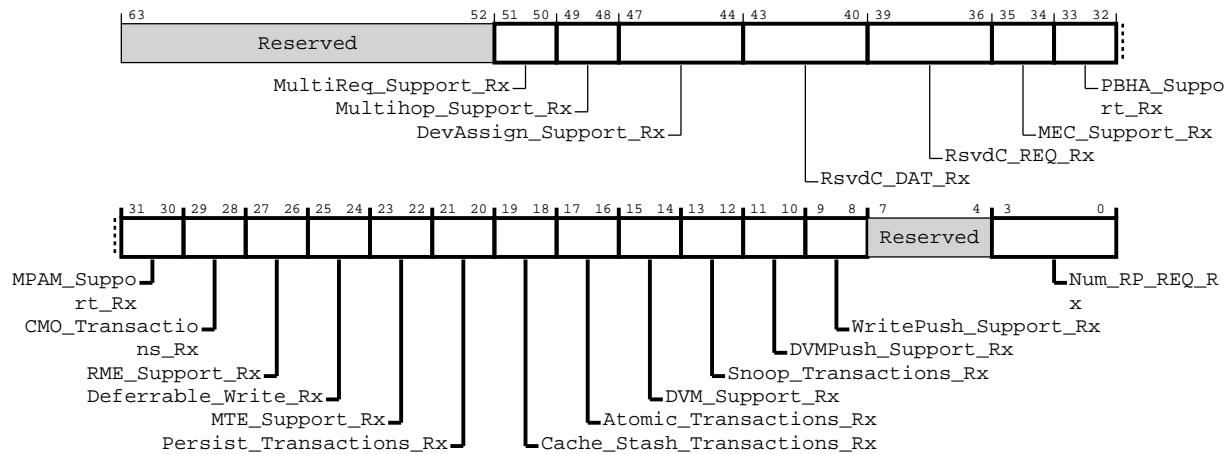
##### Usage constraints

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 8-196: por\_ccla\_c2c\_linkproperties\_advertised\_rx1\_0-0**



**Table 8-197: por\_ccla\_c2c\_linkproperties\_advertised\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.4.104 por\_ccla\_c2c\_linkproperties\_informed\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7C0 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

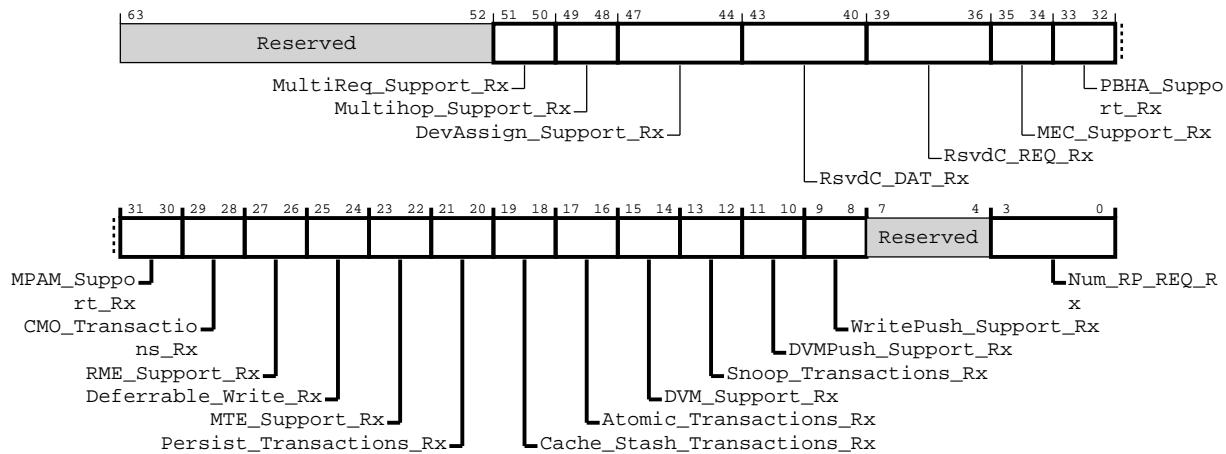
##### Usage constraints

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 8-197: por\_ccla\_c2c\_linkproperties\_informed\_rx1\_0-0**



**Table 8-198: por\_ccla\_c2c\_linkproperties\_informed\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RO	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	0b0001
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RO	0b00
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	0b00
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b00
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b00
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b00
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RO	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RO	0b00
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b00
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b00
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b00
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel: 0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0000

### 8.3.4.105 por\_ccla\_c2c\_linkproperties\_negotiated\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7C8 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

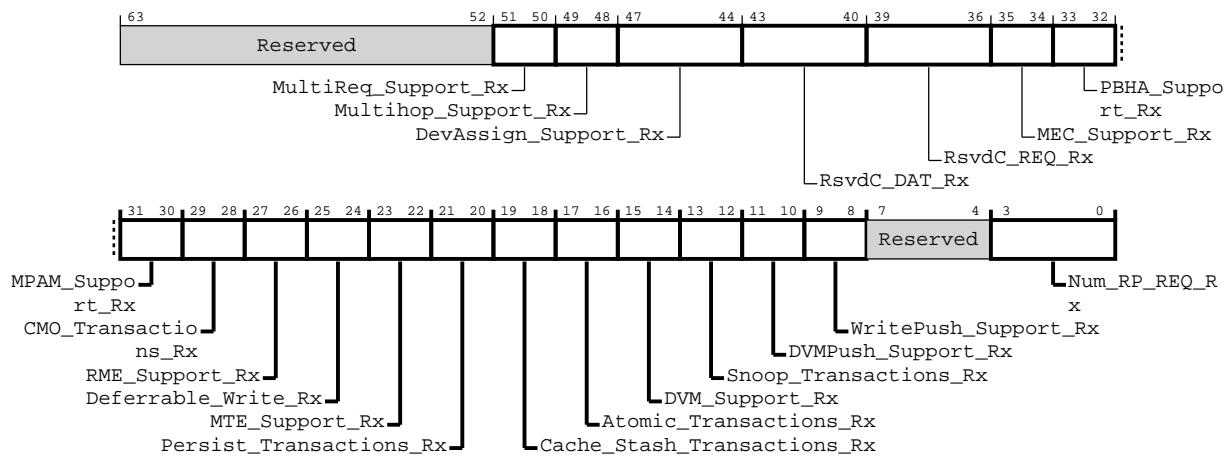
##### Usage constraints

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 8-198: por\_ccla\_c2c\_linkproperties\_negotiated\_rx1\_0-0**



**Table 8-199: por\_ccla\_c2c\_linkproperties\_negotiated\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RO	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	0b0001
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RO	0b00
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	0b00
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b00
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b00
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b00
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RO	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RO	0b00
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b00
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b00
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b00
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0000

### 8.3.4.106 por\_ccla\_c2c\_linkproperties\_supported\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7D0 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

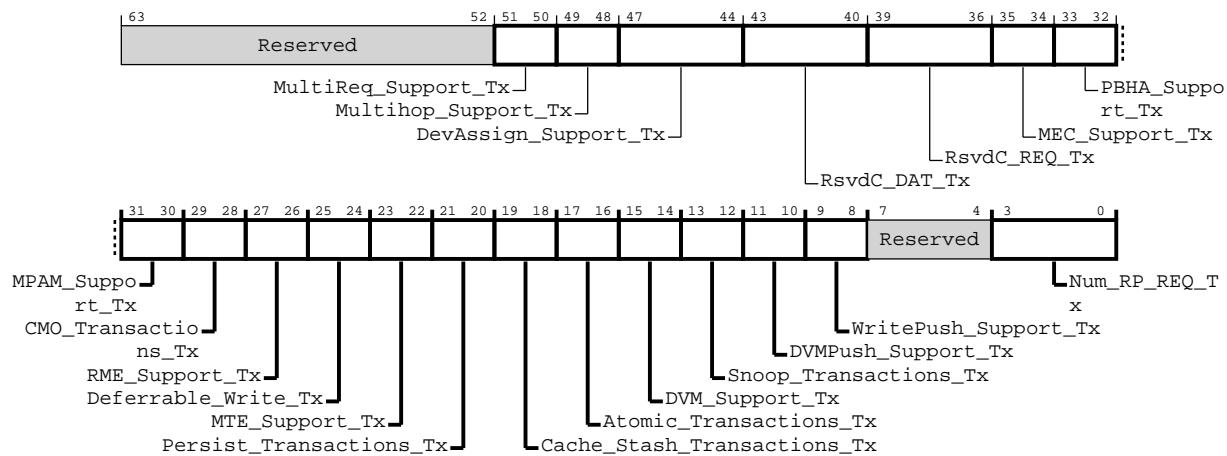
##### Usage constraints

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 8-199: por\_ccla\_c2c\_linkproperties\_supported\_tx1\_0-0**



**Table 8-200: por\_ccla\_c2c\_linkproperties\_supported\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RO	0b00
[49:48]	Multihop_Support_Tx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the transmitter chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RO	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the transmitter chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the transmitter chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a transmitter:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a transmitter:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a transmitter:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a transmitter:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RO	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a transmitter:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RO	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a transmitter:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a transmitter:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a transmitter:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a transmitter:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a transmitter:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush by the interface as a transmitter for Immediate write or Atomics:  0b00 False Will not send Immediate writes or Atomics using WritePush. 0b01 True Can send Immediate or Atomics using WritePush. Others Reserved	RO	0b01
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a transmitter on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0001

### 8.3.4.107 por\_ccla\_c2c\_linkproperties\_advertised\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7D8 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

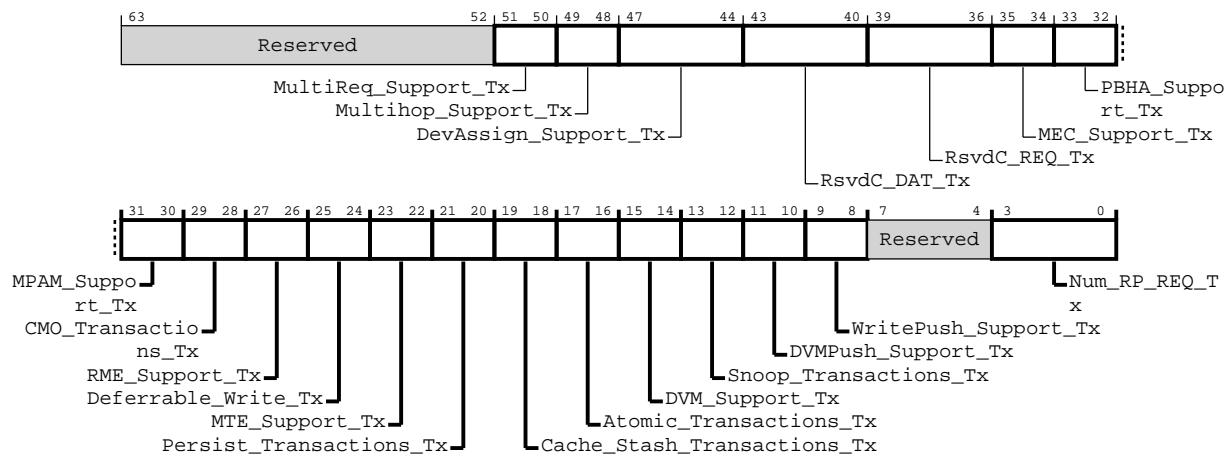
##### Usage constraints

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 8-200: por\_ccla\_c2c\_linkproperties\_advertised\_tx1\_0-0**



**Table 8-201: por\_ccla\_c2c\_linkproperties\_advertised\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RW	0b00
[49:48]	Multihop_Support_Tx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the transmitter chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the transmitter chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the transmitter chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a transmitter:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a transmitter:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a transmitter:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a transmitter:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a transmitter:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a transmitter:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a transmitter:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a transmitter:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a transmitter:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a transmitter:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush by the interface as a transmitter for Immediate write or Atomics:  0b00 False Will not send Immediate writes or Atomics using WritePush. 0b01 True Can send Immediate or Atomics using WritePush. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a transmitter on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.4.108 por\_ccla\_c2c\_linkproperties\_informed\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7E0 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

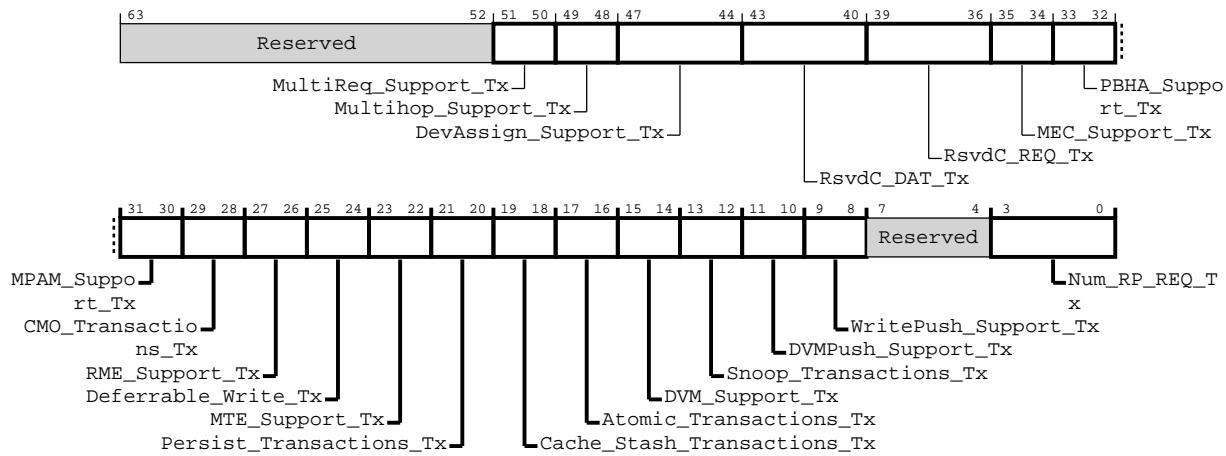
##### Usage constraints

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 8-201: por\_ccla\_c2c\_linkproperties\_informed\_tx1\_0-0**



**Table 8-202: por\_ccla\_c2c\_linkproperties\_informed\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RO	0b00
[49:48]	Multihop_Support_Tx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	0b0001
[35:34]	MEC_Support_Tx	Support for MEC on the transmitter chip:  0b00 False - MECID field not present 0b01 True - MECD field present Others Reserved	RO	0b00
[33:32]	PBHA_Support_Tx	Support for PBHA on the transmitter chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the transmitter chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	0b00
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a transmitter:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b00
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a transmitter:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b00
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a transmitter:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b00
[23:22]	MTE_Support_Tx	Support for MTE as a transmitter:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RO	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a transmitter:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RO	0b00
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a transmitter:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a transmitter:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b00
[15:14]	DVM_Support_Tx	Support for DVM messages as a transmitter:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b00
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a transmitter:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a transmitter:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b00
[9:8]	WritePush_Support_Tx	Support for WritePush by the interface as a transmitter for Immediate write or Atomics:  0b00 False Will not send Immediate writes or Atomics using WritePush. 0b01 True Can send Immediate or Atomics using WritePush. Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a transmitter on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0000

### 8.3.4.109 por\_ccla\_c2c\_linkproperties\_negotiated\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xF7E8 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

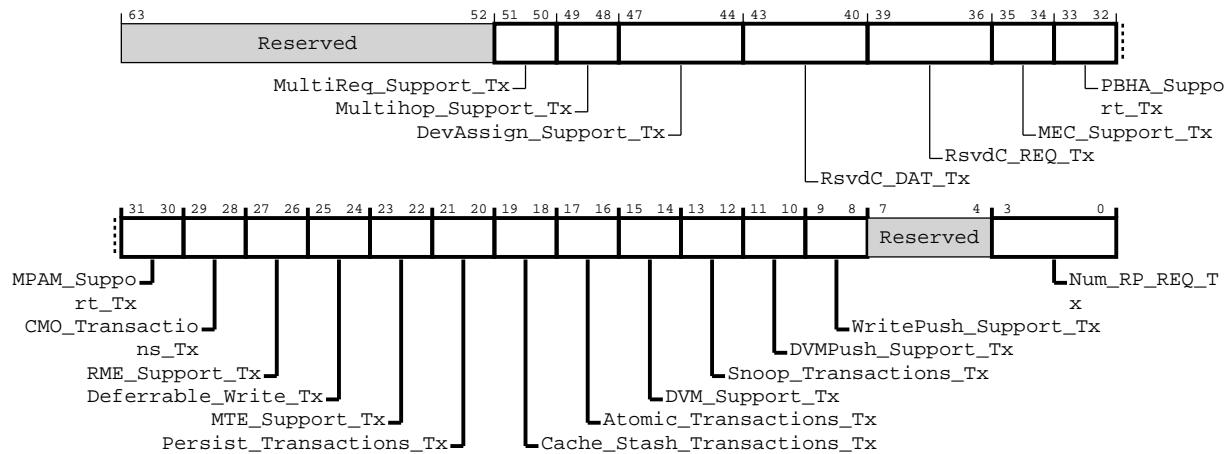
##### Usage constraints

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 8-202: por\_ccla\_c2c\_linkproperties\_negotiated\_tx1\_0-0**



**Table 8-203: por\_ccla\_c2c\_linkproperties\_negotiated\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RO	0b00
[49:48]	Multihop_Support_Tx	multihop support	RO	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RO	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RO	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the transmitter chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RO	0b0001
[35:34]	MEC_Support_Tx	Support for MEC on the transmitter chip:  0b00 False - MECID field not present 0b01 True - MECD field present Others Reserved	RO	0b00
[33:32]	PBHA_Support_Tx	Support for PBHA on the transmitter chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the transmitter chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RO	0b00
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a transmitter:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RO	0b00
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a transmitter:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RO	0b00
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a transmitter:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RO	0b00
[23:22]	MTE_Support_Tx	Support for MTE as a transmitter:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RO	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a transmitter:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RO	0b00
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a transmitter:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RO	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a transmitter:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RO	0b00
[15:14]	DVM_Support_Tx	Support for DVM messages as a transmitter:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RO	0b00
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a transmitter:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RO	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a transmitter:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RO	0b00
[9:8]	WritePush_Support_Tx	Support for WritePush by the interface as a transmitter for Immediate write or Atomics:  0b00 False Will not send Immediate writes or Atomics using WritePush. 0b01 True Can send Immediate or Atomics using WritePush. Others Reserved	RO	0b00

Bits	Name	Description	Type	Reset
[7:4]	Reserved	Reserved	RO	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a transmitter on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RO	0b0000

### 8.3.4.110 por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit1

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x17F8 + # {8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

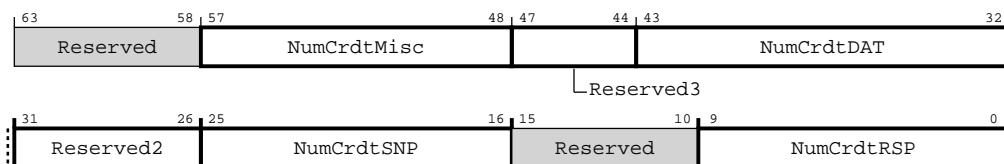
##### Usage constraints

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 8-203: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit1**



**Table 8-204: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit1 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	NumCrdtMisc	Max Number of Misc Credits allocated to this link end.	RW	0x0
[47:44]	Reserved3	Reserved	RW	0x0
[43:32]	NumCrdtDAT	Max Number of Data Credits allocated to this link end.	RW	Configuration dependent
[31:26]	Reserved2	Reserved	RW	0x0
[25:16]	NumCrdtSNP	Max Number of Snoop Credits allocated to this link end.	RW	Configuration dependent
[15:10]	Reserved	Reserved	RW	0x0
[9:0]	NumCrdtRSP	Max Number of Response Credits allocated to this link end. Num Crdt RSP[15:10] are reserved.	RW	0x100

### 8.3.4.111 por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit2

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 2

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x1800+ #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

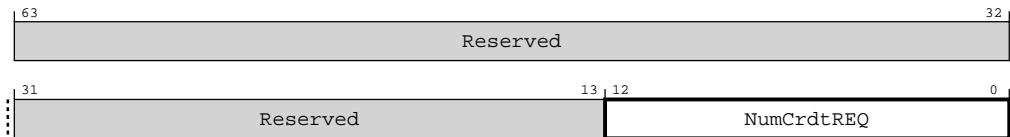
##### Usage constraints

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 8-204: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit2**



**Table 8-205: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit2 attributes**

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12:0]	NumCrdtREQ	Number of Total Request Credits allocated to this link end. Num Crdt REQ[15:13] are reserved.	RW	Configuration dependent

### 8.3.4.112 por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit3

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x1808+ #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

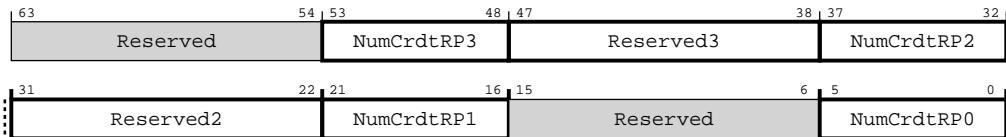
##### Usage constraints

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 8-205: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit3**



**Table 8-206: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit3 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53:48]	NumCrdtRP3	Number REQ MC credits allocated to RP3. NumCrdtRP0[15:6] are reserved.	RW	0x0
[47:38]	Reserved3	Reserved	RW	0x0
[37:32]	NumCrdtRP2	Number REQ MC credits allocated to RP2. NumCrdtRP0[15:6] are reserved.	RW	0x0
[31:22]	Reserved2	Reserved	RW	0x0
[21:16]	NumCrdtRP1	Number REQ MC credits allocated to RP1. NumCrdtRP0[15:6] are reserved.	RW	0x1
[15:6]	Reserved	Reserved	RW	0x0
[5:0]	NumCrdtRP0	Number REQ MC credits allocated to RP0. NumCrdtRP0[15:6] are reserved.	RW	0x0

### 8.3.4.113 por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit4

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 4

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x1810 + #8\*index

##### Type

RW

##### Reset value

See individual bit resets

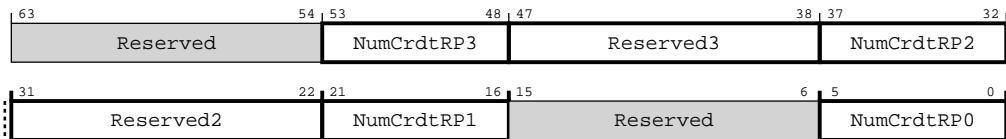
##### Usage constraints

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 8-206: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit4**



**Table 8-207: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit4 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53:48]	NumCrdtRP3	Number REQ MC credits allocated to RP3. NumCrdtRPO[15:6] are reserved.	RW	0x0
[47:38]	Reserved3	Reserved	RW	0x0
[37:32]	NumCrdtRP2	Number REQ MC credits allocated to RP2. NumCrdtRPO[15:6] are reserved.	RW	0x0
[31:22]	Reserved2	Reserved	RW	0x0
[21:16]	NumCrdtRP1	Number REQ MC credits allocated to RP1. NumCrdtRPO[15:6] are reserved.	RW	0x0
[15:6]	Reserved	Reserved	RW	0x0
[5:0]	NumCrdtRPO	Number REQ MC credits allocated to RPO. NumCrdtRPO[15:6] are reserved.	RW	0x0

### 8.3.4.114 por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit5

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 5

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE000 + 0x1818 + #8\*index}

##### Type

RW

##### Reset value

See individual bit resets

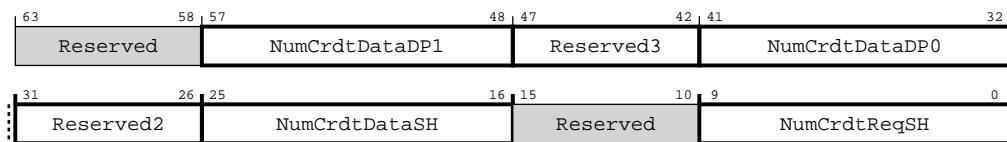
## Usage constraints

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 8-207: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit5**



**Table 8-208: por\_ccla\_c2c\_logical\_linkend\_0-0\_mc\_credit5 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	NumCrdtDataDP1	Number REQ MC credits allocated to Data DP1. NumCrdtRPO[15:6] are reserved.	RW	0x1
[47:42]	Reserved3	Reserved	RW	0x0
[41:32]	NumCrdtDataDP0	Number REQ MC credits allocated to RP2. NumCrdtRPO[15:6] are reserved.	RW	0x1
[31:26]	Reserved2	Reserved	RW	0x0
[25:16]	NumCrdtDataSH	Number DAT MC credits allocated to Shared. NumCrdtRPO[15:6] are reserved.	RW	Configuration dependent
[15:10]	Reserved	Reserved	RW	0x0
[9:0]	NumCrdtReqSH	Number REQ MC credits allocated to Shared. NumCrdtRPO[15:6] are reserved.	RW	Configuration dependent

### 8.3.4.115 por\_ccla\_errfr

Functions as the error feature register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE000

### Type

RO

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

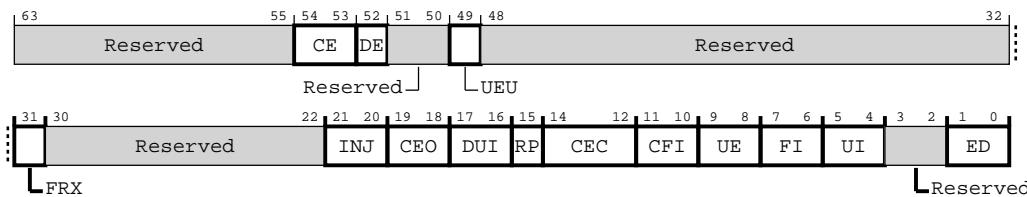
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-208: por\_ccla\_errfr**



**Table 8-209: por\_ccla\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording <b>0b00</b> Corrected Error not supported <b>0b10</b> Non-specific Corrected Error supported	RO	0b00
[52]	DE	Deferred Error recording <b>0b0</b> Deferred Error not supported <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension.  <b>0b1</b> por_ccla_errfr[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension.  <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite.  <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_ccla_errctlr.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when por_ccla_errfr.CEC != 0b000.)  <b>0b0</b> Invalid	RO	0b0
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model	RO	0b000
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_ccla_errctlr.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_ccla_errctlr.FI.	RO	0b10

Bits	Name	Description	Type	Reset
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_ccla_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_ccla_errctlr.ED	RO	0b10

### 8.3.4.116 por\_ccla\_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferral are enabled.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE008

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

##### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

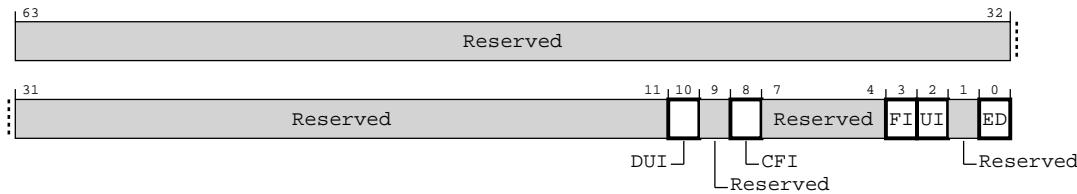
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-209: por\_ccla\_errctlr**



**Table 8-210: por\_ccla\_errctlr attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Enables error recovery interrupt for deferred error as specified in por_ccla_errfr.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_ccla_errfr.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_ccla_errfr.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_ccla_errfr.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_ccla_errfr.ED	RW	0b0

### 8.3.4.117 por\_ccla\_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

0xE010

##### Type

W1C

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

## Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

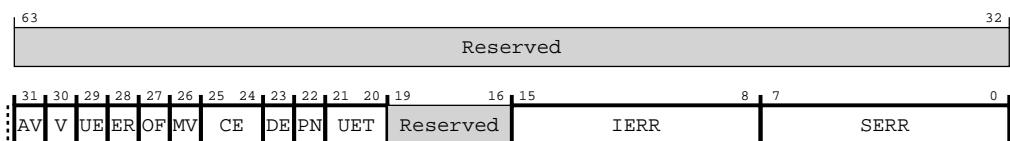
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-210: por\_ccla\_errstatus**



**Table 8-211: por\_ccla\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_ccla_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0

Bits	Name	Description	Type	Reset
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_ccla_errmisc<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b1</b> At least one error is not corrected and is deferred <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison <b>0b1</b> Uncorrected error recorded because a poison value was consumed <b>0b0</b> Other cases	W1C	0b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0 <b>0b01</b> Uncorrected error, Unrecoverable error (UEU). <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code. <b>0x00</b> No error <b>0x01</b> Partner implementation defined error	W1C	0b0

Bits	Name	Description	Type	Reset
[7:0]	SERR	<p><b>0x00</b> No error</p> <p><b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error - refer to por_ccla_errmisc1.ERRSRC</p> <p><b>0x02</b> Data value from (non-associative) internal memory. For example, Error Correction Code (ECC) from on-chip SRAM or buffer.</p> <p><b>0x05</b> Error detected on internal data path. For example, parity error in LA TX path</p> <p><b>0x12</b> Error response from Completer of access. For example, error response from cache write-back.</p> <p><b>0x13</b> External timeout. For example, timeout on interaction with another component.</p> <p><b>0x17 or 0x18</b> Deferred error from Completer/Requester passed through. For example, poisoned data received from the Completer/Requester of an access and returned to the Requester.</p>	W1C	0b0

### 8.3.4.118 por\_ccla\_erraddr

Contains the error record address.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE018

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

##### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

#### Usage constraints

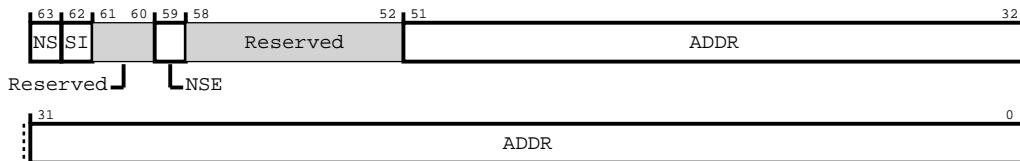
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the

por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-211: por\_ccla\_erraddr**



**Table 8-212: por\_ccla\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.4.119 por\_ccla\_errmisc1

Functions as the miscellaneous error register 1. Contains miscellaneous information about deferred/uncorrected errors.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE028

##### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_ccla_rcr.ras_secure_access_override`

### Secure group override

`por_ccla_scr.ras_secure_access_override`

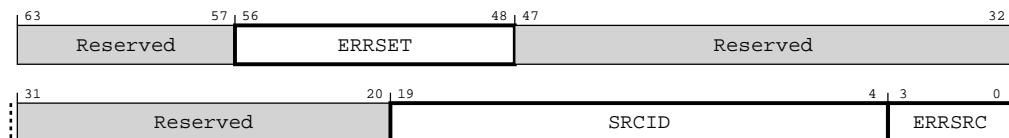
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_ccla_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_ccla_scr.ras_secure_access_override` bit and `por_ccla_rcr.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 8-212: por\_ccla\_errmisc1**



**Table 8-213: por\_ccla\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	0b0
[47:20]	Reserved	Reserved	RO	-
[19:4]	SRCID	Error source ID	RW	0b0

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	<p>Source of the parity error</p> <p><b>0b0000</b> Read data buffer 0</p> <p><b>0b0001</b> Read data buffer 1</p> <p><b>0b0010</b> Write data buffer 0</p> <p><b>0b0011</b> Write data buffer 1</p> <p><b>0b0100</b> Passive Buffer</p> <p><b>0b0101</b> CCLA Data RAM</p> <p><b>0b0110</b> PortFwd Data RAM</p> <p><b>0b0111</b> CXL Viral</p> <p><b>0b1000</b> CXL.Mem Poison</p> <p><b>0b1001</b> CXL.Cache Poison</p> <p><b>0b1010</b> CHI NDE (CXL.MEM mode)</p> <p><b>0b1011</b> Transaction timeout (CXL.MEM mode)</p> <p><b>0b1100</b> HDM decoder error</p> <p><b>0b1101</b> RA Transaction timeout</p> <p><b>0b1110</b> HA HDM LUT Parity or ECC double bit</p> <p><b>0b1111</b> Internal Datacheck error</p>	RW	0b000

### 8.3.4.120 por\_ccla\_errcfg

Functions as the Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE800

### Type

RO

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

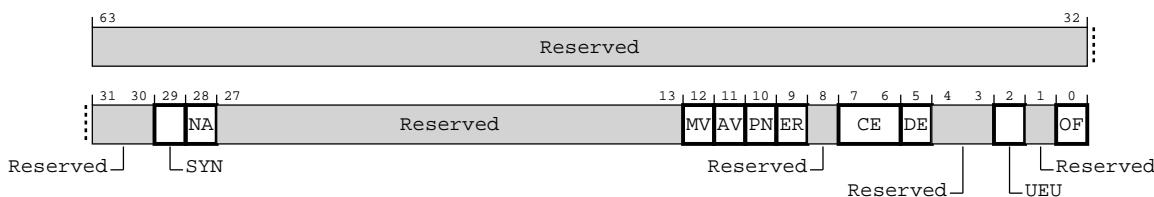
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-213: por\_ccla\_errpfgf**



**Table 8-214: por\_ccla\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update ERRSTATUS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	Miscellaneous syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	0b1
[11]	AV	Address syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	0b1
[10]	PN	Poison flag  <b>0b1</b> Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	0b1
[9]	ER	Error reported flag  <b>0b1</b> Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> No Corrected error generation.  <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 0b10; else, update ERRSTATUS.CE to 0b00	RO	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> No Deferred error generation.  <b>0b1</b> Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b1</b> If ERRPFGCTL.UEU == 1, update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01; else, update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b1</b> Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	0b1

### 8.3.4.121 por\_ccla\_errpfgctl

Functions as the Pseudo-fault Generation Control Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE808

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

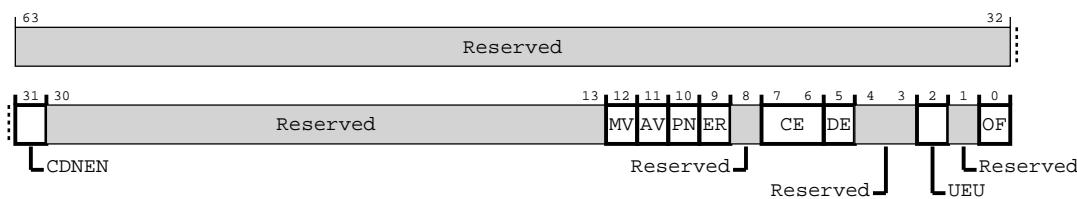
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-214: por\_ccla\_errpfgctl**



**Table 8-215: por\_ccla\_errpfgctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable.  <b>0b0</b> Countdown disabled.  <b>0b1</b> Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	<p>Miscellaneous syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.MV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.MV to 0b1</p>	RW	0b0
[11]	AV	<p>Address syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.AV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.AV to 0b1</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.PN to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.PN to 0b1</p>	RW	0b0
[9]	ER	<p>Error reported flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.ER to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.ER to 0b1</p>	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 0b00</p> <p><b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 0b10</p>	RW	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.DE to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.DE to 0b1</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00</p> <p><b>0b1</b> Fault injection update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01</p>	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	OF	<p>Overflow flag.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.OF to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.OF to 0b1</p>	RW	0b0

### 8.3.4.122 por\_ccla\_errpfgcdn

Functions as the Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE810

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

##### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

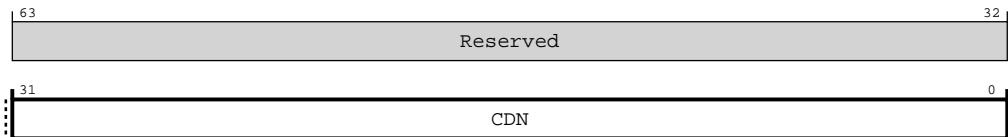
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-215: por\_ccla\_errpfgcdn**



**Table 8-216: por\_ccla\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.4.123 por\_ccla\_errfr\_NS

Functions as the non-secure error feature register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE040

##### Type

RO

##### Reset value

See individual bit resets

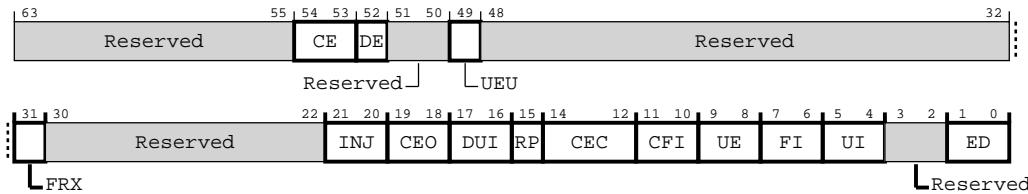
##### Usage constraints

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 8-216: por\_ccla\_errfr\_NS**



**Table 8-217: por\_ccla\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording <b>0b00</b> Corrected Error not supported <b>0b10</b> Non-specific Corrected Error supported	RO	0b00
[52]	DE	Deferred Error recording <b>0b0</b> Deferred Error not supported <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording <b>0b0</b> Unrecoverable Error not supported <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. <b>0b1</b> por_ccla_errfr_NS[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. <b>0b1</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite. <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00

Bits	Name	Description	Type	Reset
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_ccla_errctlr_NS.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when por_ccla_errfr_NS.CEC != 0b000.)  <b>0b0</b> Invalid	RO	0b0
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model	RO	0b000
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_ccla_errctlr_NS.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_ccla_errctlr_NS.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_ccla_errctlr_NS.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_ccla_errctlr_NS.ED	RO	0b10

### 8.3.4.124 por\_ccla\_errctlr\_NS

Functions as the non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE048

### Type

RW

### Reset value

See individual bit resets

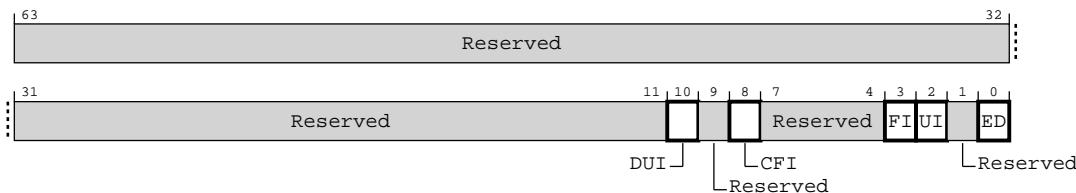
### Usage constraints

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 8-217: por\_ccla\_errctlr\_NS**



**Table 8-218: por\_ccla\_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 por_ccla_errfr_NS.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_ccla_errfr_NS.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_ccla_errfr_NS.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_ccla_errfr_NS.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_ccla_errfr_NS.ED	RW	0b0

### 8.3.4.125 por\_ccla\_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

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

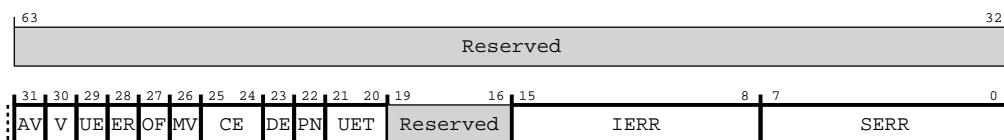
#### Usage constraints

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 8-218: por\_ccla\_errstatus\_NS**



**Table 8-219: por\_ccla\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_ccla_erraddr_NS contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0

Bits	Name	Description	Type	Reset
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_ccla_errmisc_NS<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b1</b> At least one error is not corrected and is deferred <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison <b>0b1</b> Uncorrected error recorded because a poison value was consumed <b>0b0</b> Other cases	W1C	0b0

Bits	Name	Description	Type	Reset
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	<b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error - refer to por_ccla_errmisc1.ERRSRC  <b>0x02</b> Data value from (non-associative) internal memory. For example, Error Correction Code (ECC) from on-chip SRAM or buffer.  <b>0x12</b> Error response from Completer of access. For example, error response from cache write-back.  <b>0x13</b> External timeout. For example, timeout on interaction with another component.  <b>0x17 or 0x18</b> Deferred error from Completer/Requester passed through. For example, poisoned data received from the Completer/Requester of an access and returned to the Requester.	W1C	0b0

### 8.3.4.126 por\_ccla\_erraddr\_NS

Contains the non-secure error record address.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE058

##### Type

RW

### Reset value

See individual bit resets

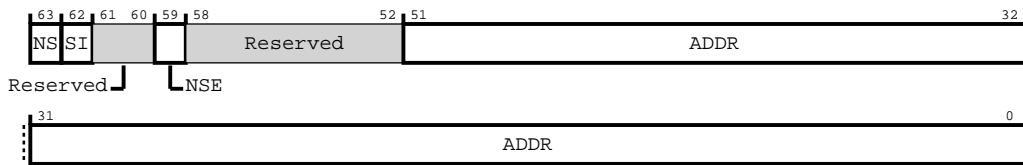
### Usage constraints

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 8-219: por\_ccla\_erraddr\_NS**



**Table 8-220: por\_ccla\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.4.127 por\_ccla\_errmisc1\_NS

Functions as the non-secure miscellaneous error register 1. Contains miscellaneous information about deferred/uncorrected errors.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

### Address offset

0xE068

### Type

RW

### Reset value

See individual bit resets

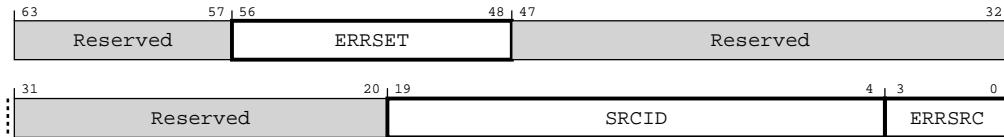
### Usage constraints

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 8-220: por\_ccla\_errmisc1\_NS**



**Table 8-221: por\_ccla\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	0b0
[47:20]	Reserved	Reserved	RO	-
[19:4]	SRCID	Error source ID	RW	0b0

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	<p>Source of the parity error</p> <p><b>0b0000</b> Read data buffer 0</p> <p><b>0b0001</b> Read data buffer 1</p> <p><b>0b0010</b> Write data buffer 0</p> <p><b>0b0011</b> Write data buffer 1</p> <p><b>0b0100</b> Passive Buffer</p> <p><b>0b0101</b> CCLA Data RAM</p> <p><b>0b0110</b> PortFwd Data RAM</p> <p><b>0b0111</b> CXL Viral</p> <p><b>0b1000</b> CXL.Mem Poison</p> <p><b>0b1001</b> CXL.Cache Poison</p> <p><b>0b1010</b> CHI NDE (CXL.MEM mode)</p> <p><b>0b1011</b> Transaction timeout (CXL.MEM mode)</p> <p><b>0b1100</b> HDM decoder error</p> <p><b>0b1101</b> RA Transaction timeout</p>	RW	0b000

### 8.3.4.128 por\_ccla\_errcfg\_NS

Functions as the non-secure Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xE840

## Type

RO

## Reset value

See individual bit resets

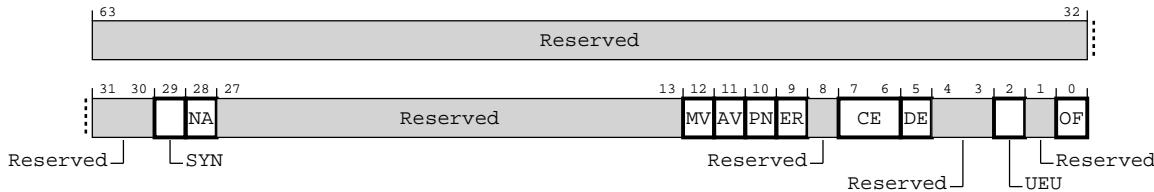
## Usage constraints

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 8-221: por\_ccla\_errpfgf\_NS**



**Table 8-222: por\_ccla\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update ERRSTATUS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b1</b> Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	0b1
[11]	AV	Address syndrome. <b>0b1</b> Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	0b1
[10]	PN	Poison flag <b>0b1</b> Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	0b1

Bits	Name	Description	Type	Reset
[9]	ER	Error reported flag  <b>0b1</b> Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> No Corrected error generation.  <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 0b10; else, update ERRSTATUS.CE to 0b00	RO	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> No Deferred error generation.  <b>0b1</b> Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b1</b> If ERRPFGCTL.UEU == 1, update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01; else, update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b1</b> Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	0b1

### 8.3.4.129 por\_ccla\_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

0xE848

##### Type

RW

##### Reset value

See individual bit resets

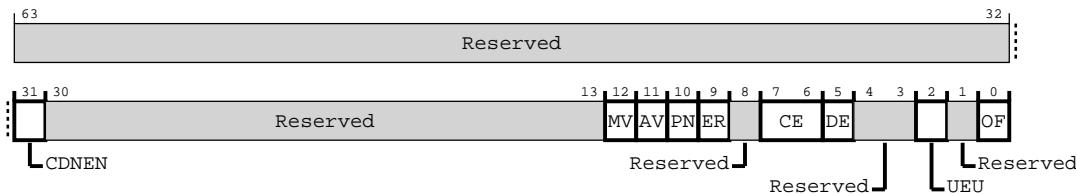
## Usage constraints

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 8-222: por\_ccla\_errpfgctl\_NS**



**Table 8-223: por\_ccla\_errpfgctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable.  <b>0b0</b> Countdown disabled.  <b>0b1</b> Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b0</b> Fault injection update ERRSTATUS.MV to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.MV to 0b1	RW	0b0
[11]	AV	Address syndrome.  <b>0b0</b> Fault injection update ERRSTATUS.AV to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.AV to 0b1	RW	0b0
[10]	PN	Poison flag  <b>0b0</b> Fault injection update ERRSTATUS.PN to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.PN to 0b1	RW	0b0

Bits	Name	Description	Type	Reset
[9]	ER	Error reported flag  <b>0b0</b> Fault injection update ERRSTATUS.ER to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 0b00  <b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 0b10	RW	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> Fault injection update ERRSTATUS.DE to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.DE to 0b1	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00  <b>0b1</b> Fault injection update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update ERRSTATUS.OF to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.OF to 0b1	RW	0b0

### 8.3.4.130 por\_ccla\_errpfcdn\_NS

Functions as the non-secure Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE850

### Type

RW

### Reset value

See individual bit resets

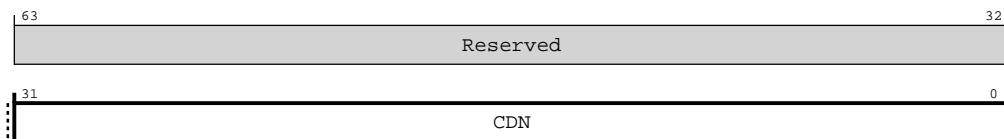
### Usage constraints

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 8-223: por\_ccla\_errpfgcdn\_NS**



**Table 8-224: por\_ccla\_errpfgcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.4.131 por\_ccla\_errcapctl

Functions as the error Capture Control Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xED00

#### Type

RW

#### Reset value

See individual bit resets

#### Root group override

por\_ccla\_rcr.ras\_secure\_access\_override

### Secure group override

por\_ccla\_scr.ras\_secure\_access\_override

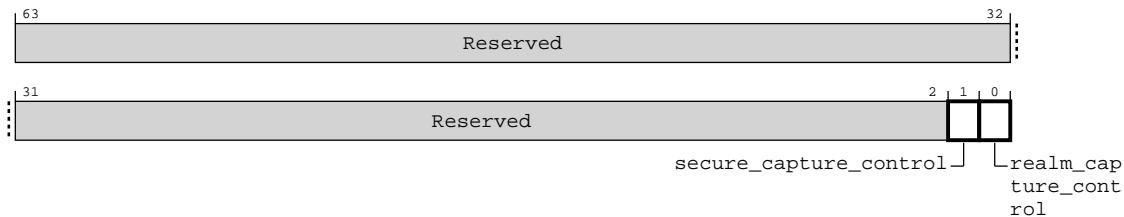
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.ras\_secure\_access\_override bit and por\_ccla\_rcr.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 8-224: por\_ccla\_errcapctl**



**Table 8-225: por\_ccla\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control <b>0b0</b> Transaction with secure PAS captured in root error record <b>0b1</b> Transaction with secure PAS captured in non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control <b>0b0</b> Transaction with realm PAS captured in root error record <b>0b1</b> Transaction with realm PAS captured in non-secure error record	RW	0b0

### 8.3.4.132 por\_ccla\_errgsr

Functions as Error Group Status Register

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEE00

### Type

RO

### Reset value

See individual bit resets

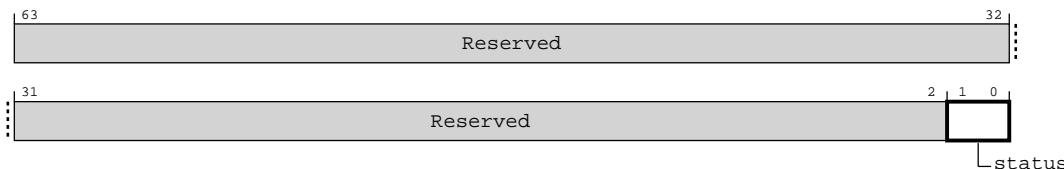
### Usage constraints

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 8-225: por\_ccla\_errgsr**



**Table 8-226: por\_ccla\_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.S.V, ERR<n>STATUS.V}	RO	0b0

## 8.3.4.133 por\_ccla\_errildr

Functions as the implementation identification register.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEE10

### Type

RO

### Reset value

See individual bit resets

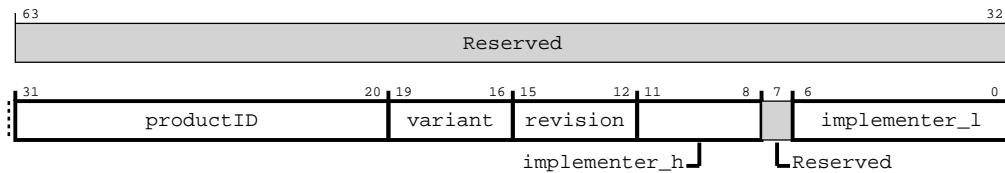
### Usage constraints

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 8-226: por\_ccla\_errildr**



**Table 8-227: por\_ccla\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.4.134 por\_ccla\_errdevaff

Functions as the device affinity register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEFA8

### Type

RO

### Reset value

See individual bit resets

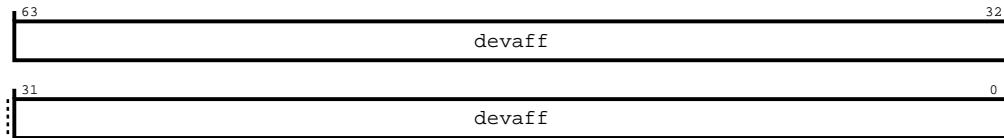
### Usage constraints

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 8-227: por\_ccla\_errdevaff**



**Table 8-228: por\_ccla\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.4.135 por\_ccla\_errdevarch

Functions as the device architecture register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEF8

#### Type

RO

#### Reset value

See individual bit resets

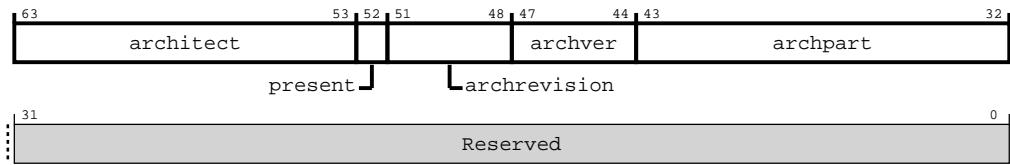
#### Usage constraints

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 8-228: por\_ccla\_errdevarch**



**Table 8-229: por\_ccla\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

### 8.3.4.136 por\_ccla\_errdevid

Functions as the device configuration register

## Configurations

This register is available in all configurations.

## Attributes

## Width

64

## Address offset

0xEF C8

## Type

RO

## Reset value

See individual bit resets

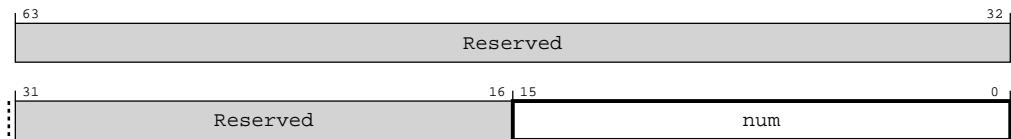
## Usage constraints

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 8-229: por\_ccla\_errdevid**



**Table 8-230: por\_ccla\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

## 8.3.4.137 por\_ccla\_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

0xEFDO

#### Type

RO

#### Reset value

See individual bit resets

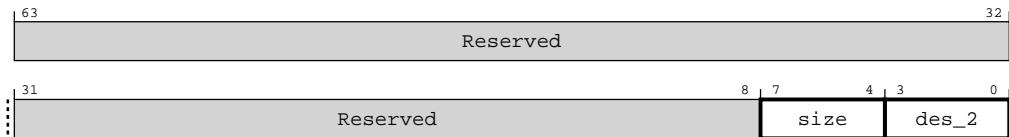
#### Usage constraints

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 8-230: por\_ccla\_errpidr45**



**Table 8-231: por\_ccla\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

### 8.3.4.138 por\_ccla\_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

0xEFE0

##### Type

RO

##### Reset value

See individual bit resets

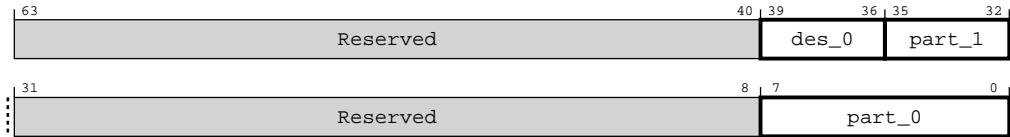
##### Usage constraints

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 8-231: por\_ccla\_errpidr01**



**Table 8-232: por\_ccla\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

### 8.3.4.139 por\_ccla\_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

0xEFE8

##### Type

RO

##### Reset value

See individual bit resets

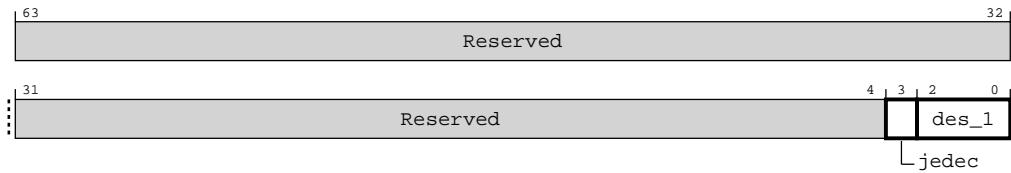
##### Usage constraints

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 8-232: por\_ccla\_errpidr23**



**Table 8-233: por\_ccla\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

### 8.3.4.140 por\_ccla\_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

0xEFF0

##### Type

RO

##### Reset value

See individual bit resets

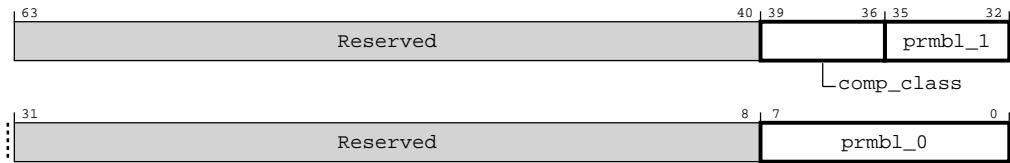
##### Usage constraints

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 8-233: por\_ccla\_errcidr01**



**Table 8-234: por\_ccla\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmbl_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmbl_0	PRMBL_0	RO	0xD

### 8.3.4.141 por\_ccla\_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

0xEFF8

##### Type

RO

##### Reset value

See individual bit resets

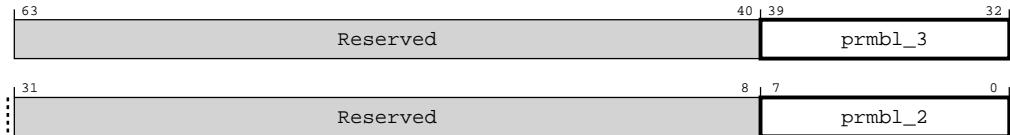
#### Usage constraints

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 8-234: por\_ccla\_errcidr23**



**Table 8-235: por\_ccla\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.4.142 por\_ccg\_multi\_mesh\_chn\_ctrl

Valid when two CHI TX response channels are enabled. Functions as the control register for Target based channel selection in Multi-Mesh Channel structure.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1BF8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_ccla\_rcr.cfg\_ctl

##### Secure group override

por\_ccla\_scr.cfg\_ctl

##### Usage constraints

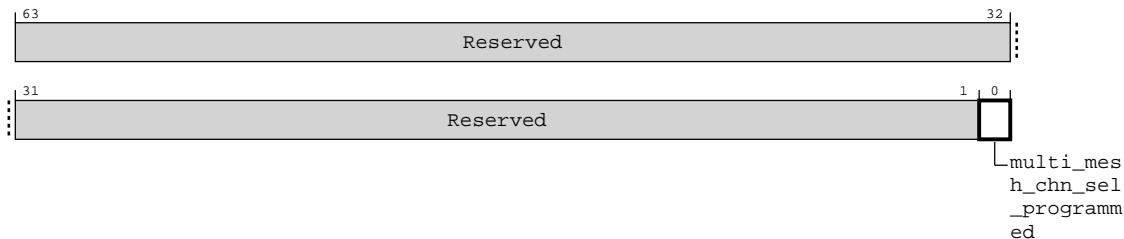
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cfg\_ctl bit and por\_ccla\_rcr.cfg\_ctl bit are set,

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 8-235: por\_ccg\_multi\_mesh\_chn\_ctrl**



**Table 8-236: por\_ccg\_multi\_mesh\_chn\_ctrl attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	multi_mesh_chn_sel_programmed	Indicates that multi CHI VC channel configured for all the targets specified in the channel select registers.	RW	0b0

### 8.3.4.143 por\_ccg\_multi\_mesh\_chn\_sel\_0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Valid when two CHI TX response channels are enabled. Functions as the CHI VC channel select per Target register in Multi-Mesh Channel structure. The register holds the target Ids for overriding the default channel selection and the response port to be used

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1C08 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_ccla\_rcr.cfg\_ctl

### Secure group override

por\_ccla\_scr.cfg\_ctl

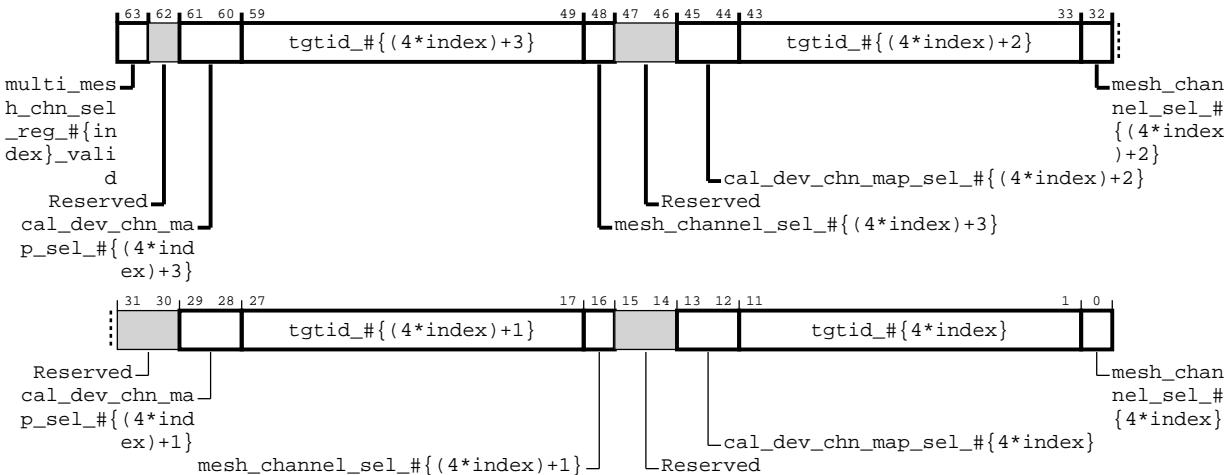
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_ccla\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_ccla\_scr.cfg\_ctl bit and por\_ccla\_rcr.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 8-236: por\_ccg\_multi\_mesh\_chn\_sel\_0-15**



**Table 8-237: por\_ccg\_multi\_mesh\_chn\_sel\_0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	multi_mesh_chn_sel_reg_{index}_valid	Indicates that multi mesh CHI VC channel configured for the targets specified in this register.	RW	0b0
[62]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[61:60]	cal_dev_chn_map_sel_{(4*index)+3}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b>            CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b>            CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b>            Reserved,</p> <p><b>0b11</b>            Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[59:49]	tgtid_{(4*index)+3}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[48]	mesh_channel_sel_{(4*index)+3}	CHI VC channel select:  1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[47:46]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[45:44]	cal_dev_chn_map_sel_{(4*index)+2}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b>            CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b>            CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b>            Reserved,</p> <p><b>0b11</b>            Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[43:33]	tgtid_{(4*index)+2}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[32]	mesh_channel_sel_{(4*index)+2}	CHI VC channel select:  1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[31:30]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[29:28]	cal_dev_chn_map_sel_#{(4*index)+1}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b>            CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b>            CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b>            Reserved,</p> <p><b>0b11</b>            Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[27:17]	tgtid_#{(4*index)+1}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[16]	mesh_channel_sel_#{(4*index)+1}	CHI VC channel select:  1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[15:14]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[13:12]	cal_dev_chn_map_sel_{4*index}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b> Reserved,</p> <p><b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[11:1]	tgtid_{4*index}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[0]	mesh_channel_sel_{4*index}	<p>CHI VC channel select:</p> <p>1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected</p>	RW	0b0

### 8.3.5 Configuration register summary

The following table describes the registers for the relevant component.

Table 8-238: por\_cfgm\_cfg register summary

Offset	Name	Type	Description
0x0	por_cfgm_node_info	RO	por_cfgm_node_info
0x8	por_cfgm_periph_id_0_periph_id_1	RO	por_cfgm_periph_id_0_periph_id_1
0x10	por_cfgm_periph_id_2_periph_id_3	RO	por_cfgm_periph_id_2_periph_id_3
0x18	por_cfgm_periph_id_4_periph_id_5	RO	por_cfgm_periph_id_4_periph_id_5
0x20	por_cfgm_periph_id_6_periph_id_7	RO	por_cfgm_periph_id_6_periph_id_7
0x28	por_cfgm_component_id_0_component_id_1	RO	por_cfgm_component_id_0_component_id_1
0x30	por_cfgm_component_id_2_component_id_3	RO	por_cfgm_component_id_2_component_id_3
0x80	por_cfgm_child_info	RO	por_cfgm_child_info
0x980	por_cfgm_secure_access	RW	por_cfgm_secure_access
0x988	por_cfgm_rcr	RW	por_cfgm_rcr

Offset	Name	Type	Description
0x990	por_cfgm_scr	RW	por_cfgm_scr
0xA00	por_cfgm_ras_mode	RW	por_cfgm_ras_mode
0x4000 : 0x10D76	por_errfr_0-671%56_sra_0-671/56	RO	por_errfr_0-671%56_sra_0-671/56
0x4010 : 0x10D86	por_errstatus_0-671%56_sra_0-671/56	W1C	por_errstatus_0-671%56_sra_0-671/56
0x4E00 : 0xFE00	por_errgsr_sra_0-11	RO	por_errgsr_sra_0-11
0x4E10 : 0xFE10	por_errildr_sra_0-11	RO	por_errildr_sra_0-11
0x4FA8 : 0xFFA8	por_errdevaff_sra_0-11	RO	por_errdevaff_sra_0-11
0x4FB8 : 0xFFB8	por_errdevarch_sra_0-11	RO	por_errdevarch_sra_0-11
0x4FC8 : 0xFFC8	por_errdevid_sra_0-11	RO	por_errdevid_sra_0-11
0x4FD0 : 0xFFFFD0	por_errpidr45_sra_0-11	RO	por_errpidr45_sra_0-11
0x4FE0 : 0xFFFFE0	por_errpidr01_sra_0-11	RO	por_errpidr01_sra_0-11
0x4FE8 : 0xFFFFE8	por_errpidr23_sra_0-11	RO	por_errpidr23_sra_0-11
0x4FF0 : 0xFFFFF0	por_errcidr01_sra_0-11	RO	por_errcidr01_sra_0-11
0x4FF8 : 0xFFFFF8	por_errcidr23_sra_0-11	RO	por_errcidr23_sra_0-11
0x900	por_info_global	RO	por_info_global
0x908	por_info_global_1	RO	por_info_global_1
0x1900	por_ppu_qactive_hyst	RW	por_ppu_qactive_hyst
{0-15} 0x1980 : 0x19F8	por_ppu_int_enable0-15	RW	por_ppu_int_enable0-15
{0-15} 0x1A00 : 0x1A78	por_ppu_int_status0-15	W1C	por_ppu_int_status0-15
{0-15} 0x1A80 : 0x1AF8	por_mpam_s_err_int_status0-15	W1C	por_mpam_s_err_int_status0-15
{0-15} 0x1B00 : 0x1B78	por_mpam_ns_err_int_status0-15	W1C	por_mpam_ns_err_int_status0-15
{0-15} 0x1B80 : 0x1BF8	por_mpam_rt_err_int_status0-15	W1C	por_mpam_rt_err_int_status0-15
{0-15} 0x1C00 : 0x1C78	por_mpam_rl_err_int_status0-15	W1C	por_mpam_rl_err_int_status0-15
0x100 : 0x8F8	por_cfgm_child_pointer_0-255	RO	por_cfgm_child_pointer_0-255

### 8.3.5.1 por\_cfgm\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

### Reset value

See individual bit resets

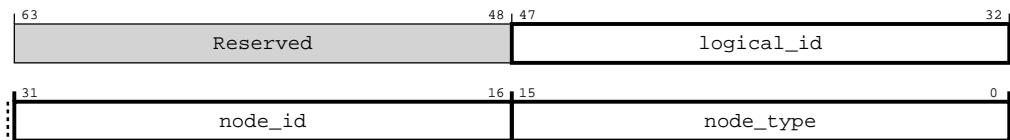
### Usage constraints

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 8-237: por\_cfgm\_node\_info**



**Table 8-239: por\_cfgm\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0002

### 8.3.5.2 por\_cfgm\_periph\_id\_0\_periph\_id\_1

Functions as the peripheral ID 0 and peripheral ID 1 register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x8

#### Type

RO

#### Reset value

See individual bit resets

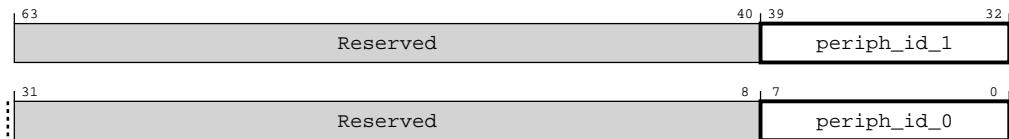
## Usage constraints

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 8-238: por\_cfgm\_periph\_id\_0\_periph\_id\_1**



**Table 8-240: por\_cfgm\_periph\_id\_0\_periph\_id\_1 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_1	Peripheral ID 1	RO	0b10110100
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_0	Peripheral ID 0	RO	0x3E

## 8.3.5.3 por\_cfgm\_periph\_id\_2\_periph\_id\_3

Functions as the peripheral ID 2 and peripheral ID 3 register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x10

#### Type

RO

#### Reset value

See individual bit resets

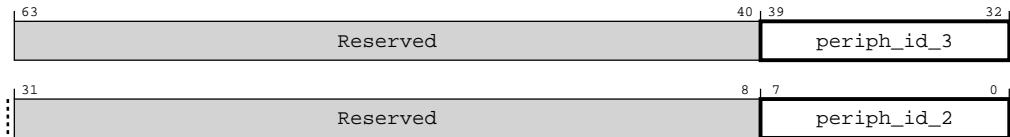
### Usage constraints

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 8-239: por\_cfgm\_periph\_id\_2\_periph\_id\_3**



**Table 8-241: por\_cfgm\_periph\_id\_2\_periph\_id\_3 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_3	Peripheral ID 3	RO	0b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_2	Peripheral ID 2[7:4] indicates revision:  0x0 r0p0 0x1 r0p1 0x2 r1p0 0x3 r2p0[3] JEDEC JEP106 identity code, 0b1 [2:0] JEP106 identity code [6:4], 0b011	RO	0x3B

## 8.3.5.4 por\_cfgm\_periph\_id\_4\_periph\_id\_5

Functions as the peripheral ID 4 and peripheral ID 5 register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x18

#### Type

RO

#### Reset value

See individual bit resets

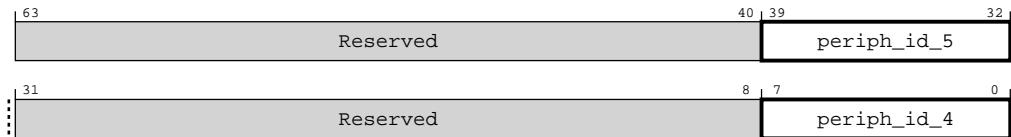
#### Usage constraints

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 8-240: por\_cfgm\_periph\_id\_4\_periph\_id\_5**



**Table 8-242: por\_cfgm\_periph\_id\_4\_periph\_id\_5 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_5	Peripheral ID 5	RO	0b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_4	Peripheral ID 4	RO	0b11000100

### 8.3.5.5 por\_cfgm\_periph\_id\_6\_periph\_id\_7

Functions as the peripheral ID 6 and peripheral ID 7 register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x20

##### Type

RO

##### Reset value

See individual bit resets

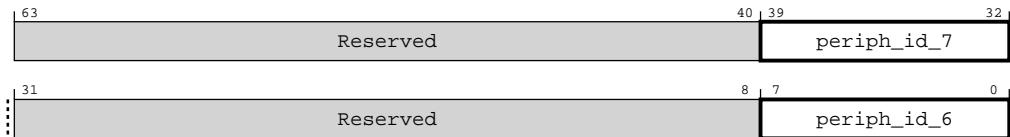
##### Usage constraints

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 8-241: por\_cfgm\_periph\_id\_6\_periph\_id\_7**



**Table 8-243: por\_cfgm\_periph\_id\_6\_periph\_id\_7 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	periph_id_7	Peripheral ID 7	RO	0b0
[31:8]	Reserved	Reserved	RO	
[7:0]	periph_id_6	Peripheral ID 6	RO	0b0

### 8.3.5.6 por\_cfgm\_component\_id\_0\_component\_id\_1

Functions as the component ID 0 and component ID 1 register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x28

##### Type

RO

##### Reset value

See individual bit resets

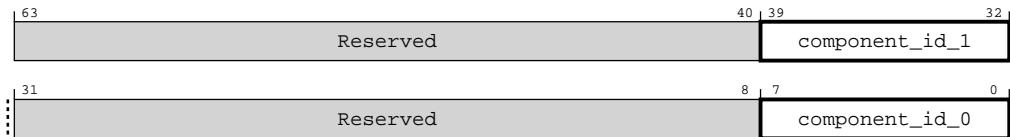
##### Usage constraints

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 8-242: por\_cfgm\_component\_id\_0\_component\_id\_1**



**Table 8-244: por\_cfgm\_component\_id\_0\_component\_id\_1 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	component_id_1	Component ID 1	RO	0b11110000
[31:8]	Reserved	Reserved	RO	-
[7:0]	component_id_0	Component ID 0	RO	0b00001101

### 8.3.5.7 por\_cfgm\_component\_id\_2\_component\_id\_3

Functions as the component ID 2 and component ID 3 register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x30

##### Type

RO

##### Reset value

See individual bit resets

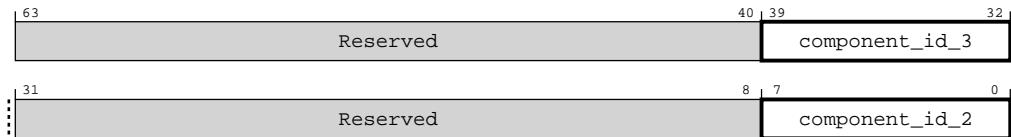
##### Usage constraints

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 8-243: por\_cfgm\_component\_id\_2\_component\_id\_3**



**Table 8-245: por\_cfgm\_component\_id\_2\_component\_id\_3 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	component_id_3	Component ID 3	RO	0b10110001
[31:8]	Reserved	Reserved	RO	-
[7:0]	component_id_2	Component ID 2	RO	0b00000101

### 8.3.5.8 por\_cfgm\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

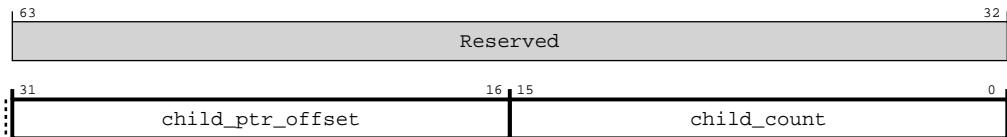
##### Usage constraints

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 8-244: por\_cfgm\_child\_info**



**Table 8-246: por\_cfgm\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x100
[15:0]	child_count	Number of child nodes; used in discovery process	RO	Configuration dependent

### 8.3.5.9 por\_cfgm\_secure\_access

Functions as the secure access control register. This register must be set up at boot time. Before initiating a write to this register, software must ensure that no other configuration accesses are in flight. Once this write is initiated, no other configuration accesses are initiated until complete.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

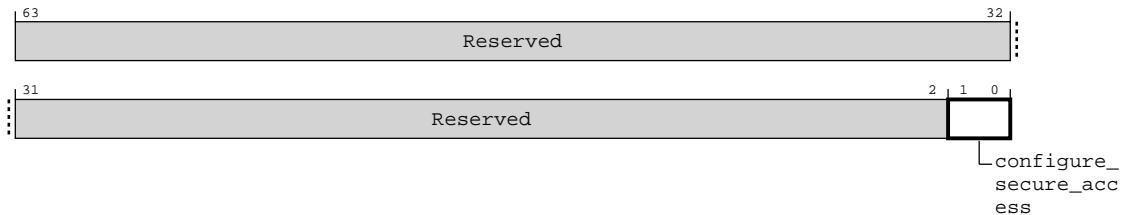
##### Usage constraints

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 8-245: por\_cfgm\_secure\_access**



**Table 8-247: por\_cfgm\_secure\_access attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	configure_secure_access	Secure access mode <b>0b00</b> Default operation <b>0b01</b> Allows non-secure access to secure registers <b>0b10</b> Allows secure access only to any configuration register regardless of its security status <b>0b11</b> Undefined behavior	RW	0b0

### 8.3.5.10 por\_cfgm\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

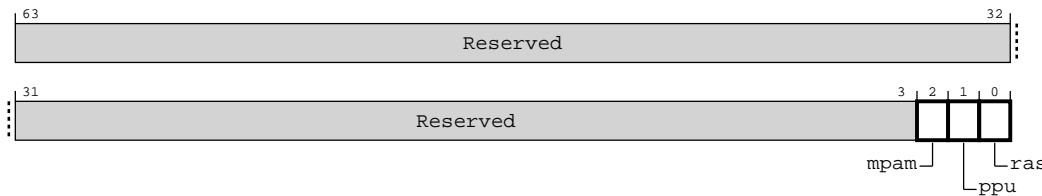
## Usage constraints

This register is owned in the Root space and is 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 8-246: por\_cfgm\_rcr**



**Table 8-248: por\_cfgm\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	mpam	Allows Root override of the MPAM registers	RW	0b0
[1]	ppu	Allows Root override of the PPU registers	RW	0b0
[0]	ras	Allows Root override of the RAS registers	RW	0b0

### 8.3.5.11 por\_cfgm\_scr

Secure register access override.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x990

### Type

RW

### Reset value

See individual bit resets

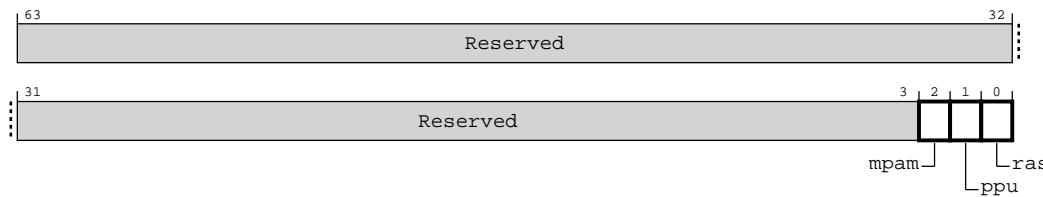
## Usage constraints

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 8-247: por\_cfgm\_scr**



**Table 8-249: por\_cfgm\_scr attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	mpam	Allows Secure override of the MPAM registers	RW	0b0
[1]	ppu	Allows Secure override of the PPU registers	RW	0b0
[0]	ras	Allows Secure override of the RAS registers	RW	0b0

### 8.3.5.12 por\_cfgm\_ras\_mode

Functions as the RAS mode control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_cfgm\_rcr.ras

## Secure group override

por\_cfgm\_scr.ras

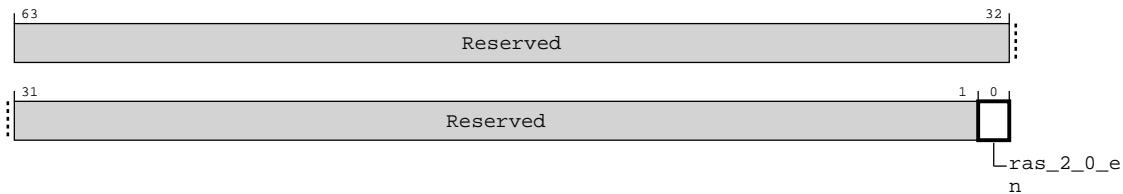
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_cfgm\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_cfgm\_scr.ras bit and por\_cfgm\_rcr.ras 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 8-248: por\_cfgm\_ras\_mode**



**Table 8-250: por\_cfgm\_ras\_mode attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	ras_2_0_en	RAS 2.0 enable  <b>0b0</b> CMN Default operation, not supporting RAS 2.0 System RAS Agents  <b>0b1</b> CMN support 12 System RAS Agents.	RW	0b0

## 8.3.5.13 por\_errfr\_0-671%56\_sra\_0-671/56

There are 672 iterations of this register. The index ranges from 0 to 671. Functions as the error feature register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

$0x4000 + \{0x1000 * (\text{index}/56)\} + \{0x40 * (\text{index}\%56)\}$

### Type

RO

### Reset value

See individual bit resets

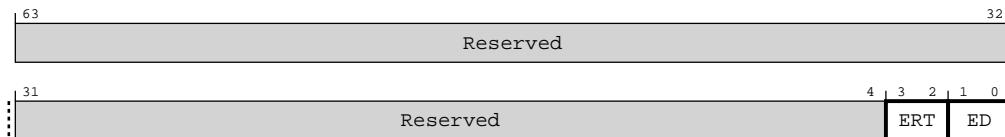
### Usage constraints

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 8-249: por\_errfr\_0-671%56\_sra\_0-671/56**



**Table 8-251: por\_errfr\_0-671%56\_sra\_0-671/56 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3:2]	ERT	Error record type <b>0b01</b> Error record <n> is a proxy for a RAS agent.	RO	0b01
[1:0]	ED	Error reporting and logging control <b>0b11</b> Error record <n> is not a true error record.	RO	0b11

### 8.3.5.14 por\_errstatus\_0-671%56\_sra\_0-671/56

There are 672 iterations of this register. The index ranges from 0 to 671. Functions as the error status register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

$0x4000 + \{0x1000 * (\text{index}/56)\} + \{0x40 * (\text{index}\%56)\} + \{0x10\}$

## Type

W1C

## Reset value

See individual bit resets

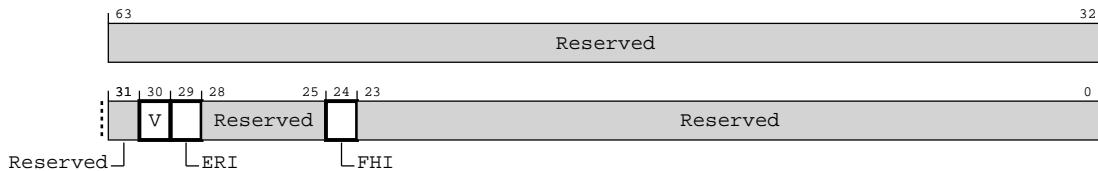
## Usage constraints

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 8-250: por\_errstatus\_0-671%56\_sra\_0-671/56**



**Table 8-252: por\_errstatus\_0-671%56\_sra\_0-671/56 attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	
[30]	V	RAS agent sticky valid status.  <b>0b0</b> RAS agent error status has not been asserted.  <b>0b1</b> RAS agent error status asserted.	W1C	0b0
[29]	ERI	Error Recovery Interrupt sticky status.  <b>0b0</b> RAS agent error recovery interrupt has not been asserted.  <b>0b1</b> RAS agent error recovery interrupt asserted.	W1C	0b0
[28:25]	Reserved	Reserved	RO	
[24]	FHI	Fault Handling Interrupt sticky status.  <b>0b0</b> RAS agent fault handling interrupt has not been asserted.  <b>0b1</b> RAS agent fault handling interrupt asserted.	W1C	0b0
[23:0]	Reserved	Reserved	RO	

### 8.3.5.15 por\_errgsr\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. Functions as Error Group Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

$0x4E00 + \{0x1000 * \text{index}\}$

##### Type

RO

##### Reset value

See individual bit resets

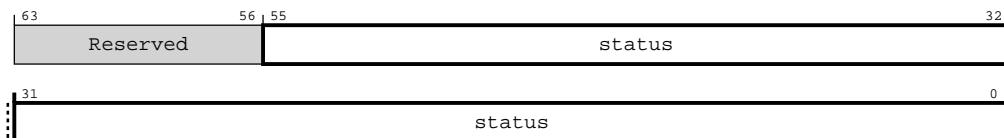
##### Usage constraints

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 8-251: por\_errgsr\_sra\_0-11**



**Table 8-253: por\_errgsr\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:56]	Reserved	Reserved	RO	
[55:0]	status	Read-only copy of ERRSTATUS_<N>.V	RO	0b0

### 8.3.5.16 por\_errildr\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. Functions as the implementation identification register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

$0x4E10 + \{0x1000 * \text{index}\}$

##### Type

RO

##### Reset value

See individual bit resets

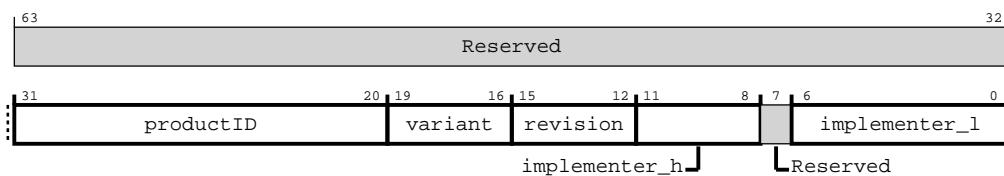
##### Usage constraints

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 8-252: por\_errildr\_sra\_0-11**



**Table 8-254: por\_errildr\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.5.17 por\_errdevaff\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. Functions as the device affinity register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

$0x4FA8 + \{0x1000 * \text{index}\}$

##### Type

RO

##### Reset value

See individual bit resets

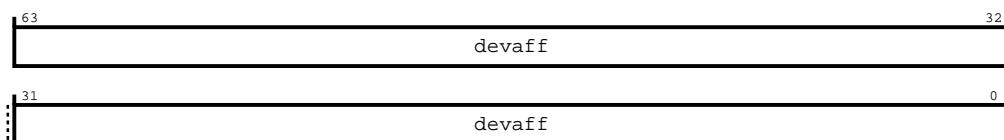
##### Usage constraints

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 8-253: por\_errdevaff\_sra\_0-11**



**Table 8-255: por\_errdevaff\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.5.18 por\_errdevarch\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. Functions as the device architecture register.

#### Configurations

This register is available in all configurations.

## Attributes

## Width

64

## Address offset

`0x4FB8 + #{0x1000 * index}`

## Type

RO

Reset value

See individual bit resets

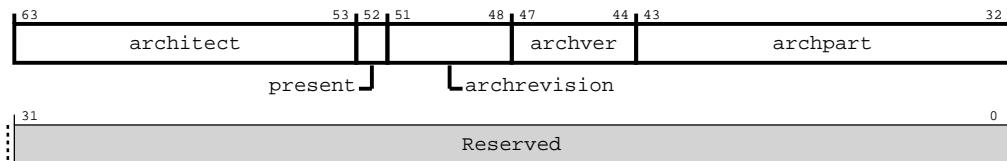
## Usage constraints

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 8-254:** por\_errdevarch\_sra\_0-11



**Table 8-256: por\_errdevarch\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

8.3.5.19 por errdevid sra 0-11

There are 12 iterations of this register. The index ranges from 0 to 11. Functions as the device configuration register

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

$0x4FC8 + \{0x1000 * \text{index}\}$

### Type

RO

### Reset value

See individual bit resets

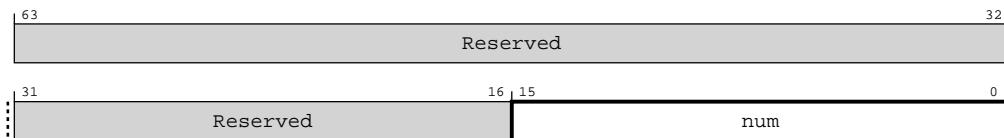
### Usage constraints

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 8-255: por\_errdevid\_sra\_0-11**



**Table 8-257: por\_errdevid\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

## 8.3.5.20 por\_errpidr45\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. 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

$0x4FD0 + \{0x1000 * \text{index}\}$

### Type

RO

### Reset value

See individual bit resets

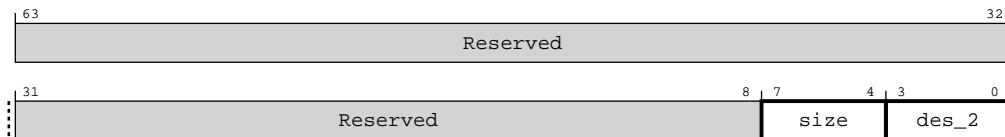
### Usage constraints

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 8-256: por\_errpidr45\_sra\_0-11**



**Table 8-258: por\_errpidr45\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

### 8.3.5.21 por\_errpidr01\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. 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

$0x4FE0 + \{0x1000 * \text{index}\}$

#### Type

RO

#### Reset value

See individual bit resets

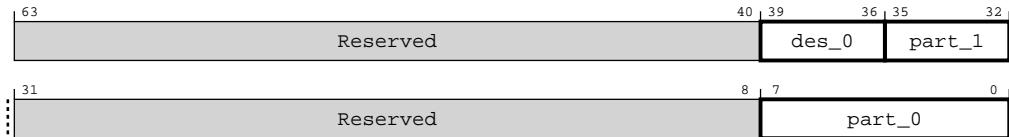
## Usage constraints

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 8-257: por\_errpidr01\_sra\_0-11**



**Table 8-259: por\_errpidr01\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

## 8.3.5.22 por\_errpidr23\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. 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

0x4FE8 + #{0x1000 \* index}

#### Type

RO

#### Reset value

See individual bit resets

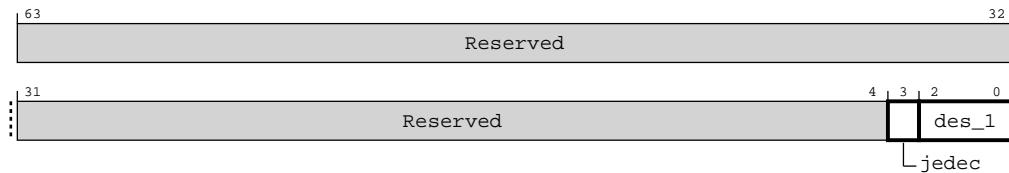
## Usage constraints

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 8-258: por\_errpidr23\_sra\_0-11**



**Table 8-260: por\_errpidr23\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

## 8.3.5.23 por\_errcidr01\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. 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

$0x4FF0 + \{0x1000 * \text{index}\}$

### Type

RO

### Reset value

See individual bit resets

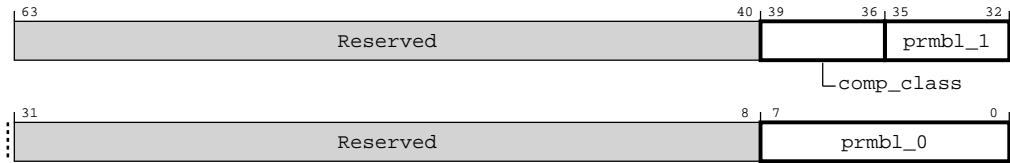
## Usage constraints

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 8-259: por\_errcidr01\_sra\_0-11**



**Table 8-261: por\_errcidr01\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmb1_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_0	PRMBL_0	RO	0xD

### 8.3.5.24 por\_errcidr23\_sra\_0-11

There are 12 iterations of this register. The index ranges from 0 to 11. 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

$0x4FF8 + \{0x1000 * \text{index}\}$

#### Type

RO

#### Reset value

See individual bit resets

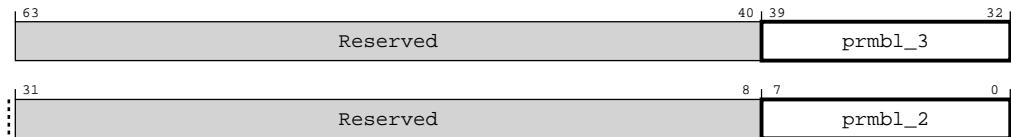
#### Usage constraints

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 8-260: por\_errcidr23\_sra\_0-11**



**Table 8-262: por\_errcidr23\_sra\_0-11 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.5.25 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

0x900

##### Type

RO

##### Reset value

See individual bit resets

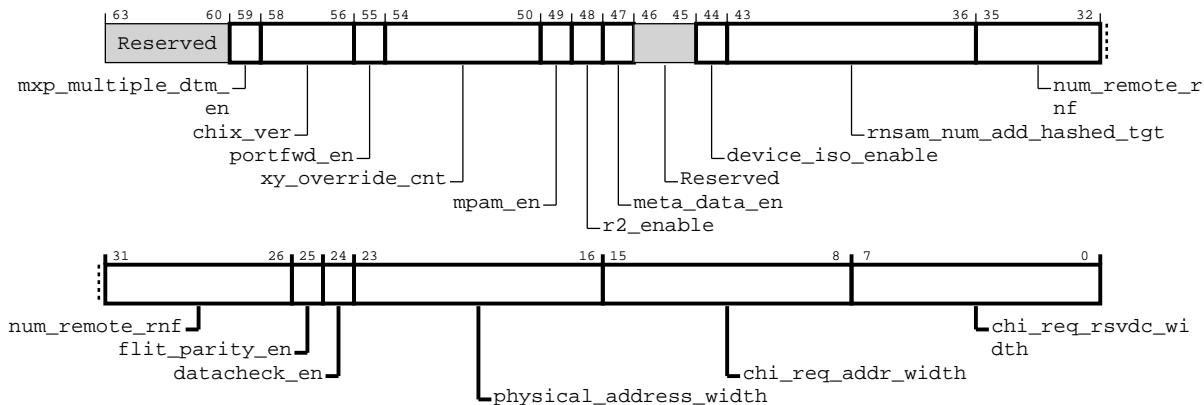
##### Usage constraints

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 8-261: por\_info\_global**



**Table 8-263: por\_info\_global attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59]	mxp_multiple_dtm_en	Multiple DTM feature enable. This is used if number of device ports on the XP is > 2	RO	Configuration dependent
[58:56]	chix_ver	CHIX Version Parameter: 2 -> CHIB 3 -> CHIC 4 -> CHID 5 -> CHIE	RO	0x4
[55]	portfwd_en	CCIX Port to Port Forwarding feature enable	RO	Configuration dependent
[54:50]	xy_override_cnt	Number of Src-Tgt pairs whose XY route path can be overridden	RO	Configuration dependent
[49]	mpam_en	MPAM enable	RO	Configuration dependent
[48]	r2_enable	CMN R2 feature enable	RO	0x0
[47]	meta_data_en	Meta Data Preservation mode enable	RO	0x0
[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

### 8.3.5.26 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

0x908

##### Type

RO

##### Reset value

See individual bit resets

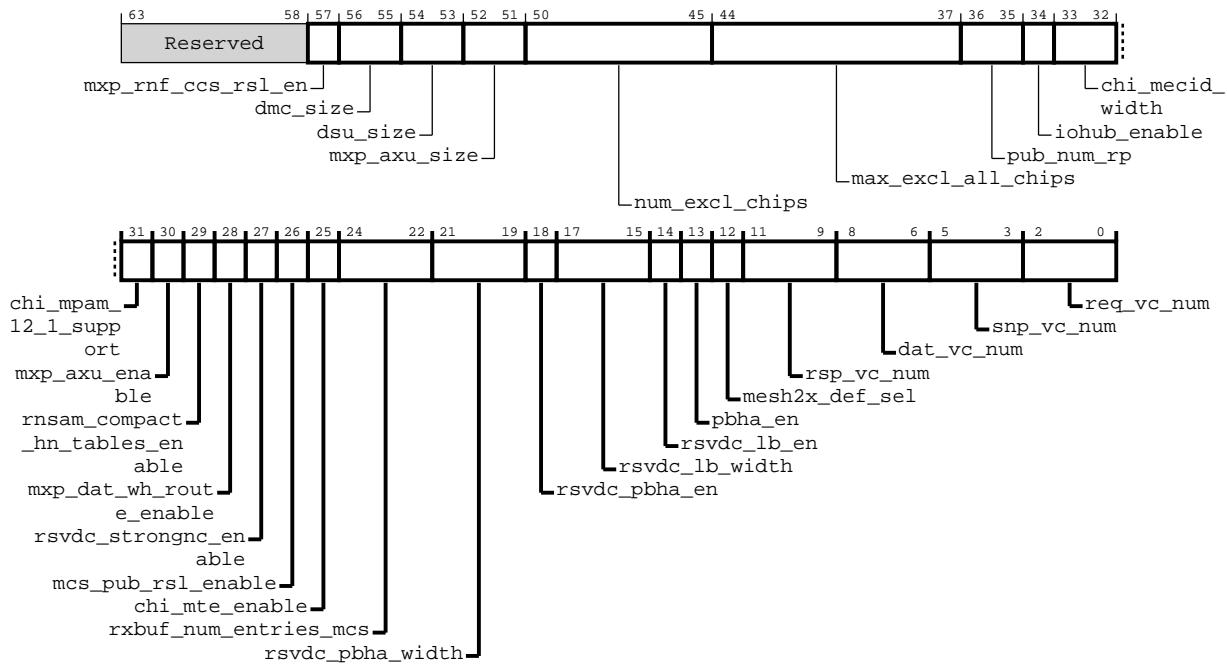
#### Usage constraints

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 8-262: por\_info\_global\_1**



**Table 8-264: por\_info\_global\_1 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57]	mxp_rnf_ccs_rsl_en	Add register slice on output of CCS for interface to MXP and CAL RN-F	RO	Configuration dependent
[56:55]	dmc_size	DMC address space <b>0b00</b> 1MB; <b>0b01</b> 4MB; <b>0b10</b> 16MB; <b>0b11</b> 64MB.	RO	Configuration dependent
[54:53]	dsu_size	DSU address space. <b>0b00</b> 1MB; <b>0b01</b> 4MB; <b>0b10</b> 16MB; <b>0b11</b> 64MB.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[52:51]	mxp_axu_size	MXP AXU address space.  <b>0b00</b> 1MB;  <b>0b01</b> 4MB;  <b>0b10</b> 16MB;  <b>0b11</b> 64MB.	RO	Configuration dependent
[50:45]	num_excl_chips	Number of SMP chips doing exclusives	RO	Configuration dependent
[44:37]	max_excl_all_chips	Maximum of the number of RNFs doing exclusives across all chips	RO	Configuration dependent
[36:35]	pub_num_rp	Enable IO HUB mesh	RO	Configuration dependent
[34]	iohub_enable	Enable IO HUB mesh	RO	Configuration dependent
[33:32]	chi_mecid_width	Width of CHI MECID field  <b>0b00</b> Disabled  <b>0b01</b> 12 bits  <b>0b10</b> 16 bits	RO	Configuration dependent
[31]	chi_mpam_12_1_support	CHI-G MPAM support property  <b>0b0</b> MPAM_9_1 mode (PARTID width is 9 bits, PMG width is 1 bit)  <b>0b1</b> MPAM_12_1 mode (PARTID width is 12 bits, PMG width is 1 bit)	RO	Configuration dependent
[30]	mxp_axu_enable	MXP AXU interface Enable	RO	Configuration dependent
[29]	rnsam_compact_hn_tables_enable	RNSAM Compact HN Tables Enable	RO	Configuration dependent
[28]	mxp_dat_wh_route_enable	Worm Hole Routing Enable for MXP DAT channel	RO	Configuration dependent
[27]	rsvdc_strongnc_enable	RSVDC StrongNC Mode Enable	RO	Configuration dependent
[26]	mcs_pub_rsl_enable	Register Slice enable for MCS PUB outputs	RO	Configuration dependent
[25]	chi_mte_enable	CHI MTE Feature Enable	RO	Configuration dependent
[24:22]	rxbuf_num_entries_mcs	RX Buffer Entries at upload interface of MCSX/MCSY	RO	Configuration dependent
[21:19]	rsvdc_pbha_width	RSVDC PBHA Field Width	RO	0x2

Bits	Name	Description	Type	Reset
[18]	rsvdc_pbha_en	RSVDC PBHA Mode Enable	RO	0x0
[17:15]	rsvdc_lb_width	RSVDC Loop Back Field Width	RO	Configuration dependent
[14]	rsvdc_lb_en	RSVDC Loop Back Mode Enable	RO	0x0
[13]	pbha_en	PBHA Mode Enable	RO	0x0
[12]	mesh2x_def_sel	Default ping-pong scheme selection for TGTID Lookup in 2xMESH	RO	Configuration dependent
[11:9]	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
[8:6]	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
[5:3]	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
[2:0]	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

### 8.3.5.27 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

0x1900

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_cfgm\_rcr.ppu

##### Secure group override

por\_cfgm\_scr.ppu

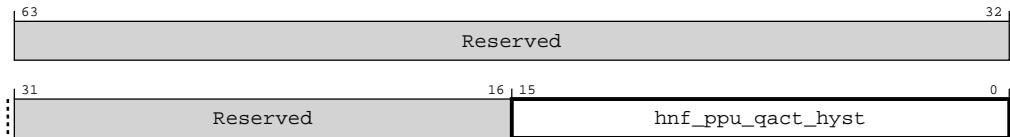
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_cfgm\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the por\_cfgm\_scr.ppu bit and por\_cfgm\_rcr.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 8-263: por\_ppu\_qactive\_hyst**



**Table 8-265: por\_ppu\_qactive\_hyst attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ppu_qact_hyst	QACTIVE hysteresis	RW	0x10

### 8.3.5.28 por\_ppu\_int\_enable0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the HN-F PPU event interrupt. Contains the interrupt mask.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-15) : 0x1980 + #8 \* index

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_cfgm\_rcr.ppu

### Secure group override

por\_cfgm\_scr.ppu

### Usage constraints

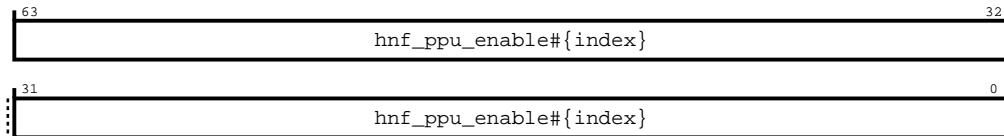
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_cfgm\_scr.ppu bit is set, Secure accesses to this

register are permitted. If both the por\_cfgm\_scr.ppu bit and por\_cfgm\_rcr.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 8-264: por\_ppu\_int\_enable0-15**



**Table 8-266: por\_ppu\_int\_enable0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>hnf_ppu_enable#{index}</code>	Interrupt mask	RW	0b0

## 8.3.5.29 por\_ppu\_int\_status0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Provides HN-F PPU event interrupt status.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

`index(0-15) : 0x1A00 + #{8 * index}`

### Type

W1C

### Reset value

See individual bit resets

### Root group override

`por_cfgm_rcr.ppu`

### Secure group override

`por_cfgm_scr.ppu`

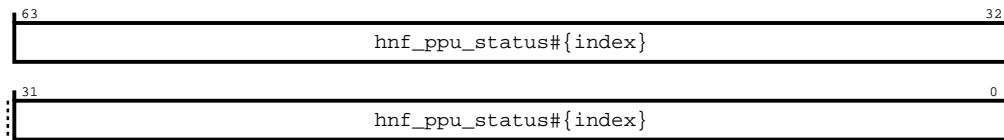
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_cfgm\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the por\_cfgm\_scr.ppu bit and por\_cfgm\_rcr.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 8-265: por\_ppu\_int\_status0-15**



**Table 8-267: por\_ppu\_int\_status0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status#{index}	Interrupt status	W1C	0b0

## 8.3.5.30 por\_mpam\_s\_err\_int\_status0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Provides HN-F MPAM Secure Error interrupt status.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

index(0-15) : 0x1A80 + #{8 \* index}

#### Type

W1C

#### Reset value

See individual bit resets

#### Root group override

por\_cfgm\_rcr.mpam

### Secure group override

por\_cfgm\_scr.mpam

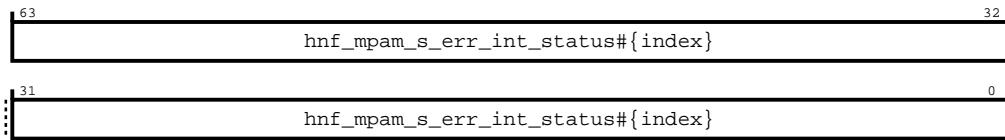
### Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-266: por\_mpam\_s\_err\_int\_status0-15**



**Table 8-268: por\_mpam\_s\_err\_int\_status0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status#{index}	MPAM S Interrupt status	W1C	0b0

### 8.3.5.31 por\_mpam\_ns\_err\_int\_status0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Provides HN-F MPAM Non-Secure Error interrupt status.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

index(0-15) : 0x1B00 + #{8 \* index}

#### Type

W1C

#### Reset value

See individual bit resets

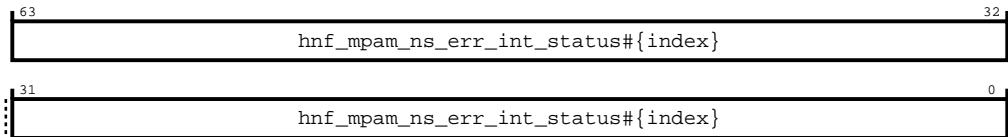
### Usage constraints

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 8-267: por\_mpam\_ns\_err\_int\_status0-15**



**Table 8-269: por\_mpam\_ns\_err\_int\_status0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status#{index}	MPAM NS Interrupt status	W1C	0b0

### 8.3.5.32 por\_mpam\_rt\_err\_int\_status0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Provides HN-F MPAM Root Error interrupt status.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-15) : 0x1B80 + #{8 \* index}

### Type

W1C

### Reset value

See individual bit resets

### Root group override

por\_cfgm\_rcr.mpam

### Secure group override

por\_cfgm\_scr.mpam

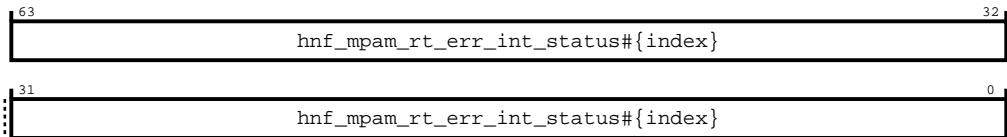
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_cfgm\_scr.mpam bit is set, Secure accesses to this register are permitted. If both the por\_cfgm\_scr.mpam bit and por\_cfgm\_rcr.mpam 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 8-268: por\_mpam\_rt\_err\_int\_status0-15**



**Table 8-270: por\_mpam\_rt\_err\_int\_status0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>hnf_mpam_rt_err_int_status#{index}</code>	MPAM RT Interrupt status	W1C	0b0

### 8.3.5.33 por\_mpam\_rl\_err\_int\_status0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Provides HN-F MPAM Realm Error interrupt status.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

`index(0-15) : 0x1C00 + #{8 * index}`

### Type

W1C

### Reset value

See individual bit resets

### Root group override

`por_cfgm_rcr.mpam`

### Secure group override

`por_cfgm_scr.mpam`

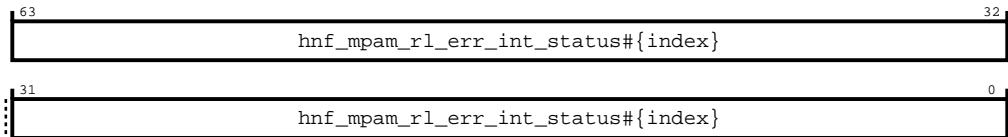
### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-269: por\_mpam\_rl\_err\_int\_status0-15**



**Table 8-271: por\_mpam\_rl\_err\_int\_status0-15 attributes**

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_rl_err_int_status#{index}	MPAM RL Interrupt status	W1C	0b0

## 8.3.5.34 por\_cfgm\_child\_pointer\_0-255

There are 256 iterations of this register. The index ranges from 0 to 255. Contains base address of child configuration node. NOTE: There will be as many child pointer registers in the Global Config Unit as the number of XP's on the chip. Each successive child pointer register will be at the next 8 byte address boundary. Each successive child pointer register will be named with the suffix corresponding to the register number. For example por\_cfgm\_child\_pointer\_<0:255>

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x100 + #{8 \* index}

### Type

RO

### Reset value

See individual bit resets

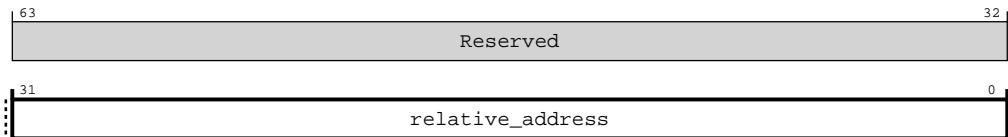
## Usage constraints

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 8-270: por\_cfgm\_child\_pointer\_0-255**



**Table 8-272: por\_cfgm\_child\_pointer\_0-255 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	relative_address	<p>Bit 31: External or internal child node <b>0b1</b>: Indicates child pointer points to a configuration node that is external to CMN <b>0b0</b>: Indicates child pointer points to a configuration node that is internal to CMN</p> <p>Bits [30]: Set to 0b0</p> <p>Bits [29:0]: Child node address offset relative to PERIPHBASE</p>	RO	0b0

### 8.3.6 CXLAPB register summary

The following table describes the registers for the relevant component.

**Table 8-273: por\_cxlapb\_cfg register summary**

Offset	Name	Type	Description
0x1110	por_cxlapb_link_rx_credit_ctl	RW	<a href="#">por_cxlapb_link_rx_credit_ctl</a>
0x1118	por_cxlapb_link_rx_credit_return_stat	RO	<a href="#">por_cxlapb_link_rx_credit_return_stat</a>
0x1120	por_cxlapb_link_tx_credit_stat	RO	<a href="#">por_cxlapb_link_tx_credit_stat</a>
0x1060	por_cxlapb_cxl_security_policy	RW	<a href="#">por_cxlapb_cxl_security_policy</a>
0x1200	por_cxlapb_cxl_hdm_decoder_capability	RO	<a href="#">por_cxlapb_cxl_hdm_decoder_capability</a>
0x1204	por_cxlapb_cxl_hdm_decoder_global_control	RW	<a href="#">por_cxlapb_cxl_hdm_decoder_global_control</a>
0x1210 : 0x12F0	por_cxlapb_cxl_hdm_decoder_0-7_base_low	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_base_low</a>
0x1214 : 0x12F4	por_cxlapb_cxl_hdm_decoder_0-7_base_high	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_base_high</a>
0x1218 : 0x12F8	por_cxlapb_cxl_hdm_decoder_0-7_size_low	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_size_low</a>

Offset	Name	Type	Description
0x121C : 0x12FC	por_cxlapb_cxl_hdm_decoder_0-7_size_high	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_size_high</a>
0x1220 : 0x1300	por_cxlapb_cxl_hdm_decoder_0-7_control	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_control</a>
0x1224 : 0x1304	por_cxlapb_cxl_hdm_decoder_0-7_dpa_skip_low	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_dpa_skip_low</a>
0x1228 : 0x1308	por_cxlapb_cxl_hdm_decoder_0-7_dpa_skip_high	RWL	<a href="#">por_cxlapb_cxl_hdm_decoder_0-7_dpa_skip_high</a>
0x1800	por_cxlapb_snoop_filter_group_id	RO	<a href="#">por_cxlapb_snoop_filter_group_id</a>
0x1804	por_cxlapb_snoop_filter_effective_size	RO	<a href="#">por_cxlapb_snoop_filter_effective_size</a>
0x120	por_cxlapb_dvsec_cxl_range_1_base_high	RWL	<a href="#">por_cxlapb_dvsec_cxl_range_1_base_high</a>
0x124	por_cxlapb_dvsec_cxl_range_1_base_low	RWL	<a href="#">por_cxlapb_dvsec_cxl_range_1_base_low</a>
0x130	por_cxlapb_dvsec_cxl_range_2_base_high	RWL	<a href="#">por_cxlapb_dvsec_cxl_range_2_base_high</a>
0x134	por_cxlapb_dvsec_cxl_range_2_base_low	RWL	<a href="#">por_cxlapb_dvsec_cxl_range_2_base_low</a>
0x118	por_cxlapb_dvsec_cxl_range_1_size_high	RO	<a href="#">por_cxlapb_dvsec_cxl_range_1_size_high</a>
0x11C	por_cxlapb_dvsec_cxl_range_1_size_low	RO	<a href="#">por_cxlapb_dvsec_cxl_range_1_size_low</a>
0x128	por_cxlapb_dvsec_cxl_range_2_size_high	RO	<a href="#">por_cxlapb_dvsec_cxl_range_2_size_high</a>
0x12C	por_cxlapb_dvsec_cxl_range_2_size_low	RO	<a href="#">por_cxlapb_dvsec_cxl_range_2_size_low</a>
0x10C	por_cxlapb_dvsec_cxl_control	RWL	<a href="#">por_cxlapb_dvsec_cxl_control</a>
0x110	por_cxlapb_dvsec_cxl_control2	RW	<a href="#">por_cxlapb_dvsec_cxl_control2</a>
0x114	por_cxlapb_dvsec_cxl_lock	RW	<a href="#">por_cxlapb_dvsec_cxl_lock</a>
0x20C	por_cxlapb_dvsec_flex_bus_port_control	RW	<a href="#">por_cxlapb_dvsec_flex_bus_port_control</a>
0x1014	por_cxlapb_err_capabilities_control	RW	<a href="#">por_cxlapb_err_capabilities_control</a>
0x1858	por_cxlapb_IDE_key_refresh_time_control	RW	<a href="#">por_cxlapb_IDE_key_refresh_time_control</a>
0x185C	por_cxlapb_IDE_truncation_transmit_delay_control	RW	<a href="#">por_cxlapb_IDE_truncation_transmit_delay_control</a>
0x0	por_cxlapb_ll_to_ull_msg	RW	<a href="#">por_cxlapb_ll_to_ull_msg</a>
0x1908	por_cxlapb_cxl_timeout_iso_ctl	RW	<a href="#">por_cxlapb_cxl_timeout_iso_ctl</a>
0x1900	por_cxlapb_cxl_timeout_iso_capability	RO	<a href="#">por_cxlapb_cxl_timeout_iso_capability</a>
0x190C	por_cxlapb_cxl_timeout_iso_status	RW	<a href="#">por_cxlapb_cxl_timeout_iso_status</a>
0x1980	por_cxlapb_CXL_Cache_ID_Decoder_Capability	RO	<a href="#">por_cxlapb_CXL_Cache_ID_Decoder_Capability</a>
0x1984	por_cxlapb_CXL_Cache_ID_Decoder_Control	RW	<a href="#">por_cxlapb_CXL_Cache_ID_Decoder_Control</a>
0x1988	por_cxlapb_CXL_Cache_ID_Decoder_Status	RO	<a href="#">por_cxlapb_CXL_Cache_ID_Decoder_Status</a>
0x1130	por_cxlapb_link_layer_defeature	RW	<a href="#">por_cxlapb_link_layer_defeature</a>
0x2000	por_c2capb_c2c_port_register_offset_header	RO	<a href="#">por_c2capb_c2c_port_register_offset_header</a>
0x2008	por_c2capb_c2c_port_register_offset_table_0	RO	<a href="#">por_c2capb_c2c_port_register_offset_table_0</a>
0x2010	por_c2capb_c2c_port_register_offset_table_1	RO	<a href="#">por_c2capb_c2c_port_register_offset_table_1</a>
0x2020	por_c2capb_c2c_port_capabilities_and_control_header	RO	<a href="#">por_c2capb_c2c_port_capabilities_and_control_header</a>
0x2028	por_c2capb_c2c_port_capabilities_1	RO	<a href="#">por_c2capb_c2c_port_capabilities_1</a>
0x2030	por_c2capb_c2c_port_capabilities_2	RO	<a href="#">por_c2capb_c2c_port_capabilities_2</a>

Offset	Name	Type	Description
0x2040	por_c2capb_c2c_port_control_and_status	RW	<a href="#">por_c2capb_c2c_port_control_and_status</a>
0x2060	por_c2capb_c2c_port_ingressid_route_table_header	RO	<a href="#">por_c2capb_c2c_port_ingressid_route_table_header</a>
0x2068	por_c2capb_c2c_port_ingressid_route_table_capabilities	RO	<a href="#">por_c2capb_c2c_port_ingressid_route_table_capabilities</a>
0x2080	por_c2capb_c2c_port_ingressid_route_table_control_and_status	RW	<a href="#">por_c2capb_c2c_port_ingressid_route_table_control_and_status</a>
0x20A0	por_c2capb_c2c_port_ingressid_route_entry_0	RW	<a href="#">por_c2capb_c2c_port_ingressid_route_entry_0</a>
0x20B0	por_c2capb_c2c_port_egressid_route_table_header	RO	<a href="#">por_c2capb_c2c_port_egressid_route_table_header</a>
0x20B8	por_c2capb_c2c_port_egressid_route_table_capabilities	RO	<a href="#">por_c2capb_c2c_port_egressid_route_table_capabilities</a>
0x20D0	por_c2capb_c2c_port_egressid_route_table_control_and_status	RW	<a href="#">por_c2capb_c2c_port_egressid_route_table_control_and_status</a>
0x20F0	por_c2capb_c2c_port_egressid_route_entry_0	RW	<a href="#">por_c2capb_c2c_port_egressid_route_entry_0</a>
0x2200	por_c2capb_c2c_logical_linkend_common_header	RO	<a href="#">por_c2capb_c2c_logical_linkend_common_header</a>
0x2220	por_c2capb_c2c_logical_linkend_offset_register	RO	<a href="#">por_c2capb_c2c_logical_linkend_offset_register</a>
0x2240	por_c2capb_c2c_logical_linkend_0-0_header	RO	<a href="#">por_c2capb_c2c_logical_linkend_0-0_header</a>
0x2258	por_c2capb_c2c_logical_linkend_0-0_control_and_status	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_control_and_status</a>
0x2268	por_c2capb_c2c_logical_linkend_0-0_map_table	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_map_table</a>
0x2280	por_c2capb_c2c_linkproperties_supported_uniform1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_supported_uniform1_0-0</a>
0x22C0	por_c2capb_c2c_linkproperties_advertised_uniform1_0-0	RW	<a href="#">por_c2capb_c2c_linkproperties_advertised_uniform1_0-0</a>
0x2300	por_c2capb_c2c_linkproperties_informed_uniform1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_informed_uniform1_0-0</a>
0x2340	por_c2capb_c2c_linkproperties_negotiated_uniform1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_negotiated_uniform1_0-0</a>
0x2290	por_c2capb_c2c_linkproperties_supported_rx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_supported_rx1_0-0</a>
0x22D0	por_c2capb_c2c_linkproperties_advertised_rx1_0-0	RW	<a href="#">por_c2capb_c2c_linkproperties_advertised_rx1_0-0</a>
0x2310	por_c2capb_c2c_linkproperties_informed_rx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_informed_rx1_0-0</a>
0x2350	por_c2capb_c2c_linkproperties_negotiated_rx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_negotiated_rx1_0-0</a>

Offset	Name	Type	Description
0x22A0 : 0x22A0	por_c2capb_c2c_linkproperties_supported_tx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_supported_tx1_0-0</a>
0x22E0 : 0x22E0	por_c2capb_c2c_linkproperties_advertised_tx1_0-0	RW	<a href="#">por_c2capb_c2c_linkproperties_advertised_tx1_0-0</a>
0x2320 : 0x2320	por_c2capb_c2c_linkproperties_informed_tx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_informed_tx1_0-0</a>
0x2360 : 0x2360	por_c2capb_c2c_linkproperties_negotiated_tx1_0-0	RO	<a href="#">por_c2capb_c2c_linkproperties_negotiated_tx1_0-0</a>
0x2388 : 0x2388	por_c2capb_c2c_logical_linkend_0-0_mc_credit1	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_mc_credit1</a>
0x2390 : 0x2390	por_c2capb_c2c_logical_linkend_0-0_mc_credit2	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_mc_credit2</a>
0x2398 : 0x2398	por_c2capb_c2c_logical_linkend_0-0_mc_credit3	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_mc_credit3</a>
0x23A0 : 0x23A0	por_c2capb_c2c_logical_linkend_0-0_mc_credit4	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_mc_credit4</a>
0x23A8 : 0x23A8	por_c2capb_c2c_logical_linkend_0-0_mc_credit5	RW	<a href="#">por_c2capb_c2c_logical_linkend_0-0_mc_credit5</a>

### 8.3.6.1 por\_cxlapb\_link\_rx\_credit\_ctl

CXL Link Rx Credit Control Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1110

##### Type

RW

##### Reset value

See individual bit resets

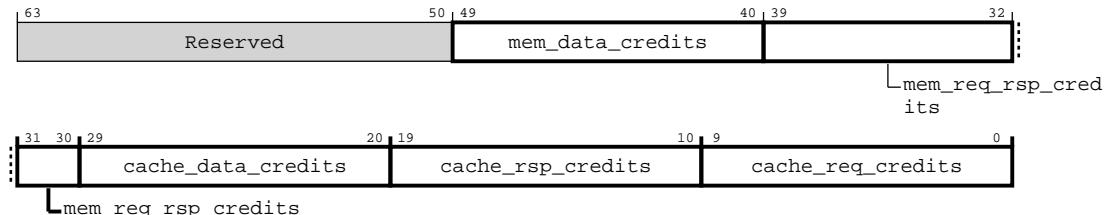
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-271: por\_cxlapb\_link\_rx\_credit\_ctl**



**Table 8-274: por\_cxlapb\_link\_rx\_credit\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Credits to advertise for Mem Data channel at init	RW	Configuration dependent
[39:30]	mem_req_rsp_credits	Credits to advertise for Mem Request or Response channel at init	RW	Configuration dependent
[29:20]	cache_data_credits	Credits to advertise for Cache Data channel at init	RW	0x100
[19:10]	cache_rsp_credits	Credits to advertise for Cache Response channel at init	RW	0x0
[9:0]	cache_req_credits	Credits to advertise for Cache Request channel at init	RW	Configuration dependent

## 8.3.6.2 por\_cxlapb\_link\_rx\_credit\_return\_stat

CXL Link Rx Credit Return Status Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x1118

#### Type

RO

#### Reset value

See individual bit resets

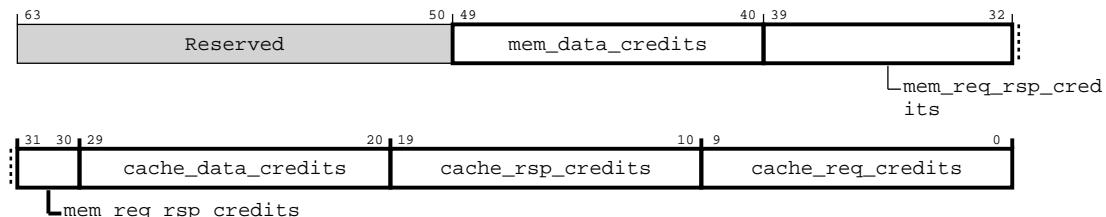
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-272: por\_cxlapb\_link\_rx\_credit\_return\_stat**



**Table 8-275: por\_cxlapb\_link\_rx\_credit\_return\_stat attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits to be returned	RO	0b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits to be returned	RO	0b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits to be returned	RO	0b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits to be returned	RO	0b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits to be returned	RO	0b0

## 8.3.6.3 por\_cxlapb\_link\_tx\_credit\_stat

CXL Link Tx Credit Status Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x1120

#### Type

RO

#### Reset value

See individual bit resets

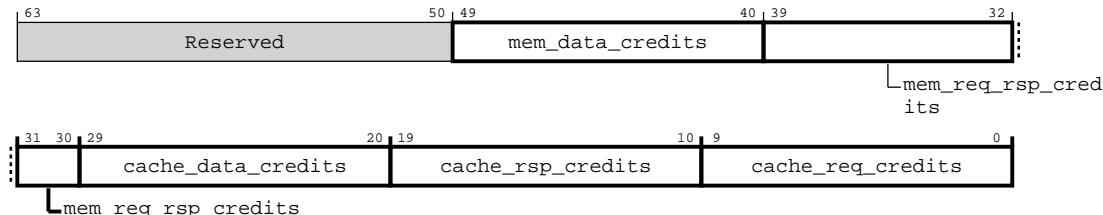
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-273: por\_cxlapb\_link\_tx\_credit\_stat**



**Table 8-276: por\_cxlapb\_link\_tx\_credit\_stat attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits for TX	RO	0b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits for TX	RO	0b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits for TX	RO	0b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits for TX	RO	0b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits for TX	RO	0b0

### 8.3.6.4 por\_cxlapb\_cxl\_security\_policy

Contains CXL Security Policy

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x1060

### Type

RW

### Reset value

See individual bit resets

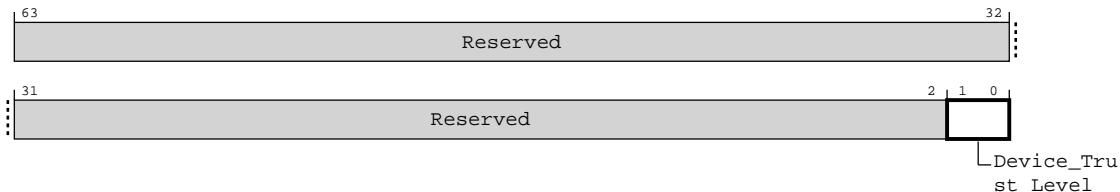
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-274: por\_cxlapb\_cxl\_security\_policy**



**Table 8-277: por\_cxlapb\_cxl\_security\_policy attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	Device_Trust_Level	0 -> Trusted CXL Device. At this setting a CXL Device will be able to get access on CXL.cache for both host-attached and device attached memory ranges. The Host can still protect security sensitive memory regions.  '1 -> Trusted for Device Attached Memory Range Only. At this setting a CXL Device will be able to get access on CXL.cache for device attached memory ranges only. Requests on CXL.cache for host-attached memory ranges will be aborted by the Host.  '2 -> Untrusted CXL Device. At this setting all requests on CXL.cache will be aborted by the Host. Please note that these settings only apply to requests on CXL.cache. The device can still source requests on CXL.io regardless of these settings. Protection on CXL.io will be implemented using IOMMU based page tables. Default value of this field is 2.	RW	0x2

## 8.3.6.5 por\_cxlapb\_cxl\_hdm\_decoder\_capability

Contains CXL\_HDM\_Decoder\_Capability\_Register. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x1200

### Type

RO

### Reset value

See individual bit resets

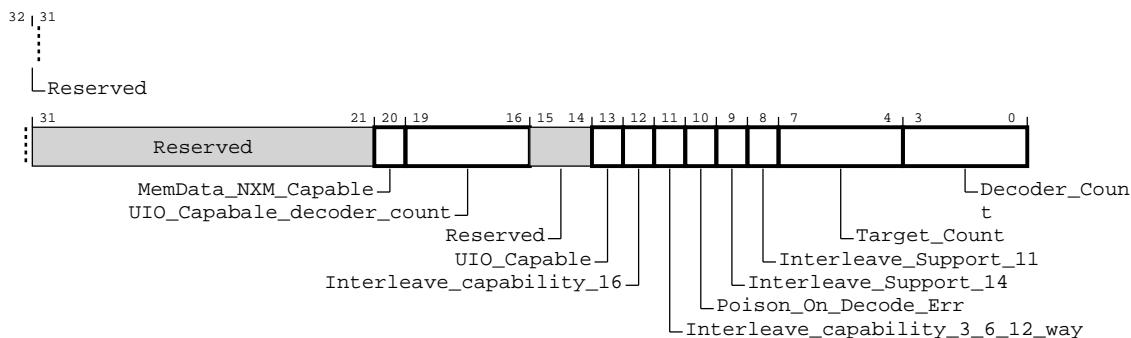
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-275: por\_cxlapb\_cxl\_hdm\_decoder\_capability**



**Table 8-278: por\_cxlapb\_cxl\_hdm\_decoder\_capability attributes**

Bits	Name	Description	Type	Reset
[31:21]	Reserved	Reserved	RO	
[20]	MemData_NXM_Capable	If set, the component supports MemData-NXM opcode. If cleared, the component does not support MemData-NXM opcode. All 256B Flit mode-capable components shall set this bit to 1.	RO	0b1
[19:16]	UIO_Capable_decoder_count	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x0
[15:14]	Reserved	Reserved	RO	
[13]	UIO_Capable	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x0
[12]	Interleave_capability_16	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1
[11]	Interleave_capability_3_6_12_way	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1
[10]	Poison_On_Decode_Err	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	0x1

Bits	Name	Description	Type	Reset
[9]	Interleave_Support_14	If set the component supports interleaving based on Address bit 14 Address bit 13 and Address bit 12. Root ports and switches shall always set this bit indicating support for interleaving based on Address bits 14-12.	RO	0x0
[8]	Interleave_Support_11	If set the component supports interleaving based on Address bit 11 Address bit 10 Address bit 9 and Address bit 8. Root Ports and Upstream Switch Ports shall always set this bit indicating support for interleaving based on Address bit 11-8.	RO	0x0
[7:4]	Target_Count	The number of target ports each decoder supports (applicable to Upstream Switch Port and Root Port only). Maximum of 8. 1: 1 target port, 2: 2 target ports, 4: 4 target ports, 8: 8 target ports, All other values are reserved	RO	0x1
[3:0]	Decoder_Count	Reports the number of memory address decoders implemented by the component. 0: 1 Decoder, 1: 2 Decoders, 2: 4 Decoders, 3: 6 Decoders, 4: 8 Decoders, 5: 10 Decoders, All other values are reserved	RO	0x4

### 8.3.6.6 por\_cxlapb\_cxl\_hdm\_decoder\_global\_control

Contains CXL\_HDM\_Decoder\_Global\_Control\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1204

##### Type

RW

##### Reset value

See individual bit resets

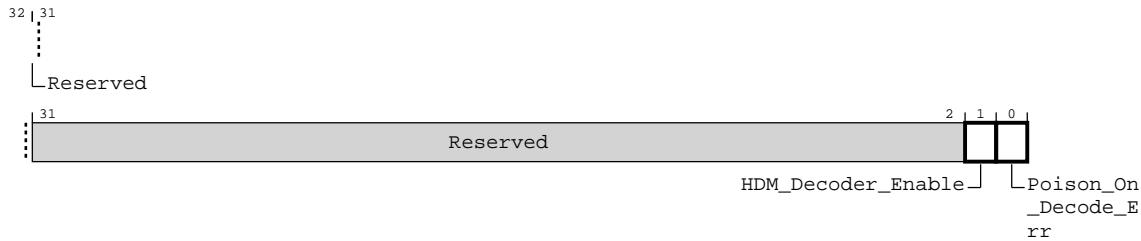
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-276: por\_cxlapb\_cxl\_hdm\_decoder\_global\_control**



**Table 8-279: por\_cxlapb\_cxl\_hdm\_decoder\_global\_attributes**

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Reserved	RO	
[1]	HDM_Decoder_Enable	This bit is only applicable to CXL.mem devices and shall return 0 on Root Ports and Upstream Switch Ports. When this bit is set device shall use HDM decoders to decode CXL.mem transactions and not use HDM Base registers in DVSEC ID 0. Root Ports and Upstream Switch Ports always use HDM Decoders to decode CXL.mem transactions.	RW	0x0
[0]	Poison_On_Decode_Error	This bit is RO and is hard-wired to 0 if Poison On Decode Error Capability=0. If set the component returns poison on read access to addresses that are not positively decoded by the component. If clear the component returns all 1s data without a poison under such scenarios.	RW	0x0

### 8.3.6.7 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Base\_High\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1210 + # {32 \* index}

##### Type

RWL

##### Reset value

See individual bit resets

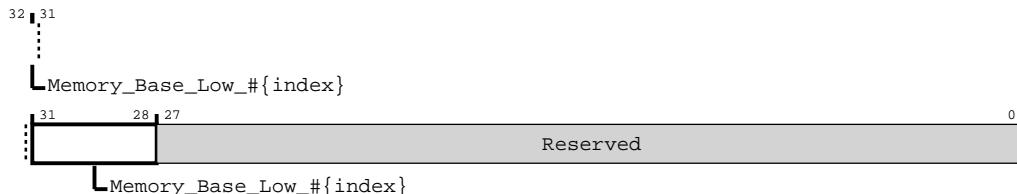
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-277: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_low**



**Table 8-280: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low_{index}	Corresponds to bits 31:28 of the base of the address range managed by decoder 0	RWL	0x0
[27:0]	Reserved	Reserved	RO	

## 8.3.6.8 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Base\_Low\_Register. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x1214 + #{32\*index}

#### Type

RWL

#### Reset value

See individual bit resets

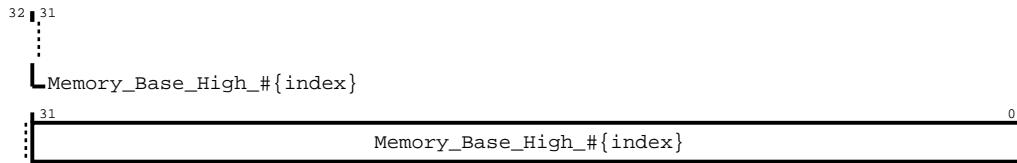
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-278: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_high**



**Table 8-281: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High_{index}	Corresponds to bits 63:32 of the base of the address range managed by decoder 0.	RWL	0x0

### 8.3.6.9 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Size\_Low\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1218 + #{32\*index}

##### Type

RWL

##### Reset value

See individual bit resets

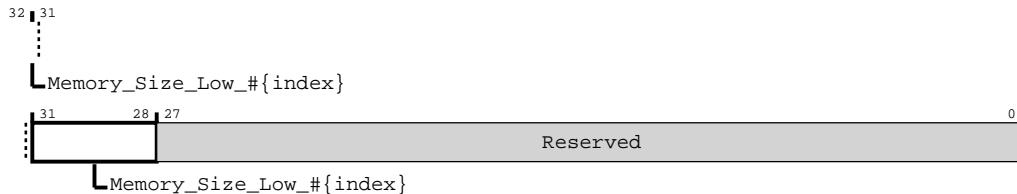
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-279: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_low**



**Table 8-282: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low_{index}	Corresponds to bits 31:28 of the size of the address range managed by decoder 0	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.6.10 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Size\_High\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x121C + #{32\*index}

##### Type

RWL

##### Reset value

See individual bit resets

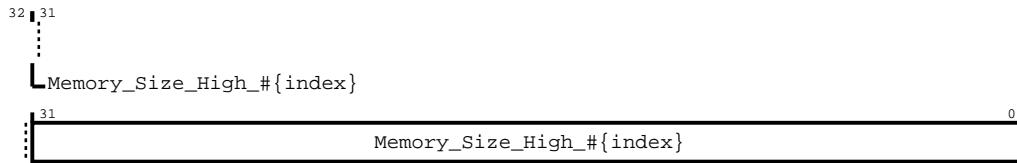
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-280: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_high**



**Table 8-283: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High_{index}	Corresponds to bits 63:32 of the size of address range managed by decoder 0.	RWL	0x0

### 8.3.6.11 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_control

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_Control\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1220 + #{32\*index}

##### Type

RWL

##### Reset value

See individual bit resets

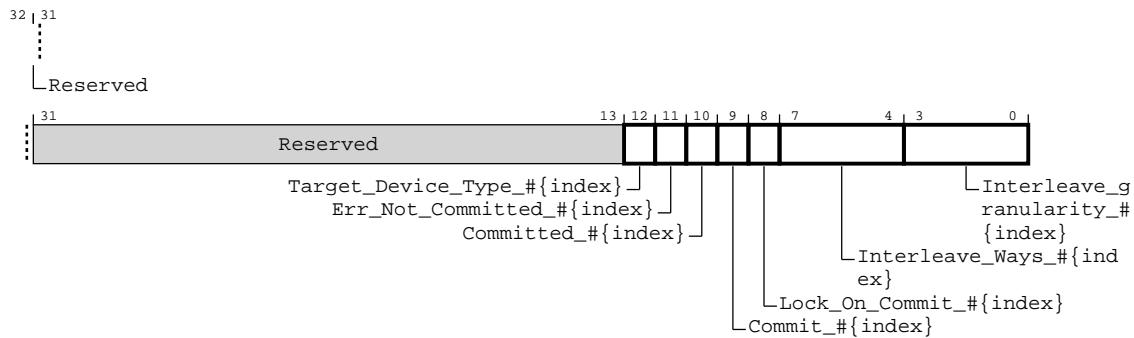
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-281: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_control**



**Table 8-284: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_control attributes**

Bits	Name	Description	Type	Reset
[31:13]	Reserved	Reserved	RO	
[12]	Target_Device_Type_{index}	<b>0</b> Target is a CXL Type 2 Device <b>1</b> Target is a CXL Type 3 Device	RWL	0x0
[11]	Err_Not_Committed_{index}	Indicates the decode programming had an error and decoder is not active.	RWL	0x0
[10]	Committed_{index}	Indicates the decoder is active	RWL	0x0
[9]	Commit_{index}	Software sets this to 1 to commit this decoder	RWL	0x0
[8]	Lock_On_Commit_{index}	If set all RWL fields in Decoder 0 registers will become read only when Committed changes to 1.	RWL	0x0
[7:4]	Interleave_Ways_{index}	The number of targets across which this memory range is interleaved. <b>0</b> : 1 way <b>1</b> : 2 way <b>2</b> : 4 way <b>3</b> : 8 way <b>Others</b> : All reserved	RWL	0x0
3:0	Commit_{index}	Interleave_Way_{index}, Interleave_Granularity_{index}, Lock_On_Commit_{index}, Commit_{index}		

Bits	Name	Description	Type	Reset
[3:0]	Interleave_granularity_{index}	The number of consecutive bytes that are assigned to each target in the Target List.	RWL	0x0
0		256 Bytes		
1		512 Bytes		
2		1024 Bytes (1KB)		
3		2048 Bytes (2KB)		
4		4096 Bytes (4KB)		
5		8192 Bytes (8KB)		
4		16384 Bytes (16KB)		

### 8.3.6.12 por\_cxladb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_DPA\_Skip\_Low\_Register. Only applicable for device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1224 + #{32\*index}

##### Type

RWL

##### Reset value

See individual bit resets

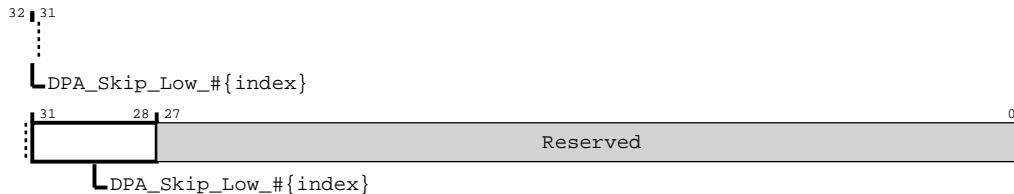
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-282: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low**



**Table 8-285: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	DPA_Skip_Low_{index}	Corresponds to bits 31:28 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.6.13 por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high

There are 8 iterations of this register. The index ranges from 0 to 7. Contains CXL\_HDM\_Decoder\_{index}\_DPA\_Skip\_High\_Register. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1228 + #{index}\*32}

##### Type

RWL

##### Reset value

See individual bit resets

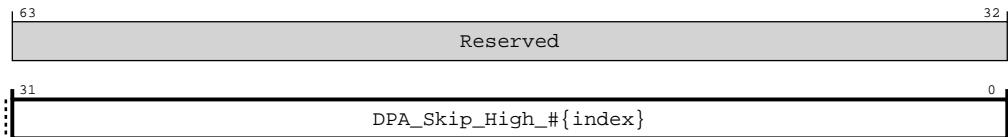
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-283: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high**



**Table 8-286: por\_cxlapb\_cxl\_hdm\_decoder\_0-7\_dpa\_skip\_high attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	DPA_Skip_High_{index}	Corresponds to bits 63:32 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	0x0

### 8.3.6.14 por\_cxlapb\_snoop\_filter\_group\_id

Contains Snoop\_Filter\_Group\_ID

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1800

##### Type

RO

##### Reset value

See individual bit resets

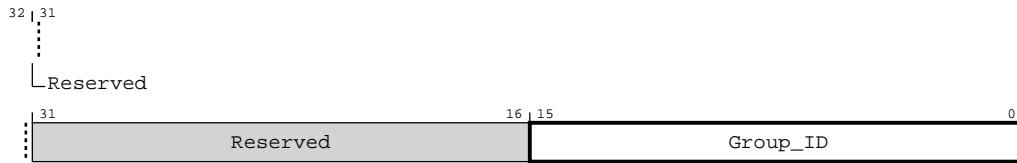
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-284: por\_cxlapb\_snoop\_filter\_group\_id**



**Table 8-287: por\_cxlapb\_snoop\_filter\_group\_id attributes**

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RO	
[15:0]	Group_ID	Uniquely identifies a snoop filter instance that is used to track CXL.cache devices below this Port. All Ports that share a single Snoop Filter instance shall set this field to the same value.	RO	0x0

### 8.3.6.15 por\_cxlapb\_snoop\_filter\_effective\_size

Contains Snoop\_Filter\_Effective\_Size

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1804

##### Type

RO

##### Reset value

See individual bit resets

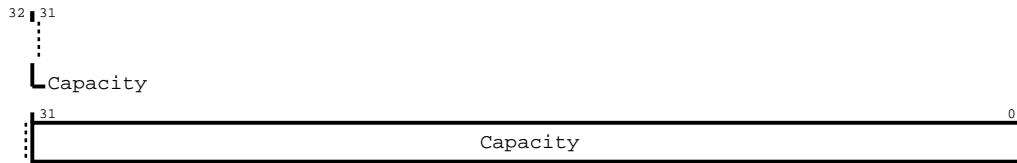
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-285: por\_cxlapb\_snoop\_filter\_effective\_size**



**Table 8-288: por\_cxlapb\_snoop\_filter\_effective\_size attributes**

Bits	Name	Description	Type	Reset
[31:0]	Capacity	Effective Snoop Filter Capacity representing the size of cache that can be effectively tracked by the Snoop Filter with this Group ID in multiples of 64K.	RO	0x0

### 8.3.6.16 por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_high

Contains DVSEC\_CXL\_Range\_1\_Base\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x120

##### Type

RWL

##### Reset value

See individual bit resets

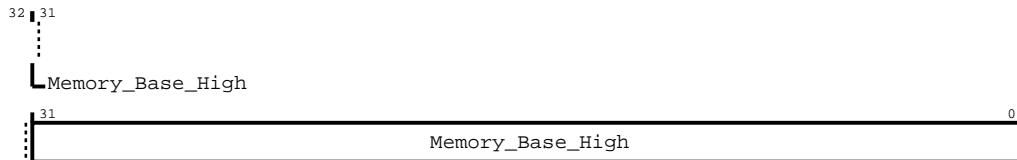
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-286: por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_high**



**Table 8-289: por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder O Base High Register for backward compatibility reasons.	RWL	0x0

### 8.3.6.17 por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_low

Contains DVSEC\_CXL\_Range\_1\_Base\_Low. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x124

##### Type

RWL

##### Reset value

See individual bit resets

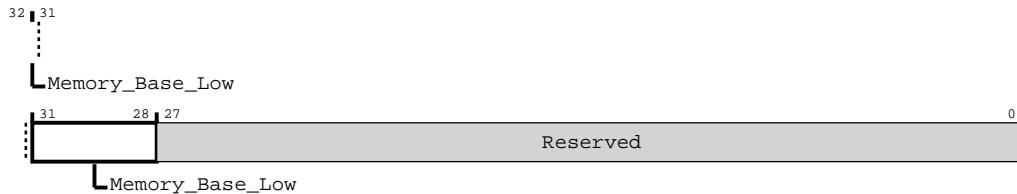
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-287: por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_low**



**Table 8-290: por\_cxlapb\_dvsec\_cxl\_range\_1\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base Low Register for backward compatibility reasons.	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.6.18 por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_high

Contains DVSEC\_CXL\_Range\_2\_Base\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x130

##### Type

RWL

##### Reset value

See individual bit resets

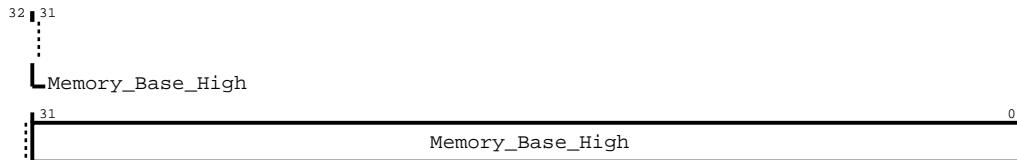
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-288: por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_high**



**Table 8-291: por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match the corresponding CXL HDM Decoder Base High Register for backward compatibility reasons.	RWL	0x0

### 8.3.6.19 por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_low

Contains DVSEC\_CXL\_Range\_2\_Base\_Low. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x134

##### Type

RWL

##### Reset value

See individual bit resets

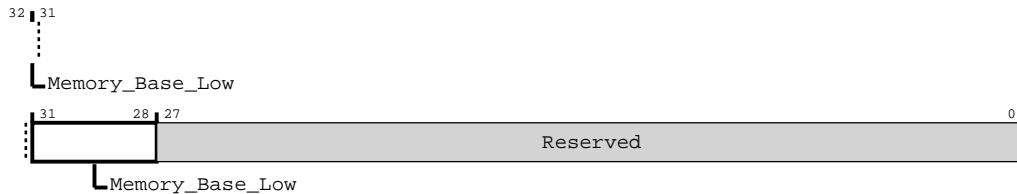
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-289: por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_low**



**Table 8-292: por\_cxlapb\_dvsec\_cxl\_range\_2\_base\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK	RWL	0x0
[27:0]	Reserved	Reserved	RO	

### 8.3.6.20 por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_high

Contains DVSEC\_CXL\_Range\_1\_Size\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x118

##### Type

RO

##### Reset value

See individual bit resets

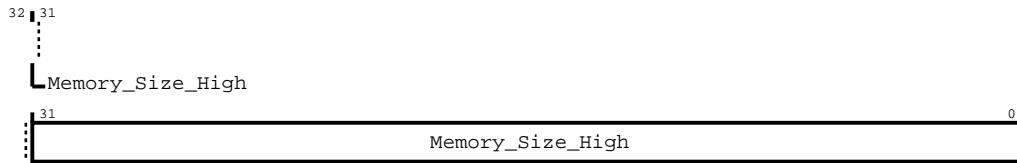
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-290: por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_high**



**Table 8-293: por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of CXL Range 1 memory size.	RO	0x0

### 8.3.6.21 por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_low

Contains DVSEC\_CXL\_Range\_1\_size\_Low. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x11C

##### Type

RO

##### Reset value

See individual bit resets

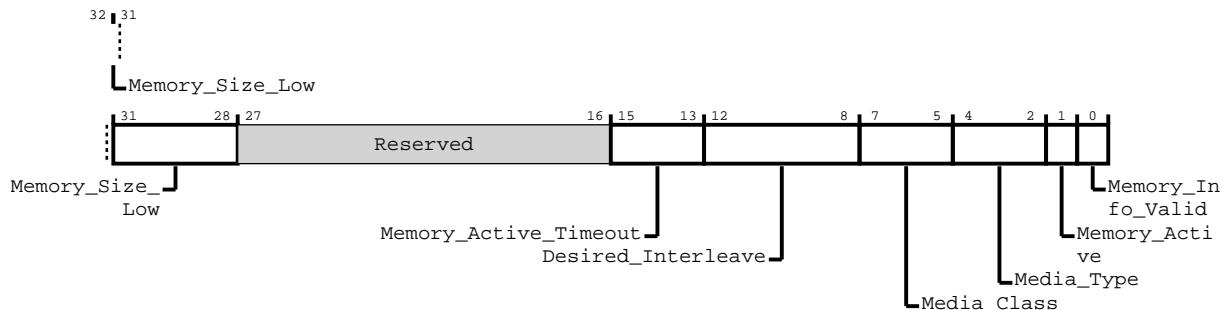
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-291: por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_low**



**Table 8-294: por\_cxlapb\_dvsec\_cxl\_range\_1\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of CXL Range 1 memory size.	RO	0x0
[27:16]	Reserved	Reserved	RO	
[15:13]	Memory_Active_Timeout	All other - Reserved	RO	0x1
[12:8]	Desired_Interleave	all other settings are reserved.	RO	0x3
[7:5]	Media_Class	CXL 2.0 and future CXL.mem devices shall set this field to 010.	RO	0x2
[4:2]	Media_Type	Other encodings are reserved. CXL 2.0 and future CXL.mem devices shall set this field to 010.	RO	0x2
[1]	Memory_Active	When set, indicates that the CXL Range 1 memory is fully initialized and available for software use. Must be set within Range 1. Memory_Active_Timeout of deassertion of reset to CXL device if CXL.mem HwInit Mode=1.	RO	0x0
[0]	Memory_Info_Valid	When set, indicates that the CXL Range 1 Size high and Size Low registers are valid. Must be set within 1 second of deassertion of reset to CXL device.	RO	0x0

### 8.3.6.22 por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_high

Contains DVSEC\_CXL\_Range\_2\_size\_High. Only applicable for Device. Host does not use this register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x128

##### Type

RO

### Reset value

See individual bit resets

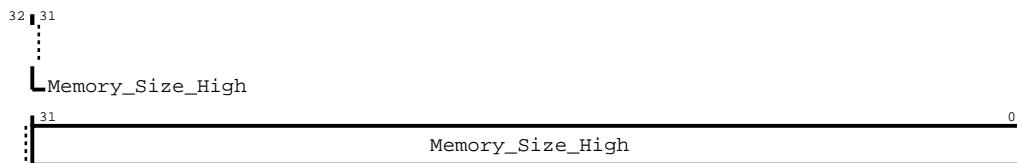
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-292: por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_high**



**Table 8-295: por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_high attributes**

Bits	Name	Description	Type	Reset
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of CXL Range 2 memory size.	RO	0x0

### 8.3.6.23 por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_low

Contains DVSEC\_CXL\_Range\_2\_size\_Low. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x12C

#### Type

RO

#### Reset value

See individual bit resets

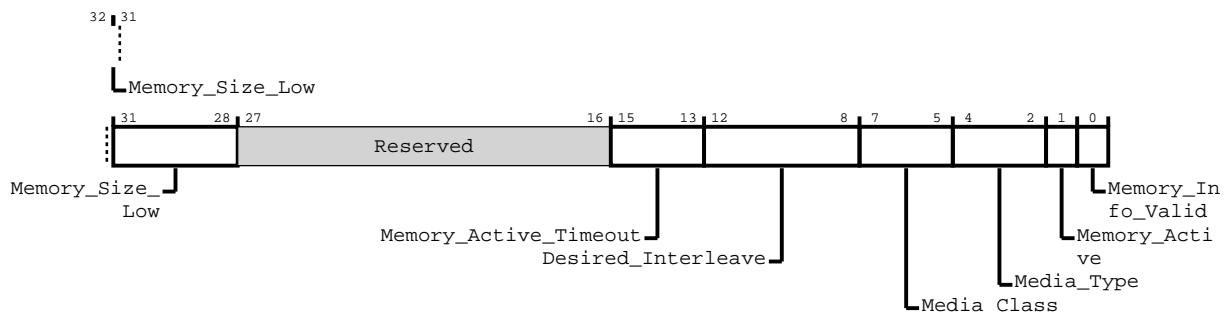
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-293: por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_low**



**Table 8-296: por\_cxlapb\_dvsec\_cxl\_range\_2\_size\_low attributes**

Bits	Name	Description	Type	Reset
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of CXL range 2 memory size.	RO	0x0
[27:16]	Reserved	Reserved	RO	
[15:13]	Memory_Active_Timeout	All other - Reserved	RO	0x1
[12:8]	Desired_Interleave	all other settings are reserved.	RO	0x3
[7:5]	Media_Class	CXL 2.0 and future CXL.mem devices shall set this field to 010.	RO	0x2
[4:2]	Media_Type	Other encodings are reserved. CXL 2.0 and future CXL.mem devices shall set this field to 010.	RO	0x2
[1]	Memory_Active	When set, indicates that the CXL range 2 memory is fully initialized and available for software use. Must be set within range 2. Memory_Active_Timeout of deassertion of reset to CXL device if CXL.mem HwInit Mode=1.	RO	0x0
[0]	Memory_Info_Valid	When set, indicates that the CXL range 2 Size high and Size Low registers are valid. Must be set within 1 second of deassertion of reset to CXL device.	RO	0x0

### 8.3.6.24 por\_cxlapb\_dvsec\_cxl\_control

Contains DVSEC\_CXL\_Control. Only applicable for Device. Host does not use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0x10C

##### Type

RWL

### Reset value

See individual bit resets

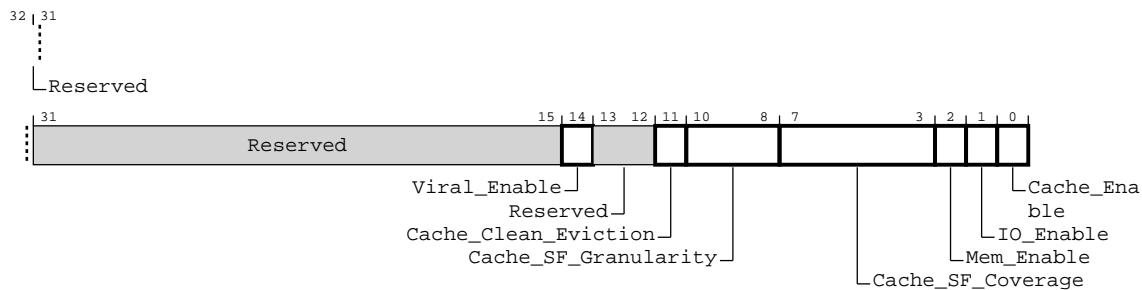
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-294: por\_cxlapb\_dvsec\_cxl\_control**



**Table 8-297: por\_cxlapb\_dvsec\_cxl\_control attributes**

Bits	Name	Description	Type	Reset
[31:15]	Reserved	Reserved	RO	
[14]	Viral_Enable	When set enables Viral handling in the CXL device. Locked by CONFIG_LOCK. If 0 the CXL device may ignore the viral that it receives	RWL	0x0
[13:12]	Reserved	Reserved	RO	
[11]	Cache_Clean_Eviction	Performance hint to the device. Locked by CONFIG_LOCK.  0      Indicates clean evictions from device caches are needed for best performance  1      Indicates clean evictions from device caches are NOT needed for best performance	RWL	0x0

Bits	Name	Description	Type	Reset
[10:8]	Cache_SF_Granularity	<p>Performance hint to the device. Locked by CONFIG_LOCK.</p> <p><b>000</b> Indicates 64B granular tracking on the Host</p> <p><b>001</b> Indicates 128B granular tracking on the Host</p> <p><b>010</b> Indicates 256B granular tracking on the Host</p> <p><b>011</b> Indicates 512B granular tracking on the Host</p> <p><b>100</b> Indicates 1KB granular tracking on the Host</p> <p><b>101</b> Indicates 2KB granular tracking on the Host</p> <p><b>110</b> Indicates 4KB granular tracking on the Host</p> <p><b>111</b> Reserved</p>	RWL	0x0
[7:3]	Cache_SF_Coverage	<p>Performance hint to the device. Locked by CONFIG_LOCK.</p> <p><b>0x00</b> Indicates no Snoop Filter coverage on the Host</p> <p><b>For all other values of N</b> Indicates Snoop Filter coverage on the Host of <math>2^{(N+15d)}</math> Bytes.</p>	RWL	0x0
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	0x0
[1]	IO_Enable	When set enables CXL.io protocol operation when in Flex Bus.CXL mode.	RWL	0x1
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	0x0

### 8.3.6.25 por\_cxlspb\_dvsec\_cxl\_control2

Contains DVSEC\_CXL\_Control2. Only applicable for Device. Host does not use this register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

32

##### Address offset

0x110

### Type

RW

### Reset value

See individual bit resets

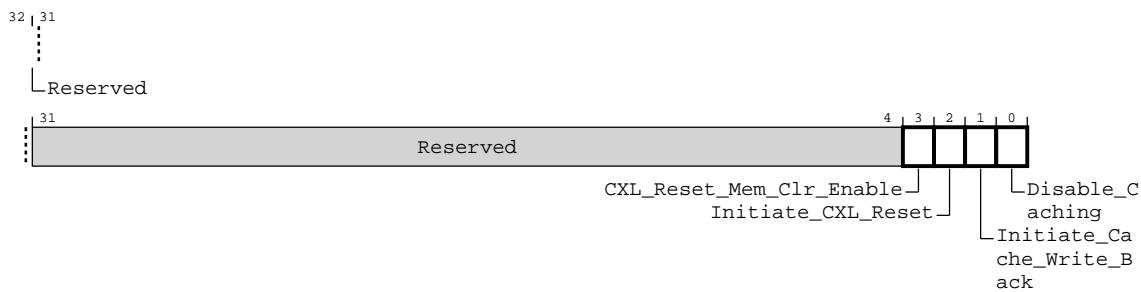
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-295: por\_cxlappb\_dvsec\_cxl\_control2**



**Table 8-298: por\_cxlappb\_dvsec\_cxl\_control2 attributes**

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	RO	
[3]	CXL_Reset_Mem_Clr_Enable	When set and CXL Reset Mem Clr Capable returns 1 Device shall clear or randomize volatile HDM ranges as part of the CXL Reset operation. When CXL Reset Mem Clr Capable is clear this bit is ignored and volatile HDM ranges may or may not be cleared or randomized during CXL Reset.	RW	0x0
[2]	Initiate_CXL_Reset	When set to 1 device shall initiate CXL Reset. This bit always returns the value of 0 when read by the software. A write of 0 is ignored.	RW	0x0
[1]	Initiate_Cache_Write_Back	When set to 1 device shall write back all modified lines in the local cache and invalidate all lines. The device shall send CacheFlushed message to host as required by CXL.Cache protocol to indicate it does not hold any modified lines.	RW	0x0
[0]	Disable_Caching	When set to 1 device shall no longer cache new modified lines in its local cache. Device shall continue to correctly respond to CXL.cache transactions.	RW	0x0

### 8.3.6.26 por\_cxlappb\_dvsec\_cxl\_lock

Contains DVSEC\_CXL\_Lock. Only applicable for Device. Host does not use this register

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0x114

### Type

RW

### Reset value

See individual bit resets

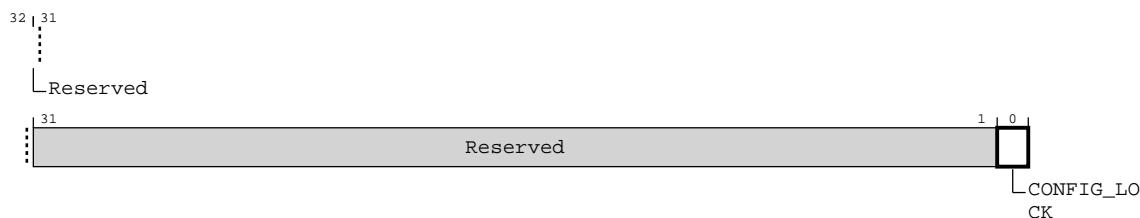
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-296: por\_cxlapb\_dvsec\_cxl\_lock**



**Table 8-299: por\_cxlapb\_dvsec\_cxl\_lock attributes**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	
[0]	CONFIG_LOCK	When set all register fields in the PCIe DVSEC for CXL Devices Capability with RWL attribute become read only. Consult individual register fields for details. This bit is cleared upon device Conventional Reset. This bit and all the fields that are locked by this bit are not affected by CXL Reset.	RW	0x0

## 8.3.6.27 por\_cxlapb\_dvsec\_flex\_bus\_port\_control

Contains DVSEC\_Flex\_Bus\_Port\_Control

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0x20C

### Type

RW

### Reset value

See individual bit resets

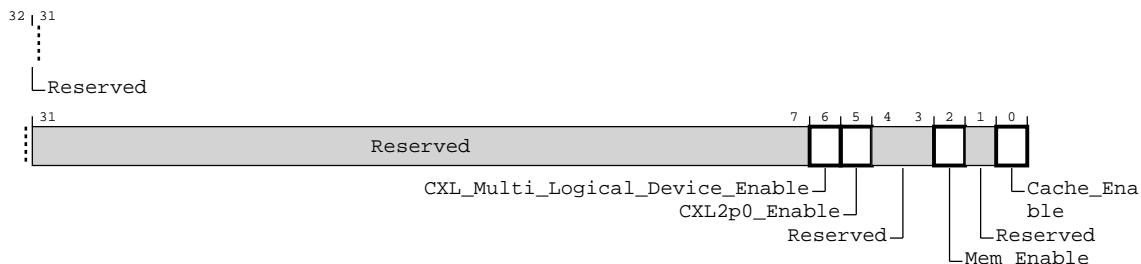
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-297: por\_cxlappb\_dvsec\_flex\_bus\_port\_control**



**Table 8-300: por\_cxlappb\_dvsec\_flex\_bus\_port\_control attributes**

Bits	Name	Description	Type	Reset
[31:7]	Reserved	Reserved	RO	
[6]	CXL_Multi_Logical_Device_Enable	When set enable Multi-Logical Device operation when in Flex Bus.CXL mode	RW	0x0
[5]	CXL2p0_Enable	When set enable CXL2.0 protocol operation when in Flex Bus.CXL mode.	RW	0x0
[4:3]	Reserved	Reserved	RO	
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode.	RW	0x0
[1]	Reserved	Reserved	RO	
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode.	RW	0x0

### 8.3.6.28 por\_cxlappb\_err\_capabilities\_control

Contains err\_capabilities\_control. Only applicable for Device. Host does not use this register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

32

### Address offset

0x1014

### Type

RW

### Reset value

See individual bit resets

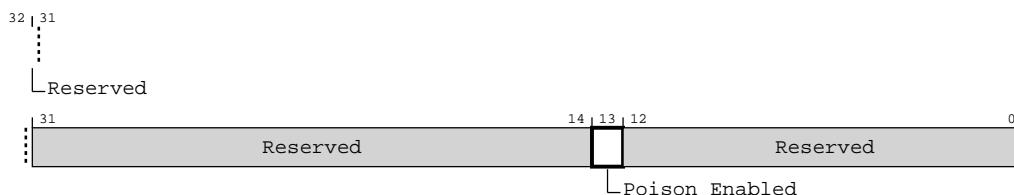
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-298: por\_cxlapb\_err\_capabilities\_control**



**Table 8-301: por\_cxlapb\_err\_capabilities\_control attributes**

Bits	Name	Description	Type	Reset
[31:14]	Reserved	Reserved	RO	
[13]	Poison_Enabled	If this bit is 0 CXL 1.1 Upstream Ports CXL 1.1 Downstream Ports and CXL 2.0 Root Port shall treat poison received on CXL.cache or CXL.mem as uncorrectable error and log the error in Uncorrectable Error Status Register. If this bit is 1 these ports shall treat poison received on CXL.cache or CXL.mem as correctable error and log the error in Correctable Error Status Register. This bit defaults to 1. This bit is hardwired to 1 in CXL 2.0 Upstream Switch Port CXL 2.0 Downstream Switch Port and CXL 2.0 device.	RW	0x0
[12:0]	Reserved	Reserved	RO	

### 8.3.6.29 por\_cxlapb\_IDE\_key\_refresh\_time\_control

Contains IDE\_key\_refresh\_time\_control. Not applicable to CMN 700

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0x1858

### Type

RW

### Reset value

See individual bit resets

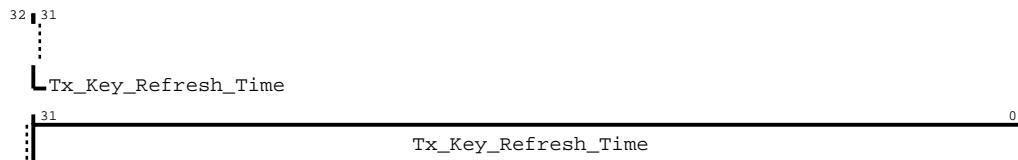
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-299: por\_cxlapp\_IDE\_key\_refresh\_time\_control**



**Table 8-302: por\_cxlapp\_IDE\_key\_refresh\_time\_control attributes**

Bits	Name	Description	Type	Reset
[31:0]	Tx_Key_Refesh_Time	Minimum number of flits transmitter needs to block transmission of protocol flits after IDE.Start has sent. Used when switching keys.	RW	0x0

## 8.3.6.30 por\_cxlapp\_IDE\_truncation\_transmit\_delay\_control

Contains IDE\_truncation\_transmit\_delay\_control. Not applicable to CMN 700

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0x185C

### Type

RW

### Reset value

See individual bit resets

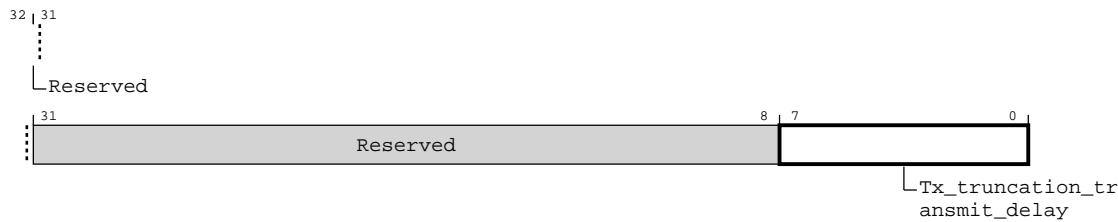
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-300: por\_cxlpb\_IDE\_truncation\_transmit\_delay\_control**



**Table 8-303: por\_cxlpb\_IDE\_truncation\_transmit\_delay\_control attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:0]	Tx_truncation_transmit_delay	This parameter feeds into the computation of minimum number of IDE idle flits Transmitter needs send after sending a truncated MAC flit.	RW	0x0

### 8.3.6.31 por\_cxlpb\_ll\_to\_ull\_msg

Contains ll\_to\_ull\_message

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x0

#### Type

RW

### Reset value

See individual bit resets

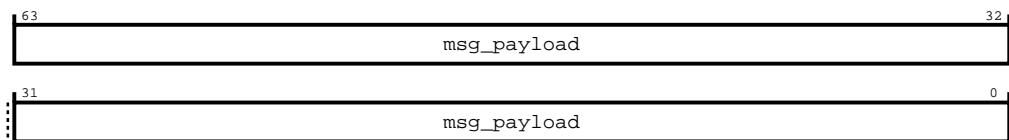
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-301: por\_cxlapb\_ll\_to\_ull\_msg**



**Table 8-304: por\_cxlapb\_ll\_to\_ull\_msg attributes**

Bits	Name	Description	Type	Reset
[63:0]	msg_payload	Contains 64 bits of message sent from ll to ull	RW	0x0

### 8.3.6.32 por\_cxlapb\_cxl\_timeout\_iso\_ctl

Contains cxl\_timeout\_iso\_ctl. Not applicable to CMN 700

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0x1908

#### Type

RW

#### Reset value

See individual bit resets

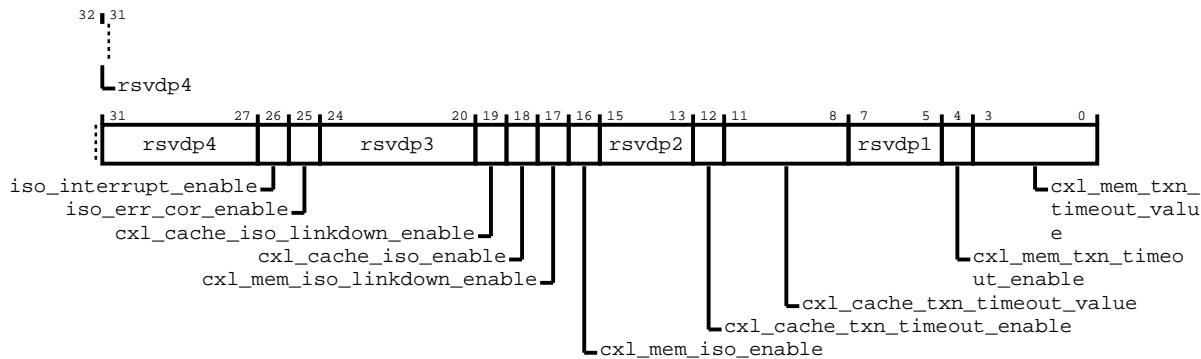
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-302: por\_cxlapb\_cxl\_timeout\_iso\_ctl**



**Table 8-305: por\_cxlapb\_cxl\_timeout\_iso\_ctl attributes**

Bits	Name	Description	Type	Reset
[31:27]	rsvdp4	Reserved	RW	0x0
[26]	iso_interrupt_enable	When Set this bit enables the generation of an interrupt to indicate that Isolation has been triggered.	RW	0x0
[25]	iso_err_cor_enable	When Set this bit enables the sending of an ERR_COR Message to indicate Isolation has been triggered.	RW	0x0
[24:20]	rsvdp3	Reserved	RW	0x0
[19]	cxl_cache_iso_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.cache enters Isolation mode.	RW	0x0
[18]	cxl_cache_iso_enable	This field allows System Software to enable CXL.cache Isolation actions.	RW	0x0
[17]	cxl_mem_iso_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.mem enters Isolation mode.	RW	0x0
[16]	cxl_mem_iso_enable	This field allows System Software to enable CXL.mem Isolation actions. If this field is set Isolation actions will be triggered if either a CXL.mem Transaction Timeout is detected or if the CXL link went down.	RW	0x0
[15:13]	rsvdp2	Reserved	RW	0x0
[12]	cxl_cache_txn_timeout_enable	-	RW	0x0
[11:8]	cxl_cache_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.cache.	RW	0x0
[7:5]	rsvdp1	Reserved	RW	0x0
[4]	cxl_mem_txn_timeout_enable	When Set this bit enables CXL.mem Transaction Timeout mechanism.	RW	0x0
[3:0]	cxl_mem_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.mem.	RW	0x0

### 8.3.6.33 por\_cxlapp\_cxl\_timeout\_iso\_capability

Contains cxl\_timeout\_iso\_capability.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1900

##### Type

RO

##### Reset value

See individual bit resets

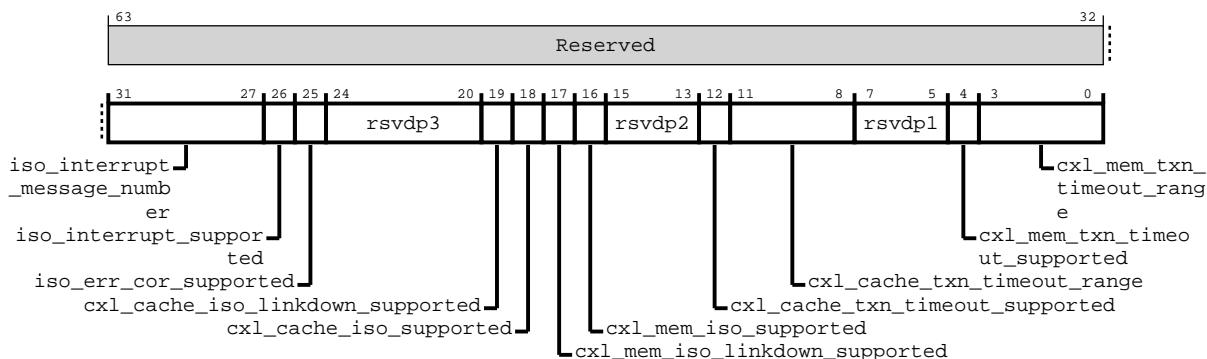
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-303: por\_cxlapp\_cxl\_timeout\_iso\_capability**



**Table 8-306: por\_cxlapp\_cxl\_timeout\_iso\_capability attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:27]	iso_interrupt_message_number	This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with the CXL Timeout and Isolation Capability structure.	RO	0x0
[26]	iso_interrupt_supported	This field indicates support for signaling interrupt when Isolation is triggered.	RO	0x1

Bits	Name	Description	Type	Reset
[25]	iso_err_cor_supported	If Set, this bit indicates that the Root Port supports the ability to signal with ERR_COR when Isolation is triggered.	RO	0x1
[24:20]	rsvdp3	Reserved	RO	0x0
[19]	cxl_cache_iso_linkdown_supported	This field indicates support for triggering Link-Down on the CXL Root Port if CXL.cache enters Isolation mode.	RO	0x0
[18]	cxl_cache_iso_supported	This field indicates support for Isolation on CXL.cache.	RO	0x0
[17]	cxl_mem_iso_linkdown_supported	This field indicates support for triggering Link-Down on the CXL port if CXL.mem enters Isolation mode.	RO	0x1
[16]	cxl_mem_iso_supported	This field indicates support for Isolation on CXL.mem.	RO	0x1
[15:13]	rsvdp2	Reserved	RO	0x0
[12]	cxl_cache_txn_timeout_supported	-	RO	0x0
[11:8]	cxl_cache_txn_timeout_range	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Range for CXL.cache.	RO	0x1
[7:5]	rsvdp1	Reserved	RO	0x0
[4]	cxl_mem_txn_timeout_supported	When Set indicates support for CXL.mem Transaction Timeout mechanism.	RO	0x1
[3:0]	cxl_mem_txn_timeout_range	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout ranges for CXL.mem.	RO	0x1

### 8.3.6.34 por\_cxlapb\_cxl\_timeout\_iso\_status

Contains cxl\_timeout\_iso\_status. Not applicable to CMN 700

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x190C

##### Type

RW

##### Reset value

See individual bit resets

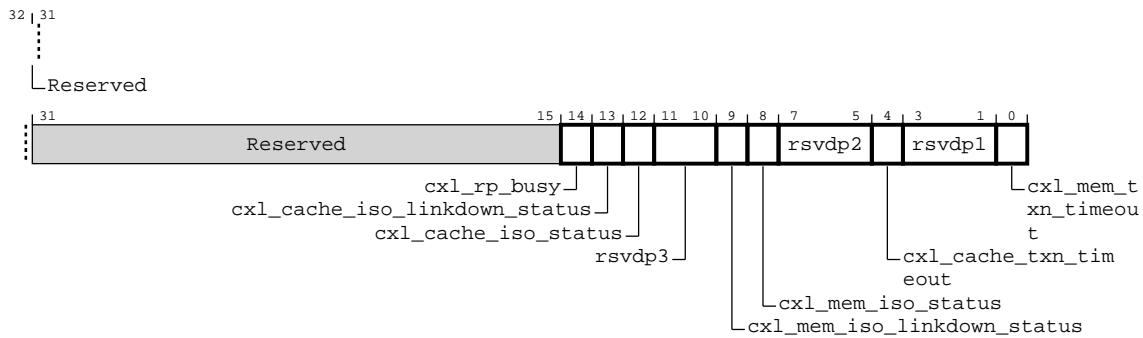
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-304: por\_cxlapb\_cxl\_timeout\_iso\_status**



**Table 8-307: por\_cxlapb\_cxl\_timeout\_iso\_status attributes**

Bits	Name	Description	Type	Reset
[31:15]	Reserved	Reserved	RO	
[14]	cxl_rp_busy	When either the CXL.mem Isolation Status bit or the CXL.cache Isolation Status bit is Set and this bit is Set, the Root Port is busy with internal activity that must complete before software is permitted to Clear the CXL.mem Isolation Status bit and the CXL.cache Isolation Status bit.	RW	0x0
[13]	cxl_cache_iso_linkdown_status	This field indicates that Isolation Mode for CXL.cache was triggered due to link down.	RW	0x0
[12]	cxl_cache_iso_status	This field indicates that Isolation mode for CXL.cache was triggered.	RW	0x0
[11:10]	rsvd3	Reserved	RW	0x0
[9]	cxl_mem_iso_linkdown_status	This field indicates that Isolation Mode for CXL.mem was triggered due to link down	RW	0x0
[8]	cxl_mem_iso_status	This field indicates that Isolation mode for CXL.mem was triggered.	RW	0x0
[7:5]	rsvd2	Reserved	RW	0x0
[4]	cxl_cache_txn_timeout	When Set, this indicates that a CXL.cache transaction timed out.	RW	0x0
[3:1]	rsvd1	Reserved	RW	0x0
[0]	cxl_mem_txn_timeout	When Set, this indicates that a CXL.mem transaction timed out.	RW	0x0

### 8.3.6.35 por\_cxlapb\_CXL\_Cache\_ID\_Decoder\_Capability

Contains CXL\_Cache\_ID\_Decoder\_Capability

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1980

### Type

RO

### Reset value

See individual bit resets

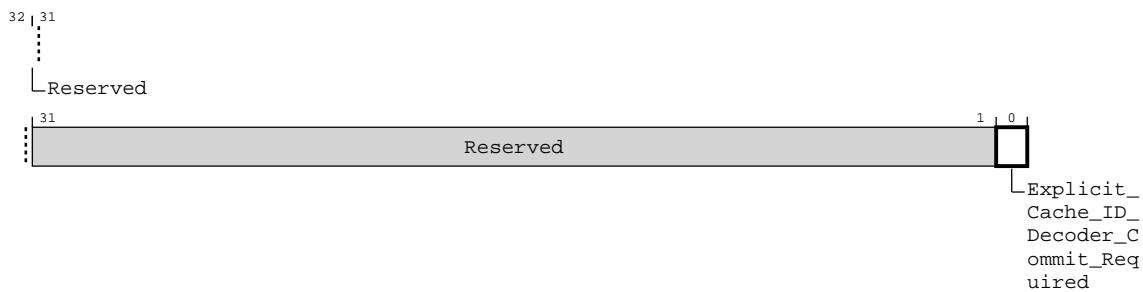
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-305: por\_cxlapp\_CXL\_Cache\_ID\_Decoder\_Capability**



**Table 8-308: por\_cxlapp\_CXL\_Cache\_ID\_Decoder\_Capability attributes**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	
[0]	Explicit_Cache_ID_Decoder_Commit_Required	If 1, indicates that the software must set Cache ID Decoder Commit bit anytime a new CXL.cache device is enabled anywhere below this port. If 1, the Cache ID Decoder Commit bit, the Cache ID Decoder Committed bit, the Cache ID Decoder Commit timeout Scale field, the Cache ID Decoder Commit Timeout Base field, and Cache ID Decoder Error Not Committed bit are implemented. Cache ID Decoder Commit operation may be used by a component to update its internal structures or perform consistency checks.	RO	0x0

### 8.3.6.36 por\_cxlapp\_CXL\_Cache\_ID\_Decoder\_Control

Contains CXL\_Cache\_ID\_Decoder\_Control

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

## Address offset

0x1984

## Type

RW

## Reset value

See individual bit resets

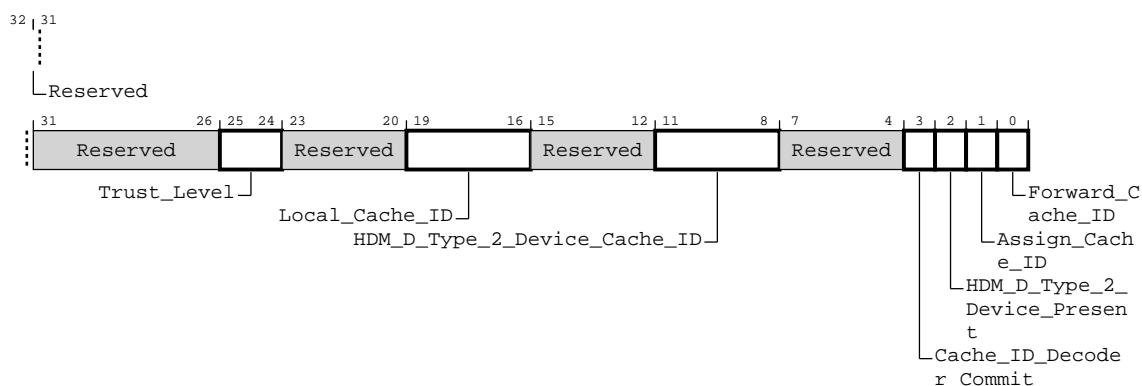
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-306: por\_cxlapb\_CXL\_Cache\_ID\_Decoder\_Control**



**Table 8-309: por\_cxlapb\_CXL\_Cache\_ID\_Decoder\_Control attributes**

Bits	Name	Description	Type	Reset
[31:26]	Reserved	Reserved	RO	
[25:24]	Trust_Level	Trust Level assigned to the directly connected device when Assign Cache ID=1. 00b = See Table 8-26 01b = Reserved 10b = See Table 8-26 11b = Reserved The reset default is 10b.	RW	0x0
[23:20]	Reserved	Reserved	RO	
[19:16]	Local_Cache_ID	If Assign Cache ID Enable=1, the Port assigns this Cache ID to the directly connected CXL.cache device regardless of whether it is using HDM-D flows or HDM-DB flows. The reset default is 0h.	RW	0x0
[15:12]	Reserved	Reserved	RO	
[11:8]	HDM_D_Type_2_Device_Cache_ID	If HDM-D Type 2 Device Present=1, this field represents the Cache ID that has been assigned to the Type 2 device below this Downstream Port that is using HDM-D flows. This field may be used by the port to identify a Type 2 device that is using HDM-D flows and must not be used for assigning Cache ID. The reset default is 0h.	RW	0x0
[7:4]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[3]	Cache_ID_Decoder_Commit	If Explicit Cache ID Decoder Commit Required=1, software must cause this bit to transition from 0 to 1 to commit the Cache ID assignment change to this Cache ID Decoder instance. The default value of this field is 0. This bit must be RW if the Explicit Cache ID Decoder Commit Required bit is set; otherwise, it is permitted to be hardwired to 0 and the Cache ID Decoder update does not require an explicit commit. Software must not set this bit unless the Explicit Cache ID Decoder Commit Required bit is set.	RW	0x0
[2]	HDM_D_Type_2_Device_Present	1 indicates there is a Type 2 Device below this Downstream Port that is using HDM-D flows. The reset default is 0.	RW	0x0
[1]	Assign_Cache_ID	1 indicates this Downstream Port is connected directly to a CXL.cache Device and assigns a cache ID=Local Cache ID to it. The reset default is 0.	RW	0x0
[0]	Forward_Cache_ID	1 indicates the Port forwards CXL.cache messages in both directions. The reset default is 0.	RW	0x0

### 8.3.6.37 por\_cxladb\_CXL\_Cache\_ID\_Decoder\_Status

Contains CXL\_Cache\_ID\_Decoder\_Status

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1988

##### Type

RO

##### Reset value

See individual bit resets

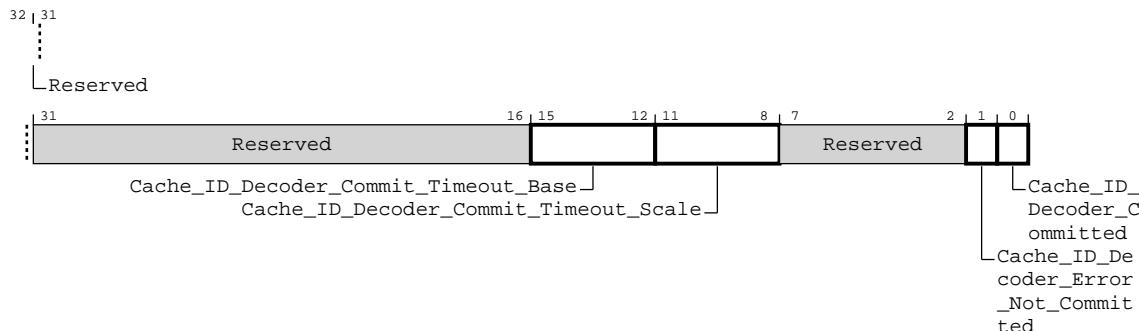
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-307: por\_cxlapb\_CXL\_Cache\_ID\_Decoder\_Status**



**Table 8-310: por\_cxlapb\_CXL\_Cache\_ID\_Decoder\_Status attributes**

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RO	
[15:12]	Cache_ID_Decoder_Commit_Timeout_Base	This field determines the Cache ID Decoder Commit timeout. The timeout duration is calculated by multiplying the Timeout Base with the Timeout Scale. Failure to set either the Cache ID Decoder Committed bit or the Cache ID Decoder Error Not Committed bit within the timeout value is treated as equivalent to commit error. In case of a timeout, the software must clear the Cache ID Decoder Commit bit to 0 prior to setting it to 1 again. This field is reserved if Explicit Cache ID Decoder Commit Required=0.	RO	0x0
[11:8]	Cache_ID_Decoder_Commit_Timeout_Scale	This field specifies the time scale associated with Cache ID Decoder Commit Timeout. 0000b = 1 us, 0001b = 10 us, 0010b = 100 us, 0011b = 1 ms, 0100b = 10 ms, 0101b = 100 ms, 0110b = 1 second, 0111b = 10 seconds. All other encodings are reserved. This field is reserved if Explicit Cache ID Decoder Commit Required=0.	RO	0x0
[7:2]	Reserved	Reserved	RO	
[1]	Cache_ID_Decoder_Error_Not_Committed	When set to 1, it indicates that the last write that caused the Cache ID Decoder Commit bit to transition from 0 to 1 was processed by the component, but resulted in an error. This bit is cleared when the software causes the Cache ID Decoder Commit bit to transition from 1 to 0. This bit is reserved if Explicit Cache ID Decoder Commit Required=0.	RO	0x0
[0]	Cache_ID_Decoder_Committed	When set to 1, it indicates that the last write that caused the Cache ID Decoder Commit bit to transition from 0 to 1 was successfully processed by the component. This bit is cleared when the software causes the Cache ID Decoder Commit bit to transition from 1 to 0. This bit is reserved if Explicit Cache ID Decoder Commit Required=0.	RO	0x0

### 8.3.6.38 por\_cxlapb\_link\_layer\_defeature

CXL Link Layer Defeature Register

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x1130

### Type

RW

### Reset value

See individual bit resets

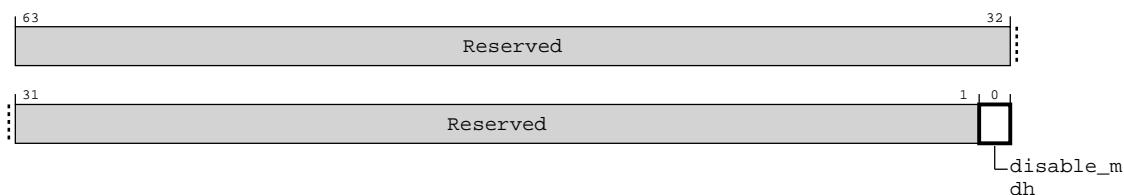
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-308: por\_cxlapb\_link\_layer\_defeature**



**Table 8-311: por\_cxlapb\_link\_layer\_defeature attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	disable_mdh	Write 1 to disable MDH. Software needs to ensure it programs this value consistently on the UP & DP. After programming, a warm reset is required for the disable to take effect.	RW	0b0

## 8.3.6.39 por\_c2capb\_c2c\_port\_register\_offset\_header

C2C Port Register Offset Header

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x2000

### Type

RO

### Reset value

See individual bit resets

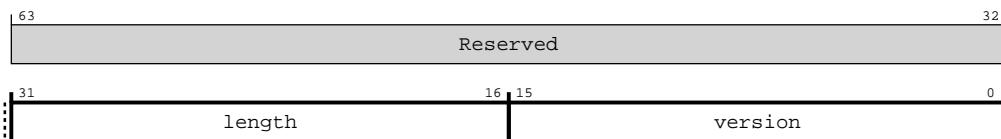
### Usage constraints

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 8-309: por\_c2capb\_c2c\_port\_register\_offset\_header**



**Table 8-312: por\_c2capb\_c2c\_port\_register\_offset\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	length	Length in bytes of this register block.	RO	0x20
[15:0]	version	Version of the Port Register Offset register block.	RO	0x0

## 8.3.6.40 por\_c2capb\_c2c\_port\_register\_offset\_table\_0

### C2C Port Register Offset Table 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

0x2008

### Type

RO

### Reset value

See individual bit resets

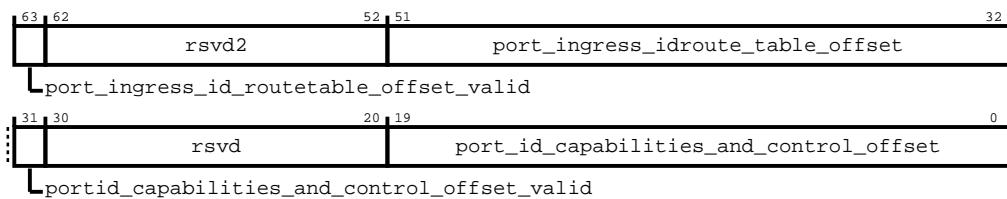
### Usage constraints

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 8-310: por\_c2capb\_c2c\_port\_register\_offset\_table\_0**



**Table 8-313: por\_c2capb\_c2c\_port\_register\_offset\_table\_0 attributes**

Bits	Name	Description	Type	Reset
[63]	port_ingress_id_routetable_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[62:52]	rsrvd2	rsrvd	RO	0x0
[51:32]	port_ingress_idroute_table_offset	Offset from the Port Registers Base for this register block.	RO	0x2060
[31]	portid_capabilities_and_control_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[30:20]	rsrvd	rsrvd	RO	0x0
[19:0]	port_id_capabilities_and_control_offset	Offset from the Port Capabilities Registers Base for this register block.	RO	0x2020

### 8.3.6.41 por\_c2capb\_c2c\_port\_register\_offset\_table\_1

C2C Port Register Offset Table 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2010

##### Type

RO

### Reset value

See individual bit resets

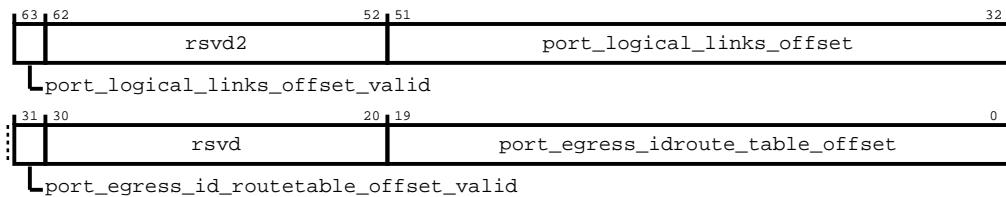
### Usage constraints

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 8-311: por\_c2capb\_c2c\_port\_register\_offset\_table\_1**



**Table 8-314: por\_c2capb\_c2c\_port\_register\_offset\_table\_1 attributes**

Bits	Name	Description	Type	Reset
[63]	port_logical_links_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[62:52]	rsvd2	rsvd	RO	0x0
[51:32]	port_logical_links_offset	Offset from the Port Registers Base for this register block.	RO	0x2200
[31]	port_egress_id_routetable_offset_valid	Indicates this register has been programmed and is valid.	RO	0b1
[30:20]	rsvd	rsvd	RO	0x0
[19:0]	port_egress_id_routetable_offset	Offset from the Port Registers Base for this register block.	RO	0x20B0

### 8.3.6.42 por\_c2capb\_c2c\_port\_capabilities\_and\_control\_header

C2C Port Register capabilities and control header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2020

##### Type

RO

### Reset value

See individual bit resets

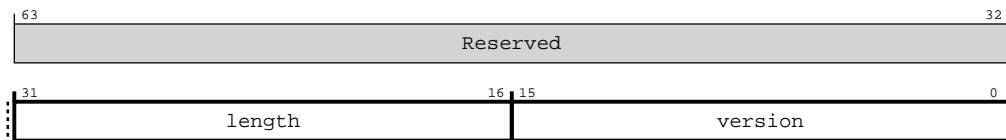
### Usage constraints

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 8-312: por\_c2capb\_c2c\_port\_capabilities\_and\_control\_header**



**Table 8-315: por\_c2capb\_c2c\_port\_capabilities\_and\_control\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	length	Length in bytes of this register block.	RO	0x40
[15:0]	version	Version value is 0h for this version of the spec.	RO	0x0

### 8.3.6.43 por\_c2capb\_c2c\_port\_capabilities\_1

C2C Port Register capabilities and control register 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2028

##### Type

RO

##### Reset value

See individual bit resets

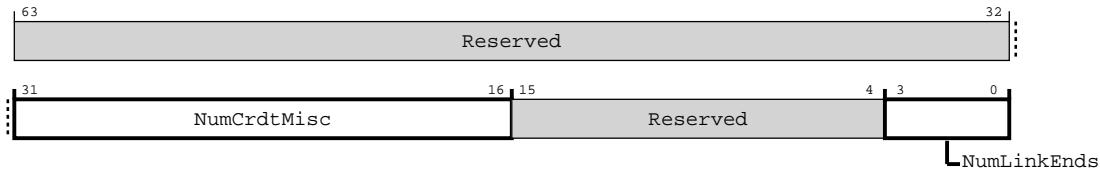
## Usage constraints

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 8-313: por\_c2capb\_c2c\_port\_capabilities\_1**



**Table 8-316: por\_c2capb\_c2c\_port\_capabilities\_1 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	NumCrdtMisc	Number of Misc Credits for this Port.	RO	0x40
[15:4]	Reserved	Reserved	RO	-
[3:0]	NumLinkEnds	Number of Link Ends for this Port. Value is 0 when the Port is disconnected.	RO	0x1

## 8.3.6.44 por\_c2capb\_c2c\_port\_capabilities\_2

C2C Port Register capabilities and control register 1

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x2030

#### Type

RO

#### Reset value

See individual bit resets

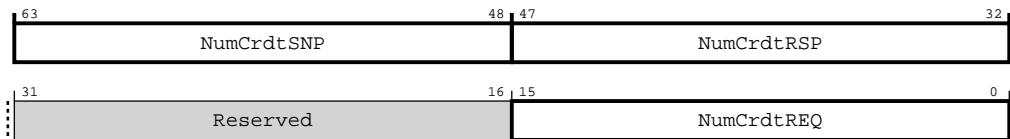
### Usage constraints

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 8-314: por\_c2capb\_c2c\_port\_capabilities\_2**



**Table 8-317: por\_c2capb\_c2c\_port\_capabilities\_2 attributes**

Bits	Name	Description	Type	Reset
[63:48]	NumCrdtSNP	Number is Snoop Credits capable.	RO	0x40
[47:32]	NumCrdtRSP	Number is Rsp Credits capable.	RO	0x40
[31:16]	Reserved	Reserved	RO	-
[15:0]	NumCrdtREQ	Number is Request Credits capable.	RO	0x0

## 8.3.6.45 por\_c2capb\_c2c\_port\_control\_and\_status

C2C Port Register capabilities and control and status

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x2040

#### Type

RW

#### Reset value

See individual bit resets

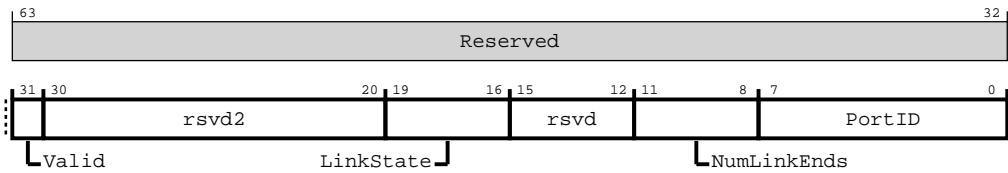
#### Usage constraints

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 8-315: por\_c2capb\_c2c\_port\_control\_and\_status**



**Table 8-318: por\_c2capb\_c2c\_port\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	Valid	Indicates this register has been programmed and is valid.	RW	0x0
[30:20]	rsvd2	rsvd	RW	0x0
[19:16]	LinkState	Link State for the Physical Link. Bit definitions: 0 Active 1 Disconnected 7:2 Reserved	RW	0x2
[15:12]	rsvd	rsvd	RW	0x0
[11:8]	NumLinkEnds	Number of Link Ends in this Port	RW	0x1
[7:0]	PortID	Port ID for this port	RW	0x0

### 8.3.6.46 por\_c2capb\_c2c\_port\_ingressid\_route\_table\_header

C2C Port ID route table header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2060

##### Type

RO

##### Reset value

See individual bit resets

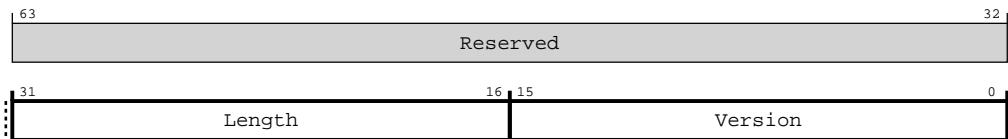
##### Usage constraints

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 8-316: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_header**



**Table 8-319: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x48
[15:0]	Version	Version of the Port ID Route Table register block. Version value is 0h for this version of the spec.	RO	0x0

## 8.3.6.47 por\_c2capb\_c2c\_port\_ingressid\_route\_table\_capabilities

C2C Port ID route table capabilities

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x2068

#### Type

RO

#### Reset value

See individual bit resets

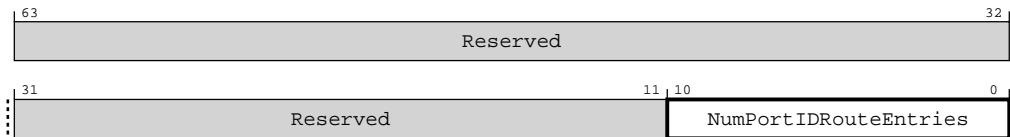
#### Usage constraints

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 8-317: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_capabilities**



**Table 8-320: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10:0]	NumPortIDRouteEntries	NumPortIdRouteEntries - Number of Route Id Entries supported. Valid values are 0 - $2^{\text{NumChipIDB}}\text{its}$ .	RO	0x0

### 8.3.6.48 por\_c2capb\_c2c\_port\_ingressid\_route\_table\_control\_and\_status

C2C Port ID route table control and status register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2080

##### Type

RW

##### Reset value

See individual bit resets

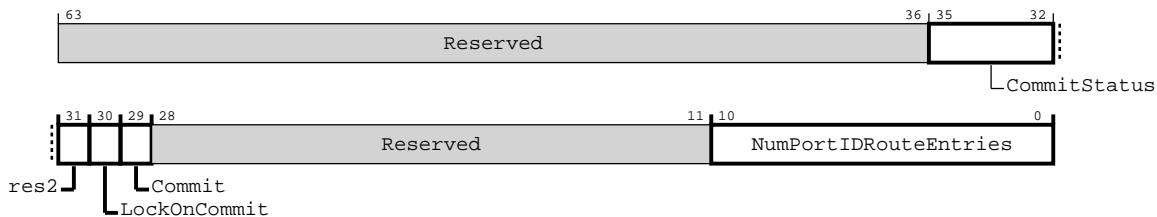
#### Usage constraints

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 8-318: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_control\_and\_status**



**Table 8-321: por\_c2capb\_c2c\_port\_ingressid\_route\_table\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35:32]	CommitStatus	Status of address decoder commit. Bit  <b>definitions</b> 0 Committed - Value of 1 indicated commit is complete 1 Error - commit failed to complete 2-3 Reserved After a commit, the Committed and/or Error bit remain set until the Commit bit is cleared. Clearing the Commit bit will cause all bits in this field to be cleared.	RW	0x0
[31]	res2	Reserved	RW	0x0
[30]	LockOnCommit	Lock the programming of the route table when committed Default  <b>Values</b> 0 - No Lock on Commit 1 - Lock On Commit When Lock On Commit is enabled, once the committed,all the route tablesare read only until reset	RW	0x0
[29]	Commit	Commit the updates to the address decoder. Commit bit will remain set until cleared by software. This bit must be cleared to initiate a subsequent commit.	RW	0x0
[28:11]	Reserved	Reserved	RW	0x0
[10:0]	NumPortIDRouteEntries	Number of Route ID Entries enabled. Valid values are 0 - 2^NumChipIDBits.	RW	0x0

### 8.3.6.49 por\_c2capb\_c2c\_port\_ingressid\_route\_entry\_0

C2C Port ID route table entry 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x20A0

##### Type

RW

## Reset value

See individual bit resets

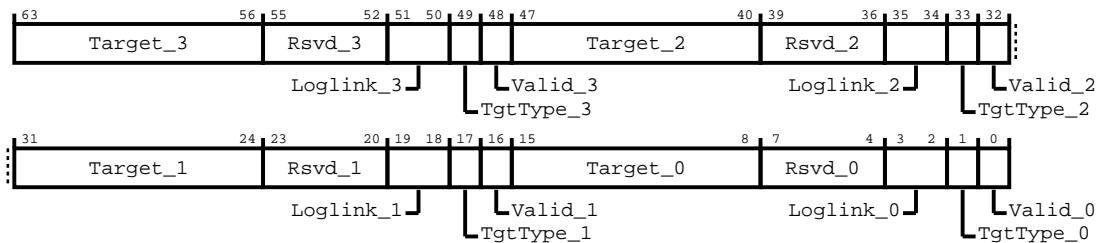
## Usage constraints

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 8-319: por\_c2capb\_c2c\_port\_ingressid\_route\_entry\_0**



**Table 8-322: por\_c2capb\_c2c\_port\_ingressid\_route\_entry\_0 attributes**

Bits	Name	Description	Type	Reset
[63:56]	Target_3	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[55:52]	Rsvd_3	Reserved	RW	0x0
[51:50]	Loglink_3	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[49]	TgtType_3	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[48]	Valid_3	Indicates this register has been programmed and is valid.	RW	0x0
[47:40]	Target_2	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[39:36]	Rsvd_2	Reserved	RW	0x0
[35:34]	Loglink_2	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[33]	TgtType_2	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[32]	Valid_2	Indicates this register has been programmed and is valid.	RW	0x0
[31:24]	Target_1	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[23:20]	Rsvd_1	Reserved	RW	0x0
[19:18]	Loglink_1	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[17]	TgtType_1	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[16]	Valid_1	Indicates this register has been programmed and is valid.	RW	0x0
[15:8]	Target_0	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[7:4]	Rsvd_0	Reserved	RW	0x0

Bits	Name	Description	Type	Reset
[3:2]	Loglink_0	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[1]	TgtType_0	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[0]	Valid_0	Indicates this register has been programmed and is valid.	RW	0x0

### 8.3.6.50 por\_c2capb\_c2c\_port\_egressid\_route\_table\_header

C2C Port ID route table header

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x20B0

##### Type

RO

##### Reset value

See individual bit resets

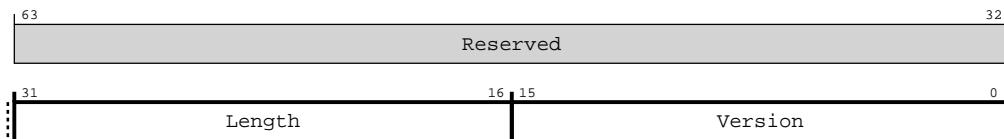
##### Usage constraints

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 8-320: por\_c2capb\_c2c\_port\_egressid\_route\_table\_header**



**Table 8-323: por\_c2capb\_c2c\_port\_egressid\_route\_table\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x48

Bits	Name	Description	Type	Reset
[15:0]	Version	Version of the Port ID Route Table register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.6.51 por\_c2capb\_c2c\_port\_egressid\_route\_table\_capabilities

C2C Port ID route table capabilities

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x20B8

##### Type

RO

##### Reset value

See individual bit resets

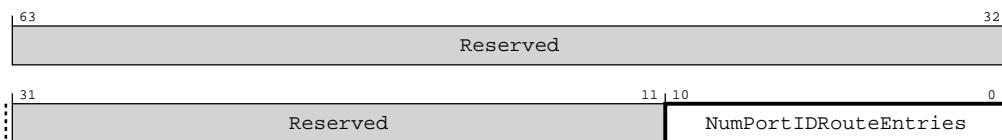
##### Usage constraints

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 8-321: por\_c2capb\_c2c\_port\_egressid\_route\_table\_capabilities**



**Table 8-324: por\_c2capb\_c2c\_port\_egressid\_route\_table\_capabilities attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10:0]	NumPortIDRouteEntries	NumPortIDRouteEntries - Number of Route Id Entries supported. Valid values are 0 - 2^NumChipIDBits.	RO	0x0

### 8.3.6.52 por\_c2capb\_c2c\_port\_egressid\_route\_table\_control\_and\_status

C2C Port ID route table control and status register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x20D0

##### Type

RW

##### Reset value

See individual bit resets

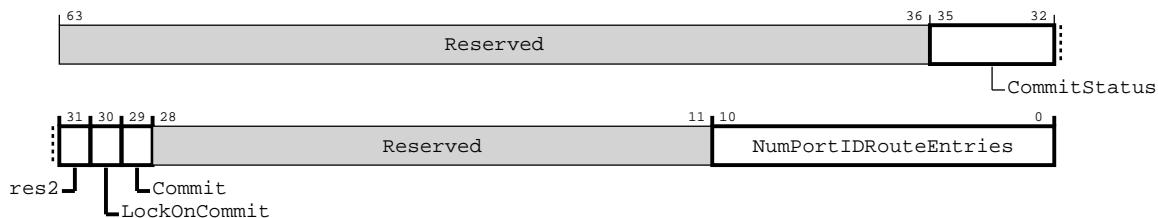
##### Usage constraints

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 8-322: por\_c2capb\_c2c\_port\_egressid\_route\_table\_control\_and\_status**



**Table 8-325: por\_c2capb\_c2c\_port\_egressid\_route\_table\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	
[35:32]	CommitStatus	Status of address decoder commit. Bit definitions:  0 Committed - Value of 1 indicated commit is complete, 1 Error - commit failed to complete, 2-3 Reserved. After a commit, the Committed and/or Error bit remain set until the Commit bit is cleared. Clearing the Commit bit will cause all bits in this field to be cleared.	RW	0x0
[31]	res2	Reserved	RW	0x0

Bits	Name	Description	Type	Reset
[30]	LockOnCommit	Lock the programming of the route table when committed Default Values: 0 - No Lock on Commit 1 - Lock On Commit When Lock On Commit is enabled, once the committed, all the route tables are read only until reset	RW	0x0
[29]	Commit	Commit the updates to the address decoder. Commit bit will remain set until cleared by software. This bit must be cleared to initiate a subsequent commit.	RW	0x0
[28:11]	Reserved	Reserved	RW	0x0
[10:0]	NumPortIDRouteEntries	Number of Route ID Entries enabled. Valid values are 0 - 2^NumChipIDBits.	RW	0x0

### 8.3.6.53 por\_c2capb\_c2c\_port\_egressid\_route\_entry\_0

C2C Port ID route table entry 0

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x20F0

##### Type

RW

##### Reset value

See individual bit resets

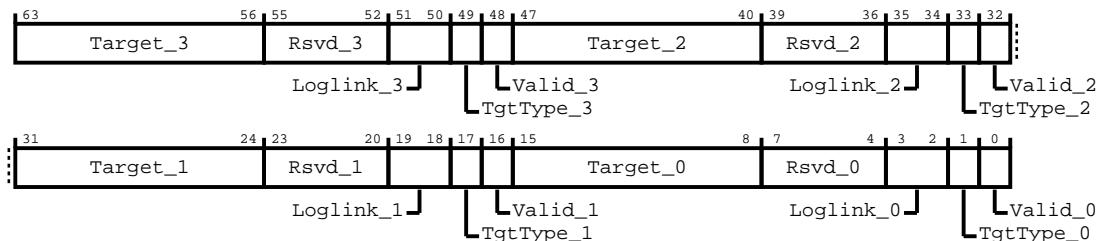
##### Usage constraints

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 8-323: por\_c2capb\_c2c\_port\_egressid\_route\_entry\_0**



**Table 8-326: por\_c2capb\_c2c\_port\_egressid\_route\_entry\_0 attributes**

Bits	Name	Description	Type	Reset
[63:56]	Target_3	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[55:52]	Rsvd_3	Reserved	RW	0x0
[51:50]	Loglink_3	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[49]	TgtType_3	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[48]	Valid_3	Indicates this register has been programmed and is valid.	RW	0x0
[47:40]	Target_2	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[39:36]	Rsvd_2	Reserved	RW	0x0
[35:34]	Loglink_2	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[33]	TgtType_2	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[32]	Valid_2	Indicates this register has been programmed and is valid.	RW	0x0
[31:24]	Target_1	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[23:20]	Rsvd_1	Reserved	RW	0x0
[19:18]	Loglink_1	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[17]	TgtType_1	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[16]	Valid_1	Indicates this register has been programmed and is valid.	RW	0x0
[15:8]	Target_0	The Target (Port or PAG) for the ID Route Entry.	RW	0x0
[7:4]	Rsvd_0	Reserved	RW	0x0
[3:2]	Loglink_0	The logical link for this ID Route Entry. Valid values are 0-3	RW	0x0
[1]	TgtType_0	Target type for this ID Route Entry. Valid Values: 0 Target ID is a Port Aggregation Group 1 Target ID is a port	RW	0x0
[0]	Valid_0	Indicates this register has been programmed and is valid.	RW	0x0

### 8.3.6.54 por\_c2capb\_c2c\_logical\_linkend\_common\_header

C2C logical link end common header register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

0x2200

### Type

RO

### Reset value

See individual bit resets

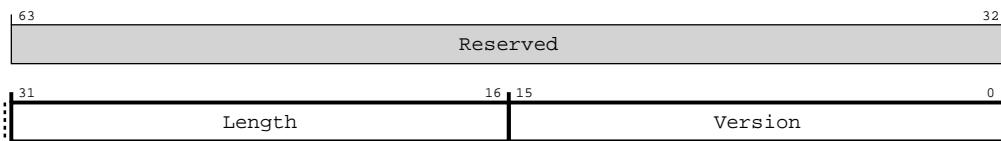
### Usage constraints

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 8-324: por\_c2capb\_c2c\_logical\_linkend\_common\_header**



**Table 8-327: por\_c2capb\_c2c\_logical\_linkend\_common\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x40
[15:0]	Version	Version of the Logical Link End register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.6.55 por\_c2capb\_c2c\_logical\_linkend\_offset\_register

C2C logical link end offset register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x2220

#### 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 8-325: por\_c2capp\_c2c\_logical\_linkend\_offset\_register**

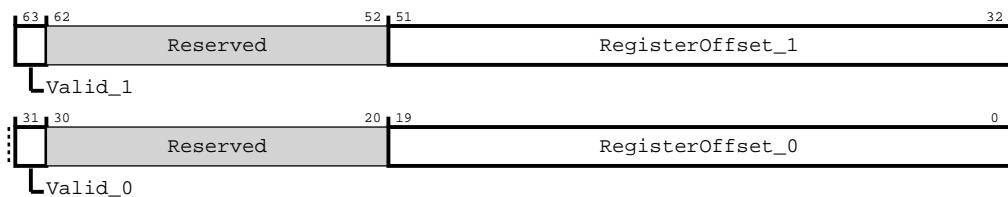


Table 8-328: por\_c2capb\_c2c\_logical\_linkend\_offset\_register attributes

Bits	Name	Description	Type	Reset
[63]	Valid_1	Indicates this register has been programmed and is valid.	RO	0x0
[62:52]	Reserved	Reserved	RO	-
[51:32]	RegisterOffset_1	Offset from the Port Registers Base for this register block.	RO	0x0
[31]	Valid_0	Indicates this register has been programmed and is valid.	RO	0x1
[30:20]	Reserved	Reserved	RO	-
[19:0]	RegisterOffset_0	Offset from the Port Registers Base for this register block.	RO	0x2240

### 8.3.6.56 por\_c2capb\_c2c\_logical\_linkend\_0-0\_header

There are 1 iterations of this register. The index ranges from 0 to 0. C2C logical link end header register

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

## Width

64

## Address offset

0x2240

### Type

RO

### Reset value

See individual bit resets

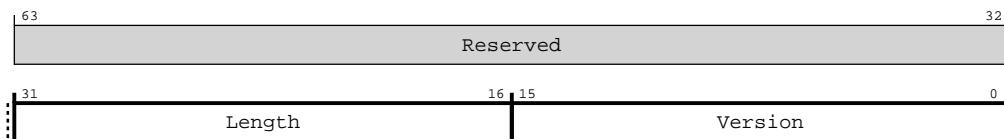
### Usage constraints

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 8-326: por\_c2capb\_c2c\_logical\_linkend\_0-0\_header**



**Table 8-329: por\_c2capb\_c2c\_logical\_linkend\_0-0\_header attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	Length	Length in bytes of this register block.	RO	0x180
[15:0]	Version	Version of the Logical Link End register block. Version value is 0h for this version of the spec.	RO	0x0

### 8.3.6.57 por\_c2capb\_c2c\_logical\_linkend\_0-0\_control\_and\_status

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Port ID route table control and status register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x2258

#### Type

RW

## Reset value

See individual bit resets

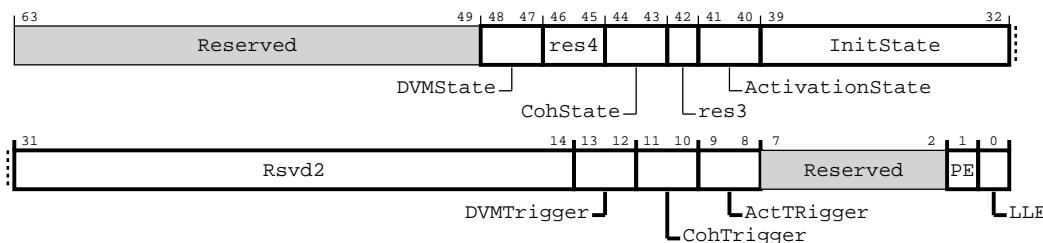
## Usage constraints

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 8-327: por\_c2capb\_c2c\_logical\_linkend\_0-0\_control\_and\_status**



**Table 8-330: por\_c2capb\_c2c\_logical\_linkend\_0-0\_control\_and\_status attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	
[49:48]	DVMState	DVM State for the Logical Link. Field values:  0 DVMDisabled 1 DVMConnect 2 DVMSigned 3 DVMDisconnect	RW	0x0
[47:46]	res4	Reserved	RW	0x0
[45:44]	CohState	Coherency State for the Logical Link. Field values:  0 CohDisabled 1 CohConnect 2 CohEnabled 3 CohDisconnect	RW	0x0
[43:42]	res3	Reserved	RW	0x0
[41:40]	ActivationState	• Activity State for the Logical Link. Field values:  0 STOP 1 ACTIVATE 2 RUN 3 DEACTIVATE	RW	0x0
[39:32]	InitState	Initialization State for the Logical Link. Bit definitions:  0 LinkStatusRcvd 1 RemoteActive 2 PropNegComplete 7:3 Reserved	RW	0x0
[31:18]	Rsvd2	Reserved	RW	0x0
[17:16]	DVMTrigger	DVM Trigger	RW	0x0
[15:14]	Reserved6	Reserved	RW	0x0
[13:12]	CohTrigger	Coherency Trigger	RW	0x0
[11:10]	Reserved5	Reserved	RW	0x0
[9:8]	ActTRigger	Activation Trigger	RW	0x0
[7:2]	Reserved	Reserved	RW	0x0

Bits	Name	Description	Type	Reset
[1]	PE	Property Exchange Enable Bit Values:  0 LogicalLinkEndEnd is disabled or not valid. 1 LogicalLinkEndEnd is enabled.	RW	0x1
[0]	LLE	LLE - LogicalLinkEndEnabled Bit Values:  0 LogicalLinkEndEnd is disabled or not valid. 1 LogicalLinkEndEnd is enabled.	RW	0x0

### 8.3.6.58 por\_c2capb\_c2c\_logical\_linkend\_0-0\_map\_table

There are 1 iterations of this register. The index ranges from 0 to 0. C2C logical link end map table register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2268

##### Type

RW

##### Reset value

See individual bit resets

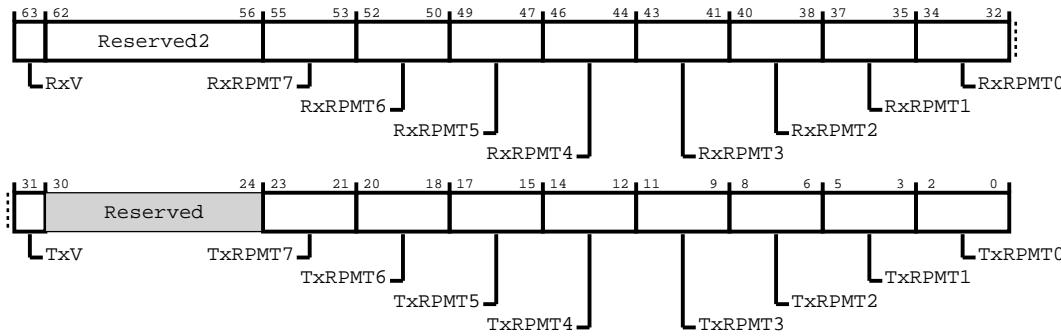
##### Usage constraints

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 8-328: por\_c2capb\_c2c\_logical\_linkend\_0-0\_map\_table**



**Table 8-331: por\_c2capb\_c2c\_logical\_linkend\_0-0\_map\_table attributes**

Bits	Name	Description	Type	Reset
[63]	RxV	Rx RP Map Valid. Bit Values: 0 Rx RP Map is not valid 1 Rx RP Map is valid	RW	0x0
[62:56]	Reserved2	Reserved	RW	0x0
[55:53]	RxRPMT7	Value indicates on-chip RP mapped to C2C RP 7	RW	0x0
[52:50]	RxRPMT6	Value indicates on-chip RP mapped to C2C RP 6	RW	0x0
[49:47]	RxRPMT5	Value indicates on-chip RP mapped to C2C RP 5	RW	0x0
[46:44]	RxRPMT4	Value indicates on-chip RP mapped to C2C RP 4	RW	0x0
[43:41]	RxRPMT3	Value indicates on-chip RP mapped to C2C RP 3	RW	0x0
[40:38]	RxRPMT2	Value indicates on-chip RP mapped to C2C RP 2	RW	0x0
[37:35]	RxRPMT1	Value indicates on-chip RP mapped to C2C RP 1	RW	0x0
[34:32]	RxRPMT0	Value indicates on-chip RP mapped to C2C RP 0	RW	0x0
[31]	TxV	Tx RP Map Valid. Bit Values: 0 Tx RP Map is not valid 1 Tx RP Map is valid	RW	0x0
[30:24]	Reserved	Reserved	RW	0x0
[23:21]	TxRPMT7	Value indicates C2C RP mapped to on-chip RP 7	RW	0x0
[20:18]	TxRPMT6	Value indicates C2C RP mapped to on-chip RP 6	RW	0x0
[17:15]	TxRPMT5	Value indicates C2C RP mapped to on-chip RP 5	RW	0x0
[14:12]	TxRPMT4	Value indicates C2C RP mapped to on-chip RP 4	RW	0x0
[11:9]	TxRPMT3	Value indicates C2C RP mapped to on-chip RP 3	RW	0x0
[8:6]	TxRPMT2	Value indicates C2C RP mapped to on-chip RP 2	RW	0x0
[5:3]	TxRPMT1	Value indicates C2C RP mapped to on-chip RP 1	RW	0x0
[2:0]	TxRPMT0	Value indicates C2C RP mapped to on-chip RP 0	RW	0x0

### 8.3.6.59 por\_c2capb\_c2c\_linkproperties\_supported\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2280 + # {8 \* index}

##### Type

RO

##### Reset value

See individual bit resets

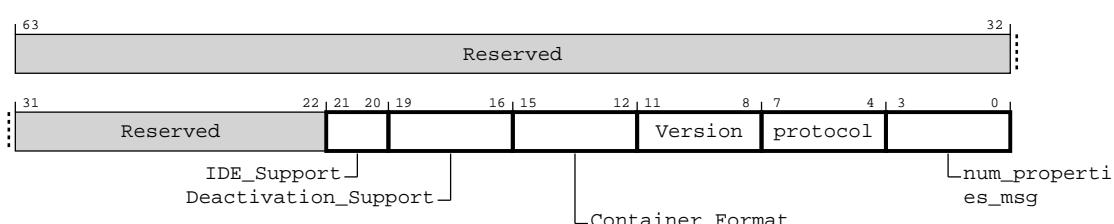
##### Usage constraints

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 8-329: por\_c2capb\_c2c\_linkproperties\_supported\_uniform1\_0-0**



**Table 8-332: por\_c2capb\_c2c\_linkproperties\_supported\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010

Bits	Name	Description	Type	Reset
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0: FormatX 0 Not supported 1 Supported  Bit 1: FormatY 0 Not supported 1 Supported  Bits 2: & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.6.60 por\_c2capb\_c2c\_linkproperties\_advertised\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x22C0 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

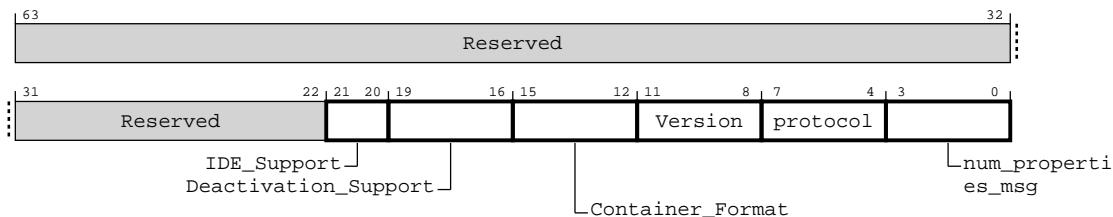
##### Usage constraints

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 8-330: por\_c2capb\_c2c\_linkproperties\_advertised\_uniform1\_0-0**



**Table 8-333: por\_c2capb\_c2c\_linkproperties\_advertised\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0: FormatX 0 Not supported 1 Supported  Bit 1: FormatY 0 Not supported 1 Supported  Bits 2: & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.6.61 por\_c2capb\_c2c\_linkproperties\_informed\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

## Address offset

$0x2300 + \#(8 * \text{index})$

## Type

RO

## Reset value

See individual bit resets

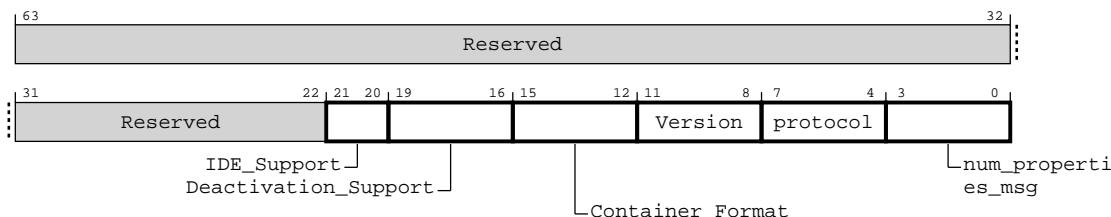
## Usage constraints

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 8-331: por\_c2capb\_c2c\_linkproperties\_informed\_uniform1\_0-0**



**Table 8-334: por\_c2capb\_c2c\_linkproperties\_informed\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0: FormatX 0 Not supported 1 Supported  Bit 1: FormatY 0 Not supported 1 Supported  Bits 2: & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.6.62 por\_c2capb\_c2c\_linkproperties\_negotiated\_uniform1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C uniform property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2340 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

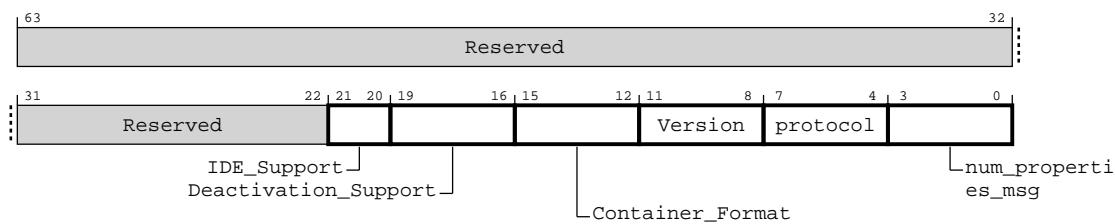
##### Usage constraints

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 8-332: por\_c2capb\_c2c\_linkproperties\_negotiated\_uniform1\_0-0**



**Table 8-335: por\_c2capb\_c2c\_linkproperties\_negotiated\_uniform1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:20]	IDE_Support	IDE support bits	RO	0b00

Bits	Name	Description	Type	Reset
[19:16]	Deactivation_Support	Deactivation modes supported by the interface:  Bit 0 Protocol Agnostic 0 Not supported 1 Supported  Bit 1 Protocol Aware 0 Not supported 1 Supported	RO	0b0010
[15:12]	Container_Format	Container format supported by the interface in both directions:  Bit 0: FormatX 0 Not supported 1 Supported  Bit 1: FormatY 0 Not supported 1 Supported  Bits 2: & 3 Reserved	RO	0b0001
[11:8]	Version	0b0000: A  Others: Reserved	RO	0b0000
[7:4]	protocol	Linked to the Protocol property.	RO	0b0000
[3:0]	num_properties_msg	Linked to the Protocol property. The transmitter should only send the correct number of Properties messages that the receiver can accept.	RO	0b0000

### 8.3.6.63 por\_c2capb\_c2c\_linkproperties\_supported\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2290 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

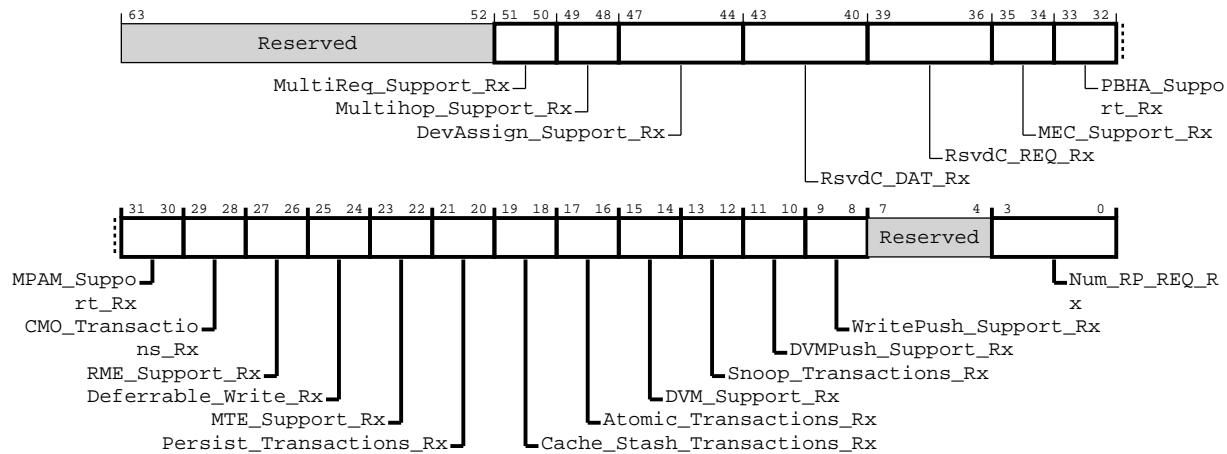
##### Usage constraints

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 8-333: por\_c2capb\_c2c\_linkproperties\_supported\_rx1\_0-0**



**Table 8-336: por\_c2capb\_c2c\_linkproperties\_supported\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.64 por\_c2capb\_c2c\_linkproperties\_advertised\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x22D0 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

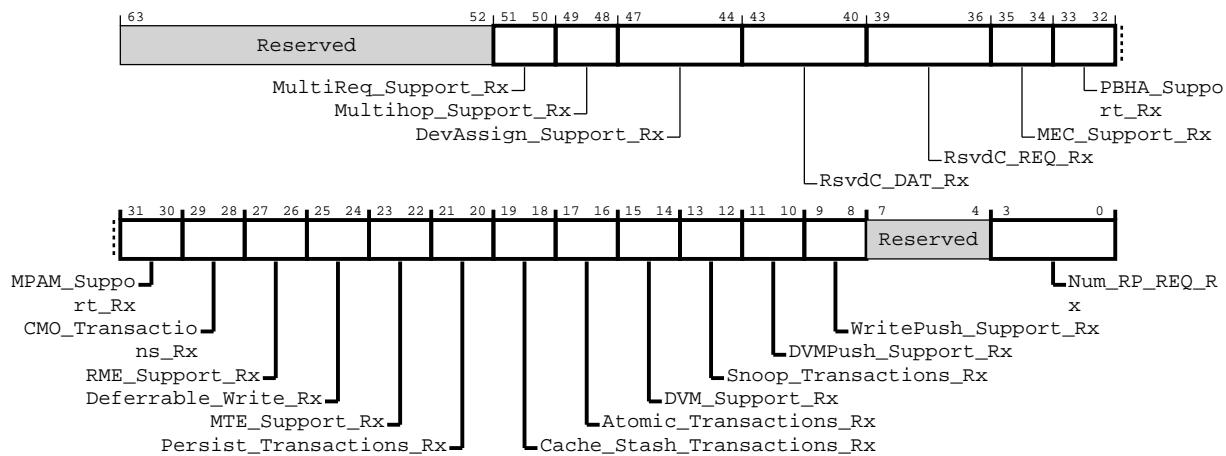
##### Usage constraints

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 8-334: por\_c2capb\_c2c\_linkproperties\_advertised\_rx1\_0-0**



**Table 8-337: por\_c2capb\_c2c\_linkproperties\_advertised\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.65 por\_c2capb\_c2c\_linkproperties\_informed\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2310 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

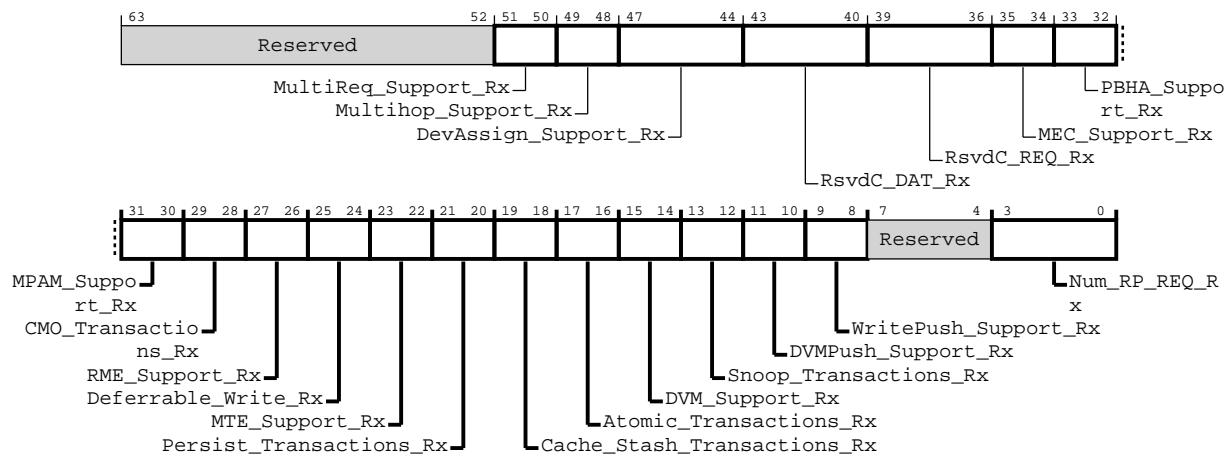
##### Usage constraints

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 8-335: por\_c2capb\_c2c\_linkproperties\_informed\_rx1\_0-0**



**Table 8-338: por\_c2capb\_c2c\_linkproperties\_informed\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.66 por\_c2capb\_c2c\_linkproperties\_negotiated\_rx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C rx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2350 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

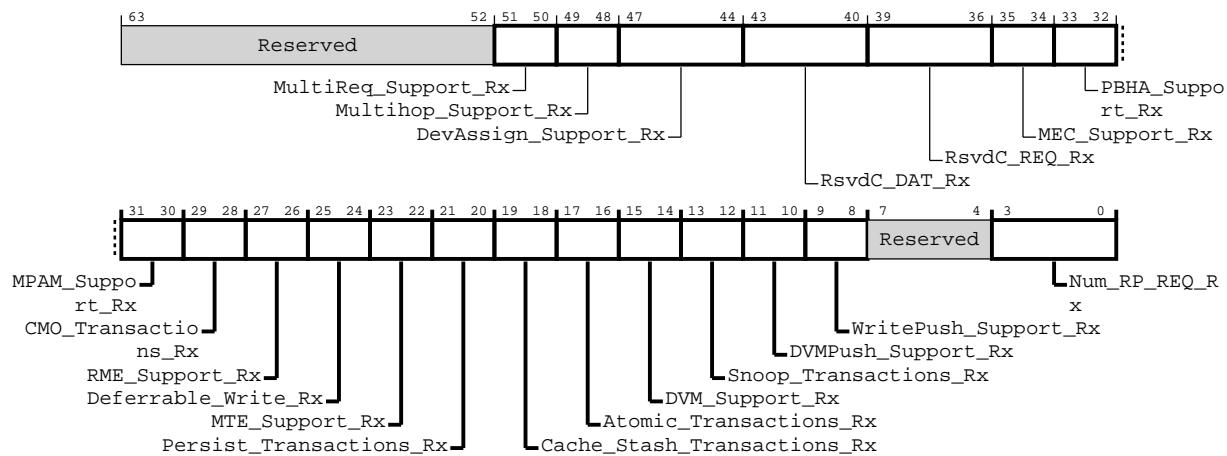
##### Usage constraints

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 8-336: por\_c2capb\_c2c\_linkproperties\_negotiated\_rx1\_0-0**



**Table 8-339: por\_c2capb\_c2c\_linkproperties\_negotiated\_rx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Rx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Rx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Rx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Rx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Rx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Rx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Rx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Rx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Rx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Rx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Rx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Rx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Rx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Rx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Rx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Rx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Rx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Rx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Rx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Rx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Rx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.67 por\_c2capb\_c2c\_linkproperties\_supported\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x22A0 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

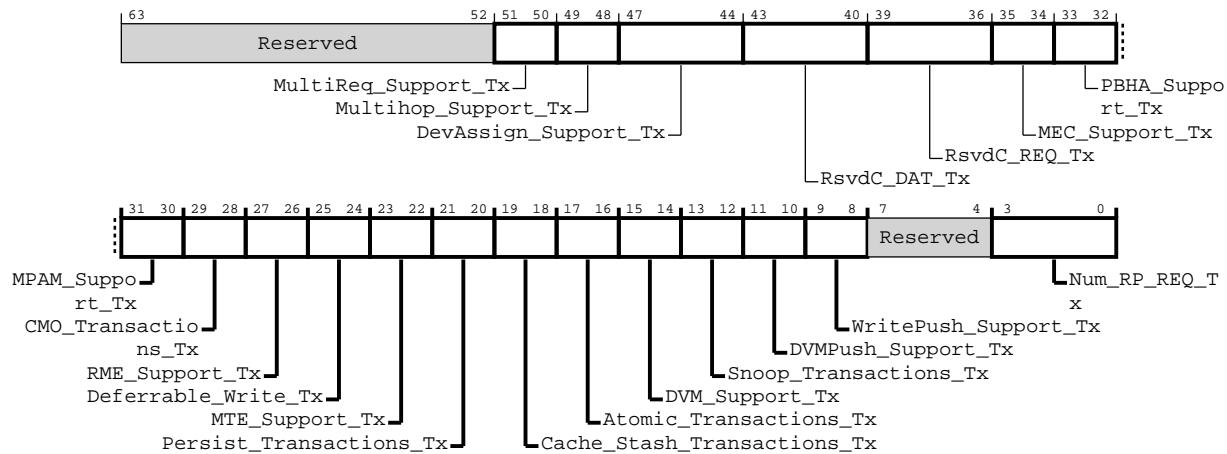
##### Usage constraints

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 8-337: por\_c2capb\_c2c\_linkproperties\_supported\_tx1\_0-0**



**Table 8-340: por\_c2capb\_c2c\_linkproperties\_supported\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Tx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Tx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.68 por\_c2capb\_c2c\_linkproperties\_advertised\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x22E0 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

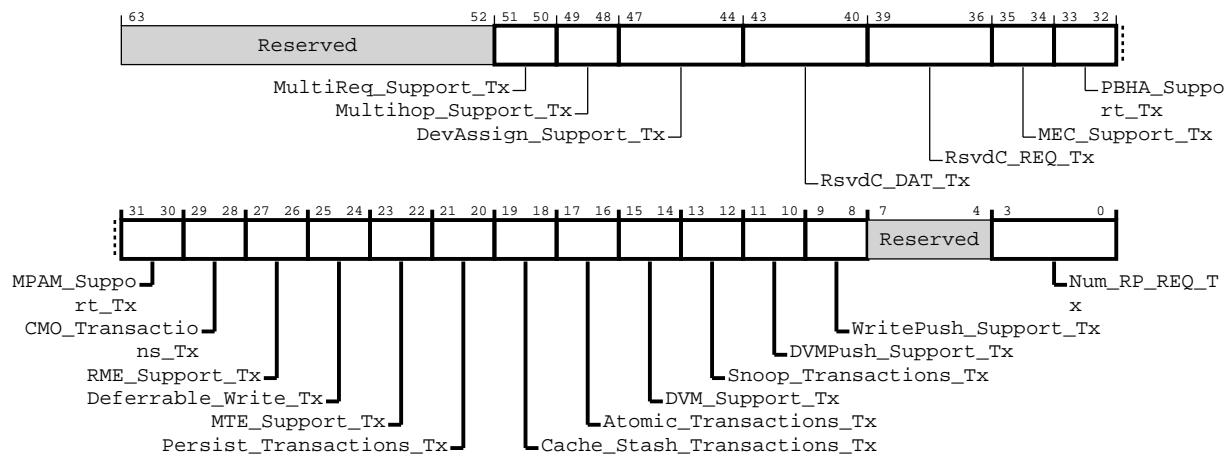
##### Usage constraints

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 8-338: por\_c2capb\_c2c\_linkproperties\_advertised\_tx1\_0-0**



**Table 8-341: por\_c2capb\_c2c\_linkproperties\_advertised\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Tx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Tx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.69 por\_c2capb\_c2c\_linkproperties\_informed\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2320 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

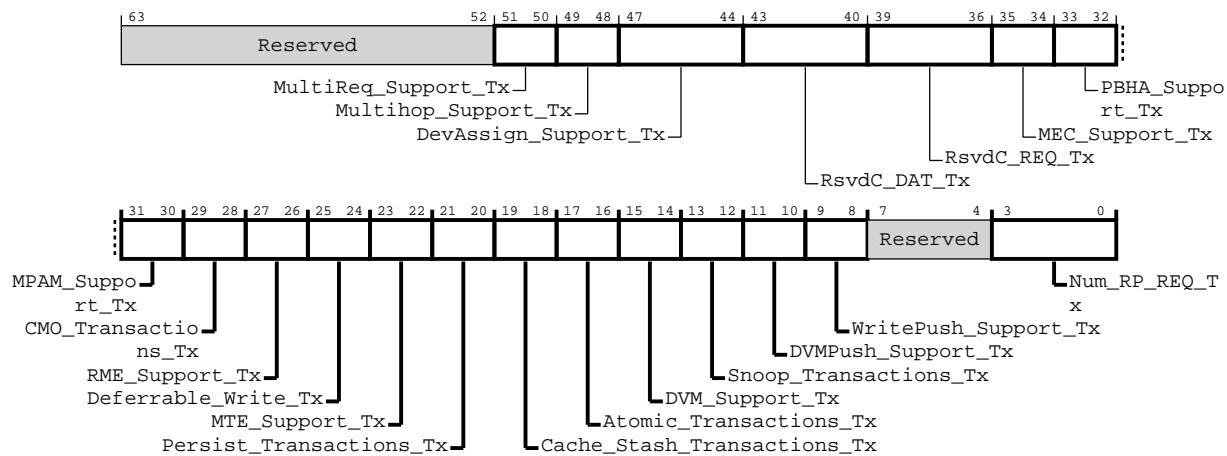
##### Usage constraints

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 8-339: por\_c2capb\_c2c\_linkproperties\_informed\_tx1\_0-0**



**Table 8-342: por\_c2capb\_c2c\_linkproperties\_informed\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Tx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Tx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.70 por\_c2capb\_c2c\_linkproperties\_negotiated\_tx1\_0-0

There are 1 iterations of this register. The index ranges from 0 to 0. C2C tx property supported set

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2360 + #{8\*index}

##### Type

RO

##### Reset value

See individual bit resets

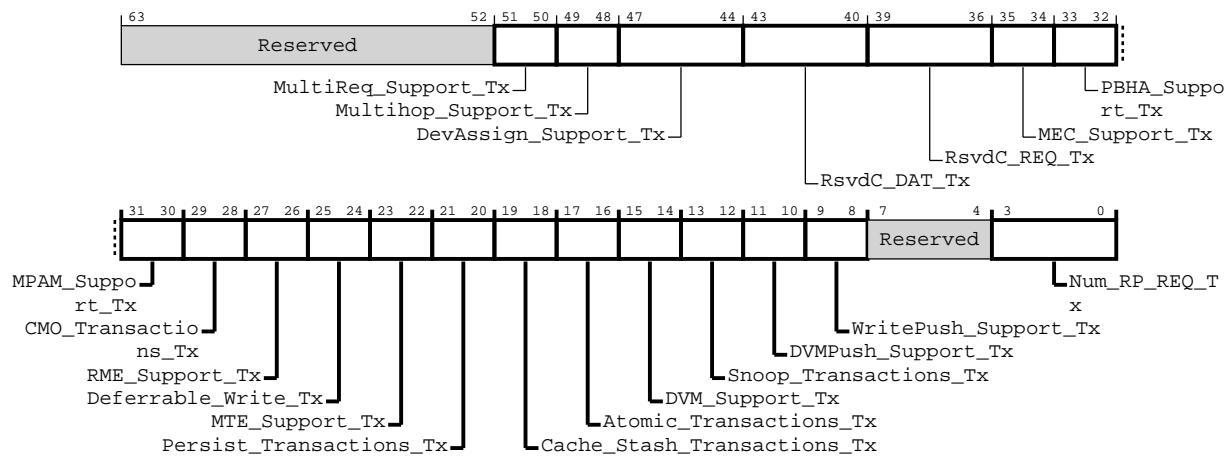
##### Usage constraints

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 8-340: por\_c2capb\_c2c\_linkproperties\_negotiated\_tx1\_0-0**



**Table 8-343: por\_c2capb\_c2c\_linkproperties\_negotiated\_tx1\_0-0 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:50]	MultiReq_Support_Tx	multireq support	RW	0b0000
[49:48]	Multihop_Support_Tx	multihop support	RW	0b00
[47:44]	DevAssign_Support_Tx	Support for RME-DA or RME-CDA. Only valid when RME_Support_Tx is True. Provided for informational purposes only, to assist with setup or debug. Not expected to have any functional effect in isolation. Each bit defines the roles of the other chip that this chip can connect to:  Bit 0 Host 0 Cannot connect to a host. 1 Can connect to a host. Bit 1 Device_StreamID_SecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Can make use of SecSID1 and StreamID fields. Bit 2 Device_NoStreamID_NoSecSID1 0 Cannot connect to a device. Does not use SecSID1 and StreamID fields. 1 Can connect to a device. Does not use SecSID1 and StreamID fields. Bit 3 Reserved	RW	0b0001
[43:40]	RsvdC_DAT_Tx	Width of RSVDC on the DAT channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits Others Reserved	RW	0b0000
[39:36]	RsvdC_REQ_Tx	Width of RSVDC on the REQ channel on the receiver chip:  0b000 0 bits 0b001 4 bits 0b010 8 bits 0b011 12 bits 0b100 16 bits 0b101 24 bits 0b110 32 bits 0b111 Reserved	RW	Configuration dependent
[35:34]	MEC_Support_Tx	Support for MEC on the receiver chip:  0b00 False - MECID field not present 0b01 True - MECID field present Others Reserved	RW	Configuration dependent
[33:32]	PBHA_Support_Tx	Support for PBHA on the receiver chip:  0b00 False - PBHA field not present 0b01 True - PBHA field present Others Reserved	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[31:30]	MPAM_Support_Tx	Support for MPAM on the receiver chip:  0b00 False - MPAM field not present 0b01 True - MPAM field present Others Reserved	RW	Configuration dependent
[29:28]	CMO_Transactions_Tx	Support for CMO transactions as a receiver:  0b00 False Will not send CMO transactions. 0b01 True Can send CMO transactions. Others Reserved	RW	0b01
[27:26]	RME_Support_Tx	Support for the Realm Management Extensions, including Realm and Root Physical Address Spaces (PAS) and PoPA CMOs as a receiver:  0b00 False RME transactions not supported. 0b01 True RME transactions supported. Others Reserved	RW	0b01
[25:24]	Deferrable_Write_Tx	Support for WriteNoSnpDef as a receiver:  0b00 False Deferrable Write transactions not supported. 0b01 True Others Reserved	RW	0b01
[23:22]	MTE_Support_Tx	Support for MTE as a receiver:  0b00 False MTE not supported. 0b01 Reduced MTE transaction support. 0b10 Full MTE transaction support. 0b11 Reserved	RW	0b00
[21:20]	Persist_Transactions_Tx	Support for CMO operations to the PoP and PoDP as a receiver:  0b00 False Persist CMO transactions not supported. 0b01 True Persist CMO transactions supported. Others Reserved	RW	0b01
[19:18]	Cache_Stash_Transactions_Tx	Support for Stashing requests and snoops as a receiver:  0b00 False Stash transactions not supported. 0b01 True Stash transactions supported. Others Reserved	RW	0b00
[17:16]	Atomic_Transactions_Tx	Support for Atomic transactions as a receiver:  0b00 False Atomic transactions not supported. 0b01 True Atomic transactions supported. Others Reserved	RW	0b01
[15:14]	DVM_Support_Tx	Support for DVM messages as a receiver:  0b00 False DVM transactions not supported. 0b01 True DVM transactions supported. Others Reserved	RW	0b01
[13:12]	Snoop_Transactions_Tx	Support for Snoop transactions as a receiver:  0b00 False Snoop transactions not supported. 0b01 True Snoop transactions supported. Others Reserved	RW	0b00
[11:10]	DVMPush_Support_Tx	Support for WritePush for DVM transactions by transactions by the interface as a receiver:  0b00 False WritePush for DVM transactions is not supported. 0b01 True WritePush for DVM transactions is supported. Others Reserved	RW	0b01

Bits	Name	Description	Type	Reset
[9:8]	WritePush_Support_Tx	Support for WritePush for Immediate writes and Atomics by the interface as a receiver:  0b00 False WritePush for Immediate writes and Atomics is not supported. 0b01 True WritePush for Immediate writes and Atomics is supported. Others Reserved	RW	0b01
[7:4]	Reserved	Reserved	RW	0x0
[3:0]	Num_RP_REQ_Tx	Number of resource planes the interface supports as a receiver on the REQ channel:  0b0000 1 0b0001 2 0b0010 3 0b0011 4 0b0100 5 0b0101 6 0b0110 7 0b0111 8 Others Reserved	RW	0b0001

### 8.3.6.71 por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit1

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2388

##### Type

RW

##### Reset value

See individual bit resets

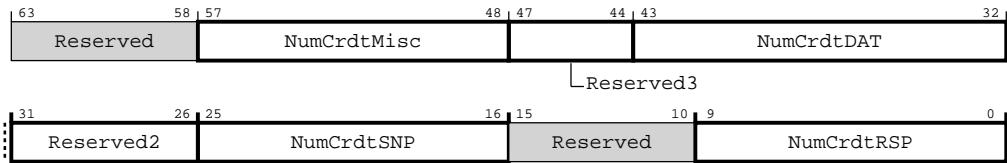
##### Usage constraints

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 8-341: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit1**



**Table 8-344: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit1 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	NumCrdtMisc	Max Number of Misc Credits allocated to this link end.	RW	0x0
[47:44]	Reserved3	Reserved	RW	0x0
[43:32]	NumCrdtDAT	Max Number of Data Credits allocated to this link end.	RW	Configuration dependent
[31:26]	Reserved2	Reserved	RW	0x0
[25:16]	NumCrdtSNP	Max Number of Snoop Credits allocated to this link end.	RW	Configuration dependent
[15:10]	Reserved	Reserved	RW	0x0
[9:0]	NumCrdtRSP	Max Number of Response Credits allocated to this link end. Num Crdt RSP[15:10] are reserved.	RW	0x100

### 8.3.6.72 por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit2

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 2

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2390

##### Type

RW

##### Reset value

See individual bit resets

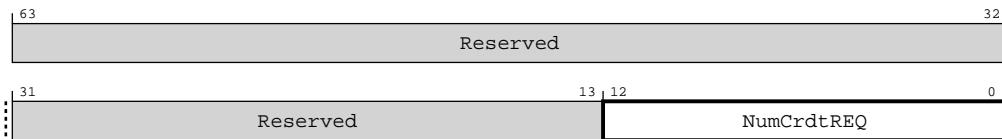
## Usage constraints

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 8-342: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit2**



**Table 8-345: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit2 attributes**

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12:0]	NumCrdtREQ	Number of Total Request Credits allocated to this link end. Num Crdt REQ[15:13] are reserved.	RW	Configuration dependent

## 8.3.6.73 por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit3

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x2398

### Type

RW

### Reset value

See individual bit resets

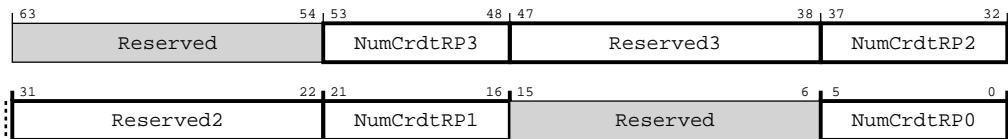
## Usage constraints

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 8-343: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit3**



**Table 8-346: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit3 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53:48]	NumCrdtRP3	Number REQ MC credits allocated to RP3. NumCrdtRPO[15:6] are reserved.	RW	0x0
[47:38]	Reserved3	Reserved	RW	0x0
[37:32]	NumCrdtRP2	Number REQ MC credits allocated to RP2. NumCrdtRPO[15:6] are reserved.	RW	0x0
[31:22]	Reserved2	Reserved	RW	0x0
[21:16]	NumCrdtRP1	Number REQ MC credits allocated to RP1. NumCrdtRPO[15:6] are reserved.	RW	0x1
[15:6]	Reserved	Reserved	RW	0x0
[5:0]	NumCrdtRPO	Number REQ MC credits allocated to RP0. NumCrdtRPO[15:6] are reserved.	RW	0x0

## 8.3.6.74 por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit4

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x23A0

#### Type

RW

#### Reset value

See individual bit resets

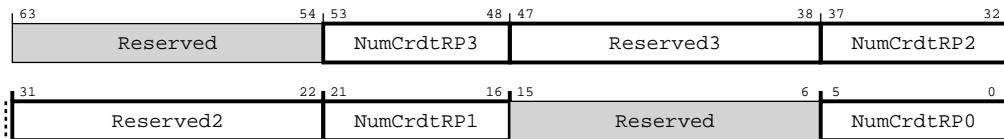
## Usage constraints

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 8-344: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit4**



**Table 8-347: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit4 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53:48]	NumCrdtRP3	Number REQ MC credits allocated to RP3. NumCrdtRPO[15:6] are reserved.	RW	0x0
[47:38]	Reserved3	Reserved	RW	0x0
[37:32]	NumCrdtRP2	Number REQ MC credits allocated to RP2. NumCrdtRPO[15:6] are reserved.	RW	0x0
[31:22]	Reserved2	Reserved	RW	0x0
[21:16]	NumCrdtRP1	Number REQ MC credits allocated to RP1. NumCrdtRPO[15:6] are reserved.	RW	0x0
[15:6]	Reserved	Reserved	RW	0x0
[5:0]	NumCrdtRPO	Number REQ MC credits allocated to RPO. NumCrdtRPO[15:6] are reserved.	RW	0x0

## 8.3.6.75 por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit5

There are 1 iterations of this register. The index ranges from 0 to 0. C2C Logical Linkend MC Credit 1

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x23A8

### Type

RW

### Reset value

See individual bit resets

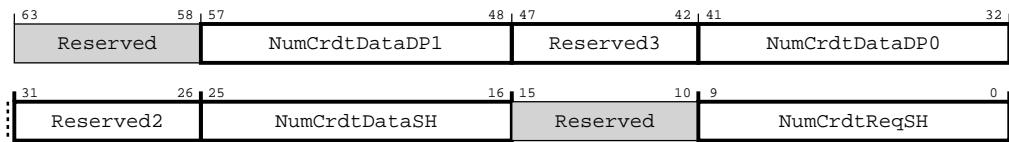
### Usage constraints

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 8-345: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit5**



**Table 8-348: por\_c2capb\_c2c\_logical\_linkend\_0-0\_mc\_credit5 attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	NumCrdtDataDP1	Number REQ MC credits allocated to Data DP1. NumCrdtRPO[15:6] are reserved.	RW	0x1
[47:42]	Reserved3	Reserved	RW	0x0
[41:32]	NumCrdtDataDP0	Number REQ MC credits allocated to RP2. NumCrdtRPO[15:6] are reserved.	RW	0x1
[31:26]	Reserved2	Reserved	RW	0x0
[25:16]	NumCrdtDataSH	Number DAT MC credits allocated to Shared. NumCrdtRPO[15:6] are reserved.	RW	Configuration dependent
[15:10]	Reserved	Reserved	RW	0x0
[9:0]	NumCrdtReqSH	Number REQ MC credits allocated to Shared. NumCrdtRPO[15:6] are reserved.	RW	Configuration dependent

### 8.3.7 DN register summary

The following table describes the registers for the relevant component.

**Table 8-349: por\_dn\_cfg register summary**

Offset	Name	Type	Description
0x0	por_dn_node_info	RO	<a href="#">por_dn_node_info</a>
0x80	por_dn_child_info	RO	<a href="#">por_dn_child_info</a>
0x900	por_dn_build_info	RO	<a href="#">por_dn_build_info</a>
0x980	por_dn_rcr	RW	<a href="#">por_dn_rcr</a>
0x988	por_dn_scr	RW	<a href="#">por_dn_scr</a>
0xA00	por_dn_cfg_ctl	RW	<a href="#">por_dn_cfg_ctl</a>

Offset	Name	Type	Description
0xA08	por_dn_aux_ctl	RW	<a href="#">por_dn_aux_ctl</a>
0xC00 : 0xF48	por_dn_vmf0-15_ctrl	RW	<a href="#">por_dn_vmf0-15_ctrl</a>
0xC08 : 0xF50	por_dn_vmf0-15_rnf0	RW	<a href="#">por_dn_vmf0-15_rnf0</a>
0xC10 : 0xF58	por_dn_vmf0-15_rnf1	RW	<a href="#">por_dn_vmf0-15_rnf1</a>
0xC18 : 0xF60	por_dn_vmf0-15_rnf2	RW	<a href="#">por_dn_vmf0-15_rnf2</a>
0xC20 : 0xF68	por_dn_vmf0-15_rnf3	RW	<a href="#">por_dn_vmf0-15_rnf3</a>
0xC28 : 0xF70	por_dn_vmf0-15_rnd0	RW	<a href="#">por_dn_vmf0-15_rnd0</a>
0xC30 : 0xF78	por_dn_vmf0-15_cxra	RW	<a href="#">por_dn_vmf0-15_cxra</a>
0xF80 : 0xF98	por_dn_domain_rnf0-3	RW	<a href="#">por_dn_domain_rnf0-3</a>
0xFA0	por_dn_domain_rnd0	RW	<a href="#">por_dn_domain_rnd0</a>
0xFA8	por_dn_domain_cxra	RW	<a href="#">por_dn_domain_cxra</a>
0xFB0 : 0x1028	por_dn_vmf0-15_rnd1	RW	<a href="#">por_dn_vmf0-15_rnd1</a>
0x1030	por_dn_domain_rnd1	RW	<a href="#">por_dn_domain_rnd1</a>
0x1038	por_dn_domain_cxra_cc	RW	<a href="#">por_dn_domain_cxra_cc</a>
0x1040 : 0x1388	por_dn_vmf16-31_ctrl	RW	<a href="#">por_dn_vmf16-31_ctrl</a>
0x1048 : 0x1390	por_dn_vmf16-31_rnf0	RW	<a href="#">por_dn_vmf16-31_rnf0</a>
0x1050 : 0x1398	por_dn_vmf16-31_rnf1	RW	<a href="#">por_dn_vmf16-31_rnf1</a>
0x1058 : 0x13A0	por_dn_vmf16-31_rnf2	RW	<a href="#">por_dn_vmf16-31_rnf2</a>
0x1060 : 0x13A8	por_dn_vmf16-31_rnf3	RW	<a href="#">por_dn_vmf16-31_rnf3</a>
0x1068 : 0x13B0	por_dn_vmf16-31_rnd0	RW	<a href="#">por_dn_vmf16-31_rnd0</a>
0x1070 : 0x13B8	por_dn_vmf16-31_cxra	RW	<a href="#">por_dn_vmf16-31_cxra</a>
0x13C0 : 0x1438	por_dn_vmf16-31_rnd1	RW	<a href="#">por_dn_vmf16-31_rnd1</a>
0xD900	por_dn_pmu_event_sel	RW	<a href="#">por_dn_pmu_event_sel</a>

### 8.3.7.1 por\_dn\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

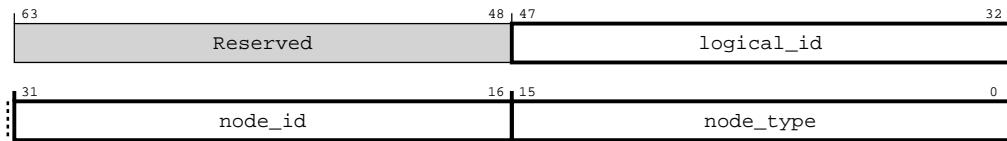
## Usage constraints

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 8-346: por\_dn\_node\_info**



**Table 8-350: por\_dn\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0001

## 8.3.7.2 por\_dn\_child\_info

Provides component child identification information.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x80

#### Type

RO

#### Reset value

See individual bit resets

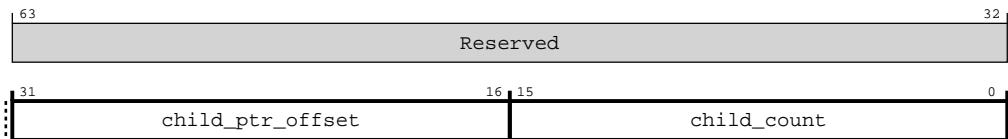
## Usage constraints

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 8-347: por\_dn\_child\_info**



**Table 8-351: por\_dn\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0x0

### 8.3.7.3 por\_dn\_build\_info

Contains the configuration parameter values. Indicates the specific DN configuration.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x900

### Type

RO

### Reset value

See individual bit resets

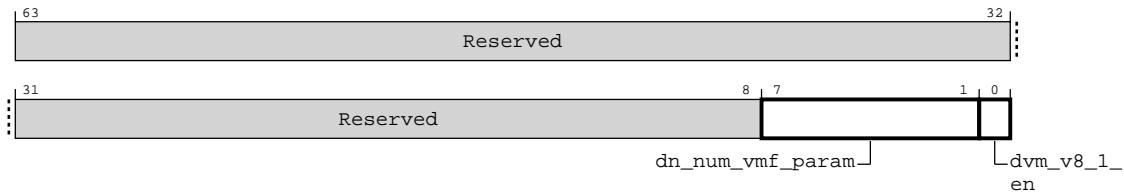
### Usage constraints

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 8-348: por\_dn\_build\_info**



**Table 8-352: por\_dn\_build\_info attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:1]	dn_num_vmf_param	Indicates the value of the DN_NUM_VMF_PARAM, with legal values of 4, 16, or 32.	RO	Configuration dependent
[0]	dvm_v8_1_en	Indicates that all nodes receiving DVM snoops support DVM v8/v8.1 operations.	RO	0b1

### 8.3.7.4 por\_dn\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

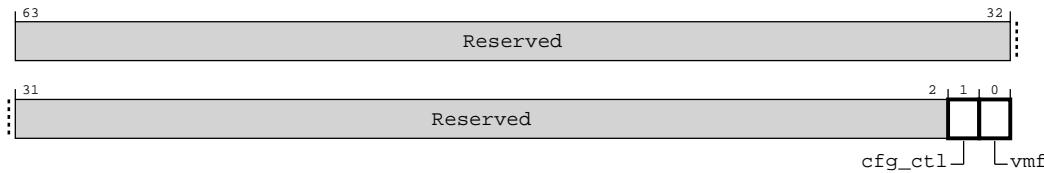
#### Usage constraints

This register is owned in the Root space and is 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 8-349: por\_dn\_rcr**



**Table 8-353: por\_dn\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0
[0]	vmf	Allows Root override of the secure VMF registers	RW	0b0

### 8.3.7.5 por\_dn\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

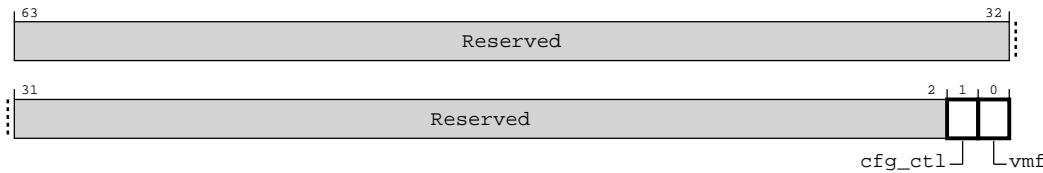
##### Usage constraints

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 8-350: por\_dn\_scr**



**Table 8-354: por\_dn\_scr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0
[0]	vmf	Allows Secure override of the VMF registers	RW	0b0

### 8.3.7.6 por\_dn\_cfg\_ctl

Functions as the configuration control register for DVM Node.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.cfg\_ctl

##### Secure group override

por\_dn\_scr.cfg\_ctl

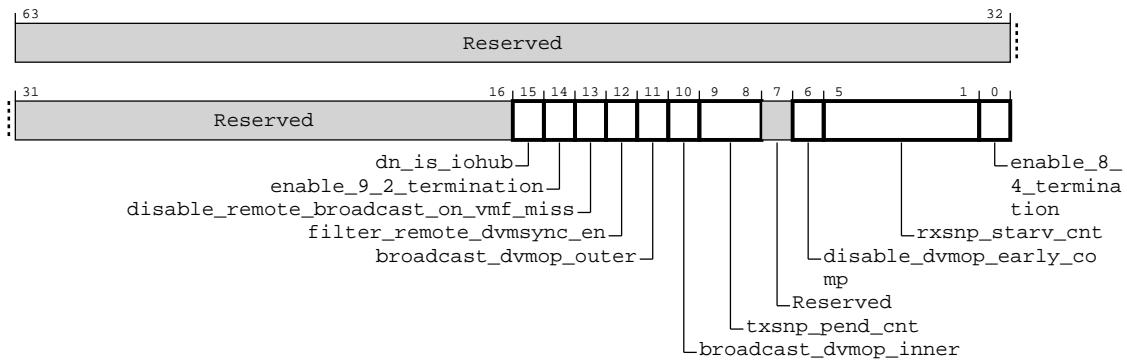
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.cfg\_ctl bit and por\_dn\_rcr.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 8-351: por\_dn\_cfg\_ctl**



**Table 8-355: por\_dn\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15]	dn_is_iohub	Indicates to the DN that it is in an IO Hub. DVMOp requests from Compute Chiplet CCGs are treated as local reqs	RW	0b0
[14]	enable_9_2_termination	Enables termination of v9.2 DVMOps in DN.	RW	0b0
[13]	disable_remote_broadcast_on_vmf_miss	Disables broadcast of VMID Filterable DVMOps to remote on VMF miss.	RW	0x0
[12]	filter_remote_dvmsync_en	Enables logic which filters redundant DVMSyncs to remote domains.	RW	0x0
[11]	broadcast_dvmop_outer	Used to filter DVMOps marked as outershareable (OS) from being sent off-chip.	RW	0x1
[10]	broadcast_dvmop_inner	Used to filter DVMOps marked as innershareable (IS) from being sent off-chip.	RW	0x1
[9:8]	txsnp_pend_cnt	Maximum number of (Non-Sync + Sync) SnpDVMOps issued on TXSNP. <b>0b00</b> Max of 4 SnpDVMOps in progress (default) <b>0b01</b> Max of 8 SnpDVMOps in progress <b>0b10</b> Max of 16 SnpDVMOps in progress <b>0b11</b> Reserved	RW	0x10
[7]	Reserved	Reserved	RO	
[6]	disable_dvmop_early_comp	Disables Early Comp (CompDBID) for DVMOps	RW	0b0
[5:1]	rxsn_starv_cnt	Credit steal threshold. The number of cycles RXSNP loses to RXREQ before credit stealing occurs. Must be set to non-zero value - a value of 0 is treated as the default value of 8.	RW	0x8
[0]	enable_8_4_termination	Enables termination of 8.4 DVMOps in DN.	RW	0b0

### 8.3.7.7 por\_dn\_aux\_ctl

Functions as the auxiliary control register for DN.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

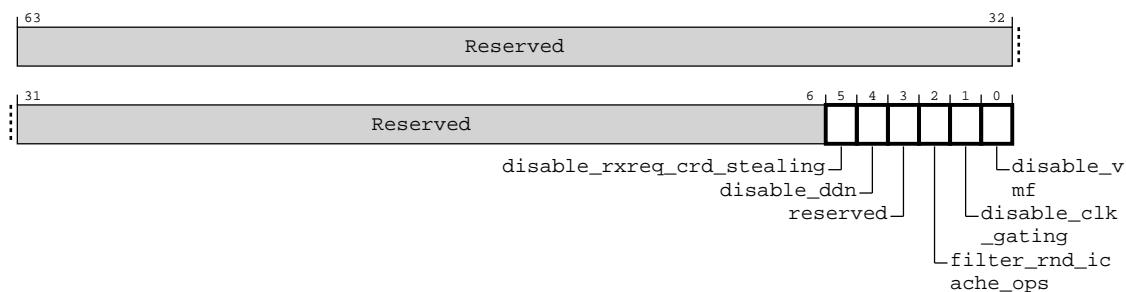
##### Usage constraints

This register is owned in the Root space and is 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 8-352: por\_dn\_aux\_ctl**



**Table 8-356: por\_dn\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	
[5]	disable_rxreq_crd_stealing	Disables credit stealing from RXREQ LinkLayer when RXSNP is starved for RCB alloc.	RW	0b0
[4]	disable_ddn	Disables Distributed DN functionality- Snoops all RNs and CML nodes in the mesh and disables snooping other DNs. Must program all RNSAMs to target HND for DVMs and then set this to 1 in HND.	RW	0b0
[3]	reserved	Reserved field	RW	0b0

Bits	Name	Description	Type	Reset
[2]	filter_rnd_icache_ops	Filters out BPI and VICI/PICI Snps to RNDs when set	RW	0x0
[1]	disable_clk_gating	Disables autonomous clock gating when set	RW	0b0
[0]	disable_vmf	Disables VMID-based DVM snoop filtering when set	RW	0x0

### 8.3.7.8 por\_dn\_vmf0-15\_ctrl

There are 16 iterations of this register. The index ranges from 0 to 15. Functions as the control register for VMID-based DVM snoop filtering. NOTE: This register has no effect when por\_dn\_aux\_ctl.disable\_vmf is set to 1.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{56\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

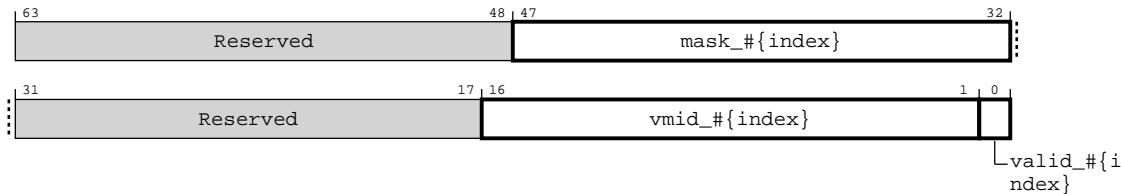
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-353: por\_dn\_vmf0-15\_ctrl**



**Table 8-357: por\_dn\_vmf0-15\_ctrl attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	mask_{index}	VMID mask; enables mapping of multiple VMID values to a single register  <b>NOTE</b> Logically, the AND operator is performed on the mask and por_dn_vmf#{index}_ctrl.vmid. Then, the AND operator is performed on the mask and the incoming request's VMID. The two results are then compared, and filtering is applied to the incoming request if the masked VMIDs match.	RW	0xffffffff
[31:17]	Reserved	Reserved	RO	-
[16:1]	vmid_{index}	VMID value  <b>NOTE</b> The incoming request's VMID is only compared with this VMID value if the request's VMID valid bit is set. If the request's VMID is valid and the two VMIDs match, filtering is applied to the incoming request.	RW	0x0000
[0]	valid_{index}	Register valid  <b>0b1</b> Register is enabled  <b>0b0</b> Register is not enabled	RW	0b0

### 8.3.7.9 por\_dn\_vmf0-15\_rnf0

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

$0xC00 + \{56 * \text{index} + 8\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

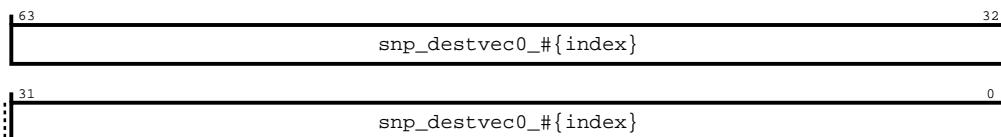
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-354: por\_dn\_vmf0-15\_rnf0**



**Table 8-358: por\_dn\_vmf0-15\_rnf0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0_{index}	RN-F bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.10 por\_dn\_vmf0-15\_rnf1

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 127:64 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xC00 + #{56\*index+16}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

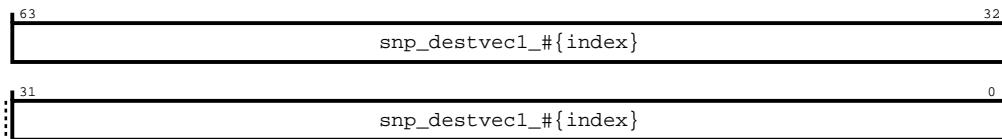
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-355: por\_dn\_vmf0-15\_rnf1**



**Table 8-359: por\_dn\_vmf0-15\_rnf1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1_{index}	RN-F bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.11 por\_dn\_vmf0-15\_rnf2

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 191:128 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{56\*index+24}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

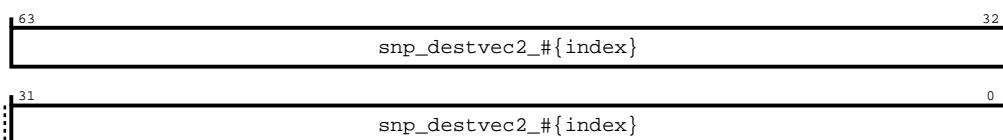
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-356: por\_dn\_vmf0-15\_rnf2**



**Table 8-360: por\_dn\_vmf0-15\_rnf2 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec2_#{index}	RN-F bit vector 191:128 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.12 por\_dn\_vmf0-15\_rnf3

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 255:192 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{56\*index+32}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-357: por\_dn\_vmf0-15\_rnf3**



**Table 8-361: por\_dn\_vmf0-15\_rnf3 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec3_{index}	RN-F bit vector 255:192 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.13 por\_dn\_vmf0-15\_rnd0

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-D bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{56\*index+40}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

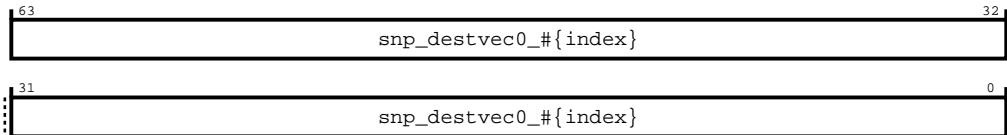
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-358: por\_dn\_vmf0-15\_rnd0**



**Table 8-362: por\_dn\_vmf0-15\_rnd0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0_{index}	RN-D bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.14 por\_dn\_vmf0-15\_cxra

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical CXRA bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering. NOTE: Not applicable in a single-chip CMN system. Does not have any effect.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{56\*index+48}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

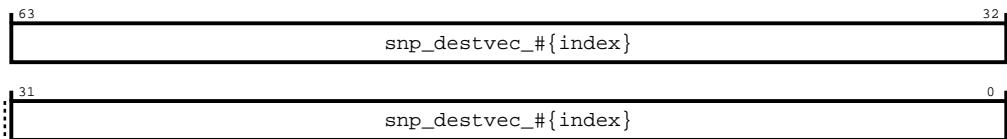
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-359: por\_dn\_vmf0-15\_cxra**



**Table 8-363: por\_dn\_vmf0-15\_cxra attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec_{index}	CXRA bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

## 8.3.7.15 por\_dn\_domain\_rnf0-3

There are 4 iterations of this register. The index ranges from 0 to 3. RN-F logical list for DDN

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xF80 + #{8\*index}

#### Type

RW

#### Reset value

See individual bit resets

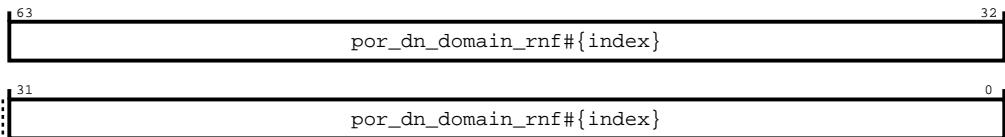
#### Usage constraints

This register is owned in the Root space and is 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 8-360: por\_dn\_domain\_rnf0-3**



**Table 8-364: por\_dn\_domain\_rnf0-3 attributes**

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnf#{index}	RN-F logical list corresponding to RN-F #{{((index+1)64)-1}:#{index64}}	RW	Configuration dependent

### 8.3.7.16 por\_dn\_domain\_rnd0

RND logical list for DDN

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xFA0

##### Type

RW

##### Reset value

See individual bit resets

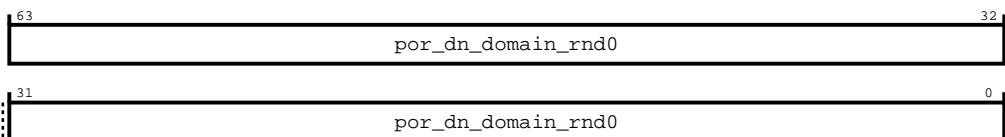
##### Usage constraints

This register is owned in the Root space and is 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 8-361: por\_dn\_domain\_rnd0**



**Table 8-365: por\_dn\_domain\_rnd0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnd0	RND logical list for DDN corresponding to RND 63:0	RW	Configuration dependent

### 8.3.7.17 por\_dn\_domain\_cxra

CXRA logical list for DDN

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xFA8

##### Type

RW

##### Reset value

See individual bit resets

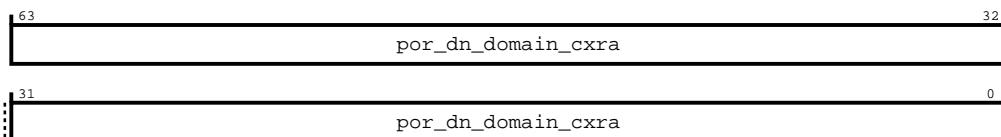
##### Usage constraints

This register is owned in the Root space and is 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 8-362: por\_dn\_domain\_cxra**



**Table 8-366: por\_dn\_domain\_cxra attributes**

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_cxra	CXRA logical list for DDN	RW	Configuration dependent

### 8.3.7.18 por\_dn\_vmf0-15\_rnd1

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-D bit vector 127:64 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xFB0 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

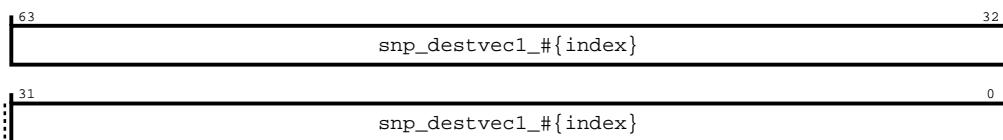
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-363: por\_dn\_vmf0-15\_rnd1**



**Table 8-367: por\_dn\_vmf0-15\_rnd1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1_#{index}	RN-D bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.19 por\_dn\_domain\_rnd1

RND logical list for DDN

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1030

##### Type

RW

##### Reset value

See individual bit resets

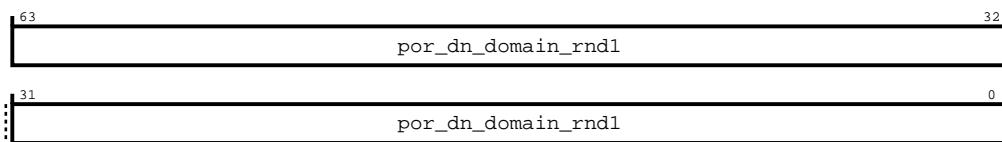
##### Usage constraints

This register is owned in the Root space and is 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 8-364: por\_dn\_domain\_rnd1**



**Table 8-368: por\_dn\_domain\_rnd1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnd1	RND logical list for DDN corresponding to RND 127:64	RW	Configuration dependent

### 8.3.7.20 por\_dn\_domain\_cxra\_cc

IOHub CXRA Compute Chiplet logical list for DDN

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1038

##### Type

RW

##### Reset value

See individual bit resets

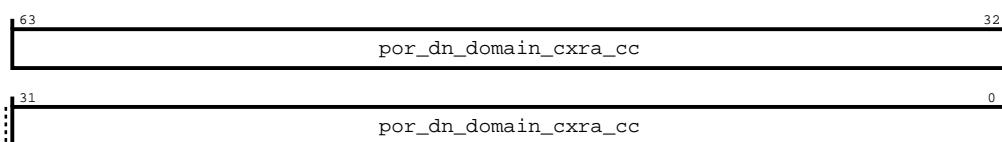
##### Usage constraints

This register is owned in the Root space and is 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 8-365: por\_dn\_domain\_cxra\_cc**



**Table 8-369: por\_dn\_domain\_cxra\_cc attributes**

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_cxra_cc	This register is only consumed by IO Hub DNs. It allows IO Hub DN to differentiate between CXRAs connected to Compute Chiplets and other IO Hubs. CXRA compute chiplet logical list for DDN	RW	0b0

### 8.3.7.21 por\_dn\_vmf16-31\_ctrl

There are 16 iterations of this register. The index ranges from 16 to 31. Functions as the control register for VMID-based DVM snoop filtering. NOTE: This register has no effect when por\_dn\_aux\_ctl.disable\_vmf is set to 1.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1040 + #{56\*(index-16)}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

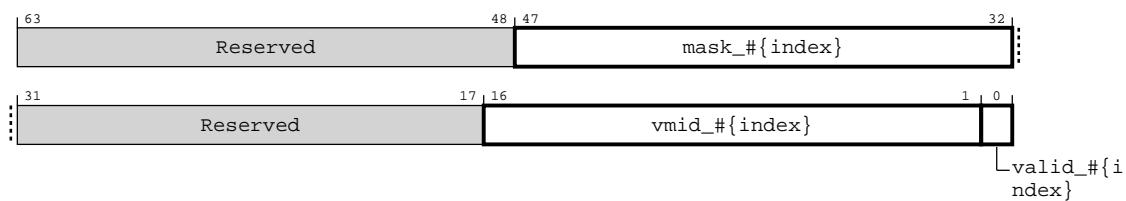
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-366: por\_dn\_vmf16-31\_ctrl**



**Table 8-370: por\_dn\_vmf16-31\_ctrl attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	mask_{index}	VMID mask; enables mapping of multiple VMID values to a single register  <b>NOTE</b> Logically, the AND operator is performed on the mask and por_dn_vmf#{index}_ctrl.vmid. Then, the AND operator is performed on the mask and the incoming request's VMID. The two results are then compared, and filtering is applied to the incoming request if the masked VMIDs match.	RW	0xFFFF
[31:17]	Reserved	Reserved	RO	-
[16:1]	vmid_{index}	VMID value  <b>NOTE</b> The incoming request's VMID is only compared with this VMID value if the request's VMID valid bit is set. If the request's VMID is valid and the two VMIDs match, filtering is applied to the incoming request.	RW	0x0000
[0]	valid_{index}	Register valid  <b>0b1</b> Register is enabled  <b>0b0</b> Register is not enabled	RW	0b0

### 8.3.7.22 por\_dn\_vmf16-31\_rnf0

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-F bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1040 + #{56\*(index-16)+8}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

## Secure group override

por\_dn\_scr.vmf

## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rrc.vmf 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 8-367: por\_dn\_vmf16-31\_rnf0**



**Table 8-371: por\_dn\_vmf16-31\_rnf0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0_{index}	RN-F bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

## 8.3.7.23 por\_dn\_vmf16-31\_rnf1

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-F bit vector 127:64 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1040 + #{56\*(index-16)+16}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

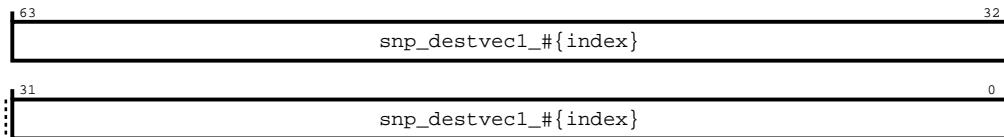
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-368: por\_dn\_vmf16-31\_rnf1**



**Table 8-372: por\_dn\_vmf16-31\_rnf1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1_{index}	RN-F bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.24 por\_dn\_vmf16-31\_rnf2

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-F bit vector 191:128 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x1040 + #{56\*(index-16)+24}

#### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

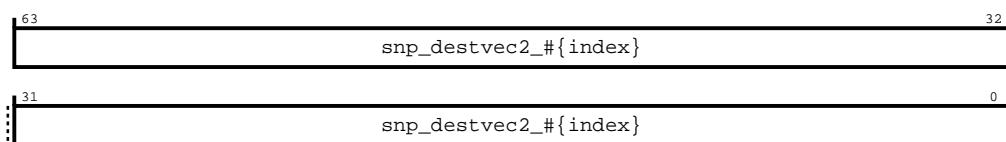
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-369: por\_dn\_vmf16-31\_rnf2**



**Table 8-373: por\_dn\_vmf16-31\_rnf2 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec2_{index}	RN-F bit vector 191:128 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.25 por\_dn\_vmf16-31\_rnf3

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-F bit vector 255:192 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x1040 + #{56\*(index-16)+32}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-370: por\_dn\_vmf16-31\_rnf3**



**Table 8-374: por\_dn\_vmf16-31\_rnf3 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec3_{index}	RN-F bit vector 255:192 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.26 por\_dn\_vmf16-31\_rnd0

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-D bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

0x1040 + #{56\*(index-16)+40}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

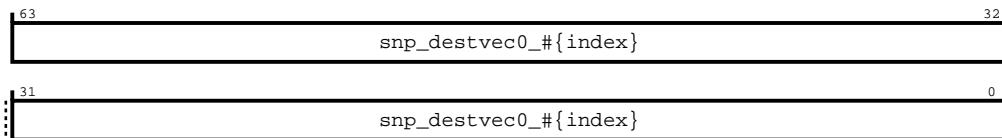
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-371: por\_dn\_vmf16-31\_rnd0**



**Table 8-375: por\_dn\_vmf16-31\_rnd0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0_{index}	RN-D bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.27 por\_dn\_vmf16-31\_cxra

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical CXRA bit vector 63:0 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering. NOTE: Not applicable in a single-chip CMN system. Does not have any effect.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1040 + #{56\*(index-16)+48}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_dn\_rcr.vmf

### Secure group override

por\_dn\_scr.vmf

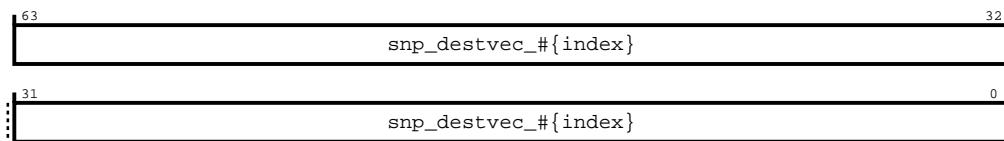
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-372: por\_dn\_vmf16-31\_cxra**



**Table 8-376: por\_dn\_vmf16-31\_cxra attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec_{index}	CXRA bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.28 por\_dn\_vmf16-31\_rnd1

There are 16 iterations of this register. The index ranges from 16 to 31. Contains the logical RN-D bit vector 127:64 corresponding to por\_dn\_vmf#{index}\_ctrl.vmid. Used for VMID-based DVM snoop filtering.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x13C0 + #{8\*(index-16)}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_dn\_rcr.vmf

##### Secure group override

por\_dn\_scr.vmf

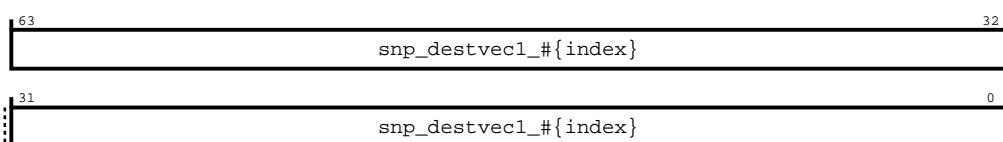
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_dn\_scr.vmf bit is set, Secure accesses to this register are permitted. If both the por\_dn\_scr.vmf bit and por\_dn\_rcr.vmf 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 8-373: por\_dn\_vmf16-31\_rnd1**



**Table 8-377: por\_dn\_vmf16-31\_rnd1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1_#{index}	RN-D bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	0b0

### 8.3.7.29 por\_dn\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

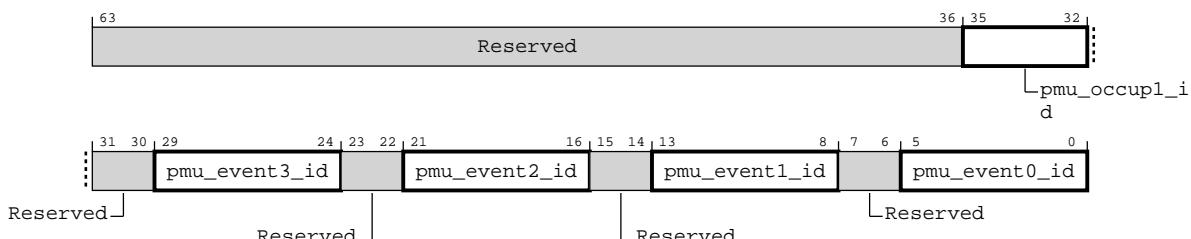
##### Usage constraints

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 8-374: por\_dn\_pmu\_event\_sel**



**Table 8-378: por\_dn\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[35:32]	pmu_occup1_id	PMU occupancy event selector ID  <b>0b0000</b> All  <b>0b0001</b> DVM ops  <b>0b0010</b> DVM syncs	RW	0b0
[31:30]	Reserved	Reserved	RO	
[29:24]	pmu_event3_id	PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[23:22]	Reserved	Reserved	RO	
[21:16]	pmu_event2_id	PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[15:14]	Reserved	Reserved	RO	
[13:8]	pmu_event1_id	PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>PMU Event 0 ID</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> Number of TLBI DVM op requests</p> <p><b>0x02</b> Number of BPI DVM op requests</p> <p><b>0x03</b> Number of PICI DVM op requests</p> <p><b>0x04</b> Number of VICI DVM op requests</p> <p><b>0x05</b> Number of DVM sync requests</p> <p><b>0x06</b> Number of DVM op requests that were filtered using VMID filtering</p> <p><b>0x07</b> Number of DVM op requests to RNDs, BPI or PICI/VICI, that were filtered</p> <p><b>0x08</b> Number of retried REQ</p> <p><b>0x09</b> Number of SNPs sent to RNs</p> <p><b>0xa</b> Number of SNPs stalled to RNs due to lack of Crds</p> <p><b>0xb</b> DVM tracker full counter</p> <p><b>0xc</b> DVM RN-F tracker occupancy counter</p> <p><b>0xd</b> DVM CXHA tracker occupancy counter</p> <p><b>0xe</b> DVM Peer DN tracker occupancy counter</p> <p><b>0xf</b> DVM RN-F tracker Alloc</p> <p><b>0x10</b> DVM CXHA tracker Alloc</p> <p><b>0x11</b> DVM Peer DN tracker Alloc</p> <p><b>0x12</b> TXSNP stall due to number outstanding limit</p> <p><b>0x13</b> RXSNP stall starvation threshold hit</p> <p><b>0x14</b> TXSNP SYNC stall due to outstanding early completed Op</p>	RW	0b0

## 8.3.8 DT register summary

The following table describes the registers for the relevant component.

**Table 8-379: por\_dt\_cfg register summary**

Offset	Name	Type	Description
0x0	por_dt_node_info	RO	por_dt_node_info
0x80	por_dt_child_info	RO	por_dt_child_info
0x980	por_dt_secure_access	RW	por_dt_secure_access
0x988	por_dt_pmscr	RW	por_dt_pmscr
0x990	por_dt_pmrcr	RW	por_dt_pmrcr
0xA00	por_dt_dtc_ctl	RW	por_dt_dtc_ctl
0xA10	por_dt_trigger_status	RO	por_dt_trigger_status
0xA20	por_dt_trigger_status_clr	WO	por_dt_trigger_status_clr
0xA30	por_dt_trace_control	RW	por_dt_trace_control
0xA48	por_dt_traceid	RW	por_dt_traceid
0xD900	por_dt_pmevcntAB	RW	por_dt_pmevcntAB
0xD910	por_dt_pmevcntCD	RW	por_dt_pmevcntCD
0xD920	por_dt_pmevcntEF	RW	por_dt_pmevcntEF
0xD930	por_dt_pmevcntGH	RW	por_dt_pmevcntGH
0xD940	por_dt_pmccntr	RW	por_dt_pmccntr
0xD950	por_dt_pmevcntsrAB	RW	por_dt_pmevcntsrAB
0xD960	por_dt_pmevcntsrCD	RW	por_dt_pmevcntsrCD
0xD970	por_dt_pmevcntsrEF	RW	por_dt_pmevcntsrEF
0xD980	por_dt_pmevcntsrGH	RW	por_dt_pmevcntsrGH
0xD990	por_dt_pmccntrsrsr	RW	por_dt_pmccntrsrsr
0xDA00	por_dt_pmcr	RW	por_dt_pmcr
0xDA18	por_dt_pmovsr	RO	por_dt_pmovsr
0xDA20	por_dt_pmovsr_clr	WO	por_dt_pmovsr_clr
0xDA28	por_dt_pmssr	RO	por_dt_pmssr
0xDA30	por_dt_pmsrr	WO	por_dt_pmsrr
0xFA0	por_dt_claim	RW	por_dt_claim
0xFA8	por_dt_devaff	RO	por_dt_devaff
0xFB0	por_dt_lsr	RO	por_dt_lsr
0xFB8	por_dt_authstatus_devarch	RO	por_dt_authstatus_devarch
0xFC0	por_dt_devid	RO	por_dt_devid
0xFC8	por_dt_devtype	RO	por_dt_devtype
0xFD0	por_dt_pidr45	RO	por_dt_pidr45
0xFD8	por_dt_pidr67	RO	por_dt_pidr67
0xFE0	por_dt_pidr01	RO	por_dt_pidr01
0xFE8	por_dt_pidr23	RO	por_dt_pidr23

Offset	Name	Type	Description
0xFF0	por_dt_cidr01	RO	por_dt_cidr01
0xFF8	por_dt_cidr23	RO	por_dt_cidr23

### 8.3.8.1 por\_dt\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

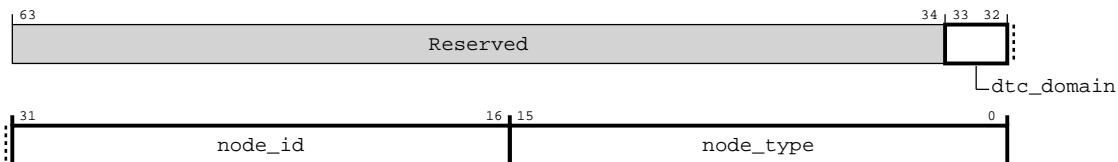
##### Usage constraints

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 8-375: por\_dt\_node\_info**



**Table 8-380: por\_dt\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33:32]	dtc_domain	DTC domain number	RO	0x00
[31:16]	node_id	Component CHI node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x3

### 8.3.8.2 por\_dt\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

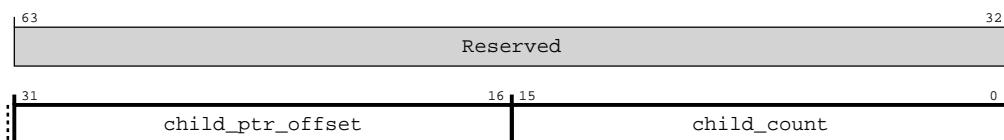
##### Usage constraints

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 8-376: por\_dt\_child\_info**



**Table 8-381: por\_dt\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0000
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.8.3 por\_dt\_secure\_access

Functions as the secure access control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

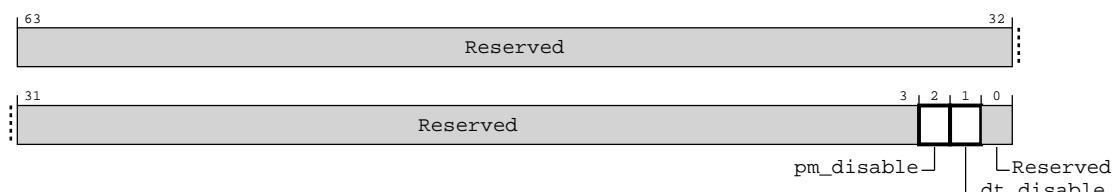
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-377: por\_dt\_secure\_access**



**Table 8-382: por\_dt\_secure\_access attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	pm_disable	PMU disable <b>0b0</b> PMU function is not affected <b>0b1</b> PMU function is disabled.	RW	0b0

Bits	Name	Description	Type	Reset
[1]	dt_disable	Debug disable  <b>0b0</b> DT function is not affected  <b>0b1</b> DT function is disabled.	RW	0b0
[0]	Reserved	Reserved	RO	

### 8.3.8.4 por\_dt\_pmscr

Functions as the secure PMU control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

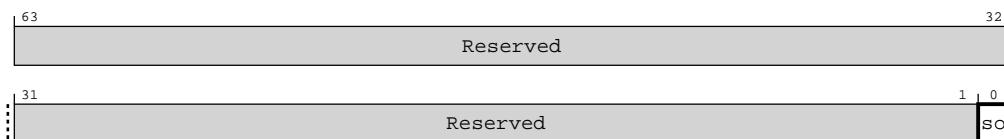
##### Usage constraints

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 8-378: por\_dt\_pmscr**



**Table 8-383: por\_dt\_pmscr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	so	Secure event observation <b>0b0</b> secure events are only monitored by the PMU if SPNIDEN is set to 1 <b>0b1</b> secure events are monitored by the PMU	RW	0b0

### 8.3.8.5 por\_dt\_pmrcr

Functions as the RME PMU control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x990

##### Type

RW

##### Reset value

See individual bit resets

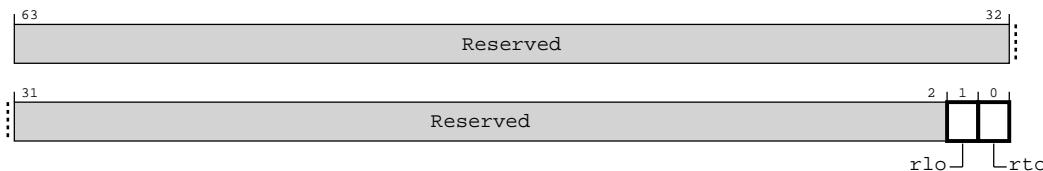
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-379: por\_dt\_pmrcr**



**Table 8-384: por\_dt\_pmrcr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[1]	rlo	Realm event observation <b>0b0</b> realm events are not monitored by the PMU <b>0b1</b> realm events are monitored by the PMU	RW	0b0
[0]	rto	Root event observation <b>0b0</b> root events are not monitored by the PMU <b>0b1</b> root events are monitored by the PMU	RW	0b0

### 8.3.8.6 por\_dt\_dtc\_ctl

Functions as the debug trace control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

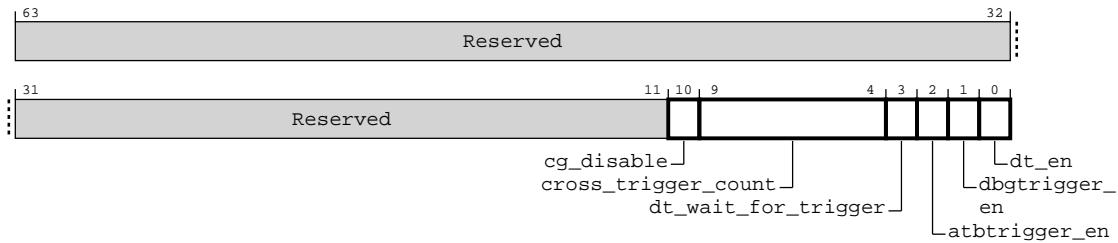
##### Usage constraints

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 8-380: por\_dt\_dtc\_ctl**



**Table 8-385: por\_dt\_dtc\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	cg_disable	Disables DT architectural clock gates	RW	0b0
[9:4]	cross_trigger_count	Number of cross triggers received before trace enable  <b>NOTE</b> Only applicable if dt_wait_for_trigger is set to 1.	RW	0b0
[3]	dt_wait_for_trigger	Enables waiting for cross trigger before trace enable	RW	0b0
[2]	atbtrigger_en	ATB trigger enable	RW	0b0
[1]	dbgtrigger_en	DBGWATCHTRIG enable	RW	0b0
[0]	dt_en	Enables debug, trace, and PMU features	RW	0b0

### 8.3.8.7 por\_dt\_trigger\_status

Provides the trigger status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA10

##### Type

RO

##### Reset value

See individual bit resets

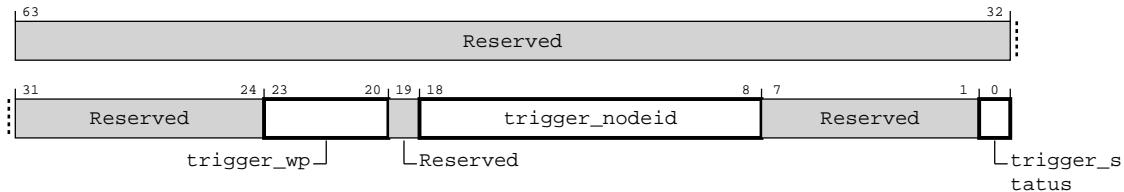
#### Usage constraints

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 8-381: por\_dt\_trigger\_status**



**Table 8-386: por\_dt\_trigger\_status attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:20]	trigger_wp	DBGWATCHTRIGREQ assertion and/or ATB trigger are caused by watchpoint	RO	0x0
[19]	Reserved	Reserved	RO	
[18:8]	trigger_nodeid	DBGWATCHTRIGREQ assertion and/or ATB trigger are caused by node ID	RO	0x0
[7:1]	Reserved	Reserved	RO	
[0]	trigger_status	Indicates DBGWATCHTRIGREQ assertion and/or ATB trigger	RO	0x0

### 8.3.8.8 por\_dt\_trigger\_status\_clr

Clears the trigger status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA20

##### Type

WO

##### Reset value

See individual bit resets

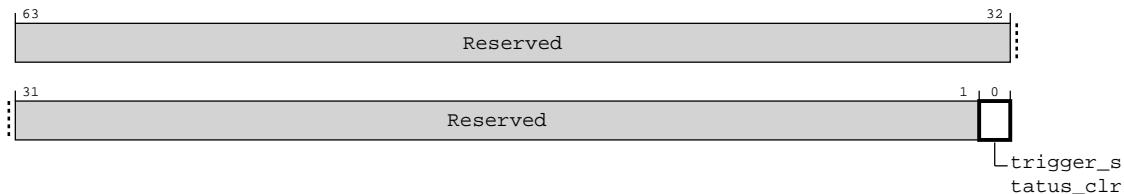
##### Usage constraints

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 8-382: por\_dt\_trigger\_status\_clr**



**Table 8-387: por\_dt\_trigger\_status\_clr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	trigger_status_clr	Write a 1 to clear por_dt_trigger_status.trigger_status	WO	0b0

### 8.3.8.9 por\_dt\_trace\_control

Functions as the trace control register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA30

### Type

RW

### Reset value

See individual bit resets

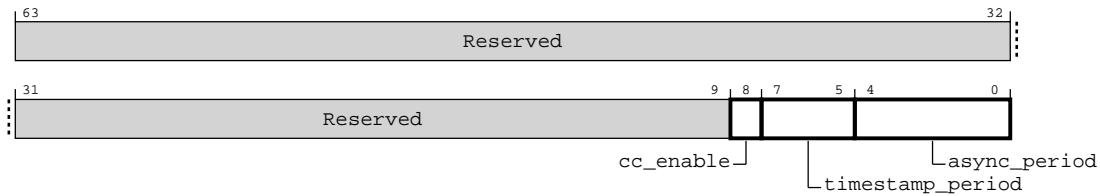
### Usage constraints

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 8-383: por\_dt\_trace\_control**



**Table 8-388: por\_dt\_trace\_control attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8]	cc_enable	Cycle count enable	RW	0b0
[7:5]	timestamp_period	Time stamp packet insertion period <b>0b000</b> Time stamp disabled <b>0b011</b> Time stamp every 8K clock cycles <b>0b100</b> Time stamp every 16K clock cycles <b>0b101</b> Time stamp every 32K clock cycles <b>0b110</b> Time stamp every 64K clock cycles	RW	0b0
[4:0]	async_period	Alignment sync packet insertion period <b>0x00</b> Alignment sync disabled <b>0x08</b> Alignment sync inserted after 256B of trace <b>0x09</b> Alignment sync inserted after 512B of trace <b>0x14</b> Alignment sync inserted after 1048576B of trace <b>NOTE</b> All other values are reserved.	RW	0b0

### 8.3.8.10 por\_dt\_traceid

Contains the ATB ID.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA48

### Type

RW

### Reset value

See individual bit resets

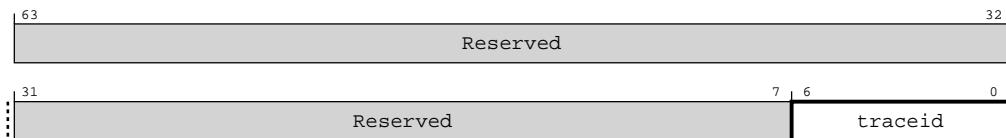
### Usage constraints

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 8-384: por\_dt\_traceid**



**Table 8-389: por\_dt\_traceid attributes**

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	
[6:0]	traceid	ATB ID	RW	0x0

## 8.3.8.11 por\_dt\_pmevcntAB

Contains the PMU event counters A and B.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD900

### Type

RW

### Reset value

See individual bit resets

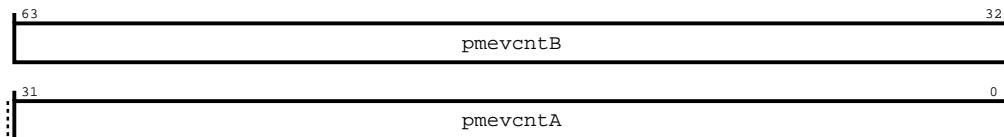
### Usage constraints

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 8-385: por\_dt\_pmevcntAB**



**Table 8-390: por\_dt\_pmevcntAB attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntB	PMU counter B	RW	0x0000
[31:0]	pmevcntA	PMU counter A	RW	0x0000

### 8.3.8.12 por\_dt\_pmevcntCD

Contains the PMU event counters C and D.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xD910

#### Type

RW

#### Reset value

See individual bit resets

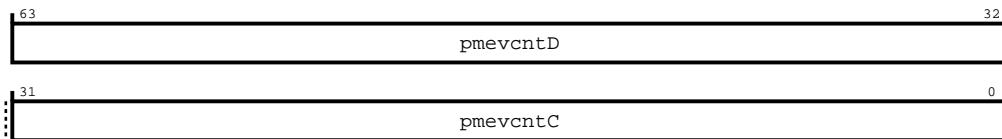
## Usage constraints

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 8-386: por\_dt\_pmevcntCD**



**Table 8-391: por\_dt\_pmevcntCD attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntD	PMU counter D	RW	0x0000
[31:0]	pmevcntC	PMU counter C	RW	0x0000

### 8.3.8.13 por\_dt\_pmevcntEF

Contains the PMU event counters E and F.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD920

### Type

RW

### Reset value

See individual bit resets

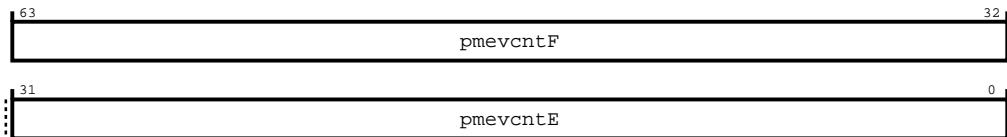
## Usage constraints

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 8-387: por\_dt\_pmevcntEF**



**Table 8-392: por\_dt\_pmevcntEF attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntF	PMU counter F	RW	0x0000
[31:0]	pmevcntE	PMU counter E	RW	0x0000

### 8.3.8.14 por\_dt\_pmevcntGH

Contains the PMU event counters G and H.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD930

##### Type

RW

##### Reset value

See individual bit resets

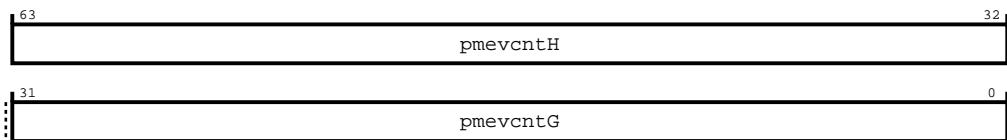
#### Usage constraints

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 8-388: por\_dt\_pmevcntGH**



**Table 8-393: por\_dt\_pmevcntGH attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntH	PMU counter H	RW	0x0000
[31:0]	pmevcntG	PMU counter G	RW	0x0000

### 8.3.8.15 por\_dt\_pmccntr

Contains the PMU cycle counter.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD940

##### Type

RW

##### Reset value

See individual bit resets

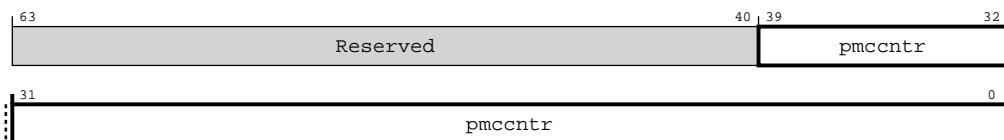
##### Usage constraints

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 8-389: por\_dt\_pmccntr**



**Table 8-394: por\_dt\_pmccntr attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:0]	pmccntr	PMU cycle counter	RW	0x0

### 8.3.8.16 por\_dt\_pmevcntsrAB

Contains the PMU event counter shadow registers A and B.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD950

##### Type

RW

##### Reset value

See individual bit resets

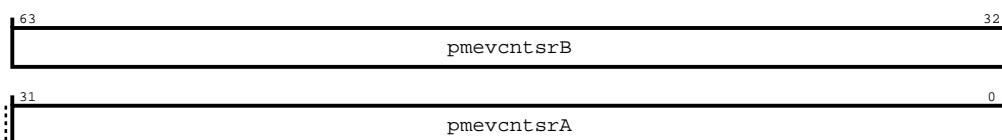
##### Usage constraints

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 8-390: por\_dt\_pmevcntsrAB**



**Table 8-395: por\_dt\_pmevcntsrAB attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrB	PMU counter B shadow register	RW	0x0000
[31:0]	pmevcntsrA	PMU counter A shadow register	RW	0x0000

### 8.3.8.17 por\_dt\_pmevcntsrCD

Contains the PMU event counter shadow registers C and D.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD960

### Type

RW

### Reset value

See individual bit resets

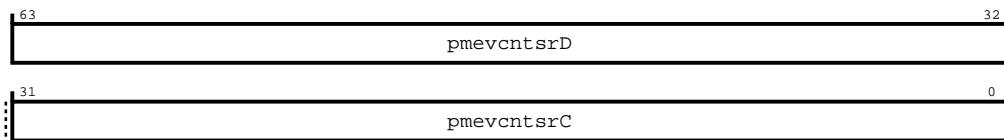
### Usage constraints

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 8-391: por\_dt\_pmevcntsrCD**



**Table 8-396: por\_dt\_pmevcntsrCD attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrD	PMU counter D shadow register	RW	0x0000
[31:0]	pmevcntsrC	PMU counter C shadow register	RW	0x0000

## 8.3.8.18 por\_dt\_pmevcntsrEF

Contains the PMU event counter shadow registers E and F.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD970

### Type

RW

### Reset value

See individual bit resets

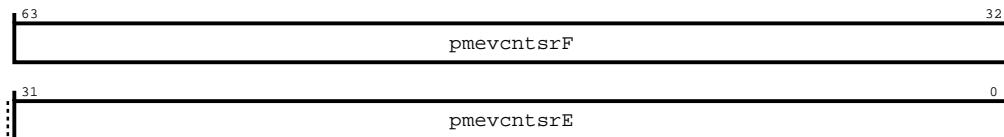
### Usage constraints

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 8-392: por\_dt\_pmevcntsrEF**



**Table 8-397: por\_dt\_pmevcntsrEF attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrF	PMU counter F shadow register	RW	0x0000
[31:0]	pmevcntsrE	PMU counter E shadow register	RW	0x0000

### 8.3.8.19 por\_dt\_pmevcntsrGH

Contains the PMU event counter shadow registers G and H.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xD980

#### Type

RW

#### Reset value

See individual bit resets

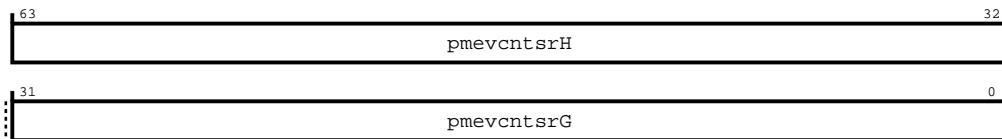
## Usage constraints

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 8-393: por\_dt\_pmevcntsrGH**



**Table 8-398: por\_dt\_pmevcntsrGH attributes**

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrH	PMU counter H shadow register	RW	0x0000
[31:0]	pmevcntsrG	PMU counter G shadow register	RW	0x0000

## 8.3.8.20 por\_dt\_pmcntrs

Contains the PMU cycle counter shadow register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD990

### Type

RW

### Reset value

See individual bit resets

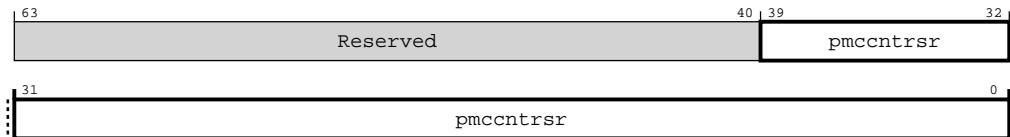
## Usage constraints

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 8-394: por\_dt\_pmccntrs**



**Table 8-399: por\_dt\_pmccntrs attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:0]	pmccntrs	PMU cycle counter shadow register	RW	0x0

### 8.3.8.21 por\_dt\_pmc

Functions as the PMU control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xDA00

##### Type

RW

##### Reset value

See individual bit resets

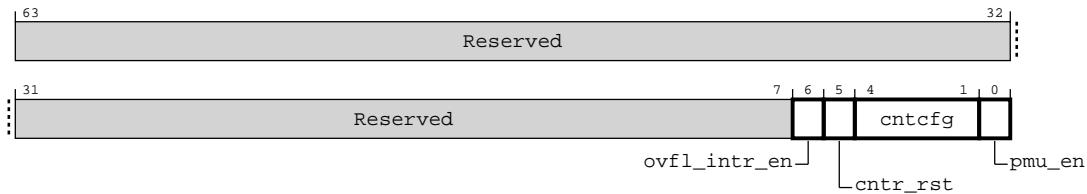
##### Usage constraints

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 8-395: por\_dt\_pmcr**



**Table 8-400: por\_dt\_pmcr attributes**

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	
[6]	ovfl_intr_en	Enables INTREQPMU assertion on PMU counter overflow	RW	0x0
[5]	cntr_rst	Enables clearing of live counters upon assertion of por_dt_pmsrr.ss_req or PMUSNAPSHOTREQ	RW	0x0
[4:1]	cntcfg	Groups adjacent 32-bit registers into a 64-bit register	RW	0x0
[0]	pmu_en	Enables PMU features	RW	0b0

### 8.3.8.22 por\_dt\_pmovsr

Provides the PMU overflow status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xDA18

##### Type

RO

##### Reset value

See individual bit resets

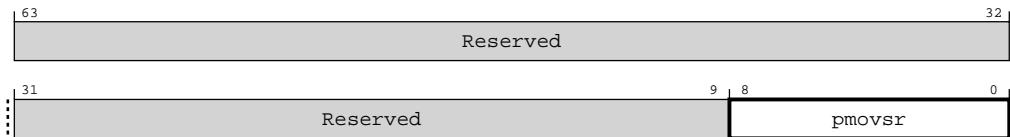
#### Usage constraints

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 8-396: por\_dt\_pmovsr**



**Table 8-401: por\_dt\_pmovsr attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8:0]	pmovsr	PMU overflow status Bit 8:  Indicates overflow from cycle counter  Bits [7:0]: Indicates overflow from counters 7 to 0	RO	0x0

### 8.3.8.23 por\_dt\_pmovsr\_clr

Clears the PMU overflow status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xDA20

##### Type

WO

##### Reset value

See individual bit resets

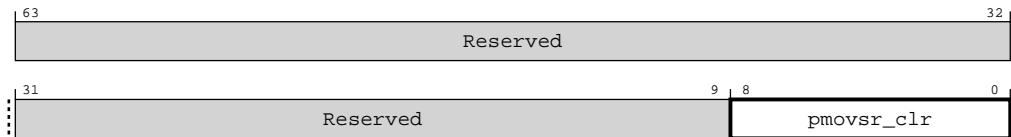
##### Usage constraints

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 8-397: por\_dt\_pmovsr\_clr**



**Table 8-402: por\_dt\_pmovsr\_clr attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8:0]	pmovsr_clr	Write a 1 to clear the corresponding bit in por_dt_pmovsr.pmovsr	WO	0b0

### 8.3.8.24 por\_dt\_pmssr

Provides the PMU snapshot status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xDA28

##### Type

RO

##### Reset value

See individual bit resets

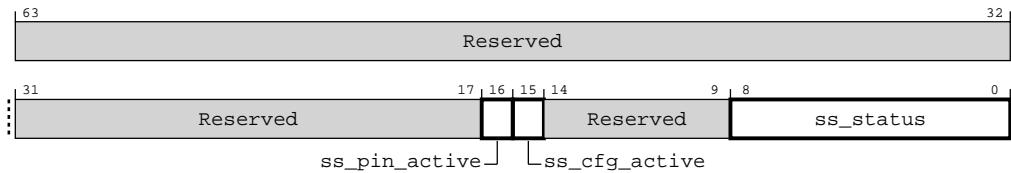
##### Usage constraints

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 8-398: por\_dt\_pmssr**



**Table 8-403: por\_dt\_pmssr attributes**

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	
[16]	ss_pin_active	Activates PMU snapshot from PMUSNAPSHOTREQ	RO	0b0
[15]	ss_cfg_active	PMU snapshot activated from configuration write	RO	0b0
[14:9]	Reserved	Reserved	RO	
[8:0]	ss_status	PMU snapshot status Bit 8:  Indicates snapshot status for cycle counter  Bits [7:0]: Indicates snapshot status for counters 7 to 0	RO	0b0

### 8.3.8.25 por\_dt\_pmsrr

Sends PMU snapshot requests.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xDA30

##### Type

WO

##### Reset value

See individual bit resets

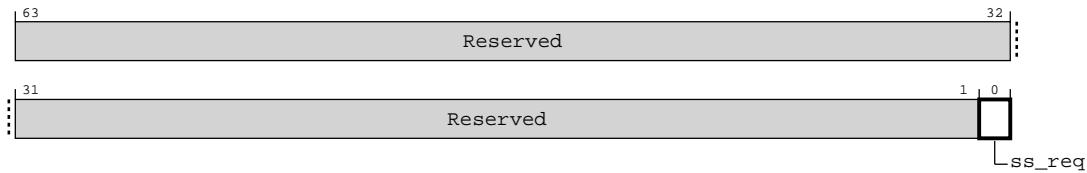
##### Usage constraints

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 8-399: por\_dt\_pmsrr**



**Table 8-404: por\_dt\_pmsrr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	ss_req	Write a 1 to request PMU snapshot	WO	0b0

### 8.3.8.26 por\_dt\_claim

Functions as the claim tag set register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFA0

##### 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 8-400: por\_dt\_claim**



**Table 8-405: por\_dt\_claim attributes**

Bits	Name	Description	Type	Reset
[63:32]	clr	Upper half of the claim tag value; enables individual bits to be cleared (write) and returns the current claim tag value (read)	RW	0b0
[31:0]	set	Lower half of the claim tag value; allows individual bits to be set (write) and returns the number of bits that can be set (read)	RW	0xffffffffffff

### 8.3.8.27 por\_dt\_devaff

Functions as the device affinity register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFA8

##### Type

RO

##### Reset value

See individual bit resets

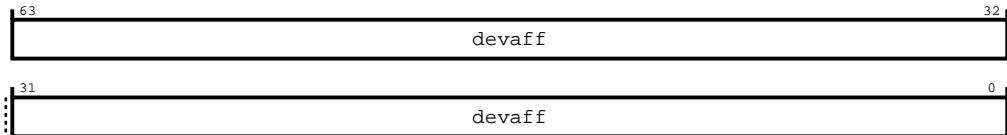
##### Usage constraints

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 8-401: por\_dt\_devaff**



**Table 8-406: por\_dt\_devaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.8.28 por\_dt\_lsr

Functions as the lock status register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFB0

##### Type

RO

##### Reset value

See individual bit resets

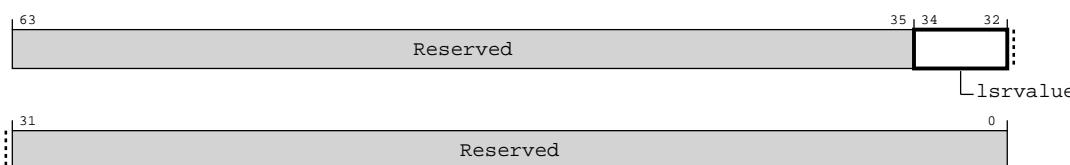
##### Usage constraints

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 8-402: por\_dt\_lsr**



**Table 8-407: por\_dt\_lsr attributes**

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	
[34:32]	lsrvalue	Lock status value	RO	0b0
[31:0]	Reserved	Reserved	RO	

### 8.3.8.29 por\_dt\_authstatus\_devarch

Functions as the authentication status register and the device architecture register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFB8

##### Type

RO

##### Reset value

See individual bit resets

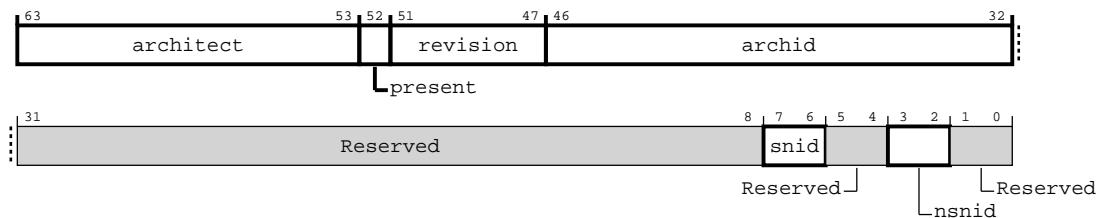
##### Usage constraints

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 8-403: por\_dt\_authstatus\_devarch**



**Table 8-408: por\_dt\_authstatus\_devarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0b0

Bits	Name	Description	Type	Reset
[52]	present	Present	RO	0b1
[51:47]	revision	Architecture revision	RO	0b0
[46:32]	archid	Architecture ID	RO	0b0
[31:8]	Reserved	Reserved	RO	
[7:6]	snid	Secure non-invasive debug	RO	0b10
[5:4]	Reserved	Reserved	RO	
[3:2]	nsnid	Non-secure non-invasive debug	RO	0b10
[1:0]	Reserved	Reserved	RO	

### 8.3.8.30 por\_dt\_devid

Functions as the device configuration register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFC0

##### Type

RO

##### Reset value

See individual bit resets

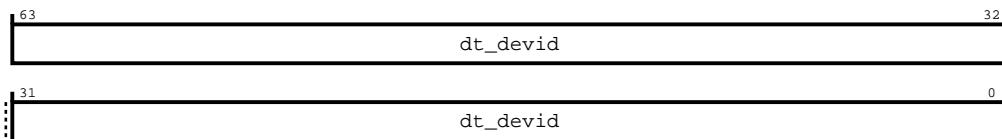
#### Usage constraints

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 8-404: por\_dt\_devid**



**Table 8-409: por\_dt\_devid attributes**

Bits	Name	Description	Type	Reset
[63:0]	dt_devid	Device ID	RO	0b0

### 8.3.8.31 por\_dt\_devtype

Functions as the device type identifier register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFC8

##### Type

RO

##### Reset value

See individual bit resets

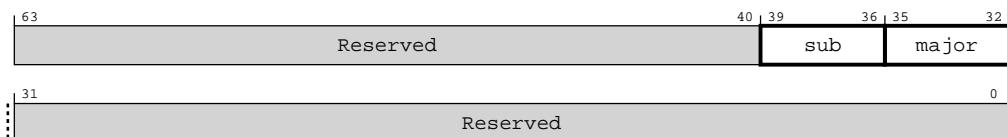
##### Usage constraints

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 8-405: por\_dt\_devtype**



**Table 8-410: por\_dt\_devtype attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	sub	Sub type	RO	0x4
[35:32]	major	Major type	RO	0x3
[31:0]	Reserved	Reserved	RO	

### 8.3.8.32 por\_dt\_pidr45

Functions as the identification register for peripheral ID 4 and peripheral ID 5.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFD0

##### Type

RO

##### Reset value

See individual bit resets

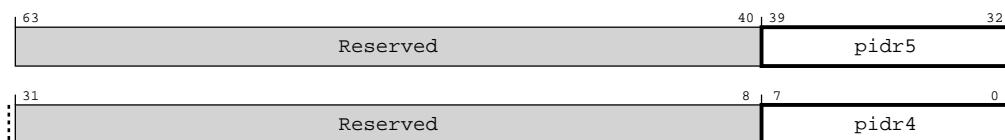
#### Usage constraints

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 8-406: por\_dt\_pidr45**



**Table 8-411: por\_dt\_pidr45 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	pidr5	Peripheral ID 5	RO	0b0
[31:8]	Reserved	Reserved	RO	
[7:0]	pidr4	Peripheral ID 4	RO	0x4

### 8.3.8.33 por\_dt\_pidr67

Functions as the identification register for peripheral ID 6 and peripheral ID 7.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFD8

##### Type

RO

##### Reset value

See individual bit resets

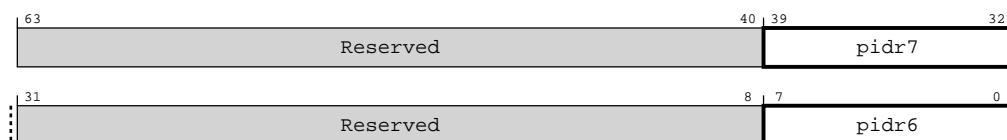
##### Usage constraints

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 8-407: por\_dt\_pidr67**



**Table 8-412: por\_dt\_pidr67 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	pidr7	Peripheral ID 7	RO	0b0
[31:8]	Reserved	Reserved	RO	
[7:0]	pidr6	Peripheral ID 6	RO	0b0

### 8.3.8.34 por\_dt\_pidr01

Functions as the identification register for peripheral ID 0 and peripheral ID 1.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFE0

##### Type

RO

##### Reset value

See individual bit resets

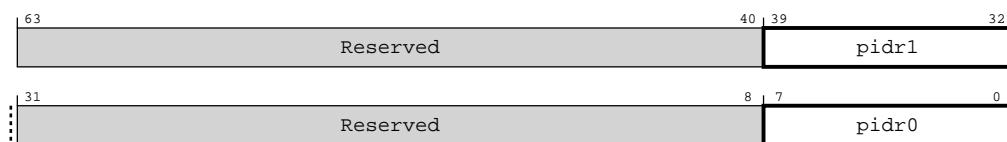
#### Usage constraints

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 8-408: por\_dt\_pidr01**



**Table 8-413: por\_dt\_pidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	pidr1	Peripheral ID 1	RO	0xb4
[31:8]	Reserved	Reserved	RO	
[7:0]	pidr0	Peripheral ID 0	RO	0x34

### 8.3.8.35 por\_dt\_pidr23

Functions as the identification register for peripheral ID 2 and peripheral ID 3.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFE8

##### Type

RO

##### Reset value

See individual bit resets

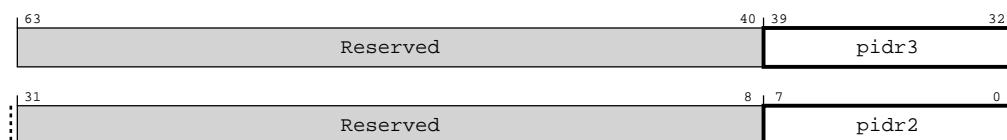
#### Usage constraints

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 8-409: por\_dt\_pidr23**



**Table 8-414: por\_dt\_pidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	pidr3	Peripheral ID 3	RO	0b0
[31:8]	Reserved	Reserved	RO	
[7:0]	pidr2	Peripheral ID 2	RO	0x7

### 8.3.8.36 por\_dt\_cidr01

Functions as the identification register for component ID 0 and component ID 1.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFF0

##### Type

RO

##### Reset value

See individual bit resets

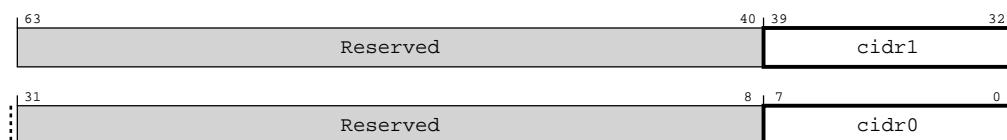
##### Usage constraints

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 8-410: por\_dt\_cidr01**



**Table 8-415: por\_dt\_cidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	cidr1	Component ID 1	RO	0x9f
[31:8]	Reserved	Reserved	RO	
[7:0]	cidr0	Component ID 0	RO	0xd

### 8.3.8.37 por\_dt\_cidr23

Functions as the identification register for component ID 2 and component ID 3.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFF8

##### Type

RO

##### Reset value

See individual bit resets

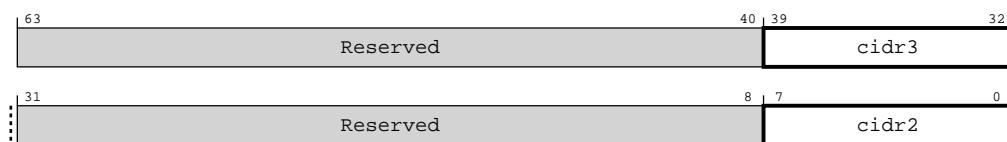
##### Usage constraints

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 8-411: por\_dt\_cidr23**



**Table 8-416: por\_dt\_cidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	cidr3	Component ID 3	RO	0xb1
[31:8]	Reserved	Reserved	RO	
[7:0]	cidr2	Component ID 2	RO	0x5

### 8.3.9 HN-S register summary

The following table describes the registers for the relevant component.

**Table 8-417: cmn\_hns\_cfg register summary**

Offset	Name	Type	Description
0x0	cmn_hns_node_info	RO	cmn_hns_node_info
0x80	cmn_hns_child_info	RO	cmn_hns_child_info
0x980	cmn_hns_scr	RW	cmn_hns_scr
0x988	cmn_hns_rcr	RW	cmn_hns_rcr
0x900	cmn_hns_unit_info	RO	cmn_hns_unit_info
0x908	cmn_hns_unit_info_1	RO	cmn_hns_unit_info_1
0xA00	cmn_hns_cfg_ctl	RW	cmn_hns_cfg_ctl
0xA08	cmn_hns_aux_ctl	RW	cmn_hns_aux_ctl
0xA10	cmn_hns_aux_ctl_1	RW	cmn_hns_aux_ctl_1
0xA18	cmn_hns_cbusy_limit_ctl	RW	cmn_hns_cbusy_limit_ctl
0xA20	cmn_hns_txrsp_arb_weight_ctl	RW	cmn_hns_txrsp_arb_weight_ctl
0xA28	cmn_hns_cbusy_mode_ctl	RW	cmn_hns_cbusy_mode_ctl
0xA30	cmn_hns_lbt_cfg_ctl	RW	cmn_hns_lbt_cfg_ctl
0xA38	cmn_hns_lbt_aux_ctl	RW	cmn_hns_lbt_aux_ctl
0xA50	cmn_hns_datasource_ctl	RW	cmn_hns_datasource_ctl
0x1900	cmn_hns_ppu_pwpr	RW	cmn_hns_ppu_pwpr
0x1908	cmn_hns_ppu_pwsr	RO	cmn_hns_ppu_pwsr
0x1914	cmn_hns_ppu_misr	RO	cmn_hns_ppu_misr
0x28B0	cmn_hns_ppu_idr0	RO	cmn_hns_ppu_idr0
0x28B4	cmn_hns_ppu_idr1	RO	cmn_hns_ppu_idr1
0x28C8	cmn_hns_ppu_iidr	RO	cmn_hns_ppu_iidr
0x28CC	cmn_hns_ppu_aidr	RO	cmn_hns_ppu_aidr
0x1A00	cmn_hns_ppu_dyn_ret_threshold	RW	cmn_hns_ppu_dyn_ret_threshold
0xA80	cmn_hns_qos_band	RO	cmn_hns_qos_band
0xE000	cmn_hns_errfr	RO	cmn_hns_errfr
0xE008	cmn_hns_errctrlr	RW	cmn_hns_errctrlr
0xE010	cmn_hns_errstatus	W1C	cmn_hns_errstatus
0xE018	cmn_hns_erraddr	RW	cmn_hns_erraddr
0xE020	cmn_hns_errmisc0	RW	cmn_hns_errmisc0
0xE028	cmn_hns_errmisc1	RW	cmn_hns_errmisc1
0xE800	cmn_hns_errpfgf	RO	cmn_hns_errpfgf
0xE808	cmn_hns_errpfctl	RW	cmn_hns_errpfctl
0xE810	cmn_hns_errpfcdn	RW	cmn_hns_errpfcdn
0xE040	cmn_hns_errfr_NS	RO	cmn_hns_errfr_NS
0xE048	cmn_hns_errctrlr_NS	RW	cmn_hns_errctrlr_NS
0xE050	cmn_hns_errstatus_NS	W1C	cmn_hns_errstatus_NS

Offset	Name	Type	Description
0xE058	cmn_hns_erraddr_NS	RW	<a href="#">cmn_hns_erraddr_NS</a>
0xE060	cmn_hns_errmisc0_NS	RW	<a href="#">cmn_hns_errmisc0_NS</a>
0xE068	cmn_hns_errmisc1_NS	RW	<a href="#">cmn_hns_errmisc1_NS</a>
0xE840	cmn_hns_errpfgf_NS	RO	<a href="#">cmn_hns_errpfgf_NS</a>
0xE848	cmn_hns_errpfgctl_NS	RW	<a href="#">cmn_hns_errpfgctl_NS</a>
0xE850	cmn_hns_errpfgcdn_NS	RW	<a href="#">cmn_hns_errpfgcdn_NS</a>
0xED00	cmn_hns_errcapctl	RW	<a href="#">cmn_hns_errcapctl</a>
0xEE00	cmn_hns_errgsr	RO	<a href="#">cmn_hns_errgsr</a>
0xEE10	cmn_hns_errildr	RO	<a href="#">cmn_hns_errildr</a>
0xEFA8	cmn_hns_errdevaff	RO	<a href="#">cmn_hns_errdevaff</a>
0xEF88	cmn_hns_errdevarch	RO	<a href="#">cmn_hns_errdevarch</a>
0xEFC8	cmn_hns_errdevid	RO	<a href="#">cmn_hns_errdevid</a>
0xEF00	cmn_hns_errpidr45	RO	<a href="#">cmn_hns_errpidr45</a>
0xEF00	cmn_hns_errpidr01	RO	<a href="#">cmn_hns_errpidr01</a>
0xEF08	cmn_hns_errpidr23	RO	<a href="#">cmn_hns_errpidr23</a>
0xEFF0	cmn_hns_errcidr01	RO	<a href="#">cmn_hns_errcidr01</a>
0xEFF8	cmn_hns_errcidr23	RO	<a href="#">cmn_hns_errcidr23</a>
0xE030	cmn_hns_err_inj	RW	<a href="#">cmn_hns_err_inj</a>
0xE938	cmn_hns_byte_par_err_inj	WO	<a href="#">cmn_hns_byte_par_err_inj</a>
0xC00	cmn_hns_slc_lock_ways	RW	<a href="#">cmn_hns_slc_lock_ways</a>
0xC08	cmn_hns_slc_lock_base0	RW	<a href="#">cmn_hns_slc_lock_base0</a>
0xC10	cmn_hns_slc_lock_base1	RW	<a href="#">cmn_hns_slc_lock_base1</a>
0xC18	cmn_hns_slc_lock_base2	RW	<a href="#">cmn_hns_slc_lock_base2</a>
0xC20	cmn_hns_slc_lock_base3	RW	<a href="#">cmn_hns_slc_lock_base3</a>
0xC28	cmn_hns_rni_region_vec	RW	<a href="#">cmn_hns_rni_region_vec</a>
0xC30	cmn_hns_rnd_region_vec	RW	<a href="#">cmn_hns_rnd_region_vec</a>
0xC38	cmn_hns_rnf_region_vec	RW	<a href="#">cmn_hns_rnf_region_vec</a>
0xC40	cmn_hns_rnf_region_vec1	RW	<a href="#">cmn_hns_rnf_region_vec1</a>
0xC48	cmn_hns_slcway_partition0_rnf_vec	RW	<a href="#">cmn_hns_slcway_partition0_rnf_vec</a>
0xC50	cmn_hns_slcway_partition1_rnf_vec	RW	<a href="#">cmn_hns_slcway_partition1_rnf_vec</a>
0xC58	cmn_hns_slcway_partition2_rnf_vec	RW	<a href="#">cmn_hns_slcway_partition2_rnf_vec</a>
0xC60	cmn_hns_slcway_partition3_rnf_vec	RW	<a href="#">cmn_hns_slcway_partition3_rnf_vec</a>
0xCB0	cmn_hns_slcway_partition0_rnf_vec1	RW	<a href="#">cmn_hns_slcway_partition0_rnf_vec1</a>
0xCB8	cmn_hns_slcway_partition1_rnf_vec1	RW	<a href="#">cmn_hns_slcway_partition1_rnf_vec1</a>
0xCC0	cmn_hns_slcway_partition2_rnf_vec1	RW	<a href="#">cmn_hns_slcway_partition2_rnf_vec1</a>
0xCC8	cmn_hns_slcway_partition3_rnf_vec1	RW	<a href="#">cmn_hns_slcway_partition3_rnf_vec1</a>
0xC68	cmn_hns_slcway_partition0_rni_vec	RW	<a href="#">cmn_hns_slcway_partition0_rni_vec</a>
0xC70	cmn_hns_slcway_partition1_rni_vec	RW	<a href="#">cmn_hns_slcway_partition1_rni_vec</a>
0xC78	cmn_hns_slcway_partition2_rni_vec	RW	<a href="#">cmn_hns_slcway_partition2_rni_vec</a>
0xC80	cmn_hns_slcway_partition3_rni_vec	RW	<a href="#">cmn_hns_slcway_partition3_rni_vec</a>

Offset	Name	Type	Description
0xC88	cmn_hns_slcway_partition0_rnd_vec	RW	cmn_hns_slcway_partition0_rnd_vec
0xC90	cmn_hns_slcway_partition1_rnd_vec	RW	cmn_hns_slcway_partition1_rnd_vec
0xC98	cmn_hns_slcway_partition2_rnd_vec	RW	cmn_hns_slcway_partition2_rnd_vec
0xCA0	cmn_hns_slcway_partition3_rnd_vec	RW	cmn_hns_slcway_partition3_rnd_vec
0xCA8	cmn_hns_rn_region_lock	RW	cmn_hns_rn_region_lock
0xCD0	cmn_hns_sf_cxg_blocked_ways	RW	cmn_hns_sf_cxg_blocked_ways
0xCE0	cmn_hns_cxg_ha_metadata_exclusion_list	RW	cmn_hns_cxg_ha_metadata_exclusion_list
0xD60	cmn_hns_cxg_ha_snp_throttle_disable_list	RW	cmn_hns_cxg_ha_snp_throttle_disable_list
0xCD8	cmn_hns_cxg_ha_smp_exclusion_list	RW	cmn_hns_cxg_ha_smp_exclusion_list
0xCF0	hn_sam_hash_addr_mask_reg	RW	hn_sam_hash_addr_mask_reg
0xCF8	hn_sam_region_cmp_addr_mask_reg	RW	hn_sam_region_cmp_addr_mask_reg
0xD48	cmn_hns_sam_cfg1_def_hashed_region	RW	cmn_hns_sam_cfg1_def_hashed_region
0xD50	cmn_hns_sam_cfg2_def_hashed_region	RW	cmn_hns_sam_cfg2_def_hashed_region
0xD00	cmn_hns_sam_control	RW	cmn_hns_sam_control
0xD28	cmn_hns_sam_control2	RW	cmn_hns_sam_control2
0xD08	cmn_hns_sam_memregion0	RW	cmn_hns_sam_memregion0
0xD38	cmn_hns_sam_memregion0_end_addr	RW	cmn_hns_sam_memregion0_end_addr
0xD10	cmn_hns_sam_memregion1	RW	cmn_hns_sam_memregion1
0xD40	cmn_hns_sam_memregion1_end_addr	RW	cmn_hns_sam_memregion1_end_addr
0xD18	cmn_hns_sam_sn_properties	RW	cmn_hns_sam_sn_properties
0xD20	cmn_hns_sam_6sn_nodeid	RW	cmn_hns_sam_6sn_nodeid
0xCE8	cmn_hns_sam_sn_properties1	RW	cmn_hns_sam_sn_properties1
0xD30	cmn_hns_sam_sn_properties2	RW	cmn_hns_sam_sn_properties2
{0-4} 0xF80 : 0xFA0	cmn_hns_cml_port_aggr_grp0-4_add_mask	RW	cmn_hns_cml_port_aggr_grp0-4_add_mask
{5-31} 0x6028 : 0x60F8			
{0-4} 0xF80 : 0xFA0	cmn_hns_cml_port_aggr_grp5-31_add_mask	RW	cmn_hns_cml_port_aggr_grp5-31_add_mask
{5-31} 0x6028 : 0x60F8			
{0-7} 0x6400 : 0x6438	cmn_hns_cml_cpag_base_idx_grp0-3	RW	cmn_hns_cml_cpag_base_idx_grp0-3
{0-1} 0xFB0 : 0xFB8	cmn_hns_cml_port_aggr_grp_reg0-12	RW	cmn_hns_cml_port_aggr_grp_reg0-12
{2-12} 0x6110 : 0x6160			
0xFD0	cmn_hns_cml_port_aggr_ctrl_reg	RW	cmn_hns_cml_port_aggr_ctrl_reg
{1-6} 0x6208 : 0x6230	cmn_hns_cml_port_aggr_ctrl_reg1-6	RW	cmn_hns_cml_port_aggr_ctrl_reg1-6
0xF50	cmn_hns_abf_lo_addr	RW	cmn_hns_abf_lo_addr
0xF58	cmn_hns_abf_hi_addr	RW	cmn_hns_abf_hi_addr
0xF60	cmn_hns_abf_pr	RW	cmn_hns_abf_pr
0xF68	cmn_hns_abf_sr	RO	cmn_hns_abf_sr
0x1000	cmn_hns_cbusy_write_limit_ctl	RW	cmn_hns_cbusy_write_limit_ctl
0x1008	cmn_hns_cbusy_resp_ctl	RW	cmn_hns_cbusy_resp_ctl
0x1010	cmn_hns_cbusy_sn_ctl	RW	cmn_hns_cbusy_sn_ctl

Offset	Name	Type	Description
0x1068	cmn_hns_cbusy_sn1_ctl	RW	<a href="#">cmn_hns_cbusy_sn1_ctl</a>
0x1070	cmn_hns_cbusy_sn1_threshold	RW	<a href="#">cmn_hns_cbusy_sn1_threshold</a>
0x1018	cmn_hns_lbt_cbusy_ctl	RW	<a href="#">cmn_hns_lbt_cbusy_ctl</a>
0x1080	cmn_hns_cbusy_ccg_ctl	RW	<a href="#">cmn_hns_cbusy_ccg_ctl</a>
0x1088	cmn_hns_cbusy_ccg_threshold	RW	<a href="#">cmn_hns_cbusy_ccg_threshold</a>
0x1020	cmn_hns_pocq_alloc_class_dedicated	RW	<a href="#">cmn_hns_pocq_alloc_class_dedicated</a>
0x1028	cmn_hns_pocq_alloc_class_max_allowed	RW	<a href="#">cmn_hns_pocq_alloc_class_max_allowed</a>
0x1030	cmn_hns_pocq_alloc_class_contented_min	RW	<a href="#">cmn_hns_pocq_alloc_class_contented_min</a>
0x1038	cmn_hns_pocq_alloc_misc_max_allowed	RW	<a href="#">cmn_hns_pocq_alloc_misc_max_allowed</a>
0x1040	cmn_hns_class_ctl	RW	<a href="#">cmn_hns_class_ctl</a>
0x1048	cmn_hns_pocq_qos_class_ctl	RW	<a href="#">cmn_hns_pocq_qos_class_ctl</a>
0x1050	cmn_hns_class_pocq_arb_weight_ctl	RW	<a href="#">cmn_hns_class_pocq_arb_weight_ctl</a>
0x1058	cmn_hns_class_retry_weight_ctl	RW	<a href="#">cmn_hns_class_retry_weight_ctl</a>
0x1060	cmn_hns_pocq_misc_retry_weight_ctl	RW	<a href="#">cmn_hns_pocq_misc_retry_weight_ctl</a>
0xFE0	cmn_hns_partner_scratch_reg0	RW	<a href="#">cmn_hns_partner_scratch_reg0</a>
0xFE8	cmn_hns_partner_scratch_reg1	RW	<a href="#">cmn_hns_partner_scratch_reg1</a>
0xB80	cmn_hns_cfg_slcsf_dbgrd	WO	<a href="#">cmn_hns_cfg_slcsf_dbgrd</a>
0xB88	cmn_hns_slc_cache_access_slc_tag	RO	<a href="#">cmn_hns_slc_cache_access_slc_tag</a>
0xB90	cmn_hns_slc_cache_access_slc_tag1	RO	<a href="#">cmn_hns_slc_cache_access_slc_tag1</a>
0xB98	cmn_hns_slc_cache_access_slc_data	RO	<a href="#">cmn_hns_slc_cache_access_slc_data</a>
0xBC0	cmn_hns_slc_cache_access_slc_mte_tag	RO	<a href="#">cmn_hns_slc_cache_access_slc_mte_tag</a>
0xBA0	cmn_hns_slc_cache_access_sf_tag	RO	<a href="#">cmn_hns_slc_cache_access_sf_tag</a>
0xBA8	cmn_hns_slc_cache_access_sf_tag1	RO	<a href="#">cmn_hns_slc_cache_access_sf_tag1</a>
0xBB0	cmn_hns_slc_cache_access_sf_tag2	RO	<a href="#">cmn_hns_slc_cache_access_sf_tag2</a>
0xD900	cmn_hns_pmu_event_sel	RW	<a href="#">cmn_hns_pmu_event_sel</a>
0xD908	cmn_hns_pmu_mpam_sel	RW	<a href="#">cmn_hns_pmu_mpam_sel</a>
0xC900 : 0xC920	cmn_hns_amevcntr0-4	RW	<a href="#">cmn_hns_amevcntr0-4</a>
0xCD00 : 0xCD10	cmn_hns_amevtyper0-4	RO	<a href="#">cmn_hns_amevtyper0-4</a>
0xD500	cmn_hns_amcntensem	W1S	<a href="#">cmn_hns_amcntensem</a>
0xD520	cmn_hns_amcntenclr	W1C	<a href="#">cmn_hns_amcntenclr</a>
0xD700	cmn_hns_amcfg	RO	<a href="#">cmn_hns_amcfg</a>
0xD704	cmn_hns_amcr	RW	<a href="#">cmn_hns_amcr</a>
0xD708	cmn_hns_amiidr	RO	<a href="#">cmn_hns_amiidr</a>
0xD8BC	cmn_hns_amdevarch	RO	<a href="#">cmn_hns_amdevarch</a>
0xD8C8	cmn_hns_amdevid	RO	<a href="#">cmn_hns_amdevid</a>
0xD8CC	cmn_hns_amdevtype	RO	<a href="#">cmn_hns_amdevtype</a>
0xD8D0	cmn_hns_ampidr4	RO	<a href="#">cmn_hns_ampidr4</a>
0xD8D4 : 0xD8DC	cmn_hns_ampidr5-7	RO	<a href="#">cmn_hns_ampidr5-7</a>
0xD8E0	cmn_hns_ampidr0	RO	<a href="#">cmn_hns_ampidr0</a>
0xD8E4	cmn_hns_ampidr1	RO	<a href="#">cmn_hns_ampidr1</a>

Offset	Name	Type	Description
0xD8E8	cmn_hns_ampidr2	RO	cmn_hns_ampidr2
0xD8EC	cmn_hns_ampidr3	RO	cmn_hns_ampidr3
0xD8F0	cmn_hns_amcidr0	RO	cmn_hns_amcidr0
0xD8F4	cmn_hns_amcidr1	RO	cmn_hns_amcidr1
0xD8F8	cmn_hns_amcidr2	RO	cmn_hns_amcidr2
0xD8FC	cmn_hns_amcidr3	RO	cmn_hns_amcidr3
0xD910 + #8*index}	cmn_hns_pmu_mpam_pardid_mask0-7	RW	cmn_hns_pmu_mpam_pardid_mask0-7
0x3C00 : 0x43E0	cmn_hns_rn_cluster0-63_physid_reg0	RW	cmn_hns_rn_cluster0-63_physid_reg0
0x4400 : 0x4BE0	cmn_hns_rn_cluster64-127_physid_reg0	RW	cmn_hns_rn_cluster64-127_physid_reg0
0x3C08 : 0x4BE8	cmn_hns_rn_cluster0-127_physid_reg1	RW	cmn_hns_rn_cluster0-127_physid_reg1
0x3C10 : 0x4BF0	cmn_hns_rn_cluster0-127_physid_reg2	RW	cmn_hns_rn_cluster0-127_physid_reg2
0x3C18 : 0x4BF8	cmn_hns_rn_cluster0-127_physid_reg3	RW	cmn_hns_rn_cluster0-127_physid_reg3
0x5010 : 0x51F8	cmn_hns_sam_nonhash_cfg1_memregion2-63	RW	cmn_hns_sam_nonhash_cfg1_memregion2-63
0x5210 : 0x53F8	cmn_hns_sam_nonhash_cfg2_memregion2-63	RW	cmn_hns_sam_nonhash_cfg2_memregion2-63
0x5400 : 0x5478	cmn_hns_sam_htg_cfg1_memregion0-15	RW	cmn_hns_sam_htg_cfg1_memregion0-15
0x5480 : 0x54F8	cmn_hns_sam_htg_cfg2_memregion0-15	RW	cmn_hns_sam_htg_cfg2_memregion0-15
0x5500 : 0x5578	cmn_hns_sam_htg_cfg3_memregion0-15	RW	cmn_hns_sam_htg_cfg3_memregion0-15
{0-15} 0x5600 : 0x5678	cmn_hns_sam_htg_sn_nodeid_reg0-31	RW	cmn_hns_sam_htg_sn_nodeid_reg0-31
{16-31} 0x5808 : 0x5880			
0x5680 : 0x56F8	cmn_hns_sam_htg_sn_attr0-15	RW	cmn_hns_sam_htg_sn_attr0-15
0x5700 : 0x5718	cmn_hns_sam_ccg_sa_nodeid_reg0-3	RW	cmn_hns_sam_ccg_sa_nodeid_reg0-3
0x5740 : 0x5758	cmn_hns_sam_ccg_sa_attr0-3	RW	cmn_hns_sam_ccg_sa_attr0-3
0x5780 : 0x57B8	hns_generic_regs0-7	RW	hns_generic_regs0-7
0x5900	cmn_hns_pa2setaddr_slc	RW	cmn_hns_pa2setaddr_slc
0x5908	cmn_hns_pa2setaddr_sf	RW	cmn_hns_pa2setaddr_sf
0x5910	cmn_hns_pa2setaddr_flex_slc	RW	cmn_hns_pa2setaddr_flex_slc
0x5918	cmn_hns_pa2setaddr_flex_sf	RW	cmn_hns_pa2setaddr_flex_sf
{0-31} 0x7000 : 0x70F8	lcn_hashed_tgt_grp_cfg1_region0-31	RW	lcn_hashed_tgt_grp_cfg1_region0-31
{0-31} 0x7100 : 0x71F8	lcn_hashed_tgt_grp_cfg2_region0-31	RW	lcn_hashed_tgt_grp_cfg2_region0-31
{0-31} 0x7200 : 0x72F8	lcn_hashed_target_grp_secondary_cfg1_reg0-31	RW	lcn_hashed_target_grp_secondary_cfg1_reg0-31
{0-31} 0x7300 : 0x73F8	lcn_hashed_target_grp_secondary_cfg2_reg0-31	RW	lcn_hashed_target_grp_secondary_cfg2_reg0-31
{0-31} 0x7400 : 0x74F8	lcn_hashed_target_grp_hash_cntl_reg0-31	RW	lcn_hashed_target_grp_hash_cntl_reg0-31
{0-3} 0x7500 : 0x7518	lcn_hashed_target_group_hn_count_reg0-3	RW	lcn_hashed_target_group_hn_count_reg0-3
{0-7} 0x7520 : 0x7558	lcn_hashed_target_grp_cal_mode_reg0-7	RW	lcn_hashed_target_grp_cal_mode_reg0-7
{0-1} 0x7560 : 0x7568	lcn_hashed_target_grp_hnf_cpa_en_reg0-1	RW	lcn_hashed_target_grp_hnf_cpa_en_reg0-1
{0-15} 0x7580 : 0x75F8	lcn_hashed_target_grp_cpag_perhnf_reg0-15	RW	lcn_hashed_target_grp_cpag_perhnf_reg0-15
{0-31} 0x7700 : 0x77F8	lcn_hashed_target_grp_compact_cpag_ctrl0-31	RW	lcn_hashed_target_grp_compact_cpag_ctrl0-31
{0-31} 0x7800 : 0x78F8	lcn_hashed_target_grp_compact_hash_ctrl0-31	RW	lcn_hashed_target_grp_compact_hash_ctrl0-31

### 8.3.9.1 cmn\_hns\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

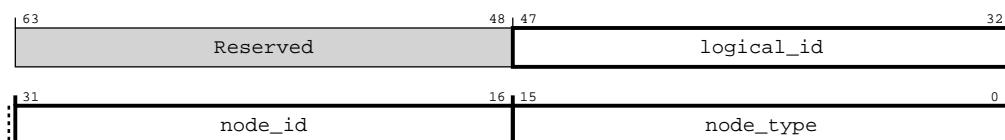
##### Usage constraints

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 8-412: cmn\_hns\_node\_info**



**Table 8-418: cmn\_hns\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x00

### 8.3.9.2 cmn\_hns\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

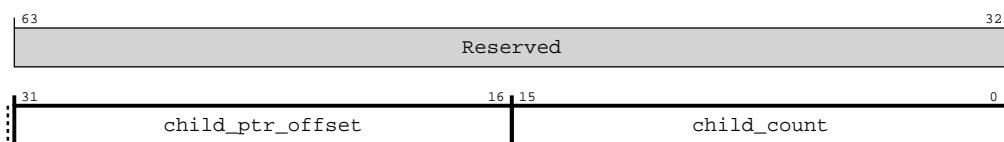
#### Usage constraints

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 8-413: cmn\_hns\_child\_info**



**Table 8-419: 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	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.9.3 cmn\_hns\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x980

### Type

RW

### Reset value

See individual bit resets

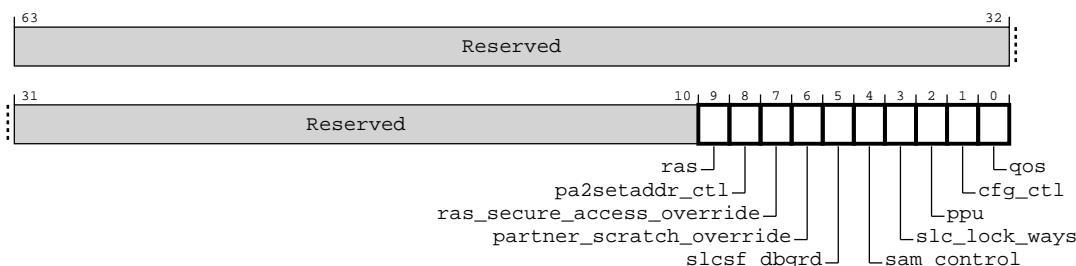
### Usage constraints

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 8-414: cmn\_hns\_scr**



**Table 8-420: cmn\_hns\_scr attributes**

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	
[9]	ras	Allow Secure override of the RAS registers	RW	0b0
[8]	pa2setaddr_ctl	Allow Secure override of the PA2SETADDR registers	RW	0b0
[7]	ras_secure_access_override	Allow Secure override of the RAS ERMISC<n> registers	RW	0b0
[6]	partner_scratch_override	Allows Secure override of the Partenr scratch registers	RW	0b0
[5]	slcsf_dbgrd	Allows Secure override of the SLC/SF debug read registers	RW	0b0
[4]	sam_control	Allows Secure override of the HN-F SAM control registers	RW	0b0
[3]	slc_lock_ways	Allows Secure override of the cache way locking registers	RW	0b0
[2]	ppu	Allows Secure override of the power policy registers	RW	0b0
[1]	cfg_ctl	Allows Secure override of the configuration control register (cmn_hns_cfg_ctl)	RW	0b0
[0]	qos	Allows Secure override of the QoS registers	RW	0b0

### 8.3.9.4 cmn\_hns\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

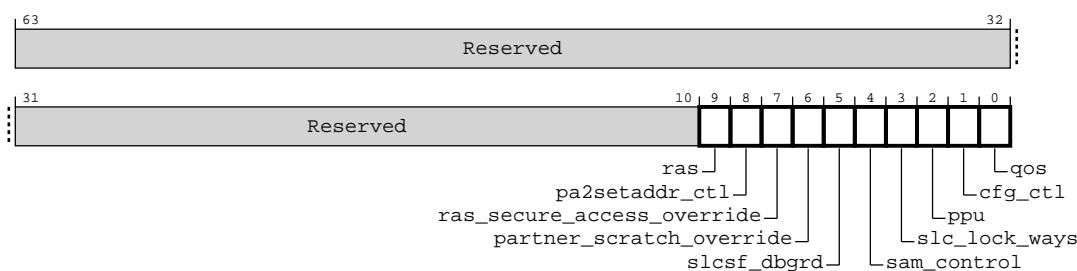
#### Usage constraints

This register is owned in the Root space and is 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 8-415: cmn\_hns\_rcr**



**Table 8-421: cmn\_hns\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	
[9]	ras	Allow Root override of the RAS registers	RW	0b0
[8]	pa2setaddr_ctl	Allows Root override of the PA2SETADDR registers	RW	0b0
[7]	ras_secure_access_override	Allows Root override of the RAS ERMISC<n> registers	RW	0b0
[6]	partner_scratch_override	Allows Root override of the Partner scratch registers	RW	0b0
[5]	slcsf_dbgrd	Allows Root override of the SLC/SF debug read registers	RW	0b0

Bits	Name	Description	Type	Reset
[4]	sam_control	Allows Root override of the HN-F SAM control registers	RW	0b0
[3]	slc_lock_ways	Allows Root override of the cache way locking registers	RW	0b0
[2]	ppu	Allows Root override of the power policy registers	RW	0b0
[1]	cfg_ctl	Allows Root override of the configuration control register (cmn_hns_cfg_ctl)	RW	0b0
[0]	qos	Allows Root override of the QoS registers	RW	0b0

### 8.3.9.5 cmn\_hns\_unit\_info

Provides component identification information for HN-F.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

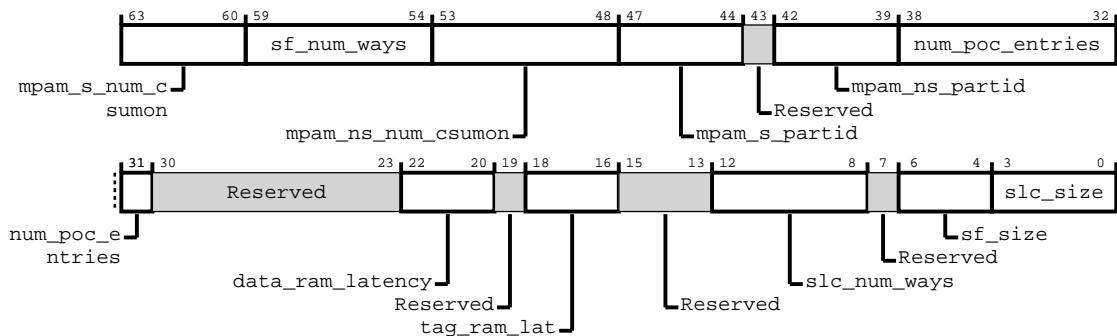
##### Usage constraints

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 8-416: cmn\_hns\_unit\_info**



**Table 8-422: 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	0x2
[59:54]	sf_num_ways	Number of cache ways in the SF	RO	Configuration dependent
[53:48]	mpam_ns_num_csumon	Number of Non-Secure Cache Storage Usage Monitors for MPAM	RO	0x8
[47:44]	mpam_s_partid	MPAM Secure supported PARTIDs  <b>0b0000</b> 1 S PARTID  <b>0b0001</b> Reserved  <b>0b0010</b> Reserved  <b>0b0011</b> 8 S PARTID  <b>0b0100</b> 16 S PARTID	RO	Configuration dependent
[43]	Reserved	Reserved	RO	-
[42:39]	mpam_ns_partid	MPAM Non-Secure supported PARTIDs  <b>0b0000</b> 1 NS PARTID  <b>0b0001</b> Reserved  <b>0b0010</b> Reserved  <b>0b0011</b> Reserved  <b>0b0100</b> Reserved  <b>0b0101</b> 32 NS PARTID  <b>0b0110</b> 64 NS PARTID  <b>0b0111</b> 128 NS PARTID  <b>0b1000</b> 256 NS PARTID  <b>0b1001</b> 512 NS PARTID	RO	Configuration dependent
[38:31]	num_poc_entries	Number of POCQ entries	RO	0x0
[30:23]	Reserved	Reserved	RO	-
[22:20]	data_ram_latency	SLC data RAM latency (in cycles)	RO	Configuration dependent
[19]	Reserved	Reserved	RO	-
[18:16]	tag_ram_lat	SLC tag RAM latency (in cycles)	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:13]	Reserved	Reserved	RO	-
[12:8]	slc_num_ways	Number of cache ways in the SLC	RO	Configuration dependent
[7]	Reserved	Reserved	RO	-
[6:4]	sf_size	SF size  <b>0b000</b> ( 32KB * sf_num_ways) <b>0b001</b> ( 64KB * sf_num_ways) <b>0b010</b> ( 128KB * sf_num_ways) <b>0b011</b> ( 256KB * sf_num_ways) <b>0b100</b> ( 512KB * sf_num_ways) <b>0b101</b> (1024KB * sf_num_ways) <b>0b111</b> (0KB)	RO	Configuration dependent
[3:0]	slc_size	SLC size  <b>0b0000</b> No SLC <b>0b0001</b> 128KB <b>0b0010</b> 256KB <b>0b0011</b> 512KB <b>0b0100</b> 1MB <b>0b0101</b> 1.5MB <b>0b0110</b> 2MB <b>0b0111</b> 3MB <b>0b1000</b> 4MB <b>0b1001</b> 384KB	RO	Configuration dependent

### 8.3.9.6 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

0x908

##### Type

RO

##### Reset value

See individual bit resets

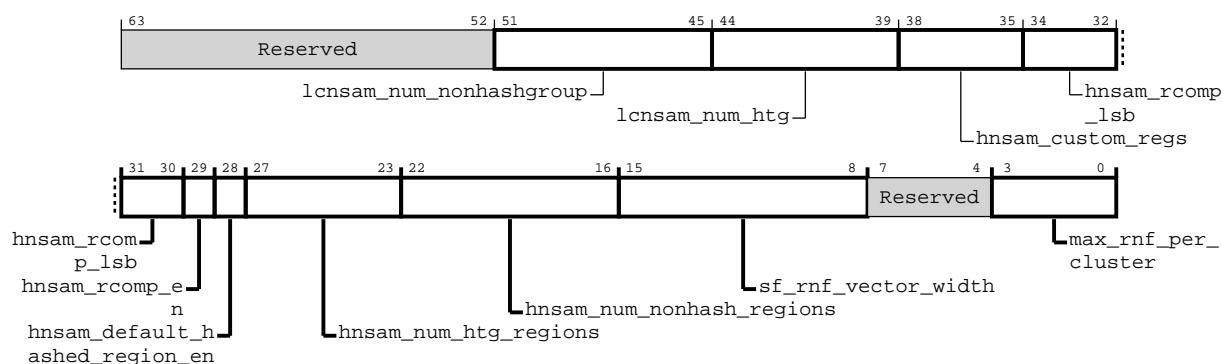
##### Usage constraints

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 8-417: cmn\_hns\_unit\_info\_1**



**Table 8-423: cmn\_hns\_unit\_info\_1 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:45]	lcnsam_num_nonhashgroup	Number of non hashed regions supported by LCN SAM	RO	0x08
[44:39]	lcnsam_num_htg	Number of hashed regions supported by LCN SAM	RO	0x04
[38:35]	hnsam_custom_regs	Number of customer specific registers for customer implemented logic	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[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	0x1
[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 RN-F tracking vector in the Snoop Filter (Total SF_VEC_WIDTH = (TOTAL_RN-F/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	0x1

### 8.3.9.7 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

0xA00

##### 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

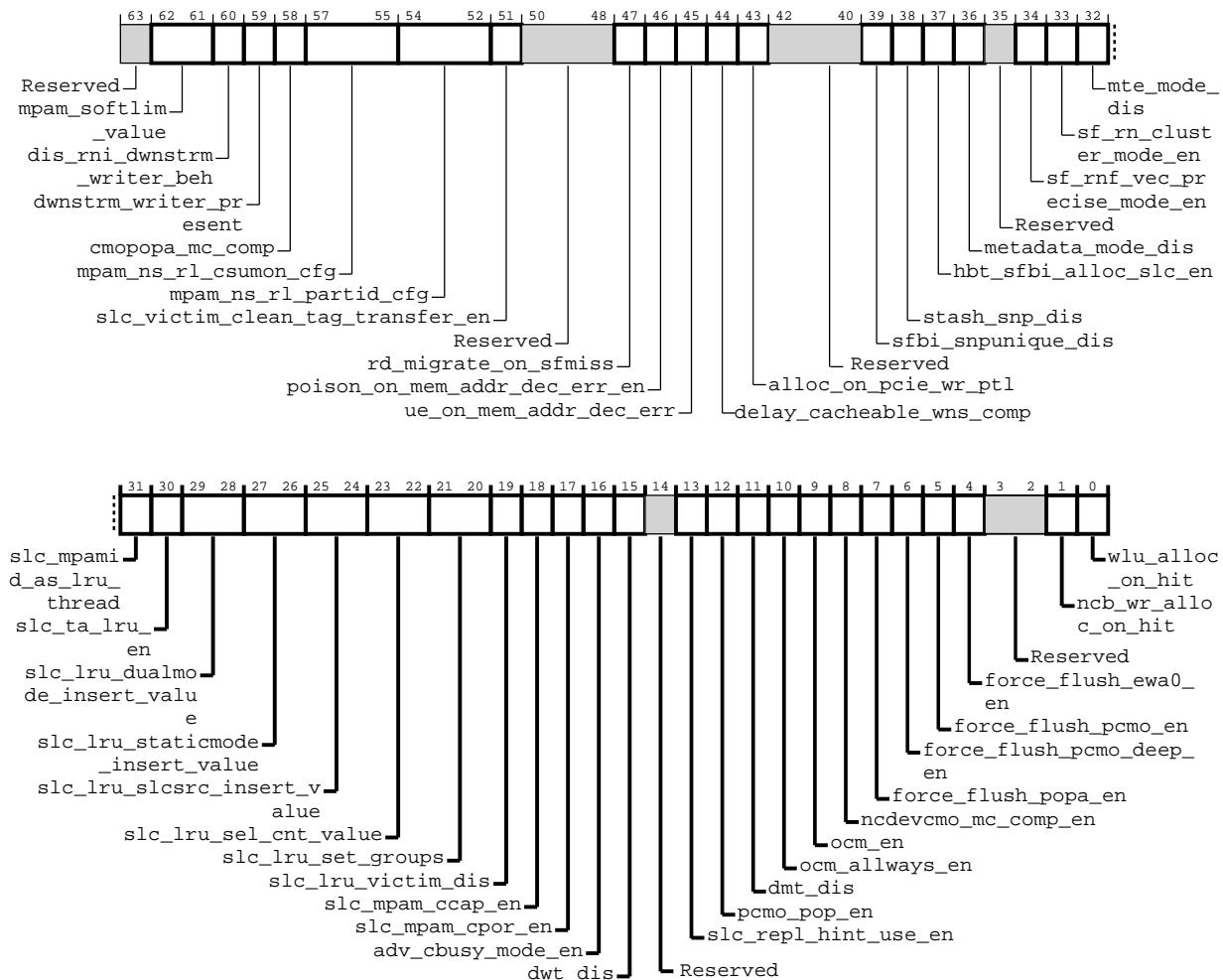
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.cfg\_ctl bit are

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 8-418: cmn\_hns\_cfg\_ctl**



**Table 8-424: cmn\_hns\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63]	mpam_partid_oor_use_default_partid_en	<p>When set, use Default MPAM PartID for CMAX &amp; CPBM if Request PartID is Out Of Range.</p> <p><b>Note</b> Override is only for internal use to determine CMAX &amp; CPBM. Original PartID is stored and propagated to SN.</p> <p><b>Note</b> Default PartID is Zero for each NS/NSE Space.</p>	RW	0b0

Bits	Name	Description	Type	Reset
[62:61]	mpam_softlim_value	<p>Soft Limit value for MPAM capacity partitioning.</p> <p><b>0b00</b> Soft limit is 0% below hardlimit.</p> <p><b>0b01</b> Soft limit is 3.13% (1/32) below hardlimit</p> <p><b>0b10</b> Soft limit is 6.25% (1/16) below hardlimit</p> <p><b>0b11</b> Soft limit is 9.38% (3/32) below hardlimit</p> <p><b>NOTE</b> Default is 3.13% below hardlimit. If CMAX value set is at or below 12.5%, soft limit is ignored.</p>	RW	0b01
[60]	dis_rni_dwnstrm_writer_beh	When set, HN-S doesn't do SN read for NC requests from RNI/RND when clean data is available in SLC or upstream RNs Only applicable when dwnstrm_writer_present is set	RW	0b0
[59]	dwnstrm_writer_present	When set, HN-S does SN read for NC requests when clean data is available in SLC or upstream RNs	RW	0b0
[58]	cmopopa_mc_comp	When set, HN-F sends completion for CMOPoPA after completion from SN	RW	0b0
[57:55]	mpam_ns_rl_csumon_cfg	<p>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</p> <p><b>0b000</b> 1 CSUMON allocated to Realm; Remaining CSUMONs allocated to Non-Secure</p> <p><b>0b001</b> 1 CSUMON allocated to Non-Secure; Remaining CSUMONs allocated to Realm</p> <p><b>0b010</b> 1/2 CSUMONS allocated to Non-Secure; 1/2 to Realm. Remaining 1 CSUMON allocated to Non-Secure</p> <p><b>0b011</b> 3/4 CSUMONS allocated to Non-Secure; 1/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure</p> <p><b>0b100</b> 1/4 CSUMONS allocated to Non-Secure; 3/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure</p>	RW	0b0

Bits	Name	Description	Type	Reset
[54:52]	mpam_ns_rl_partid_cfg	<p>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</p> <p><b>0b000</b> 1 PARTID allocated to Realm; Remaining PARTIDs allocated to Non-Secure</p> <p><b>0b001</b> 1 PARTID allocated to Non-Secure; Remaining PARTIDs allocated to Realm</p> <p><b>0b010</b> 1/2 PARTIDs allocated to Non-Secure; 1/2 to Realm. Remaining 1 PARTID allocated to Non-Secure</p> <p><b>0b011</b> 3/4 PARTIDs allocated to Non-Secure; 1/4 to Realm. Remaining 1 PARTID allocated to Non-Secure</p> <p><b>0b100</b> 1/4 PARTIDs allocated to Non-Secure; 3/4 to Realm. Remaining 1 PARTID allocated to Non-Secure</p>	RW	0b0
[51]	slc_victim_clean_tag_transfer_en	When set, HN-S propagates clean tag to SN when SLC victim has clean tag	RW	0b0
[50:48]	Reserved	Reserved	RO	-
[47]	rd_migrate_on_sfmiss	Migrates a read from LCC/SLC if sf miss	RW	0b1
[46]	poison_on_mem_addr_dec_err_en	When set, set poison in read data for CXL address decode error	RW	0b1
[45]	ue_on_mem_addr_dec_err	Log CXL address decode error as UE in error register	RW	0b0
[44]	delay_cacheable_wns_comp	Sends late completion for cacheable WriteNoSnoop	RW	0b0
[43]	alloc_on_pcie_wr_ptl	Forces HBT PCIE partial writes to allocate in SLC	RW	0b0
[42:40]	Reserved	Reserved	RO	-
[39]	sfbi_snpunique_dis	<p>Disable SnpUnique when SFBI is allowed to allocate in SLC. When Set, SFBI only sends SnpCleanInvalid.</p> <p><b>Note</b> Applicable for HBT &amp; LBT both.</p>	RW	0b1
[38]	stash_snp_dis	Disables stashing snoop in HN-S when set to 0b1	RW	0b0
[37]	hbt_sfbi_alloc_slc_en	<p>Enable SLC allocation of Implicit WB data from RN due to SF Eviction of HBT line.</p> <p><b>Note</b> Only Capacity SF Eviction affected.</p>	RW	0b0
[36]	metadata_mode_dis	Disables the METADATA features in HN-S when set to 0b1	RW	0b0
[35]	Reserved	Reserved	RO	-
[34]	sf_rnf_vec_precise_mode_en	Enables the snoop filter's precise RN-F vector in clustered mode when set to 0b1	RW	0b1
[33]	sf_rn_cluster_mode_en	Enables the snoop filter clustering of the RN-F ID's using programmable registers	RW	0b1
[32]	mte_mode_dis	Disables the MTE features in HN-S when set to 0b1. Can only program when POR_CHI_MTE_ENABLE_PARAM is set	RW	0b0

Bits	Name	Description	Type	Reset
[31]	slc_mpamid_as_lru_thread	<p>Use MPAM PARTID as ThreadID for Thread Aware eLRU</p> <p><b>0b0</b> ThreadID is based on LPID+LID for Thread Aware eLRU.</p> <p><b>0b1</b> ThreadID is based on MPAM PARTID+NS for Thread Aware eLRU.</p> <p><b>Note</b> MPAM PARTID is used only if MPAM is enabled.</p>	RW	0b0
[30]	slc_ta_lru_en	<p>Thread Aware eLRU enable</p> <p><b>0b0</b> ThreadID used for eLRU is zero.</p> <p><b>0b1</b> ThreadID used for eLRU is based on MPAMID or LPID+LID.</p> <p><b>Note</b> If SLC size is less than 256KB, this bit is ignore.</p>	RW	0b0
[29:28]	slc_lru_dualmode_insert_value	<p>Insertion value for Dual mode eLRU</p> <p><b>NOTE</b> Default is 0b11.</p>	RW	0b11
[27:26]	slc_lru_staticmode_insert_value	<p>Insertion value for Static mode eLRU</p> <p><b>NOTE</b> Default is 0b10.</p>	RW	0b10
[25:24]	slc_lru_slcsrc_insert_value	<p>Insertion value if SLC source bit is set</p> <p><b>NOTE</b> Default is 0b00.</p>	RW	0b00
[23:22]	slc_lru_sel_cnt_value	<p>Selection counter value for eLRU to determine which group policy is more effective</p> <p><b>0b00</b> Sel counter is like an 8-bit range; upper limit is 255; middle point is 128</p> <p><b>0b01</b> Sel counter is like a 9-bit range; upper limit is 511; middle point is 256</p> <p><b>0b10</b> Sel counter is like a 10-bit range; upper limit is 1023; middle point is 512</p> <p><b>0b11</b> Sel counter is like an 11-bit range; upper limit is 2047; middle point is 1024</p> <p><b>NOTE</b> Default is 10-bit with counter reset to a value of 512.</p>	RW	0b10

Bits	Name	Description	Type	Reset
[21:20]	slc_lru_set_groups	<p>Number of sets in monitor group for enhance LRU</p> <p><b>0b00</b> 16</p> <p><b>0b01</b> 32</p> <p><b>0b10</b> 64</p> <p><b>0b11</b> 128</p> <p><b>NOTE</b> Default is 32 sets per monitor group. If cache size is small (128KB or less), there would be only one set per group.</p>	RW	0b01
[19]	slc_lru_victim_dis	<p>Disable enhanced LRU based victim selection for SLC</p> <p><b>0b0</b> SLC victim selection is based on eLRU.</p> <p><b>0b1</b> SLC victim selection is based on LFSR.</p> <p><b>NOTE</b> Victim selection for SF is always LFSR-based.</p>	RW	0b1
[18]	slc_mpam_ccap_en	<p>Enable MPAM Cache Capacity Partitioning for SLC</p> <p><b>0b1</b> Cache Capacity Partitioning is enabled if supported in Hardware.</p> <p><b>0b0</b> Cache Capacity Partitioning is disabled for SLC.</p> <p><b>NOTE</b> If MPAM is disabled at build time, this bit has no meaning.</p>	RW	0b0
[17]	slc_mpam_cpor_en	<p>Enable MPAM Cache Portion Partitioning for SLC</p> <p><b>0b1</b> Cache Portion Partitioning is enabled if supported in Hardware.</p> <p><b>0b0</b> Cache Portion Partitioning is disabled for SLC.</p> <p><b>NOTE</b> If MPAM is disabled at build time, this bit has no meaning.</p>	RW	0b0
[16]	adv_cbusy_mode_en	Enables the advanced features of HN-S CBusy handling	RW	0b0
[15]	dwt_dis	Disables DWT when set	RW	0b1
[14]	Reserved	Reserved	RO	-
[13]	slc_repl_hint_use_en	<p><b>0b0</b> Interconnect generated SLC Replacement hints are used for eLRU.</p> <p><b>0b1</b> RN-F provided SLC Replacement hints are used for eLRU.</p>	RW	0b1
[12]	pcmo_pop_en	Terminates PCMO in HN-S when this bit is set to 0b1. PCMO is dropped and dirty data is not flushed	RW	0b0

Bits	Name	Description	Type	Reset
[11]	dmt_dis	Disables DMT when set	RW	0b0
[10]	ocm_allways_en	Enables all SLC ways with OCM	RW	0b0
[9]	ocm_en	Enables region locking with OCM support	RW	0b0
[8]	ncdevcmo_mc_comp_en	<p>Disables HN-F completion when set</p> <p><b>NOTE</b></p> <p>When set, HN-F sends completion for the following transactions received after completion from SN</p> <ol style="list-style-type: none"> <li>1. Non-cacheable WriteNoSnp</li> <li>2. Device WriteNoSnp</li> <li>3. CMO (cache maintenance operations)</li> </ol> <p><b>CONSTRAINT</b></p> <p>When this bit is set, por_rni_cfg_ctl.dis_ncwr_stream and por_rnd_cfg_ctl.dis_ncwr_stream must also be set.</p>	RW	0b0
[7]	force_flush_popa_en	<p>Generate PoPA request for SLC and SF flush generated SN writes.</p> <p><b>CONSTRAINT</b></p> <p>force_flush_popa_en is valid only if ABF CleanInvalid mode is set.</p>	RW	0b0
[6]	force_flush_pcmo_deep_en	<p>Make PCMO request for SLC and SF flush generated SN writes as Deep PCMO.</p> <p><b>CONSTRAINT</b></p> <p>hns_force_flush_pcמו_deep_en is valid only if hns_force_flush_pcמו_en bit is set.</p> <p><b>CONSTRAINT</b></p> <p>This bit can be set only if ALL SNs in the system support deep attribute.</p>	RW	0b0
[5]	force_flush_pcמו_en	Generate PCMO request for SLC and SF flush generated SN writes	RW	0b0
[4]	force_flush_ewa0_en	Force SLC and SF flush to use EWA 0 for SN writes	RW	0b0
[3:2]	Reserved	Reserved	RO	-
[1]	ncb_wr_alloc_on_hit	Forces non CopyBack write requests to allocate if the line hit in SLC/SF	RW	0b0
[0]	wlu_alloc_on_hit	Forces WLU requests to allocate if the line hit in SLC	RW	0b0

### 8.3.9.8 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

0xA08

## Type

RW

## Reset value

See individual bit resets

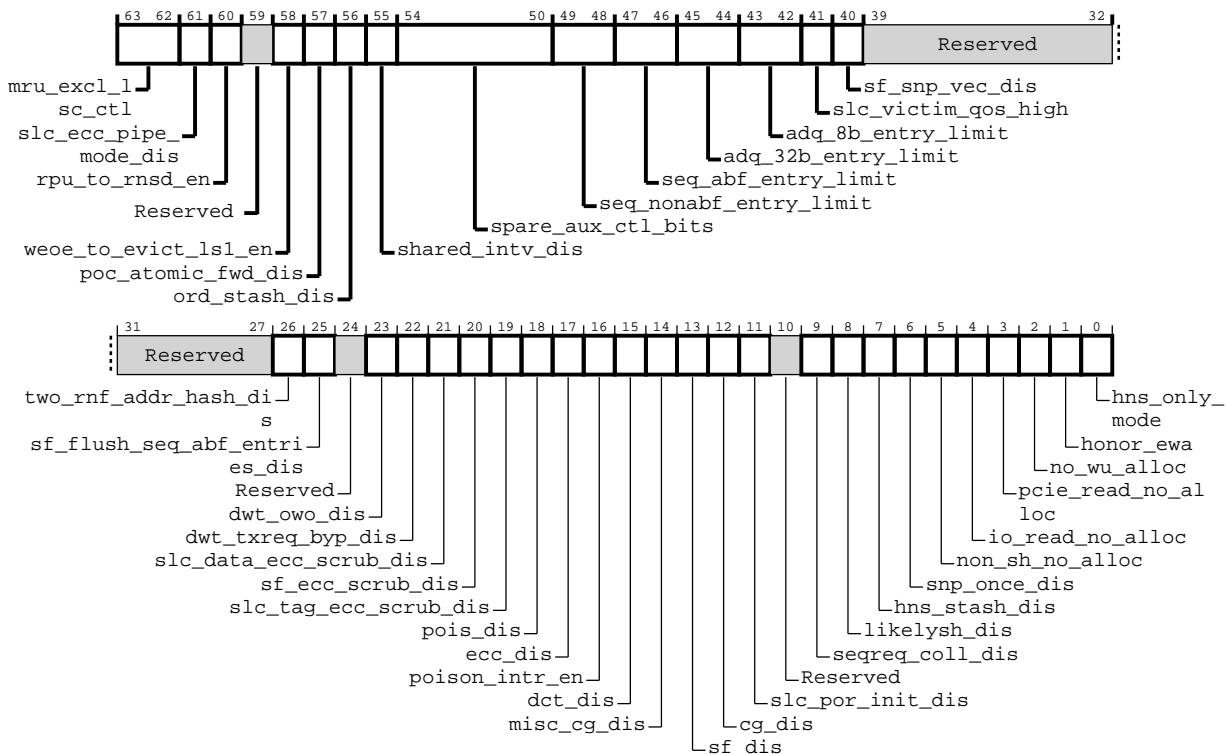
## Usage constraints

This register is owned in the Root space and is 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 8-419: cmn\_hns\_aux\_ctl**



**Table 8-425: cmn\_hns\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:62]	mru_excl_lsc_ctl	MRU Exclusive control for LSC.	RW	0b00
[61]	slc_ecc_pipe_mode_dis	Disables inline ECC pipe mode in SLC. <b>CONSTRAINT</b> Must be programmed at boot time.	RW	0b00

Bits	Name	Description	Type	Reset
[60]	rpu_to_rnsd_en	Enables HN-F to treat ReadPrefUnique ops as ReadNotSharedDirty	RW	0b0
[59]	Reserved	Reserved	RO	-
[58]	woeo_to_evict_ls1_en	Enables HN-F to treat WOE ops as EVICT when LS=1	RW	0b1
[57]	poc_atomic_fwd_dis	Disable the atomic data forwarding in POCQ	RW	0b0
[56]	ord_stash_dis	Disables stash operation for ordered write stash requests	RW	0b0
[55]	shared_intv_dis	Disables snoop requests to CHIB RN-F with shared copy	RW	0x0
[54:50]	spare_aux_ctl_bits	Spare aux control bits for potential ECO fixes.	RW	0b00000
[49:48]	seq_nonabf_entry_limit	<p>Limit number of SEQ Entries used for SF Eviction req.</p> <p><b>00</b> SF Eviction SEQ entry limit is disabled. Default 8 entries are used.</p> <p><b>01</b> Only 2 SF Eviction SEQ entry [1:0] can be used. NOT SUPPORTED FOR HN-S config</p> <p><b>10</b> Only 3 SF Eviction SEQ entry [2:0] can be used.</p> <p><b>11</b> Only 4 SF Eviction SEQ entry [3:0] can be used.</p> <p><b>Note</b> 0b01 (2 entry) is not supported for HN-S config.</p>	RW	0b00
[47:46]	seq_abf_entry_limit	<p>Limit number of SEQ Entries used for ABF Flush req.</p> <p><b>00</b> ABF SEQ entry limit is disabled. Default 4 entries are used.</p> <p><b>01</b> Only 1 ABF SEQ entry [0:0] can be used.</p> <p><b>10</b> Only 2 ABF SEQ entry [1:0] can be used.</p> <p><b>11</b> Only 3 ABF SEQ entry [2:0] can be used.</p>	RW	0b00
[45:44]	adq_32b_entry_limit	<p>Limit number of ADQ 32B entry for debug purpose.</p> <p><b>00</b> ADQ 32B entry limit is disabled. Default 4 entries are used.</p> <p><b>01</b> Only 1 ADQ 32B entry [0:0] can be used. NOT SUPPORTED FOR HN-S config</p> <p><b>10</b> Only 2 ADQ 32B entry [1:0] can be used.</p> <p><b>11</b> Only 3 ADQ 32B entry [2:0] can be used.</p> <p><b>Note</b> 0b01 (1 entry) is not supported for HN-S config as one entry needs to be reserved for HBT.</p>	RW	0b00

Bits	Name	Description	Type	Reset
[43:42]	adq_8b_entry_limit	<p>Limit number of ADQ 8B entry for debug purpose.</p> <p><b>00</b> ADQ 8B entry limit is disabled. Default 8 entries are used.</p> <p><b>01</b> Only 1 ADQ 8B entry [0:0] can be used. NOT SUPPORTED FOR HN-S config</p> <p><b>10</b> Only 2 ADQ 8B entry [1:0] can be used.</p> <p><b>11</b> Only 4 ADQ 8B entry [3:0] can be used.</p> <p><b>Note</b> 0b01 (1 entry) is not supported for HN-S config as one entry needs to be reserved for HBT.</p>	RW	0b00
[41]	slc_victim_qos_high	<p>SLC victim QoS behavior for SN write request</p> <p><b>0b0</b> Each victim inherits the QoS value of the request which caused it</p> <p><b>0b1</b> All victims use high QoS class (14)</p>	RW	0b0
[40]	sf_snp_vec_dis	Disables SF snoop vector when set	RW	0b0
[39:27]	Reserved	Reserved	RO	-
[26]	two_rnf_addr_hash_dis	Disable address hash and use single bit (bit 7) to select RN-F interface when 2RNF is supported	RW	0b0
[25]	sf_flush_seq_abf_entries_dis	When set, disables using SEQ ABF entries for PM SF flush	RW	0b0
[24]	Reserved	Reserved	RO	-
[23]	dwt_owo_dis	Disables DWT for OWO requests	RW	0b0
[22]	dwt_txreq_byp_dis	Disables DWT TXREQ bypass when set	RW	0b1
[21]	slc_data_ecc_scrub_dis	Disables data single-bit ECC error scrubbing for non-migrating reads when set	RW	0b1
[20]	sf_ecc_scrub_dis	Disables SF tag single-bit ECC error scrubbing when set	RW	0b0
[19]	slc_tag_ecc_scrub_dis	Disables SLC tag single-bit ECC error scrubbing when set	RW	0b0
[18]	pois_dis	Disables parity error data poison when set	RW	0b0
[17]	ecc_dis	Disables SLC and SF ECC generation/detection when set	RW	0b0
[16]	poison_intr_en	Enables reporting an interrupt by HN-F when poison is detected at SLC	RW	0x0
[15]	dct_dis	Disables DCT when set	RW	0x0
[14]	misc_cg_dis	Disables architectural clock gates in hns_misc hierarchy	RW	0b0
[13]	sf_dis	Disables SF	RW	0b0
[12]	cg_dis	Disables HN-F architectural clock gates	RW	0b0
[11]	slc_por_init_dis	Disables SLC and SF initialization on Reset	RW	0b0
[10]	Reserved	Reserved	RO	-
[9]	seqreq_coll_dis	-	RW	0b0
[8]	likelysh_dis	Disables Likely Shared based allocations	RW	0b0
[7]	hns_stash_dis	Disables HN-F stash support	RW	0x0
[6]	snp_once_dis	When set, disables SnpOnce and converts to SnpShared	RW	0x0

Bits	Name	Description	Type	Reset
[5]	non_sh_no_alloc	Disables SLC allocation for non-shareable cacheable transactions when set	RW	0b0
[4]	io_read_no_alloc	When set, disables ReadOnce and ReadNoSnp allocation in SLC from RN-Is	RW	0b0
[3]	pcie_read_no_alloc	When set, disables ReadOnce and ReadNoSnp allocation in SLC from PCIE	RW	0b0
[2]	no_wu_alloc	Disables WriteUnique/WriteLineUnique allocations in SLC when set	RW	0b0
[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	0b1
[0]	hns_only_mode	Enables HN-F only mode; disables SLC and SF when set	RW	0b0

### 8.3.9.9 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

0xA10

##### Type

RW

##### Reset value

See individual bit resets

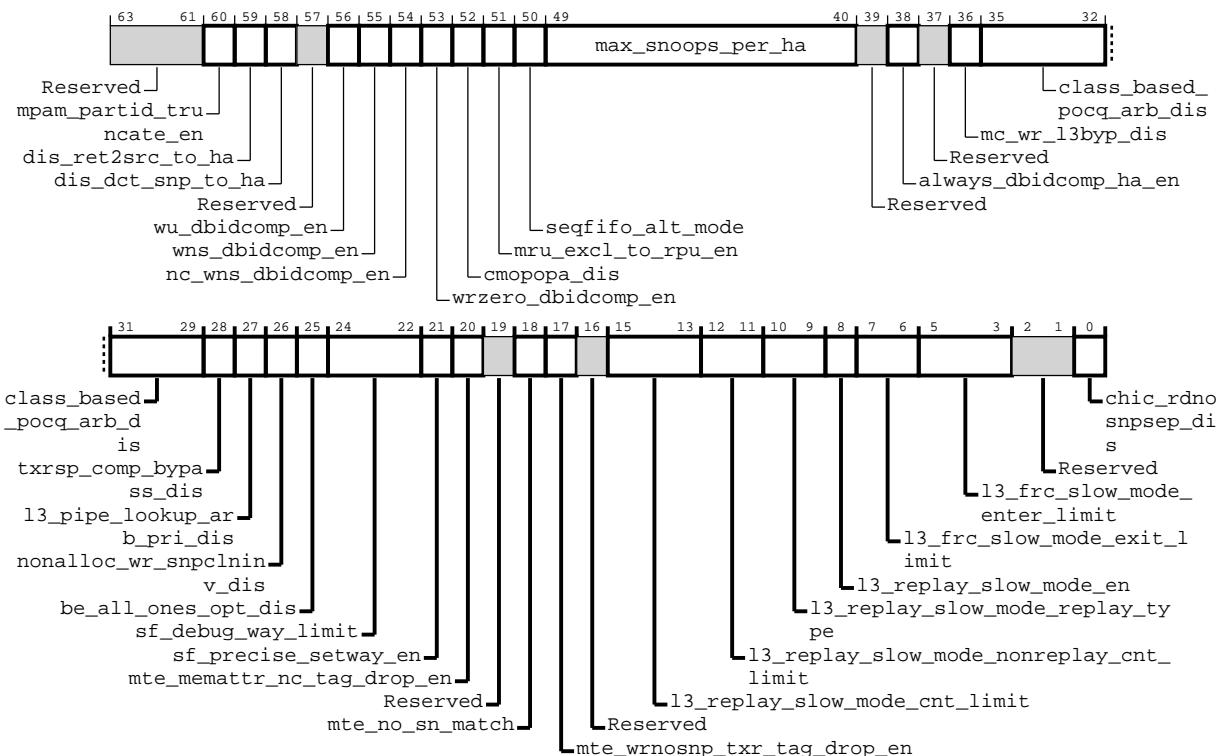
##### Usage constraints

This register is owned in the Root space and is 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 8-420: cmn\_hns\_aux\_ctl\_1**



**Table 8-426: cmn\_hns\_aux\_ctl\_1 attributes**

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60]	mpam_partid_truncate_en	MPAM PartID for all address spaces (S, NS, RL, RT) is truncated if PartID value is above HNS_MPAM_NS_PARTID_MAX_PARAM. <b>Note</b> This truncation is only applied for internal use of MPAM PartID. Propagation is not truncated.	RW	0b0
[59]	dis_ret2src_to_ha	When set to 1, Clears RetToSrc for all snoops going to HA	RW	0b0
[58]	dis_dct.snp_to_ha	When set to 1, disables forwarding snoops to HA	RW	0b0
[57]	Reserved	Reserved	RO	-
[56]	wu_dbidcomp_en	When set, HN-S will combine DBID and Comp response for all WriteUnique requests to post SLC/SF lookup and snoops. Only applicable to HBT transactions	RW	0b0
[55]	wns_dbidcomp_en	When set, HN-S will combine DBID and Comp response for all WriteNoSnoop requests to post SLC/SF lookup and snoops	RW	0b0
[54]	nc_wns_dbidcomp_en	When set, HN-S will combine DBID and Comp response for all NC WriteNoSnoop requests to post SLC/SF lookup and snoops	RW	0b0
[53]	wrzero_dbidcomp_en	When set, HN-S will combine DBID and Comp response for all WriteZero requests to post SLC/SF lookup and snoops Only applicable to HBT transactions	RW	0b1
[49]	max_snoops_per_ha	Maximum number of snoops per HA		
[40:32]	class_based_pocq_arb_dis, mc_wr_13byp_dis, always_dbidcomp_ha_en	Configuration for class-based POCQ arbitration and MC write bypass		
[31:29]	chic_rdnosnpsep_d	Configuration for CHIC RDNO and SNPSEPD		
[28]	txrsp_comp_bypa_ss_dis	TX/RSP Comp bypass and SS disable		
[27]	13_pipe_lookup_ar_b_pri_dis	13-Pipe lookup and B-Priority disable		
[26]	nonalloc_wr_snplnin_v_dis	Non-alloc write snoop enable and V-disable		
[25]	be_all_ones_opt_dis_sf_debug_way_limit	BE All Ones Opt Dis and SF Debug Way Limit		
[24]	sf_precise_setway_en_mte_memattr_nc_tag_drop_en	SF Precise Setway enable and MTE MemAttr NC Tag Drop enable		
[22]	mte_no_sn_match	MTE No SN Match		
[19]	13_frc_slow_mode_enter_limit	13-FRC Slow Mode Enter Limit		
[18]	13_frc_slow_mode_exit_1	13-FRC Slow Mode Exit 1		
[17]	13_replay_slow_mode_en	13-Replay Slow Mode enable		
[16]	13_replay_slow_mode_replay_type	13-Replay Slow Mode Replay Type		
[15]	13_replay_slow_mode_nonreplay_cnt_limit	13-Replay Slow Mode Nonreplay Count Limit		
[13]	13_replay_slow_mode_cnt_limit	13-Replay Slow Mode Count Limit		
[12]	13_replay_slow_mode_mte_wrnosnp_txr_tag_drop_en	13-Replay Slow Mode MTE WRNOSNP TXR Tag Drop enable		

Bits	Name	Description	Type	Reset
[52]	cmopopa_dis	When set, HN-F downgrades CleanInvalidPoPA to CleanInvalid	RW	0b0
[51]	mru_excl_to_rpu_en	When set, Excl MRU will send RPU flow when fail and SF hit EU but not self hit	RW	0b1
[50]	seqfifo_alt_mode	HN-S will use alternate mode to select SEQ entries.	RW	0b0
[49:40]	max_snoops_per_ha	HN-S will use the register value instead of the parameter to control Max snoops per HA	RW	Configuration dependent
[39]	Reserved	Reserved	RO	-
[38]	always_dbidcomp_ha_en	When set, HN-S will combine DBID and Comp response for all writeunique and writemosnoop requests from CXHA to post SLC/SF lookup and snoops	RW	0b0
[37]	Reserved	Reserved	RO	-
[36]	mc_wr_l3byp_dis	When set, disables l3 bypass path to mc request for writes	RW	0b0
[35:29]	class_based_pocq_arb_dis	<p>Disables Class based arbitration for various POCQ arbiters. For each</p> <p><b>0b0</b> Use Class based arbitration.</p> <p><b>0b1</b> Use QoS based arbitration. Legacy mode.</p> <p>[35] POCQ entry selection for SN Static Credit Grant return.</p> <p>[34] POCQ entry selection for SLC/SF pipeline request.</p> <p>[33] POCQ entry selection for TXRSP.</p> <p>[32] POCQ entry selection for TXDAT.</p> <p>[31] POCQ entry selection for TXREQ.</p> <p>[30] POCQ entry selection for ADQ.</p> <p>[29] Reserved for future use.</p>	RW	0b0000000
[28]	txrsp_comp_bypass_dis	When set, TXRSP COMP bypass gets disabled for WEOE/EVICT	RW	0b1
[27]	l3_pipe_lookup_arb_pri_dis	When set to 0b1, L3 pipe arbiter will not give priority to lookup requests over fill requests.	RW	0b0
[26]	nonalloc_wr_snpclninv_dis	Disable the.snp type of.snp_cln_inv on non-allocating writes. Send.snp_uniq instead	RW	0b0
[25]	be_all_ones_opt_dis	Disable the optimizations related to BE=1's hint on WR_PTL from RNI	RW	0b0

Bits	Name	Description	Type	Reset
[24:22]	sf_debug_way_limit	<p>Limit number of SF ways for debug purpose.</p> <p><b>000</b> SF Way limit is disabled. SF_NUM_WAYS_PARAM determines number of ways supported</p> <p><b>001</b> Only 1 SF way[0:0] can be allocated. NOT SUPPORTED FOR HN-S config</p> <p><b>010</b> Only 2 SF ways[1:0] can be allocated.</p> <p><b>011</b> Only 4 SF ways[3:0] can be allocated.</p> <p><b>100</b> Only 8 SF ways[7:0] can be allocated.</p> <p><b>101</b> Only 12 SF ways[11:0] can be allocated.</p> <p><b>110</b> Only 16 SF ways[15:0] can be allocated.</p> <p><b>Note</b> HBT/CCG has a reserved way (MSB way) that's independent of this limit.</p> <p><b>Note</b> 0b001 (1 entry) is not supported for HN-S config. It's only supported for LCN or HNF config.</p> <p><b>Note</b> SF Way Blocking (cmn_hns_sf_cxg_blocked_ways) is not supported when sf_debug_way_limit is non-zero.</p>	RW	0b000
[21]	sf_precise_setway_en	Enables Precise setway hazard, when set to 0b1	RW	0b1
[20]	mte_memattr_nc_tag_drop_en	Enables HN-S to drop any dirty tags for Non-Cacheable memory, when set to 0b1	RW	0b0
[19]	Reserved	Reserved	RO	-
[18]	mte_no_sn_match	When set, HN-S does MTE match locally without propagating to SN	RW	0b1
[17]	mte_wrnosnp_txr_tag_drop_en	When set to 0b1, HN-S will drop clean tags from a WriteNoSnp with tagop Transfer	RW	0b0
[16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:13]	l3_replay_slow_mode_cnt_limit	<p>L3 arbitration throttle count limit, when enabled.</p> <p><b>000</b> L3 Throttle is enabled after 64 replays</p> <p><b>001</b> L3 Throttle is enabled after 128 replays</p> <p><b>010</b> L3 Throttle is enabled after 256 replays</p> <p><b>011</b> L3 Throttle is enabled after 512 replays</p> <p><b>100</b> L3 Throttle is enabled after 1024 replays</p> <p><b>101</b> L3 Throttle is enabled after 2048 replays</p> <p><b>110</b> L3 Throttle is enabled after 4096 replays</p> <p><b>111</b> L3 Throttle is enabled after 8192 replays</p>	RW	0b101
[12:11]	l3_replay_slow_mode_nonreplay_cnt_limit	<p>Number of required back to back transactions that don't replay to exit replay mode.</p> <p><b>00</b> Exit slow mode after 8 back to back non-replayed ops</p> <p><b>01</b> Exit slow mode after 16 back to back non-replayed ops</p> <p><b>10</b> Exit slow mode after 32 back to back non-replayed ops</p> <p><b>11</b> Exit slow mode after 64 back to back non-replayed ops</p>	RW	0b01
[10:9]	l3_replay_slow_mode_replay_type	<p>HN-S replay type that will enable slow mode.</p> <p><b>00</b> Replay slow mode is enabled for any replays</p> <p><b>01</b> Replay slow mode is enabled for setway replays</p> <p><b>10</b> Replay slow mode is enabled for seq full replays</p>	RW	0b00
[8]	l3_replay_slow_mode_en	Enables L3 arbitration slow mode in case of constant replays, when set to 0b1	RW	0b1

Bits	Name	Description	Type	Reset
[7:6]	I3_frc_slow_mode_exit_limit	Exit L3 pipe slow mode based on cycle count. <b>00</b> Slow mode forced exit disabled <b>01</b> Exit slow mode 1K cycles after entering slow mode <b>10</b> Exit slow mode 2K cycles after entering slow mode <b>11</b> Exit slow mode 4K cycles after entering slow mode	RW	0b00
[5:3]	I3_frc_slow_mode_enter_limit	Enter L3 pipe slow mode based on cycle count. <b>000</b> Slow mode forced entry disabled <b>001</b> Enter slow mode 4K cycles after last slow mode exit <b>010</b> Enter slow mode 8K cycles after last slow mode exit <b>011</b> Enter slow mode 16K cycles after last slow mode exit <b>100</b> Enter slow mode 32K cycles after last slow mode exit <b>101</b> Enter slow mode 64K cycles after last slow mode exit <b>110</b> Enter slow mode 128K cycles after last slow mode exit <b>111</b> Enter slow mode 256K cycles after last slow mode exit	RW	0b000
[2:1]	Reserved	Reserved	RO	-
[0]	chic_rdnosnpsep_dis	Disables separation of Data and Comp in CHIC mode	RW	0b0

### 8.3.9.10 cmn\_hns\_cbusy\_limit\_ctl

Cbusy threshold limits for POCQ entries.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA18

## Type

RW

## Reset value

See individual bit resets

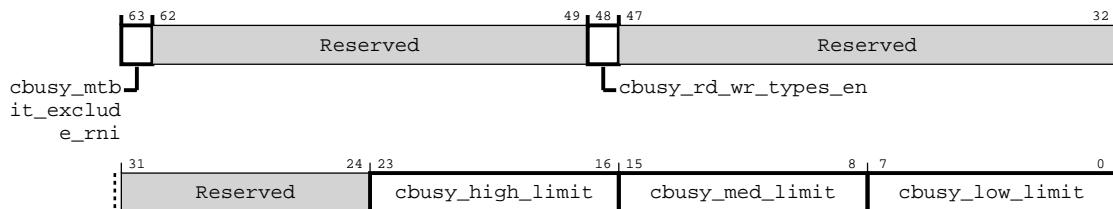
## Usage constraints

This register is owned in the Root space and is 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 8-421: cmn\_hns\_cbusy\_limit\_ctl**



**Table 8-427: cmn\_hns\_cbusy\_limit\_ctl attributes**

Bits	Name	Description	Type	Reset
[63]	cbusy_mtbit_exclude_rni	Exclude RNI sources in multi-source mode	RW	0b0
[62:49]	Reserved	Reserved	RO	-
[48]	cbusy_rd_wr_types_en	When set, CBusy for Reads and Writes are handled independently. The thresholds specified in this register are used for Read request types in POCQ	RW	0b0
[47:24]	Reserved	Reserved	RO	-
[23:16]	cbusy_high_limit	POCQ limit for CBusy High	RW	0x18
[15:8]	cbusy_med_limit	POCQ limit for CBusy Med	RW	0x10
[7:0]	cbusy_low_limit	POCQ limit for CBusy Low	RW	0x8

## 8.3.9.11 cmn\_hns\_txrsp\_arb\_weight\_ctl

TXRSP arbitration weight controls.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA20

### Type

RW

### Reset value

See individual bit resets

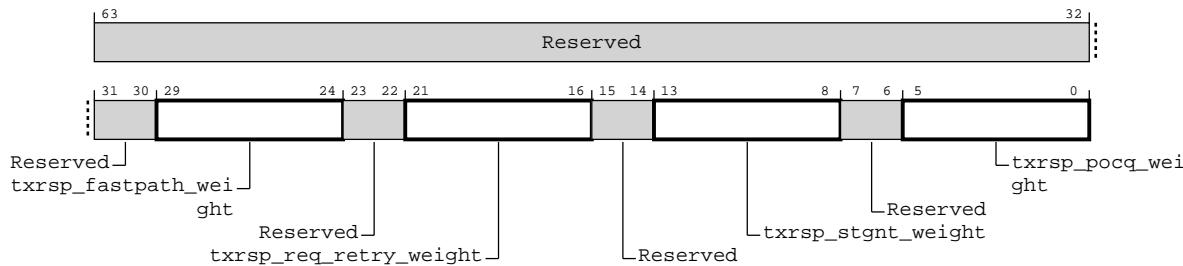
### Usage constraints

This register is owned in the Root space and is 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 8-422: cmn\_hns\_txrsp\_arb\_weight\_ctl**



**Table 8-428: cmn\_hns\_txrsp\_arb\_weight\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	txrsp_fastpath_weight	Fastpath response weights for TXRSP channel	RW	0b111111
[23:22]	Reserved	Reserved	RO	-
[21:16]	txrsp_req_retry_weight	Request retry response weights for TXRSP channel	RW	0b000001
[15:14]	Reserved	Reserved	RO	-
[13:8]	txrsp_stgnt_weight	Static Credit Grant response weights for TXRSP channel	RW	0b000001
[7:6]	Reserved	Reserved	RO	-
[5:0]	txrsp_pocq_weight	POCQ response weights for TXRSP channel	RW	0b000001

### 8.3.9.12 cmn\_hns\_cbusy\_mode\_ctl

Control register for additional CBusy controls

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA28

### Type

RW

### Reset value

See individual bit resets

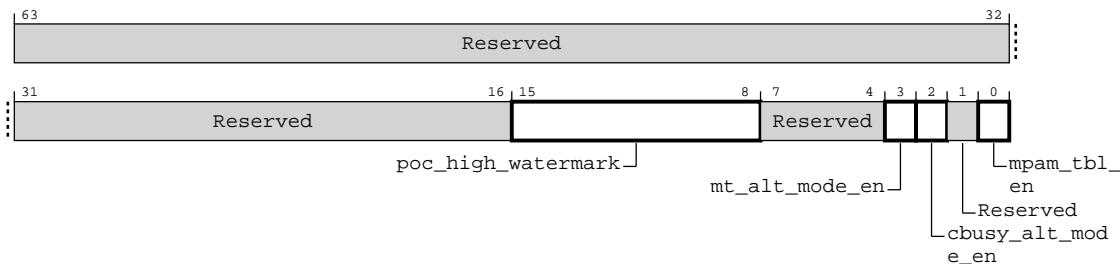
### Usage constraints

This register is owned in the Root space and is 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 8-423: cmn\_hns\_cbusy\_mode\_ctl**



**Table 8-429: 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	0x18
[7:4]	Reserved	Reserved	RO	-
[3]	mt_alt_mode_en	Enable CBusy[2] alternate reporting mode:  <b>0b0</b> POCQ has requests from more than one source  <b>0b1</b> POCQ Occupancy is higher than the poc_high_watermark	RW	0b0
[2]	cbusy_alt_mode_en	Enables an alternate mode of SN CBusy[1:0] capture for mpam_tbl_en=1 mode:  <b>0b0</b> For each MPAM partID, CBusy[1:0] = SN_CBusy[1:0]  <b>0b1</b> For each MPAM partID, CBusy[1] = SN's CBusy[2], CBusy[0] = (SN_CBusy[1] & SN_CBusy[0])	RW	0b0

Bits	Name	Description	Type	Reset
[1]	Reserved	Reserved	RO	-
[0]	mpam_tbl_en	Enables cbusy reporting based on MPAM part ID	RW	0b0

### 8.3.9.13 cmn\_hns\_lbt\_cfg\_ctl

Functions as the configuration control register for HN-F. Only applicable to LBT transactions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA30

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.cfg\_ctl

##### Secure group override

cmn\_hns\_scr.cfg\_ctl

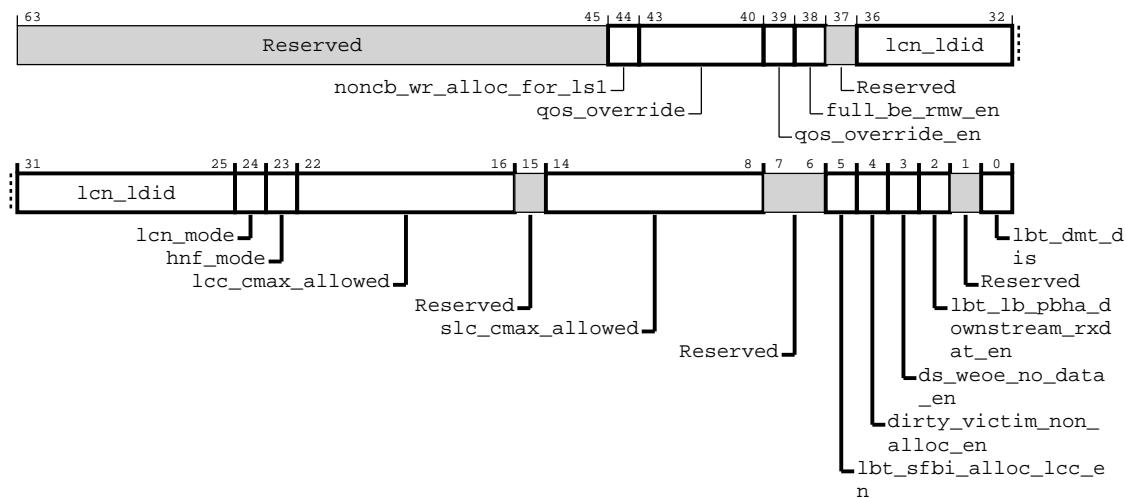
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-424: cmn\_hns\_lbt\_cfg\_ctl**



**Table 8-430: cmn\_hns\_lbt\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44]	noncb_wr_alloc_for_ls1	Drive memattr.allocate for Non Copyback writes when LS=1 for LBT requests	RW	0b1
[43:40]	qos_override	QoS override value for LBT requests	RW	0x0
[39]	qos_override_en	When set, QoS value for LBT requests is driven from QoS override value in this register	RW	0b0
[38]	full_be_rmw_en	When set, LCN issues CleanUnique followed by WriteBackFull for non-allocating full BE and full opcode writes from RNI	RW	0b1
[37]	Reserved	Reserved	RO	-
[36:25]	lcn_ldid	Assigned LDID for LBT TXREQ.	RW	0b0000000000000000
[24]	lcn_mode	Remove all HBT reservations. Can program to 0b1 when HNS only receives LBT transactions	RW	0b0
[23]	hnf_mode	When set, HNS is in HNF mode and some reservations are removed. Can program to 0b1 when HNS only receives HBT transactions	RW	0b0
[22:16]	lcc_cmax_allowed	Maximum cache capacity usage in fixed-point fraction of the cache capacity by LBT lines	RW	0b1111111
[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	0b1111111
[7:6]	Reserved	Reserved	RO	-
[5]	lbt_sfbi_alloc_lcc_en	Enable LCC allocation of Implicit WB data from RN due to SF Eviction of LBT line.  <b>Note</b> Only Capacity SF Eviction affected.	RW	0b0
[4]	dirty_victim_non_alloc_en	When HNS issues dirty CopyBack writes for LCC victim or SFBI, set non-allocating type	RW	0b0

Bits	Name	Description	Type	Reset
[3]	ds_weoe_no_data_en	When HNS issues WriteEvictOrEvict downstream, force no data transfer if config bit is set	RW	0b0
[2]	lbt_lb_pbha_downstream_rxdat_en	Takes LB/PBHA values from downstream RXDAT for LBT lines	RW	0b0
[1]	Reserved	Reserved	RO	-
[0]	lbt_dmt_dis	Disables DMT when set	RW	0b0

### 8.3.9.14 cmn\_hns\_lbt\_aux\_ctl

Functions as the auxiliary control register for HN-F. Only applicable to LBT transactions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA38

##### Type

RW

##### Reset value

See individual bit resets

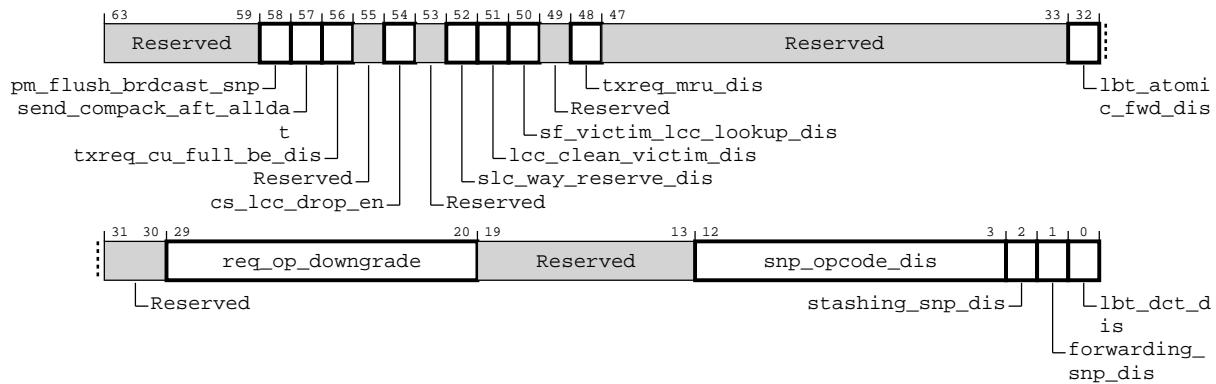
##### Usage constraints

This register is owned in the Root space and is 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 8-425: cmn\_hns\_lbt\_aux\_ctl**



**Table 8-431: 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 0b0 only when it's guaranteed that no coherence traffic comes to HN-S during PM flush.	RW	0b1
[57]	send_compack_aft_alldat	Enables sending CompAck after all data beats are received	RW	0b0
[56:55]	Reserved	Reserved	RO	-
[54]	cs_lcc_drop_en	Enables LCC to drop clean copy after writing dirty data for CleanShared.	RW	0b1
[53]	Reserved	Reserved	RO	-
[52]	slc_way_reserve_dis	Disables SLC reserved way for HBT lines in HN-S mode and allow LBT lines to take all ways in system cache.	RW	0b1
[51]	lcc_clean_victim_dis	Disables LCC sending clean eviction to Home Node.	RW	0b0
[50]	sf_victim_lcc_lookup_dis	Disables LCC lookup for a SF victim.	RW	0b0
[49]	Reserved	Reserved	RO	-
[48]	txreq_mru_dis	Disables sending MRU opcode downstream. Use RDUNIQ instead	RW	0b0
[47:33]	Reserved	Reserved	RO	-
[32]	lbt_atomic_fwd_dis	Disable the atomic data forwarding in POCQ for LBT atomics	RW	0b0
[31:30]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[29:20]	req_op_downgrade	<p>Downgrades req opcode when set</p> <p>[20] Change READSHARED to READUNIQ</p> <p>[21] Change READNOTSHAREDDIRTY to READUNIQ</p> <p>[22] Change READPREFERUNIQ to READUNIQ</p> <p>[23] Change READONCEMKINV to READONCE</p> <p>[24] Change READONCECLNINV to READONCE</p> <p>[25] Change MAKEUNIQ to CLNUINQ</p> <p>[26] Change MAKEREADUNIQ to READUNIQ</p> <p>[27] Change WRITEEVICTOREVICT to EVICT</p> <p>[28] Change MAKEINVALID to CLEANINVALID</p> <p>[29] Change WRITEUNIQUEFULL to non-allocating WRITEUNIQUEPTL</p>	RW	0x000
[19:13]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[12:3]	snp_opcode_dis	<p>Disables support for RXSNP different snoop opcodes and changes to similar snoop opcode when set</p> <ul style="list-style-type: none"> <li>[3] Change SNPSTASHSHARED to SNPQUERY</li> <li>[4] Change SNPSTASHUNIQUE to SNPQUERY</li> <li>[5] Change SNPMAKEINVALIDSTASH to SNPMAKEINVALID</li> <li>[6] Change SNPCLEANSHARED to SNPCLEANINVALID</li> <li>[7] Change SNPPREFERUNIQUE(FWD) to SNPUNIQUE</li> <li>[8] Change SNPNOTSHAREDDIRTY(FWD) to SNPUNIQUE</li> <li>[9] Change SNPCLEAN(FWD) to SNPUNIQUE</li> <li>[10] Change SNPUNIQUESTASH to SNPUNIQUE</li> <li>[11] Change SNPONCE(FWD) to SNPUNIQUE</li> <li>[12] Change SNPSHARED(FWD) to SNPUNIQUE</li> </ul>	RW	0x000
[2]	stashing_snp_dis	Disables Stashing type of snoops when set for RXSNP	RW	0b1
[1]	forwarding_snp_dis	Disables Forwarding type of snoops when set for RXSNP	RW	0b1
[0]	lbt_dct_dis	Disables DCT when set	RW	0b0

### 8.3.9.15 cmn\_hns\_datasource\_ctl

Functions as control register to determine how to drive the CHI DAT.DataSource field for data provided from SLC

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA50

##### Type

RW

### Reset value

See individual bit resets

### Root group override

`cmn_hns_rcr.cfg_ctl`

### Secure group override

`cmn_hns_scr.cfg_ctl`

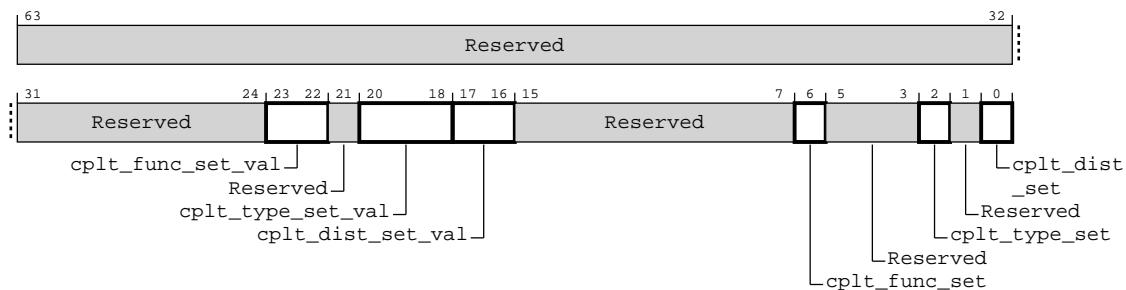
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.cfg_ctl` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.cfg_ctl` bit and `cmn_hns_rcr.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 8-426: cmn\_hns\_datasource\_ctl**



**Table 8-432: cmn\_hns\_datasource\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:22]	cplt_func_set_val	Completer Functional value to use when cplt_func_set is set to 1.	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	Completer Type value to use when cplt_type_set is set to 1. <b>CONSTRAINT</b> Must be set to one of the cache groups	RW	0b101
[17:16]	cplt_dist_set_val	Completer Distance value to use when cplt_dist_set is set to 1.	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	cplt_func_set	Completer Functional Set. Set Completer Functional field to cplt_func_set_val when set to 1	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	Completer Type Set. 0b0 - Reserved, 0b1 - Set Completer Type field to cplt_type_set_val	RW	0b1
[1]	Reserved	Reserved	RO	-
[0]	cplt_dist_set	Completer Distance Set. 0b0 - Reserved, 0b1 - Set Completer Distance field to cplt_dist_set_val	RW	0b1

### 8.3.9.16 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

0x1900

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ppu

##### Secure group override

cmn\_hns\_scr.ppu

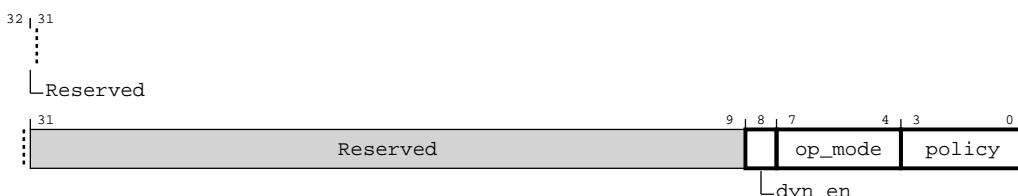
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ppu bit and cmn\_hns\_rcr.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 8-427: cmn\_hns\_ppu\_pwpr**



**Table 8-433: cmn\_hns\_ppu\_pwpr attributes**

Bits	Name	Description	Type	Reset
[31:9]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[8]	dyn_en	Dynamic transition enable	RW	0b0
[7:4]	op_mode	HN-F operational power mode <b>0b0011</b> FAM <b>0b0010</b> HAM <b>0b0001</b> SFONLY <b>0b0000</b> NOSFSLC	RW	0b0
[3:0]	policy	HN-F power mode policy <b>0b1000</b> ON <b>0b0111</b> FUNC_RET <b>0b0010</b> MEM_RET <b>0b0000</b> OFF	RW	0b0

### 8.3.9.17 cmn\_hns\_ppu\_pwsr

Provides power status information for HN-F.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0x1908

##### Type

RO

##### Reset value

See individual bit resets

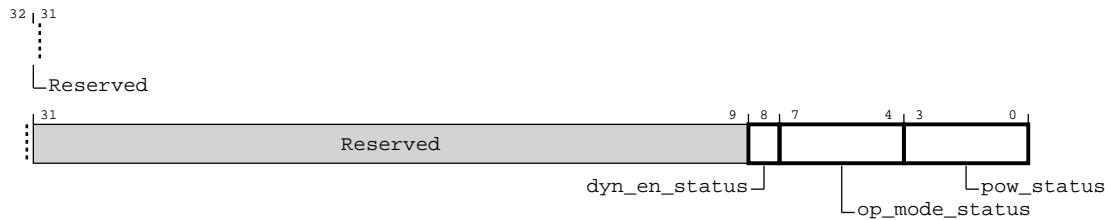
##### Usage constraints

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 8-428: cmn\_hns\_ppu\_pwsr**



**Table 8-434: cmn\_hns\_ppu\_pwsr attributes**

Bits	Name	Description	Type	Reset
[31:9]	Reserved	Reserved	RO	
[8]	dyn_en_status	Dynamic transition status	RO	0b0
[7:4]	op_mode_status	HN-F operational mode status <b>0b0011</b> FAM <b>0b0010</b> HAM <b>0b0001</b> SFONLY <b>0b0000</b> NOSFSLC	RO	0b0
[3:0]	pow_status	HN-F power mode status <b>0b1000</b> ON <b>0b0111</b> FUNC_RET <b>0b0010</b> MEM_RET <b>0b0000</b> OFF	RO	0b0

### 8.3.9.18 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

0x1914

### Type

RO

### Reset value

See individual bit resets

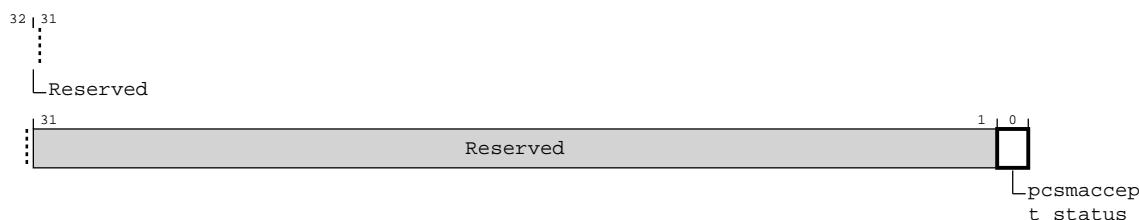
### Usage constraints

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 8-429: cmn\_hns\_ppu\_misr**



**Table 8-435: cmn\_hns\_ppu\_misr attributes**

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	
[0]	pcsmaccep_t_status	HN-F RAM PCSMACCEPT status	RO	0b0

## 8.3.9.19 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

0x28B0

## Type

RO

## Reset value

See individual bit resets

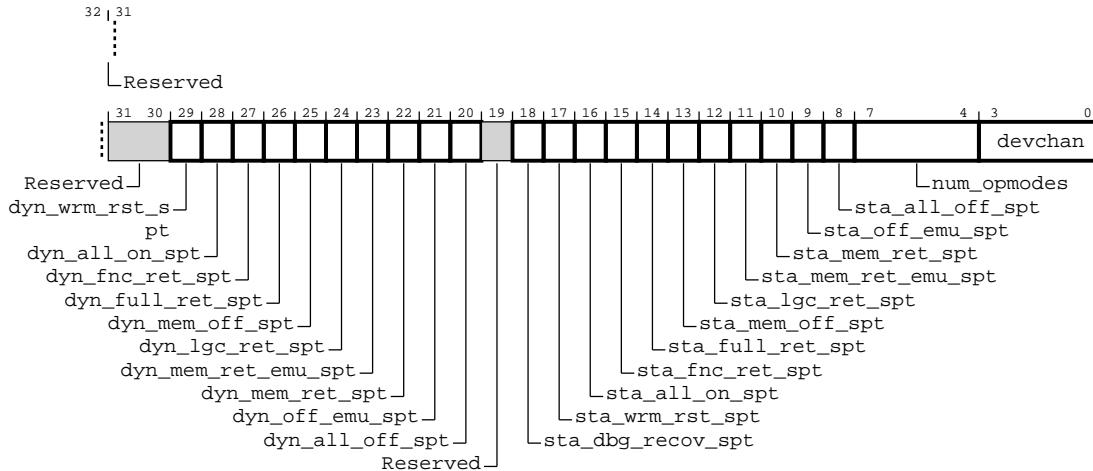
## Usage constraints

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 8-430: cmn\_hns\_ppu\_idr0**



**Table 8-436: 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	0b0
[28]	dyn_all_on_spt	Dynamic on support	RO	0b0
[27]	dyn_fnc_ret_spt	Dynamic func_ret support	RO	0b1
[26]	dyn_full_ret_spt	Dynamic full_ret support	RO	0b0
[25]	dyn_mem_off_spt	Dynamic mem_off support	RO	0b0
[24]	dyn_lgc_ret_spt	Dynamic logic_ret support	RO	0b0
[23]	dyn_mem_ret_emu_spt	Dynamic mem_ret_emu support	RO	0b0
[22]	dyn_mem_ret_spt	Dynamic mem_ret support	RO	0b0
[21]	dyn_off_emu_spt	Dynamic off_emu support	RO	0b0
[20]	dyn_all_off_spt	Dynamic off support	RO	0b0

Bits	Name	Description	Type	Reset
[19]	Reserved	Reserved	RO	-
[18]	sta_dbg_recov_spt	Static dbg_recov support	RO	0b0
[17]	sta_wrm_RST_spt	Static warm_RST support	RO	0b0
[16]	sta_all_on_spt	Static on support	RO	0b1
[15]	sta_fnc_ret_spt	Static func_ret support	RO	0b1
[14]	sta_full_ret_spt	Static full_ret support	RO	0b0
[13]	sta_mem_off_spt	Static mem_off support	RO	0b1
[12]	sta_lgc_ret_spt	Static logic_ret support	RO	0b0
[11]	sta_mem_ret_emu_spt	Static mem_ret_emu support	RO	0b0
[10]	sta_mem_ret_spt	Static mem_ret support	RO	0b1
[9]	sta_off_emu_spt	Static off_emu support	RO	0b0
[8]	sta_all_off_spt	Static off support	RO	0b1
[7:4]	num_opmodes	Number of operational modes	RO	0b0100
[3:0]	devchan	Number of device interface channels	RO	0b0

### 8.3.9.20 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

0x28B4

##### Type

RO

##### Reset value

See individual bit resets

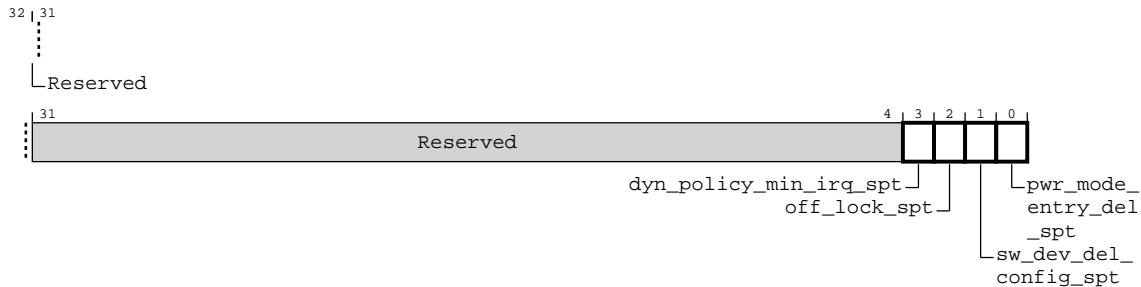
#### Usage constraints

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 8-431: cmn\_hns\_ppu\_idr1**



**Table 8-437: cmn\_hns\_ppu\_idr1 attributes**

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	RO	
[3]	dyn_policy_min_irq_spt	Dynamic minimum policy interrupt support	RO	0b0
[2]	off_lock_spt	Off and mem_ret lock support	RO	0b0
[1]	sw_dev_del_config_spt	Software device delay control configuration support	RO	0b0
[0]	pwr_mode_entry_del_spt	Power mode entry delay support	RO	0b0

### 8.3.9.21 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

0x28C8

##### Type

RO

##### Reset value

See individual bit resets

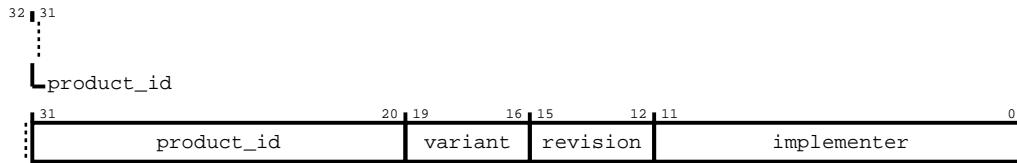
#### Usage constraints

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 8-432: cmn\_hns\_ppu\_iidr**



**Table 8-438: cmn\_hns\_ppu\_iidr attributes**

Bits	Name	Description	Type	Reset
[31:20]	product_id	Implementation identifier	RO	0x434
[19:16]	variant	Implementation variant	RO	0x0
[15:12]	revision	Implementation revision	RO	0x0
[11:0]	implementer	Arm implementation	RO	0x43B

### 8.3.9.22 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

0x28CC

##### Type

RO

##### Reset value

See individual bit resets

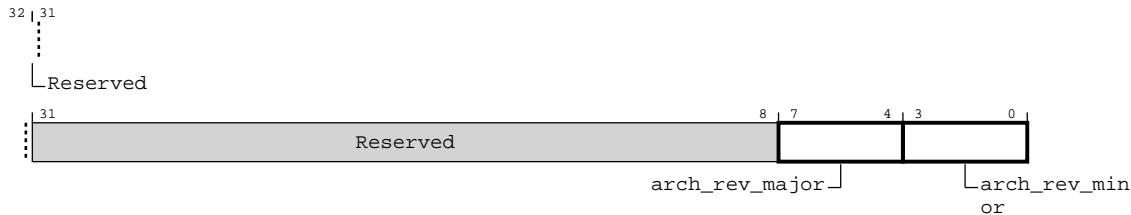
##### Usage constraints

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 8-433: cmn\_hns\_ppu\_aistr**



**Table 8-439: cmn\_hns\_ppu\_aistr attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	arch_rev_major	PPU architecture major revision	RO	0x1
[3:0]	arch_rev_minor	PPU architecture minor revision	RO	0x1

### 8.3.9.23 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

0x1A00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ppu

##### Secure group override

cmn\_hns\_scr.ppu

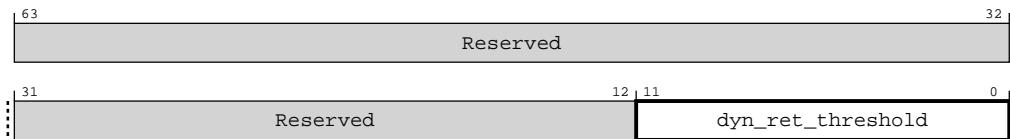
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ppu bit and cmn\_hns\_rcr.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 8-434: cmn\_hns\_ppu\_dyn\_ret\_threshold**



**Table 8-440: 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	0b0

### 8.3.9.24 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

0xA80

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.qos

##### Secure group override

cmn\_hns\_scr.qos

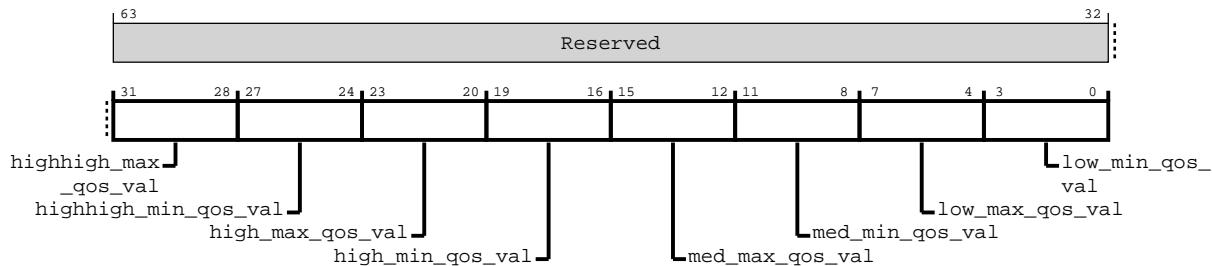
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.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 8-435: cmn\_hns\_qos\_band**



**Table 8-441: 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	0xF
[27:24]	highhigh_min_qos_val	Minimum value for HighHigh QoS class	RO	0xF
[23:20]	high_max_qos_val	Maximum value for High QoS class	RO	0xE
[19:16]	high_min_qos_val	Minimum value for High QoS class	RO	0xC
[15:12]	med_max_qos_val	Maximum value for Medium QoS class	RO	0xB
[11:8]	med_min_qos_val	Minimum value for Medium QoS class	RO	0x8
[7:4]	low_max_qos_val	Maximum value for Low QoS class	RO	0x7
[3:0]	low_min_qos_val	Minimum value for Low QoS class	RO	0x0

## 8.3.9.25 cmn\_hns\_errfr

Functions as the error feature register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xE000

#### Type

RO

#### Reset value

See individual bit resets

## Root group override

cmn\_hns\_rcr.ras

## Secure group override

cmn\_hns\_scr.ras

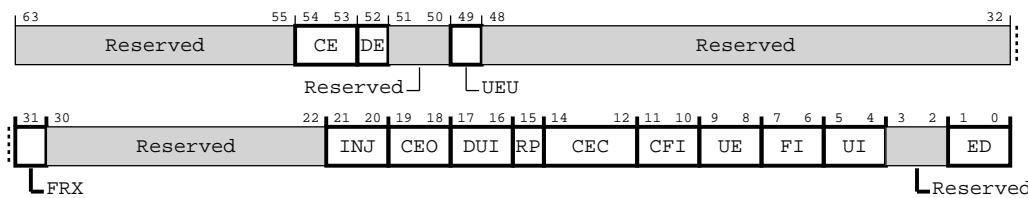
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-436: cmn\_hns\_errfr**



**Table 8-442: cmn\_hns\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording  <b>0b00</b> Corrected Error not supported  <b>0b10</b> Non-specific Corrected Error supported	RO	0b10
[52]	DE	Deferred Error recording  <b>0b0</b> Deferred Error not supported  <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	FRX	Feature Register extension.  <b>0b1</b> cmn_hns_errfr[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension.  <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite.  <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using cmn_hns_errctlr.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when cmn_hns_errfr.CEC != 0b000.)  <b>0b0</b> Invalid;  <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in cmn_hns_errmisc0	RO	0b1
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter in cmn_hns_errmisc0	RO	0b100
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using cmn_hns_errctlr.CFI.	RO	0b10
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using cmn_hns_errctlr.FI.	RO	0b10

Bits	Name	Description	Type	Reset
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using cmn_hns_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using cmn_hns_errctlr.ED	RO	0b10

### 8.3.9.26 cmn\_hns\_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferral are enabled.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE008

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ras

##### Secure group override

cmn\_hns\_scr.ras

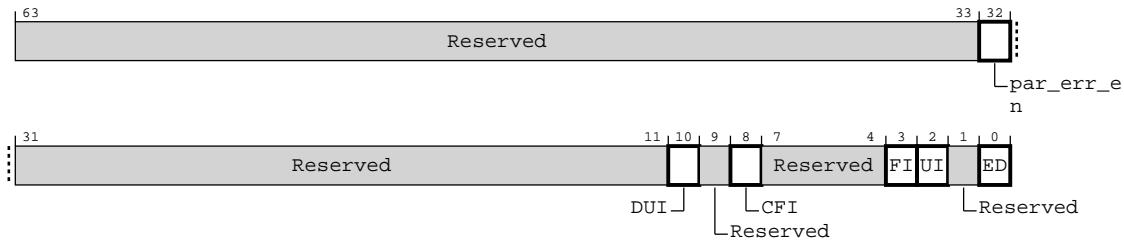
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-437: cmn\_hns\_errctlr**



**Table 8-443: 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 0b1	RW	0b0
[31:11]	Reserved	Reserved	RO	
[10]	DUI	Enables error recovery interrupt for deferred error as specified in cmn_hns_errfr.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in cmn_hns_errfr.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in cmn_hns_errfr.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in cmn_hns_errfr.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in cmn_hns_errfr.ED	RW	0b0

### 8.3.9.27 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

0xE010

##### Type

W1C

##### Reset value

See individual bit resets

### Root group override

`cmn_hns_rcr.ras`

### Secure group override

`cmn_hns_scr.ras`

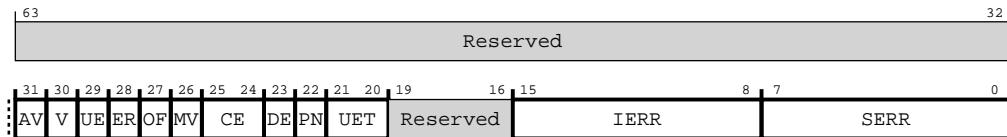
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.ras` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.ras` bit and `cmn_hns_rcr.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 8-438: cmn\_hns\_errstatus**



**Table 8-444: cmn\_hns\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; cmn_hns_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0

Bits	Name	Description	Type	Reset
[28]	ER	Error Reported  <b>0b1</b> In-band error response signaled to the Requester  <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected  <b>0b1</b> More than one error detected  <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	cmn_hns_errmisc<01> valid  <b>0b1</b> Miscellaneous registers are valid  <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors  <b>0b10</b> At least one corrected error recorded  <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors  <b>0b1</b> At least one error is not corrected and is deferred  <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison  <b>0b1</b> Uncorrected error recorded because a poison value was consumed  <b>0b0</b> Other cases	W1C	0b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0

Bits	Name	Description	Type	Reset
[7:0]	SERR	<p>Architecturally-defined primary error code.</p> <p><b>0x00</b> No error</p> <p><b>0x01</b> CMN implementation defined error. Refer to cmn_hns_errmisc1.ERRSRC for error type details.</p> <p><b>0x06</b> ECC error on L3 data</p> <p><b>0x07</b> ECC error on L3/SF Tag</p> <p><b>0x0A</b> Producer write data was poisoned but ACE-Lite does not support poisoned data</p> <p><b>0x0D</b> Illegal address</p> <p><b>0x1A</b> Parity error</p>	W1C	0b0

### 8.3.9.28 cmn\_hns\_erraddr

Contains the error record address.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE018

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ras

##### Secure group override

cmn\_hns\_scr.ras

#### Usage constraints

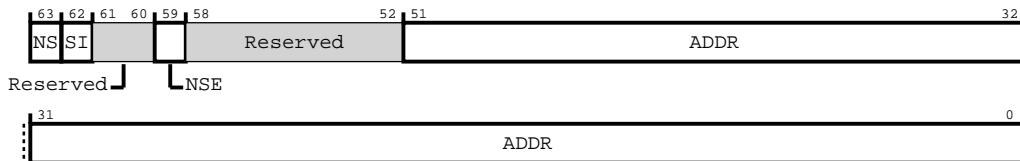
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this

register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-439: cmn\_hns\_erraddr**



**Table 8-445: cmn\_hns\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.9.29 cmn\_hns\_errmisc0

Functions as miscellaneous error register 0. Contains information about corrected errors.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE020

##### 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`

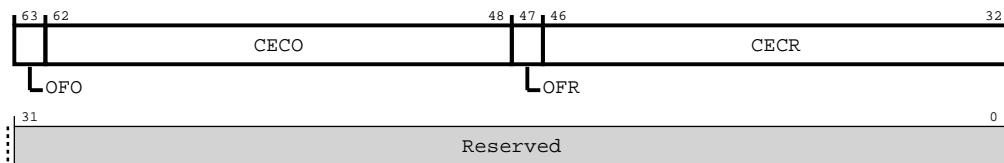
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.ras_secure_access_override` bit and `cmn_hns_rcr.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 8-440: cmn\_hns\_errmisc0**



**Table 8-446: cmn\_hns\_errmisc0 attributes**

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	0b0
[62:48]	CECO	Corrected ECC error count	RW	0b0
[47]	OFR	Corrected error counter overflow	RW	0b0
[46:32]	CECR	Corrected ECC error count	RW	0b0
[31:0]	Reserved	Reserved	RO	

### 8.3.9.30 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

0xE028

### 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

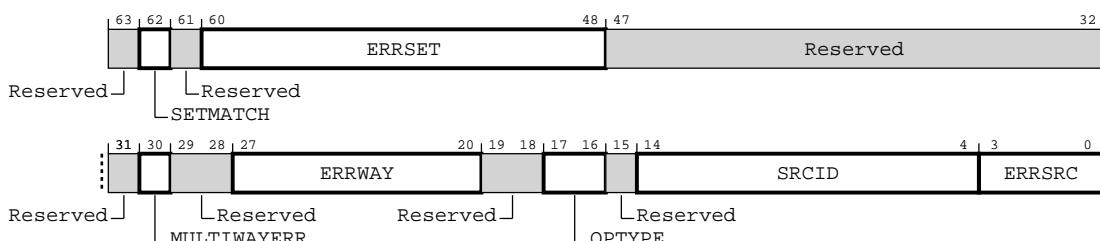
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras\_secure\_access\_override bit and cmn\_hns\_rcr.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 8-441: cmn\_hns\_errmisc1**



**Table 8-447: cmn\_hns\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	SETMATCH	Set address match	RW	0b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	0b0
[47:31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	0b0

Bits	Name	Description	Type	Reset
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	0b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type  <b>0b00</b> Writes, CleanShared, Atomics and stash requests with invalid targets  <b>0b01</b> WriteBack, Evict, and Stash requests with valid target  <b>0b10</b> CMO  <b>0b11</b> Other op types	RW	0b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	0b0
[3:0]	ERRSRC	Error source  <b>0b0001</b> Data single-bit ECC  <b>0b0010</b> Data double-bit ECC  <b>0b0011</b> Single-bit ECC overflow  <b>0b0100</b> Tag single-bit ECC  <b>0b0101</b> Tag double-bit ECC  <b>0b0111</b> SF tag single-bit ECC  <b>0b1000</b> SF tag double-bit ECC  <b>0b1010</b> Data parity error  <b>0b1011</b> Data parity and poison  <b>0b1100</b> NDE	RW	0b0000

### 8.3.9.31 cmn\_hns\_errpfg

Functions as the Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE800

### Type

RO

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.ras

### Secure group override

cmn\_hns\_scr.ras

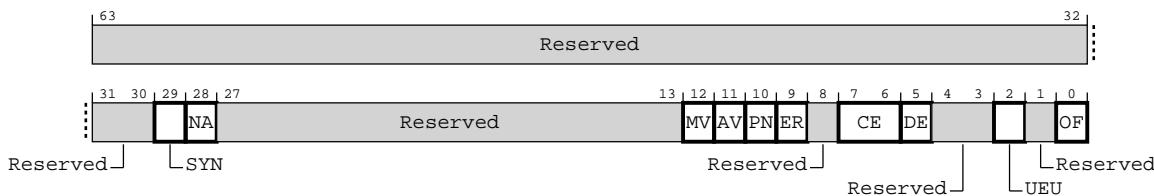
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-442: cmn\_hns\_errpfgf**



**Table 8-448: cmn\_hns\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update ERRSTATUS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	Miscellaneous syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	0b1
[11]	AV	Address syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	0b1
[10]	PN	Poison flag  <b>0b1</b> Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	0b1
[9]	ER	Error reported flag  <b>0b1</b> Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> No Corrected error generation.  <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 0b10; else, update ERRSTATUS.CE to 0b00	RO	0b01
[5]	DE	Deferred error generation.  <b>0b0</b> No Deferred error generation.  <b>0b1</b> Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b1</b> If ERRPFGCTL.UEU == 1, update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01; else, update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b1</b> Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	0b1

### 8.3.9.32 cmn\_hns\_errpfgctl

Functions as the Pseudo-fault Generation Control Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE808

### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.ras

### Secure group override

cmn\_hns\_scr.ras

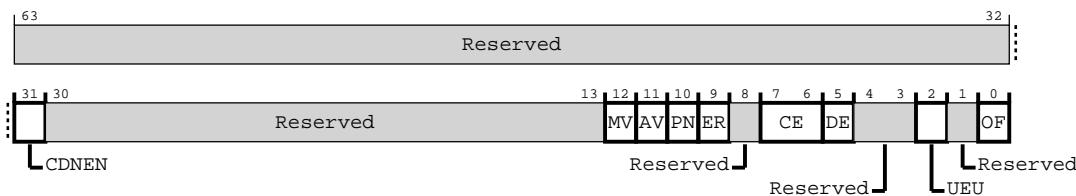
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-443: cmn\_hns\_errpfgctl**



**Table 8-449: cmn\_hns\_errpfgctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	<p>Countdown Enable.</p> <p><b>0b0</b> Countdown disabled.</p> <p><b>0b1</b> Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.</p>	RW	0b0
[30:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	<p>Miscellaneous syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.MV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.MV to 0b1</p>	RW	0b0
[11]	AV	<p>Address syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.AV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.AV to 0b1</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.PN to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.PN to 0b1</p>	RW	0b0
[9]	ER	<p>Error reported flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.ER to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.ER to 0b1</p>	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 0b00</p> <p><b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 0b10</p>	RW	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.DE to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.DE to 0b1</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00</p> <p><b>0b1</b> Fault injection update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01</p>	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	OF	<p>Overflow flag.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.OF to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.OF to 0b1</p>	RW	0b0

### 8.3.9.33 cmn\_hns\_errpfgcdn

Functions as the Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE810

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ras

##### Secure group override

cmn\_hns\_scr.ras

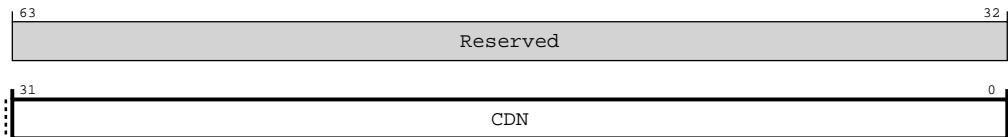
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-444: cmn\_hns\_errpfgcdn**



**Table 8-450: cmn\_hns\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.9.34 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

0xE040

##### Type

RO

##### Reset value

See individual bit resets

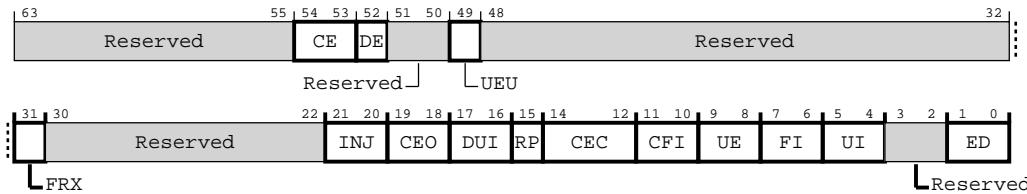
##### Usage constraints

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 8-445: cmn\_hns\_errfr\_NS**



**Table 8-451: cmn\_hns\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording <b>0b00</b> : Corrected Error not supported <b>0b10</b> : Non-specific Corrected Error supported	RO	0b10
[52]	DE	Deferred Error recording <b>0b0</b> : Deferred Error not supported <b>0b1</b> : Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording <b>0b0</b> : Unrecoverable Error not supported <b>0b1</b> : Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. <b>0b1</b> : cmn_hns_errfr_NS[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. <b>0b1</b> : Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite. <b>0b00</b> : Keep the first Corrected Error syndrome	RO	0b00

Bits	Name	Description	Type	Reset
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using cmn_hns_errctlr.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when cmn_hns_errfr_NS.CEC != 0b000.)  <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in cmn_hns_errmisc0	RO	0b1
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter in cmn_hns_errmisc0	RO	0b100
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using cmn_hns_errctlr.CFI.	RO	0b10
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using cmn_hns_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using cmn_hns_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using cmn_hns_errctlr.ED	RO	0b10

### 8.3.9.35 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

0xE048

##### Type

RW

##### Reset value

See individual bit resets

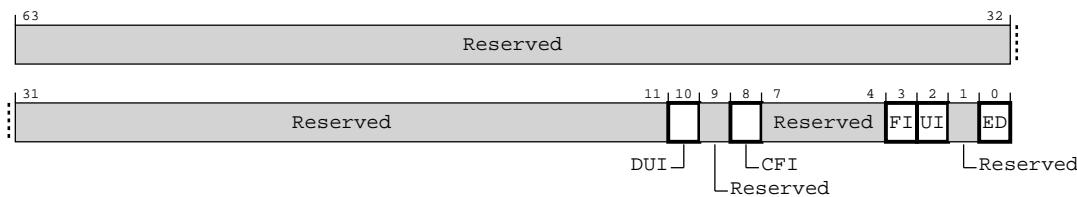
##### Usage constraints

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 8-446: cmn\_hns\_errctlr\_NS**



**Table 8-452: 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 cmn_hns_errfr_NS.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in cmn_hns_errfr_NS.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in cmn_hns_errfr_NS.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in cmn_hns_errfr_NS.UI	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	ED	Enables error detection as specified in cmn_hns_errfr_NS.ED	RW	0b0

### 8.3.9.36 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

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

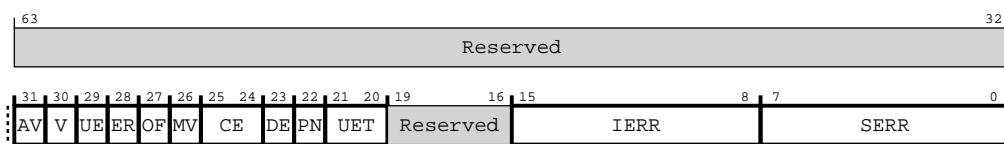
##### Usage constraints

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 8-447: cmn\_hns\_errstatus\_NS**



**Table 8-453: cmn\_hns\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	AV	Address register valid <b>0b1</b> Address is valid; cmn_hns_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	cmn_hns_errmisc<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b1</b> At least one error is not corrected and is deferred <b>0b0</b> No errors deferred	W1C	0b0

Bits	Name	Description	Type	Reset
[22]	PN	Poison <b>0b1</b> Uncorrected error recorded because a poison value was consumed <b>0b0</b> Other cases	W1C	0b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0 <b>0b01</b> Uncorrected error, Unrecoverable error (UEU). <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code. <b>0x00</b> No error <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code. <b>0x00</b> No error <b>0x01</b> CMN implementation defined error. Refer to cmn_hns_errmisc1_NS.ERRSRC for error type details. <b>0x06</b> ECC error on cache data <b>0x07</b> ECC error on L3/SF Tag <b>0x0A</b> Producer write data was poisoned but ACE-Lite does not support poisoned data <b>0x0D</b> Illegal address <b>0x1A</b> Parity error	W1C	0b0

### 8.3.9.37 cmn\_hns\_erraddr\_NS

Contains the non-secure error record address.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE058

### Type

RW

### Reset value

See individual bit resets

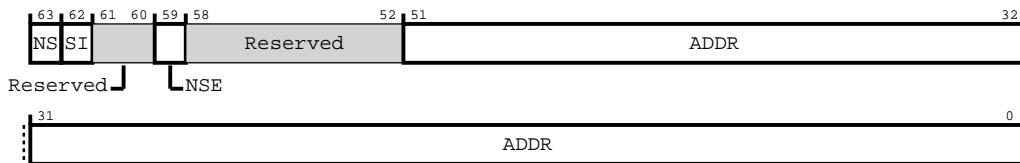
### Usage constraints

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 8-448: cmn\_hns\_erraddr\_NS**



**Table 8-454: cmn\_hns\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.9.38 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

0xE060

##### Type

RW

##### Reset value

See individual bit resets

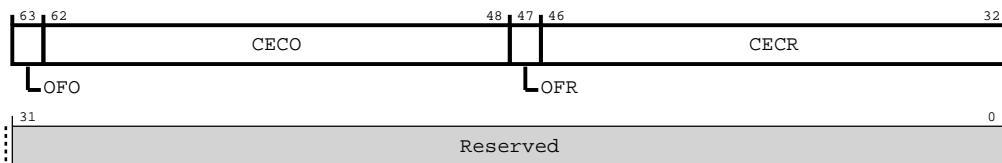
##### Usage constraints

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 8-449: cmn\_hns\_errmisc0\_NS**



**Table 8-455: cmn\_hns\_errmisc0\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	0b0
[62:48]	CECO	Corrected ECC error count	RW	0b0
[47]	OFR	Corrected error counter overflow	RW	0b0
[46:32]	CECR	Corrected ECC error count	RW	0b0
[31:0]	Reserved	Reserved	RO	

### 8.3.9.39 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

0xE068

##### Type

RW

##### Reset value

See individual bit resets

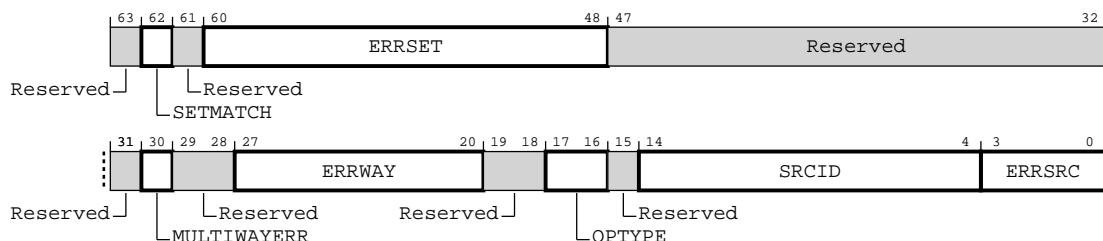
##### Usage constraints

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 8-450: cmn\_hns\_errmisc1\_NS**



**Table 8-456: cmn\_hns\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	SETMATCH	Set address match	RW	0b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	0b0
[47:31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	0b0

Bits	Name	Description	Type	Reset
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	0b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type  <b>0b00</b> Writes, CleanShared, Atomics and stash requests with invalid targets  <b>0b01</b> WriteBack, Evict, and Stash requests with valid target  <b>0b10</b> CMO  <b>0b11</b> Other op types	RW	0b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	0b0
[3:0]	ERRSRC	Error source  <b>0b0001</b> Data single-bit ECC  <b>0b0010</b> Data double-bit ECC  <b>0b0011</b> Single-bit ECC overflow  <b>0b0100</b> Tag single-bit ECC  <b>0b0101</b> Tag double-bit ECC  <b>0b0111</b> SF tag single-bit ECC  <b>0b1000</b> SF tag double-bit ECC  <b>0b1010</b> Data parity error  <b>0b1011</b> Data parity and poison  <b>0b1100</b> NDE	RW	0b0000

### 8.3.9.40 cmn\_hns\_errcfgf\_NS

Functions as the non-secure Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE840

### Type

RO

### Reset value

See individual bit resets

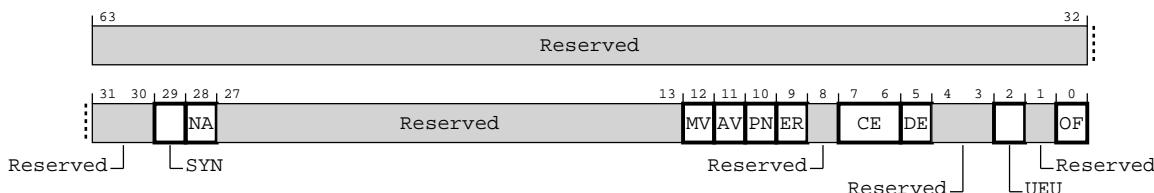
### Usage constraints

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 8-451: cmn\_hns\_errpfgf\_NS**



**Table 8-457: cmn\_hns\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection.  <b>0b1</b> Fault injection does not update ERRSTATUS_NS.SERR	RO	0b1
[28]	NA	No access required.  <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to ERRPFGCTL_NS.MV	RO	0b1
[11]	AV	Address syndrome.  <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to ERRPFGCTL_NS.AV	RO	0b1

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to ERRPFGCTL_NS.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to ERRPFGCTL_NS.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation. <b>0b00</b> No Corrected error generation. <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL_NS.CE == 1, update ERRSTATUS_NS.CE to 0b10; else, update ERRSTATUS_NS.CE to 0b00	RO	0b01
[5]	DE	Deferred error generation. <b>0b0</b> No Deferred error generation. <b>0b1</b> Fault injection update ERRSTATUS_NS.DE to ERRPFGCTL_NS.DE	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation. <b>0b1</b> If ERRPFGCTL_NS.UEU == 1, update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01; else, update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag. <b>0b1</b> Fault injection update ERRSTATUS_NS.OF to ERRPFGCTL_NS.OF	RO	0b1

### 8.3.9.41 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

0xE848

## Type

RW

## Reset value

See individual bit resets

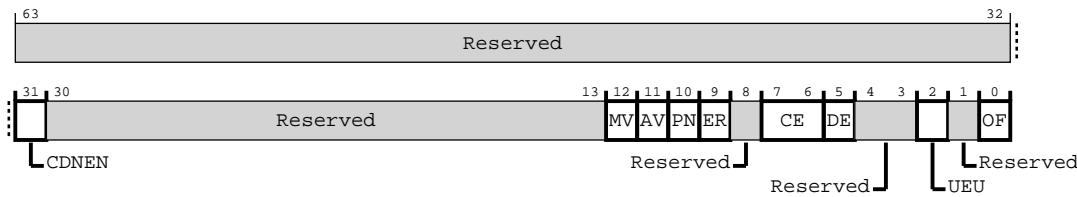
## Usage constraints

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 8-452: cmn\_hns\_errpfgctl\_NS**



**Table 8-458: cmn\_hns\_errpfgctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable. <b>0b0</b> Countdown disabled. <b>0b1</b> Error generation counter is set to ERRPFGCDN_NS.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.MV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to 0b1	RW	0b0
[11]	AV	Address syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.AV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to 0b1	RW	0b0

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag  <b>0b0</b> Fault injection update ERRSTATUS_NS.PN to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to 0b1	RW	0b0
[9]	ER	Error reported flag  <b>0b0</b> Fault injection update ERRSTATUS_NS.ER to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS_NS.CE to 0b00  <b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS_NS.CE to 0b10	RW	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> Fault injection update ERRSTATUS_NS.DE to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.DE to 0b1	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00  <b>0b1</b> Fault injection update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update ERRSTATUS_NS.OF to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.OF to 0b1	RW	0b0

### 8.3.9.42 cmn\_hns\_errfgcdn\_NS

Functions as the non-secure Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE850

### Type

RW

### Reset value

See individual bit resets

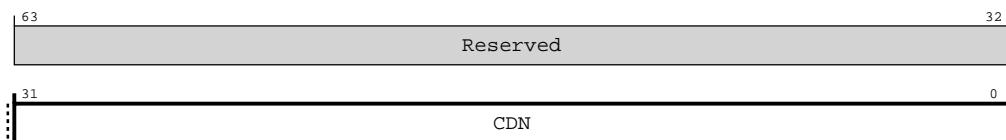
### Usage constraints

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 8-453: cmn\_hns\_errpfcdn\_NS**



**Table 8-459: cmn\_hns\_errpfcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

## 8.3.9.43 cmn\_hns\_errcapctl

Functions as the error Capture Control Register

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xED00

## Type

RW

## Reset value

See individual bit resets

## Root group override

cmn\_hns\_rcr.ras

## Secure group override

cmn\_hns\_scr.ras

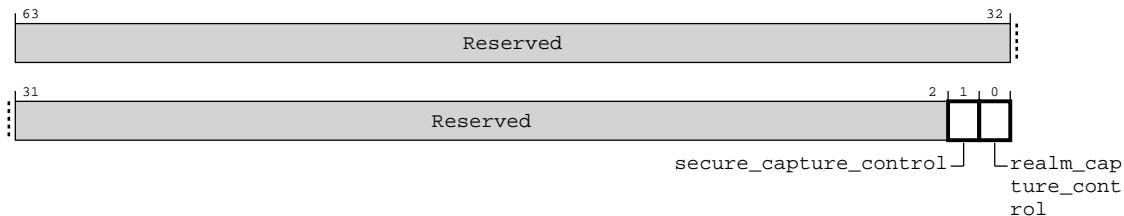
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-454: cmn\_hns\_errcapctl**



**Table 8-460: cmn\_hns\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control <b>0b0</b> Transaction with secure PAS captured in root error record <b>0b1</b> Transaction with secure PAS captured in non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control <b>0b0</b> Transaction with realm PAS captured in root error record <b>0b1</b> Transaction with realm PAS captured in non-secure error record	RW	0b0

### 8.3.9.44 cmn\_hns\_errgsr

Functions as Error Group Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEE00

##### Type

RO

##### Reset value

See individual bit resets

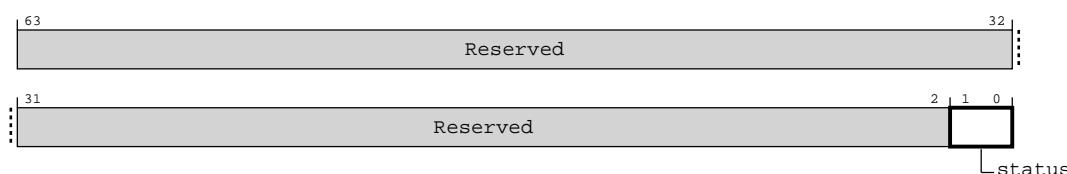
##### Usage constraints

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 8-455: cmn\_hns\_errgsr**



**Table 8-461: 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	0b0

### 8.3.9.45 cmn\_hns\_erriidsr

Functions as the implementation identification register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEE10

### Type

RO

### Reset value

See individual bit resets

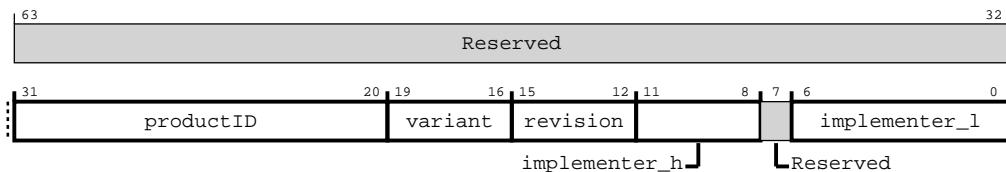
### Usage constraints

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 8-456: cmn\_hns\_errildr**



**Table 8-462: cmn\_hns\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.9.46 cmn\_hns\_errdevaff

Functions as the device affinity register.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEFA8

### Type

RO

### Reset value

See individual bit resets

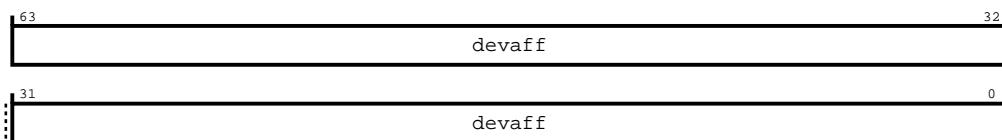
### Usage constraints

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 8-457: cmn\_hns\_errdevaff**



**Table 8-463: cmn\_hns\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

## 8.3.9.47 cmn\_hns\_errdevarch

Functions as the device architecture register.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEF8

## Type

RO

## Reset value

See individual bit resets

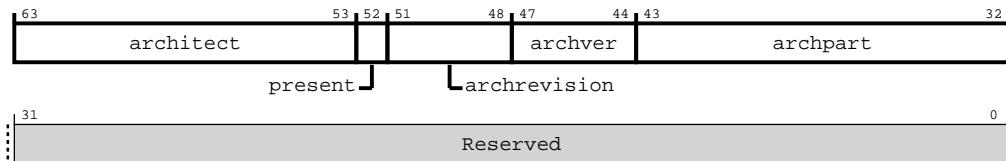
## Usage constraints

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 8-458: cmn\_hns\_errdevarch**



**Table 8-464: cmn\_hns\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

### 8.3.9.48 cmn hns errdevid

Functions as the device configuration register

## Configurations

This register is available in all configurations.

## Attributes

## Width

64

### Address offset

0xEFEC8

## Type

RO

### Reset value

See individual bit resets

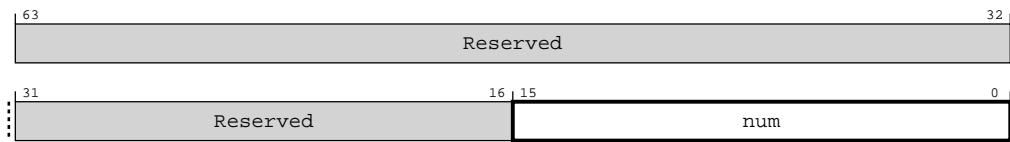
### Usage constraints

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 8-459: cmn\_hns\_errdevid**



**Table 8-465: cmn\_hns\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

### 8.3.9.49 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

0xEFDO

#### Type

RO

#### Reset value

See individual bit resets

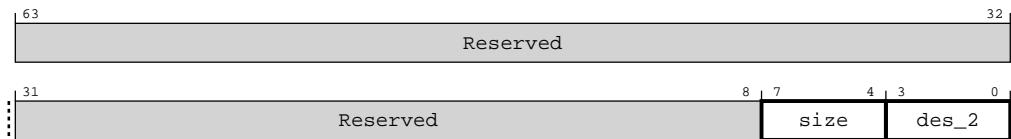
#### Usage constraints

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 8-460: cmn\_hns\_errpidr45**



**Table 8-466: cmn\_hns\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

## 8.3.9.50 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

0xEFE0

#### Type

RO

#### Reset value

See individual bit resets

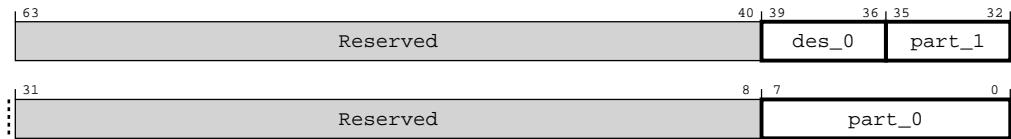
#### Usage constraints

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 8-461: cmn\_hns\_errpidr01**



**Table 8-467: cmn\_hns\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

### 8.3.9.51 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

0xEFE8

##### Type

RO

##### Reset value

See individual bit resets

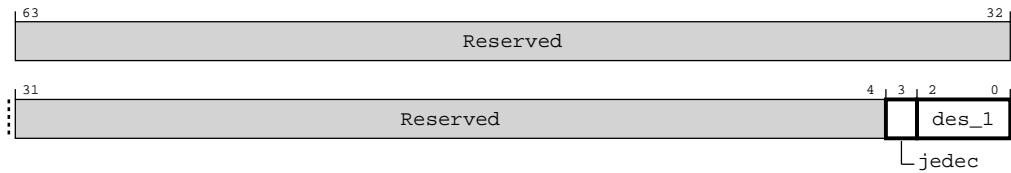
##### Usage constraints

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 8-462: cmn\_hns\_errpidr23**



**Table 8-468: cmn\_hns\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

### 8.3.9.52 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

0xEFF0

##### Type

RO

##### Reset value

See individual bit resets

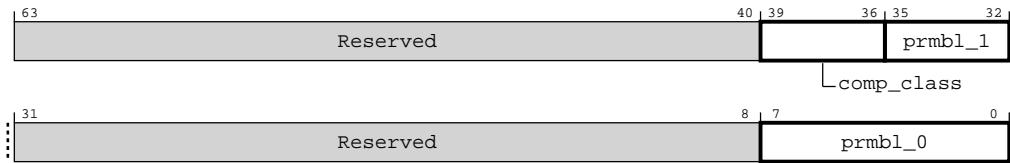
##### Usage constraints

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 8-463: cmn\_hns\_errcidr01**



**Table 8-469: cmn\_hns\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmb1_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_0	PRMBL_0	RO	0xD

### 8.3.9.53 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

0xEFF8

##### Type

RO

##### Reset value

See individual bit resets

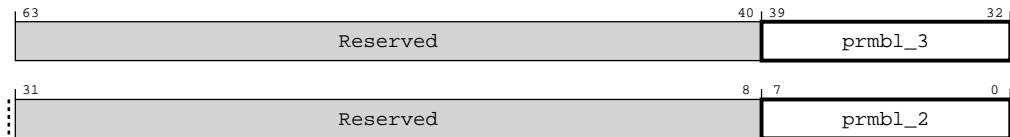
#### Usage constraints

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 8-464: cmn\_hns\_errcidr23**



**Table 8-470: cmn\_hns\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.9.54 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

0xE030

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ras

##### Secure group override

cmn\_hns\_scr.ras

#### Usage constraints

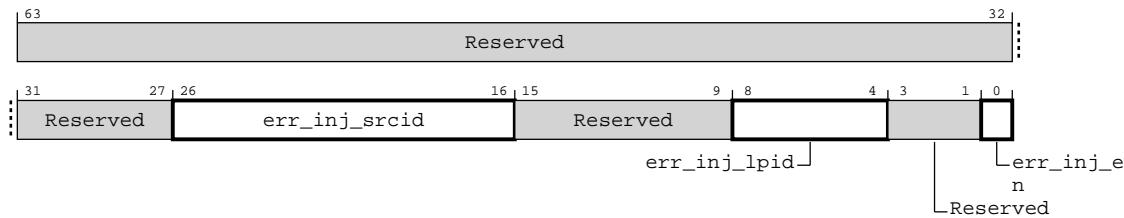
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ras bit is set, Secure accesses to this

register are permitted. If both the cmn\_hns\_scr.ras bit and cmn\_hns\_rcr.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 8-465: cmn\_hns\_err\_inj**



**Table 8-471: 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	0x0
[15:9]	Reserved	Reserved	RO	
[8:4]	err_inj_lpid	LPID used to match for error injection	RW	0x0
[3:1]	Reserved	Reserved	RO	
[0]	err_inj_en	Enables error injection and report	RW	0b0

### 8.3.9.55 cmn\_hns\_byte\_par\_err\_inj

Functions as the byte parity error injection register for HN-F.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE938

### Type

WO

### Reset value

See individual bit resets

### Root group override

`cmn_hns_rcr.ras`

### Secure group override

`cmn_hns_scr.ras`

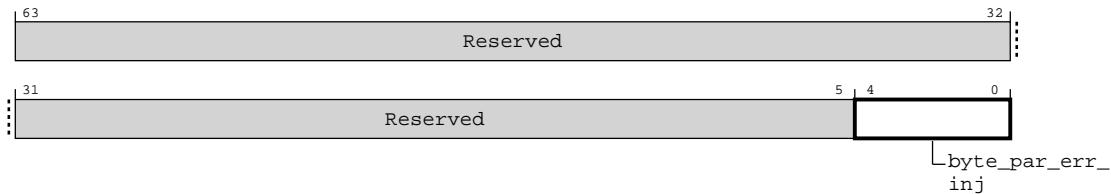
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.ras` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.ras` bit and `cmn_hns_rcr.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 8-466: cmn\_hns\_byte\_par\_err\_inj**



**Table 8-472: 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	0x0

### 8.3.9.56 cmn\_hns\_slc\_lock\_ways

Controls SLC way lock settings.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xC00

#### 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`

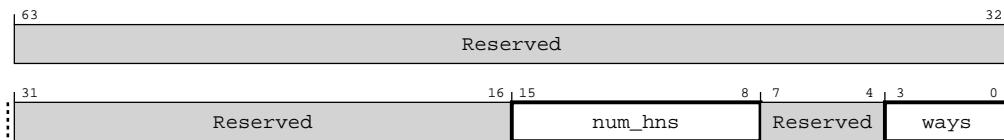
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-467: `cmn_hns_slc_lock_ways`**



**Table 8-473: `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	0b0

### 8.3.9.57 `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

0xC08

### 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

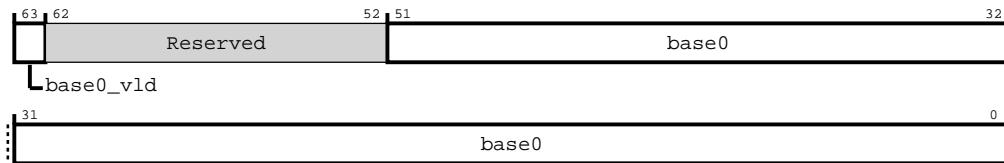
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-468: cmn\_hns\_slc\_lock\_base0**



**Table 8-474: cmn\_hns\_slc\_lock\_base0 attributes**

Bits	Name	Description	Type	Reset
[63]	base0_vld	Lock region 0 base valid	RW	0b0
[62:52]	Reserved	Reserved	RO	
[51:0]	base0	Lock region 0 base address	RW	0b0

### 8.3.9.58 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

0xC10

### 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

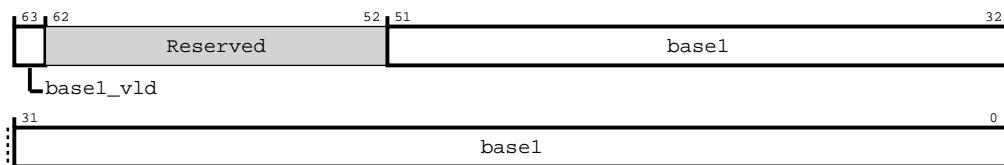
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-469: cmn\_hns\_slc\_lock\_base1**



**Table 8-475: cmn\_hns\_slc\_lock\_base1 attributes**

Bits	Name	Description	Type	Reset
[63]	base1_vld	Lock region 1 base valid	RW	0b0
[62:52]	Reserved	Reserved	RO	
[51:0]	base1	Lock region 1 base address	RW	0b0

### 8.3.9.59 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

0xC18

##### 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

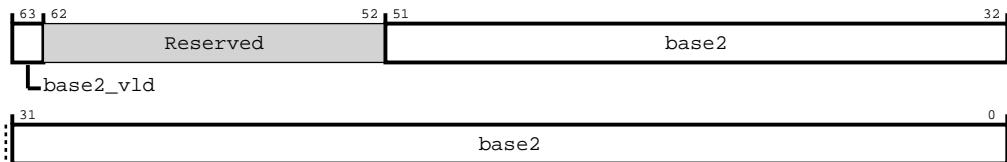
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-470: cmn\_hns\_slc\_lock\_base2**



**Table 8-476: cmn\_hns\_slc\_lock\_base2 attributes**

Bits	Name	Description	Type	Reset
[63]	base2_vld	Lock region 2 base valid	RW	0b0

Bits	Name	Description	Type	Reset
[62:52]	Reserved	Reserved	RO	
[51:0]	base2	Lock region 2 base address	RW	0b0

### 8.3.9.60 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

0xC20

##### 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

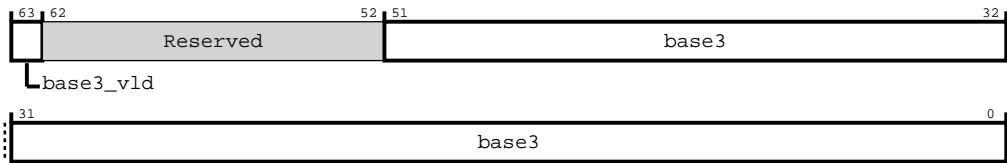
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-471: cmn\_hns\_slc\_lock\_base3**



**Table 8-477: cmn\_hns\_slc\_lock\_base3 attributes**

Bits	Name	Description	Type	Reset
[63]	base3_vld	Lock region 3 base valid	RW	0b0
[62:52]	Reserved	Reserved	RO	
[51:0]	base3	Lock region 3 base address	RW	0b0

### 8.3.9.61 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

0xC28

##### 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

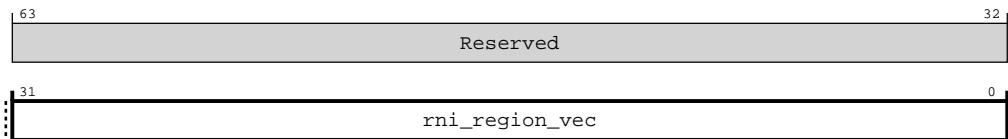
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-472: cmn\_hns\_rni\_region\_vec**



**Table 8-478: 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  <b>NOTE</b> Must be set to 0b0 if range-based region locking or OCM is enabled.	RW	0b0

### 8.3.9.62 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

0xC30

### 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

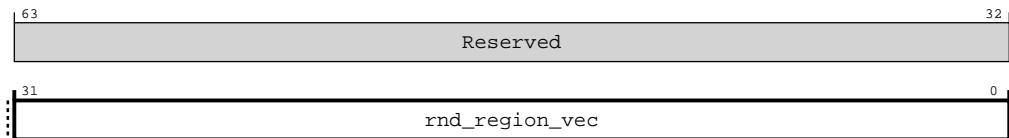
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and

`cmn_hns_rcr.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 8-473: cmn\_hns\_rnd\_region\_vec**



**Table 8-479: 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  <b>NOTE</b> Must be set to 0b0 if range-based region locking or OCM is enabled.	RW	0b0

## 8.3.9.63 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

0xC38

#### 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`

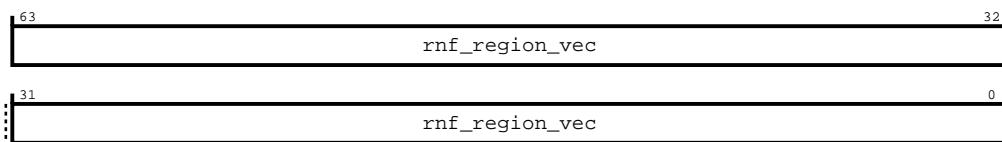
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-474: cmn\_hns\_rnf\_region\_vec**



**Table 8-480: 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  <b>NOTE</b> Must be 0b0 if range-based region locking or OCM is enabled.	RW	0b0

### 8.3.9.64 cmn\_hns\_rnf\_region\_vec1

Functions as the control register for RN-F source SLC way allocation for logical IDs 64 through 127.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC40

##### 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`

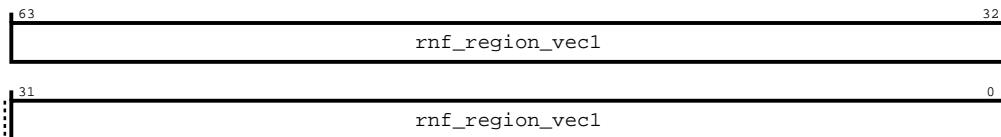
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-475: cmn\_hns\_rnf\_region\_vec1**



**Table 8-481: cmn\_hns\_rnf\_region\_vec1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>rnf_region_vec1</code>	Bit vector mask; identifies which logical IDs of the RN-Fs to allocate to the locked region  <b>NOTE</b> Must be 0b0 if range-based region locking or OCM is enabled.	RW	0b0

### 8.3.9.65 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

0xC48

#### 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`

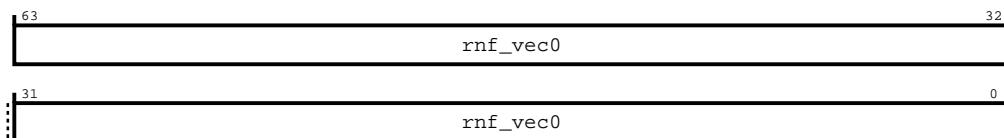
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-476: cmn\_hns\_slcway\_partition0\_rnf\_vec**



**Table 8-482: cmn\_hns\_slcway\_partition0\_rnf\_vec attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>rnf_vec0</code>	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	0xFFFFFFFFFFFFFFF

### 8.3.9.66 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

0xC50

#### 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`

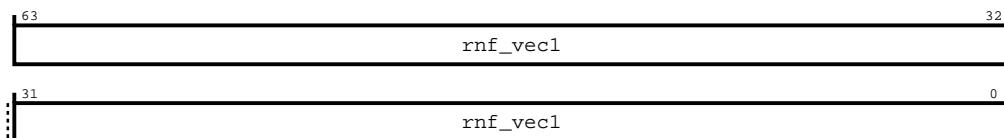
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-477: cmn\_hns\_slcway\_partition1\_rnf\_vec**



**Table 8-483: cmn\_hns\_slcway\_partition1\_rnf\_vec attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>rnf_vec1</code>	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	0xFFFFFFFFFFFFFFF

### 8.3.9.67 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

0xC58

#### 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`

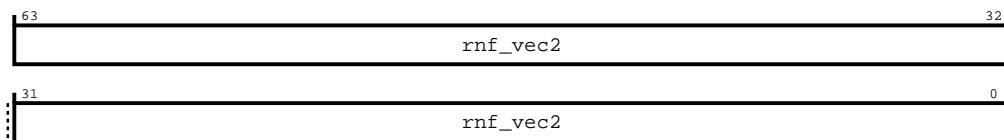
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-478: cmn\_hns\_slcway\_partition2\_rnf\_vec**



**Table 8-484: cmn\_hns\_slcway\_partition2\_rnf\_vec attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>rnf_vec2</code>	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	0xFFFFFFFFFFFFFFF

### 8.3.9.68 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

0xC60

### 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`

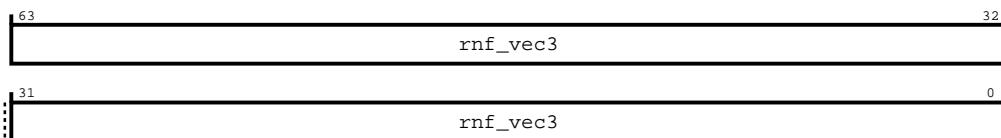
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.slc_lock_ways` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.slc_lock_ways` bit and `cmn_hns_rcr.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 8-479: cmn\_hns\_slcway\_partition3\_rnf\_vec**



**Table 8-485: 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	0xFFFFFFFFFFFFFFFFF

### 8.3.9.69 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

0xCB0

### 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

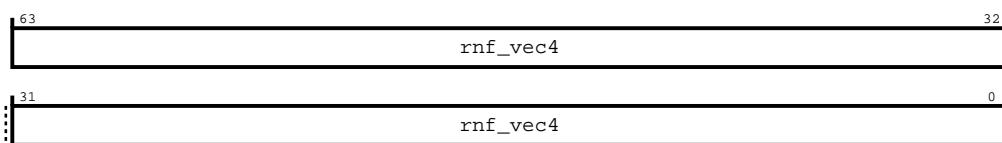
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-480: cmn\_hns\_slcway\_partition0\_rnf\_vec1**



**Table 8-486: 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	0xFFFFFFFFFFFFFFFFF

### 8.3.9.70 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xCB8

### 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

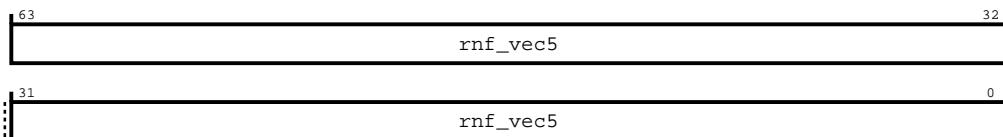
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-481: cmn\_hns\_slcway\_partition1\_rnf\_vec1**



**Table 8-487: 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	0xFFFFFFFFFFFFFFFFF

### 8.3.9.71 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xCC0

##### 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

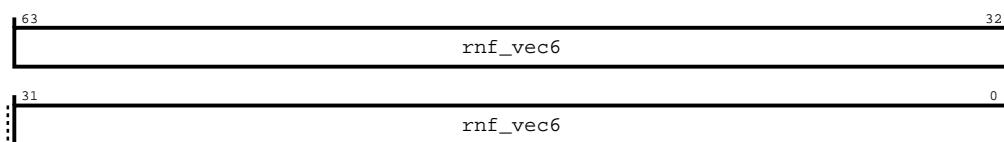
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-482: cmn\_hns\_slcway\_partition2\_rnf\_vec1**



**Table 8-488: 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	0xFFFFFFFFFFFFFFFFF

### 8.3.9.72 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 RN-F IDs 64 to 127.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xCC8

##### 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

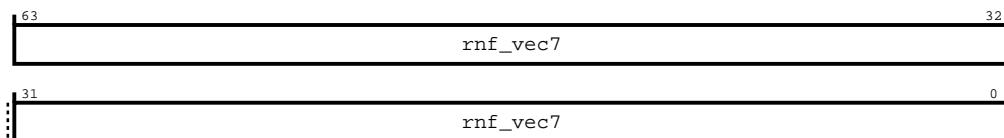
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-483: cmn\_hns\_slcway\_partition3\_rnf\_vec1**



**Table 8-489: cmn\_hns\_slcway\_partition3\_rnf\_vec1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec7	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	0xFFFFFFFFFFFFFFFFF

### 8.3.9.73 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

0xC68

##### 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

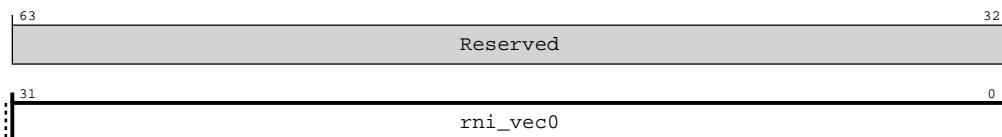
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-484: cmn\_hns\_slcway\_partition0\_rni\_vec**



**Table 8-490: 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	0xFFFFFFFF

### 8.3.9.74 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

0xC70

##### 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

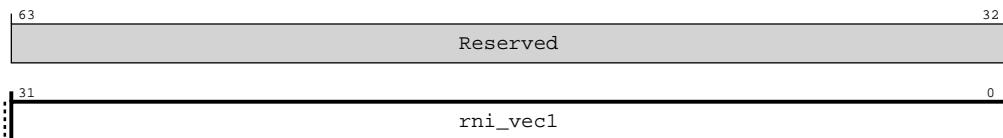
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-485: cmn\_hns\_slcway\_partition1\_rni\_vec**



**Table 8-491: 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	0xFFFFFFFF

### 8.3.9.75 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

0xC78

##### 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

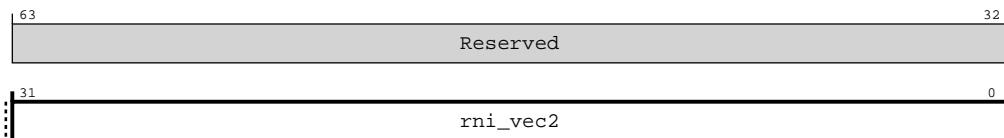
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-486: cmn\_hns\_slcway\_partition2\_rni\_vec**



**Table 8-492: 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	0xFFFFFFFF

### 8.3.9.76 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

0xC80

##### 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

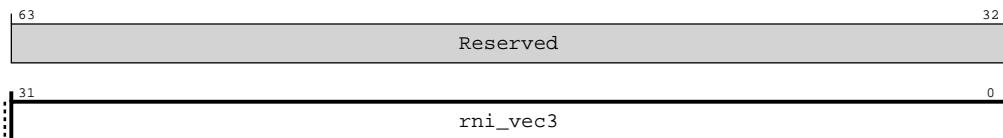
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-487: cmn\_hns\_slcway\_partition3\_rni\_vec**



**Table 8-493: 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	0xFFFFFFFF

### 8.3.9.77 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

0xC88

##### 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

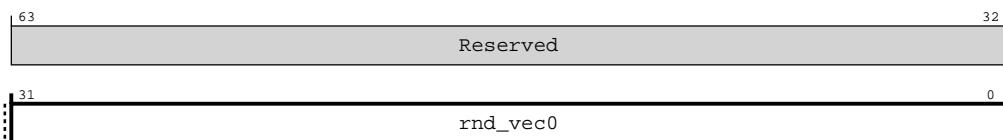
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-488: cmn\_hns\_slcway\_partition0\_rnd\_vec**



**Table 8-494: 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	0xFFFFFFFF

### 8.3.9.78 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

0xC90

##### 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

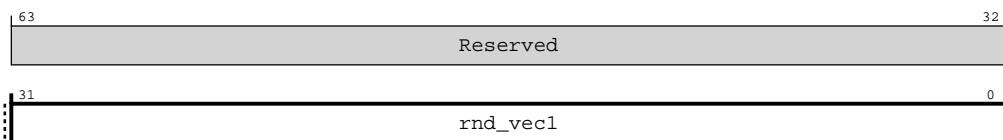
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-489: cmn\_hns\_slcway\_partition1\_rnd\_vec**



**Table 8-495: 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	0xFFFFFFFF

### 8.3.9.79 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

0xC98

##### 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

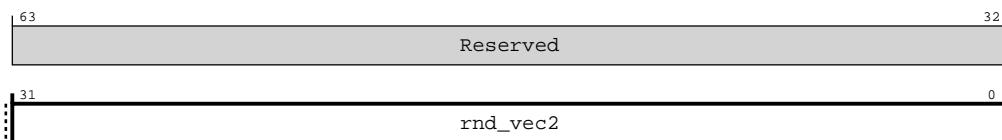
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-490: cmn\_hns\_slcway\_partition2\_rnd\_vec**



**Table 8-496: 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	0xFFFFFFFF

### 8.3.9.80 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

0xCA0

##### 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

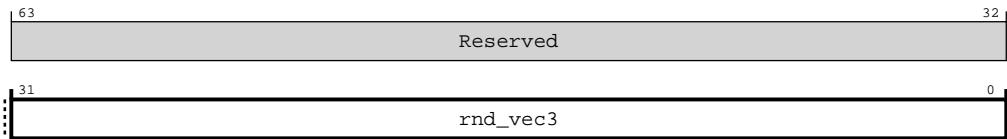
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-491: cmn\_hns\_slcway\_partition3\_rnd\_vec**



**Table 8-497: 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	0xFFFFFFFF

### 8.3.9.81 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

0xCA8

##### 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

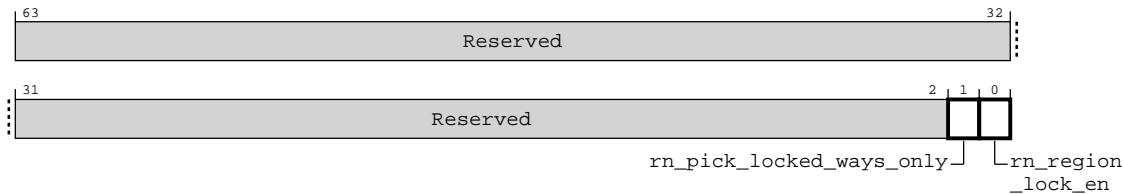
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slc\_lock\_ways bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slc\_lock\_ways bit and cmn\_hns\_rcr.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 8-492: cmn\_hns\_rn\_region\_lock**



**Table 8-498: 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 <b>0b0</b> Programmed RN will choose all ways including locked <b>0b1</b> Programmed RN will only allocate in locked ways	RW	0b0
[0]	rn_region_lock_en	Enables SRC-based region locking <b>0b0</b> SRC based way locking is disabled <b>0b1</b> SRC based way locking is enabled	RW	0b0

### 8.3.9.82 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

0xCD0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

## Secure group override

cmn\_hns\_scr.sam\_control

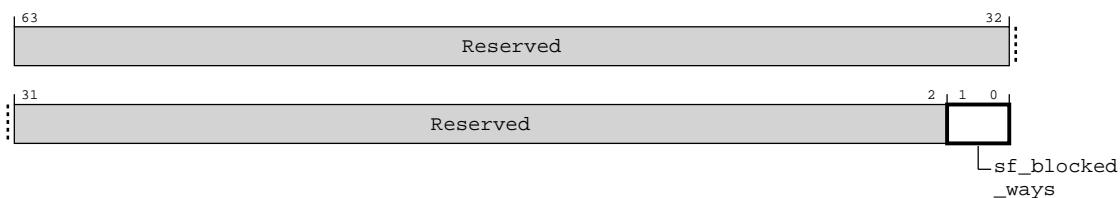
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-493: cmn\_hns\_sf\_cxg\_blocked\_ways**



**Table 8-499: cmn\_hns\_sf\_cxg\_blocked\_ways attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	sf_blocked_ways	<p>Number of SF ways blocked from remote chips to be able to use in CML mode.</p> <p><b>0b00</b> No ways are blocked; all SF ways could be used by local or remote RN-Fs</p> <p><b>0b01</b> SF_NUM_WAYS = 16 ways 3:0 for local RN-Fs only; ways 15:4 for local and remote RN-Fs SF_NUM_WAYS &gt; 16 ways 7:0 for local RN-Fs only; ways 31:8 for local and remote RN-Fs</p> <p><b>0b10</b> SF_NUM_WAYS = 16 ways 7:0 for local RN-Fs only; ways 15:8 for local and remote RN-Fs SF_NUM_WAYS &gt; 16 ways 15:0 for local RN-Fs only; ways 31:16 for local and remote RN-Fs</p> <p><b>0b11</b> SF_NUM_WAYS &lt; 26 ways 11:0 for local RN-Fs only; ways 15:12 for local and remote RN-Fs SF_NUM_WAYS &gt;= 26 ways 23:0 for local RN-Fs only; ways 31:24 for local and remote RN-Fs</p>	RW	0b00

### 8.3.9.83 cmn\_hns\_cxg\_ha\_metadata\_exclusion\_list

Functions as the control register to identify CXG HA which does not support metadata

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xCE0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

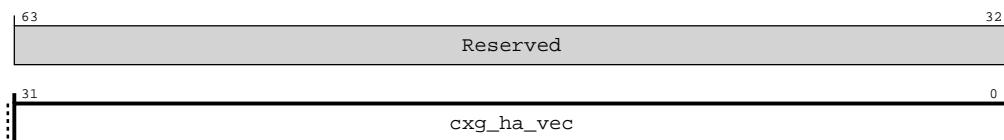
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-494: cmn\_hns\_cxg\_ha\_metadata\_exclusion\_list**



**Table 8-500: cmn\_hns\_cxg\_ha\_metadata\_exclusion\_list attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31:0]	cxg_ha_vec	Bit vector mask; identifies which logical IDs of the CXG HA does not support metadata	RW	0x00000000

### 8.3.9.84 cmn\_hns\_cxg\_ha\_snp\_throttle\_disable\_list

Functions as the control register to identify CXG HA which do not need to throttle snoops

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD60

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

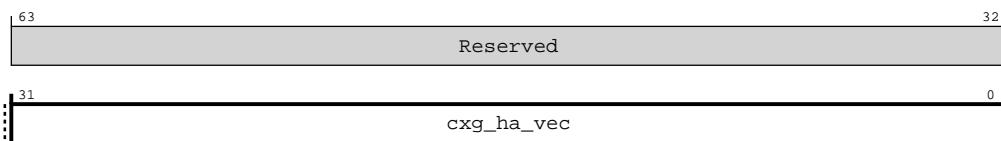
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-495: cmn\_hns\_cxg\_ha\_snp\_throttle\_disable\_list**



**Table 8-501: cmn\_hns\_cxg\_ha\_snp\_throttle\_disable\_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 do not require snoops to be throttled	RW	0x00000000

### 8.3.9.85 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

0xCD8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

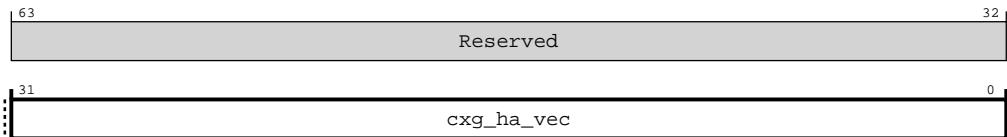
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-496: cmn\_hns\_cxg\_ha\_smp\_exclusion\_list**



**Table 8-502: 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	0x00000000

### 8.3.9.86 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

0xCFO

##### Type

RW

##### Reset value

See individual bit resets

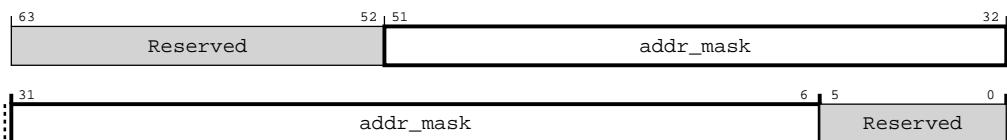
##### Usage constraints

This register is owned in the Root space and is 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 8-497: hn\_sam\_hash\_addr\_mask\_reg**



**Table 8-503: 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	0xFFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

### 8.3.9.87 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

0xCF8

##### Type

RW

##### Reset value

See individual bit resets

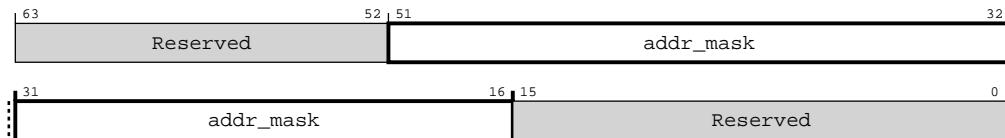
##### Usage constraints

This register is owned in the Root space and is 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 8-498: hn\_sam\_region\_cmp\_addr\_mask\_reg**



**Table 8-504: 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	0xFFFFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

### 8.3.9.88 cmn\_hns\_sam\_cfg1\_def\_hashed\_region

Configures default hashed region in HN-F SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD48

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

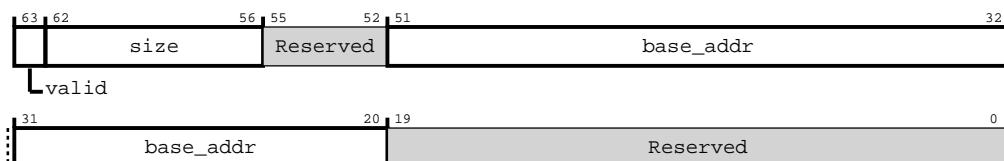
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-499: cmn\_hns\_sam\_cfg1\_def\_hashed\_region**



**Table 8-505: cmn\_hns\_sam\_cfg1\_def\_hashed\_region attributes**

Bits	Name	Description	Type	Reset
[63]	valid	Default hashed region valid  <b>0b0</b> not valid  <b>0b1</b> valid for memory region comparison	RW	0x1
[62:56]	size	Default hashed region 0 size  <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0x7F
[55:52]	Reserved	Reserved	RO	
[51:20]	base_addr	Bits [51:20] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[19:0]	Reserved	Reserved	RO	

### 8.3.9.89 cmn\_hns\_sam\_cfg2\_def\_hashed\_region

Configures default hashed region in HN-F SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD50

##### 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

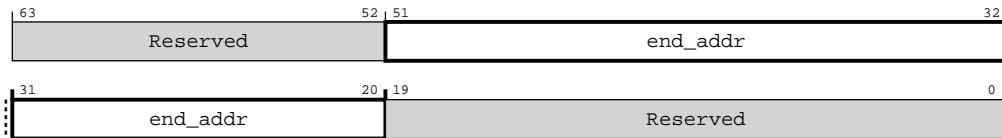
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and

cmn\_hns\_rcr.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 8-500: cmn\_hns\_sam\_cfg2\_def\_hashed\_region**



**Table 8-506: cmn\_hns\_sam\_cfg2\_def\_hashed\_region attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	Bits [51:20] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0xFFFFFFFF
[19:0]	Reserved	Reserved	RO	-

## 8.3.9.90 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xD00

#### Type

RW

#### Reset value

See individual bit resets

#### Root group override

cmn\_hns\_rcr.sam\_control

## Secure group override

cmn hns scr.sam control

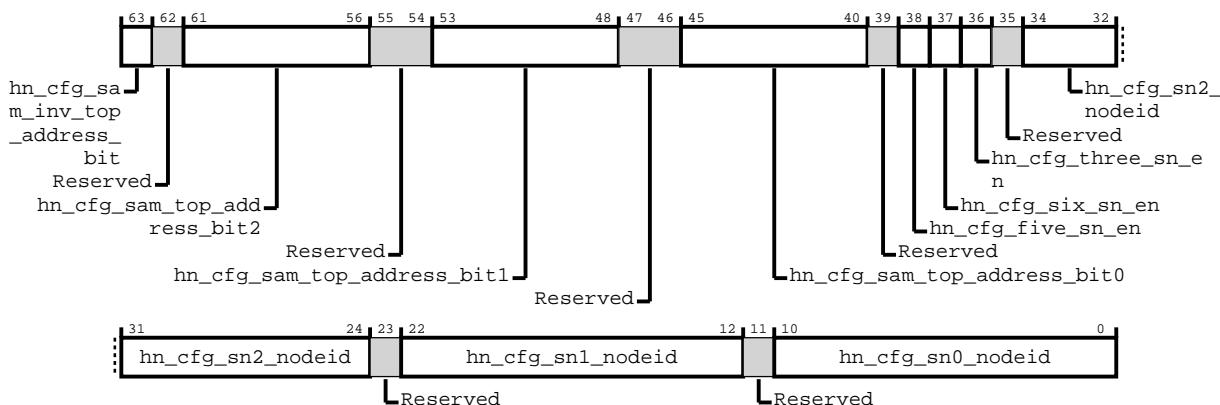
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-501: cmn\_hns\_sam\_control**



**Table 8-507: 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)  <b>NOTE</b> Can only be used when the address map does not have unique address bit combinations.	RW	0x0
[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	0x00
[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	0x00
[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	0x00
[39]	Reserved	Reserved	RO	
[38]	hn_cfg_five_sn_en	Enables 5-SN configuration	RW	0b0

Bits	Name	Description	Type	Reset
[37]	hn_cfg_six_sn_en	Enables 6-SN configuration	RW	0b0
[36]	hn_cfg_three_sn_en	Enables 3-SN configuration	RW	0b0
[35]	Reserved	Reserved	RO	
[34:24]	hn_cfg_sn2_nodeid	SN 2 node ID	RW	0x0
[23]	Reserved	Reserved	RO	
[22:12]	hn_cfg_sn1_nodeid	SN 1 node ID	RW	0x0
[11]	Reserved	Reserved	RO	
[10:0]	hn_cfg_sn0_nodeid	SN 0 node ID	RW	0x0

### 8.3.9.91 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD28

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

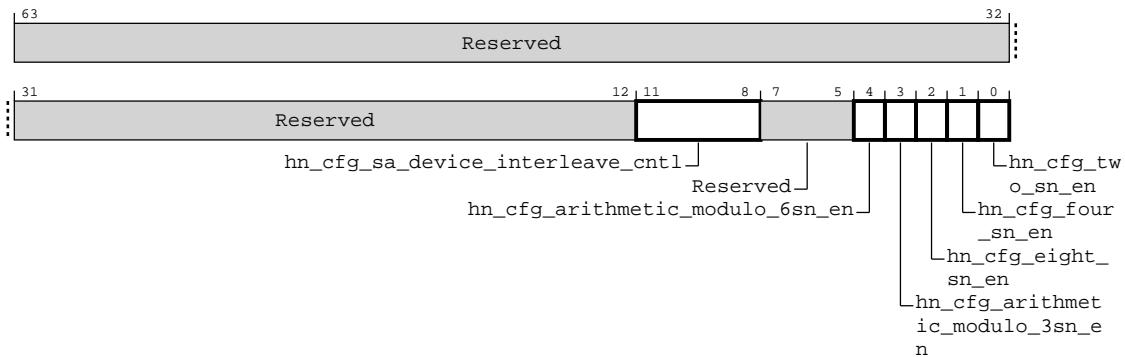
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-502: cmn\_hns\_sam\_control2**



**Table 8-508: cmn\_hns\_sam\_control2 attributes**

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	
[11:8]	hn_cfg_sa_device_interleave_cntl	controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits	RW	0b0
0x0		64B Interleaved (CXSA), [52:6] - arithmetic modulo		
0x1		128B Interleaved (CXSA), [52:7] - arithmetic modulo		
0x2		256B Interleaved (CXSA), [52:8] - arithmetic modulo		
0x3		512B Interleaved (CXSA), [52:9] - arithmetic modulo		
0x4		1KB Interleaved (CXSA), [52:10] - arithmetic modulo		
0x5		2KB Interleaved (CXSA), [52:11] - arithmetic modulo		
0x6		4KB Interleaved (CXSA), [52:12] - arithmetic modulo		
0x7		8KB Interleaved (CXSA), [52:13] - arithmetic modulo		
0x8		16KB Interleaved (CXSA), [52:14] - arithmetic modulo		
0x9		Reserved (CXSA), [52:15] - arithmetic modulo		
Others		Reserved		
[7:5]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[4]	hn_cfg_arithmetic_modulo_6sn_en	Enables 6-SN configuration (Arithmetic Modulo)	RW	0b0
[3]	hn_cfg_arithmetic_modulo_3sn_en	Enables 3-SN configuration (Arithmetic Modulo)	RW	0b0
[2]	hn_cfg_eight_sn_en	Enables 8-SN configuration	RW	0b0
[1]	hn_cfg_four_sn_en	Enables 4-SN configuration	RW	0b0
[0]	hn_cfg_two_sn_en	Enables 2-SN configuration	RW	0b0

### 8.3.9.92 cmn\_hns\_sam\_memregion0

Configures range-based memory region 0 in HN-F SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD08

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

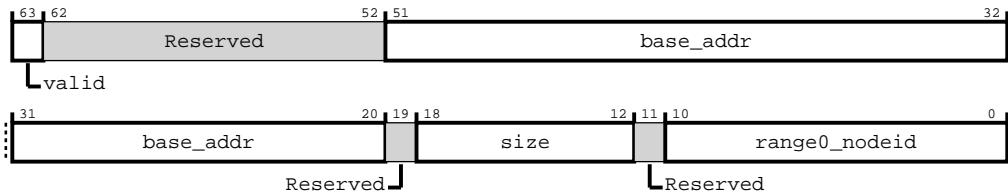
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-503: cmn\_hns\_sam\_memregion0**



**Table 8-509: cmn\_hns\_sam\_memregion0 attributes**

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 0 valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0x0
[62:52]	Reserved	Reserved	RO	
[51:20]	base_addr	Base address of memory region 0 <b>CONSTRAINT</b> Must be an integer multiple of region size.	RW	0x0
[19]	Reserved	Reserved	RO	
[18:12]	size	Memory region 0 size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0x0
[11]	Reserved	Reserved	RO	
[10:0]	range0_nodeid	Memory region 0 target node ID	RW	0x0

### 8.3.9.93 cmn\_hns\_sam\_memregion0\_end\_addr

Configures end address memory region 0 in HN-F SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD38

## Type

RW

## Reset value

See individual bit resets

## Root group override

cmn\_hns\_rcr.sam\_control

## Secure group override

cmn\_hns\_scr.sam\_control

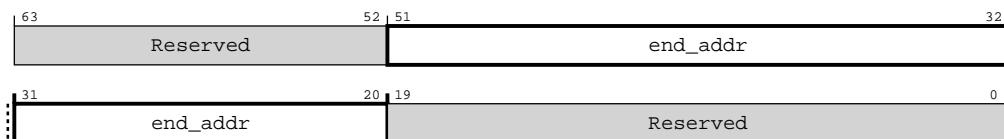
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-504: cmn\_hns\_sam\_memregion0\_end\_addr**



**Table 8-510: 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	0x0
[19:0]	Reserved	Reserved	RO	

## 8.3.9.94 cmn\_hns\_sam\_memregion1

Configures range-based memory region 1 in HN-F SAM.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

## Width

64

## Address offset

0xD10

## 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 is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-505: cmn\_hns\_sam\_memregion1**

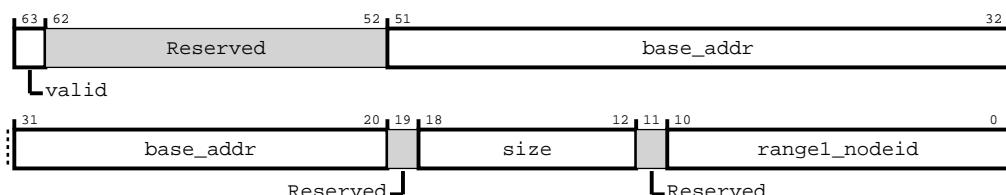


Table 8-511: cnp hns sam memregion1 attributes

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 1 valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0x0
[62:52]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[51:20]	base_addr	Base address of memory region 1  <b>CONSTRAINT</b> Must be an integer multiple of region size.	RW	0x0
[19]	Reserved	Reserved	RO	
[18:12]	size	Memory region 1 size  <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0x0
[11]	Reserved	Reserved	RO	
[10:0]	range1_nodeid	Memory region 1 target node ID	RW	0x0

### 8.3.9.95 cmn\_hns\_sam\_memregion1\_end\_addr

Configures end address memory region 1 in HN-F SAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD40

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

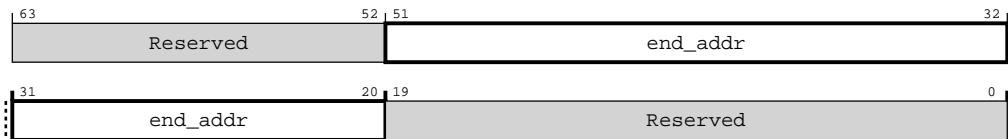
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-506: cmn\_hns\_sam\_memregion1\_end\_addr**



**Table 8-512: cmn\_hns\_sam\_memregion1\_end\_addr attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:20]	end_addr	End address of memory region 1	RW	0x0
[19:0]	Reserved	Reserved	RO	

## 8.3.9.96 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

0xD18

#### 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

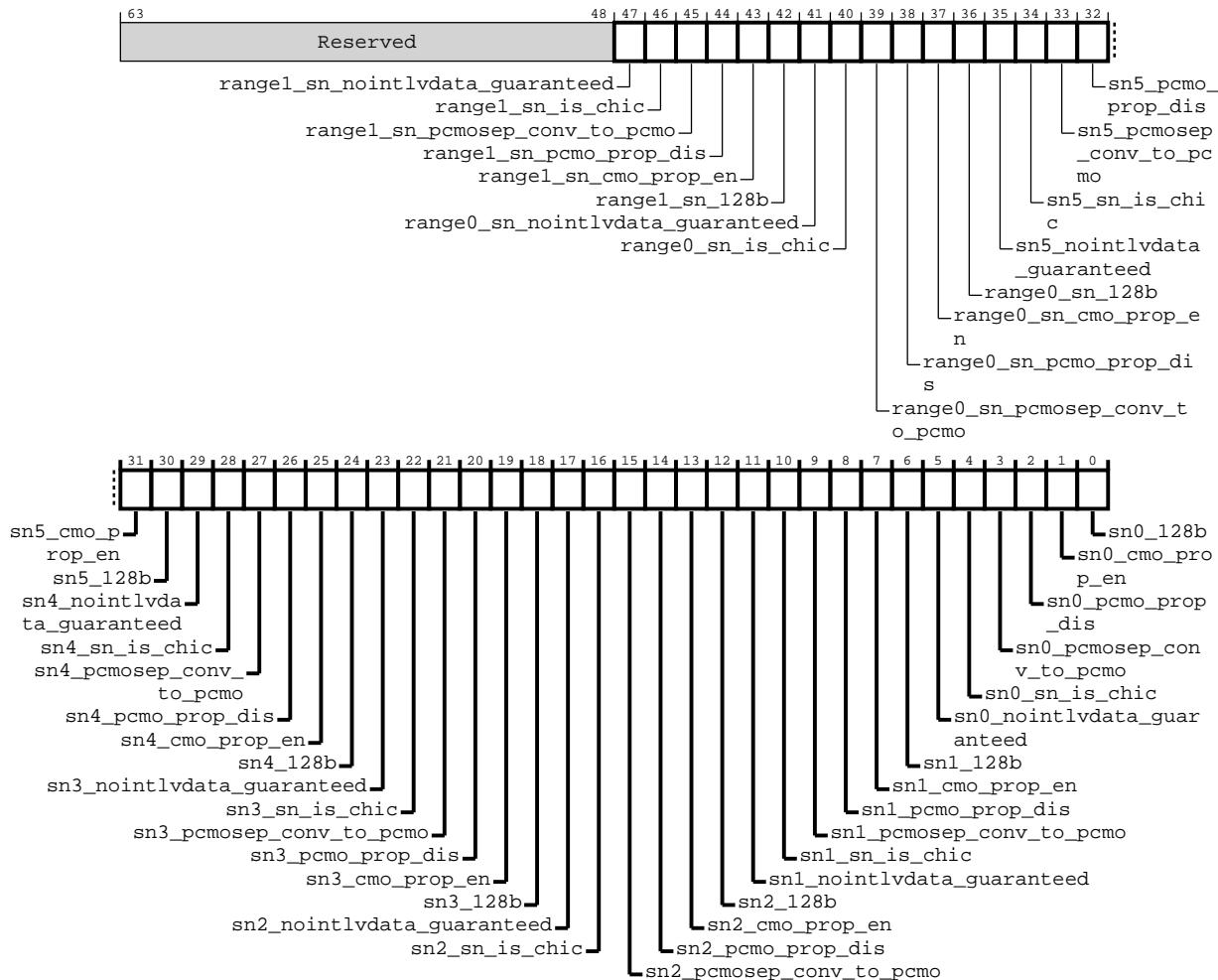
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.sam\_control bit 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 8-507: cmn\_hns\_sam\_sn\_properties**



**Table 8-513: 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	0b0
[46]	range1_sn_is_chic	Indicates that the range 1 SN is a CHI-C SN when set	RW	0b0
[45]	range1_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 1 SN when set <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0

Bits	Name	Description	Type	Reset
[44]	range1_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 1 SN when set	RW	0b0
[43]	range1_sn_cmo_prop_en	Enables CMO propagation for range 1 SN	RW	0b0
[42]	range1_sn_128b	Data width of range 1 SN  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[41]	range0_sn_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[40]	range0_sn_is_chic	Indicates that the range 0 SN is a CHI-C SN when set	RW	0b0
[39]	range0_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 0 SN when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[38]	range0_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 0 SN when set	RW	0b0
[37]	range0_sn_cmo_prop_en	Enables CMO propagation for range 0 SN	RW	0b0
[36]	range0_sn_128b	Data width of range 0 SN  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[35]	sn5_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[34]	sn5_sn_is_chic	Indicates that SN5 is a CHI-C SN when set	RW	0b0
[33]	sn5_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 5 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[32]	sn5_pcmo_prop_dis	Disables PCMO propagation for SN 5 when set	RW	0b0
[31]	sn5_cmo_prop_en	Enables CMO propagation for SN 5 when set	RW	0b0
[30]	sn5_128b	Data width of SN 5  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[29]	sn4_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[28]	sn4_sn_is_chic	Indicates that SN4 is a CHI-C SN when set	RW	0b0
[27]	sn4_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 4 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[26]	sn4_pcmo_prop_dis	Disables PCMO propagation for SN 4 when set	RW	0b0
[25]	sn4_cmo_prop_en	Enables CMO propagation for SN 4 when set	RW	0b0

Bits	Name	Description	Type	Reset
[24]	sn4_128b	Data width of SN 4  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[23]	sn3_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[22]	sn3_sn_is_chic	Indicates that SN3 is a CHI-C SN when set	RW	0b0
[21]	sn3_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 3 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[20]	sn3_pcmo_prop_dis	Disables PCMO propagation for SN 3 when set	RW	0b0
[19]	sn3_cmo_prop_en	Enables CMO propagation for SN 3 when set	RW	0b0
[18]	sn3_128b	Data width of SN 3  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[17]	sn2_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[16]	sn2_sn_is_chic	Indicates that SN2 is a CHI-C SN when set	RW	0b0
[15]	sn2_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 2 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[14]	sn2_pcmo_prop_dis	Disables PCMO propagation for SN 2 when set	RW	0b0
[13]	sn2_cmo_prop_en	Enables CMO propagation for SN 2 when set	RW	0b0
[12]	sn2_128b	Data width of SN 2  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[11]	sn1_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[10]	sn1_sn_is_chic	Indicates that SN1 is a CHI-C SN when set	RW	0b0
[9]	sn1_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 1 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[8]	sn1_pcmo_prop_dis	Disables PCMO propagation for SN 1 when set	RW	0b0
[7]	sn1_cmo_prop_en	Enables CMO propagation for SN 1 when set	RW	0b0
[6]	sn1_128b	Data width of SN 1  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0

Bits	Name	Description	Type	Reset
[5]	sn0_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[4]	sn0_sn_is_chic	Indicates that SN0 is a CHI-C SN when set	RW	0b0
[3]	sn0_pcmodsep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 0 when set <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[2]	sn0_pcmo_prop_dis	Disables PCMO propagation for SN 0 when set	RW	0b0
[1]	sn0_cmo_prop_en	Enables CMO propagation for SN 0 when set	RW	0b0
[0]	sn0_128b	Data width of SN 0 <b>0b1</b> 128 bits <b>0b0</b> 256 bits	RW	0b0

### 8.3.9.97 cmn\_hns\_sam\_6sn\_nodeid

Configures node IDs for subordinate nodes 3 to 5 in 6-SN configuration mode.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD20

##### 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

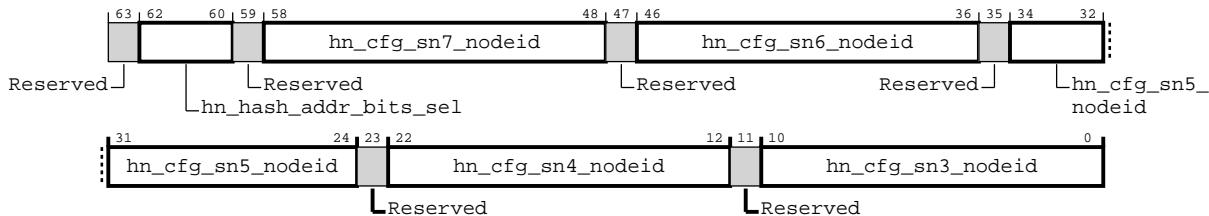
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.sam\_control bit 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 8-508: cmn\_hns\_sam\_6sn\_nodeid**



**Table 8-514: 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) <b>0b000</b> [16:8] address bits (Default) <b>0b001</b> [17:9] address bits <b>0b010</b> [18:10] address bits <b>0b011</b> [19:11] address bits <b>0b100</b> [20:12] address bits <b>0b101</b> [21:13] address bits <b>0b110</b> [22:14] address bits <b>0b111</b> [23:15] address bits <b>Others</b> Reserved	RW	0x0
[59]	Reserved	Reserved	RO	
[58:48]	hn_cfg_sn7_nodeid	SN 7 node ID	RW	0x0
[47]	Reserved	Reserved	RO	
[46:36]	hn_cfg_sn6_nodeid	SN 6 node ID	RW	0x0
[35]	Reserved	Reserved	RO	
[34:24]	hn_cfg_sn5_nodeid	SN 5 node ID	RW	0x0

Bits	Name	Description	Type	Reset
[23]	Reserved	Reserved	RO	
[22:12]	hn_cfg_sn4_nodeid	SN 4 node ID	RW	0x0
[11]	Reserved	Reserved	RO	
[10:0]	hn_cfg_sn3_nodeid	SN 3 node ID	RW	0x0

### 8.3.9.98 cmn\_hns\_sam\_sn\_properties1

Configures additional properties for all six SN targets and two range-based SN targets.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xCE8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

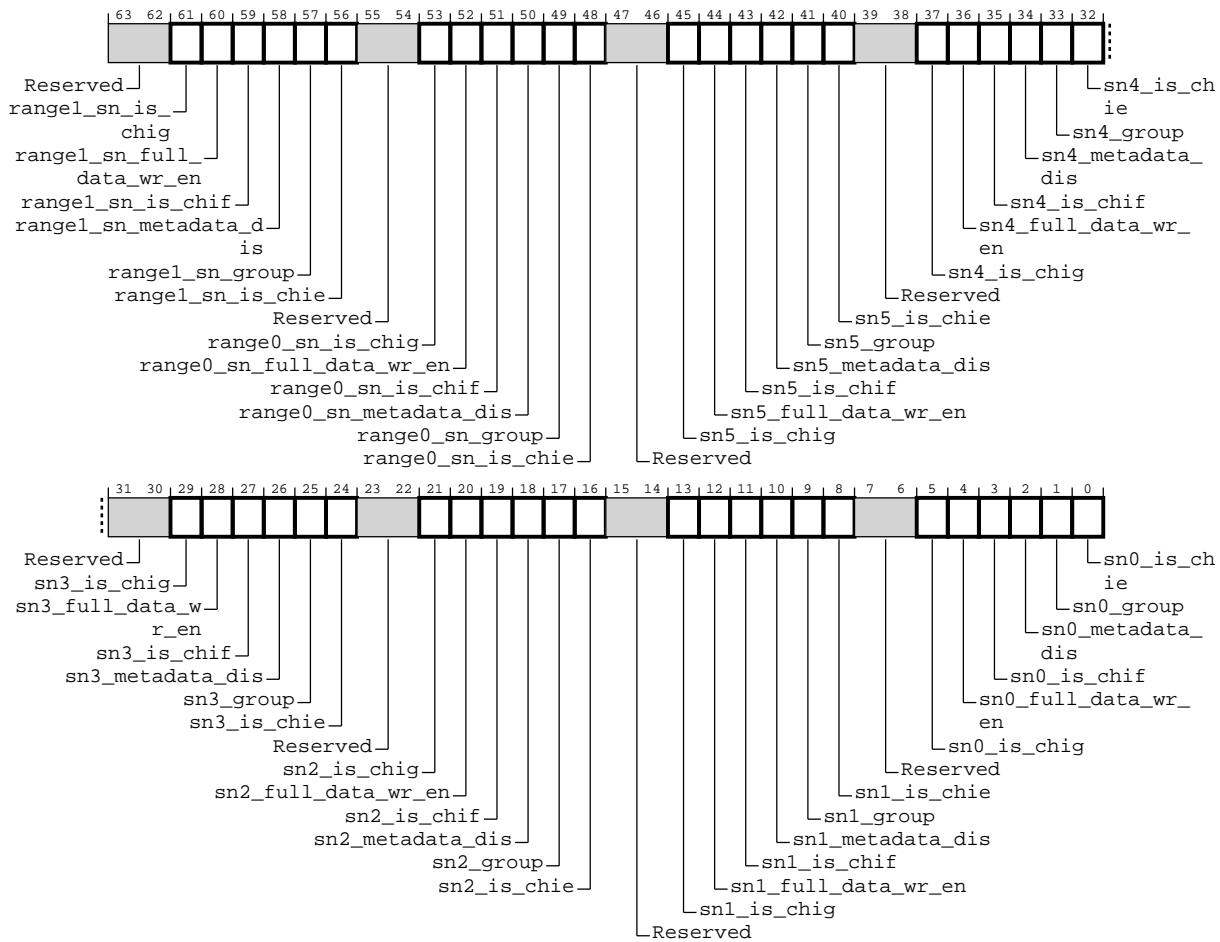
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-509: cmn\_hns\_sam\_sn\_properties1**



**Table 8-515: cmn\_hns\_sam\_sn\_properties1 attributes**

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61]	range1_sn_is_chig	Range 1 SN supports CHI-G	RW	0b0
[60]	range1_sn_full_data_wr_en	HNS implements always sending 64B write for Range 1 SN when set	RW	0b0
[59]	range1_sn_is_chif	Range 1 SN supports CHI-F	RW	0b0
[58]	range1_sn_metadata_dis	HNS implements metadata termination flow for Range 1 SN when set	RW	0b0
[57]	range1_sn_group	Specifies the SN-F grouping	RW	0b0
		<b>0b0</b> Group A		
		<b>0b1</b> Group B		
[56]	range1_sn_is_chie	Range 1 SN supports CHI-E	RW	0b0
[55:54]	Reserved	Reserved	RO	-
[53]	range0_sn_is_chig	Range 0 SN supports CHI-G	RW	0b0

Bits	Name	Description	Type	Reset
[52]	range0_sn_full_data_wr_en	HNS implements always sending 64B write for Range 0 SN when set	RW	0b0
[51]	range0_sn_is_chif	Range 0 SN supports CHI-F	RW	0b0
[50]	range0_sn_metadata_dis	HNS implements metadata termination flow for Range 0 SN when set	RW	0b0
[49]	range0_sn_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[48]	range0_sn_is_chie	Range 0 SN supports CHI-E	RW	0b0
[47:46]	Reserved	Reserved	RO	-
[45]	sn5_is_chig	SN5 supports CHI-G	RW	0b0
[44]	sn5_full_data_wr_en	HNS implements always sending 64B write for SN 5 when set	RW	0b0
[43]	sn5_is_chif	SN5 supports CHI-F	RW	0b0
[42]	sn5_metadata_dis	HNS implements metadata termination flow for SN 5 when set	RW	0b0
[41]	sn5_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[40]	sn5_is_chie	SN 5 supports CHI-E	RW	0b0
[39:38]	Reserved	Reserved	RO	-
[37]	sn4_is_chig	SN4 supports CHI-G	RW	0b0
[36]	sn4_full_data_wr_en	HNS implements always sending 64B write for SN 4 when set	RW	0b0
[35]	sn4_is_chif	SN4 supports CHI-F	RW	0b0
[34]	sn4_metadata_dis	HNS implements metadata termination flow for SN 4 when set	RW	0b0
[33]	sn4_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[32]	sn4_is_chie	SN 4 supports CHI-E	RW	0b0
[31:30]	Reserved	Reserved	RO	-
[29]	sn3_is_chig	SN3 supports CHI-G	RW	0b0
[28]	sn3_full_data_wr_en	HNS implements always sending 64B write for SN 3 when set	RW	0b0
[27]	sn3_is_chif	SN3 supports CHI-F	RW	0b0
[26]	sn3_metadata_dis	HNS implements metadata termination flow for SN 3 when set	RW	0b0
[25]	sn3_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0

Bits	Name	Description	Type	Reset
[24]	sn3_is_chie	SN 3 supports CHI-E	RW	0b0
[23:22]	Reserved	Reserved	RO	-
[21]	sn2_is_chig	SN2 supports CHI-G	RW	0b0
[20]	sn2_full_data_wr_en	HNS implements always sending 64B write for SN 2 when set	RW	0b0
[19]	sn2_is_chif	SN2 supports CHI-F	RW	0b0
[18]	sn2_metadata_dis	HNS implements metadata termination flow for SN 2 when set	RW	0b0
[17]	sn2_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[16]	sn2_is_chie	SN 2 supports CHI-E	RW	0b0
[15:14]	Reserved	Reserved	RO	-
[13]	sn1_is_chig	SN1 supports CHI-G	RW	0b0
[12]	sn1_full_data_wr_en	HNS implements always sending 64B write for SN 1 when set	RW	0b0
[11]	sn1_is_chif	SN1 supports CHI-F	RW	0b0
[10]	sn1_metadata_dis	HNS implements metadata termination flow for SN 1 when set	RW	0b0
[9]	sn1_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[8]	sn1_is_chie	SN 1 supports CHI-E	RW	0b0
[7:6]	Reserved	Reserved	RO	-
[5]	sn0_is_chig	SNO supports CHI-G	RW	0b0
[4]	sn0_full_data_wr_en	HNS implements always sending 64B write for SN 0 when set	RW	0b0
[3]	sn0_is_chif	SNO supports CHI-F	RW	0b0
[2]	sn0_metadata_dis	HNS implements metadata termination flow for SN 0 when set	RW	0b0
[1]	sn0_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[0]	sn0_is_chie	SN 0 supports CHI-E	RW	0b0

### 8.3.9.99 cmn\_hns\_sam\_sn\_properties2

Configures properties for SN-7 & SN-8.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD30

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

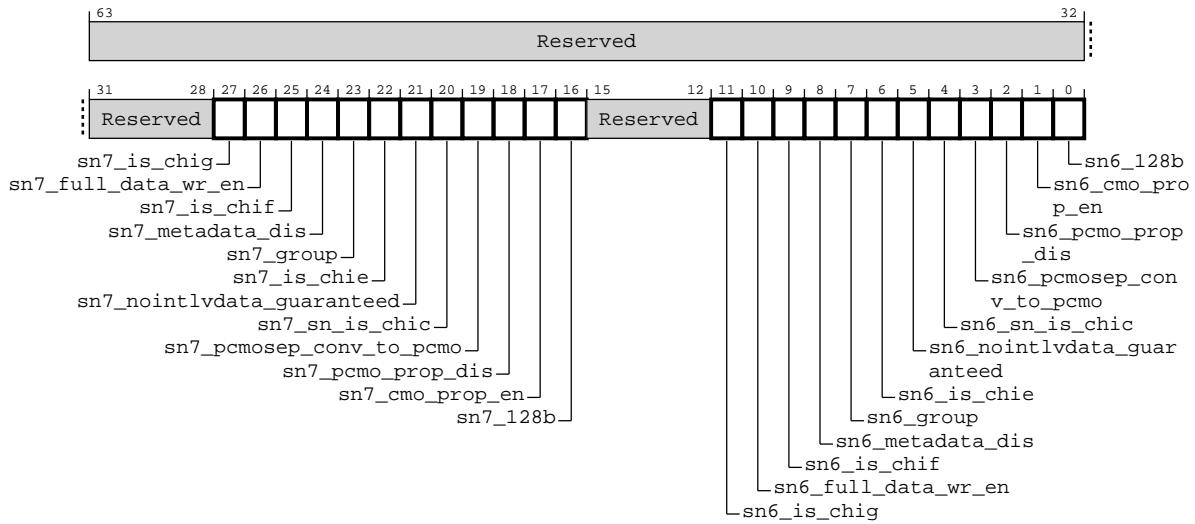
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-510: cmn\_hns\_sam\_sn\_properties2**



**Table 8-516: cmn\_hns\_sam\_sn\_properties2 attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	
[27]	sn7_is_chig	SN7 supports CHI-G	RW	0b0
[26]	sn7_full_data_wr_en	HNS implements always sending 64B write for SN 7 when set	RW	0b0
[25]	sn7_is_chif	SN7 supports CHI-F	RW	0b0
[24]	sn7_metadata_dis	HNS implements metadata termination flow for SN 7 when set	RW	0b0
[23]	sn7_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[22]	sn7_is_chie	SN 7 supports CHI-E	RW	0b0
[21]	sn7_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[20]	sn7_sn_is_chic	Indicates that SN7 is a CHI-C SN when set	RW	0b0
[19]	sn7_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 7 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[18]	sn7_pcmo_prop_dis	Disables PCMO propagation for SN 7 when set	RW	0b0
[17]	sn7_cmo_prop_en	Enables CMO propagation for SN 7 when set	RW	0b0
[16]	sn7_128b	Data width of SN 7  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0

Bits	Name	Description	Type	Reset
[15:12]	Reserved	Reserved	RO	
[11]	sn6_is_chig	SN6 supports CHI-G	RW	0b0
[10]	sn6_full_data_wr_en	HNS implements always sending 64B write for SN 6 when set	RW	0b0
[9]	sn6_is_chif	SN6 supports CHI-F	RW	0b0
[8]	sn6_metadata_dis	HNS implements metadata termination flow for SN 6 when set	RW	0b0
[7]	sn6_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[6]	sn6_is_chie	SN 6 supports CHI-E	RW	0b0
[5]	sn6_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[4]	sn6_sn_is_chic	Indicates that SN6 is a CHI-C SN when set	RW	0b0
[3]	sn6_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 6 when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[2]	sn6_pcmo_prop_dis	Disables PCMO propagation for SN 6 when set	RW	0b0
[1]	sn6_cmo_prop_en	Enables CMO propagation for SN 6 when set	RW	0b0
[0]	sn6_128b	Data width of SN 6  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0

### 8.3.9.100 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-4) : 0xF80 + #{8 \* index}

##### index(5-31)

0x6000 + #{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

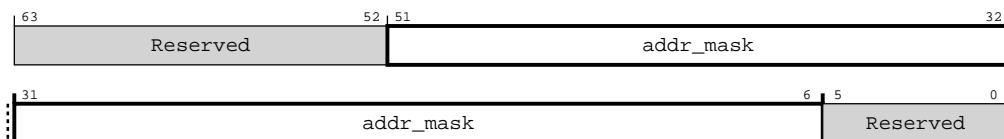
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-511: cmn\_hns\_cml\_port\_aggr\_grp0-4\_add\_mask**



**Table 8-517: 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	0xFFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

## 8.3.9.101 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

Address offset : index(0-4) : 0xF80 + #{8 \* index}

### index(5-31)

0x6000 + #{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 is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-512: cmn\_hns\_cml\_port\_aggr\_grp5-31\_add\_mask**



**Table 8-518: 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	0x3FFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

8.3.9.102 cmn\_hns\_cml\_cpag\_base\_idx\_grp0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures the CPAG base indexes.

# Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

## Width

64

## Address offset

index(0-7) : 0x6400 + #{{8 \* index}}

## Type

RW

## Reset value

See individual bit resets

## Usage constraints

This register is owned in the Root space and is 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 8-513: cmn\_hns\_cml\_cpag\_base\_idx\_grp0-3**

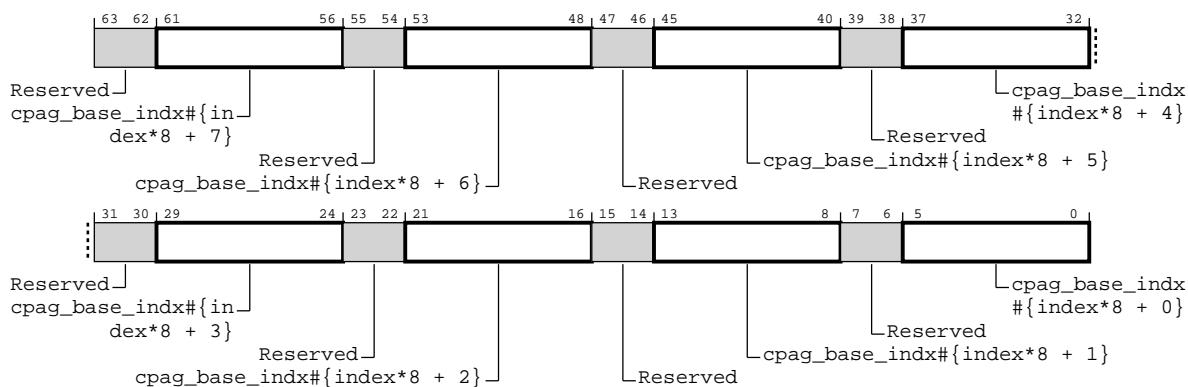


Table 8-519: cmn\_hns\_cml\_cpag\_base\_idx\_grp0-3 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	
[61:56]	cpag_base_idx#{index*8 + 7}	Configures the CPAG base index #{8*index + 7}	RW	0x3F

Bits	Name	Description	Type	Reset
[55:54]	Reserved	Reserved	RO	
[53:48]	cpag_base_idx#{index*8 + 6}	Configures the CPAG base index #{8*index + 6}	RW	0x3F
[47:46]	Reserved	Reserved	RO	
[45:40]	cpag_base_idx#{index*8 + 5}	Configures the CPAG base index #{8*index + 5}	RW	0x3F
[39:38]	Reserved	Reserved	RO	
[37:32]	cpag_base_idx#{index*8 + 4}	Configures the CPAG base index #{8*index + 4}	RW	0x3F
[31:30]	Reserved	Reserved	RO	
[29:24]	cpag_base_idx#{index*8 + 3}	Configures the CPAG base index #{8*index + 3}	RW	0x3F
[23:22]	Reserved	Reserved	RO	
[21:16]	cpag_base_idx#{index*8 + 2}	Configures the CPAG base index #{8*index + 2}	RW	0x3F
[15:14]	Reserved	Reserved	RO	
[13:8]	cpag_base_idx#{index*8 + 1}	Configures the CPAG base index #{8*index + 1}	RW	0x3F
[7:6]	Reserved	Reserved	RO	
[5:0]	cpag_base_idx#{index*8 + 0}	Configures the CPAG base index #{8*index + 0}	RW	0x3F

### 8.3.9.103 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-1) : 0xFB0 + #{8 \* index}

##### index(2-12)

0x6100 + #{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

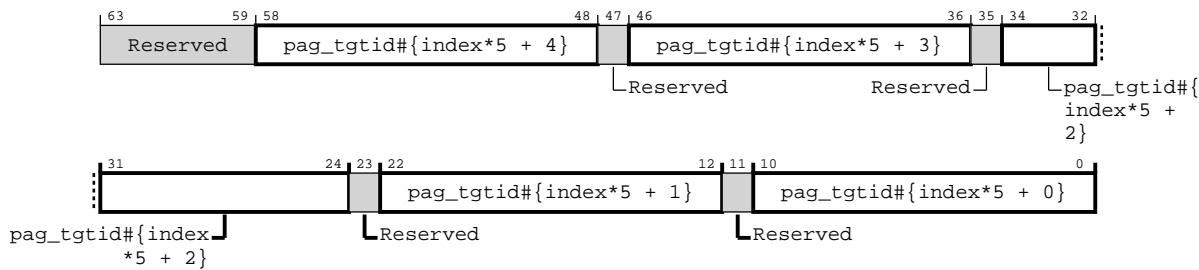
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-514: cmn\_hns\_cml\_port\_aggr\_grp\_reg0-12**



**Table 8-520: cmn\_hns\_cml\_port\_aggr\_grp\_reg0-12 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	
[58:48]	pag_tgtid#{index*5 + 4}	Specifies the target ID #{index*5 + 4} for CPAG	RW	0b0
[47]	Reserved	Reserved	RO	
[46:36]	pag_tgtid#{index*5 + 3}	Specifies the target ID #{index*5 + 3} for CPAG	RW	0b0
[35]	Reserved	Reserved	RO	
[34:24]	pag_tgtid#{index*5 + 2}	Specifies the target ID {index*5 + 2} for CPAG	RW	0b0
[23]	Reserved	Reserved	RO	
[22:12]	pag_tgtid#{index*5 + 1}	Specifies the target ID {index*5 + 1} for CPAG	RW	0b0
[11]	Reserved	Reserved	RO	
[10:0]	pag_tgtid#{index*5 + 0}	Specifies the target ID {index*5 + 0} for CPAG	RW	0b0

## 8.3.9.104 cmn\_hns\_cml\_port\_aggr\_ctrl\_reg

Configures the CCIX port aggregation port groups

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xFD0

### 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 is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-515: cmn\_hns\_cml\_port\_aggr\_ctrl\_reg**



**Table 8-521: 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	0b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag4	<p>Specifies the number of CXRAs in CPAG4</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid3	Valid programming for CPAG + 3}, Enabled by default (backward compatible)	RW	0b1
[38:36]	num_cxg_pag3	<p>Specifies the number of CXRAs in CPAG3</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[35:28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27]	cpag_valid2	Valid programming for CPAG + 2}, Enabled by default (backward compatible)	RW	0b1
[26:24]	num_cxg_pag2	<p>Specifies the number of CXRAs in CPAG2</p> <p><b>Constraint</b></p> <p>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</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid1	Valid programming for CPAG + 1}, Enabled by default (backward compatible)	RW	0b1
[14:12]	num_cxg_pag1	<p>Specifies the number of CXRAs in CPAG1</p> <p><b>Constraint</b></p> <p>May use pag_tgtid2 through pag_tgtid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

Bits	Name	Description	Type	Reset
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid0	Valid programming for CPAG, Enabled by default (backward compatible)	RW	0b1
[2:0]	num_cxg_pago	<p>Specifies the number of CXRAs in CPAGO</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

### 8.3.9.105 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(1-6) : 0x6200 + #{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`

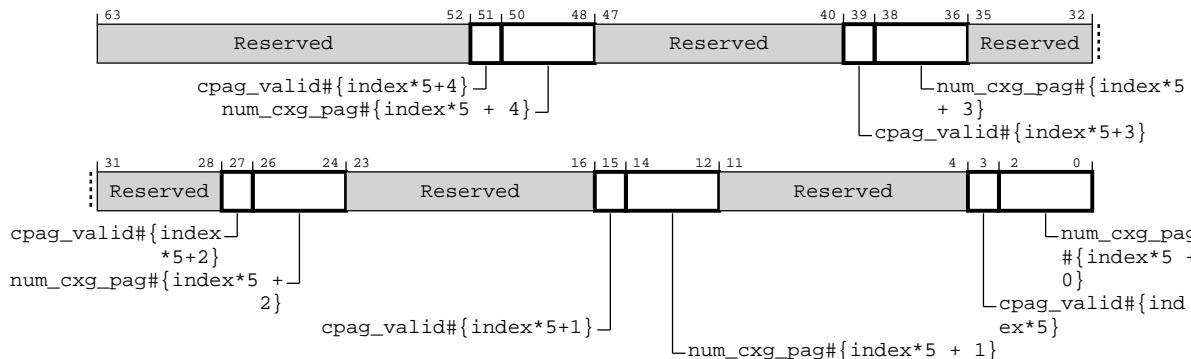
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.sam_control` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.sam_control` bit and `cmn_hns_rcr.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 8-516: cmn\_hns\_cml\_port\_aggr\_ctrl\_reg1-6**



**Table 8-522: cmn\_hns\_cml\_port\_aggr\_ctrl\_reg1-6 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	cpag_valid#{index*5+4}	Valid programming for CPAG #{index*5 + 4}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag#{index*5 + 4}	<p>Specifies the number of CXRAs in CPAG4#{index*5 + 4}</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{index*5 + 3}, Enabled by default (backward compatible)	RW	0b1
[38:36]	num_cxg_pag#{index*5 + 3}	<p>Specifies the number of CXRAs in CPAG3#{index*5 + 3}</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[35:28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{index*5 + 2}, Enabled by default (backward compatible)	RW	0b1
[26:24]	num_cxg_pag#{index*5 + 2}	<p>Specifies the number of CXRAs in CPAG2#{index*5 + 2}</p> <p><b>Constraint</b></p> <p>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</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{index*5 + 1}, Enabled by default (backward compatible)	RW	0b1
[14:12]	num_cxg_pag#{index*5 + 1}	<p>Specifies the number of CXRAs in CPAG1#{index*5 + 1}</p> <p><b>Constraint</b></p> <p>May use pag_tgtid2 through pag_tgtid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

Bits	Name	Description	Type	Reset
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	0b1
[2:0]	num_cxg_pag#{index*5 + 0}	<p>Specifies the number of CXRAs in CPAG#{index*5 + 0}</p> <p><b>Constraint</b></p> <p>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><b>0b000</b> 1 port used</p> <p><b>0b001</b> 2 ports used</p> <p><b>0b010</b> 4 ports used</p> <p><b>0b011</b> 8 ports used</p> <p><b>0b100</b> 16 ports used</p> <p><b>0b101</b> 32 ports used</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b000

### 8.3.9.106 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

0xF50

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.ppu

## Secure group override

cmn\_hns\_scr.ppu

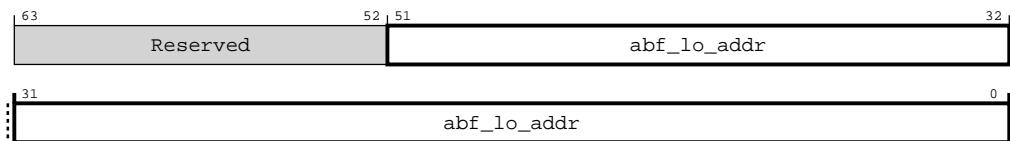
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ppu bit and cmn\_hns\_rcr.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 8-517: cmn\_hns\_abf\_lo\_addr**



**Table 8-523: 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	0b0

## 8.3.9.107 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

0xF58

#### Type

RW

#### Reset value

See individual bit resets

### **Root group override**

cmn\_hns\_rcr.ppu

### **Secure group override**

cmn\_hns\_scr.ppu

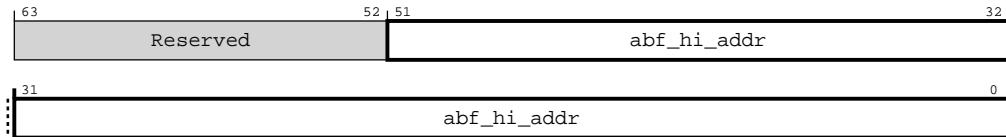
### **Usage constraints**

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ppu bit and cmn\_hns\_rcr.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 8-518: cmn\_hns\_abf\_hi\_addr**



**Table 8-524: 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	0b0

### [8.3.9.108 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**

0xF60

#### **Type**

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.ppu

### Secure group override

cmn\_hns\_scr.ppu

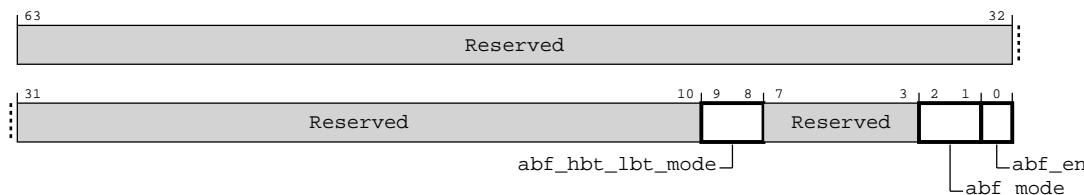
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.ppu bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.ppu bit and cmn\_hns\_rcr.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 8-519: cmn\_hns\_abf\_pr**



**Table 8-525: 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 <b>0b00</b> All addresses in ABF range (HBT and LBT) flushed <b>0b01</b> All HBT addresses in ABF range flushed <b>0b10</b> All LBT addresses in ABF range flushed <b>0b11</b> Reserved <b>Note</b> At this time, only 0b01 is supported.	RW	0b01
[7:3]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[2:1]	abf_mode	ABF mode <b>0b00</b> Clean Invalidate; WB dirty data and invalidate local copy <b>0b01</b> Make Invalidate; invalidate without writing back dirty data <b>0b10</b> Clean Shared; WB dirty data and can keep clean copy <b>0b11</b> Reserved	RW	0b00
[0]	abf_en	Start Address Based Flushing based on high and low address ranges	RW	0b0

### 8.3.9.109 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

0xF68

##### Type

RO

##### Reset value

See individual bit resets

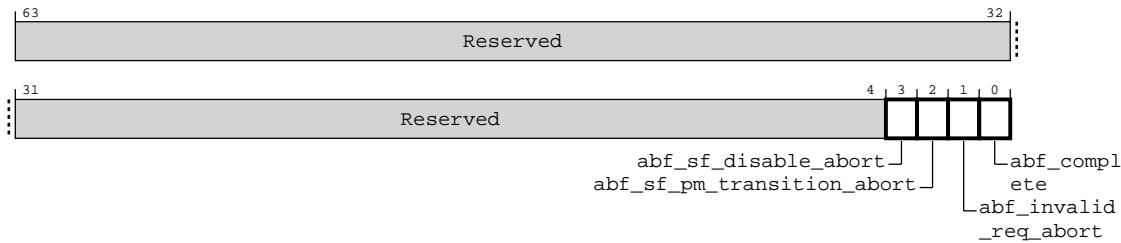
##### Usage constraints

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 8-520: cmn\_hns\_abf\_sr**



**Table 8-526: 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	0b0
[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	0b0
[1]	abf_invalid_req_abort	ABF request made while PM is not in FAM/HAM/SF_ONLY mode; request aborted in this case	RO	0b0
[0]	abf_complete	ABF completed	RO	0b0

### 8.3.9.110 cmn\_hns\_cbusy\_write\_limit\_ctl

Cbusy threshold limits for POCQ write entries. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1000

##### Type

RW

##### Reset value

See individual bit resets

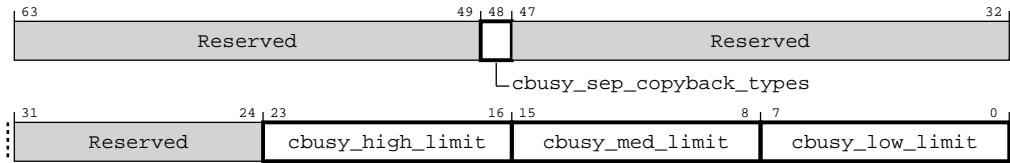
##### Usage constraints

This register is owned in the Root space and is 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 8-521: cmn\_hns\_cbusy\_write\_limit\_ctl**



**Table 8-527: 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	0b0
[47:24]	Reserved	Reserved	RO	-
[23:16]	cbusy_high_limit	POCQ limit for Write CBusy High	RW	0x18
[15:8]	cbusy_med_limit	POCQ limit for Write CBusy Med	RW	0x10
[7:0]	cbusy_low_limit	POCQ limit for Write CBusy Low	RW	0x8

## 8.3.9.111 cmn\_hns\_cbusy\_resp\_ctl

Controls the responses sent from HN-S to RN-F. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x1008

#### Type

RW

#### Reset value

See individual bit resets

#### Root group override

cmn\_hns\_rcr.sam\_control

#### Secure group override

cmn\_hns\_scr.sam\_control

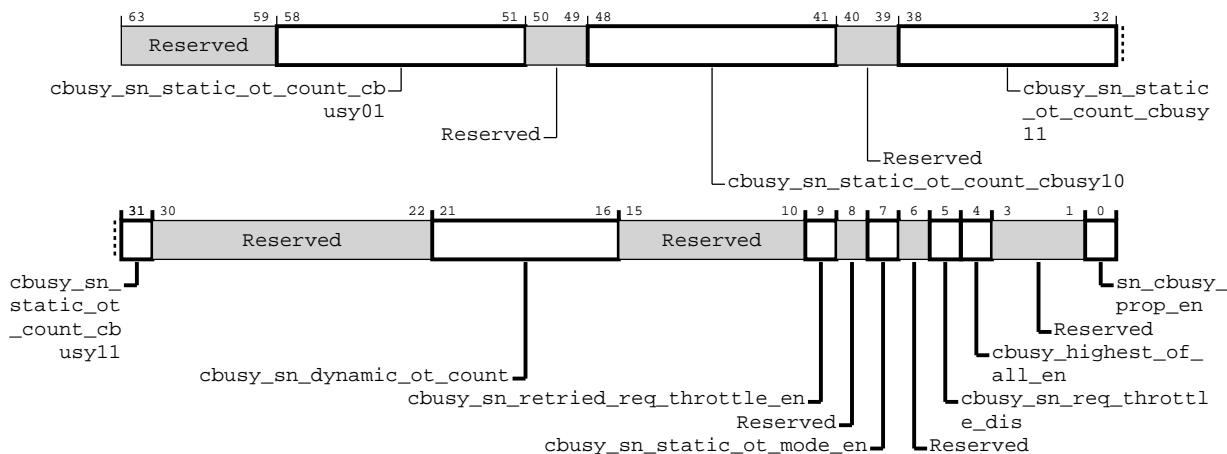
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-522: cmn\_hns\_cbusy\_resp\_ctl**



**Table 8-528: 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.  <b>CONSTRAINT</b> 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.  <b>CONSTRAINT</b> Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy01	RW	Configuration dependent
[40:39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:31]	cbusy_sn_static_ot_count_cbusy11	<p>Specifies the maximum number of transactions to SN-F when SN Cbusy=11 in static throttling mode.</p> <p><b>CONSTRAINT</b></p> <p>Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy10</p>	RW	Configuration dependent
[30:22]	Reserved	Reserved	RO	-
[21:16]	cbusy_sn_dynamic_ot_count	<p>Specifies the granularity at which HN-F will dynamically throttle transactions to SN-F.</p> <p><b>CONSTRAINT</b></p> <p>1,2,4,8 are the the allowed values</p>	RW	0b000100
[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	0b0
[8]	Reserved	Reserved	RO	-
[7]	cbusy_sn_static_ot_mode_en	<p>Controls cbusy between HN-F and SN-F</p> <p><b>0b0</b> HN-F will dynamically throttle outstanding requests to SN-F</p> <p><b>0b1</b> HN-F will use fixed transactions count at each CBusy level at 1/4th POCQ granularity</p> <p><b>CONSTRAINT</b></p> <p>For SN request throttling, CBusy aggregation is always based on SN_CBusy[1:0] and cbusy_alt_mode_en is inapplicable</p>	RW	0b0
[6]	Reserved	Reserved	RO	-
[5]	cbusy_sn_req_throttle_dis	Disables Cbusy based request throttling from HN-S to SN-F when set to 0b1	RW	0b0
[4]	cbusy_highest_of_all_en	<p>Controls cbusy between HN-F and SN-F</p> <p><b>0b0</b> Will send the HN-F or SN-F as configured</p> <p><b>0b1</b> Will select highest CBusy value between the SN-F and HN-F</p>	RW	0b0
[3:1]	Reserved	Reserved	RO	-
[0]	sn_cbusy_prop_en	<p>Controls HN-F and SN-F cbusy on responses to RN-F</p> <p><b>0b0</b> HN-F's POCQ Cbusy is sent</p> <p><b>0b1</b> SN-F's Cbusy is sent</p>	RW	0b0

### 8.3.9.112 cmn\_hns\_cbusy\_sn\_ctl

Controls the SN-F cbusy thresholds. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1010

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

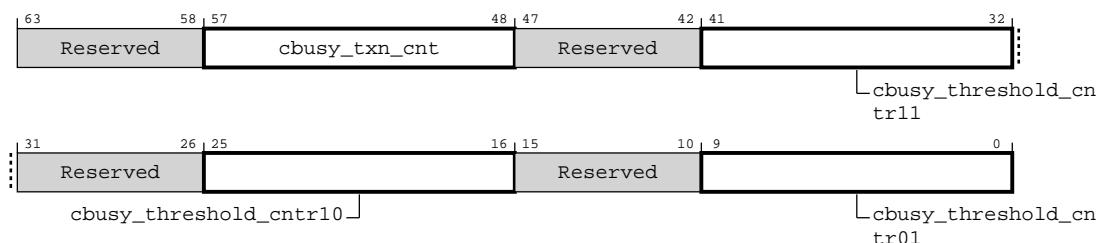
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-523: cmn\_hns\_cbusy\_sn\_ctl**



**Table 8-529: 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	0b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	cbusy_threshold_cntr11	CBusy threshold at which SN-F is considered busy for Counter_11	RW	0b0000010000
[31:26]	Reserved	Reserved	RO	-
[25:16]	cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10	RW	0b0000100000
[15:10]	Reserved	Reserved	RO	-
[9:0]	cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01	RW	0b0001000000

### 8.3.9.113 cmn\_hns\_cbusy\_sn1\_ctl

Controls the cbusy throttling to SN group 1. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1068

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

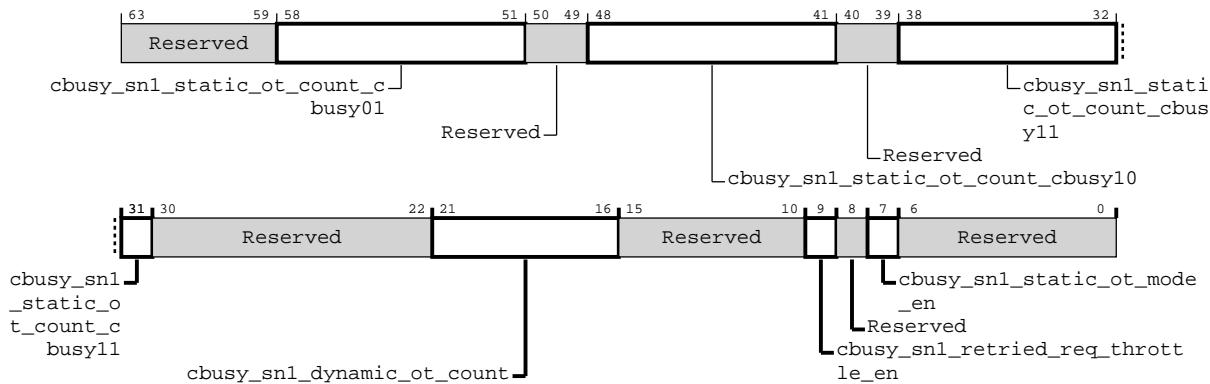
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-524: cmn\_hns\_cbusy\_sn1\_ctl**



**Table 8-530: cmn\_hns\_cbusy\_sn1\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:51]	cbusy_sn1_static_ot_count_cbusy01	Specifies the maximum number of transactions to SN-F sn group 1 when SN Cbusy=01 in static throttling mode.  <b>CONSTRAINT</b> Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1	RW	Configuration dependent
[50:49]	Reserved	Reserved	RO	-
[48:41]	cbusy_sn1_static_ot_count_cbusy10	Specifies the maximum number of transactions to SN-F sn group 1 when SN Cbusy=10 in static throttling mode.  <b>CONSTRAINT</b> 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_sn1_static_ot_count_cbusy11	Specifies the maximum number of transactions to SN-F sn group 1 when SN Cbusy=11 in static throttling mode.  <b>CONSTRAINT</b> 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_sn1_dynamic_ot_count	Specifies the granularity at which HN-F will dynamically throttle transactions to SN-F for sn group 1.  <b>CONSTRAINT</b> 1,2,4,8 are the allowed values	RW	0b000100
[15:10]	Reserved	Reserved	RO	-
[9]	cbusy_sn1_retried_req_throttle_en	Enables throttling retried requests with static grants (from SN group 1) along with dynamic credit requests	RW	0b0
[8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7]	cbusy_sn1_static_ot_mode_en	<p>Controls cbusy between HN-F and SN-F for sn group 1</p> <p><b>0b0</b> HN-F will dynamically throttle outstanding requests to SN-F</p> <p><b>0b1</b> HN-F will use fixed transactions count at each CBusy level at 1/4th POCQ granularity</p> <p><b>CONSTRAINT</b> For SN request throttling, CBusy aggregation is always based on SN_CBusy[1:0] and cbusy_alt_mode_en is inapplicable</p>	RW	0b0
[6:0]	Reserved	Reserved	RO	-

### 8.3.9.114 cmn\_hns\_cbusy\_sn1\_threshold

Controls the SN-F group 1 cbusy thresholds. **CONSTRAINT:** The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1070

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

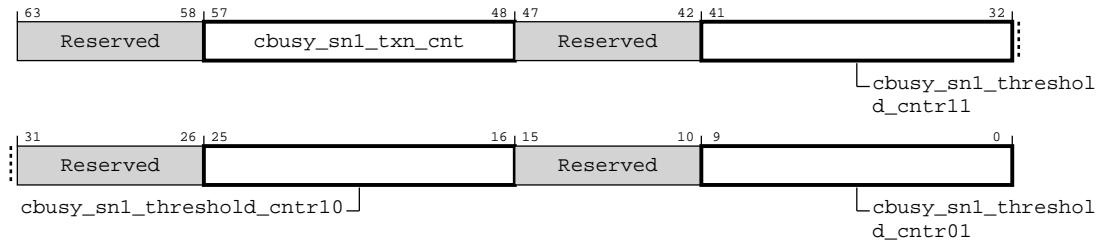
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-525: cmn\_hns\_cbusy\_sn1\_threshold**



**Table 8-531: cmn\_hns\_cbusy\_sn1\_threshold attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	cbusy_sn1_txn_cnt	Number of sn group 1 transactions over which the counters are tracked	RW	0b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	cbusy_sn1_threshold_cntr11	CBusy threshold at which SN-F sn group 1 is considered busy for Counter_11	RW	0b0000010000
[31:26]	Reserved	Reserved	RO	-
[25:16]	cbusy_sn1_threshold_cntr10	CBusy threshold at which SN-F sn group 1 is considered busy for Counter_10	RW	0b0000100000
[15:10]	Reserved	Reserved	RO	-
[9:0]	cbusy_sn1_threshold_cntr01	CBusy threshold at which SN-F sn group 1 is considered busy for Counter_01	RW	0b0001000000

### 8.3.9.115 cmn\_hns\_lbt\_cbusy\_ctl

Controls the CBusy response for LCN Bound Transactions. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1018

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

`cmn_hns_rcr.sam_control`

### Secure group override

`cmn_hns_scr.sam_control`

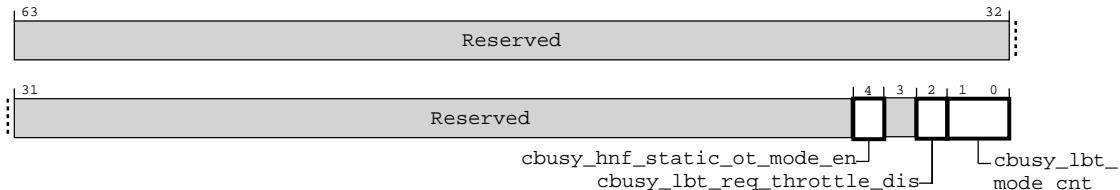
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.sam_control` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.sam_control` bit and `cmn_hns_rcr.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 8-526: cmn\_hns\_lbt\_cbusy\_ctl**



**Table 8-532: cmn\_hns\_lbt\_cbusy\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4]	cbusy_lbt_retried_req_throttle_en	Enables throttling retried requests with static grants (from LBT) along with dynamic credit requests	RW	0b0
[3]	Reserved	Reserved	RO	-
[2]	cbusy_lbt_req_throttle_dis	Disables Cbusy based request throttling from HNS to LBT when set to 0b1	RW	0b0
[1:0]	cbusy_lbt_mode_cnt	<p>Controls the propagation of Cbusy field for LCN bound transactions.</p> <p><b>0b00</b> Send HNS POCQ Cbusy on all responses based on the limits programmed in <code>cmn_hns_cbusy_limit_ctl</code></p> <p><b>0b01</b> Pass through HNF CBusy on late completion responses (CompData, Comp)</p> <p><b>0b10</b> Greater of POCQ Cbusy or HNF Cbusy. Applicable to responses where remote Cbusy can be sent</p>	RW	0b00

### 8.3.9.116 cmn\_hns\_cbusy\_ccg\_ctl

Controls the CBusy response for LCN Bound Transactions to CCG. CONSTRAINT: The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1080

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

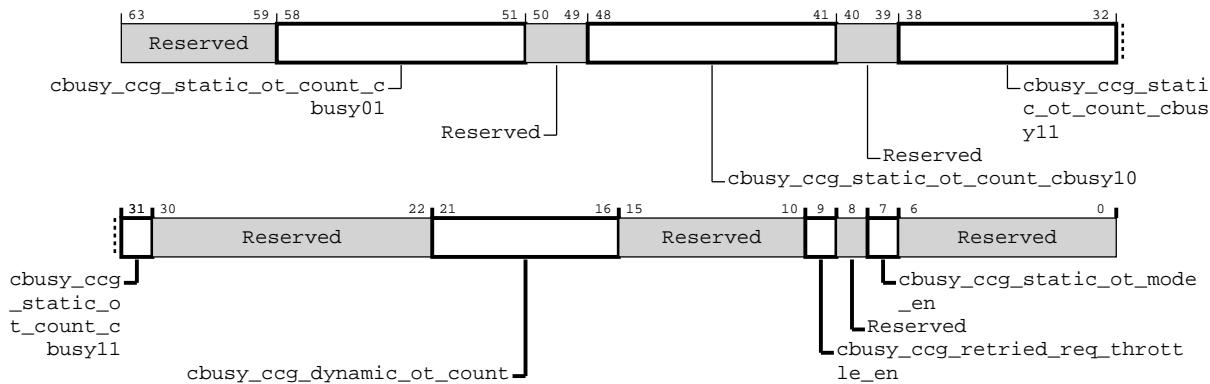
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-527: cmn\_hns\_cbusy\_ccg\_ctl**



**Table 8-533: cmn\_hns\_cbusy\_ccg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:51]	cbusy_ccg_static_ot_count_cbusy01	Specifies the maximum number of transactions to CCG when HN Cbusy=01 in static throttling mode.  <b>CONSTRAINT</b> Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1	RW	Configuration dependent
[50:49]	Reserved	Reserved	RO	-
[48:41]	cbusy_ccg_static_ot_count_cbusy10	Specifies the maximum number of transactions to CCG when HN Cbusy=10 in static throttling mode.  <b>CONSTRAINT</b> 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_ccg_static_ot_count_cbusy11	Specifies the maximum number of transactions to CCG when HN Cbusy=11 in static throttling mode.  <b>CONSTRAINT</b> 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_ccg_dynamic_ot_count	Specifies the granularity at which HN-F will dynamically throttle transactions to CCG.  <b>CONSTRAINT</b> 1,2,4,8 are the allowed values	RW	0b000100
[15:10]	Reserved	Reserved	RO	-
[9]	cbusy_ccg_retried_req_throttle_en	Enables throttling retried requests with static grants (from CCG) along with dynamic credit requests	RW	0b0
[8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7]	cbusy_ccg_static_ot_mode_en	<p>Controls cbusy between HN-F and CCG</p> <p><b>0b0</b> HN-F will dynamically throttle outstanding requests to HN-F</p> <p><b>0b1</b> HN-F will use fixed transactions count at each CBusy level at 1/4th POCQ granularity</p> <p><b>CONSTRAINT</b> For SN request throttling, CBusy aggregation is always based on SN_CBusy[1:0] and cbusy_alt_mode_en is inapplicable</p>	RW	0b0
[6:0]	Reserved	Reserved	RO	-

### 8.3.9.117 cmn\_hns\_cbusy\_ccg\_threshold

Controls the CCG cbusy thresholds. **CONSTRAINT:** The hns\_adv\_cbusy\_mode\_en must be 0b1 to use this feature.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1088

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.sam\_control

##### Secure group override

cmn\_hns\_scr.sam\_control

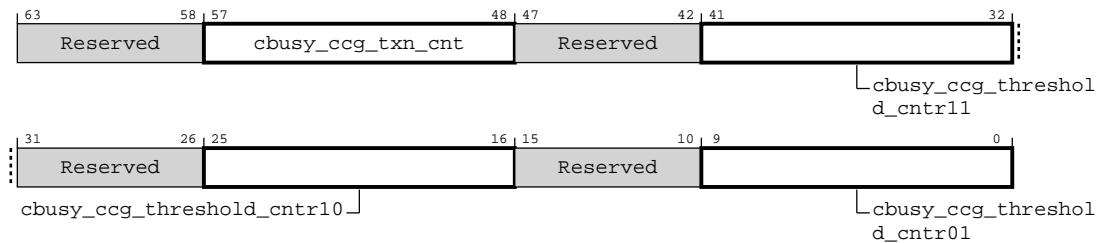
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-528: cmn\_hns\_cbusy\_ccg\_threshold**



**Table 8-534: cmn\_hns\_cbusy\_ccg\_threshold attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	cbusy_ccg_txn_cnt	Number of HN-F transactions over which the counters are tracked	RW	0b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	cbusy_ccg_threshold_cntr11	CBusy threshold at which HN-F is considered busy for Counter_11	RW	0b0000010000
[31:26]	Reserved	Reserved	RO	-
[25:16]	cbusy_ccg_threshold_cntr10	CBusy threshold at which HN-F is considered busy for Counter_10	RW	0b0000100000
[15:10]	Reserved	Reserved	RO	-
[9:0]	cbusy_ccg_threshold_cntr01	CBusy threshold at which HN-F is considered busy for Counter_01	RW	0b0001000000

### 8.3.9.118 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

0x1020

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.qos

##### Secure group override

cmn\_hns\_scr.qos

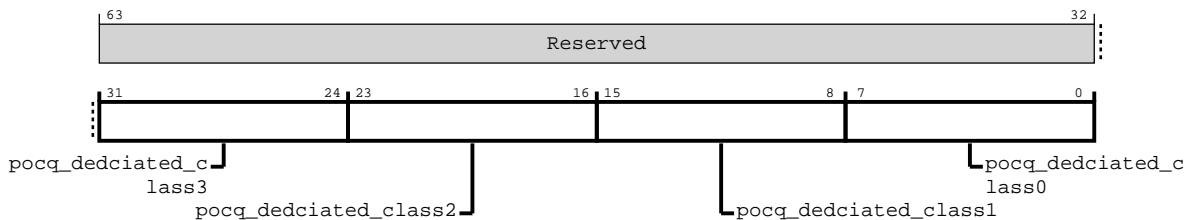
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.qos 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 8-529: cmn\_hns\_pocq\_alloc\_class\_dedicated**



**Table 8-535: cmn\_hns\_pocq\_alloc\_class\_dedicated attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pocq_dedicated_class3	<p>Dedicated number of entries for Class 3 in POCQ</p> <p><b>CONSTRAINT</b> Sum of dedicated entries for classes &amp; SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM.</p> <p><b>CONSTRAINT</b> <math>\text{hns\_pocq\_dedicated\_class3} &lt; \text{hns\_pocq\_max\_allowed\_class3}</math></p>	RW	0b00000000
[23:16]	pocq_dedicated_class2	<p>Dedicated number of entries for Class 2 in POCQ</p> <p><b>CONSTRAINT</b> Sum of dedicated entries for classes &amp; SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM.</p> <p><b>CONSTRAINT</b> <math>\text{hns\_pocq\_dedicated\_class2} &lt; \text{hns\_pocq\_max\_allowed\_class2}</math></p>	RW	0b00000000
[15:8]	pocq_dedicated_class1	<p>Dedicated number of entries for Class 1 in POCQ</p> <p><b>CONSTRAINT</b> Sum of dedicated entries for classes &amp; SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM.</p> <p><b>CONSTRAINT</b> <math>\text{hns\_pocq\_dedicated\_class1} &lt; \text{hns\_pocq\_max\_allowed\_class1}</math></p>	RW	0b00000000

Bits	Name	Description	Type	Reset
[7:0]	pocq_dedicated_class0	Dedicated number of entries for Class 0 in POCQ  <b>CONSTRAINT</b> Sum of dedicated entries for classes & SEQ can not exceed <code>HNS_NUM_ENTRIES_POCQ_PARAM</code> .  <b>CONSTRAINT</b> <code>hns_pocq_dedicated_class0 &lt; hns_pocq_max_allowed_class0</code>	RW	0b00000000

### 8.3.9.119 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

0x1028

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

`cmn_hns_rcr.qos`

##### Secure group override

`cmn_hns_scr.qos`

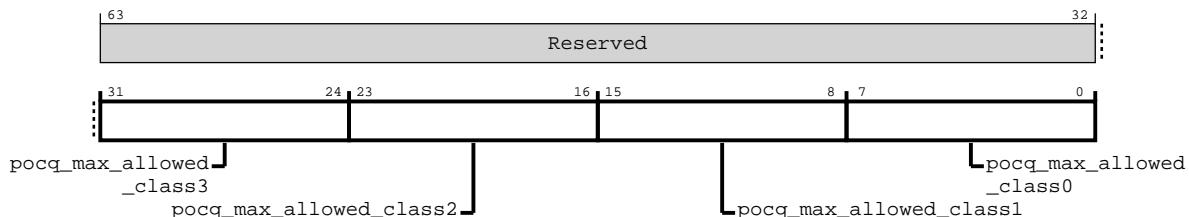
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.qos` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.qos` bit and `cmn_hns_rcr.qos` 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 8-530: cmn\_hns\_pocq\_alloc\_class\_max\_allowed**



**Table 8-536: cmn\_hns\_pocq\_alloc\_class\_max\_allowed attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pocq_max_allowed_class3	Maximum number of entries for Class 3 in POCQ <b>CONSTRAINT</b> hns_pocq_dedicated_class3 < hns_pocq_max_allowed_class3	RW	Configuration dependent
[23:16]	pocq_max_allowed_class2	Maximum number of entries for Class 2 in POCQ <b>CONSTRAINT</b> hns_pocq_dedicated_class2 < hns_pocq_max_allowed_class2	RW	Configuration dependent
[15:8]	pocq_max_allowed_class1	Maximum number of entries for Class 1 in POCQ <b>CONSTRAINT</b> hns_pocq_dedicated_class1 < hns_pocq_max_allowed_class1	RW	Configuration dependent
[7:0]	pocq_max_allowed_class0	Maximum number of entries for Class 0 in POCQ <b>CONSTRAINT</b> hns_pocq_dedicated_class0 < hns_pocq_max_allowed_class0	RW	Configuration dependent

### 8.3.9.120 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

0x1030

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.qos

### Secure group override

cmn\_hns\_scr.qos

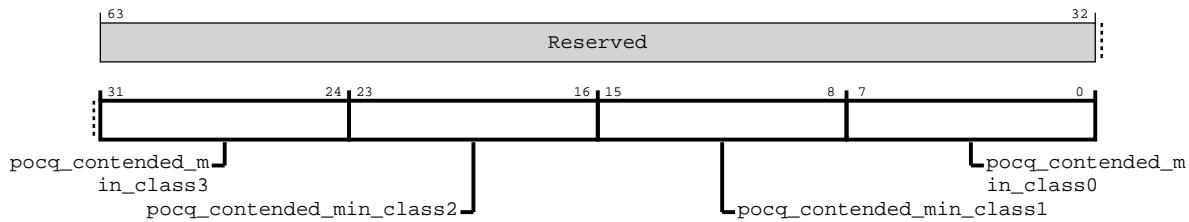
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.qos 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 8-531: cmn\_hns\_pocq\_alloc\_class\_contended\_min**



**Table 8-537: 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	0x8
[23:16]	pocq_contended_min_class2	Contended min entries for Class 2 in POCQ	RW	0x8
[15:8]	pocq_contended_min_class1	Contended min entries for Class 1 in POCQ	RW	0x8
[7:0]	pocq_contended_min_class0	Contended min entries for Class 0 in POCQ	RW	0x8

### 8.3.9.121 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

0x1038

## 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 is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.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 8-532: cmn\_hns\_pocq\_alloc\_misc\_max\_allowed**



**Table 8-538: 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. <b>Constraint</b> Only values of 1 or 2 supported.	RW	0x02
[7:0]	pocq_max_allowed_snpreq	Maximum number of entries for RXSNP requests in POCQ	RW	0x04

## 8.3.9.122 cmn\_hns\_class\_ctl

Class misc controls.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x1040

### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.qos

### Secure group override

cmn\_hns\_scr.qos

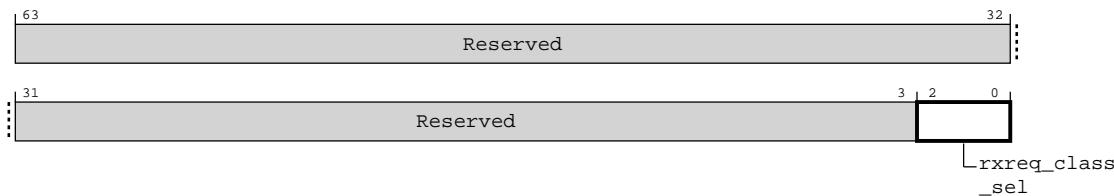
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.qos 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 8-533: cmn\_hns\_class\_ctl**



**Table 8-539: cmn\_hns\_class\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2:0]	rxreq_class_sel	RxReq Class select: <b>0b000</b> QoS based class selection <b>0b001</b> Request Opcode based class selection <b>Note</b> If un-supported value is programmed, default selection of QoS based is chosen.	RW	0b000

### 8.3.9.123 cmn\_hns\_pocq\_qos\_class\_ctl

QoS bases class identification controls.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1048

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.qos

##### Secure group override

cmn\_hns\_scr.qos

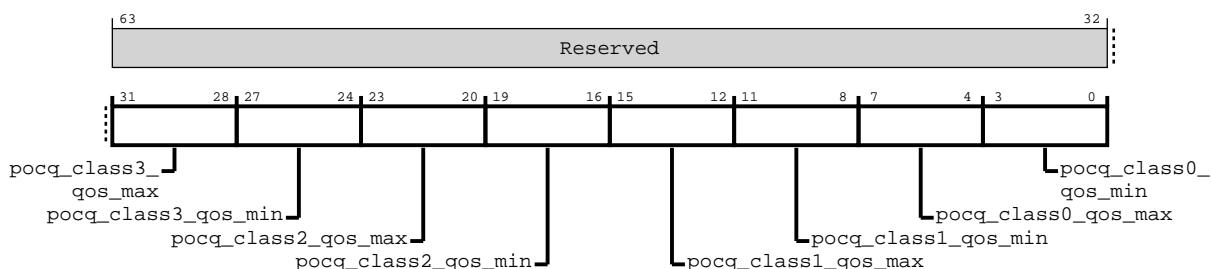
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.qos 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 8-534: cmn\_hns\_pocq\_qos\_class\_ctl**



**Table 8-540: 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	0b0111
[27:24]	pocq_class3_qos_min	QoS minimum value for Class 3	RW	0b0000
[23:20]	pocq_class2_qos_max	QoS maximum value for Class 2	RW	0b1011
[19:16]	pocq_class2_qos_min	QoS minimum value for Class 2	RW	0b1000
[15:12]	pocq_class1_qos_max	QoS maximum value for Class 1	RW	0b1110
[11:8]	pocq_class1_qos_min	QoS minimum value for Class 1	RW	0b1100
[7:4]	pocq_class0_qos_max	QoS maximum value for Class 0	RW	0b1111
[3:0]	pocq_class0_qos_min	QoS minimum value for Class 0	RW	0b1111

### 8.3.9.124 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

0x1050

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.qos

##### Secure group override

cmn\_hns\_scr.qos

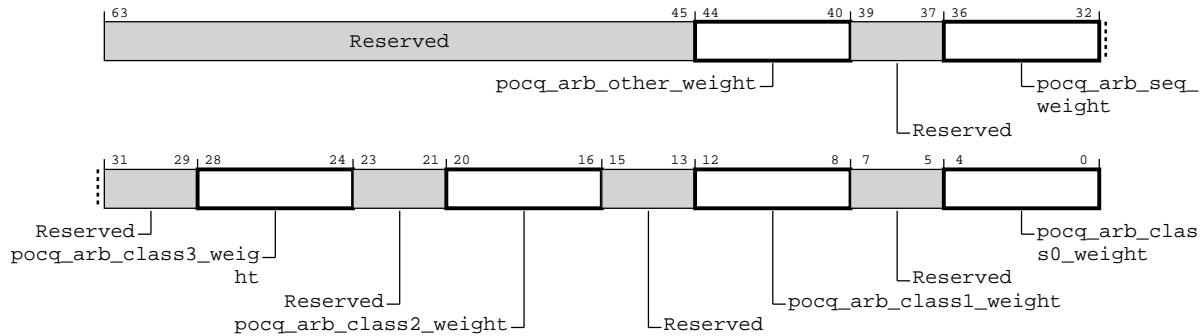
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.qos 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 8-535: cmn\_hns\_class\_pocq\_arb\_weight\_ctl**



**Table 8-541: 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	0b00000
[39:37]	Reserved	Reserved	RO	-
[36:32]	pocq_arb_seq_weight	SEQ weight for scheduling requests from POCQ	RW	0b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	pocq_arb_class3_weight	Class3 weight for scheduling requests from POCQ	RW	0b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	pocq_arb_class2_weight	Class2 weight for scheduling requests from POCQ	RW	0b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	pocq_arb_class1_weight	Class1 weight for scheduling requests from POCQ	RW	0b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	pocq_arb_class0_weight	Class0 weight for scheduling requests from POCQ	RW	0b00000

### 8.3.9.125 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

0x1058

##### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.qos

### Secure group override

cmn\_hns\_scr.qos

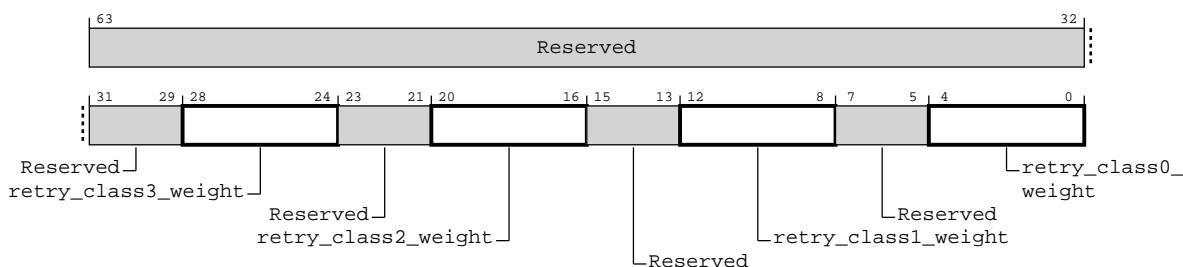
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.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 8-536: cmn\_hns\_class\_retry\_weight\_ctl**



**Table 8-542: cmn\_hns\_class\_retry\_weight\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:24]	retry_class3_weight	Overall Class3 weight for credit grant arbitration	RW	0b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	retry_class2_weight	Overall Class2 weight for credit grant arbitration	RW	0b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	retry_class1_weight	Overall Class1 weight for credit grant arbitration	RW	0b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	retry_class0_weight	Overall Class0 weight for credit grant arbitration	RW	0b00000

### 8.3.9.126 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

0x1060

### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.qos

### Secure group override

cmn\_hns\_scr.qos

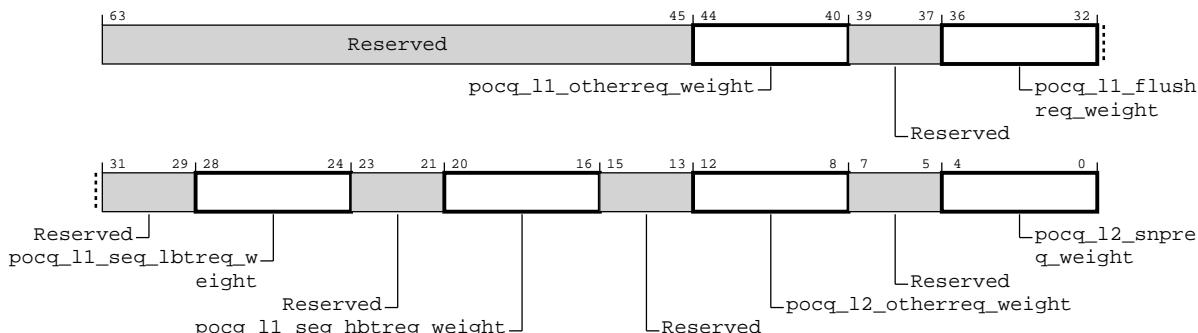
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.qos bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.qos bit and cmn\_hns\_rcr.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 8-537: cmn\_hns\_pocq\_misc\_retry\_weight\_ctl**



**Table 8-543: 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	0b00000
[39:37]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[36:32]	pocq_l1_flushreq_weight	Weight for SLF/SF Flush requests for POCQ allocation arbitration for Level 1.  <b>Note</b> This is first level arb weight control. Second level after this arb is for snpreq.	RW	0b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	pocq_l1_seq_lbreq_weight	Weight for SEQ-LBT requests for POCQ allocation arbitration for Level 1.  <b>Note</b> This is first level arb weight control. Second level after this arb is for snpreq.	RW	0b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	pocq_l1_seq_hbtreq_weight	Weight for SEQ-HBT requests for POCQ allocation arbitration for Level 1.  <b>Note</b> This is first level arb weight control. Second level after this arb is for snpreq.	RW	0b00000
[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.  <b>Note</b> This is second level arb weight control. First level is seq, flush, dbgrd, etc.	RW	0b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	pocq_l2_snpreq_weight	Weight for external snoop requests for POCQ allocation arbitration for Level 2.  <b>Note</b> This is second level arb weight control. First level is seq, flush, dbgrd, etc	RW	0b00000

### 8.3.9.127 cmn\_hns\_partner\_scratch\_reg0

Partner scratch register 0

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFE0

##### 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`

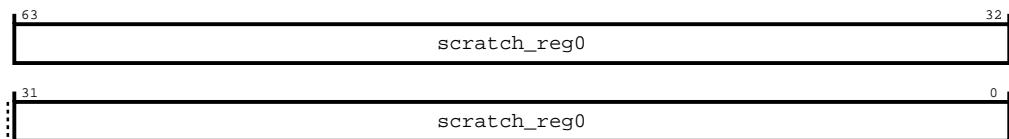
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.partner_scratch_override` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.partner_scratch_override` bit and `cmn_hns_rcr.partner_scratch_override` 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 8-538: cmn\_hns\_partner\_scratch\_reg0**



**Table 8-544: cmn\_hns\_partner\_scratch\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg0	64 bit scratch register 0 with read/write access	RW	0x00000000

### 8.3.9.128 cmn\_hns\_partner\_scratch\_reg1

Partner scratch register 1

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xFE8

##### 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`

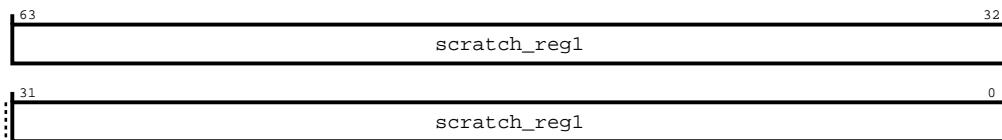
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.partner_scratch_override` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.partner_scratch_override` bit and `cmn_hns_rcr.partner_scratch_override` 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 8-539: cmn\_hns\_partner\_scratch\_reg1**



**Table 8-545: 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	0x00000000

### 8.3.9.129 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

0xB80

## Type

WO

## Reset value

See individual bit resets

## Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

## Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

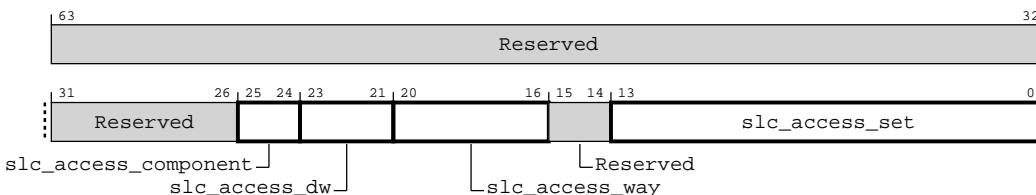
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-540: cmn\_hns\_cfg\_slcsf\_dbgrd**



**Table 8-546: 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  <b>0b01</b> SLC data read  <b>0b10</b> SLC tag read  <b>0b11</b> SF tag read	WO	0b00
[23:21]	slc_access_dw	64-bit chunk address for SLC data debug read access	WO	0x0
[20:16]	slc_access_way	Way address for SLC/SF debug read access	WO	0x00
[15:14]	Reserved	Reserved	RO	
[13:0]	slc_access_set	Set address for SLC/SF debug read access	WO	0x0

### 8.3.9.130 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

0xB88

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

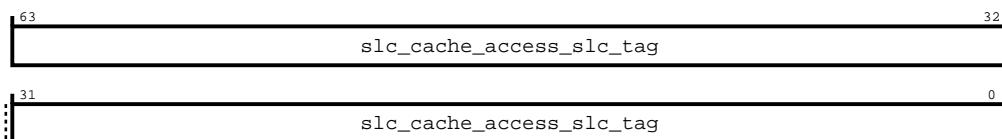
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-541: cmn\_hns\_slc\_cache\_access\_slc\_tag**



**Table 8-547: 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	0x0

### 8.3.9.131 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

0xB90

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

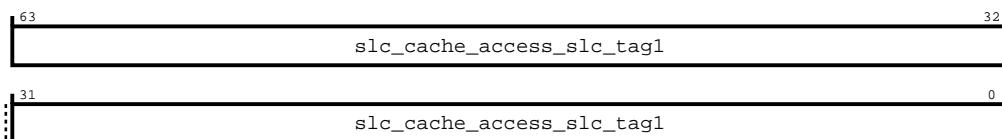
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-542: cmn\_hns\_slc\_cache\_access\_slc\_tag1**



**Table 8-548: 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	0x0

### 8.3.9.132 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

0xB98

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

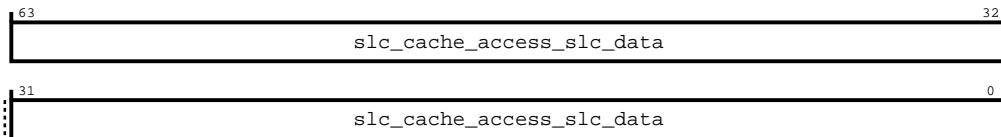
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-543: cmn\_hns\_slc\_cache\_access\_slc\_data**



**Table 8-549: 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	0x0

### 8.3.9.133 cmn\_hns\_slc\_cache\_access\_slc\_mte\_tag

Contains MTE Tag data for the corresponding SLC data RAM debug read.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xBC0

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

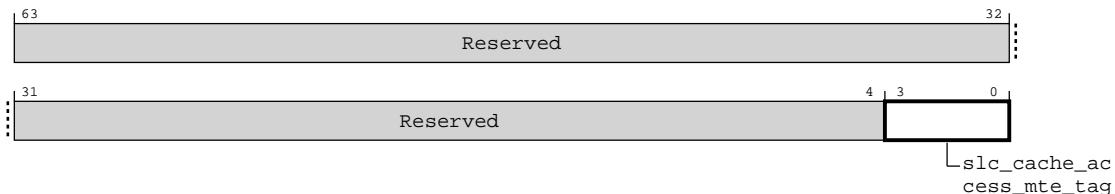
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-544: cmn\_hns\_slc\_cache\_access\_slc\_mte\_tag**



**Table 8-550: cmn\_hns\_slc\_cache\_access\_slc\_mte\_tag attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[3:0]	slc_cache_access_mte_tag	SLC MTE TAG corresponding to data RAM debug read data (128bit chunk of data)	RO	0x0

### 8.3.9.134 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

0xBA0

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

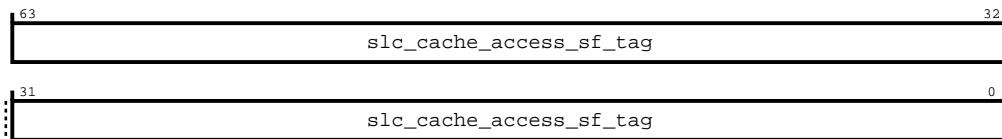
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-545: cmn\_hns\_slc\_cache\_access\_sf\_tag**



**Table 8-551: 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	0x0

### 8.3.9.135 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

0xBA8

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

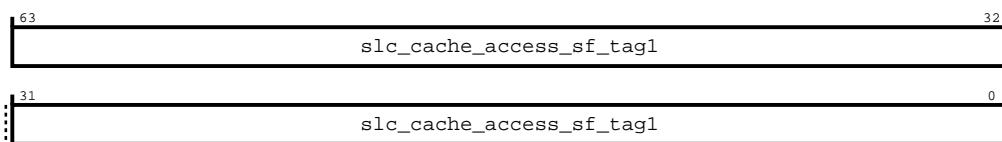
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-546: cmn\_hns\_slc\_cache\_access\_sf\_tag1**



**Table 8-552: 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	0x0

### 8.3.9.136 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

0xBB0

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.slcsf\_dbgrd

##### Secure group override

cmn\_hns\_scr.slcsf\_dbgrd

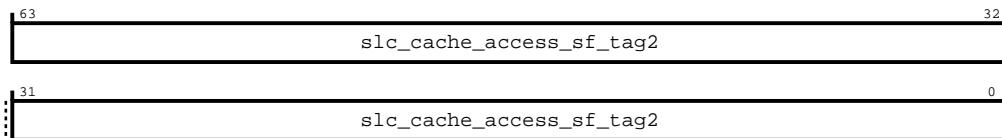
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.slcsf\_dbgrd bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.slcsf\_dbgrd bit and cmn\_hns\_rcr.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 8-547: cmn\_hns\_slc\_cache\_access\_sf\_tag2**



**Table 8-553: 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	0x0

### 8.3.9.137 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

0xD900

##### Type

RW

##### Reset value

See individual bit resets

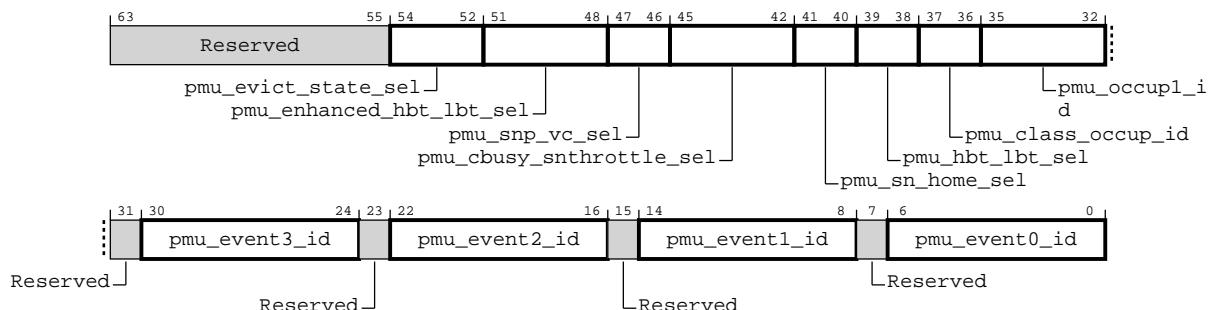
##### Usage constraints

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 8-548: cmn\_hns\_pmu\_event\_sel**



**Table 8-554: cmn\_hns\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[54:52]	pmu_evict_state_sel	HN-F SF/SLC Eviction data state select <b>0b000</b> All state <b>0b001</b> State EU <b>0b010</b> State EN <b>0b011</b> State SU <b>0b100</b> State SN <b>0b101</b> State MU <b>0b110</b> State MN	RW	0x0
[51:48]	pmu_enhanced_hbt_lbt_sel	HN-F PMU occupancy 1 select <b>0b0000</b> All occupancy selected <b>0b0001</b> HBT requests <b>0b0010</b> LBT requests <b>0b0011</b> All requests from local RN-F <b>0b0100</b> All requests from local RN-I <b>0b0101</b> All requests from CCG-LCN <b>0b0110</b> All requests from CCG RN-I/RN-F	RW	0x0
[47:46]	pmu_snp_vc_sel	HN-F PMU Snoop VC select when 3TXSNP channels are enabled <b>0b00</b> Snoop VCO to CCG <b>0b01</b> Snoop VC1 to local LCNs <b>0b10</b> Snoop VC2 to local RNFs <b>0b11</b> Reserved	RW	0b10

Bits	Name	Description	Type	Reset
[45:42]	pmu_cbusy_snthrottle_sel	<p>Filter for selecting specific SN throttle type</p> <p><b>0b0000</b> All SN types throttled</p> <p><b>0b0001</b> SN Group 0 Reads</p> <p><b>0b0010</b> SN Group 0 Non-Reads</p> <p><b>0b0011</b> SN Group 1 Reads</p> <p><b>0b0100</b> SN Group 1 Non-Reads</p> <p><b>0b0101</b> All SN Reads</p> <p><b>0b0110</b> All SN Non-Reads</p> <p><b>0b1001</b> Remote CCG Reads</p> <p><b>0b1010</b> Remote CCG Non-Reads</p> <p><b>0b1011</b> All LBT Reads</p> <p><b>0b1100</b> All LBT Non-Reads</p> <p><b>0b1101</b> All LBT throttled</p>	RW	0x0
[41:40]	pmu_sn_home_sel	<p>HN-F PMU SN/Home select</p> <p><b>0b00</b> All requests selected</p> <p><b>0b01</b> SN bound requests selected</p> <p><b>0b10</b> CCG/Remote home bound requests selected</p>	RW	0x0
[39:38]	pmu_hbt_lbt_sel	<p>HN-F PMU HBT/LBT select</p> <p><b>0b00</b> All requests selected</p> <p><b>0b01</b> HBT requests selected</p> <p><b>0b10</b> LBT requests selected</p>	RW	0x0

Bits	Name	Description	Type	Reset
[37:36]	pmu_class_occup_id	HN-F PMU Class select <b>0b00</b> Class 0 selected <b>0b01</b> Class 1 selected <b>0b10</b> Class 2 selected <b>0b11</b> Class 3 selected	RW	0x0
[35:32]	pmu_occup1_id	HN-F PMU occupancy 1 select <b>0b0000</b> All occupancy selected <b>0b0001</b> Read requests <b>0b0010</b> Write requests <b>0b0011</b> Atomic operation requests <b>0b0100</b> Stash requests <b>0b0101</b> RxSnp requests <b>0b0110</b> LBT requests <b>0b0111</b> HBT requests <b>0b1000</b> All requests from local RN-F <b>0b1001</b> All requests from lcal RN-I <b>0b1010</b> All requests from CCG-LCN <b>0b1011</b> All requests from CCG RN-I/RN-F	RW	0x0
[31]	Reserved	Reserved	RO	
[30:24]	pmu_event3_id	HN-F PMU Event 3 select; see pmu_event0_id for encodings	RW	0x00
[23]	Reserved	Reserved	RO	
[22:16]	pmu_event2_id	HN-F PMU Event 2 select; see pmu_event0_id for encodings	RW	0x00
[15]	Reserved	Reserved	RO	
[14:8]	pmu_event1_id	HN-F PMU Event 1 select; see pmu_event0_id for encodings	RW	0x00
[7]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>HN-F PMU Event 0 select</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> PMU_HN_CACHE_MISS_EVENT; counts total cache misses in first lookup result (high priority). Filtering is programmed in pmu_enhanced_hbt_lbt_sel. HN-S PMU doesn't count event for copyback write requests, since HN-S guarantees the the cacheline must be in certain case of SF/SLC hit/miss, for example with WrBackFull it must be SF hit SLC miss, otherwise the hit/miss rate calculated based on event counter doesn't reflect the real hit rate.</p> <p><b>0x02</b> PMU_HN_SLCSF_CACHE_ACCESS_EVENT; counts number of cache accesses in first access (high priority). Filtering is programmed in pmu_enhanced_hbt_lbt_sel. HN-S PMU doesn't count event for copyback write requests, since HN-S guarantees the the cacheline must be in certain case of SF/SLC hit/miss, for example with WrBackFull it must be SF hit SLC miss, otherwise the hit/miss rate calculated based on event counter doesn't reflect the real hit rate.</p> <p><b>0x03</b> PMU_HN_CACHE_FILL_EVENT; counts total allocations in HN SLC (all cache line allocations to SLC). Filtering is programmed in pmu_enhanced_hbt_lbt_sel.</p> <p><b>0x04</b> PMU_HN_POCQ_RETRY_EVENT; counts number of retried requests. Filtering is programmed in pmu_occup1_id.</p> <p><b>0x05</b> PMU_HN_POCQ_REQS_RECVD_EVENT; counts number of requests received by HN. Filtering is programmed in pmu_occup1_id.</p> <p><b>0x06</b> PMU_HN_SF_HIT_EVENT; counts number of SF hits. Filtering is programmed in pmu_enhanced_hbt_lbt_sel. HN-S PMU doesn't count event for copyback write requests, since HN-S guarantees the the cacheline must be in certain case of SF/SLC hit/miss, for example with WrBackFull it must be SF hit SLC miss, otherwise the hit/miss rate calculated based on event counter doesn't reflect the real hit rate.</p> <p><b>0x07</b> PMU_HN_SF_EVICTIONS_EVENT; counts number of SF eviction cache invalidations initiated for specific SF state. Filtering is programmed in pmu_hbt_lbt_sel and pmu_evict_state_sel</p> <p><b>0x08</b> PMU_HN_DIR_SNOOPS_SENT_EVENT; counts number of directed snoops sent (not including SF back invalidation). Filtering is programmed in pmu_snp_vc_sel</p> <p><b>0x09</b> PMU_HN_BRD_SNOOPS_SENTEVENT; counts number of multicast snoops send (not including SF back invalidation). Filtering is programmed in pmu_snp_vc_sel</p> <p><b>0x0A</b> PMU_HN_SLC_EVICTION_EVENT; counts number of SLC evictions for specific SLC state. Filtering is programmed in pmu_hbt_lbt_sel and pmu_evict_state_sel</p>	RW	0x00

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>HN-F PMU Event 0 select</p> <p><b>0x0B</b> PMU_HN_SLC_FILL_INVALID_WAY_EVENT; counts number of SLC fills to an invalid way. Filtering is programmed in pmu_enhanced_hbt_lbt_sel</p> <p><b>0x0C</b> PMU_HN_MC_RETRY_LOCAL_EVENT; counts number of local retried transactions by the MC. Filtering is programmed in pmu_sn_home_sel</p> <p><b>0x0D</b> PMU_HN_MC_REQS_LOCAL_EVENT; counts number of local requests sent to MC. Filtering is programmed in pmu_sn_home_sel</p> <p><b>0x0E</b> PMU_HN_QOS_HH_RETRY_EVENT; counts number of times a HighHigh priority request is protocol retried at the HN-F</p> <p><b>0x0F</b> PMU_HN_POCQ_OCCUPANCY_EVENT; counts the POCQ occupancy in HN-F; occupancy filtering is programmed in pmu_occup1_id</p> <p><b>0x10</b> PMU_HN_POCQ_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation. Filterfing is programmed in pmu_enhanced_hbt_lbt_sel</p> <p><b>0x11</b> PMU_HN_POCQ_ATOMICS_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation for atomic operations . Filterfing is programmed in pmu_enhanced_hbt_lbt_sel</p> <p><b>0x14</b> PMU_HN_TXDAT_STALL_EVENT; counts number of times HN-F has a pending TXDAT flit but no credits to upload</p> <p><b>0x15</b> PMU_HN_RXRSP_STALL_EVENT; counts number of times HN-F has a pending TXRSP flit but no credits to upload</p> <p><b>0x17</b> PMU_HN_SEQ_HIT_EVENT; counts number of times a request in SLC hit a pending SF eviction in SEQ</p> <p><b>0x18</b> PMU_HN_SNP_SENT_EVENT; counts number of snoops sent including directed/multicast/SF back invalidation. Filtering is programmed in pmu.snp_vc_sel</p> <p><b>0x19</b> PMU_HN_SFBI_DIR_SNP_SENT_EVENT; counts number of times directed snoops were sent due to SF back invalidation. Filtering is programmed in pmu.snp_vc_sel</p> <p><b>0x1a</b> PMU_HN_SFBI_BRD_SNP_SENT_EVENT; counts number of times multicast snoops were sent due to SF back invalidation. Filtering is programmed in pmu.snp_vc_sel</p> <p><b>0x1b</b> Reserved</p>	RW	0x00

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>HN-F PMU Event 0 select</p> <p><b>0x1c</b> PMU_HN_INTV_DIRTY_EVENT; counts number of times SF back invalidation resulted in dirty line intervention from the RN</p> <p><b>0x1f</b> PMU_HN_SNP_FWDED_EVENT; counts number of times data forward snoops sent. Filtering is programmed in pmu_snp_vc_sel</p> <p><b>0x21</b> PMU_HN_MPAM_REQ_OVER_HARDLIM_EVENT; counts number of times write req can't allocate in SLC due to being over hardlimit. Filtering is programmed in cmn_hns_pmu_mpam_pardid_mask</p> <p><b>0x22</b> PMU_HN_MPAM_REQ_OVER_SOFTLIM_EVENT; counts number of times write req is above soft limit. Filtering is programmed in cmn_hns_pmu_mpam_pardid_mask</p> <p><b>0x23</b> PMU_HN_SNP_SENT_CLUSTER_EVENT; counts number of snoops sent to clusters excluding individual snoops within a cluster. Filtering is programmed in pmu_snp_vc_sel</p> <p><b>0x24</b> PMU_HN_SF_IMPRECISE_EVICT_EVENT; counts number of times an evict op was dropped due to SF clustering. Filtering is programmed in pmu_hbt_lbt_sel</p> <p><b>0x25</b> PMU_HN_SF_EVICT_SHARED_LINE_EVENT; counts number of times a shared line was evicted from SF</p> <p><b>0x26</b> 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</p> <p><b>0x27</b> PMU_HN_POCQ_CLASS_RETRY_EVENT; counts number of retried requests for a given class; Class filtering is programmed in pmu_class_occup_id</p> <p><b>0x28</b> 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</p>	RW	0x00

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>HN-F PMU Event 0 select</p> <p><b>0x2A</b> 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</p> <p><b>0x2B</b> 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</p> <p><b>0x2C</b> PMU_HN_SF_PRECISE_TO_IMPRECISE_EVENT; counts when number sharers exceeds how many RN's could be precisely tracked in SF</p> <p><b>0x2D</b> PMU_HN_SNP_INTV_CLN_EVENT; counts the number of times clean data intervened for a snoop request</p> <p><b>0x2E</b> PMU_HN_NC_EXCL_EVENT; counts the number of times non-cacheable exclusive request arrived at HNF</p> <p><b>0x2F</b> Reserved</p> <p><b>0x30</b> PMU_HN_SNP_REQ_RECVD_EVENT; counts number of incoming snoop requests</p> <p><b>0x31</b> PMU_HN_SNP_REQ_BYP_POCQ_EVENT; counts number of times incoming snoop request bypass Snoop Queue and allocates in POCQ</p> <p><b>0x32</b> PMU_HN_DIR_CCGHA_SNP_SENT_EVENT; counts number of directed snoops sent to CCG HA. (SFBI and non-SFBI included).</p> <p><b>0x33</b> PMU_HN_BRD_CCGHA_SNP_SENT_EVENT; counts number of broadcast snoops sent to CCG HA. (SFBI and non-SFBI included).</p> <p><b>0x35</b> 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</p> <p><b>0x36</b> 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</p>	RW	0x00

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>HN-F PMU Event 0 select</p> <p><b>0x37</b> PMU_HN_SF_REUPDATE_EVENT; counts number of times SF needs to be re-updated</p> <p><b>0x38</b> PMU_HN_EXCL_SF_IMPRECISE_EVENT; counts number of times exclusive req hits SF in imprecise mode. Filtering is programmed in pmu_hbt_lbt_sel</p> <p><b>0x39</b> PMU_HN_SNP_POCQ_ADDRHAZ_EVENT; counts number of POCQ address hazards involving external snoop requests at allocation</p> <p><b>0x3A</b> PMU_HN_MC_RETRY_REMOTE_EVENT; counts number of remote retried transactions by the MC. Filtering is programmed in pmu_sn_home_sel</p> <p><b>0x3B</b> PMU_HN_MC_REQS_REMOTE_EVENT; counts number of remote requests sent to MC. Filtering is programmed in pmu_sn_home_sel</p> <p><b>0x3C</b> 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> <p><b>0x3D</b> PMU_HN_READONCE_HAZARD_DETECTED_EVENT; Dependent RdOnce or RNSD eligible for forwarding. Filtering is programmed in pmu_enhanced_hbt_lbt_sel</p> <p><b>0x3E</b> PMU_HN_READONCE_FWD_DATA_COMPLETED_EVENT; Dependent RdOnce or RNSD entry got forwarded data. Filtering is programmed in pmu_enhanced_hbt_lbt_sel</p> <p><b>0x40</b> PMU_HN_CBUSY00_EVENT; Monitor CBUSY 00 is executed. Filtering is programmed in pmu_cbusy_snthrottle_sel</p> <p><b>0x41</b> PMU_HN_CBUSY01_EVENT; Monitor CBUSY 01 is executed. Filtering is programmed in pmu_cbusy_snthrottle_sel</p> <p><b>0x42</b> PMU_HN_CBUSY10_EVENT; Monitor CBUSY 10 is executed. Filtering is programmed in pmu_cbusy_snthrottle_sel</p> <p><b>0x43</b> PMU_HN_CBUSY11_EVENT; Monitor CBUSY 11 is executed. Filtering is programmed in pmu_cbusy_snthrottle_sel</p> <p><b>0x44</b> PMU_HN_RO_RNSD_NEW_ALLOC_HINT_EVENT; Monitor the new hint/special requests. Filtering is programmed in pmu_enhanced_hbt_lbt_sel</p>	RW	0x00

### 8.3.9.138 cmn\_hns\_pmu\_mpam\_sel

Specifies details of MPAM event to be counted

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD908

##### Type

RW

##### Reset value

See individual bit resets

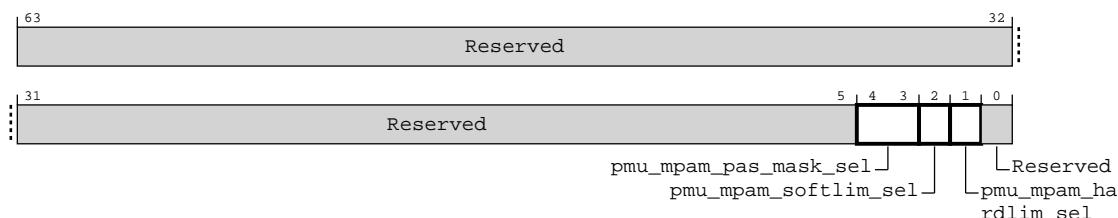
##### Usage constraints

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 8-549: cmn\_hns\_pmu\_mpam\_sel**



**Table 8-555: cmn\_hns\_pmu\_mpam\_sel attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[4:3]	pmu_mpam_pas_mask_sel	PAS select for PARTID Mask <b>0b00</b> PMU MPAM mask is for Secure MPAMID. <b>0b01</b> PMU MPAM mask is for Non-Secure MPAMID. <b>0b10</b> PMU MPAM mask is for Root MPAMID. <b>0b11</b> PMU MPAM mask is for Realm MPAMID.	RW	0b01
[2]	pmu_mpam_softlim_sel	When set, HN-F PMU MPAM Softlimit count is filtered for specific PARTIDs <b>0b0</b> PMU Softlimit count is total for all PARDIDs. <b>0b1</b> PMU Softlimit count is only for PARDIDs indicated in fliter register	RW	0b0
[1]	pmu_mpam_hardlim_sel	When set, HN-F PMU MPAM Hardlimit count is filtered for specific PARTIDs <b>0b0</b> PMU Hardlimit count is total for all PARDIDs. <b>0b1</b> PMU Hardlimit count is only for PARDIDs indicated in fliter register	RW	0b0
[0]	Reserved	Reserved	RO	

### 8.3.9.139 cmn\_hns\_amevcntr0-4

There are 5 iterations of this register. The index ranges from 0 to 4. AMU event counter register #{{index}}

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC900 + #{{8\*index}}

##### Type

RW

##### Reset value

See individual bit resets

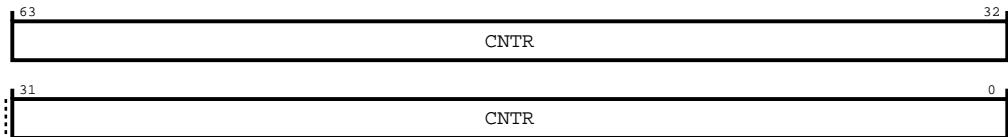
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-550: cmn\_hns\_amevcntr0-4**



**Table 8-556: cmn\_hns\_amevcntr0-4 attributes**

Bits	Name	Description	Type	Reset
[63:0]	CNTR	Monitor value	RW	0b0

## 8.3.9.140 cmn\_hns\_amevtyper0-4

There are 5 iterations of this register. The index ranges from 0 to 4. AMU event type register #{index}

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xCD00 + #{4\*index}

#### Type

RO

#### Reset value

See individual bit resets

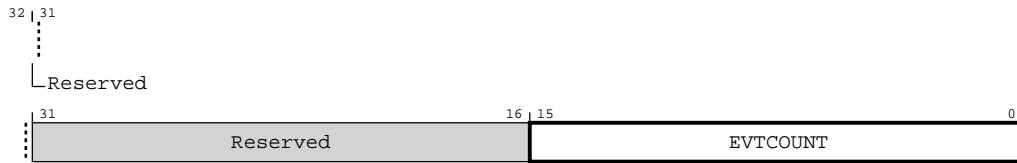
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-551: cmn\_hns\_amevtyper0-4**



**Table 8-557: cmn\_hns\_amevtyper0-4 attributes**

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RO	-
[15:0]	EVTCOUNT	Event type	RO	0-4

### 8.3.9.141 cmn\_hns\_amcntset

AMU count enable set register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD500

##### Type

W1S

##### Reset value

See individual bit resets

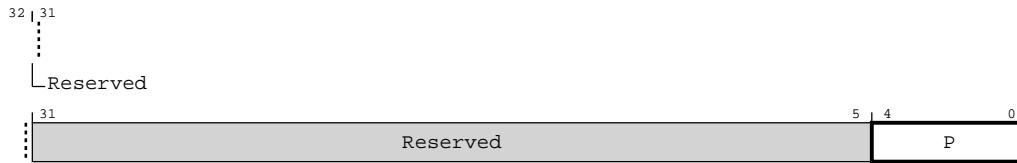
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-552: cmn\_hns\_amcntensem**



**Table 8-558: cmn\_hns\_amcntensem attributes**

Bits	Name	Description	Type	Reset
[31:5]	Reserved	Reserved	RO	-
[4:0]	P	AMEVCNTR<m> enable	W1S	Configuration dependent

### 8.3.9.142 cmn\_hns\_amcntenclr

AMU count enable clear register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD520

##### Type

W1C

##### Reset value

See individual bit resets

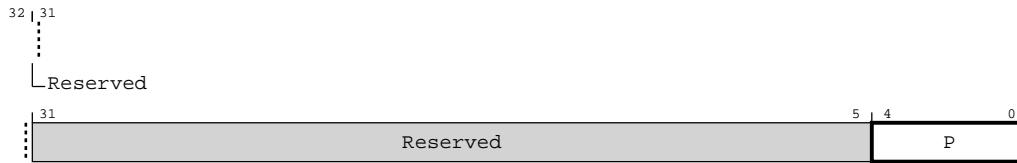
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-553: cmn\_hns\_amcntenclr**



**Table 8-559: cmn\_hns\_amcntenclr attributes**

Bits	Name	Description	Type	Reset
[31:5]	Reserved	Reserved	RO	-
[4:0]	P	AMEVCNTR<m> disable	W1C	Configuration dependent

### 8.3.9.143 cmn\_hns\_amcfg

AMU configuration register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD700

##### Type

RO

##### Reset value

See individual bit resets

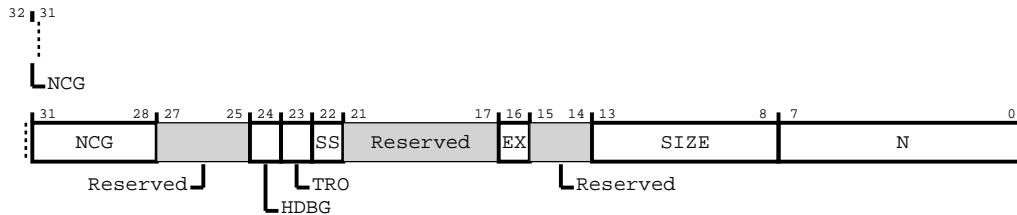
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-554: cmn\_hns\_amcfg**



**Table 8-560: cmn\_hns\_amcfg attributes**

Bits	Name	Description	Type	Reset
[31:28]	NCG	Monitor groups not implemented	RO	0x0
[27:25]	Reserved	Reserved	RO	
[24]	HDBG	Halt-on-debug feature not supported	RO	0x0
[23]	TRO	Trace features not supported	RO	0x0
[22]	SS	Snapshot not supported	RO	0x0
[21:17]	Reserved	Reserved	RO	
[16]	EX	Export not supported	RO	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	SIZE	Monitor size, 64 bit monitors supported	RO	0x3f
[7:0]	N	Number of monitors minor one, 5 monitors supported	RO	0x04

### 8.3.9.144 cmn\_hns\_amcr

AMU control register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD704

##### Type

RW

##### Reset value

See individual bit resets

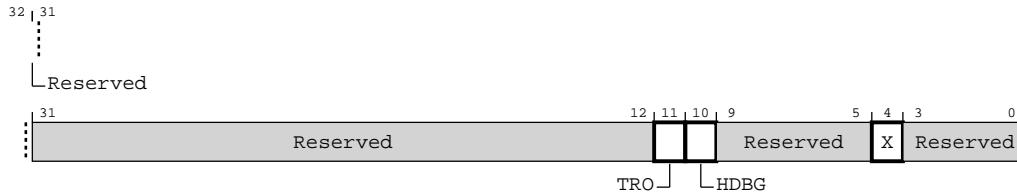
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-555: cmn\_hns\_amcr**



**Table 8-561: cmn\_hns\_amcr attributes**

Bits	Name	Description	Type	Reset
[31:12]	Reserved	Reserved	RO	
[11]	TRO	Trace enable, <b>RES0</b>	RW	0x0
[10]	HDBG	Halt-on-debug, <b>RES0</b>	RW	0x0
[9:5]	Reserved	Reserved	RO	
[4]	X	Export enable, <b>RES0</b>	RW	0x0
[3:0]	Reserved	Reserved	RO	

## 8.3.9.145 cmn\_hns\_amiidr

AMU implementation identification register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xD708

#### Type

RO

#### Reset value

See individual bit resets

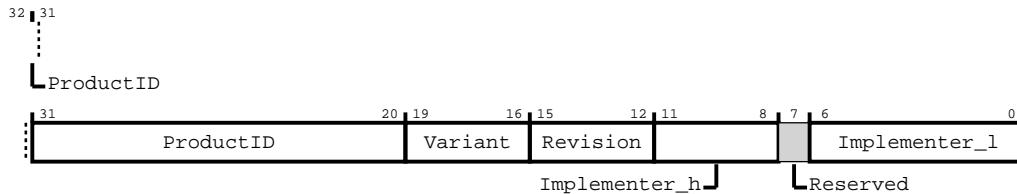
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-556: cmn\_hns\_amiidr**



**Table 8-562: cmn\_hns\_amiidr attributes**

Bits	Name	Description	Type	Reset
[31:20]	ProductID	Part number, bits[11:0]	RO	0x0
[19:16]	Variant	Component major revision	RO	0x0
[15:12]	Revision	Component minor revision	RO	0x0
[11:8]	Implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	Implementer_l	Implementer[6:0]	RO	0x3b

### 8.3.9.146 cmn\_hns\_amdevarch

AMU device architecture register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8BC

##### Type

RO

##### Reset value

See individual bit resets

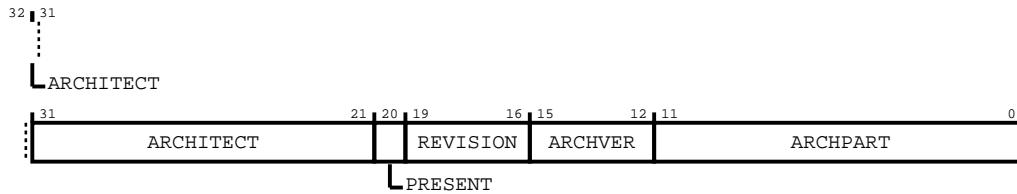
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-557: cmn\_hns\_amdevarch**



**Table 8-563: cmn\_hns\_amdevarch attributes**

Bits	Name	Description	Type	Reset
[31:21]	ARCHITECT	Architect	RO	0x23b
[20]	PRESENT	DEVARCH present	RO	0x1
[19:16]	REVISION	Revision	RO	0x0
[15:12]	ARCHVER	Architecture version	RO	0x0
[11:0]	ARCHPART	Architecture part	RO	0xa66

### 8.3.9.147 cmn\_hns\_amdevid

AMU device configuration register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8C8

##### Type

RO

##### Reset value

See individual bit resets

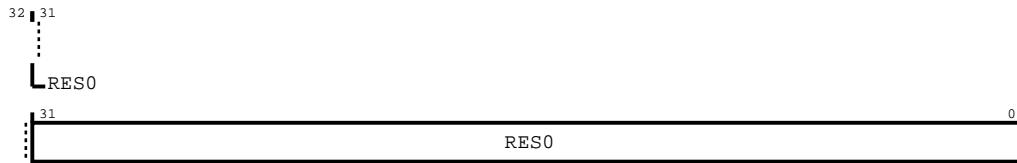
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-558: cmn\_hns\_amdevid**



**Table 8-564: cmn\_hns\_amdevid attributes**

Bits	Name	Description	Type	Reset
[31:0]	RES0	RAZ/WI	RO	0x0

### 8.3.9.148 cmn\_hns\_amdevtype

AMU device type register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8CC

##### Type

RO

##### Reset value

See individual bit resets

##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-559: cmn\_hns\_amdevtype**



**Table 8-565: cmn\_hns\_amdevtype attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	SUB	Component sub-type	RO	0x1
[3:0]	MAJOR	Component major type	RO	0x6

### 8.3.9.149 cmn\_hns\_ampidr4

AMU peripheral identification register 4

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8D0

##### Type

RO

##### Reset value

See individual bit resets

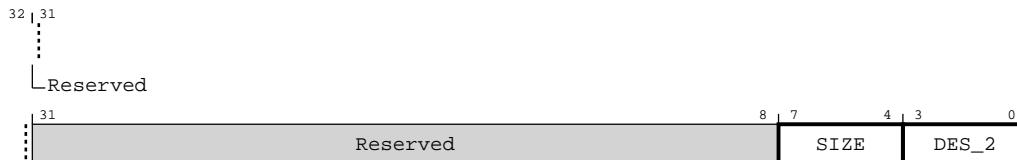
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-560: cmn\_hns\_ampidr4**



**Table 8-566: cmn\_hns\_ampidr4 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	SIZE	Size of the component	RO	0x0
[3:0]	DES_2	Designer, JEP106 continuation code	RO	0x4

### 8.3.9.150 cmn\_hns\_ampidr5-7

There are 3 iterations of this register. The index ranges from 5 to 7. AMU peripheral identification register #{{index}}

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8C0 + #{{4\*index}}

##### Type

RO

##### Reset value

See individual bit resets

##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-561: cmn\_hns\_ampidr5-7**



**Table 8-567: cmn\_hns\_ampidr5-7 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:0]	RES0	RAZ/WI	RO	0x0

### 8.3.9.151 cmn\_hns\_ampidr0

AMU peripheral identification register 0

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8E0

##### Type

RO

##### Reset value

See individual bit resets

##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-562: cmn\_hns\_ampidr0**



**Table 8-568: cmn\_hns\_ampidr0 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:0]	PART_0	Part number, bits[7:0]	RO	0x0

### 8.3.9.152 cmn\_hns\_ampidr1

AMU peripheral identification register 1

#### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xD8E4

### Type

RO

### Reset value

See individual bit resets

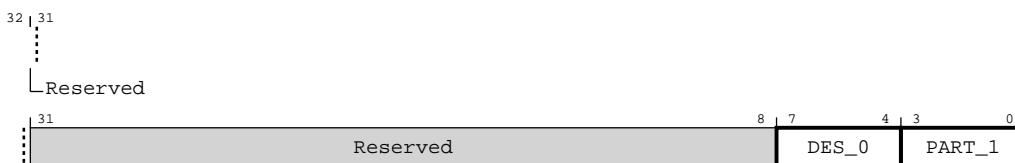
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-563: cmn\_hns\_ampidr1**



**Table 8-569: cmn\_hns\_ampidr1 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	DES_0	Designer, JEP106 identification code, bits[3:0]	RO	0xb
[3:0]	PART_1	Part number, bits[11:8]	RO	0x0

## 8.3.9.153 cmn\_hns\_ampidr2

AMU peripheral identification register 2

### Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xD8E8

### Type

RO

### Reset value

See individual bit resets

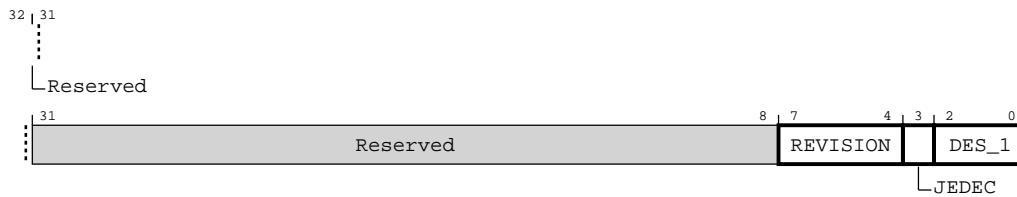
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-564: cmn\_hns\_ampidr2**



**Table 8-570: cmn\_hns\_ampidr2 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	REVISION	Component major revision	RO	0x0
[3]	JEDEC	JEDEC-assigned JEP106 implementer code is used	RO	0x1
[2:0]	DES_1	Designer, JEP106 identification code, bits[6:4]	RO	0x3

### 8.3.9.154 cmn\_hns\_ampidr3

AMU peripheral identification register 3

### Configurations

This register is available in all configurations.

### Attributes

#### Width

32

#### Address offset

0xD8EC

#### Type

RO

#### Reset value

See individual bit resets

## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-565: cmn\_hns\_ampidr3**



**Table 8-571: cmn\_hns\_ampidr3 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:4]	REVAND	Component minor revision	RO	0x0
[3:0]	CMOD	Customer modified	RO	0x0

## 8.3.9.155 cmn\_hns\_amcidr0

AMU component identification register 0

## Configurations

This register is available in all configurations.

## Attributes

### Width

32

### Address offset

0xD8F0

### Type

RO

### Reset value

See individual bit resets

## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-566: cmn\_hns\_amcidr0**



**Table 8-572: cmn\_hns\_amcidr0 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	-
[7:0]	PRMBL_0	Component identification preamble, segment 0	RO	Configuration dependent

### 8.3.9.156 cmn\_hns\_amcidr1

AMU component identification register 1

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8F4

##### Type

RO

##### Reset value

See individual bit resets

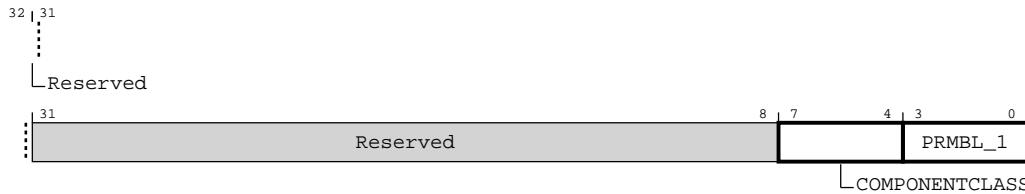
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-567: cmn\_hns\_amcidr1**



**Table 8-573: cmn\_hns\_amcidr1 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	-
[7:4]	COMPONENTCLASS	Component class	RO	Configuration dependent
[3:0]	PRMBL_1	Component identification preamble, segment 1	RO	Configuration dependent

### 8.3.9.157 cmn\_hns\_amcidr2

AMU component identification register 2

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8F8

##### Type

RO

##### Reset value

See individual bit resets

##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-568: cmn\_hns\_amcidr2**



**Table 8-574: cmn\_hns\_amcidr2 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	-
[7:0]	PRMBL_2	Component identification preamble, segment 2	RO	Configuration dependent

### 8.3.9.158 cmn\_hns\_amcidr3

AMU component identification register 3

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

32

##### Address offset

0xD8FC

##### Type

RO

##### Reset value

See individual bit resets

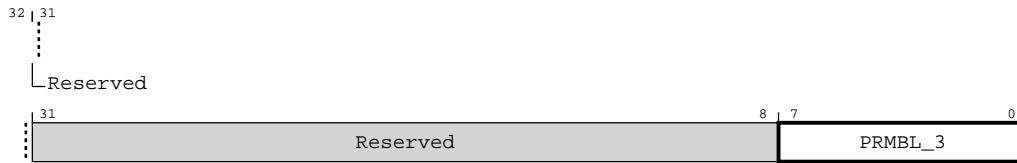
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-569: cmn\_hns\_amcidr3**



**Table 8-575: cmn\_hns\_amcidr3 attributes**

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	-
[7:0]	PRMBL_3	Component identification preamble, segment 3	RO	Configuration dependent

### 8.3.9.159 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}:#{64index}] filter for MPM PMU events

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD910 + #{8\*index}

##### index(0)

0x0 + (POR\_CHI\_MPAM\_ENABLE\_PARAM == 1)

##### index(1)

0x0 + (HNS\_MPAM\_NS\_PARTID\_MAX\_PARAM > 64) ||  
(HNS\_MPAM\_S\_PARTID\_MAX\_PARAM > 64)

##### index(2-3)

0x0 + (HNS\_MPAM\_NS\_PARTID\_MAX\_PARAM > 128) ||  
(HNS\_MPAM\_S\_PARTID\_MAX\_PARAM > 128)

##### index(4-7)

0x0 + (HNS\_MPAM\_NS\_PARTID\_MAX\_PARAM > 256) ||  
(HNS\_MPAM\_S\_PARTID\_MAX\_PARAM > 256)

##### Type

RW

### Reset value

See individual bit resets

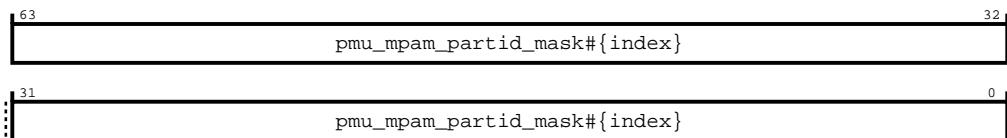
### Usage constraints

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 8-570: cmn\_hns\_pmu\_mpam\_pardid\_mask0-7**



**Table 8-576: cmn\_hns\_pmu\_mpam\_pardid\_mask0-7 attributes**

Bits	Name	Description	Type	Reset
[63:0]	pmu_mpam_partid_mask#{index}	<p>MPAM PMU hardlimit and softlimit mask for PARTID [#{64(index+1)-1}:#{64index}]</p> <p><b>0b0</b> PARTID specified is not counted in PMU count.</p> <p><b>0b1</b> PARTID specified is counted in PMU count.</p> <p><b>Note</b> This mask is used only when <code>cmn_hns_pmu_mpam_sel</code> is set for PARTID based counting.</p>	RW	0b0

### 8.3.9.160 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

`0x3C00 + #{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

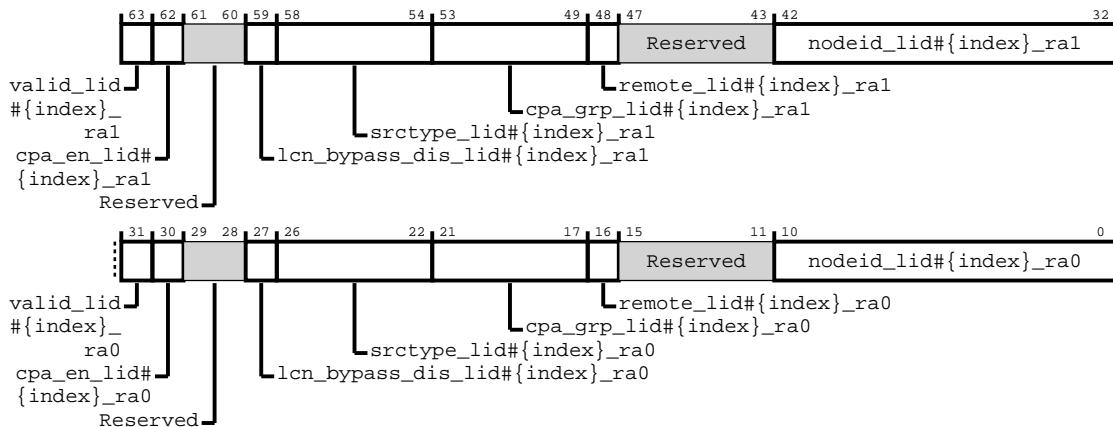
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-571: cmn\_hns\_rn\_cluster0-63\_physid\_reg0**



**Table 8-577: cmn\_hns\_rn\_cluster0-63\_physid\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra1	<p>Specifies whether the RN is valid</p> <p><b>0b0</b> RN ID is not valid</p> <p><b>0b1</b> RN ID is pointing to a valid CHI device</p>	RW	0x0

Bits	Name	Description	Type	Reset
[62]	cpa_en_lid#{index}_ra1	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[61:60]	Reserved	Reserved	RO	-
[59]	lcn_bypass_dis_lid#{index}_ra1	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra1 = 1	RW	0x0
[58:54]	srctype_lid#{index}_ra1	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[53:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[48]	remote_lid#{index}_ra1	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra1	Specifies the node ID	RW	0x0
[31]	valid_lid#{index}_ra0	Specifies whether the RN is valid  <b>0b0</b> RN ID is not valid  <b>0b1</b> RN ID is pointing to a valid CHI device	RW	0x0

Bits	Name	Description	Type	Reset
[30]	cpa_en_lid#{index}_ra0	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[29:28]	Reserved	Reserved	RO	-
[27]	lcn_bypass_dis_lid#{index}_ra0	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra0 = 1	RW	0x0
[26:22]	srctype_lid#{index}_ra0	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[21:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[16]	remote_lid#{index}_ra0	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	0x0

### 8.3.9.161 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x3C00 + #{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

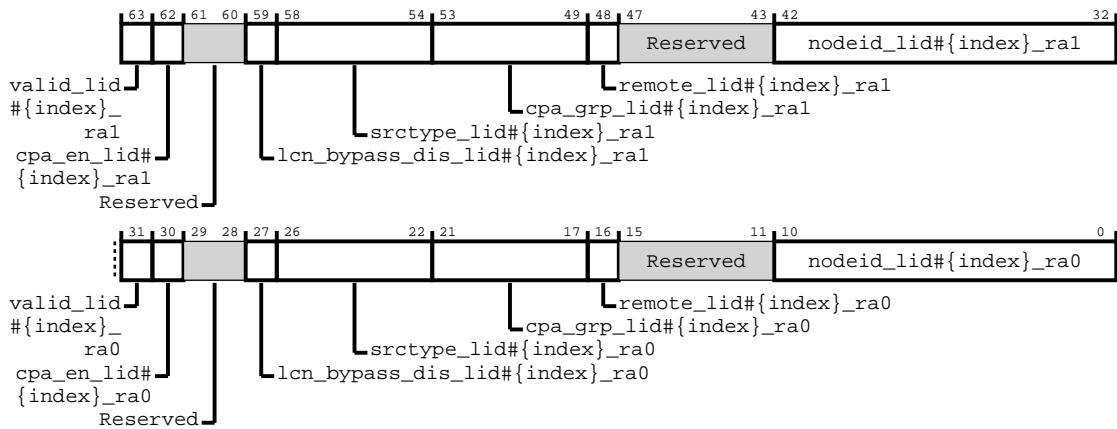
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-572: cmn\_hns\_rn\_cluster64-127\_physid\_reg0**



**Table 8-578: cmn\_hns\_rn\_cluster64-127\_physid\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra1	Specifies whether the RN is valid  <b>0b0</b> RN ID is not valid  <b>0b1</b> RN ID is pointing to a valid CHI device	RW	0x0
[62]	cpa_en_lid#{index}_ra1	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[61:60]	Reserved	Reserved	RO	-
[59]	lcn_bypass_dis_lid#{index}_ra1	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra1 = 1	RW	0x0

Bits	Name	Description	Type	Reset
[58:54]	srctype_lid#{index}_ra1	<p>Specifies the CHI source type of the RN</p> <p><b>0b01000</b> HN-S</p> <p><b>0b01010</b> 256 bit CHI-B RN-F</p> <p><b>0b01011</b> 256 bit CHI-C RN-F</p> <p><b>0b01100</b> 256 bit CHI-D RN-F</p> <p><b>0b01101</b> 256 bit CHI-E RN-F</p> <p><b>0b10000</b> 256 bit CHI-F RN-F</p> <p><b>0b10001</b> 256 bit CHI-G RN-F</p> <p><b>Others</b> Reserved</p>	RW	0b00000
[53:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[48]	remote_lid#{index}_ra1	<p>Specifies whether the RN is remote or local</p> <p><b>0b0</b> Local RN</p> <p><b>0b1</b> Remote RN</p>	RW	0x0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra1	Specifies the node ID	RW	0x0
[31]	valid_lid#{index}_ra0	<p>Specifies whether the RN is valid</p> <p><b>0b0</b> RN ID is not valid</p> <p><b>0b1</b> RN ID is pointing to a valid CHI device</p>	RW	0x0
[30]	cpa_en_lid#{index}_ra0	<p>Specifies whether the CCIX port aggregation is enabled</p> <p><b>0b0</b> CPA not enabled</p> <p><b>0b1</b> CPA enabled</p>	RW	0x0
[29:28]	Reserved	Reserved	RO	-
[27]	lcn_bypass_dis_lid#{index}_ra0	<p>Specifies whether transactions through HA bypasses LCN or not</p> <p><b>0b0</b> Bypass LCN</p> <p><b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra0 = 1</p>	RW	0x0

Bits	Name	Description	Type	Reset
[26:22]	srctype_lid#{index}_ra0	Specifies the CHI source type of the RN <b>0b01000</b> HN-S <b>0b01010</b> 256 bit CHI-B RN-F <b>0b01011</b> 256 bit CHI-C RN-F <b>0b01100</b> 256 bit CHI-D RN-F <b>0b01101</b> 256 bit CHI-E RN-F <b>0b10000</b> 256 bit CHI-F RN-F <b>0b10001</b> 256 bit CHI-G RN-F <b>Others</b> Reserved	RW	0b00000
[21:17]	cpx_grp_lid#{index}_ra0	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[16]	remote_lid#{index}_ra0	Specifies whether the RN is remote or local <b>0b0</b> Local RN <b>0b1</b> Remote RN	RW	0x0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	0x0

### 8.3.9.162 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x3C08 + #{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

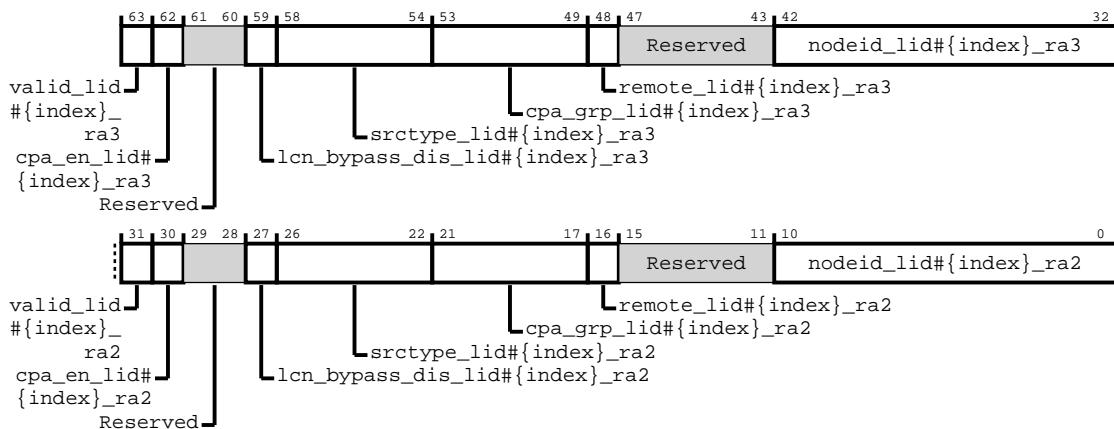
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-573: cmn\_hns\_rn\_cluster0-127\_physid\_reg1**



**Table 8-579: cmn\_hns\_rn\_cluster0-127\_physid\_reg1 attributes**

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra3	<p>Specifies whether the RN is valid</p> <p><b>0b0</b> RN ID is not valid</p> <p><b>0b1</b> RN ID is pointing to a valid CHI device</p>	RW	0x0

Bits	Name	Description	Type	Reset
[62]	cpa_en_lid#{index}_ra3	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[61:60]	Reserved	Reserved	RO	-
[59]	lcn_bypass_dis_lid#{index}_ra3	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra3 = 1	RW	0x0
[58:54]	srctype_lid#{index}_ra3	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[53:49]	cpa_grp_lid#{index}_ra3	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[48]	remote_lid#{index}_ra3	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra3	Specifies the node ID	RW	0x0
[31]	valid_lid#{index}_ra2	Specifies whether the RN is valid  <b>0b0</b> RN ID is not valid  <b>0b1</b> RN ID is pointing to a valid CHI device	RW	0x0

Bits	Name	Description	Type	Reset
[30]	cpa_en_lid#{index}_ra2	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[29:28]	Reserved	Reserved	RO	-
[27]	lcn_bypass_dis_lid#{index}_ra2	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra2 = 1	RW	0x0
[26:22]	srctype_lid#{index}_ra2	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[21:17]	cpa_grp_lid#{index}_ra2	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[16]	remote_lid#{index}_ra2	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra2	Specifies the node ID	RW	0x0

### 8.3.9.163 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x3C10 + #{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

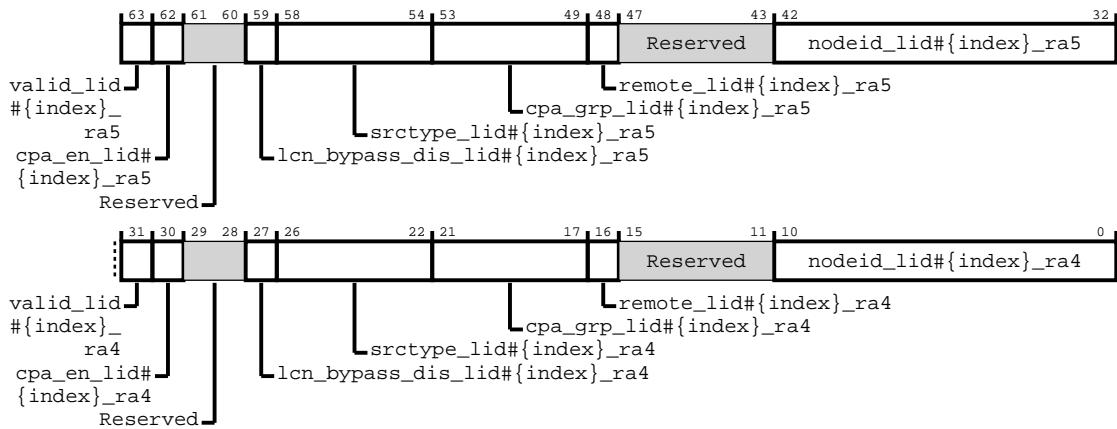
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-574: cmn\_hns\_rn\_cluster0-127\_physid\_reg2**



**Table 8-580: cmn\_hns\_rn\_cluster0-127\_physid\_reg2 attributes**

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra5	Specifies whether the RN is valid  <b>0b0</b> RN ID is not valid  <b>0b1</b> RN ID is pointing to a valid CHI device	RW	0x0
[62]	cpa_en_lid#{index}_ra5	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[61:60]	Reserved	Reserved	RO	-
[59]	lcn_bypass_dis_lid#{index}_ra5	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra5 = 1	RW	0x0

Bits	Name	Description	Type	Reset
[58:54]	srctype_lid#{index}_ra5	<p>Specifies the CHI source type of the RN</p> <p><b>0b01000</b> HN-S</p> <p><b>0b01010</b> 256 bit CHI-B RN-F</p> <p><b>0b01011</b> 256 bit CHI-C RN-F</p> <p><b>0b01100</b> 256 bit CHI-D RN-F</p> <p><b>0b01101</b> 256 bit CHI-E RN-F</p> <p><b>0b10000</b> 256 bit CHI-F RN-F</p> <p><b>0b10001</b> 256 bit CHI-G RN-F</p> <p><b>Others</b> Reserved</p>	RW	0b00000
[53:49]	cpa_grp_lid#{index}_ra5	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[48]	remote_lid#{index}_ra5	<p>Specifies whether the RN is remote or local</p> <p><b>0b0</b> Local RN</p> <p><b>0b1</b> Remote RN</p>	RW	0x0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra5	Specifies the node ID	RW	0x0
[31]	valid_lid#{index}_ra4	<p>Specifies whether the RN is valid</p> <p><b>0b0</b> RN ID is not valid</p> <p><b>0b1</b> RN ID is pointing to a valid CHI device</p>	RW	0x0
[30]	cpa_en_lid#{index}_ra4	<p>Specifies whether the CCIX port aggregation is enabled</p> <p><b>0b0</b> CPA not enabled</p> <p><b>0b1</b> CPA enabled</p>	RW	0x0
[29:28]	Reserved	Reserved	RO	-
[27]	lcn_bypass_dis_lid#{index}_ra4	<p>Specifies whether transactions through HA bypasses LCN or not</p> <p><b>0b0</b> Bypass LCN</p> <p><b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra4 = 1</p>	RW	0x0

Bits	Name	Description	Type	Reset
[26:22]	srctype_lid#{index}_ra4	<p>Specifies the CHI source type of the RN</p> <p><b>0b01000</b> HN-S</p> <p><b>0b01010</b> 256 bit CHI-B RN-F</p> <p><b>0b01011</b> 256 bit CHI-C RN-F</p> <p><b>0b01100</b> 256 bit CHI-D RN-F</p> <p><b>0b01101</b> 256 bit CHI-E RN-F</p> <p><b>0b10000</b> 256 bit CHI-F RN-F</p> <p><b>0b10001</b> 256 bit CHI-G RN-F</p> <p><b>Others</b> Reserved</p>	RW	0b00000
[21:17]	cpx_grp_lid#{index}_ra4	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[16]	remote_lid#{index}_ra4	<p>Specifies whether the RN is remote or local</p> <p><b>0b0</b> Local RN</p> <p><b>0b1</b> Remote RN</p>	RW	0x0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra4	Specifies the node ID	RW	0x0

### 8.3.9.164 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x3C18 + #{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

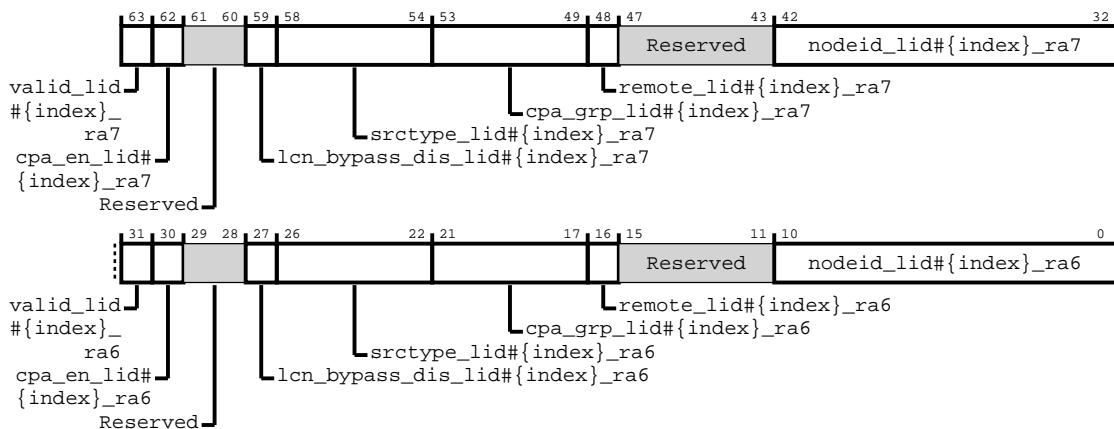
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-575: cmn\_hns\_rn\_cluster0-127\_physid\_reg3**



**Table 8-581: cmn\_hns\_rn\_cluster0-127\_physid\_reg3 attributes**

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra7	Specifies whether the RN is valid  0b0 RN ID is not valid  0b1 RN ID is pointing to a valid CHI device	RW	0x0

Bits	Name	Description	Type	Reset
[62]	cpa_en_lid#{index}_ra7	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[61:60]	Reserved	Reserved	RO	-
[59]	lcn_bypass_dis_lid#{index}_ra7	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra7 = 1	RW	0x0
[58:54]	srctype_lid#{index}_ra7	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[53:49]	cpa_grp_lid#{index}_ra7	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[48]	remote_lid#{index}_ra7	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra7	Specifies the node ID	RW	0x0
[31]	valid_lid#{index}_ra6	Specifies whether the RN is valid  <b>0b0</b> RN ID is not valid  <b>0b1</b> RN ID is pointing to a valid CHI device	RW	0x0

Bits	Name	Description	Type	Reset
[30]	cpa_en_lid#{index}_ra6	Specifies whether the CCIX port aggregation is enabled  <b>0b0</b> CPA not enabled  <b>0b1</b> CPA enabled	RW	0x0
[29:28]	Reserved	Reserved	RO	-
[27]	lcn_bypass_dis_lid#{index}_ra6	Specifies whether transactions through HA bypasses LCN or not  <b>0b0</b> Bypass LCN  <b>0b1</b> LCN bypass disabled Only applicable when remote_lid#{index}_ra6 = 1	RW	0x0
[26:22]	srctype_lid#{index}_ra6	Specifies the CHI source type of the RN  <b>0b01000</b> HN-S  <b>0b01010</b> 256 bit CHI-B RN-F  <b>0b01011</b> 256 bit CHI-C RN-F  <b>0b01100</b> 256 bit CHI-D RN-F  <b>0b01101</b> 256 bit CHI-E RN-F  <b>0b10000</b> 256 bit CHI-F RN-F  <b>0b10001</b> 256 bit CHI-G RN-F  <b>Others</b> Reserved	RW	0b00000
[21:17]	cpa_grp_lid#{index}_ra6	Specifies CCIX port aggregation group ID(0-31)	RW	0x0
[16]	remote_lid#{index}_ra6	Specifies whether the RN is remote or local  <b>0b0</b> Local RN  <b>0b1</b> Remote RN	RW	0x0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra6	Specifies the node ID	RW	0x0

### 8.3.9.165 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5000 + #{index}\*8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rsr.cfg\_ctl

##### Secure group override

cmn\_hns\_ssr.cfg\_ctl

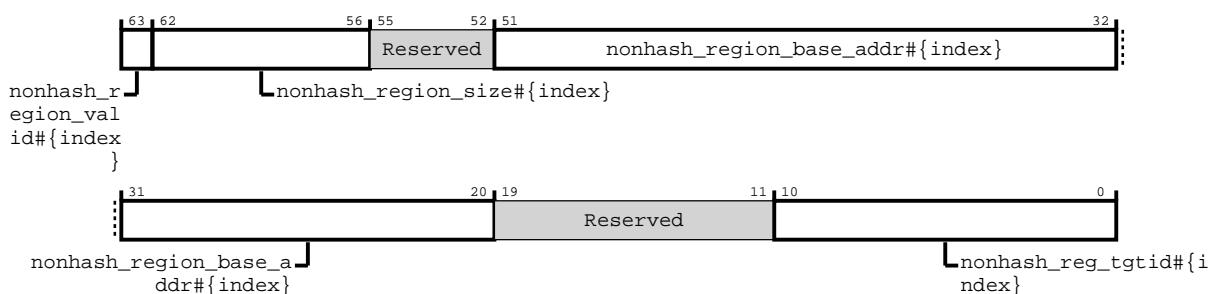
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_ssr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_ssr.cfg\_ctl bit and cmn\_hns\_rsr.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 8-576: cmn\_hns\_sam\_nonhash\_cfg1\_memregion2-63**



**Table 8-582: cmn\_hns\_sam\_nonhash\_cfg1\_memregion2-63 attributes**

Bits	Name	Description	Type	Reset
[63]	nonhash_region_valid#{index}	valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0
[62:56]	nonhash_region_size#{index}	Memory region size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b0
[55:52]	Reserved	Reserved	RO	
[51:20]	nonhash_region_base_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[19:11]	Reserved	Reserved	RO	
[10:0]	nonhash_reg_tgtid#{index}	SN TgtID for the non-hashed region	RW	0x0

### 8.3.9.166 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5200 + #{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

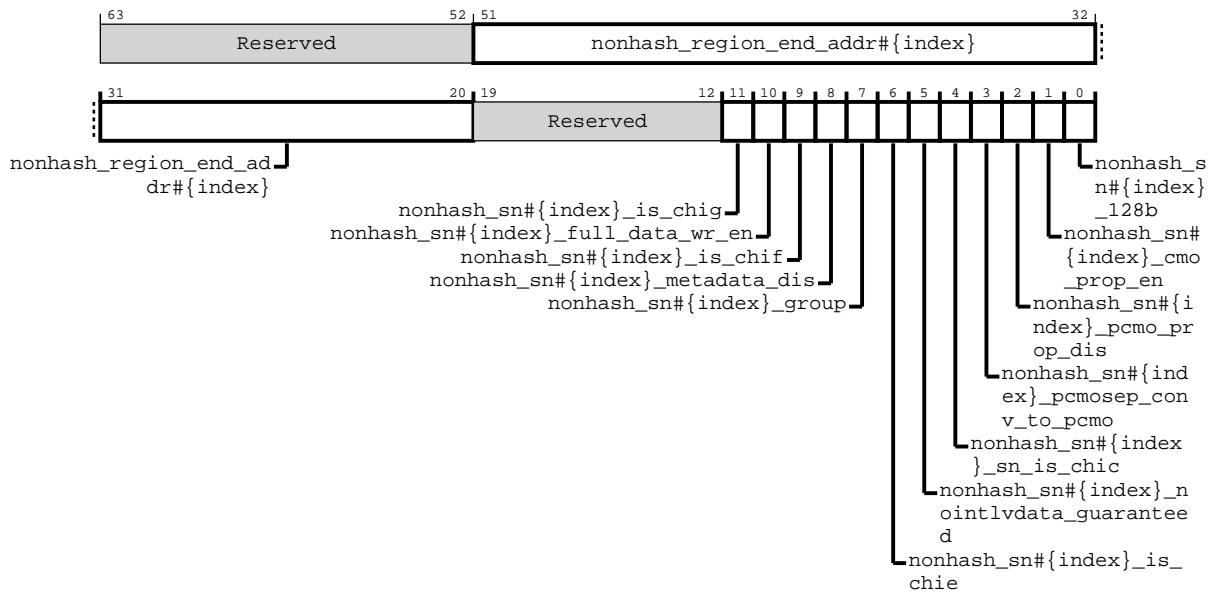
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-577: cmn\_hns\_sam\_nonhash\_cfg2\_memregion2-63**



**Table 8-583: 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	0x0
[19:12]	Reserved	Reserved	RO	
[11]	nonhash_sn#{index}_is_chig	nonhash SN #{index} supports CHI-G	RW	0b0
[10]	nonhash_sn#{index}_full_data_wr_en	HNS implements always sending 64B write for nonhash SN #{index} when set	RW	0b0
[9]	nonhash_sn#{index}_is_chif	nonhash SN #{index} supports CHI-F	RW	0b0
[8]	nonhash_sn#{index}_metadata_dis	HNS implements metadata termination flow for nonhash SN #{index} when set	RW	0b0

Bits	Name	Description	Type	Reset
[7]	nonhash_sn#{index}_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[6]	nonhash_sn#{index}_is_chie	nonhash SN #{index} supports CHI-E	RW	0b0
[5]	nonhash_sn#{index}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[4]	nonhash_sn#{index}_sn_is_chic	Indicates that nonhash sn is a CHI-C SN when set	RW	0b0
[3]	nonhash_sn#{index}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for nonhash SN #{index} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[2]	nonhash_sn#{index}_pcm0_prop_dis	Disables PCMO propagation for nonhash SN #{index} when set	RW	0b0
[1]	nonhash_sn#{index}_cmo_prop_en	Enables CMO propagation for nonhash SN #{index} when set	RW	0b0
[0]	nonhash_sn#{index}_128b	Data width of nonhash SN #{index}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0

### 8.3.9.167 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5400 + #{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

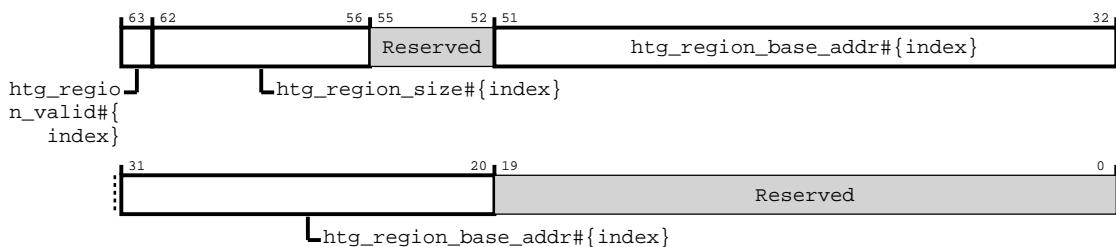
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-578: cmn\_hns\_sam\_htg\_cfg1\_memregion0-15**



**Table 8-584: cmn\_hns\_sam\_htg\_cfg1\_memregion0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	htg_region_valid#{index}	valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0
[62:56]	htg_region_size#{index}	Memory region size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[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	0x0
[19:0]	Reserved	Reserved	RO	-

### 8.3.9.168 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5480 + #{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 is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-579: cmn\_hns\_sam\_htg\_cfg2\_memregion0-15**



**Table 8-585: cmn\_hns\_sam\_htg\_cfg2\_memregion0-15 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:20]	htg_region_end_addr#[{index}]	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[19:0]	Reserved	Reserved	RO	

### 8.3.9.169 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5500 + #{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

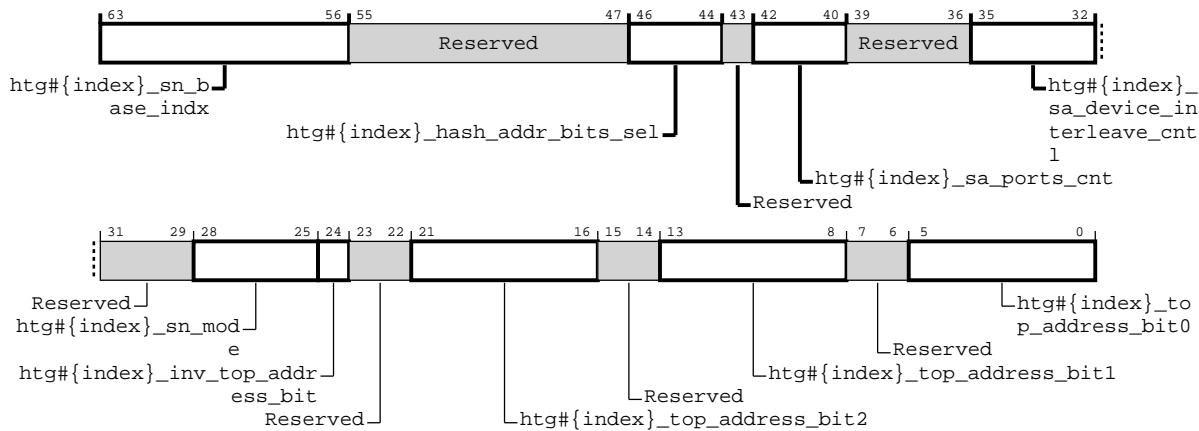
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-580: cmn\_hns\_sam\_htg\_cfg3\_memregion0-15**



**Table 8-586: cmn\_hns\_sam\_htg\_cfg3\_memregion0-15 attributes**

Bits	Name	Description	Type	Reset
[63:56]	htg#{index}_sn_base_idx	Base index for the HTG SN TgtID table	RW	0xFF
[55:47]	Reserved	Reserved	RO	
[46:44]	htg#{index}_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN)  <b>0b000</b> [16:8] address bits (Default)  <b>0b001</b> [17:9] address bits  <b>0b010</b> [18:10] address bits  <b>0b011</b> [19:11] address bits  <b>0b100</b> [20:12] address bits  <b>0b101</b> [21:13] address bits  <b>0b110</b> [22:14] address bits  <b>0b111</b> [23:15] address bits  <b>Others</b> Reserved	RW	0x0
[43]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[42:40]	htg#[index]_sa_ports_cnt	<p>Specifies the number of CXSA/CXLSA device aggregated</p> <p><b>0b000</b> 1 port</p> <p><b>0b001</b> 2 ports</p> <p><b>0b010</b> 4 ports</p> <p><b>0b011</b> 8 ports</p> <p><b>0b100</b> 16 ports</p> <p><b>0b101</b> 3 ports</p> <p><b>0b110</b> 6 ports</p> <p><b>0b111</b> 12 ports</p>	RW	0b0
[39:36]	Reserved	Reserved	RO	
[35:32]	htg#[index]_sa_device_interleave_cnt	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[31:29]	Reserved	Reserved	RO	
[28:25]	htg#[{index}_sn_mode	SN selection mode <b>0b0000</b> Reserved <b>0b0001</b> 3-SN mode (SNO, SN1, SN2) <b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5) <b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4) <b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing <b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing <b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing <b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function <b>0b1000</b> 3-SN mode (arithmetic modulo) <b>0b1001</b> 6-SN mode (arithmetic modulo) <b>Others</b> Reserved	RW	0b0
[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	0x0
[23:22]	Reserved	Reserved	RO	
[21:16]	htg#[{index}_top_address_bit2	Top address bit 2	RW	0x00
[15:14]	Reserved	Reserved	RO	
[13:8]	htg#[{index}_top_address_bit1	Top address bit 1	RW	0x00
[7:6]	Reserved	Reserved	RO	
[5:0]	htg#[{index}_top_address_bit0	Top address bit 0	RW	0x00

### 8.3.9.170 cmn\_hns\_sam\_htg\_sn\_nodeid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures SN node IDs for HTGs in the HNSAM . Controls target SN node IDs #{index4 + 0} to #{index4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

Address offset : index(0-15) : 0x5600 + #{8 \* index}

### index(16-31)

0x5800 + #{8 \* (index - 15)}

### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.cfg\_ctl

### Secure group override

cmn\_hns\_scr.cfg\_ctl

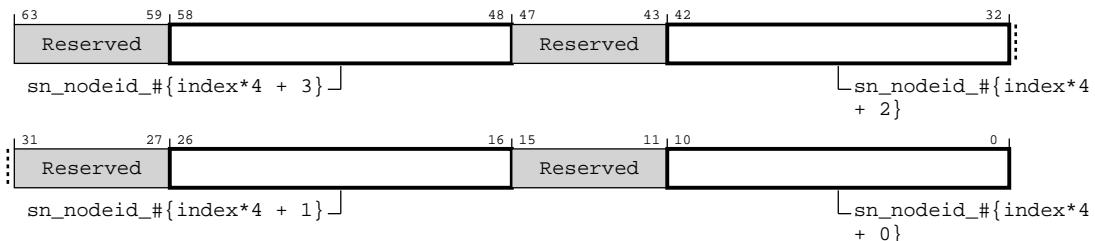
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-581: cmn\_hns\_sam\_htg\_sn\_nodeid\_reg0-31**



**Table 8-587: cmn\_hns\_sam\_htg\_sn\_nodeid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	<code>sn_nodeid_#[index*4 + 3]</code>	Hashed target SN node ID <code>#[index*4 + 3]</code>	RW	0b000000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	<code>sn_nodeid_#[index*4 + 2]</code>	Hashed target SN node ID <code>#[index*4 + 2]</code>	RW	0b000000000000
[31:27]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[26:16]	sn_nodeid_{index*4 + 1}	Hashed target SN node ID #{index*4 + 1}	RW	0b000000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_{index*4 + 0}	Hashed target SN node ID #{index*4 + 0}	RW	0b000000000000

### 8.3.9.171 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 #{index4 + 0} to #{index4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5680 + #{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

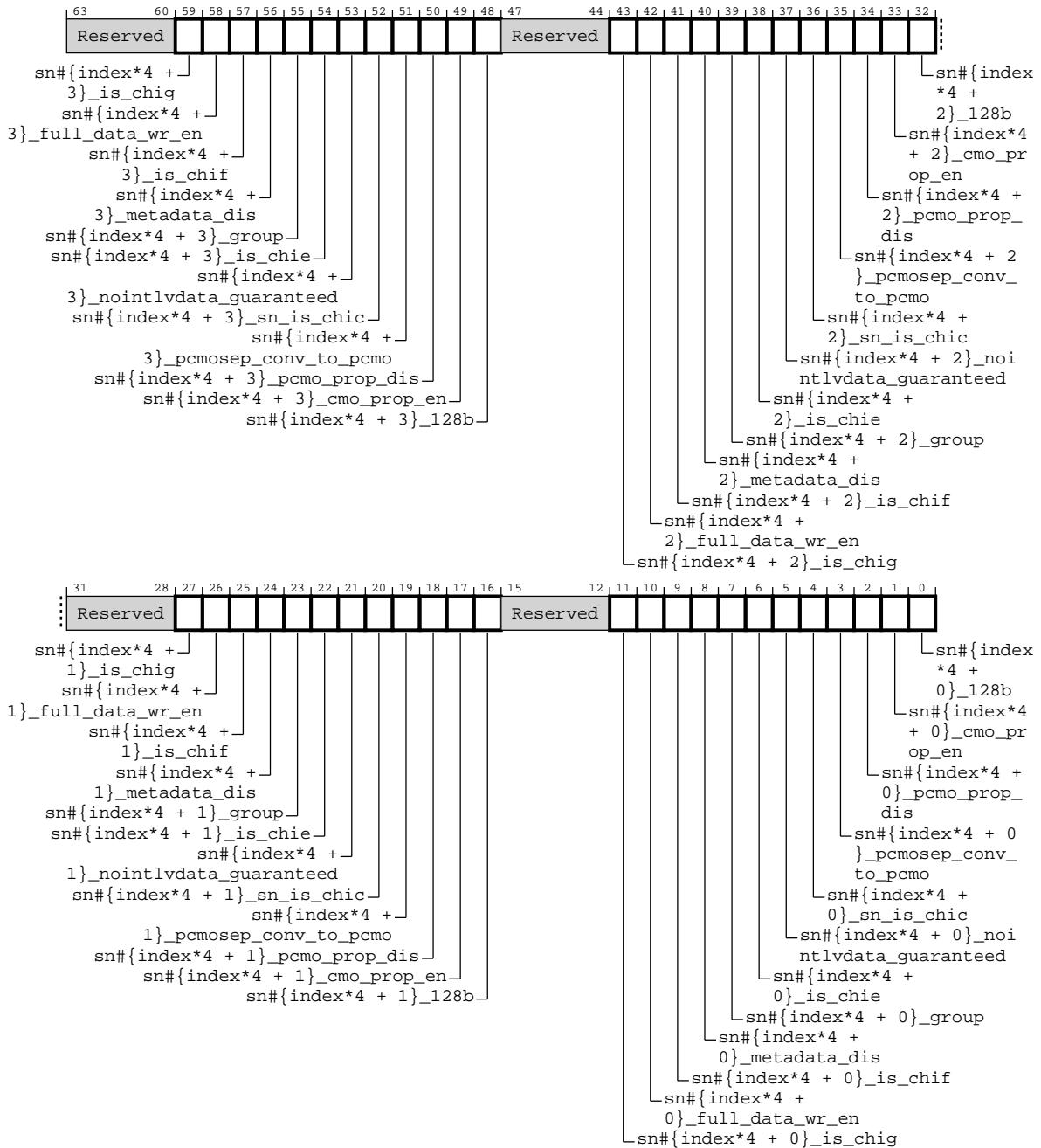
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-582: cmn\_hns\_sam\_htg\_sn\_attr0-15**



**Table 8-588: cmn\_hns\_sam\_htg\_sn\_attr0-15 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59]	sn#[index*4 + 3]_is_chig	SN #[index*4 + 3] supports CHI-G	RW	0b0
[58]	sn#[index*4 + 3]_full_data_wr_en	HNS implements always sending 64B write for SN #[index*4 + 3] when set	RW	0b0
[57]	sn#[index*4 + 3]_is_chif	SN #[index*4 + 3] supports CHI-F	RW	0b0

Bits	Name	Description	Type	Reset
[56]	sn#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 3} when set	RW	0b0
[55]	sn#{index*4 + 3}_group	Specifies the SN-F grouping <b>0b0</b> Group A <b>0b1</b> Group B	RW	0b0
[54]	sn#{index*4 + 3}_is_chie	SN #{index*4 + 3} supports CHI-E	RW	0b0
[53]	sn#{index*4 + 3}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[52]	sn#{index*4 + 3}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	0b0
[51]	sn#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 3} when set <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[50]	sn#{index*4 + 3}_pcm_o_prop_dis	Disables PCMO propagation for SN #{index*4 + 3} when set	RW	0b0
[49]	sn#{index*4 + 3}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 3} when set	RW	0b0
[48]	sn#{index*4 + 3}_128b	Data width of SN #{index*4 + 3} <b>0b1</b> 128 bits <b>0b0</b> 256 bits	RW	0b0
[47:44]	Reserved	Reserved	RO	-
[43]	sn#{index*4 + 2}_is_chig	SN #{index*4 + 2} supports CHI-G	RW	0b0
[42]	sn#{index*4 + 2}_full_data_wr_en	HNS implements always sending 64B write for SN #{index*4 + 2} when set	RW	0b0
[41]	sn#{index*4 + 2}_is_chif	SN #{index*4 + 2} supports CHI-F	RW	0b0
[40]	sn#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 2} when set	RW	0b0
[39]	sn#{index*4 + 2}_group	Specifies the SN-F grouping <b>0b0</b> Group A <b>0b1</b> Group B	RW	0b0
[38]	sn#{index*4 + 2}_is_chie	SN #{index*4 + 2} supports CHI-E	RW	0b0
[37]	sn#{index*4 + 2}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[36]	sn#{index*4 + 2}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	0b0
[35]	sn#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 2} when set <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[34]	sn#{index*4 + 2}_pcm_o_prop_dis	Disables PCMO propagation for SN #{index*4 + 2} when set	RW	0b0
[33]	sn#{index*4 + 2}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 2} when set	RW	0b0

Bits	Name	Description	Type	Reset
[32]	sn#{index*4 + 2}_128b	Data width of SN #{index*4 + 2}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[31:28]	Reserved	Reserved	RO	-
[27]	sn#{index*4 + 1}_is_chig	SN #{index*4 + 1} supports CHI-G	RW	0b0
[26]	sn#{index*4 + 1}_full_data_wr_en	HNS implements always sending 64B write for SN #{index*4 + 1} when set	RW	0b0
[25]	sn#{index*4 + 1}_is_chif	SN #{index*4 + 1} supports CHI-F	RW	0b0
[24]	sn#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 1} when set	RW	0b0
[23]	sn#{index*4 + 1}_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[22]	sn#{index*4 + 1}_is_chie	SN #{index*4 + 1} supports CHI-E	RW	0b0
[21]	sn#{index*4 + 1}_noIntlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[20]	sn#{index*4 + 1}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	0b0
[19]	sn#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 1} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[18]	sn#{index*4 + 1}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 1} when set	RW	0b0
[17]	sn#{index*4 + 1}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 1} when set	RW	0b0
[16]	sn#{index*4 + 1}_128b	Data width of SN #{index*4 + 1}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[15:12]	Reserved	Reserved	RO	-
[11]	sn#{index*4 + 0}_is_chig	SN #{index*4 + 0} supports CHI-G	RW	0b0
[10]	sn#{index*4 + 0}_full_data_wr_en	HNS implements always sending 64B write for SN #{index*4 + 0} when set	RW	0b0
[9]	sn#{index*4 + 0}_is_chif	SN #{index*4 + 0} supports CHI-F	RW	0b0
[8]	sn#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 0} when set	RW	0b0
[7]	sn#{index*4 + 0}_group	Specifies the SN-F grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[6]	sn#{index*4 + 0}_is_chie	SN #{index*4 + 0} supports CHI-E	RW	0b0

Bits	Name	Description	Type	Reset
[5]	sn#{index*4 + 0}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	0b0
[4]	sn#{index*4 + 0}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	0b0
[3]	sn#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 0} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[2]	sn#{index*4 + 0}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 0} when set	RW	0b0
[1]	sn#{index*4 + 0}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 0} when set	RW	0b0
[0]	sn#{index*4 + 0}_128b	Data width of SN #{index*4 + 0}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0

### 8.3.9.172 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5700 + #{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

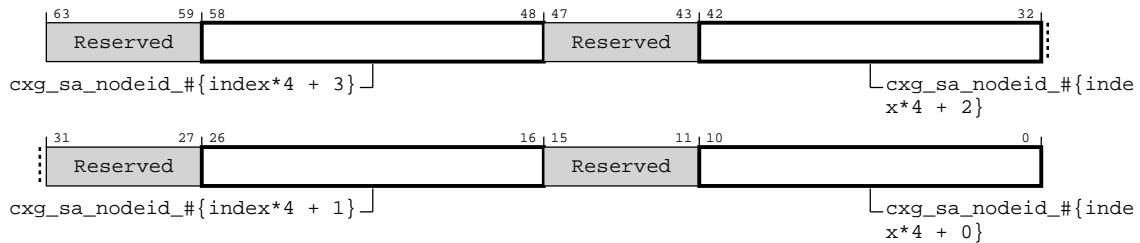
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.cfg\_ctl bit are

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 8-583: cmn\_hns\_sam\_ccg\_sa\_nodeid\_reg0-3**



**Table 8-589: cmn\_hns\_sam\_ccg\_sa\_nodeid\_reg0-3 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	$\text{cxg\_sa\_nodeid}_{\{\text{index}*4 + 3\}}$	Hashed target CCG SA node ID $\{\text{index}*4 + 3\}$	RW	0b000000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	$\text{cxg\_sa\_nodeid}_{\{\text{index}*4 + 2\}}$	Hashed target CCG SA node ID $\{\text{index}*4 + 2\}$	RW	0b000000000000
[31:27]	Reserved	Reserved	RO	-
[26:16]	$\text{cxg\_sa\_nodeid}_{\{\text{index}*4 + 1\}}$	Hashed target CCG SA node ID $\{\text{index}*4 + 1\}$	RW	0b000000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	$\text{cxg\_sa\_nodeid}_{\{\text{index}*4 + 0\}}$	Hashed target CCG SA node ID $\{\text{index}*4 + 0\}$	RW	0b000000000000

## 8.3.9.173 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

$0x5740 + \{\text{index}\}*8$

#### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.cfg\_ctl

### Secure group override

cmn\_hns\_scr.cfg\_ctl

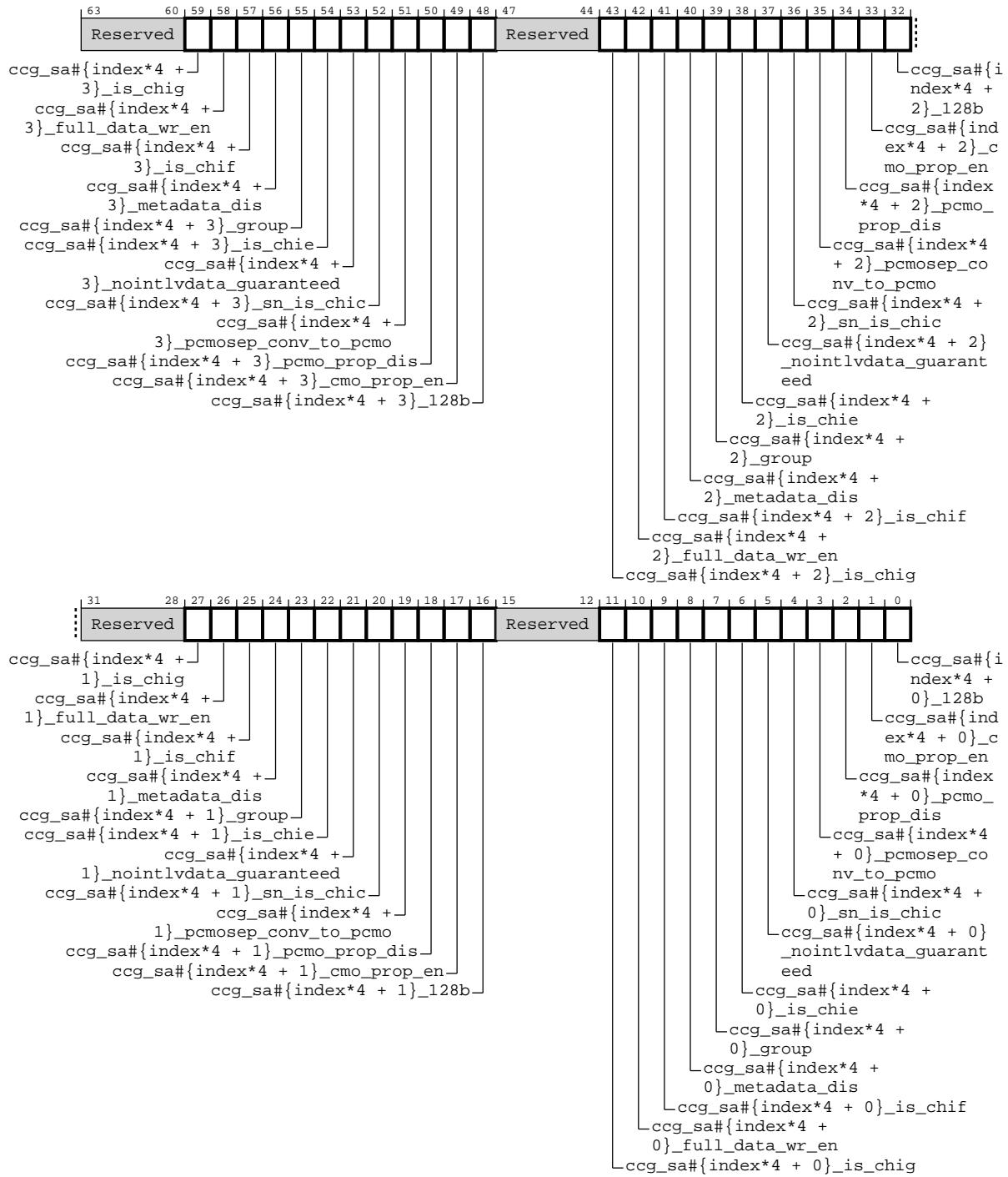
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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.

**Figure 8-584: cmn\_hns\_sam\_ccg\_sa\_attr0-3**



**Table 8-590: cmn\_hns\_sam\_ccg\_sa\_attr0-3 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[59]	ccg_sa#{index*4 + 3}_is_chig	CCG_SA #{index*4 + 3} supports CHI-G	RW	0b0
[58]	ccg_sa#{index*4 + 3}_full_data_wr_en	HNS implements always sending 64B write for CCG_SA #{index*4 + 3} when set	RW	0b0
[57]	ccg_sa#{index*4 + 3}_is_chif	CCG_SA #{index*4 + 3} supports CHI-F	RW	0b0
[56]	ccg_sa#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 3} when set	RW	0b0
[55]	ccg_sa#{index*4 + 3}_group	Specifies the CCG_SA grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[54]	ccg_sa#{index*4 + 3}_is_chie	CCG_SA #{index*4 + 3} supports CHI-E	RW	0b0
[53]	ccg_sa#{index*4 + 3}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	0b0
[52]	ccg_sa#{index*4 + 3}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	0b0
[51]	ccg_sa#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 3} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[50]	ccg_sa#{index*4 + 3}_pcmoe_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 3} when set	RW	0b0
[49]	ccg_sa#{index*4 + 3}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 3} when set	RW	0b0
[48]	ccg_sa#{index*4 + 3}_128b	Data width of CCG_SA #{index*4 + 3}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[47:44]	Reserved	Reserved	RO	-
[43]	ccg_sa#{index*4 + 2}_is_chig	CCG_SA #{index*4 + 2} supports CHI-G	RW	0b0
[42]	ccg_sa#{index*4 + 2}_full_data_wr_en	HNS implements always sending 64B write for CCG_SA #{index*4 + 2} when set	RW	0b0
[41]	ccg_sa#{index*4 + 2}_is_chif	CCG_SA #{index*4 + 2} supports CHI-F	RW	0b0
[40]	ccg_sa#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 2} when set	RW	0b0
[39]	ccg_sa#{index*4 + 2}_group	Specifies the CCG_SA grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[38]	ccg_sa#{index*4 + 2}_is_chie	CCG_SA #{index*4 + 2} supports CHI-E	RW	0b0
[37]	ccg_sa#{index*4 + 2}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	0b0
[36]	ccg_sa#{index*4 + 2}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	0b0

Bits	Name	Description	Type	Reset
[35]	ccg_sa#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{{index*4 + 2}} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[34]	ccg_sa#{index*4 + 2}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{{index*4 + 2}} when set	RW	0b0
[33]	ccg_sa#{index*4 + 2}_cmo_prop_en	Enables CMO propagation for CCG_SA #{{index*4 + 2}} when set	RW	0b0
[32]	ccg_sa#{index*4 + 2}_128b	Data width of CCG_SA #{{index*4 + 2}}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[31:28]	Reserved	Reserved	RO	-
[27]	ccg_sa#{index*4 + 1}_is_chig	CCG_SA #{{index*4 + 1}} supports CHI-G	RW	0b0
[26]	ccg_sa#{index*4 + 1}_full_data_wr_en	HNS implements always sending 64B write for CCG_SA #{{index*4 + 1}} when set	RW	0b0
[25]	ccg_sa#{index*4 + 1}_is_chif	CCG_SA #{{index*4 + 1}} supports CHI-F	RW	0b0
[24]	ccg_sa#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{{index*4 + 1}} when set	RW	0b0
[23]	ccg_sa#{index*4 + 1}_group	Specifies the CCG_SA grouping  <b>0b0</b> Group A  <b>0b1</b> Group B	RW	0b0
[22]	ccg_sa#{index*4 + 1}_is_chie	CCG_SA #{{index*4 + 1}} supports CHI-E	RW	0b0
[21]	ccg_sa#{index*4 + 1}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	0b0
[20]	ccg_sa#{index*4 + 1}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	0b0
[19]	ccg_sa#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{{index*4 + 1}} when set  <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[18]	ccg_sa#{index*4 + 1}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{{index*4 + 1}} when set	RW	0b0
[17]	ccg_sa#{index*4 + 1}_cmo_prop_en	Enables CMO propagation for CCG_SA #{{index*4 + 1}} when set	RW	0b0
[16]	ccg_sa#{index*4 + 1}_128b	Data width of CCG_SA #{{index*4 + 1}}  <b>0b1</b> 128 bits  <b>0b0</b> 256 bits	RW	0b0
[15:12]	Reserved	Reserved	RO	-
[11]	ccg_sa#{index*4 + 0}_is_chig	CCG_SA #{{index*4 + 0}} supports CHI-G	RW	0b0
[10]	ccg_sa#{index*4 + 0}_full_data_wr_en	HNS implements always sending 64B write for CCG_SA #{{index*4 + 0}} when set	RW	0b0
[9]	ccg_sa#{index*4 + 0}_is_chif	CCG_SA #{{index*4 + 0}} supports CHI-F	RW	0b0

Bits	Name	Description	Type	Reset
[8]	ccg_sa#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 0} when set	RW	0b0
[7]	ccg_sa#{index*4 + 0}_group	Specifies the CCG_SA grouping <b>0b0</b> Group A <b>0b1</b> Group B	RW	0b0
[6]	ccg_sa#{index*4 + 0}_is_chie	CCG_SA #{index*4 + 0} supports CHI-E	RW	0b0
[5]	ccg_sa#{index*4 + 0}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	0b0
[4]	ccg_sa#{index*4 + 0}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	0b0
[3]	ccg_sa#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 0} when set <b>CONSTRAINT</b> Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	0b0
[2]	ccg_sa#{index*4 + 0}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 0} when set	RW	0b0
[1]	ccg_sa#{index*4 + 0}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 0} when set	RW	0b0
[0]	ccg_sa#{index*4 + 0}_128b	Data width of CCG_SA #{index*4 + 0} <b>0b1</b> 128 bits <b>0b0</b> 256 bits	RW	0b0

### 8.3.9.174 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x5780 + #{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

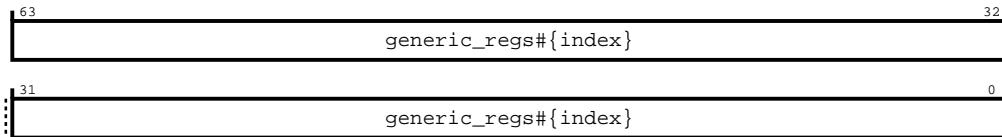
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.cfg\_ctl bit and cmn\_hns\_rcr.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 8-585: hns\_generic\_regs0-7**



**Table 8-591: hns\_generic\_regs0-7 attributes**

Bits	Name	Description	Type	Reset
[63:0]	generic_regs#{index}	Configuration register for the custom logic	RW	0x0

### 8.3.9.175 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

0x5900

##### Type

RW

##### Reset value

See individual bit resets

## Root group override

`cmn_hns_rcr.pa2setaddr_ctl`

## Secure group override

`cmn_hns_scr.pa2setaddr_ctl`

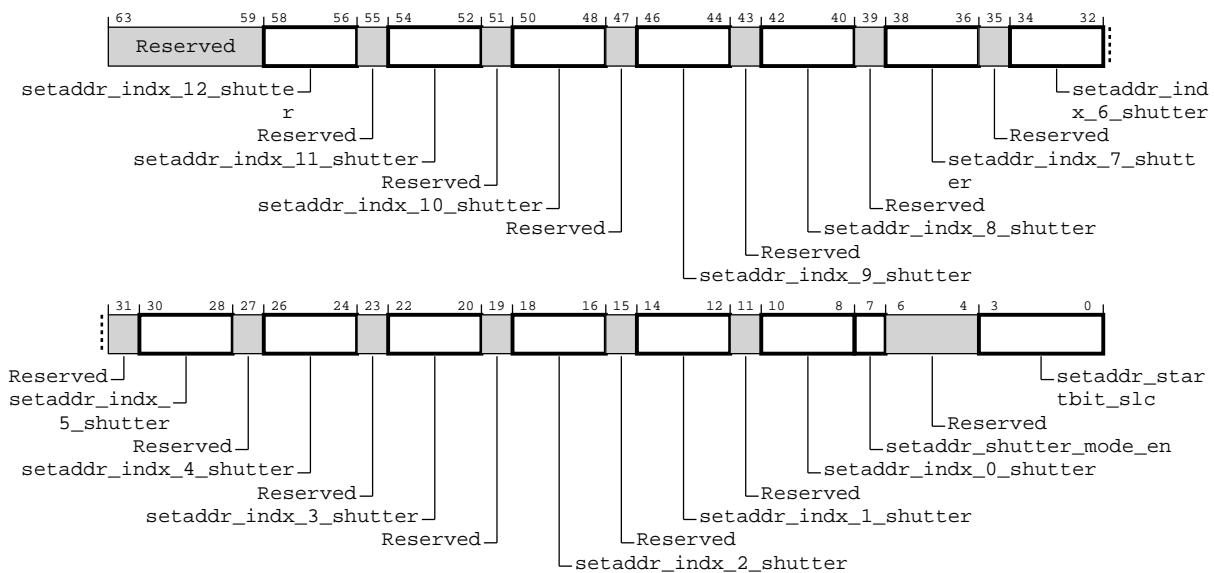
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.pa2setaddr_ctl` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.pa2setaddr_ctl` bit and `cmn_hns_rcr.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 8-586: cmn\_hns\_pa2setaddr\_slc**



**Table 8-592: cmn\_hns\_pa2setaddr\_slc attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[58:56]	setaddr_idx_12_shutter	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[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 <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[51]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[50:48]	setaddr_idx_10_shutter	<p>Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_idx_9_shutter	<p>Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[43]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[42:40]	setaddr_idx_8_shutter	<p>Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_idx_7_shutter	<p>Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[35]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[34:32]	setaddr_idx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[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 <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[27]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[26:24]	setaddr_idx_4_shutter	<p>Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_idx_3_shutter	<p>Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	setaddr_idx_2_shutter	<p>Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_idx_1_shutter	<p>Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_slc</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[11]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[10:8]	setaddr_idx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SLC as programmed by setaddr_idx_X_shutter registers	RW	0b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_slc	<b>SLC</b> SetAddr starting bit for SLC <b>0b0110</b> Setaddr starts from PA[6] <b>0b0111</b> Setaddr starts from PA[7] <b>0b1000</b> Setaddr starts from PA[8] <b>0b1001</b> Setaddr starts from PA[9] <b>0b1010</b> Setaddr starts from PA[10] <b>0b1011</b> Setaddr starts from PA[11] <b>0b1100</b> Setaddr starts from PA[12]	RW	0b0110

### 8.3.9.176 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

0x5908

### Type

RW

### Reset value

See individual bit resets

### Root group override

`cmn_hns_rcr.pa2setaddr_ctl`

### Secure group override

`cmn_hns_scr.pa2setaddr_ctl`

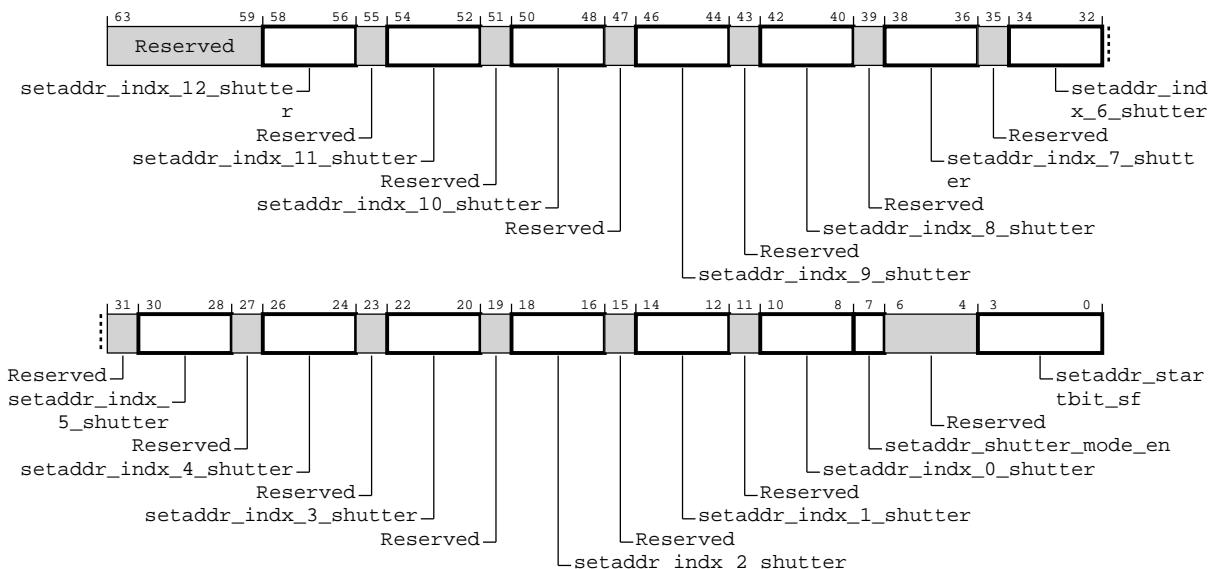
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.pa2setaddr_ctl` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.pa2setaddr_ctl` bit and `cmn_hns_rcr.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 8-587: cmn\_hns\_pa2setaddr\_sf**



**Table 8-593: cmn\_hns\_pa2setaddr\_sf 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_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_idx_11_shutter	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[51]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[50:48]	setaddr_idx_10_shutter	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_idx_9_shutter	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[43]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[42:40]	setaddr_idx_8_shutter	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_idx_7_shutter	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[35]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[34:32]	setaddr_idx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_idx_5_shutter	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[27]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[26:24]	setaddr_idx_4_shutter	<p>Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_sf</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_idx_3_shutter	<p>Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_sf</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p>	RW	0b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	setaddr_idx_2_shutter	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_idx_1_shutter	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[11]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[10:8]	setaddr_idx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_sf <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SF as programmed by setaddr_idx_X_shutter registers	RW	0b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_sf	<b>SF</b> SetAddr starting bit for SF <b>0b0110</b> Setaddr starts from PA[6] <b>0b0111</b> Setaddr starts from PA[7] <b>0b1000</b> Setaddr starts from PA[8] <b>0b1001</b> Setaddr starts from PA[9] <b>0b1010</b> Setaddr starts from PA[10] <b>0b1011</b> Setaddr starts from PA[11] <b>0b1100</b> Setaddr starts from PA[12]	RW	0b0110

### 8.3.9.177 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

0x5910

### Type

RW

### Reset value

See individual bit resets

### Root group override

cmn\_hns\_rcr.pa2setaddr\_ctl

### Secure group override

cmn\_hns\_scr.pa2setaddr\_ctl

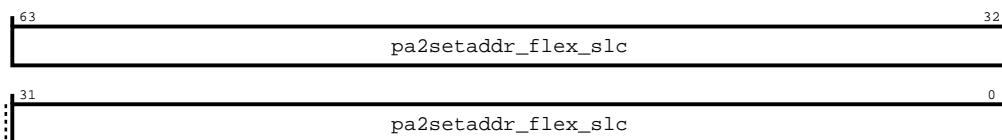
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.pa2setaddr\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.pa2setaddr\_ctl bit and cmn\_hns\_rcr.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 8-588: cmn\_hns\_pa2setaddr\_flex\_slc**



**Table 8-594: cmn\_hns\_pa2setaddr\_flex\_slc attributes**

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_slc	<b>FLEXIBLE</b> PA to SET/TAG ADDR and vice versa conversion config field for SLC	RW	0b0

### 8.3.9.178 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

0x5918

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

cmn\_hns\_rcr.pa2setaddr\_ctl

##### Secure group override

cmn\_hns\_scr.pa2setaddr\_ctl

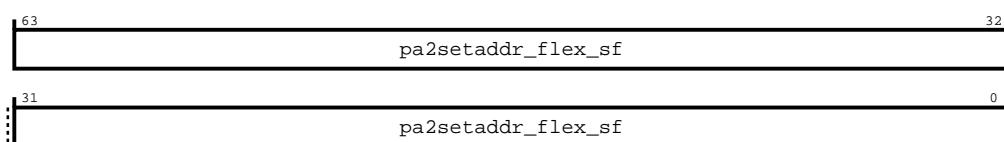
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.pa2setaddr\_ctl bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.pa2setaddr\_ctl bit and cmn\_hns\_rcr.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 8-589: cmn\_hns\_pa2setaddr\_flex\_sf**



**Table 8-595: cmn\_hns\_pa2setaddr\_flex\_sf attributes**

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_sf	<b>FLEXIBLE</b> PA to SET/TAG ADDR conversion and vice versa config field for SF	RW	0b0

### 8.3.9.179 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7000 + #{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

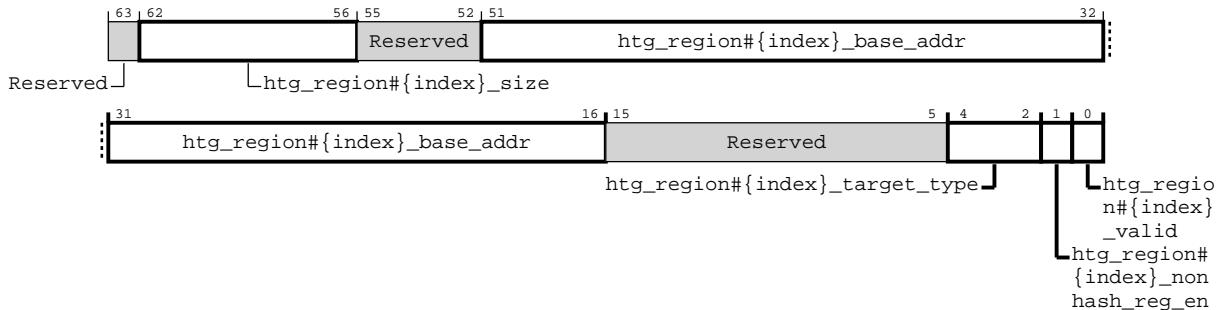
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-590: lcn\_hashed\_tgt\_grp\_cfg1\_region0-31**



**Table 8-596: lcn\_hashed\_tgt\_grp\_cfg1\_region0-31 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_region#{index}_size	<p>Memory region #{index} size</p> <p><b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size (<math>2^{\text{address width}}</math>).</p>	RW	0b0000000
[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_RNSAMHTGRCOMP_LSB_PARAM	RW	0x0
[15:5]	Reserved	Reserved	RO	-
[4:2]	htg_region#{index}_target_type	<p>Indicates node type</p> <p><b>0b000</b> HN-F</p> <p><b>0b001</b> HN-I</p> <p><b>0b010</b> CXRA</p> <p><b>0b011</b> HN-P</p> <p><b>0b100</b> PCI-CXRA</p> <p><b>0b101</b> HN-S</p> <p><b>Others</b> Reserved</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b000
[1]	htg_region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	0b0

Bits	Name	Description	Type	Reset
[0]	htg_region#{index}_valid	<p>Memory region #{index} valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.9.180 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7100 + #{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

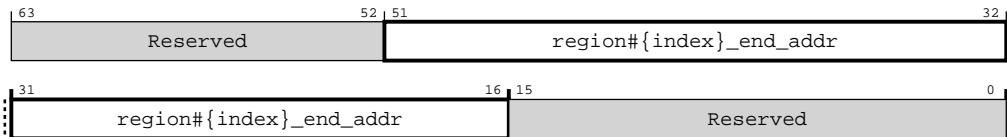
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-591: lcn\_hashed\_tgt\_grp\_cfg2\_region0-31**



**Table 8-597: 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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15:0]	Reserved	Reserved	RO	

### 8.3.9.181 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7200 + #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

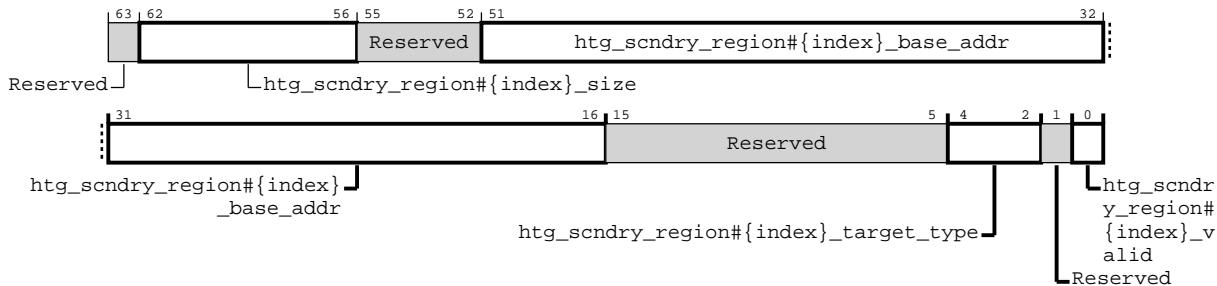
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.sam\_control bit 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 8-592: lcn\_hashed\_target\_grp\_secondary\_cfg1\_reg0-31**



**Table 8-598: 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  <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:2]	htg_scndry_region#{index}_target_type	<p>Indicates node type</p> <p><b>0b000</b> HN-F</p> <p><b>0b001</b> HN-I</p> <p><b>0b010</b> CXRA</p> <p><b>0b011</b> HN-P</p> <p><b>0b100</b> PCI-CXRA</p> <p><b>0b101</b> HN-S</p> <p><b>Others</b> Reserved</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b000
[1]	Reserved	Reserved	RO	-
[0]	htg_scndry_region#{index}_valid	<p>Secondary memory region #{index} valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.9.182 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7300 + #{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

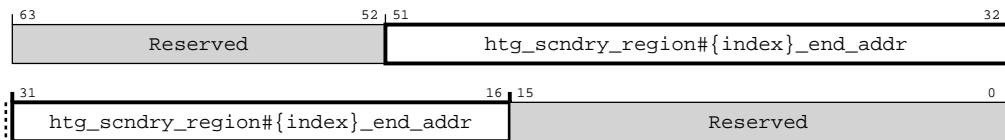
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-593: lcn\_hashed\_target\_grp\_secondary\_cfg2\_reg0-31**



**Table 8-599: 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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0b00000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

### 8.3.9.183 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

## Address offset

`index(0-31) : 0x7400 + #{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`

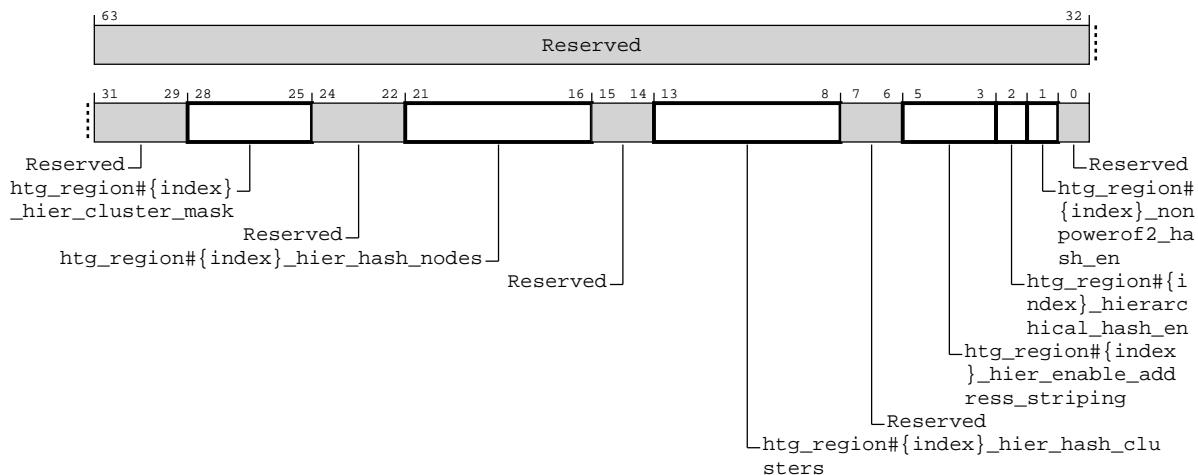
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `cmn_hns_scr.sam_control` bit is set, Secure accesses to this register are permitted. If both the `cmn_hns_scr.sam_control` bit and `cmn_hns_rcr.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 8-594: lcn\_hashed\_target\_grp\_hash\_cntl\_reg0-31**



**Table 8-600: lcn\_hashed\_target\_grp\_hash\_cntl\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[28:25]	htg_region#{index}_hier_cluster_mask	<p><b>Hierarchical hashing</b></p> <p>Enable cluster masking to achieve different interleave granularity across clusters.</p> <p><b>0b0000</b> 64 byte interleave granularity across clusters</p> <p><b>0b0001</b> 128 byte interleave granularity across clusters</p> <p><b>0b0010</b> 256 byte interleave granularity across clusters</p> <p><b>0b0011</b> 512 byte interleave granularity across clusters</p> <p><b>0b0100</b> 1024 byte interleave granularity across clusters</p> <p><b>0b0101</b> 2048 byte interleave granularity across clusters</p> <p><b>0b0110</b> 4096 byte interleave granularity across clusters</p> <p><b>0b0111</b> 8192 byte interleave granularity across clusters</p> <p><b>others</b> Reserved</p>	RW	0b0
[24:22]	Reserved	Reserved	RO	
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	0x0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:3]	htg_region#{index}_hier_enable_address_striping	<p><b>Hierarchical hashing</b></p> <p>configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask).</p> <p><b>0b000</b> no address shuttering</p> <p><b>0b001</b> one addr bit shuttered (2 clusters)</p> <p><b>0b010</b> two addr bit shuttered (4 clusters)</p> <p><b>0b011</b> three addr bit shuttered (8 clusters)</p> <p><b>0b100</b> four addr bit shuttered (16 clusters)</p> <p><b>0b101</b> five addr bit shuttered (32 clusters)</p> <p><b>0b110</b> six addr bit shuttered (These are configured when the cachelines are &gt; 64B interleaved and clusters are enabled)</p> <p><b>0b111</b> seven addr bit shuttered</p> <p><b>others</b> Reserved</p>	RW	0b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	0b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	0b0
[0]	Reserved	Reserved	RO	

### 8.3.9.184 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 #{index8} to #{index8 + 7}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-3) : 0x7500 + #{8 \* index}

## Type

RW

## Reset value

See individual bit resets

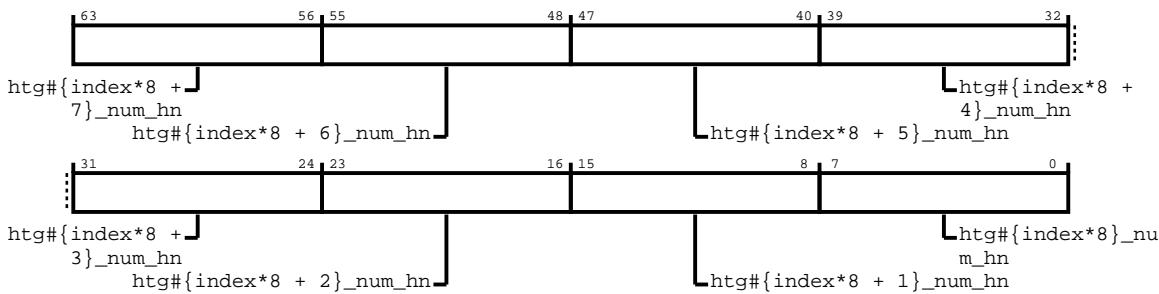
## Usage constraints

This register is owned in the Root space and is 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 8-595: lcn\_hashed\_target\_group\_hn\_count\_reg0-3**



**Table 8-601: 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	0x00
[55:48]	htg#{index*8 + 6}_num_hn	HN count for hashed target group 6	RW	0x00
[47:40]	htg#{index*8 + 5}_num_hn	HN count for hashed target group 5	RW	0x00
[39:32]	htg#{index*8 + 4}_num_hn	HN count for hashed target group 4	RW	0x00
[31:24]	htg#{index*8 + 3}_num_hn	HN count for hashed target group 3	RW	0x00
[23:16]	htg#{index*8 + 2}_num_hn	HN count for hashed target group 2	RW	0x00
[15:8]	htg#{index*8 + 1}_num_hn	HN count for hashed target group 1	RW	0x00
[7:0]	htg#{index*8}_num_hn	HN count for hashed target group 0	RW	0x00

## 8.3.9.185 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-7) : 0x7520 + #{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

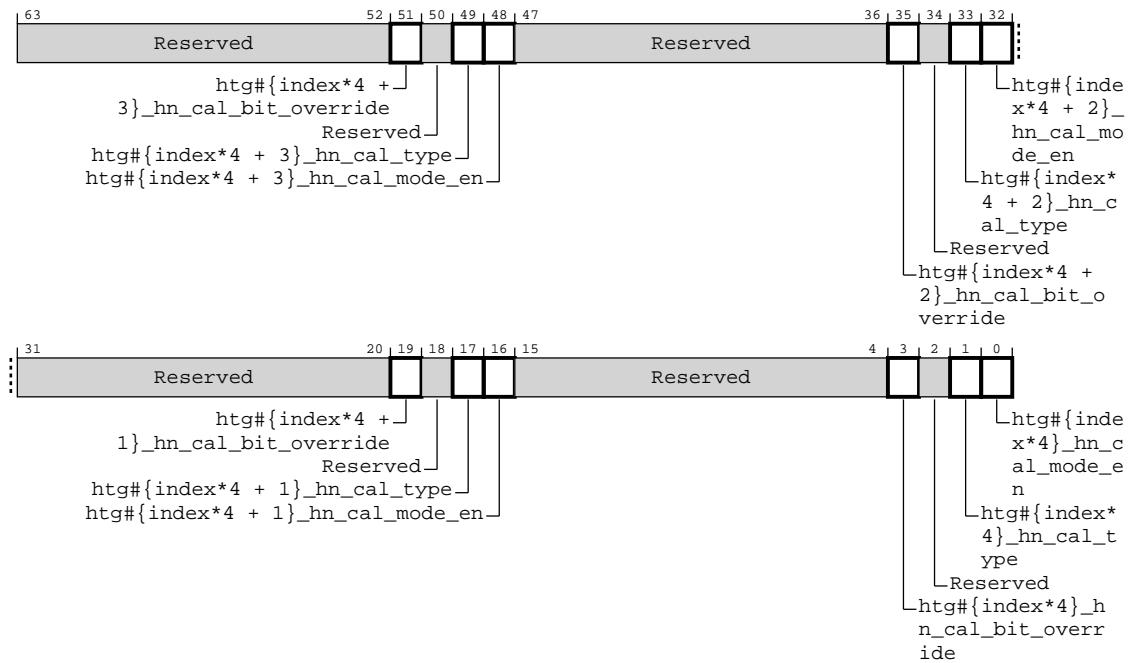
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the cmn\_hns\_scr.sam\_control bit is set, Secure accesses to this register are permitted. If both the cmn\_hns\_scr.sam\_control bit and cmn\_hns\_rcr.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 8-596: lcn\_hashed\_target\_grp\_cal\_mode\_reg0-7**



**Table 8-602: lcn\_hashed\_target\_grp\_cal\_mode\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51]	htg#{index*4 + 3}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 3}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[50]	Reserved	Reserved	RO	
[49]	htg#{index*4 + 3}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 3}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[48]	htg#{index*4 + 3}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 3}	RW	0b0
[47:36]	Reserved	Reserved	RO	
[35]	htg#{index*4 + 2}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 2}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0

Bits	Name	Description	Type	Reset
[34]	Reserved	Reserved	RO	
[33]	htg#[index*4 + 2]_hn_cal_type	Enables type of HN CAL for HTG #[index*4 + 2]  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[32]	htg#[index*4 + 2]_hn_cal_mode_en	Enables support for HN CAL for HTG #[index*4 + 2]	RW	0b0
[31:20]	Reserved	Reserved	RO	
[19]	htg#[index*4 + 1]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #[index*4 + 1]  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[18]	Reserved	Reserved	RO	
[17]	htg#[index*4 + 1]_hn_cal_type	Enables type of HN CAL for HTG #[index*4 + 1]  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[16]	htg#[index*4 + 1]_hn_cal_mode_en	Enables support for HN CAL for HTG #[index*4 + 1]	RW	0b0
[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]  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[2]	Reserved	Reserved	RO	
[1]	htg#[index*4]_hn_cal_type	Enables type of HN CAL for HTG #[index*4]  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[0]	htg#[index*4]_hn_cal_mode_en	Enables support for HN CAL for HTG #[index*4]	RW	0b0

### 8.3.9.186 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-1) : 0x7560 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

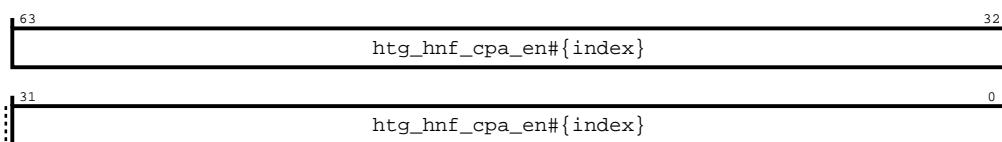
##### Usage constraints

This register is owned in the Root space and is 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 8-597: lcn\_hashed\_target\_grp\_hnf\_cpa\_en\_reg0-1**



**Table 8-603: 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	0x0000000000000000

### 8.3.9.187 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15) : 0x7580 + #8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

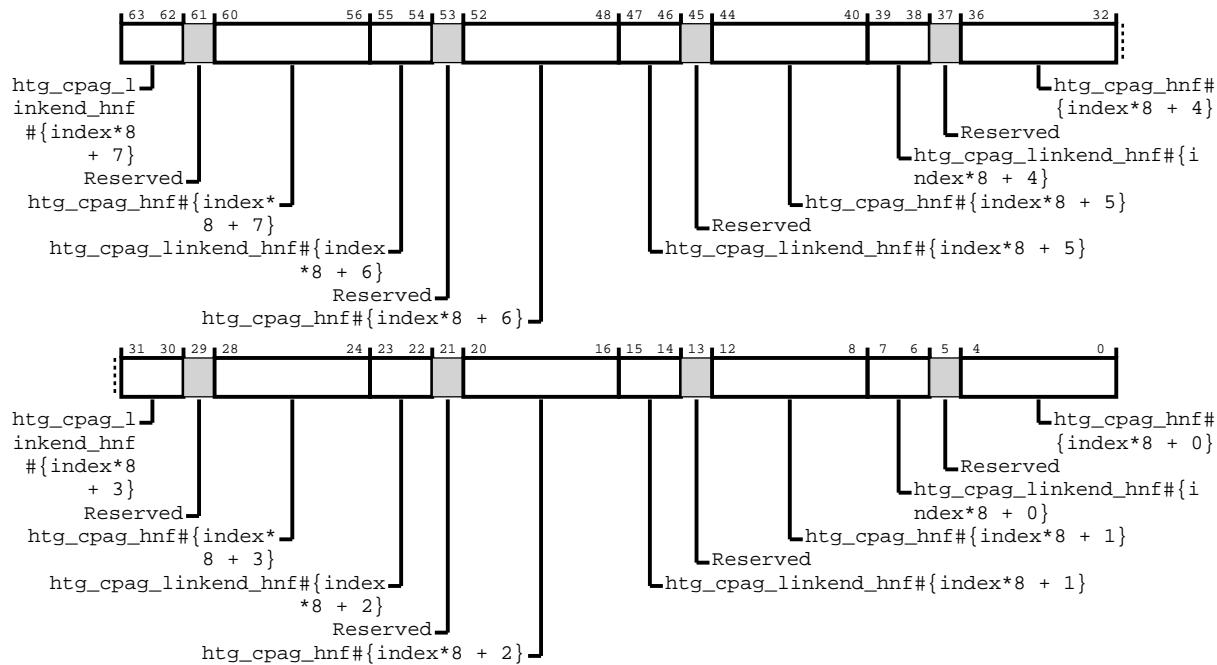
##### Usage constraints

This register is owned in the Root space and is 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 8-598: lcn\_hashed\_target\_grp\_cpag\_perhnf\_reg0-15**



**Table 8-604: lcn\_hashed\_target\_grp\_cpag\_perhnf\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63:62]	htg_cpag_linkend_hnf#{index*8 + 7}	CPAG Linkend associated to the HNF#{index*8 + 7}	RW	0b0
[61]	Reserved	Reserved	RO	
[60:56]	htg_cpag_hnf#{index*8 + 7}	CPAG associated to the HNF#{index*8 + 7}	RW	0b0
[55:54]	htg_cpag_linkend_hnf#{index*8 + 6}	CPAG Linkend associated to the HNF#{index*8 + 6}	RW	0b0
[53]	Reserved	Reserved	RO	
[52:48]	htg_cpag_hnf#{index*8 + 6}	CPAG associated to the HNF#{index*8 + 6}	RW	0b0
[47:46]	htg_cpag_linkend_hnf#{index*8 + 5}	CPAG Linkend associated to the HNF#{index*8 + 5}	RW	0b0
[45]	Reserved	Reserved	RO	
[44:40]	htg_cpag_hnf#{index*8 + 5}	CPAG associated to the HNF#{index*8 + 5}	RW	0b0
[39:38]	htg_cpag_linkend_hnf#{index*8 + 4}	CPAG Linkend associated to the HNF#{index*8 + 4}	RW	0b0
[37]	Reserved	Reserved	RO	
[36:32]	htg_cpag_hnf#{index*8 + 4}	CPAG associated to the HNF#{index*8 + 4}	RW	0b0
[31:30]	htg_cpag_linkend_hnf#{index*8 + 3}	CPAG Linkend associated to the HNF#{index*8 + 3}	RW	0b0
[29]	Reserved	Reserved	RO	
[28:24]	htg_cpag_hnf#{index*8 + 3}	CPAG associated to the HNF#{index*8 + 3}	RW	0b0
[23:22]	htg_cpag_linkend_hnf#{index*8 + 2}	CPAG Linkend associated to the HNF#{index*8 + 2}	RW	0b0
[21]	Reserved	Reserved	RO	
[20:16]	htg_cpag_hnf#{index*8 + 2}	CPAG associated to the HNF#{index*8 + 2}	RW	0b0
[15:14]	htg_cpag_linkend_hnf#{index*8 + 1}	CPAG Linkend associated to the HNF#{index*8 + 1}	RW	0b0
[13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12:8]	htg_cpag_hnf#[index*8 + 1]	CPAG associated to the HNF#[index*8 + 1]	RW	0b0
[7:6]	htg_cpag_linkend_hnf#[index*8 + 0]	CPAG Linkend associated to the HNF#[index*8 + 0]	RW	0b0
[5]	Reserved	Reserved	RO	
[4:0]	htg_cpag_hnf#[index*8 + 0]	CPAG associated to the HNF#[index*8 + 0]	RW	0b0

### 8.3.9.188 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7700 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

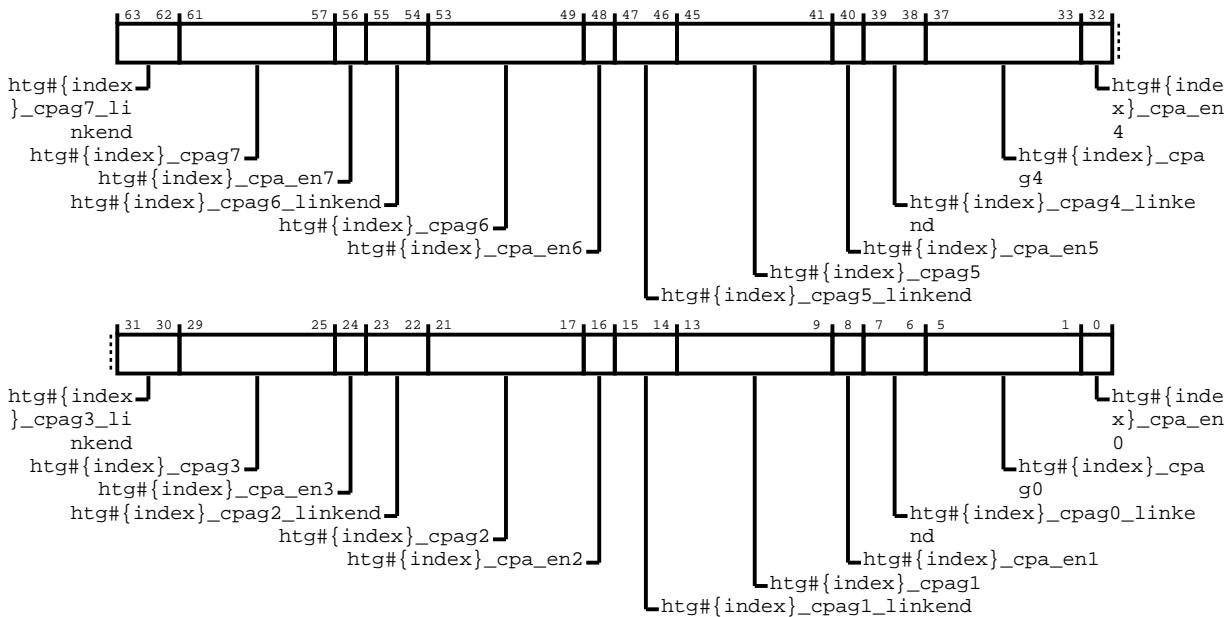
##### Usage constraints

This register is owned in the Root space and is 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 8-599: lcn\_hashed\_target\_grp\_compact\_cpag\_ctrl0-31**



**Table 8-605: lcn\_hashed\_target\_grp\_compact\_cpag\_ctrl0-31 attributes**

Bits	Name	Description	Type	Reset
[63:62]	htg#{index}_cpag7_linkend	cpag id for index7	RW	0b0
[61:57]	htg#{index}_cpag7	cpag id for index7	RW	0b0
[56]	htg#{index}_cpa_en7	cpa enable for index7	RW	0b0
[55:54]	htg#{index}_cpag6_linkend	cpag id for index6	RW	0b0
[53:49]	htg#{index}_cpag6	cpag id for index6	RW	0b0
[48]	htg#{index}_cpa_en6	cpa enable for index6	RW	0b0
[47:46]	htg#{index}_cpag5_linkend	cpag id for index5	RW	0b0
[45:41]	htg#{index}_cpag5	cpag id for index5	RW	0b0
[40]	htg#{index}_cpa_en5	cpa enable for index5	RW	0b0
[39:38]	htg#{index}_cpag4_linkend	cpag id for index4	RW	0b0
[37:33]	htg#{index}_cpag4	cpag id for index4	RW	0b0
[32]	htg#{index}_cpa_en4	cpa enable for index4	RW	0b0
[31:30]	htg#{index}_cpag3_linkend	cpag id for index3	RW	0b0
[29:25]	htg#{index}_cpag3	cpag id for index0	RW	0b0
[24]	htg#{index}_cpa_en3	cpa enable for index3	RW	0b0
[23:22]	htg#{index}_cpag2_linkend	cpag id for index2	RW	0b0
[21:17]	htg#{index}_cpag2	cpag id for index2	RW	0b0
[16]	htg#{index}_cpa_en2	cpa enable for index2	RW	0b0
[15:14]	htg#{index}_cpag1_linkend	cpag id for index1	RW	0b0
[13:9]	htg#{index}_cpag1	cpag id for index1	RW	0b0
[8]	htg#{index}_cpa_en1	cpa enable for index1	RW	0b0

Bits	Name	Description	Type	Reset
[7:6]	htg#{index}_cpag0_linkend	cpag id for index0	RW	0b0
[5:1]	htg#{index}_cpag0	cpag id for index0	RW	0b0
[0]	htg#{index}_cpa_en0	cpa enable for index0	RW	0b0

### 8.3.9.189 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x7800 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

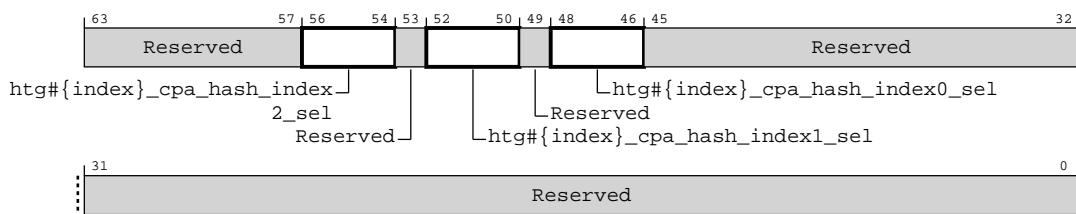
##### Usage constraints

This register is owned in the Root space and is 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 8-600: lcn\_hashed\_target\_grp\_compact\_hash\_ctrl0-31**



**Table 8-606: 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	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index2.</p> <p><b>0b001</b> SMP hash index2 + 1.</p> <p><b>0b010</b> SMP hash index2 + 2.</p> <p><b>0b011</b> SMP hash index2 + 3.</p> <p><b>0b100</b> SMP hash index2 + 4.</p> <p><b>0b101</b> SMP hash index2 + 5.</p> <p><b>0b110</b> SMP hash index2 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[53]	Reserved	Reserved	RO	
[52:50]	htg#{index}_cpa_hash_index1_sel	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index1.</p> <p><b>0b001</b> SMP hash index1 + 1.</p> <p><b>0b010</b> SMP hash index1 + 2.</p> <p><b>0b011</b> SMP hash index1 + 3.</p> <p><b>0b100</b> SMP hash index1 + 4.</p> <p><b>0b101</b> SMP hash index1 + 5.</p> <p><b>0b110</b> SMP hash index1 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[49]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[48:46]	htg#{index}_cpa_hash_index0_sel	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index0.</p> <p><b>0b001</b> SMP hash index0 + 1.</p> <p><b>0b010</b> SMP hash index0 + 2.</p> <p><b>0b011</b> SMP hash index0 + 3.</p> <p><b>0b100</b> SMP hash index0 + 4.</p> <p><b>0b101</b> SMP hash index0 + 5.</p> <p><b>0b110</b> SMP hash index0 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[45:0]	Reserved	Reserved	RO	

### 8.3.10 HN-S MPAM NS register summary

The following table describes the registers for the relevant component.

**Table 8-607: cmn\_hns\_mpam\_ns\_cfg register summary**

Offset	Name	Type	Description
0x0	cmn_hns_mpam_ns_node_info	RO	<a href="#">cmn_hns_mpam_ns_node_info</a>
0x80	cmn_hns_mpam_ns_child_info	RO	<a href="#">cmn_hns_mpam_ns_child_info</a>
0x1000	cmn_hns_ns_mpam_idr	RO	<a href="#">cmn_hns_ns_mpam_idr</a>
0x1018	cmn_hns_mpam_iidr	RO	<a href="#">cmn_hns_mpam_iidr</a>
0x1020	cmn_hns_mpam_aidr	RO	<a href="#">cmn_hns_mpam_aidr</a>
0x1028	cmn_hns_ns_mpam_impl_idr	RO	<a href="#">cmn_hns_ns_mpam_impl_idr</a>
0x1030	cmn_hns_ns_mpam_cpor_idr	RO	<a href="#">cmn_hns_ns_mpam_cpor_idr</a>
0x1038	cmn_hns_ns_mpam_ccap_idr	RO	<a href="#">cmn_hns_ns_mpam_ccap_idr</a>
0x1040	cmn_hns_ns_mpam_mbw_idr	RO	<a href="#">cmn_hns_ns_mpam_mbw_idr</a>
0x1048	cmn_hns_ns_mpam_pri_idr	RO	<a href="#">cmn_hns_ns_mpam_pri_idr</a>
0x1050	cmn_hns_ns_mpam_partid_nrw_idr	RO	<a href="#">cmn_hns_ns_mpam_partid_nrw_idr</a>
0x1080	cmn_hns_ns_mpam_msmon_idr	RO	<a href="#">cmn_hns_ns_mpam_msmon_idr</a>
0x1088	cmn_hns_ns_mpam_csumon_idr	RO	<a href="#">cmn_hns_ns_mpam_csumon_idr</a>
0x1090	cmn_hns_ns_mpam_mbwumon_idr	RO	<a href="#">cmn_hns_ns_mpam_mbwumon_idr</a>
0x10F0	cmn_hns_ns_mpam_ecr	RW	<a href="#">cmn_hns_ns_mpam_ecr</a>
0x10F8	cmn_hns_ns_mpam_esr	RW	<a href="#">cmn_hns_ns_mpam_esr</a>

Offset	Name	Type	Description
0x1100	cmn_hns_ns_mpamcfg_part_sel	RW	<a href="#">cmn_hns_ns_mpamcfg_part_sel</a>
0x1108	cmn_hns_ns_mpamcfg_cmax	RW	<a href="#">cmn_hns_ns_mpamcfg_cmax</a>
0x1200	cmn_hns_ns_mpamcfg_mbw_min	RW	<a href="#">cmn_hns_ns_mpamcfg_mbw_min</a>
0x1208	cmn_hns_ns_mpamcfg_mbw_max	RW	<a href="#">cmn_hns_ns_mpamcfg_mbw_max</a>
0x1220	cmn_hns_ns_mpamcfg_mbw_winwd	RW	<a href="#">cmn_hns_ns_mpamcfg_mbw_winwd</a>
0x1400	cmn_hns_ns_mpamcfg_pri	RW	<a href="#">cmn_hns_ns_mpamcfg_pri</a>
0x1500	cmn_hns_ns_mpamcfg_mbw_prop	RW	<a href="#">cmn_hns_ns_mpamcfg_mbw_prop</a>
0x1600	cmn_hns_ns_mpamcfg_intpartid	RW	<a href="#">cmn_hns_ns_mpamcfg_intpartid</a>
0x1800	cmn_hns_ns_msmon_cfg_mon_sel	RW	<a href="#">cmn_hns_ns_msmon_cfg_mon_sel</a>
0x1808	cmn_hns_ns_msmon_capt_evnt	RW	<a href="#">cmn_hns_ns_msmon_capt_evnt</a>
0x1810	cmn_hns_ns_msmon_cfg_csu_ftl	RW	<a href="#">cmn_hns_ns_msmon_cfg_csu_ftl</a>
0x1818	cmn_hns_ns_msmon_cfg_csu_ctl	RW	<a href="#">cmn_hns_ns_msmon_cfg_csu_ctl</a>
0x1820	cmn_hns_ns_msmon_cfg_mbwu_ftl	RW	<a href="#">cmn_hns_ns_msmon_cfg_mbwu_ftl</a>
0x1828	cmn_hns_ns_msmon_cfg_mbwu_ctl	RW	<a href="#">cmn_hns_ns_msmon_cfg_mbwu_ctl</a>
0x1840	cmn_hns_ns_msmon_csu	RW	<a href="#">cmn_hns_ns_msmon_csu</a>
0x1848	cmn_hns_ns_msmon_csu_capture	RW	<a href="#">cmn_hns_ns_msmon_csu_capture</a>
0x1860	cmn_hns_ns_msmon_mbwu	RW	<a href="#">cmn_hns_ns_msmon_mbwu</a>
0x1868	cmn_hns_ns_msmon_mbwu_capture	RW	<a href="#">cmn_hns_ns_msmon_mbwu_capture</a>
0x2000	cmn_hns_ns_mpamcfg_cpbm	RW	<a href="#">cmn_hns_ns_mpamcfg_cpbm</a>
0x8000	cmn_hns_rl_mpam_idr	RO	<a href="#">cmn_hns_rl_mpam_idr</a>
0x8018	cmn_hns_rl_mpam_iidr	RO	<a href="#">cmn_hns_rl_mpam_iidr</a>
0x8020	cmn_hns_rl_mpam_aidr	RO	<a href="#">cmn_hns_rl_mpam_aidr</a>
0x8028	cmn_hns_rl_mpam_impl_idr	RO	<a href="#">cmn_hns_rl_mpam_impl_idr</a>
0x8030	cmn_hns_rl_mpam_cpor_idr	RO	<a href="#">cmn_hns_rl_mpam_cpor_idr</a>
0x8038	cmn_hns_rl_mpam_ccap_idr	RO	<a href="#">cmn_hns_rl_mpam_ccap_idr</a>
0x8040	cmn_hns_rl_mpam_mbw_idr	RO	<a href="#">cmn_hns_rl_mpam_mbw_idr</a>
0x8048	cmn_hns_rl_mpam_pri_idr	RO	<a href="#">cmn_hns_rl_mpam_pri_idr</a>
0x8050	cmn_hns_rl_mpam_partid_nrw_idr	RO	<a href="#">cmn_hns_rl_mpam_partid_nrw_idr</a>
0x8080	cmn_hns_rl_mpam_msmon_idr	RO	<a href="#">cmn_hns_rl_mpam_msmon_idr</a>
0x8088	cmn_hns_rl_mpam_csumon_idr	RO	<a href="#">cmn_hns_rl_mpam_csumon_idr</a>
0x8090	cmn_hns_rl_mpam_mbwumon_idr	RO	<a href="#">cmn_hns_rl_mpam_mbwumon_idr</a>
0x80F0	cmn_hns_rl_mpam_ecr	RW	<a href="#">cmn_hns_rl_mpam_ecr</a>
0x80F8	cmn_hns_rl_mpam_esr	RW	<a href="#">cmn_hns_rl_mpam_esr</a>
0x8100	cmn_hns_rl_mpamcfg_part_sel	RW	<a href="#">cmn_hns_rl_mpamcfg_part_sel</a>
0x8108	cmn_hns_rl_mpamcfg_cmax	RW	<a href="#">cmn_hns_rl_mpamcfg_cmax</a>
0x8200	cmn_hns_rl_mpamcfg_mbw_min	RW	<a href="#">cmn_hns_rl_mpamcfg_mbw_min</a>
0x8208	cmn_hns_rl_mpamcfg_mbw_max	RW	<a href="#">cmn_hns_rl_mpamcfg_mbw_max</a>
0x8220	cmn_hns_rl_mpamcfg_mbw_winwd	RW	<a href="#">cmn_hns_rl_mpamcfg_mbw_winwd</a>
0x8400	cmn_hns_rl_mpamcfg_pri	RW	<a href="#">cmn_hns_rl_mpamcfg_pri</a>
0x8500	cmn_hns_rl_mpamcfg_mbw_prop	RW	<a href="#">cmn_hns_rl_mpamcfg_mbw_prop</a>

Offset	Name	Type	Description
0x8600	cmn_hns_rl_mpamcfg_intpartid	RW	cmn_hns_rl_mpamcfg_intpartid
0x8800	cmn_hns_rl_msmon_cfg_mon_sel	RW	cmn_hns_rl_msmon_cfg_mon_sel
0x8808	cmn_hns_rl_msmon_capt_evnt	RW	cmn_hns_rl_msmon_capt_evnt
0x8810	cmn_hns_rl_msmon_cfg_csu_flg	RW	cmn_hns_rl_msmon_cfg_csu_flg
0x8818	cmn_hns_rl_msmon_cfg_csu_ctl	RW	cmn_hns_rl_msmon_cfg_csu_ctl
0x8820	cmn_hns_rl_msmon_cfg_mbwu_flg	RW	cmn_hns_rl_msmon_cfg_mbwu_flg
0x8828	cmn_hns_rl_msmon_cfg_mbwu_ctl	RW	cmn_hns_rl_msmon_cfg_mbwu_ctl
0x8840	cmn_hns_rl_msmon_csu	RW	cmn_hns_rl_msmon_csu
0x8848	cmn_hns_rl_msmon_csu_capture	RW	cmn_hns_rl_msmon_csu_capture
0x8860	cmn_hns_rl_msmon_mbwu	RW	cmn_hns_rl_msmon_mbwu
0x8868	cmn_hns_rl_msmon_mbwu_capture	RW	cmn_hns_rl_msmon_mbwu_capture
0x9000	cmn_hns_rl_mpamcfg_cpbm	RW	cmn_hns_rl_mpamcfg_cpbm

### 8.3.10.1 cmn\_hns\_mpam\_ns\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

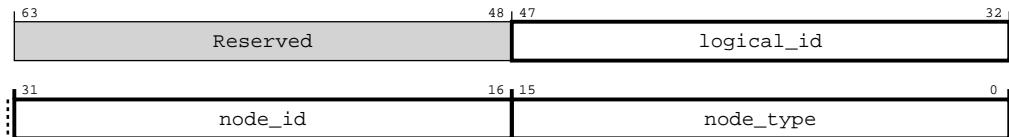
##### Usage constraints

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 8-601: cmn\_hns\_mpam\_ns\_node\_info**



**Table 8-608: cmn\_hns\_mpam\_ns\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x00

### 8.3.10.2 cmn\_hns\_mpam\_ns\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

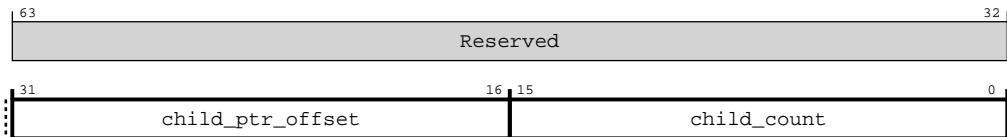
##### Usage constraints

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 8-602: cmn\_hns\_mpam\_ns\_child\_info**



**Table 8-609: 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	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.10.3 cmn\_hns\_ns\_mpam\_idr

MPAM features ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1000

##### Type

RO

##### Reset value

See individual bit resets

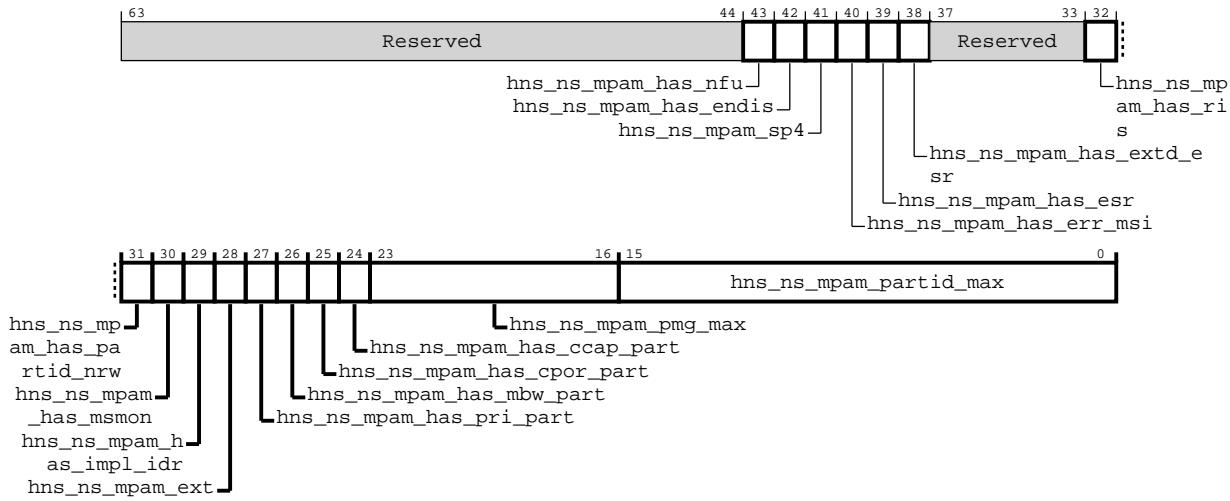
#### Usage constraints

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 8-603: cmn\_hns\_ns\_mpam\_idr**



**Table 8-610: cmn\_hns\_ns\_mpam\_idr attributes**

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_ns_mpam_has_nfu	<p><b>0</b> HN-F does not support no future use field</p> <p><b>1</b> HN-F supports no future use field</p>	RO	0b0
[42]	hns_ns_mpam_has_endis	<p><b>0</b> HN-F does not support PARTID enable and disable functionality</p> <p><b>1</b> HN-F supports PARTID enable and disable functionality</p>	RO	0b0
[41]	hns_ns_mpam_sp4	<p><b>0</b> HN-F supports two PARTID spaces</p> <p><b>1</b> HN-F supports four PARTID spaces</p>	RO	0b1
[40]	hns_ns_mpam_has_err_msi	<p><b>0</b> HN-F does not support MSI writes to signal MPAM error interrupt</p> <p><b>1</b> HN-F supports MSI writes to signal MPAM error interrupt</p>	RO	0b0
[39]	hns_ns_mpam_has_esr	<p><b>0</b> HN-F does not support MPAM error handling</p> <p><b>1</b> HN-F supports MPAM error handling</p>	RO	0b1

Bits	Name	Description	Type	Reset
[38]	hns_ns_mpam_has_extd_esr	<p><b>0</b> MPAMF_ESR is 32 bits</p> <p><b>1</b> MPAMF_ESR is 64 bits</p>	RO	0b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_ns_mpam_has_ris	<p><b>0</b> HN-F does not support MPAM resource instance selector</p> <p><b>1</b> HN-F supports MPAM resource instance selector</p>	RO	0b0
[31]	hns_ns_mpam_has_partid_nrw	<p><b>0</b> HN-F does not support MPAM PARTID Narrowing</p> <p><b>1</b> HN-F supports MPAM PARTID Narrowing</p>	RO	0x0
[30]	hns_ns_mpam_has_msmon	<p><b>0</b> MPAM performance monitoring is not supported</p> <p><b>1</b> MPAM performance monitoring is supported</p>	RO	Configuration dependent
[29]	hns_ns_mpam_has_impl_idr	<p><b>0</b> MPAM implementation specific partitioning features not supported</p> <p><b>1</b> MPAM implementation specific partitioning features supported</p>	RO	0x0
[28]	hns_ns_mpam_ext	<p><b>0</b> HN-F has no defined bits in [63:32]</p> <p><b>1</b> HN-F has bits defined in [63:32]</p>	RO	0b1
[27]	hns_ns_mpam_has_pri_part	<p><b>0</b> MPAM priority partitioning is not supported</p> <p><b>1</b> MPAM priority partitioning is supported</p>	RO	0x0
[26]	hns_ns_mpam_has_mbw_part	<p><b>0</b> MPAM memory bandwidth partitioning is not supported</p> <p><b>1</b> MPAM memory bandwidth partitioning is supported</p>	RO	0x0
[25]	hns_ns_mpam_has_cpor_part	<p><b>0</b> MPAM cache portion partitioning is not supported</p> <p><b>1</b> MPAM cache portion partitioning is supported</p>	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[24]	hns_ns_mpam_has_ccap_part	<b>0</b> MPAM cache maximum capacity partitioning is not supported <b>1</b> 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	0x1
[15:0]	hns_ns_mpam_partid_max	Maximum value of non-secure PARTID supported by this HN-F	RO	0xf

### 8.3.10.4 cmn\_hns\_mpam\_iidr

MPAM Implementation ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1018

##### Type

RO

##### Reset value

See individual bit resets

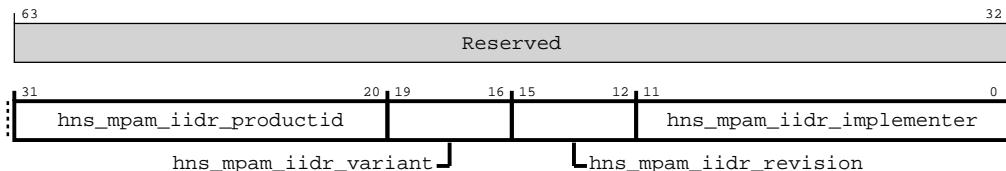
##### Usage constraints

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 8-604: cmn\_hns\_mpam\_iidr**



**Table 8-611: 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	0x3e
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	Configuration dependent
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	0b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	0x43B

### 8.3.10.5 cmn\_hns\_mpam\_aeidr

MPAM architecture ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1020

##### Type

RO

##### Reset value

See individual bit resets

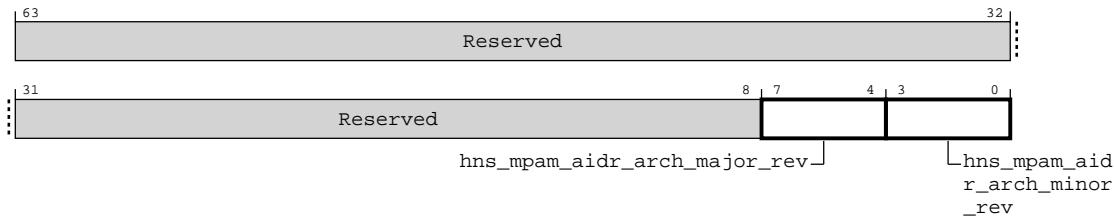
##### Usage constraints

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 8-605: cmn\_hns\_mpam\_aidr**



**Table 8-612: 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	0b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	0b0001

### 8.3.10.6 cmn\_hns\_ns\_mpam\_impl\_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1028

##### Type

RO

##### Reset value

See individual bit resets

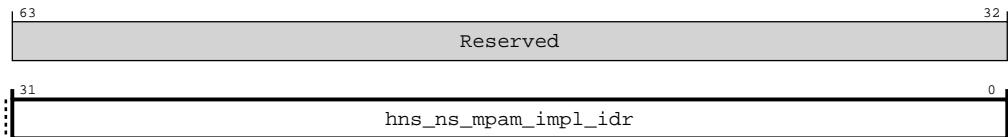
##### Usage constraints

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 8-606: cmn\_hns\_ns\_mpam\_impl\_idr**



**Table 8-613: 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	0x00000000

### 8.3.10.7 cmn\_hns\_ns\_mpam\_cpor\_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1030

##### Type

RO

##### Reset value

See individual bit resets

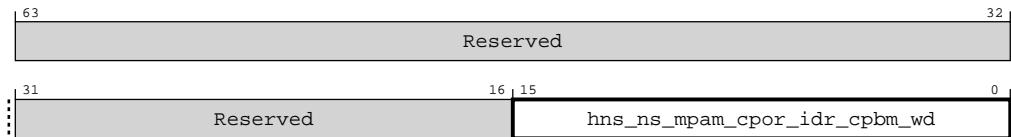
#### Usage constraints

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 8-607: cmn\_hns\_ns\_mpam\_cpor\_idr**



**Table 8-614: 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	0x10

### 8.3.10.8 cmn\_hns\_ns\_mpam\_ccap\_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1038

##### Type

RO

##### Reset value

See individual bit resets

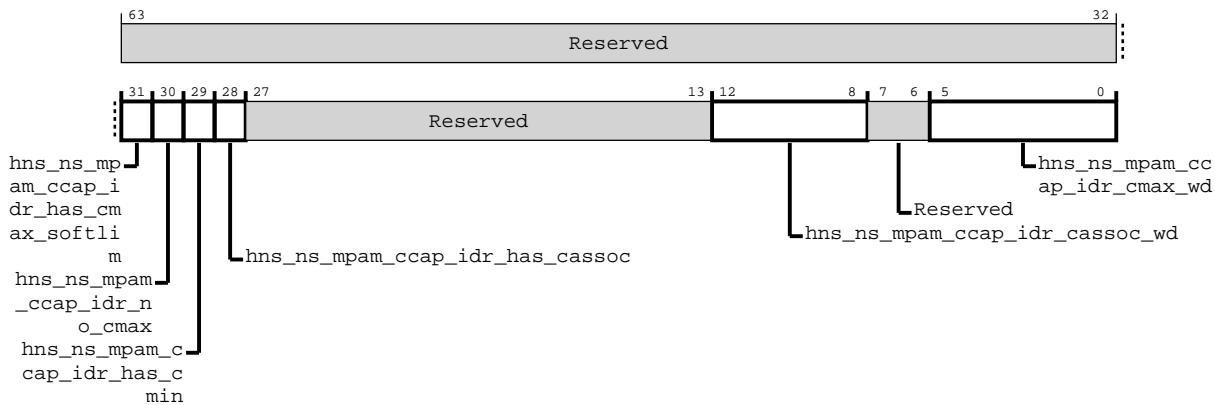
#### Usage constraints

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 8-608: cmn\_hns\_ns\_mpam\_ccap\_idr**



**Table 8-615: cmn\_hns\_ns\_mpam\_ccap\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpam_ccap_idr_has_cassoc	<p><b>0</b> HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit</p> <p><b>1</b> HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit</p>	RO	0x0
[30]	hns_ns_mpam_ccap_idr_no_cmax	<p><b>0</b> HN-F support MPAMCFG_CMAX</p> <p><b>1</b> HN-F doesn't support MPAMCFG_CMAX</p>	RO	0x0
[29]	hns_ns_mpam_ccap_idr_has_cmin	<p><b>0</b> HN-F does not support MPAMCFG_CMIN</p> <p><b>1</b> HN-F supports MPAMCFG_CMIN</p>	RO	0x0
[28]	hns_ns_mpam_ccap_idr_cassoc_wd	<p><b>0</b> HN-F does not support MPAMCFG_CASSOC</p> <p><b>1</b> HN-F supports MPAMCFG_CASSOC</p>	RO	0x0
[27:13]	Reserved	Reserved	RO	-
[12:8]	hns_ns_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	0x0
[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	0x7

### 8.3.10.9 cmn\_hns\_ns\_mpam\_mbw\_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1040

##### Type

RO

##### Reset value

See individual bit resets

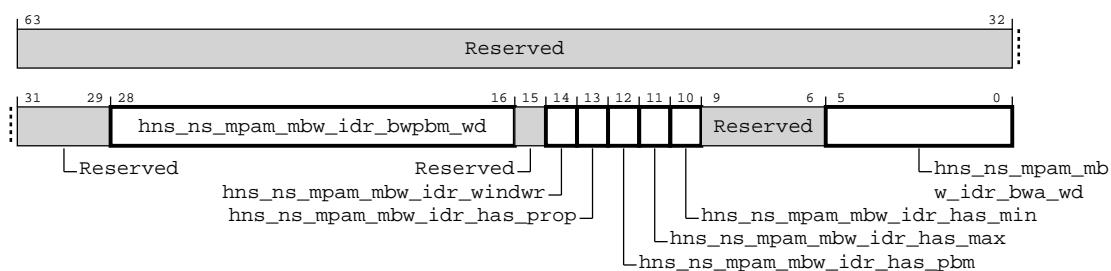
##### Usage constraints

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 8-609: cmn\_hns\_ns\_mpam\_mbw\_idr**



**Table 8-616: 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	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14]	hns_ns_mpam_mbw_idr_windwr	<p><b>0</b> The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed.</p> <p><b>1</b> The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.</p>	RO	0x0
[13]	hns_ns_mpam_mbw_idr_has_prop	<p><b>0</b> There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register</p> <p><b>1</b> MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.</p>	RO	0x0
[12]	hns_ns_mpam_mbw_idr_has_pbm	<p><b>0</b> There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register</p> <p><b>1</b> MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.</p>	RO	0x0
[11]	hns_ns_mpam_mbw_idr_has_max	<p><b>0</b> There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register</p> <p><b>1</b> MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[10]	hns_ns_mpam_mbw_idr_has_min	<p><b>0</b> There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register</p> <p><b>1</b> MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_ns_mpam_mbw_idr_bwa_wd	Number of implemented bits in bandwidth allocation <b>fields</b> MIN, MAX, and STRIDE. Value must be between 1 to 16	RO	0b0000

### 8.3.10.10 cmn\_hns\_ns\_mpam\_pri\_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1048

### Type

RO

### Reset value

See individual bit resets

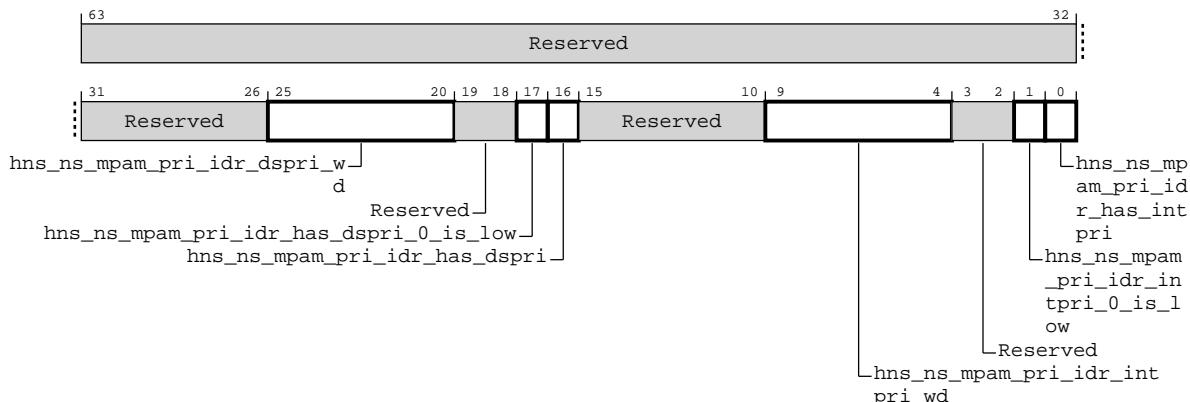
### Usage constraints

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 8-610: cmn\_hns\_ns\_mpam\_pri\_idr**



**Table 8-617: cmn\_hns\_ns\_mpam\_pri\_idr attributes**

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	
[25:20]	hns_ns_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_ns_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	0x0
[15:10]	Reserved			
[9:4]	Reserved			
[3:2]	hns_ns_mpam_pri_idr_in_tpri_0_is_low			
[1:0]	hns_ns_mpam_pri_idr_int_pri_wd			

Bits	Name	Description	Type	Reset
[16]	hns_ns_mpam_pri_idr_has_dspri	<p><b>0</b> This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports downstream priority and has an DSPRI field.</p>	RO	0x0
[15:10]	Reserved	Reserved	RO	
[9:4]	hns_ns_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	0x0
[3:2]	Reserved	Reserved	RO	
[1]	hns_ns_mpam_pri_idr_intpri_0_is_low	<p><b>0</b> In the INTPRI field, a value of 0 means highest priority.</p> <p><b>1</b> In the INTPRI field, a value of 0 means lowest priority.</p>	RO	0x0
[0]	hns_ns_mpam_pri_idr_has_intpri	<p><b>0</b> This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports internal priority and has an INTPRI field.</p>	RO	0x0

### 8.3.10.11 cmn\_hns\_ns\_mpam\_partid\_nrw\_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1050

##### Type

RO

##### Reset value

See individual bit resets

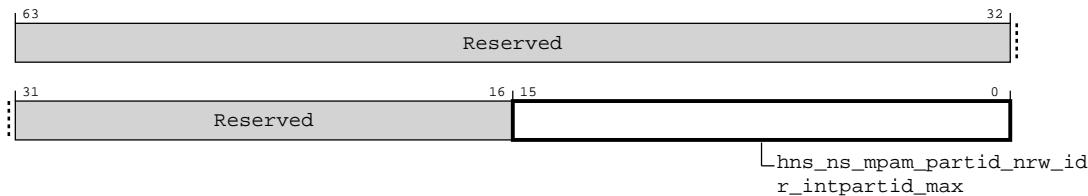
##### Usage constraints

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 8-611: cmn\_hns\_ns\_mpam\_partid\_nrw\_idr**



**Table 8-618: 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	0x00

### 8.3.10.12 cmn\_hns\_ns\_mpam\_msmon\_idr

MPAM performance monitoring ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1080

##### Type

RO

##### Reset value

See individual bit resets

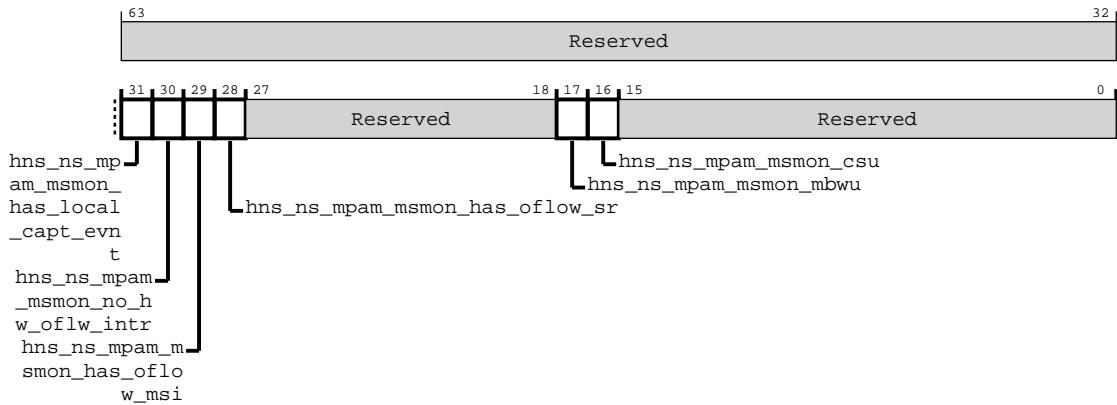
##### Usage constraints

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 8-612: cmn\_hns\_ns\_mpam\_msmon\_idr**



**Table 8-619: cmn\_hns\_ns\_mpam\_msmon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpam_msmon_has_local_capt_evt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	0x1
[30]	hns_ns_mpam_msmon_no_hw_oflw_intr	<p><b>0</b> HNF has hardwired MPAM overflow interrupt</p> <p><b>1</b> HNF doesn't have hardwired MPAM overflow interrupt</p>	RO	0b1
[29]	hns_ns_mpam_msmon_has_oflow_msi	<p><b>0</b> HNF doesn't have support for MSI writes to signal MPAM monitor overflow interrupt</p> <p><b>1</b> HNF has support for MSI writes to signal MPAM monitor overflow interrupt</p>	RO	0b0
[28]	hns_ns_mpam_msmon_has_oflow_sr	<p><b>0</b> HNF doesn't have overflow status register</p> <p><b>1</b> HNF has overflow status register</p>	RO	0b0
[27:18]	Reserved	Reserved	RO	-
[17]	hns_ns_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	0x0
[16]	hns_ns_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

### 8.3.10.13 cmn\_hns\_ns\_mpam\_csumon\_idr

MPAM cache storage usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1088

##### Type

RO

##### Reset value

See individual bit resets

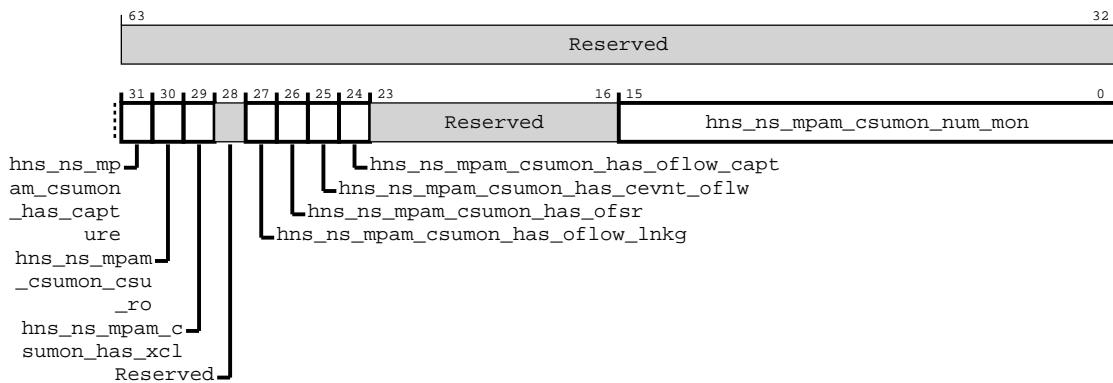
##### Usage constraints

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 8-613: cmn\_hns\_ns\_mpam\_csumon\_idr**



**Table 8-620: cmn\_hns\_ns\_mpam\_csumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_ns_mpam_csumon_has_capture	<p><b>0</b> MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature.</p> <p><b>1</b> This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.</p>	RO	0x1
[30]	hns_ns_mpam_csumon_csu_ro	<p><b>0</b> MSMON_CSU is read/write.</p> <p><b>1</b> MSMON_CSU is read-only.</p>	RO	0b0
[29]	hns_ns_mpam_csumon_has_xcl	<p><b>0</b> MSMON_CFG_CSU_FLT does not implement the XCL field</p> <p><b>1</b> MSMON_CFG_CSU_FLT implements the XCL field</p>	RO	0b0
[28]	Reserved	Reserved	RO	-
[27]	hns_ns_mpam_csumon_has_oflow_lnk	<p><b>0</b> HNF doesn't support CSU overflow linkage</p> <p><b>1</b> HNF supports CSU overflow linkage</p>	RO	0b0
[26]	hns_ns_mpam_csumon_has_ofsr	<p><b>0</b> MSMON_CSU_OFSR register is not implemented</p> <p><b>1</b> MSMON_CSU_OFSR register is implemented</p>	RO	0b0
[25]	hns_ns_mpam_csumon_has_cevnt_oflw	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW</p>	RO	0b0
[24]	hns_ns_mpam_csumon_has_oflow_capt	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT</p>	RO	0b0
[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

### 8.3.10.14 cmn\_hns\_ns\_mpam\_mbwumon\_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1090

##### Type

RO

##### Reset value

See individual bit resets

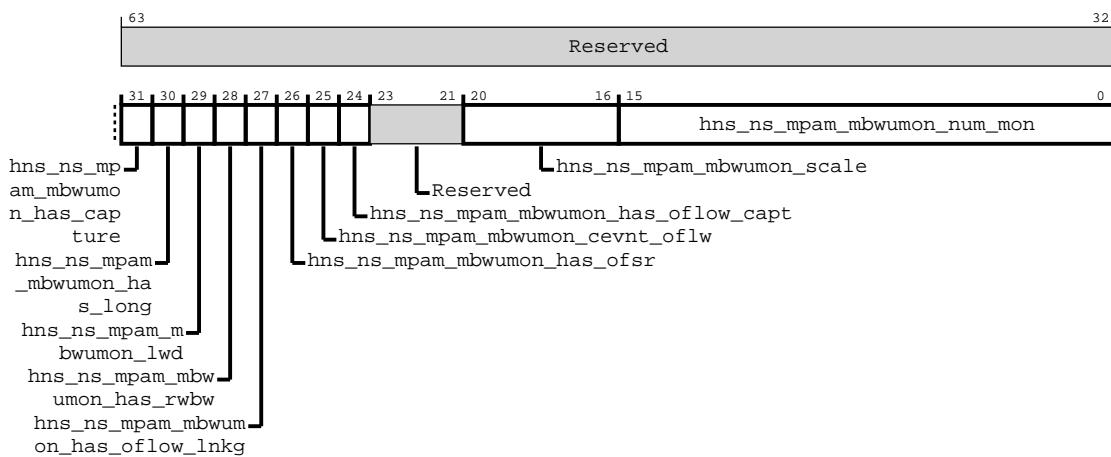
##### Usage constraints

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 8-614: cmn\_hns\_ns\_mpam\_mbwumon\_idr**



**Table 8-621: cmn\_hns\_ns\_mpam\_mbwumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	hns_ns_mpam_mbwumon_has_capture	<p><b>0</b> MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature.</p> <p><b>1</b> This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.</p>	RO	0x0
[30]	hns_ns_mpam_mbwumon_has_long	<p><b>0</b> MSMON_MBWU_L is not implemented.</p> <p><b>1</b> MSMON_MBWU_L is implemented.</p>	RO	0b0
[29]	hns_ns_mpam_mbwumon_lwd	<p><b>0</b> MSMON_MBWU_L has 44-bit VALUE field in bits [43:0].</p> <p><b>1</b> MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].</p>	RO	0b0
[28]	hns_ns_mpam_mbwumon_has_rwbw	<p><b>0</b> Read/write bandwidth selection is not implemented.</p> <p><b>1</b> Read/write bandwidth selection is implemented.</p>	RO	0b0
[27]	hns_ns_mpam_mbwumon_has_oflow_lnkg	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p>	RO	0b0
[26]	hns_ns_mpam_mbwumon_has_ofsr	<p><b>0</b> MSMON_MBWU_OFSR register is not implemented</p> <p><b>1</b> MSMON_MBWU_OFSR register is implemented</p>	RO	0b0
[25]	hns_ns_mpam_mbwumon_cevnt_oflw	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p>	RO	0b0
[24]	hns_ns_mpam_mbwumon_has_oflow_capt	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p>	RO	0b0
[23:21]	Reserved	Reserved	RO	
[20:16]	hns_ns_mpam_mbwumon_scale	Scalling of MSMON_MBWU.VALUE in bits.	RO	0x0
[15:0]	hns_ns_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	0x0

### 8.3.10.15 cmn\_hns\_ns\_mpam\_ecr

MPAM Error Control Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x10F0

##### Type

RW

##### Reset value

See individual bit resets

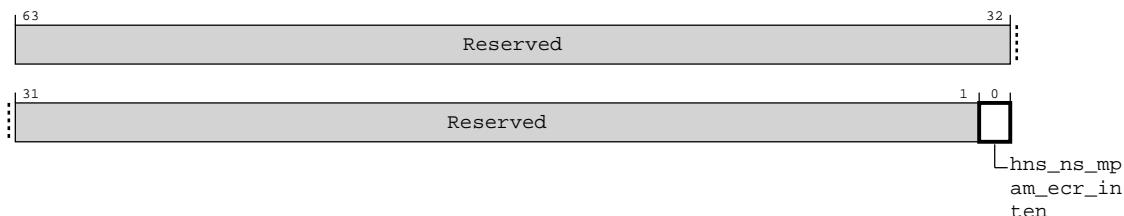
##### Usage constraints

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 8-615: cmn\_hns\_ns\_mpam\_ecr**



**Table 8-622: 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	0x0

### 8.3.10.16 cmn\_hns\_ns\_mpam\_esr

MPAM Error Status Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x10F8

##### Type

RW

##### Reset value

See individual bit resets

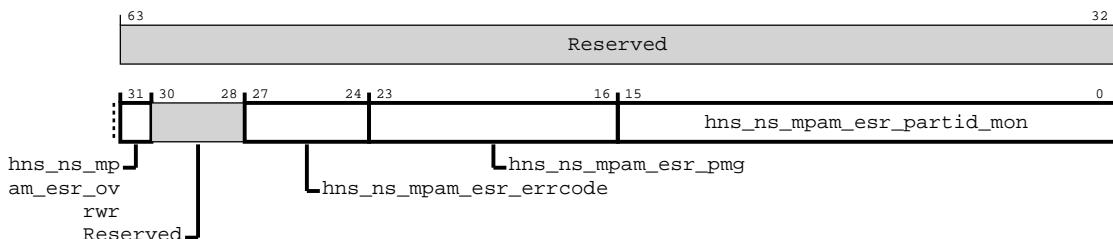
##### Usage constraints

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 8-616: cmn\_hns\_ns\_mpam\_esr**



**Table 8-623: 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	0x0
[30:28]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[27:24]	hns_ns_mpam_esr_errcode	Error code	RW	0x0
[23:16]	hns_ns_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	0x0
[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	0x0

### 8.3.10.17 cmn\_hns\_ns\_mpamcfg\_part\_sel

MPAM partition configuration selection register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1100

##### Type

RW

##### Reset value

See individual bit resets

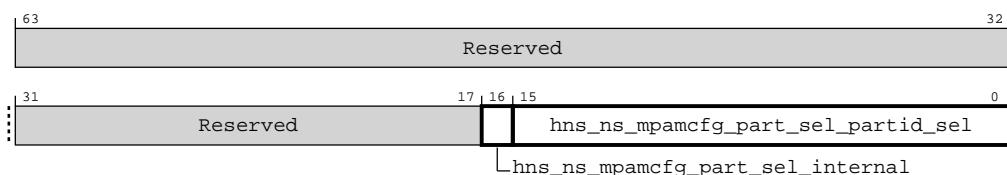
##### Usage constraints

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 8-617: cmn\_hns\_ns\_mpamcfg\_part\_sel**



**Table 8-624: 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 <b>RAZ/WI</b> . If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interpret PARTID_SEL.	RW	0x0
[15:0]	hns_ns_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	0x0

### 8.3.10.18 cmn\_hns\_ns\_mpamcfg\_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1108

##### Type

RW

##### Reset value

See individual bit resets

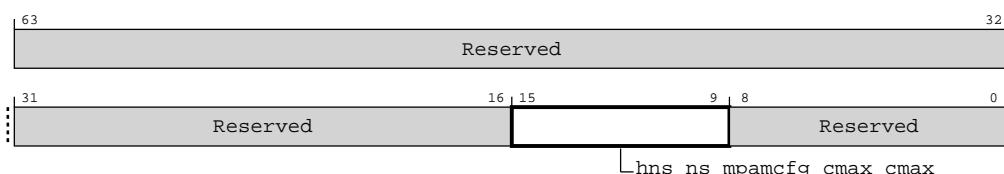
##### Usage constraints

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 8-618: cmn\_hns\_ns\_mpamcfg\_cmax**



**Table 8-625: 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	0b1111111
[8:0]	Reserved	Reserved	RO	-

### 8.3.10.19 cmn\_hns\_ns\_mpamcfg\_mbw\_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1200

##### Type

RW

##### Reset value

See individual bit resets

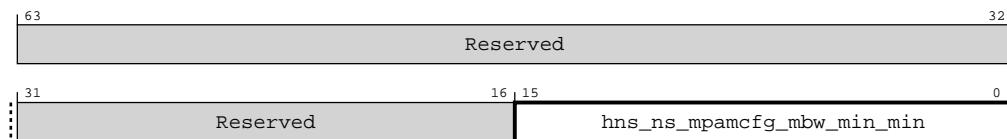
##### Usage constraints

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 8-619: cmn\_hns\_ns\_mpamcfg\_mbw\_min**



**Table 8-626: 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	0x0

### 8.3.10.20 cmn\_hns\_ns\_mpamcfg\_mbw\_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1208

##### Type

RW

##### Reset value

See individual bit resets

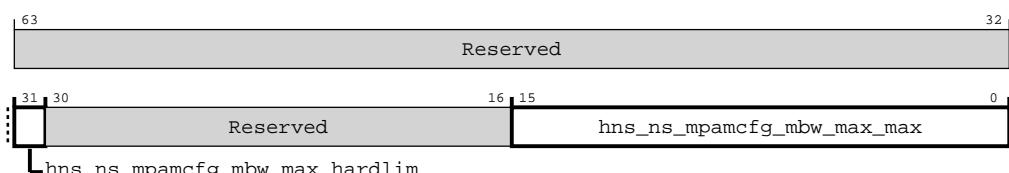
##### Usage constraints

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 8-620: cmn\_hns\_ns\_mpamcfg\_mbw\_max**



**Table 8-627: cmn\_hns\_ns\_mpamcfg\_mbw\_max attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	hns_ns_mpamcfg_mbw_max_hardlim	<p><b>0</b> When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth.</p> <p><b>1</b> When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.</p>	RW	0x0
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_ns_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	0x0

### 8.3.10.21 cmn\_hns\_ns\_mpamcfg\_mbw\_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1220

##### Type

RW

##### Reset value

See individual bit resets

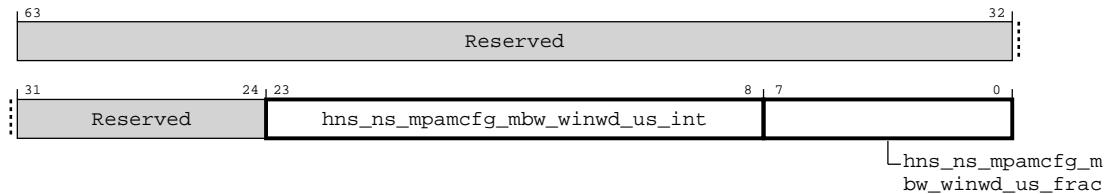
##### Usage constraints

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 8-621: cmn\_hns\_ns\_mpamcfg\_mbw\_winwd**



**Table 8-628: 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	0x0
[7:0]	hns_ns_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	0x0

### 8.3.10.22 cmn\_hns\_ns\_mpamcfg\_pri

MPAM priority partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1400

##### Type

RW

##### Reset value

See individual bit resets

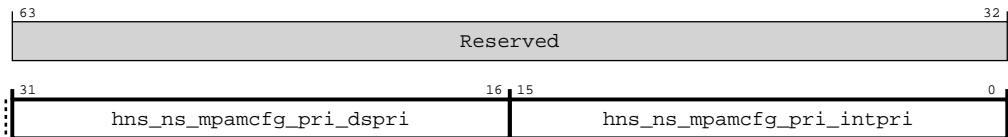
##### Usage constraints

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 8-622: cmn\_hns\_ns\_mpamcfg\_pri**



**Table 8-629: 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	0x0
[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	0x0

### 8.3.10.23 cmn\_hns\_ns\_mpamcfg\_mbw\_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1500

##### Type

RW

##### Reset value

See individual bit resets

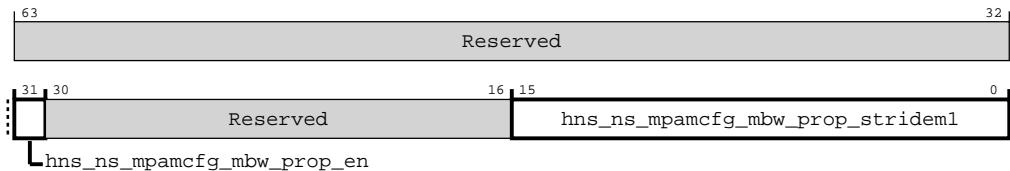
##### Usage constraints

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 8-623: cmn\_hns\_ns\_mpamcfg\_mbw\_prop**



**Table 8-630: cmn\_hns\_ns\_mpamcfg\_mbw\_prop attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_ns_mpamcfg_mbw_prop_en	<p><b>0</b> The selected partition is not regulated by proportional stride bandwidth partitioning.</p> <p><b>1</b> The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.</p>	RW	0x0
[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	0x0

### 8.3.10.24 cmn\_hns\_ns\_mpamcfg\_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1600

##### Type

RW

##### Reset value

See individual bit resets

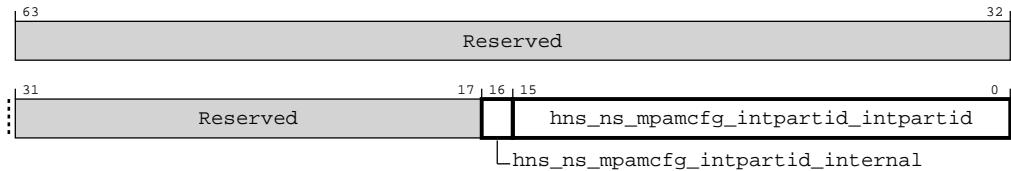
##### Usage constraints

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 8-624: cmn\_hns\_ns\_mpamcfg\_intpartid**



**Table 8-631: cmn\_hns\_ns\_mpamcfg\_intpartid attributes**

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	
[16]	hns_ns_mpamcfg_intpartid_internal	This bit must be 1 when written to the register. If written as 0, the write will not update the reqPARTID to intPARTID association.	RW	0x0
[15:0]	hns_ns_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	0x0

### 8.3.10.25 cmn\_hns\_ns\_msmon\_cfg\_mon\_sel

Memory system performance monitor selection register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1800

##### Type

RW

##### Reset value

See individual bit resets

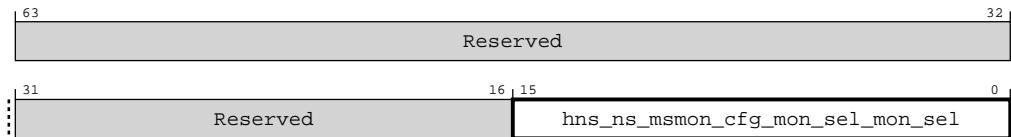
#### Usage constraints

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 8-625: cmn\_hns\_ns\_msmon\_cfg\_mon\_sel**



**Table 8-632: 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	0x0

### 8.3.10.26 cmn\_hns\_ns\_msmon\_capt\_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1808

##### Type

RW

##### Reset value

See individual bit resets

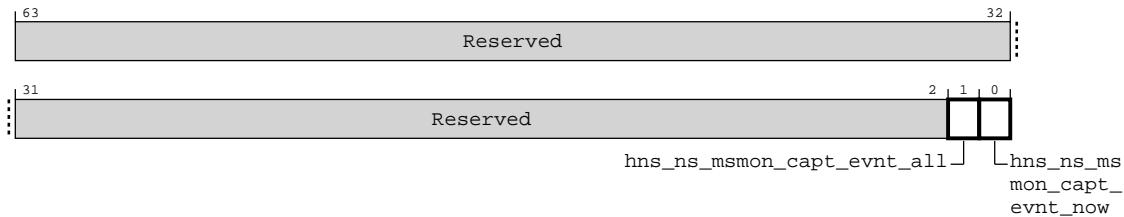
##### Usage constraints

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 8-626: cmn\_hns\_ns\_msmon\_capt\_evnt**



**Table 8-633: cmn\_hns\_ns\_msmon\_capt\_evnt attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	hns_ns_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version all bits are RAZ/WI. 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	0x0
[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	0x0

### 8.3.10.27 cmn\_hns\_ns\_msmon\_cfg\_csu\_filt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1810

##### Type

RW

### Reset value

See individual bit resets

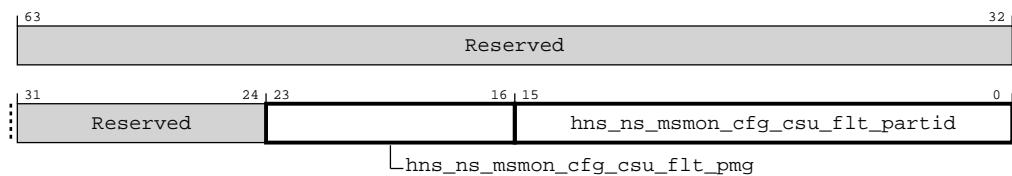
### Usage constraints

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 8-627: cmn\_hns\_ns\_msmon\_cfg\_csu\_flt**



**Table 8-634: 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	0x0
[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	0x0

### 8.3.10.28 cmn\_hns\_ns\_msmon\_cfg\_csu\_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1818

## Type

RW

## Reset value

See individual bit resets

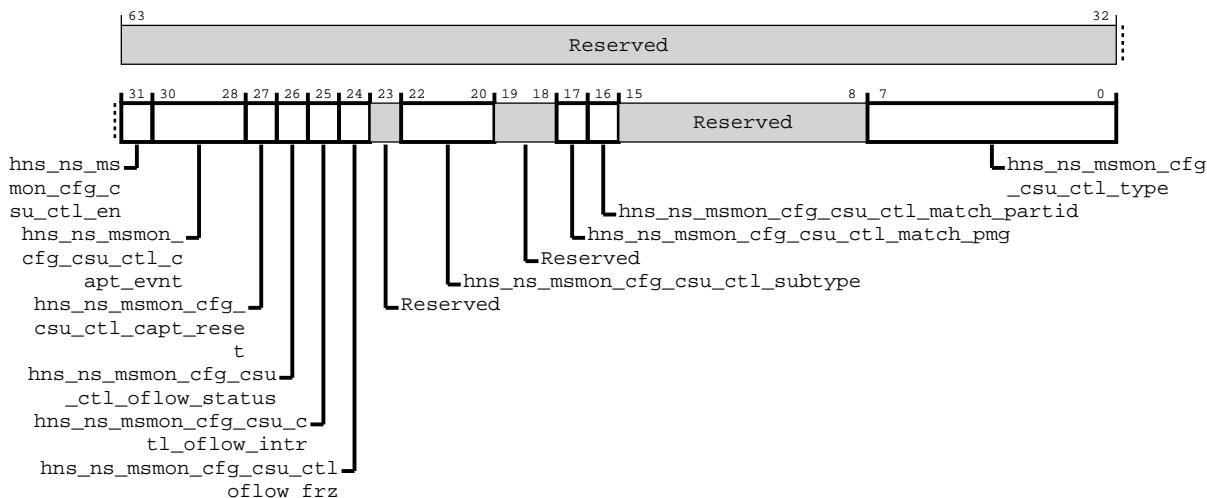
## Usage constraints

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 8-628: cmn\_hns\_ns\_msmon\_cfg\_csu\_ctl**



**Table 8-635: cmn\_hns\_ns\_msmon\_cfg\_csu\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_ns_msmon_cfg_csu_ctl_en	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_ns_msmon_cfg_csu_ctl_capt_evt	Select the event that triggers capture from the following:	RW	0x0
		<p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>		
[27]	hns_ns_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	0x0

Bits	Name	Description	Type	Reset
[26]	hns_ns_msmon_cfg_csu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0
[25]	hns_ns_msmon_cfg_csu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_ns_msmon_cfg_csu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[23]	Reserved	Reserved	RO	
[22:20]	hns_ns_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_ns_msmon_cfg_csu_ctl_match_pmg	<p><b>0</b> Monitor storage used by all PMG values.</p> <p><b>1</b> Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0
[16]	hns_ns_msmon_cfg_csu_ctl_match_partid	<p><b>0</b> Monitor storage used by all PARTIDs.</p> <p><b>1</b> Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_ns_msmon_cfg_csu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	0x43

### 8.3.10.29 cmn\_hns\_ns\_msmon\_cfg\_mbwu\_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1820

### Type

RW

### Reset value

See individual bit resets

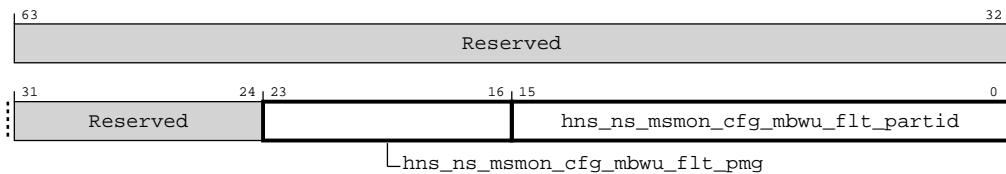
### Usage constraints

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 8-629: cmn\_hns\_ns\_msmon\_cfg\_mbwu\_flt**



**Table 8-636: cmn\_hns\_ns\_msmon\_cfg\_mbwu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_ns_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0
[15:0]	hns_ns_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0

### 8.3.10.30 cmn\_hns\_ns\_msmon\_cfg\_mbwu\_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1828

##### Type

RW

##### Reset value

See individual bit resets

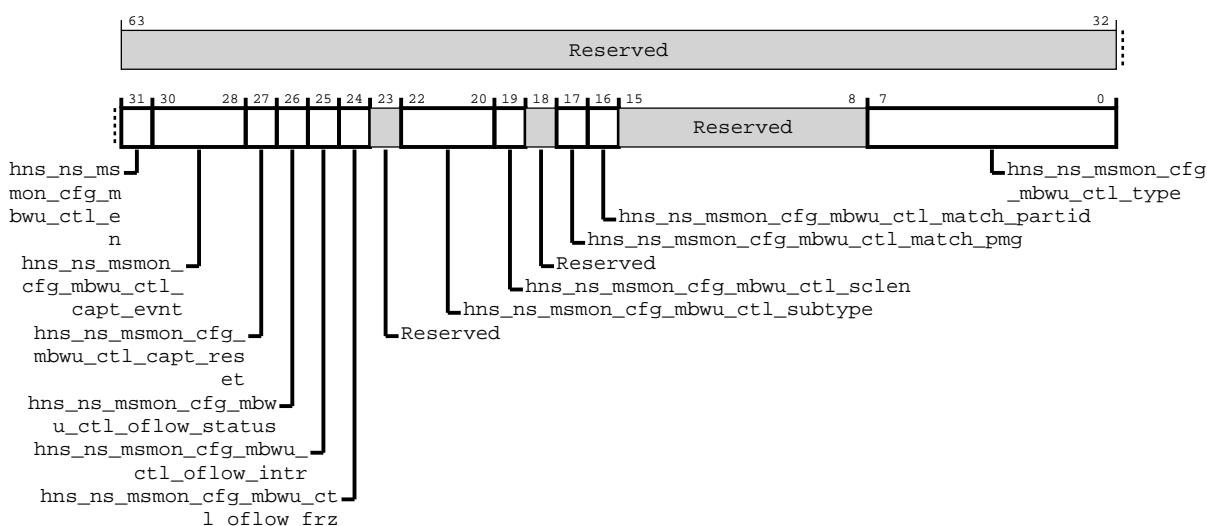
##### Usage constraints

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 8-630: cmn\_hns\_ns\_msmon\_cfg\_mbwu\_ctl**



**Table 8-637: 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	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_ns_msmon_cfg_mbwu_ctl_capt_evnt	<p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>	RW	0x0
[27]	hns_ns_msmon_cfg_mbwu_ctl_capt_reset	<p><b>0</b> Monitor is not reset on capture.</p> <p><b>1</b> Monitor is reset on capture.</p>	RW	0x0
[26]	hns_ns_msmon_cfg_mbwu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0
[25]	hns_ns_msmon_cfg_mbwu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_ns_msmon_cfg_mbwu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[23]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[22:20]	hns_ns_msmon_cfg_mbwu_ctl_subtype	<p>A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports:</p> <ul style="list-style-type: none"> <li><b>0</b> Do not count any bandwidth.</li> <li><b>1</b> Count bandwidth used by memory reads</li> <li><b>2</b> Count bandwidth used by memory writes</li> <li><b>3</b> 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 <b>UNPREDICTABLE</b>.</li> </ul>	RW	0x0
[19]	hns_ns_msmon_cfg_mbwu_ctl_sclen	<ul style="list-style-type: none"> <li><b>0</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance.</li> <li><b>1</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.</li> </ul>	RW	0x0
[18]	Reserved	Reserved	RO	
[17]	hns_ns_msmon_cfg_mbwu_ctl_match_pmg	<ul style="list-style-type: none"> <li><b>0</b> Monitor bandwidth used by all PMG values.</li> <li><b>1</b> Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</li> </ul>	RW	0x0
[16]	hns_ns_msmon_cfg_mbwu_ctl_match_partid	<ul style="list-style-type: none"> <li><b>0</b> Monitor bandwidth used by all PARTIDs.</li> <li><b>1</b> Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.</li> </ul>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_ns_msmon_cfg_mbwu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	0x42

### 8.3.10.31 cmn\_hns\_ns\_msmon\_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1840

### Type

RW

### Reset value

See individual bit resets

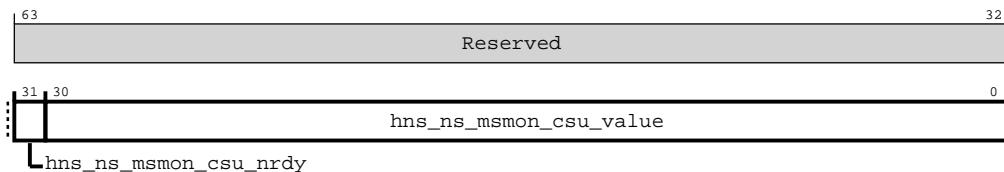
### Usage constraints

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 8-631: cmn\_hns\_ns\_msmon\_csu**



**Table 8-638: 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	0x0
[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	0x0

### 8.3.10.32 cmn\_hns\_ns\_msmon\_csu\_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1848

### Type

RW

### Reset value

See individual bit resets

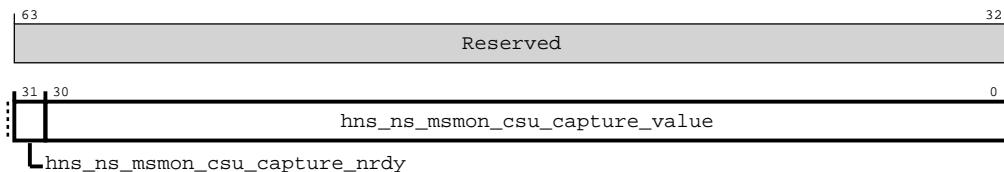
### Usage constraints

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 8-632: cmn\_hns\_ns\_msmon\_csu\_capture**



**Table 8-639: 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	0x0
[30:0]	hns_ns_msmon_csu_capture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	0x0

### 8.3.10.33 cmn\_hns\_ns\_msmon\_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1860

### Type

RW

### Reset value

See individual bit resets

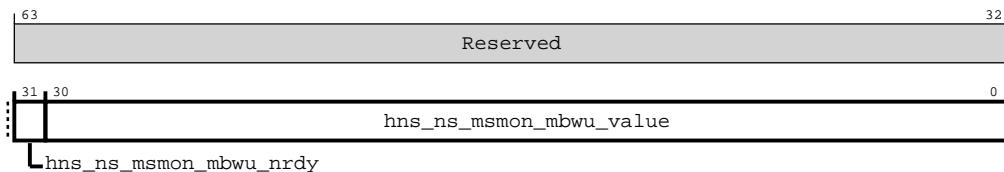
### Usage constraints

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 8-633: cmn\_hns\_ns\_msmon\_mbwu**



**Table 8-640: 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	0x0
[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	0x0

### 8.3.10.34 cmn\_hns\_ns\_msmon\_mbwu\_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1868

### Type

RW

### Reset value

See individual bit resets

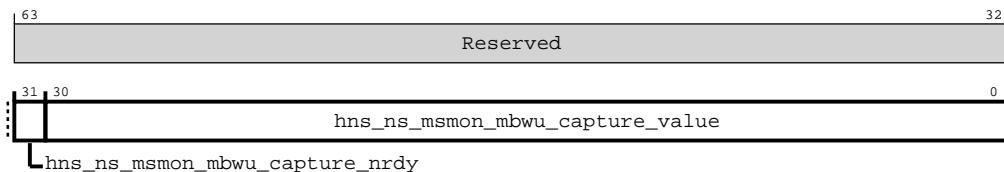
### Usage constraints

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 8-634: cmn\_hns\_ns\_msmon\_mbwu\_capture**



**Table 8-641: 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	0x0
[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	0x0

### 8.3.10.35 cmn\_hns\_ns\_mpamcfg\_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x2000

### Type

RW

### Reset value

See individual bit resets

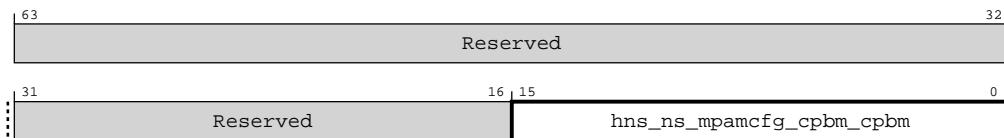
### Usage constraints

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 8-635: cmn\_hns\_ns\_mpamcfg\_cpbm**



**Table 8-642: cmn\_hns\_ns\_mpamcfg\_cpbm attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL.	RW	0xFFFF

### 8.3.10.36 cmn\_hns\_rl\_mpam\_idr

MPAM features ID register. This is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

## Address offset

0x8000

## Type

RO

## Reset value

See individual bit resets

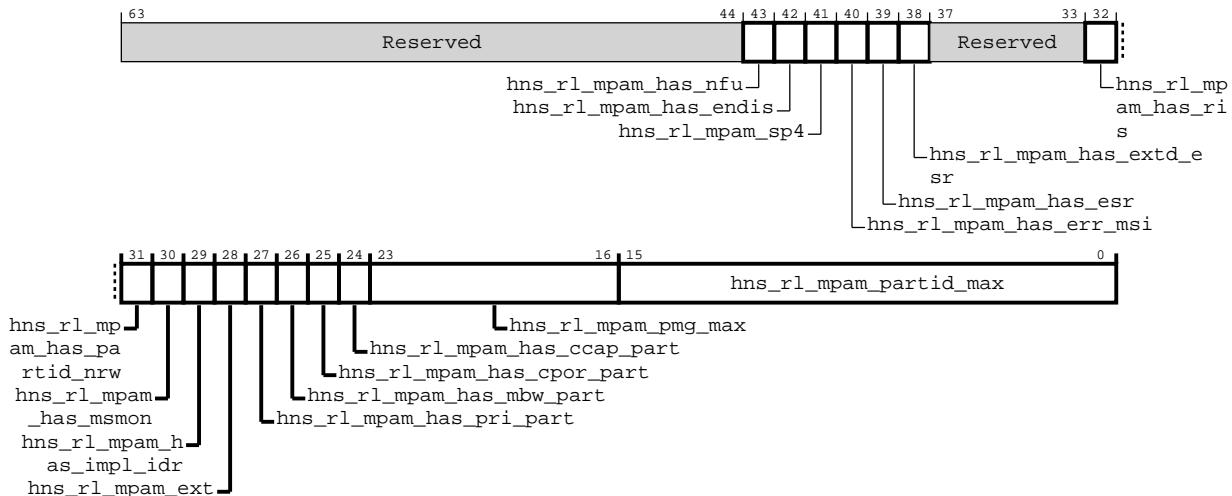
## Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-636: cmn\_hns\_rl\_mpam\_idr**



**Table 8-643: cmn\_hns\_rl\_mpam\_idr attributes**

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_rl_mpam_has_nfu	<p><b>0</b> HN-F does not support no future use field</p> <p><b>1</b> HN-F supports no future use field</p>	RO	0b0
[42]	hns_rl_mpam_has_endis	<p><b>0</b> HN-F does not support PARTID enable and disable functionality</p> <p><b>1</b> HN-F supports PARTID enable and disable functionality</p>	RO	0b0

Bits	Name	Description	Type	Reset
[41]	hns_rl_mpam_sp4	<p><b>0</b> HN-F supports two PARTID spaces</p> <p><b>1</b> HN-F supports four PARTID spaces</p>	RO	0b1
[40]	hns_rl_mpam_has_err_msi	<p><b>0</b> HN-F does not support MSI writes to signal MPAM error interrupt</p> <p><b>1</b> HN-F supports MSI writes to signal MPAM error interrupt</p>	RO	0b0
[39]	hns_rl_mpam_has_esr	<p><b>0</b> HN-F does not support MPAM error handling</p> <p><b>1</b> HN-F supports MPAM error handling</p>	RO	0b1
[38]	hns_rl_mpam_has_extd_esr	<p><b>0</b> MPAMF_ESR is 32 bits</p> <p><b>1</b> MPAMF_ESR is 64 bits</p>	RO	0b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_rl_mpam_has_ris	<p><b>0</b> HN-F does not support MPAM resource instance selector</p> <p><b>1</b> HN-F supports MPAM resource instance selector</p>	RO	0b0
[31]	hns_rl_mpam_has_partid_nrw	<p><b>0</b> HN-F does not support MPAM PARTID Narrowing</p> <p><b>1</b> HN-F supports MPAM PARTID Narrowing</p>	RO	Configuration dependent
[30]	hns_rl_mpam_has_msmon	<p><b>0</b> MPAM performance monitoring is not supported</p> <p><b>1</b> MPAM performance monitoring is supported</p>	RO	Configuration dependent
[29]	hns_rl_mpam_has_impl_idr	<p><b>0</b> MPAM implementation specific partitioning features not supported</p> <p><b>1</b> MPAM implementation specific partitioning features supported</p>	RO	Configuration dependent
[28]	hns_rl_mpam_ext	<p><b>0</b> HN-F has no defined bits in [63:32]</p> <p><b>1</b> HN-F has bits defined in [63:32]</p>	RO	0b1

Bits	Name	Description	Type	Reset
[27]	hns_rl_mpam_has_pri_part	<p><b>0</b> MPAM priority partitioning is not supported</p> <p><b>1</b> MPAM priority partitioning is supported</p>	RO	0x0
[26]	hns_rl_mpam_has_mbw_part	<p><b>0</b> MPAM memory bandwidth partitioning is not supported</p> <p><b>1</b> MPAM memory bandwidth partitioning is supported</p>	RO	0x0
[25]	hns_rl_mpam_has_cpor_part	<p><b>0</b> MPAM cache portion partitioning is not supported</p> <p><b>1</b> MPAM cache portion partitioning is supported</p>	RO	Configuration dependent
[24]	hns_rl_mpam_has_ccap_part	<p><b>0</b> MPAM cache maximum capacity partitioning is not supported</p> <p><b>1</b> MPAM cache maximum capacity partitioning is supported</p>	RO	Configuration dependent
[23:16]	hns_rl_mpam_pmg_max	Maximum value of realm PMG supported by this HN-F	RO	0x2
[15:0]	hns_rl_mpam_partid_max	Maximum value of realm PARTID supported by this HN-F	RO	0x1

### 8.3.10.37 cmn\_hns\_rl\_mpam\_iidr

MPAM Implementation ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8018

##### Type

RO

##### Reset value

See individual bit resets

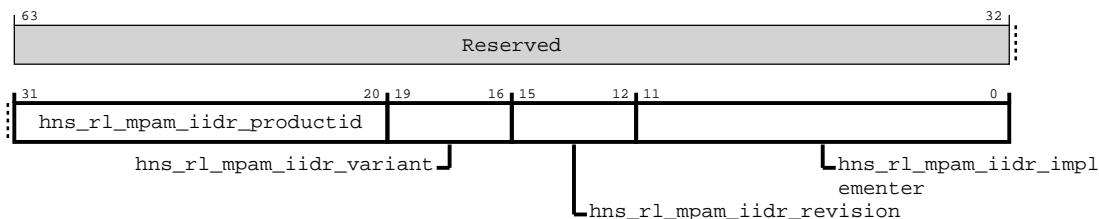
##### Usage constraints

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 8-637: cmn\_hns\_rl\_mpam\_iidr**



**Table 8-644: cmn\_hns\_rl\_mpam\_iidr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_rl_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	0x3e
[19:16]	hns_rl_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	Configuration dependent
[15:12]	hns_rl_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	0b0000
[11:0]	hns_rl_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	0x43B

### 8.3.10.38 cmn\_hns\_rl\_mpam\_aidr

MPAM architecture ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8020

##### Type

RO

##### Reset value

See individual bit resets

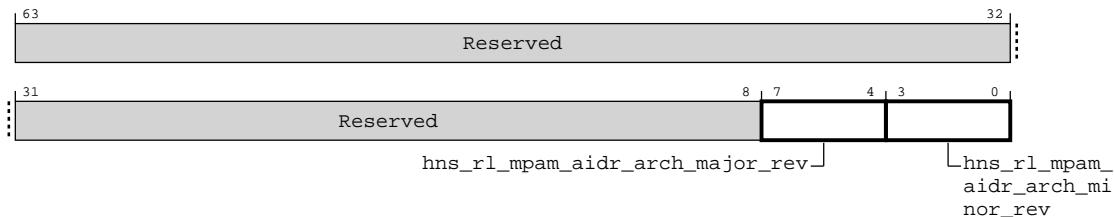
## Usage constraints

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 8-638: cmn\_hns\_rl\_mpam\_aistr**



**Table 8-645: cmn\_hns\_rl\_mpam\_aistr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_rl_mpam_aistr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	0b0001
[3:0]	hns_rl_mpam_aistr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	0b0001

### 8.3.10.39 cmn\_hns\_rl\_mpam\_impl\_idr

MPAM Implementation defined partitioning feature ID register. This is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8028

### Type

RO

### Reset value

See individual bit resets

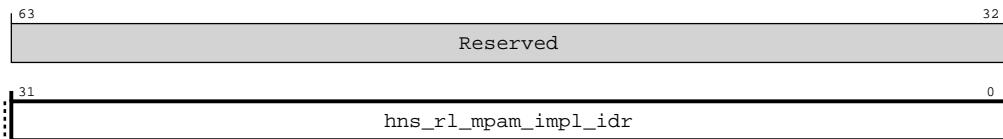
## Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-639: cmn\_hns\_rl\_mpam\_impl\_idr**



**Table 8-646: cmn\_hns\_rl\_mpam\_impl\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_rl_mpam_impl_idr	Implementation defined partitioning features.	RO	0x00000000

## 8.3.10.40 cmn\_hns\_rl\_mpam\_cpor\_idr

MPAM cache portion partitioning ID register. This is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8030

### Type

RO

### Reset value

See individual bit resets

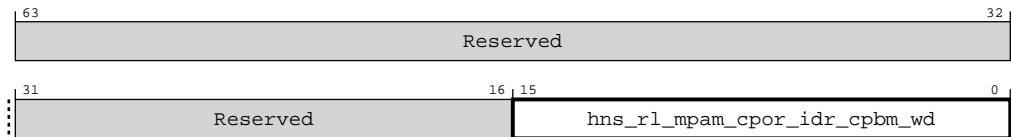
## Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-640: cmn\_hns\_rl\_mpam\_cpor\_idr**



**Table 8-647: cmn\_hns\_rl\_mpam\_cpor\_idr attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	0x10

### 8.3.10.41 cmn\_hns\_rl\_mpam\_ccap\_idr

MPAM cache capacity partitioning ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8038

##### Type

RO

##### Reset value

See individual bit resets

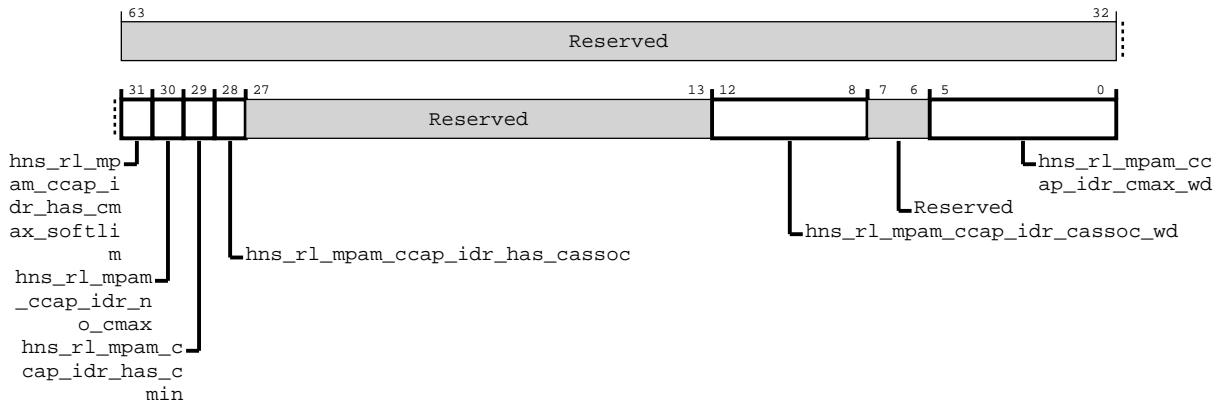
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-641: cmn\_hns\_rl\_mpam\_ccap\_idr**



**Table 8-648: cmn\_hns\_rl\_mpam\_ccap\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_mpam_ccap_idr_has_cmax_softlim	<p><b>0</b> HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit</p> <p><b>1</b> HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit</p>	RO	0x0
[30]	hns_rl_mpam_ccap_idr_no_cmax	<p><b>0</b> HN-F support MPAMCFG_CMAX</p> <p><b>1</b> HN-F doesn't support MPAMCFG_CMAX</p>	RO	0x0
[29]	hns_rl_mpam_ccap_idr_has_cmin	<p><b>0</b> HN-F does not support MPAMCFG_CMIN</p> <p><b>1</b> HN-F supports MPAMCFG_CMIN</p>	RO	0x0
[28]	hns_rl_mpam_ccap_idr_has_cassoc	<p><b>0</b> HN-F does not support MPAMCFG_CASSOC</p> <p><b>1</b> HN-F supports MPAMCFG_CASSOC</p>	RO	0x0
[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	0x0
[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	0x7

### 8.3.10.42 cmn\_hns\_rl\_mpam\_mbw\_idr

MPAM Memory Bandwidth partitioning ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8040

##### Type

RO

##### Reset value

See individual bit resets

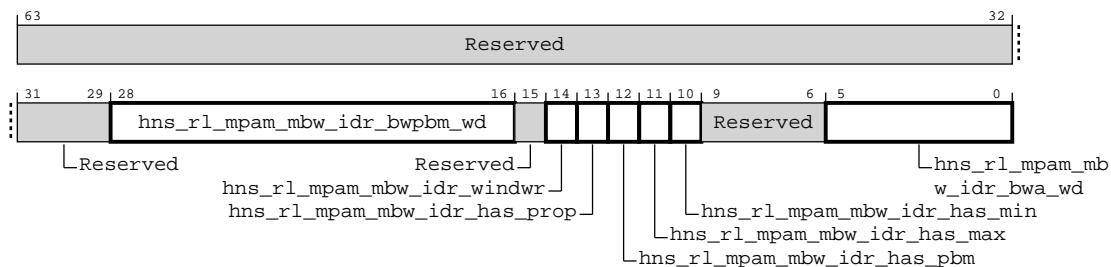
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-642: cmn\_hns\_rl\_mpam\_mbw\_idr**



**Table 8-649: 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	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14]	hns_rl_mpam_mbw_idr_windwr	<p><b>0</b> The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed.</p> <p><b>1</b> The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.</p>	RO	0x0
[13]	hns_rl_mpam_mbw_idr_has_prop	<p><b>0</b> There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register</p> <p><b>1</b> MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.</p>	RO	0x0
[12]	hns_rl_mpam_mbw_idr_has_pbm	<p><b>0</b> There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register</p> <p><b>1</b> MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.</p>	RO	0x0
[11]	hns_rl_mpam_mbw_idr_has_max	<p><b>0</b> There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register</p> <p><b>1</b> MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[10]	hns_rl_mpam_mbw_idr_has_min	<p><b>0</b> There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register</p> <p><b>1</b> MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_rl_mpam_mbw_idr_bwa_wd	<p>Number of implemented bits in bandwidth allocation</p> <p><b>fields</b></p> <p>MIN, MAX, and STRIDE. Value must be between 1 to 16</p>	RO	0b0000

### 8.3.10.43 cmn\_hns\_rl\_mpam\_pri\_idr

MPAM Priority partitioning ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8048

### Type

RO

### Reset value

See individual bit resets

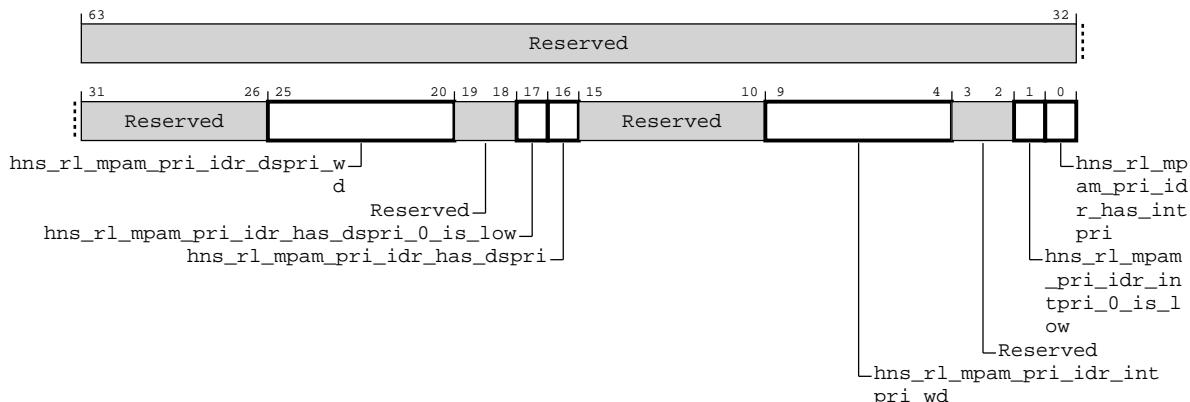
### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-643: cmn\_hns\_rl\_mpam\_pri\_idr**



**Table 8-650: cmn\_hns\_rl\_mpam\_pri\_idr attributes**

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	
[25:20]	hns_rl_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_rl_mpam_pri_idr_has_dspri_0_is_low	<p><b>0</b> In the DSPRI field, a value of 0 means highest priority.</p> <p><b>1</b> In the DSPRI field, a value of 0 means lowest priority.</p>	RO	0x0

Bits	Name	Description	Type	Reset
[16]	hns_rl_mpam_pri_idr_has_dspri	<p><b>0</b> This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports downstream priority and has an DSPRI field.</p>	RO	0x0
[15:10]	Reserved	Reserved	RO	
[9:4]	hns_rl_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	0x0
[3:2]	Reserved	Reserved	RO	
[1]	hns_rl_mpam_pri_idr_intpri_0_is_low	<p><b>0</b> In the INTPRI field, a value of 0 means highest priority.</p> <p><b>1</b> In the INTPRI field, a value of 0 means lowest priority.</p>	RO	0x0
[0]	hns_rl_mpam_pri_idr_has_intpri	<p><b>0</b> This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports internal priority and has an INTPRI field.</p>	RO	0x0

### 8.3.10.44 cmn\_hns\_rl\_mpam\_partid\_nrw\_idr

MPAM PARTID narrowing ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8050

##### Type

RO

##### Reset value

See individual bit resets

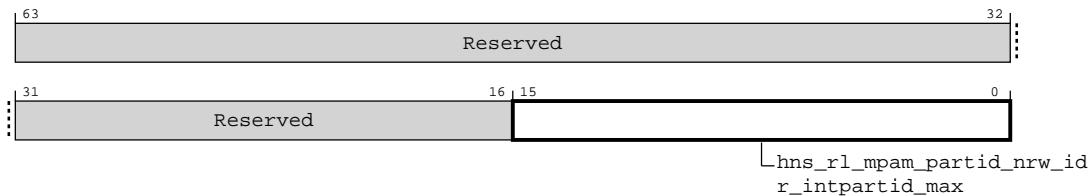
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-644: cmn\_hns\_rl\_mpam\_partid\_nrw\_idr**



**Table 8-651: cmn\_hns\_rl\_mpam\_partid\_nrw\_idr attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	<code>hns_rl_mpam_partid_nrw_idr_intpartid_max</code>	This field indicates the largest intPARTID supported in this component.	RO	0x00

### 8.3.10.45 cmn\_hns\_rl\_mpam\_msmon\_idr

MPAM performance monitoring ID register. This is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8080

### Type

RO

### Reset value

See individual bit resets

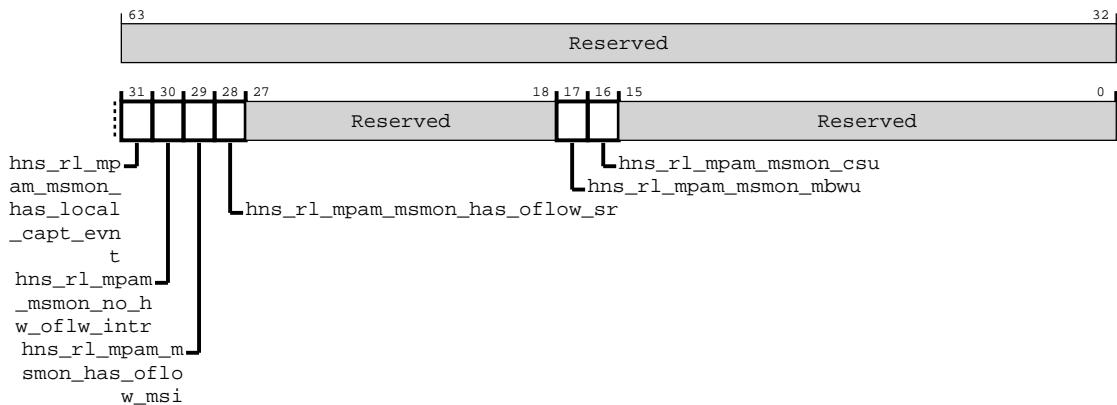
### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-645: cmn\_hns\_rl\_mpam\_msmon\_idr**



**Table 8-652: cmn\_hns\_rl\_mpam\_msmon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_mpam_msmon_has_local_capt_evnt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	0x1
[30]	hns_rl_mpam_msmon_no_hw_oflw_intr	0 HNF has hardwired MPAM overflow interrupt 1 HNF doesn't have hardwired MPAM overflow interrupt	RO	0b1
[29]	hns_rl_mpam_msmon_has_oflow_msi	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	0b0
[28]	hns_rl_mpam_msmon_has_oflow_sr	0 HNF doesn't have overflow status register 1 HNF has overflow status register	RO	0b0
[27:18]	Reserved	Reserved	RO	-
[17]	hns_rl_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	0x0
[16]	hns_rl_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

### 8.3.10.46 cmn\_hns\_rl\_mpam\_csumon\_idr

MPAM cache storage usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8088

##### Type

RO

##### Reset value

See individual bit resets

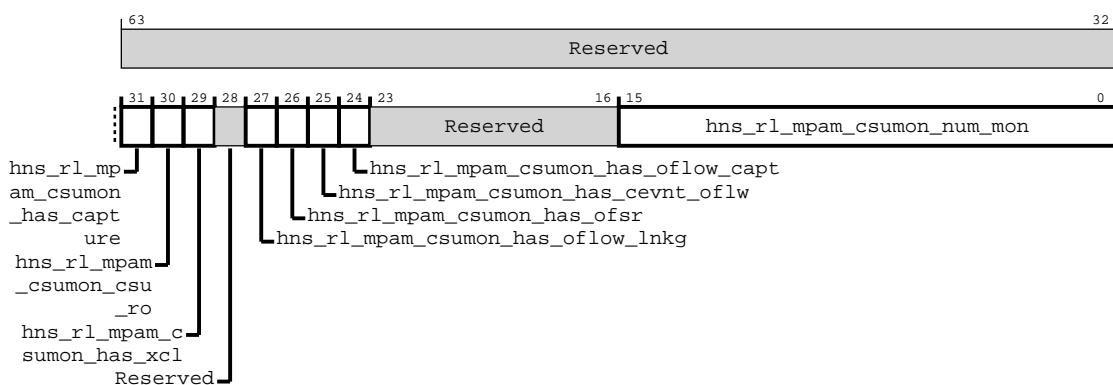
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-646: cmn\_hns\_rl\_mpam\_csumon\_idr**



**Table 8-653: cmn\_hns\_rl\_mpam\_csumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_rl_mpam_csumon_has_capture	<p><b>0</b> MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature.</p> <p><b>1</b> This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.</p>	RO	0x1
[30]	hns_rl_mpam_csumon_csu_ro	<p><b>0</b> MSMON_CSU is read/write.</p> <p><b>1</b> MSMON_CSU is read-only.</p>	RO	0b0
[29]	hns_rl_mpam_csumon_has_xcl	<p><b>0</b> MSMON_CFG_CSU_FLT does not implement the XCL field</p> <p><b>1</b> MSMON_CFG_CSU_FLT implements the XCL field</p>	RO	0b0
[28]	Reserved	Reserved	RO	-
[27]	hns_rl_mpam_csumon_has_oflow_lnk	<p><b>0</b> HNF doesn't support CSU overflow linkage</p> <p><b>1</b> HNF supports CSU overflow linkage</p>	RO	0b0
[26]	hns_rl_mpam_csumon_has_ofsr	<p><b>0</b> MSMON_CSU_OFSR register is not implemented</p> <p><b>1</b> MSMON_CSU_OFSR register is implemented</p>	RO	0b0
[25]	hns_rl_mpam_csumon_has_cevnt_oflw	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW</p>	RO	0b0
[24]	hns_rl_mpam_csumon_has_oflow_capt	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT</p>	RO	0b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	0x1

### 8.3.10.47 cmn\_hns\_rl\_mpam\_mbwumon\_idr

MPAM memory bandwidth usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8090

### Type

RO

### Reset value

See individual bit resets

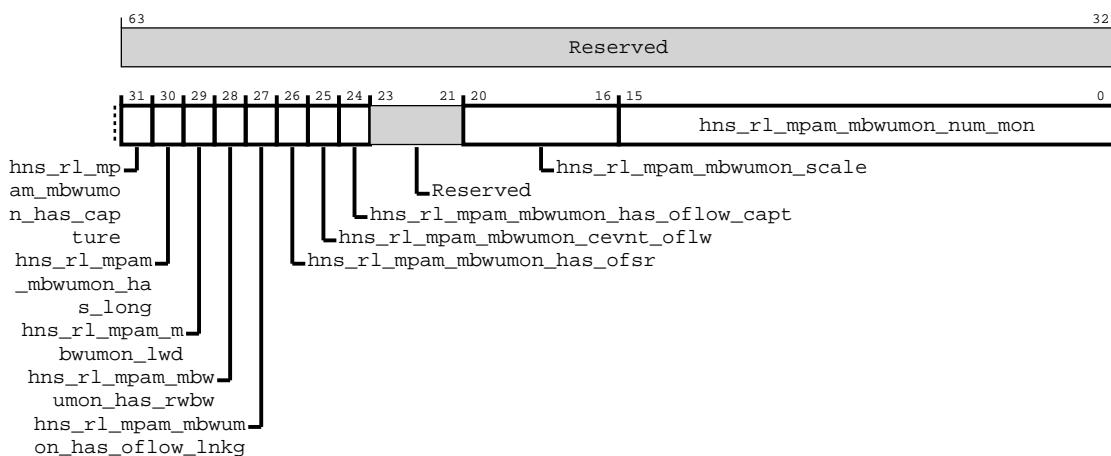
### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-647: cmn\_hns\_rl\_mpam\_mbwumon\_idr**



**Table 8-654: cmn\_hns\_rl\_mpam\_mbwumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rl_mpam_mbwumon_has_capture	<p><b>0</b> MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature.</p> <p><b>1</b> This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.</p>	RO	0x0

Bits	Name	Description	Type	Reset
[30]	hns_rl_mpam_mbwumon_has_long	<p><b>0</b> MSMON_MBWU_L is not implemented.</p> <p><b>1</b> MSMON_MBWU_L is implemented.</p>	RO	0b0
[29]	hns_rl_mpam_mbwumon_lwd	<p><b>0</b> MSMON_MBWU_L has 44-bit VALUE field in bits [43:0].</p> <p><b>1</b> MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].</p>	RO	0b0
[28]	hns_rl_mpam_mbwumon_has_rwbw	<p><b>0</b> Read/write bandwidth selection is not implemented.</p> <p><b>1</b> Read/write bandwidth selection is implemented.</p>	RO	0b0
[27]	hns_rl_mpam_mbwumon_has_oflow_lnkg	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p>	RO	0b0
[26]	hns_rl_mpam_mbwumon_has_ofsr	<p><b>0</b> MSMON_MBWU_OFSR register is not implemented</p> <p><b>1</b> MSMON_MBWU_OFSR register is implemented</p>	RO	0b0
[25]	hns_rl_mpam_mbwumon_cevnt_oflw	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p>	RO	0b0
[24]	hns_rl_mpam_mbwumon_has_oflow_capt	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p>	RO	0b0
[23:21]	Reserved	Reserved	RO	
[20:16]	hns_rl_mpam_mbwumon_scale	Scalling of MSMON_MBWU.VALUE in bits.	RO	0x0
[15:0]	hns_rl_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	0x0

### 8.3.10.48 cmn\_hns\_rl\_mpam\_ecr

MPAM Error Control Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x80F0

### Type

RW

### Reset value

See individual bit resets

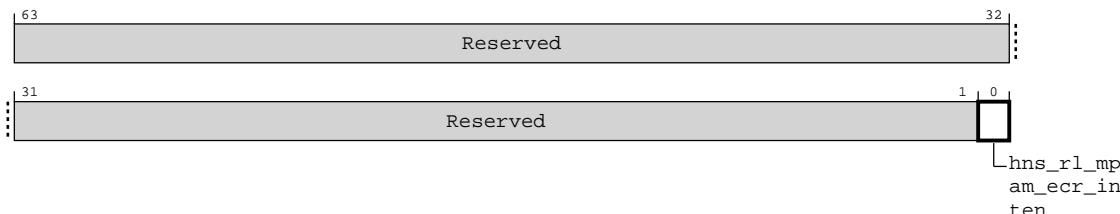
### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-648: cmn\_hns\_rl\_mpam\_ecr**



**Table 8-655: 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	0x0

## 8.3.10.49 cmn\_hns\_rl\_mpam\_esr

MPAM Error Status Register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

## Address offset

0x80F8

## Type

RW

## Reset value

See individual bit resets

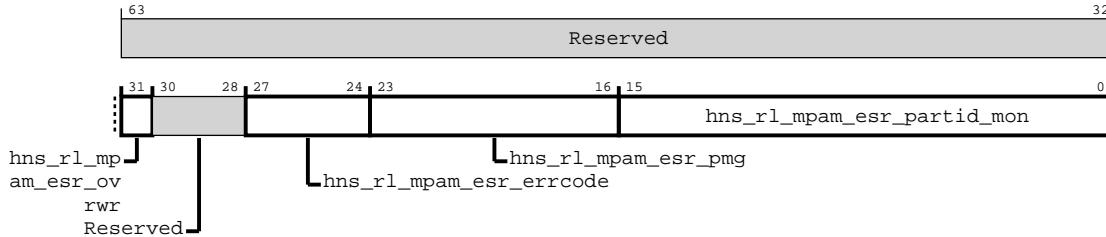
## Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-649: cmn\_hns\_rl\_mpam\_esr**



**Table 8-656: cmn\_hns\_rl\_mpam\_esr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rl_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	0x0
[30:28]	Reserved	Reserved	RO	
[27:24]	hns_rl_mpam_esr_errcode	Error code	RW	0x0
[23:16]	hns_rl_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	0x0
[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	0x0

### 8.3.10.50 cmn\_hns\_rl\_mpamcfg\_part\_sel

MPAM partition configuration selection register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8100

##### Type

RW

##### Reset value

See individual bit resets

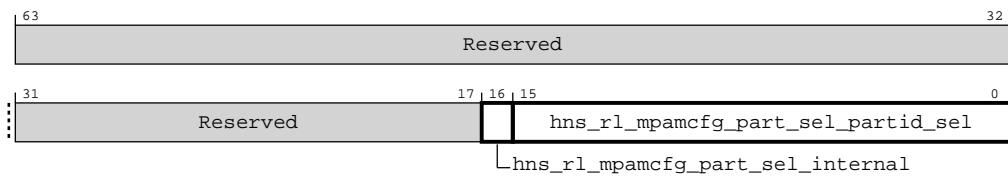
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-650: cmn\_hns\_rl\_mpamcfg\_part\_sel**



**Table 8-657: cmn\_hns\_rl\_mpamcfg\_part\_sel attributes**

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	
[16]	hns_rl_mpamcfg_part_sel_internal	If MPAMF_IDR.HAS_PARTID_NRW = 0, this field is <b>RAZ/WI</b> . If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interpret PARTID_SEL.	RW	0x0
[15:0]	hns_rl_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	0x0

### 8.3.10.51 cmn\_hns\_rl\_mpamcfg\_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8108

##### Type

RW

##### Reset value

See individual bit resets

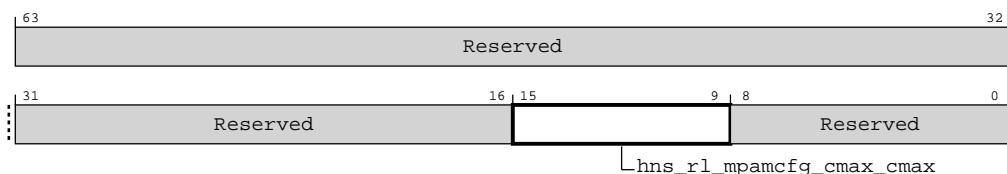
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-651: cmn\_hns\_rl\_mpamcfg\_cmax**



**Table 8-658: cmn\_hns\_rl\_mpamcfg\_cmax attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[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	0b1111111
[8:0]	Reserved	Reserved	RO	-

### 8.3.10.52 cmn\_hns\_rl\_mpamcfg\_mbw\_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8200

##### Type

RW

##### Reset value

See individual bit resets

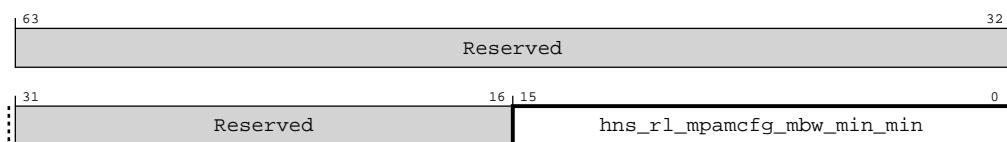
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-652: cmn\_hns\_rl\_mpamcfg\_mbw\_min**



**Table 8-659: 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	0x0

### 8.3.10.53 cmn\_hns\_rl\_mpamcfg\_mbw\_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8208

##### Type

RW

##### Reset value

See individual bit resets

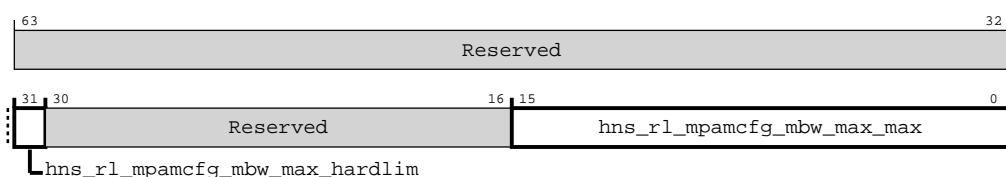
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-653: cmn\_hns\_rl\_mpamcfg\_mbw\_max**



**Table 8-660: cmn\_hns\_rl\_mpamcfg\_mbw\_max attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rl_mpamcfg_mbw_max_hardlim	<p><b>0</b> When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth.</p> <p><b>1</b> When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_rl_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	0x0

### 8.3.10.54 cmn\_hns\_rl\_mpamcfg\_mbw\_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8220

##### Type

RW

##### Reset value

See individual bit resets

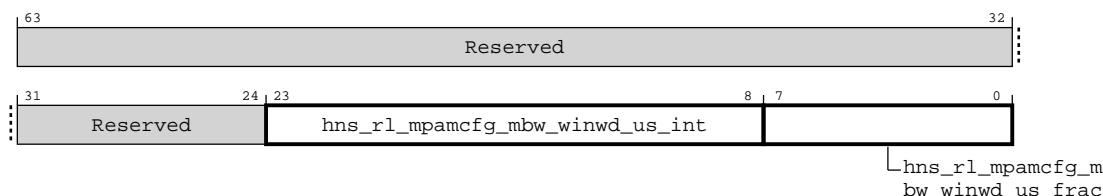
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-654: cmn\_hns\_rl\_mpamcfg\_mbw\_winwd**



**Table 8-661: 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	0x0

Bits	Name	Description	Type	Reset
[7:0]	hns_rl_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	0x0

### 8.3.10.55 cmn\_hns\_rl\_mpamcfg\_pri

MPAM priority partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8400

##### Type

RW

##### Reset value

See individual bit resets

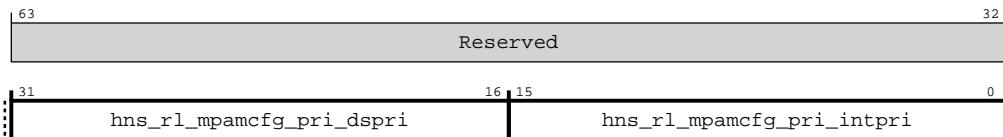
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-655: cmn\_hns\_rl\_mpamcfg\_pri**



**Table 8-662: 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	0x0
[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	0x0

### 8.3.10.56 cmn\_hns\_rl\_mpamcfg\_mbw\_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8500

##### Type

RW

##### Reset value

See individual bit resets

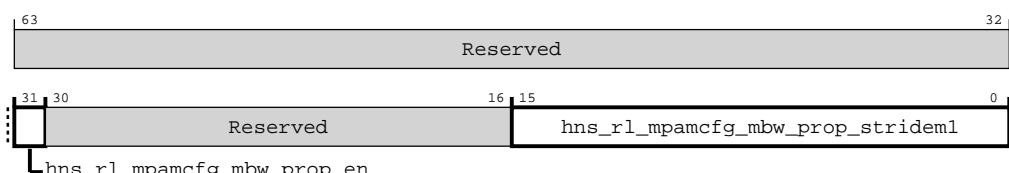
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-656: cmn\_hns\_rl\_mpamcfg\_mbw\_prop**



**Table 8-663: cmn\_hns\_rl\_mpamcfg\_mbw\_prop attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rl_mpamcfg_mbw_prop_en	<p><b>0</b> The selected partition is not regulated by proportional stride bandwidth partitioning.</p> <p><b>1</b> The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_rl_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	0x0

### 8.3.10.57 cmn\_hns\_rl\_mpamcfg\_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8600

##### Type

RW

##### Reset value

See individual bit resets

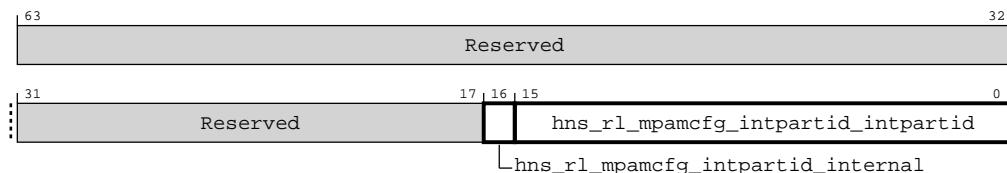
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-657: cmn\_hns\_rl\_mpamcfg\_intpartid**



**Table 8-664: cmn\_hns\_rl\_mpamcfg\_intpartid attributes**

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	
[16]	hns_rl_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	0x0

Bits	Name	Description	Type	Reset
[15:0]	hns_rl_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	0x0

### 8.3.10.58 cmn\_hns\_rl\_msmon\_cfg\_mon\_sel

Memory system performance monitor selection register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8800

##### Type

RW

##### Reset value

See individual bit resets

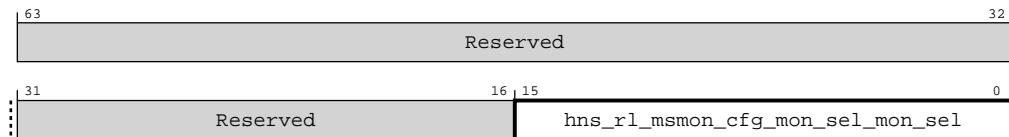
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-658: cmn\_hns\_rl\_msmon\_cfg\_mon\_sel**



**Table 8-665: 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	0x0

### 8.3.10.59 cmn\_hns\_rl\_msmon\_capt\_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8808

##### Type

RW

##### Reset value

See individual bit resets

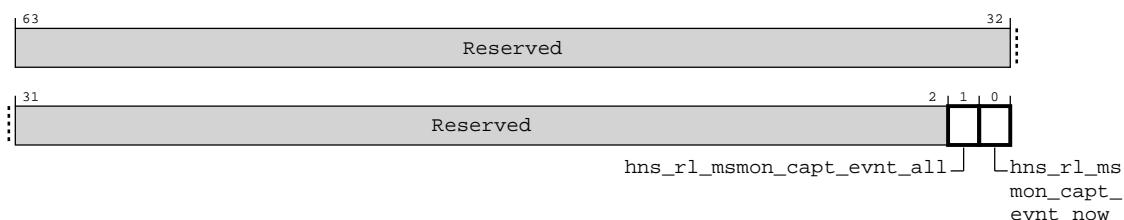
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-659: cmn\_hns\_rl\_msmon\_capt\_evnt**



**Table 8-666: 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	0x0
[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	0x0

### 8.3.10.60 cmn\_hns\_rl\_msmon\_cfg\_csu\_flt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8810

##### Type

RW

##### Reset value

See individual bit resets

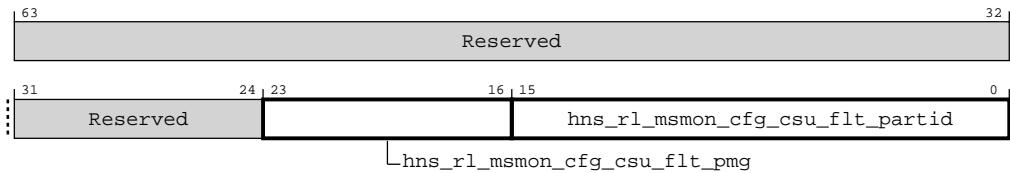
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-660: cmn\_hns\_rl\_msmon\_cfg\_csu\_flt**



**Table 8-667: cmn\_hns\_rl\_msmon\_cfg\_csu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_rl_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	0x0
[15:0]	hns_rl_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	0x0

### 8.3.10.61 cmn\_hns\_rl\_msmon\_cfg\_csu\_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8818

##### Type

RW

##### Reset value

See individual bit resets

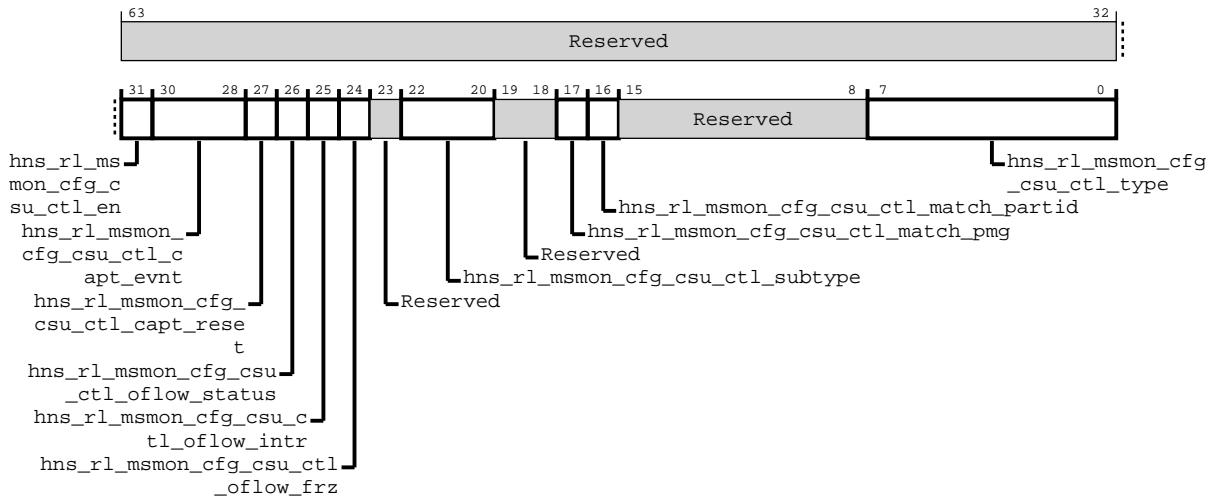
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-661: cmn\_hns\_rl\_msmon\_cfg\_csu\_ctl**



**Table 8-668: cmn\_hns\_rl\_msmon\_cfg\_csu\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rl_msmon_cfg_csu_ctl_en	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_rl_msmon_cfg_csu_ctl_capt_evnt	Select the event that triggers capture from the following:	RW	0x0
		<p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>		
[27]	hns_rl_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	0x0
[26]	hns_rl_msmon_cfg_csu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0
[25]	hns_rl_msmon_cfg_csu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_rl_msmon_cfg_csu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[23]	Reserved	Reserved	RO	
[22:20]	hns_rl_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_rl_msmon_cfg_csu_ctl_match_pmg	<p><b>0</b> Monitor storage used by all PMG values.</p> <p><b>1</b> Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0
[16]	hns_rl_msmon_cfg_csu_ctl_match_partid	<p><b>0</b> Monitor storage used by all PARTIDs.</p> <p><b>1</b> Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_rl_msmon_cfg_csu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	0x43

### 8.3.10.62 cmn\_hns\_rl\_msmon\_cfg\_mbwu\_fl

Memory system performance monitor configure memory bandwidth usage monitor filter register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8820

##### Type

RW

##### Reset value

See individual bit resets

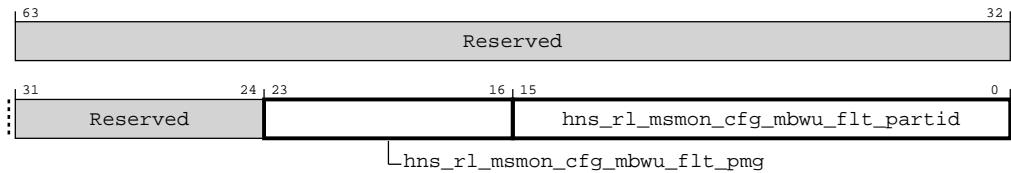
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-662: cmn\_hns\_rl\_msmon\_cfg\_mbwu\_flt**



**Table 8-669: cmn\_hns\_rl\_msmon\_cfg\_mbwu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_rl_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0
[15:0]	hns_rl_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0

### 8.3.10.63 cmn\_hns\_rl\_msmon\_cfg\_mbwu\_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8828

##### Type

RW

##### Reset value

See individual bit resets

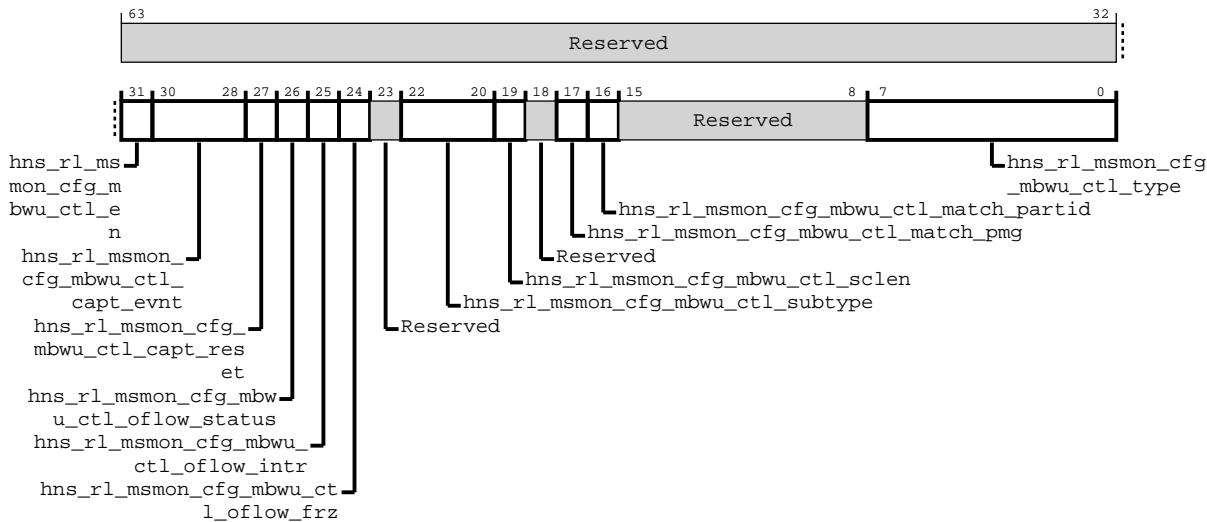
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-663: cmn\_hns\_rl\_msmon\_cfg\_mbwu\_ctl**



**Table 8-670: 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	0x0
[30:28]	hns_rl_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	0x0
[27]	hns_rl_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	0x0
[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	0x0

Bits	Name	Description	Type	Reset
[25]	hns_rl_msmon_cfg_mbwu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_rl_msmon_cfg_mbwu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[23]	Reserved	Reserved	RO	
[22:20]	hns_rl_msmon_cfg_mbwu_ctl_subtype	<p>A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports:</p> <p><b>0</b> Do not count any bandwidth.</p> <p><b>1</b> Count bandwidth used by memory reads</p> <p><b>2</b> Count bandwidth used by memory writes</p> <p><b>3</b> 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 <b>UNPREDICTABLE</b>.</p>	RW	0x0
[19]	hns_rl_msmon_cfg_mbwu_ctl_sclen	<p><b>0</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance.</p> <p><b>1</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.</p>	RW	0x0
[18]	Reserved	Reserved	RO	
[17]	hns_rl_msmon_cfg_mbwu_ctl_match_pmg	<p><b>0</b> Monitor bandwidth used by all PMG values.</p> <p><b>1</b> Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0
[16]	hns_rl_msmon_cfg_mbwu_ctl_match_partid	<p><b>0</b> Monitor bandwidth used by all PARTIDs.</p> <p><b>1</b> Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_rl_msmon_cfg_mbwu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	0x42

### 8.3.10.64 cmn\_hns\_rl\_msmon\_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8840

##### Type

RW

##### Reset value

See individual bit resets

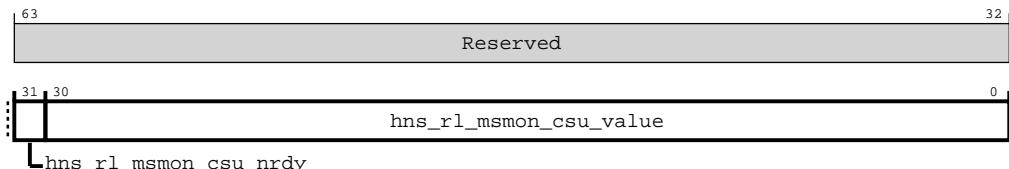
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-664: cmn\_hns\_rl\_msmon\_csu**



**Table 8-671: 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	0x0
[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	0x0

### 8.3.10.65 cmn\_hns\_rl\_msmon\_csu\_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8848

##### Type

RW

##### Reset value

See individual bit resets

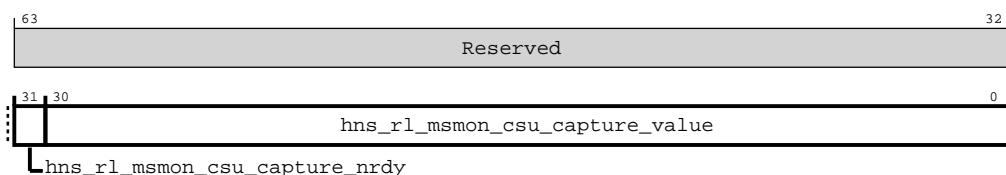
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-665: cmn\_hns\_rl\_msmon\_csu\_capture**



**Table 8-672: 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	0x0
[30:0]	hns_rl_msmon_csu_capture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	0x0

### 8.3.10.66 cmn\_hns\_rl\_msmon\_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8860

##### Type

RW

##### Reset value

See individual bit resets

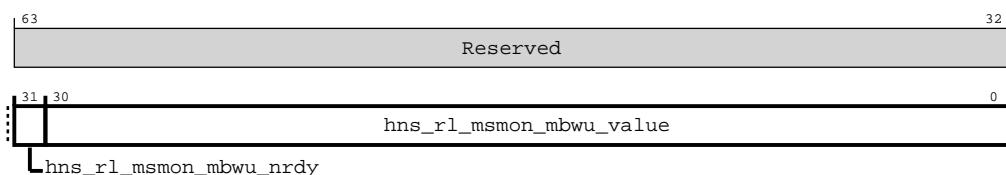
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-666: cmn\_hns\_rl\_msmon\_mbwu**



**Table 8-673: 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	0x0
[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	0x0

### 8.3.10.67 cmn\_hns\_rl\_msmon\_mbwu\_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8868

##### Type

RW

##### Reset value

See individual bit resets

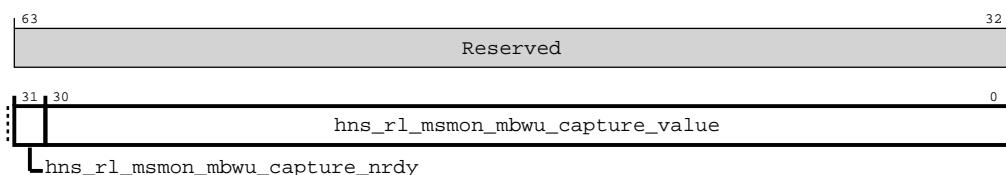
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-667: cmn\_hns\_rl\_msmon\_mbwu\_capture**



**Table 8-674: 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	0x0
[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	0x0

### 8.3.10.68 cmn\_hns\_rl\_mpamcfg\_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x9000

##### Type

RW

##### Reset value

See individual bit resets

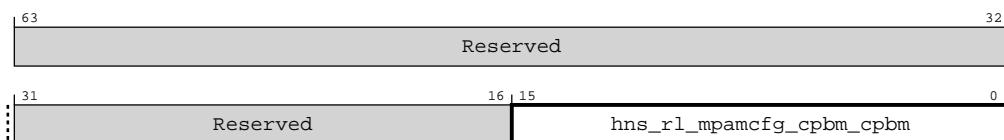
##### Usage constraints

This register is owned in the Realm space and is accessible by using Realm and Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-668: cmn\_hns\_rl\_mpamcfg\_cpbm**



**Table 8-675: 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.  <b>NOTE</b> CPBM can not be all zeros for any PARTID.	RW	0xFFFF

### 8.3.11 HN-S MPAM S register summary

The following table describes the registers for the relevant component.

**Table 8-676: cmn\_hns\_mpam\_s\_cfg register summary**

Offset	Name	Type	Description
0x0	cmn_hns_mpam_s_node_info	RO	<a href="#">cmn_hns_mpam_s_node_info</a>
0x80	cmn_hns_mpam_s_child_info	RO	<a href="#">cmn_hns_mpam_s_child_info</a>
0x1000	cmn_hns_s_mpam_idr	RO	<a href="#">cmn_hns_s_mpam_idr</a>
0x1008	cmn_hns_mpam_sidr	RO	<a href="#">cmn_hns_mpam_sidr</a>
0x1018	cmn_hns_s_mpam_iidr	RO	<a href="#">cmn_hns_s_mpam_iidr</a>
0x1020	cmn_hns_s_mpam_aidr	RO	<a href="#">cmn_hns_s_mpam_aidr</a>
0x1028	cmn_hns_s_mpam_impl_idr	RO	<a href="#">cmn_hns_s_mpam_impl_idr</a>
0x1030	cmn_hns_s_mpam_cpor_idr	RO	<a href="#">cmn_hns_s_mpam_cpor_idr</a>
0x1038	cmn_hns_s_mpam_ccap_idr	RO	<a href="#">cmn_hns_s_mpam_ccap_idr</a>
0x1040	cmn_hns_s_mpam_mbw_idr	RO	<a href="#">cmn_hns_s_mpam_mbw_idr</a>
0x1048	cmn_hns_s_mpam_pri_idr	RO	<a href="#">cmn_hns_s_mpam_pri_idr</a>
0x1050	cmn_hns_s_mpam_partid_nrw_idr	RO	<a href="#">cmn_hns_s_mpam_partid_nrw_idr</a>
0x1080	cmn_hns_s_mpam_msmon_idr	RO	<a href="#">cmn_hns_s_mpam_msmon_idr</a>
0x1088	cmn_hns_s_mpam_csumon_idr	RO	<a href="#">cmn_hns_s_mpam_csumon_idr</a>
0x1090	cmn_hns_s_mpam_mbwumon_idr	RO	<a href="#">cmn_hns_s_mpam_mbwumon_idr</a>
0x10F0	cmn_hns_s_mpam_ecr	RW	<a href="#">cmn_hns_s_mpam_ecr</a>
0x10F8	cmn_hns_s_mpam_esr	RW	<a href="#">cmn_hns_s_mpam_esr</a>
0x1100	cmn_hns_s_mpamcfg_part_sel	RW	<a href="#">cmn_hns_s_mpamcfg_part_sel</a>
0x1108	cmn_hns_s_mpamcfg_cmax	RW	<a href="#">cmn_hns_s_mpamcfg_cmax</a>
0x1200	cmn_hns_s_mpamcfg_mbw_min	RW	<a href="#">cmn_hns_s_mpamcfg_mbw_min</a>
0x1208	cmn_hns_s_mpamcfg_mbw_max	RW	<a href="#">cmn_hns_s_mpamcfg_mbw_max</a>
0x1220	cmn_hns_s_mpamcfg_mbw_winwd	RW	<a href="#">cmn_hns_s_mpamcfg_mbw_winwd</a>
0x1400	cmn_hns_s_mpamcfg_pri	RW	<a href="#">cmn_hns_s_mpamcfg_pri</a>
0x1500	cmn_hns_s_mpamcfg_mbw_prop	RW	<a href="#">cmn_hns_s_mpamcfg_mbw_prop</a>
0x1600	cmn_hns_s_mpamcfg_intpartid	RW	<a href="#">cmn_hns_s_mpamcfg_intpartid</a>
0x1800	cmn_hns_s_msmon_cfg_mon_sel	RW	<a href="#">cmn_hns_s_msmon_cfg_mon_sel</a>
0x1808	cmn_hns_s_msmon_capt_evnt	RW	<a href="#">cmn_hns_s_msmon_capt_evnt</a>
0x1810	cmn_hns_s_msmon_cfg_csu_flt	RW	<a href="#">cmn_hns_s_msmon_cfg_csu_flt</a>
0x1818	cmn_hns_s_msmon_cfg_csu_ctl	RW	<a href="#">cmn_hns_s_msmon_cfg_csu_ctl</a>
0x1820	cmn_hns_s_msmon_cfg_mbwu_flt	RW	<a href="#">cmn_hns_s_msmon_cfg_mbwu_flt</a>
0x1828	cmn_hns_s_msmon_cfg_mbwu_ctl	RW	<a href="#">cmn_hns_s_msmon_cfg_mbwu_ctl</a>
0x1840	cmn_hns_s_msmon_csu	RW	<a href="#">cmn_hns_s_msmon_csu</a>
0x1848	cmn_hns_s_msmon_csu_capture	RW	<a href="#">cmn_hns_s_msmon_csu_capture</a>
0x1860	cmn_hns_s_msmon_mbwu	RW	<a href="#">cmn_hns_s_msmon_mbwu</a>
0x1868	cmn_hns_s_msmon_mbwu_capture	RW	<a href="#">cmn_hns_s_msmon_mbwu_capture</a>
0x2000	cmn_hns_s_mpamcfg_cpbm	RW	<a href="#">cmn_hns_s_mpamcfg_cpbm</a>

Offset	Name	Type	Description
0x8000	cmn_hns_rt_mpam_idr	RO	<a href="#">cmn_hns_rt_mpam_idr</a>
0x8018	cmn_hns_rt_mpam_iidr	RO	<a href="#">cmn_hns_rt_mpam_iidr</a>
0x8020	cmn_hns_rt_mpam_aidr	RO	<a href="#">cmn_hns_rt_mpam_aidr</a>
0x8028	cmn_hns_rt_mpam_impl_idr	RO	<a href="#">cmn_hns_rt_mpam_impl_idr</a>
0x8030	cmn_hns_rt_mpam_cpor_idr	RO	<a href="#">cmn_hns_rt_mpam_cpor_idr</a>
0x8038	cmn_hns_rt_mpam_ccap_idr	RO	<a href="#">cmn_hns_rt_mpam_ccap_idr</a>
0x8040	cmn_hns_rt_mpam_mbw_idr	RO	<a href="#">cmn_hns_rt_mpam_mbw_idr</a>
0x8048	cmn_hns_rt_mpam_pri_idr	RO	<a href="#">cmn_hns_rt_mpam_pri_idr</a>
0x8050	cmn_hns_rt_mpam_partid_nrw_idr	RO	<a href="#">cmn_hns_rt_mpam_partid_nrw_idr</a>
0x8080	cmn_hns_rt_mpam_msmon_idr	RO	<a href="#">cmn_hns_rt_mpam_msmon_idr</a>
0x8088	cmn_hns_rt_mpam_csumon_idr	RO	<a href="#">cmn_hns_rt_mpam_csumon_idr</a>
0x8090	cmn_hns_rt_mpam_mbwumon_idr	RO	<a href="#">cmn_hns_rt_mpam_mbwumon_idr</a>
0x80F0	cmn_hns_rt_mpam_ecr	RW	<a href="#">cmn_hns_rt_mpam_ecr</a>
0x80F8	cmn_hns_rt_mpam_esr	RW	<a href="#">cmn_hns_rt_mpam_esr</a>
0x8100	cmn_hns_rt_mpamcfg_part_sel	RW	<a href="#">cmn_hns_rt_mpamcfg_part_sel</a>
0x8108	cmn_hns_rt_mpamcfg_cmax	RW	<a href="#">cmn_hns_rt_mpamcfg_cmax</a>
0x8200	cmn_hns_rt_mpamcfg_mbw_min	RW	<a href="#">cmn_hns_rt_mpamcfg_mbw_min</a>
0x8208	cmn_hns_rt_mpamcfg_mbw_max	RW	<a href="#">cmn_hns_rt_mpamcfg_mbw_max</a>
0x8220	cmn_hns_rt_mpamcfg_mbw_winwd	RW	<a href="#">cmn_hns_rt_mpamcfg_mbw_winwd</a>
0x8400	cmn_hns_rt_mpamcfg_pri	RW	<a href="#">cmn_hns_rt_mpamcfg_pri</a>
0x8500	cmn_hns_rt_mpamcfg_mbw_prop	RW	<a href="#">cmn_hns_rt_mpamcfg_mbw_prop</a>
0x8600	cmn_hns_rt_mpamcfg_intpartid	RW	<a href="#">cmn_hns_rt_mpamcfg_intpartid</a>
0x8800	cmn_hns_rt_msmon_cfg_mon_sel	RW	<a href="#">cmn_hns_rt_msmon_cfg_mon_sel</a>
0x8808	cmn_hns_rt_msmon_capt_evnt	RW	<a href="#">cmn_hns_rt_msmon_capt_evnt</a>
0x8810	cmn_hns_rt_msmon_cfg_csu_flt	RW	<a href="#">cmn_hns_rt_msmon_cfg_csu_flt</a>
0x8818	cmn_hns_rt_msmon_cfg_csu_ctl	RW	<a href="#">cmn_hns_rt_msmon_cfg_csu_ctl</a>
0x8820	cmn_hns_rt_msmon_cfg_mbwu_flt	RW	<a href="#">cmn_hns_rt_msmon_cfg_mbwu_flt</a>
0x8828	cmn_hns_rt_msmon_cfg_mbwu_ctl	RW	<a href="#">cmn_hns_rt_msmon_cfg_mbwu_ctl</a>
0x8840	cmn_hns_rt_msmon_csu	RW	<a href="#">cmn_hns_rt_msmon_csu</a>
0x8848	cmn_hns_rt_msmon_csu_capture	RW	<a href="#">cmn_hns_rt_msmon_csu_capture</a>
0x8860	cmn_hns_rt_msmon_mbwu	RW	<a href="#">cmn_hns_rt_msmon_mbwu</a>
0x8868	cmn_hns_rt_msmon_mbwu_capture	RW	<a href="#">cmn_hns_rt_msmon_mbwu_capture</a>
0x9000	cmn_hns_rt_mpamcfg_cpbm	RW	<a href="#">cmn_hns_rt_mpamcfg_cpbm</a>

### 8.3.11.1 cmn\_hns\_mpam\_s\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x0

### Type

RO

### Reset value

See individual bit resets

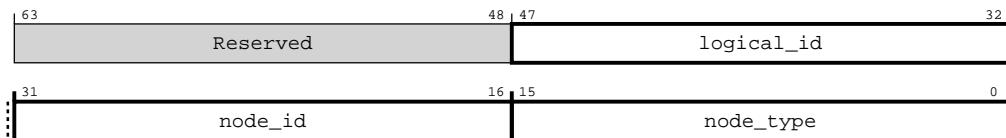
### Usage constraints

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 8-669: cmn\_hns\_mpam\_s\_node\_info**



**Table 8-677: cmn\_hns\_mpam\_s\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0

## 8.3.11.2 cmn\_hns\_mpam\_s\_child\_info

Provides component child identification information.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x80

### Type

RO

### Reset value

See individual bit resets

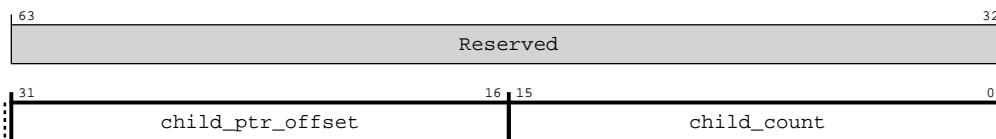
### Usage constraints

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 8-670: cmn\_hns\_mpam\_s\_child\_info**



**Table 8-678: 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	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.11.3 cmn\_hns\_s\_mpam\_idr

MPAM features ID register. This is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x1000

#### Type

RO

## Reset value

See individual bit resets

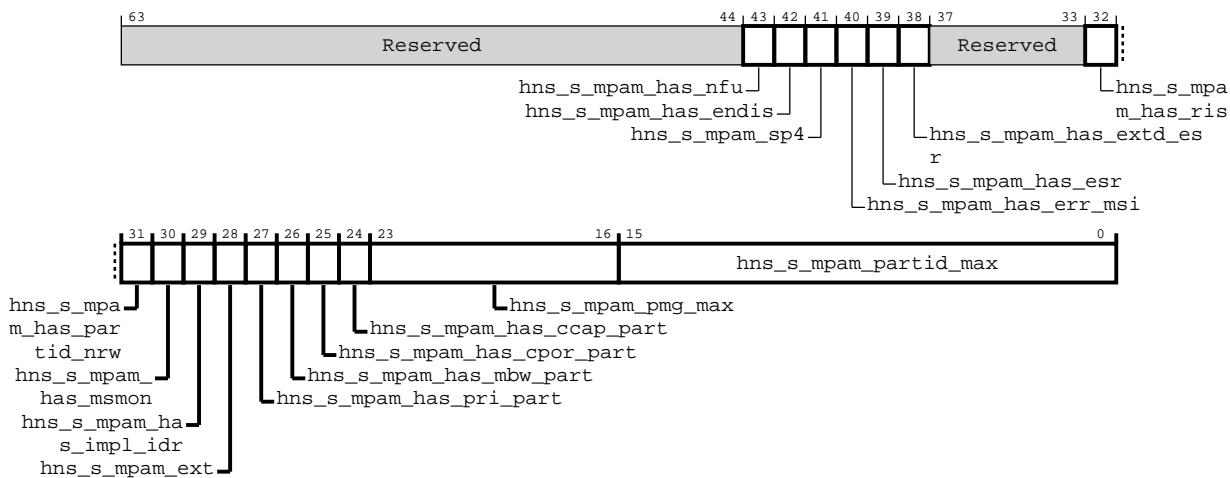
## Usage constraints

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 8-671: cmn\_hns\_s\_mpam\_idr**



**Table 8-679: cmn\_hns\_s\_mpam\_idr attributes**

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_s_mpam_has_nfu	<p><b>0</b> HN-F does not support no future use field</p> <p><b>1</b> HN-F supports no future use field</p>	RO	0b0
[42]	hns_s_mpam_has_endis	<p><b>0</b> HN-F does not support PARTID enable and disable functionality</p> <p><b>1</b> HN-F supports PARTID enable and disable functionality</p>	RO	0b0
[41]	hns_s_mpam_sp4	<p><b>0</b> HN-F supports two PARTID spaces</p> <p><b>1</b> HN-F supports four PARTID spaces</p>	RO	0b1

Bits	Name	Description	Type	Reset
[40]	hns_s_mpam_has_err_msi	<p><b>0</b> HN-F does not support MSI writes to signal MPAM error interrupt</p> <p><b>1</b> HN-F supports MSI writes to signal MPAM error interrupt</p>	RO	0b0
[39]	hns_s_mpam_has_esr	<p><b>0</b> HN-F does not support MPAM error handling</p> <p><b>1</b> HN-F supports MPAM error handling</p>	RO	0b1
[38]	hns_s_mpam_has_extd_esr	<p><b>0</b> MPAMF_ESR is 32 bits</p> <p><b>1</b> MPAMF_ESR is 64 bits</p>	RO	0b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_s_mpam_has_ris	<p><b>0</b> HN-F does not support MPAM resource instance selector</p> <p><b>1</b> HN-F supports MPAM resource instance selector</p>	RO	0b0
[31]	hns_s_mpam_has_partid_nrw	<p><b>0</b> HN-F does not support MPAM PARTID Narrowing</p> <p><b>1</b> HN-F supports MPAM PARTID Narrowing</p>	RO	0x0
[30]	hns_s_mpam_has_msmon	<p><b>0</b> MPAM performance monitoring is not supported</p> <p><b>1</b> MPAM performance monitoring is supported</p>	RO	0x0
[29]	hns_s_mpam_has_impl_idr	<p><b>0</b> MPAM implementation specific partitioning features not supported</p> <p><b>1</b> MPAM implementation specific partitioning features supported</p>	RO	Configuration dependent
[28]	hns_s_mpam_ext	<p><b>0</b> HN-F has no defined bits in [63:32]</p> <p><b>1</b> HN-F has bits defined in [63:32]</p>	RO	0b1
[27]	hns_s_mpam_has_pri_part	<p><b>0</b> MPAM priority partitioning is not supported</p> <p><b>1</b> MPAM priority partitioning is supported</p>	RO	0x0

Bits	Name	Description	Type	Reset
[26]	hns_s_mpam_has_mbw_part	<p><b>0</b> MPAM memory bandwidth partitioning is not supported</p> <p><b>1</b> MPAM memory bandwidth partitioning is supported</p>	RO	0x0
[25]	hns_s_mpam_has_cpor_part	<p><b>0</b> MPAM cache portion partitioning is not supported</p> <p><b>1</b> MPAM cache portion partitioning is supported</p>	RO	Configuration dependent
[24]	hns_s_mpam_has_ccap_part	<p><b>0</b> MPAM cache maximum capacity partitioning is not supported</p> <p><b>1</b> MPAM cache maximum capacity partitioning is supported</p>	RO	Configuration dependent
[23:16]	hns_s_mpam_pmg_max	Maximum value of non-secure PMG supported by this HN-F	RO	0x1
[15:0]	hns_s_mpam_partid_max	Maximum value of secure PARTID supported by this HN-F	RO	0xF

### 8.3.11.4 cmn\_hns\_mpam\_sldr

MPAM features Secure ID register. This is Secure (S) register only.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1008

##### Type

RO

##### Reset value

See individual bit resets

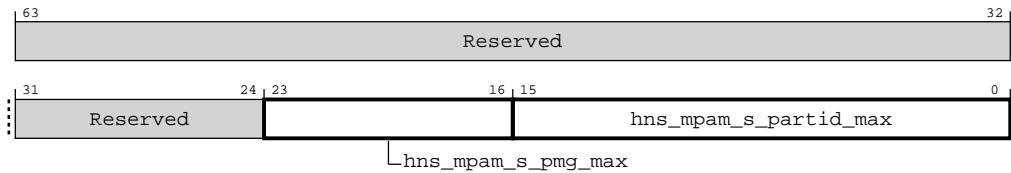
##### Usage constraints

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 8-672: cmn\_hns\_mpam\_sidr**



**Table 8-680: 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	0x1
[15:0]	hns_mpam_s_partid_max	Maximum value of secure PARTID supported by this HN-F	RO	0xF

### 8.3.11.5 cmn\_hns\_s\_mpam\_iidr

MPAM Implementation ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1018

##### Type

RO

##### Reset value

See individual bit resets

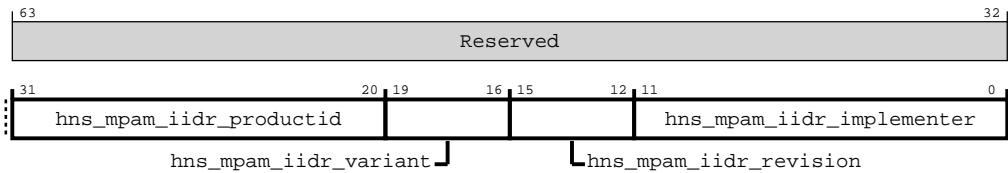
##### Usage constraints

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 8-673: cmn\_hns\_s\_mpam\_iidr**



**Table 8-681: 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	0x3e
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	Configuration dependent
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	0b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	0x43B

### 8.3.11.6 cmn\_hns\_s\_mpam\_aidr

MPAM architecture ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1020

##### Type

RO

##### Reset value

See individual bit resets

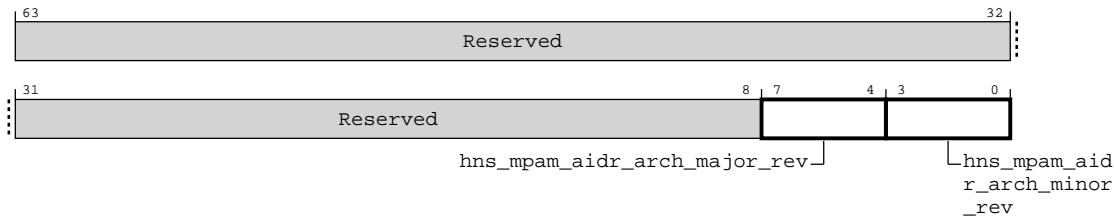
##### Usage constraints

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 8-674: cmn\_hns\_s\_mpam\_aidr**



**Table 8-682: 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	0b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	0b0001

### 8.3.11.7 cmn\_hns\_s\_mpam\_impl\_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1028

##### Type

RO

##### Reset value

See individual bit resets

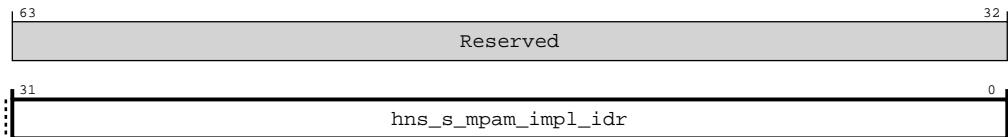
##### Usage constraints

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 8-675: cmn\_hns\_s\_mpam\_impl\_idr**



**Table 8-683: 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	0x00000000

### 8.3.11.8 cmn\_hns\_s\_mpam\_cpor\_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1030

##### Type

RO

##### Reset value

See individual bit resets

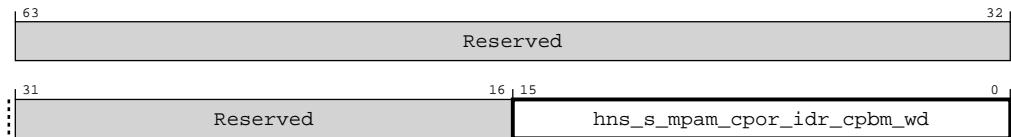
#### Usage constraints

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 8-676: cmn\_hns\_s\_mpam\_cpor\_idr**



**Table 8-684: 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	0x10

### 8.3.11.9 cmn\_hns\_s\_mpam\_ccap\_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1038

##### Type

RO

##### Reset value

See individual bit resets

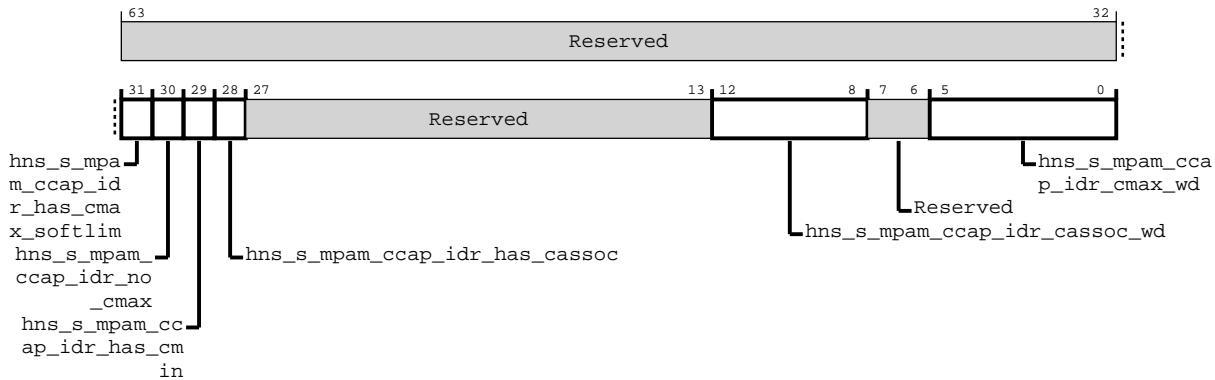
#### Usage constraints

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 8-677: cmn\_hns\_s\_mpam\_ccap\_idr**



**Table 8-685: 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>	<p><b>0</b> HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit</p> <p><b>1</b> HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit</p>	RO	0x0
[30]	<code>hns_s_mpam_ccap_idr_no_cmax</code>	<p><b>0</b> HN-F support MPAMCFG_CMAX</p> <p><b>1</b> HN-F doesn't support MPAMCFG_CMAX</p>	RO	0x0
[29]	<code>hns_s_mpam_ccap_idr_has_cmin</code>	<p><b>0</b> HN-F does not support MPAMCFG_CMIN</p> <p><b>1</b> HN-F supports MPAMCFG_CMIN</p>	RO	0x0
[28]	<code>hns_s_mpam_ccap_idr_has_cassoc</code>	<p><b>0</b> HN-F does not support MPAMCFG_CASSOC</p> <p><b>1</b> HN-F supports MPAMCFG_CASSOC</p>	RO	0x0
[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	0x0
[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	0x7

### 8.3.11.10 cmn\_hns\_s\_mpam\_mbw\_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1040

##### Type

RO

##### Reset value

See individual bit resets

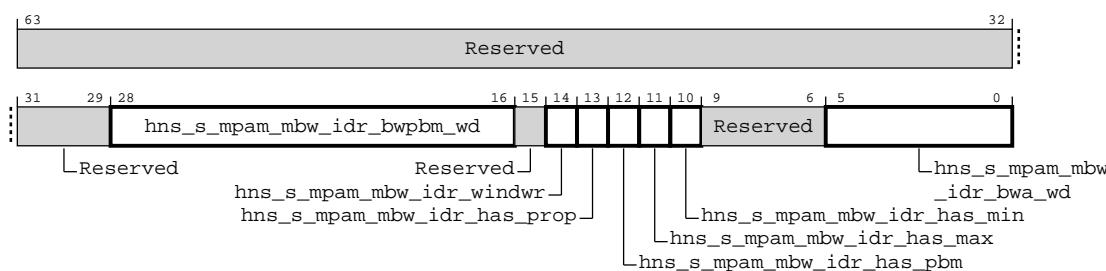
##### Usage constraints

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 8-678: cmn\_hns\_s\_mpam\_mbw\_idr**



**Table 8-686: 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	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14]	hns_s_mpam_mbw_idr_windwr	<p><b>0</b> The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed.</p> <p><b>1</b> The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.</p>	RO	0x0
[13]	hns_s_mpam_mbw_idr_has_prop	<p><b>0</b> There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register</p> <p><b>1</b> MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.</p>	RO	0x0
[12]	hns_s_mpam_mbw_idr_has_pbm	<p><b>0</b> There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register</p> <p><b>1</b> MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.</p>	RO	0x0
[11]	hns_s_mpam_mbw_idr_has_max	<p><b>0</b> There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register</p> <p><b>1</b> MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[10]	hns_s_mpam_mbw_idr_has_min	<p><b>0</b> There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register</p> <p><b>1</b> MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[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	0b0000

### 8.3.11.11 cmn\_hns\_s\_mpam\_pri\_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1048

### Type

RO

### Reset value

See individual bit resets

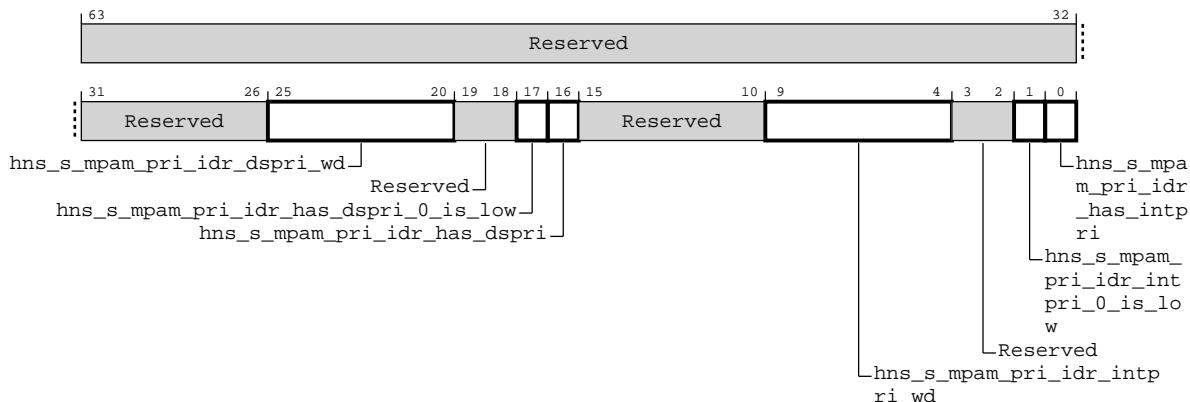
### Usage constraints

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 8-679: cmn\_hns\_s\_mpam\_pri\_idr**



**Table 8-687: 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	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_s_mpam_pri_idr_has_dspri_0_is_low	<p>0 In the DSPRI field, a value of 0 means highest priority.</p> <p>1 In the DSPRI field, a value of 0 means lowest priority.</p>	RO	0x0
[16:15]	Reserved			
[10:9]	Reserved			
[4:3]	Reserved			
[2]	hns_s_mpam_pri_idr_intpri_rw			
[1]	hns_s_mpam_pri_idr_intpri_wd			
[0]	hns_s_mpam_pri_idr_intpri_rd			

Bits	Name	Description	Type	Reset
[16]	hns_s_mpam_pri_idr_has_dspri	<p><b>0</b> This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports downstream priority and has an DSPRI field.</p>	RO	0x0
[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	0x0
[3:2]	Reserved	Reserved	RO	
[1]	hns_s_mpam_pri_idr_intpri_0_is_low	<p><b>0</b> In the INTPRI field, a value of 0 means highest priority.</p> <p><b>1</b> In the INTPRI field, a value of 0 means lowest priority.</p>	RO	0x0
[0]	hns_s_mpam_pri_idr_has_intpri	<p><b>0</b> This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports internal priority and has an INTPRI field.</p>	RO	0x0

### 8.3.11.12 cmn\_hns\_s\_mpam\_partid\_nrw\_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1050

##### Type

RO

##### Reset value

See individual bit resets

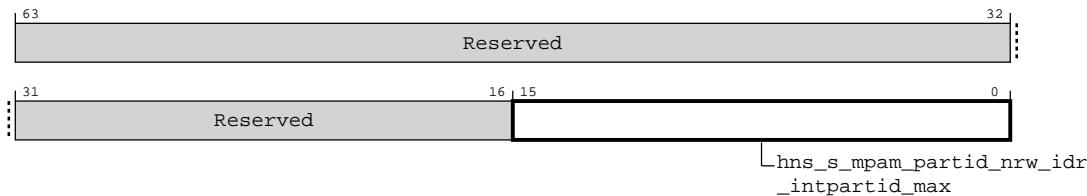
##### Usage constraints

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 8-680: cmn\_hns\_s\_mpam\_partid\_nrw\_idr**



**Table 8-688: 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	0x00

### 8.3.11.13 cmn\_hns\_s\_mpam\_msmon\_idr

MPAM performance monitoring ID register. This is a shared register for S and NS

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x1080

### Type

RO

### Reset value

See individual bit resets

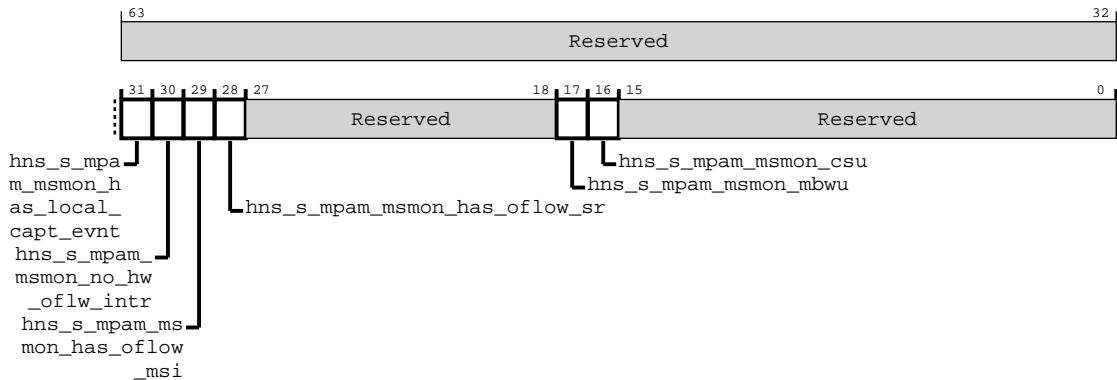
### Usage constraints

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 8-681: cmn\_hns\_s\_mpam\_msmon\_idr**



**Table 8-689: cmn\_hns\_s\_mpam\_msmon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_mpam_msmon_has_local_capt_evnt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	0x1
[30]	hns_s_mpam_msmon_no_hw_oflw_intr	<p><b>0</b> HNF has hardwired MPAM overflow interrupt</p> <p><b>1</b> HNF doesn't have hardwired MPAM overflow interrupt</p>	RO	0b1
[29]	hns_s_mpam_msmon_has_oflow_msi	<p><b>0</b> HNF doesn't have support for MSI writes to signal MPAM monitor overflow interrupt</p> <p><b>1</b> HNF has support for MSI writes to signal MPAM monitor overflow interrupt</p>	RO	0b0
[28]	hns_s_mpam_msmon_has_oflow_sr	<p><b>0</b> HNF doesn't have overflow status register</p> <p><b>1</b> HNF has overflow status register</p>	RO	0b0
[27:18]	Reserved	Reserved	RO	-
[17]	hns_s_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	0x0
[16]	hns_s_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

### 8.3.11.14 cmn\_hns\_s\_mpam\_csumon\_idr

MPAM cache storage usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1088

##### Type

RO

##### Reset value

See individual bit resets

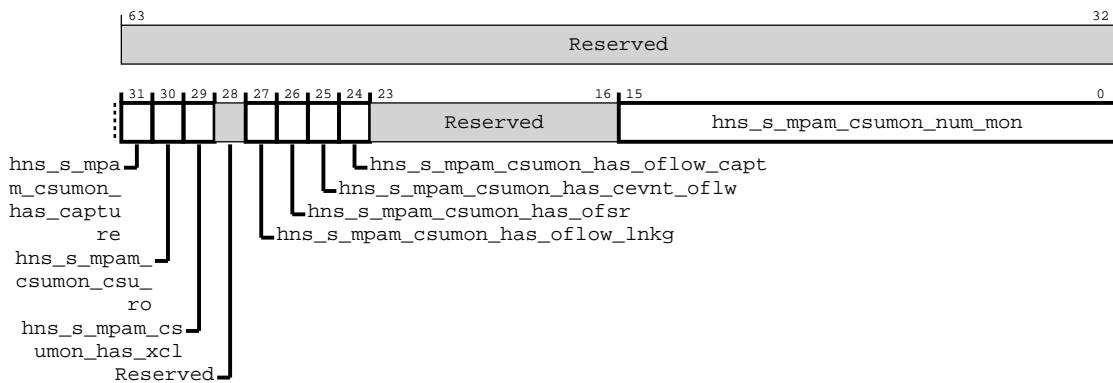
##### Usage constraints

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 8-682: cmn\_hns\_s\_mpam\_csumon\_idr**



**Table 8-690: cmn\_hns\_s\_mpam\_csumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_s_mpam_csumon_has_capture	<p><b>0</b> MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature.</p> <p><b>1</b> This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.</p>	RO	0x1
[30]	hns_s_mpam_csumon_csu_ro	<p><b>0</b> MSMON_CSU is read/write.</p> <p><b>1</b> MSMON_CSU is read-only.</p>	RO	0b0
[29]	hns_s_mpam_csumon_has_xcl	<p><b>0</b> MSMON_CFG_CSU_FLT does not implement the XCL field</p> <p><b>1</b> MSMON_CFG_CSU_FLT implements the XCL field</p>	RO	0b0
[28]	Reserved	Reserved	RO	-
[27]	hns_s_mpam_csumon_has_oflow_lnk	<p><b>0</b> HNF doesn't support CSU overflow linkage</p> <p><b>1</b> HNF supports CSU overflow linkage</p>	RO	0b0
[26]	hns_s_mpam_csumon_has_ofsr	<p><b>0</b> MSMON_CSU_OFSR register is not implemented</p> <p><b>1</b> MSMON_CSU_OFSR register is implemented</p>	RO	0b0
[25]	hns_s_mpam_csumon_has_cevnt_oflw	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW</p>	RO	0b0
[24]	hns_s_mpam_csumon_has_oflow_capt	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT</p>	RO	0b0
[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

### 8.3.11.15 cmn\_hns\_s\_mpam\_mbwumon\_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1090

##### Type

RO

##### Reset value

See individual bit resets

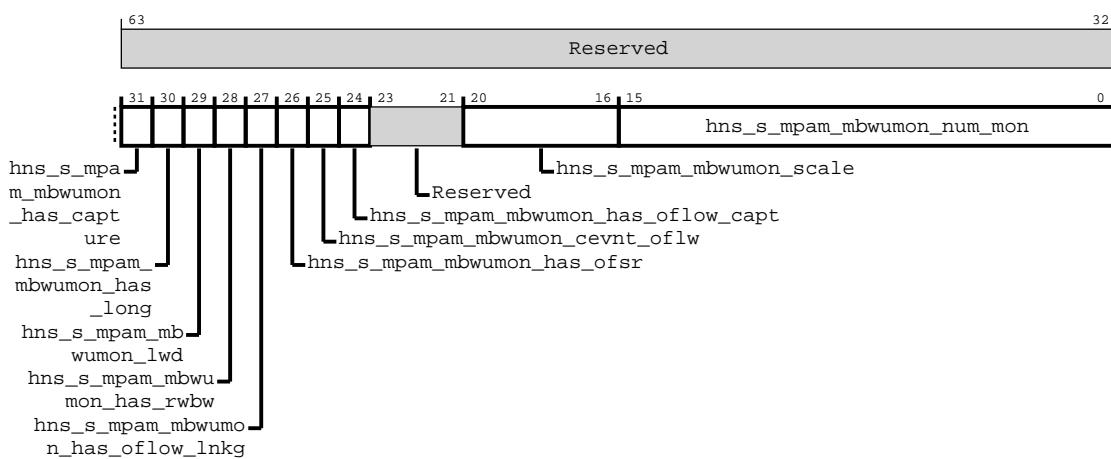
##### Usage constraints

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 8-683: cmn\_hns\_s\_mpam\_mbwumon\_idr**



**Table 8-691: cmn\_hns\_s\_mpam\_mbwumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	hns_s_mpam_mbwumon_has_capture	<p><b>0</b> MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature.</p> <p><b>1</b> This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.</p>	RO	0x0
[30]	hns_s_mpam_mbwumon_has_long	<p><b>0</b> MSMON_MBWU_L is not implemented.</p> <p><b>1</b> MSMON_MBWU_L is implemented.</p>	RO	0b0
[29]	hns_s_mpam_mbwumon_lwd	<p><b>0</b> MSMON_MBWU_L has 44-bit VALUE field in bits [43:0].</p> <p><b>1</b> MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].</p>	RO	0b0
[28]	hns_s_mpam_mbwumon_has_rwbw	<p><b>0</b> Read/write bandwidth selection is not implemented.</p> <p><b>1</b> Read/write bandwidth selection is implemented.</p>	RO	0b0
[27]	hns_s_mpam_mbwumon_has_oflow_lnk	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p>	RO	0b0
[26]	hns_s_mpam_mbwumon_has_ofsr	<p><b>0</b> MSMON_MBWU_OFSR register is not implemented</p> <p><b>1</b> MSMON_MBWU_OFSR register is implemented</p>	RO	0b0
[25]	hns_s_mpam_mbwumon_cevnt_oflw	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p>	RO	0b0
[24]	hns_s_mpam_mbwumon_has_oflow_capt	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p>	RO	0b0
[23:21]	Reserved	Reserved	RO	
[20:16]	hns_s_mpam_mbwumon_scale	Scaling of MSMON_MBWU.VALUE in bits.	RO	0x0
[15:0]	hns_s_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	0x0

### 8.3.11.16 cmn\_hns\_s\_mpam\_ecr

MPAM Error Control Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x10F0

##### Type

RW

##### Reset value

See individual bit resets

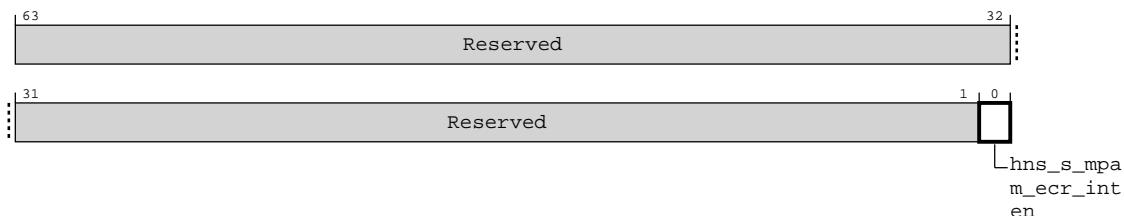
##### Usage constraints

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 8-684: cmn\_hns\_s\_mpam\_ecr**



**Table 8-692: 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	0x0

### 8.3.11.17 cmn\_hns\_s\_mpam\_esr

MPAM Error Status Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x10F8

##### Type

RW

##### Reset value

See individual bit resets

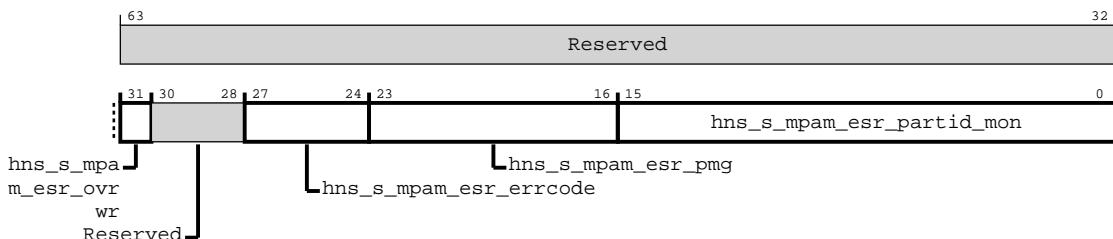
##### Usage constraints

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 8-685: cmn\_hns\_s\_mpam\_esr**



**Table 8-693: cmn\_hns\_s\_mpam\_esr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_s_mpam_esr_ovrwr	Overwritten. If 0 and ERRCODE is zero, no errors have occurred. If 0 and ERRCODE is non-zero, a single error has occurred and is recorded in this register. If 1 and ERRCODE is non-zero, multiple errors have occurred and this register records the most recent error. The state where this bit is 1 and ERRCODE is zero is not produced by hardware and is only reached when software writes this combination into this register.	RW	0x0
[30:28]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[27:24]	hns_s_mpam_esr_errcode	Error code	RW	0x0
[23:16]	hns_s_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	0x0
[15:0]	hns_s_mpam_esr_partid_mon	PARTID captured if the error code captures PARTID. MON selector captured if the error code captures MON. Otherwise 0x0000.	RW	0x0

### 8.3.11.18 cmn\_hns\_s\_mpamcfg\_part\_sel

MPAM partition configuration selection register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1100

##### Type

RW

##### Reset value

See individual bit resets

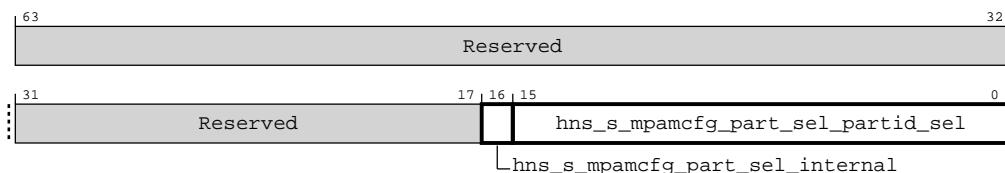
##### Usage constraints

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 8-686: cmn\_hns\_s\_mpamcfg\_part\_sel**



**Table 8-694: cmn\_hns\_s\_mpamcfg\_part\_sel attributes**

Bits	Name	Description	Type	Reset
[63:17]	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 <b>RAZ/WI</b> . If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interpret PARTID_SEL.	RW	0x0
[15:0]	hns_s_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	0x0

### 8.3.11.19 cmn\_hns\_s\_mpamcfg\_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1108

##### Type

RW

##### Reset value

See individual bit resets

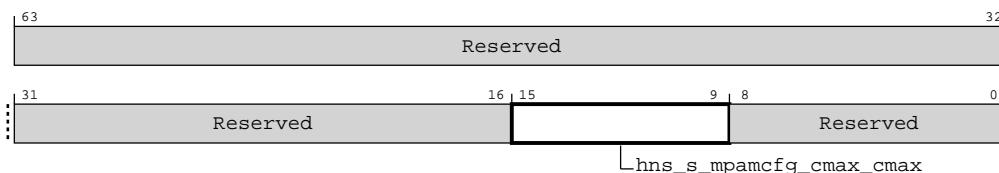
##### Usage constraints

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 8-687: cmn\_hns\_s\_mpamcfg\_cmax**



**Table 8-695: cmn\_hns\_s\_mpamcfg\_cmax attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:9]	hns_s_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	0b1111111
[8:0]	Reserved	Reserved	RO	-

### 8.3.11.20 cmn\_hns\_s\_mpamcfg\_mbw\_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1200

##### Type

RW

##### Reset value

See individual bit resets

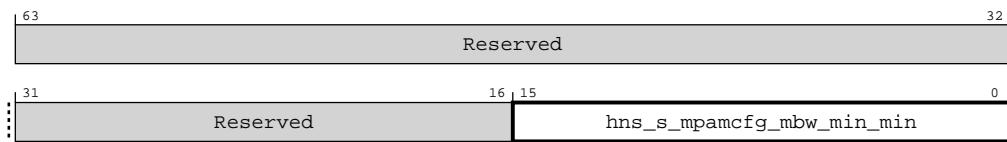
##### Usage constraints

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 8-688: cmn\_hns\_s\_mpamcfg\_mbw\_min**



**Table 8-696: cmn\_hns\_s\_mpamcfg\_mbw\_min attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	hns_s_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	0x0

### 8.3.11.21 cmn\_hns\_s\_mpamcfg\_mbw\_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1208

##### Type

RW

##### Reset value

See individual bit resets

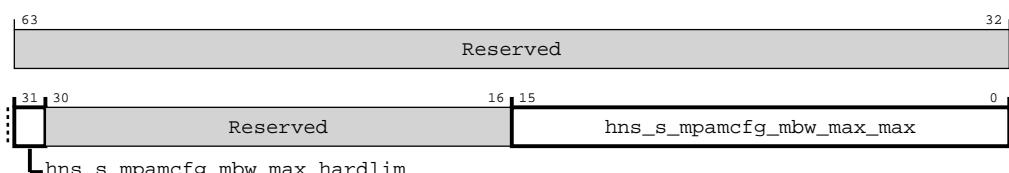
##### Usage constraints

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 8-689: cmn\_hns\_s\_mpamcfg\_mbw\_max**



**Table 8-697: cmn\_hns\_s\_mpamcfg\_mbw\_max attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	hns_s_mpamcfg_mbw_max_hardlim	<p><b>0</b> When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth.</p> <p><b>1</b> When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.</p>	RW	0x0
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_s_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	0x0

### 8.3.11.22 cmn\_hns\_s\_mpamcfg\_mbw\_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1220

##### Type

RW

##### Reset value

See individual bit resets

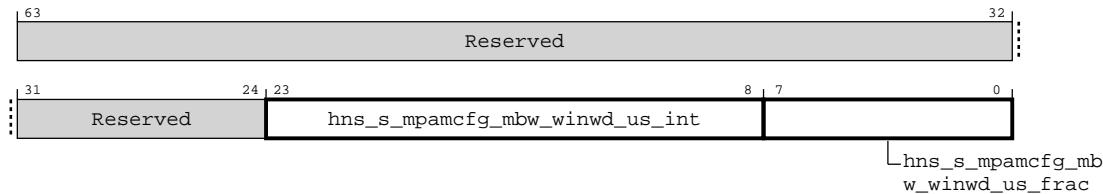
##### Usage constraints

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 8-690: cmn\_hns\_s\_mpamcfg\_mbw\_winwd**



**Table 8-698: 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	0x0
[7:0]	hns_s_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	0x0

### 8.3.11.23 cmn\_hns\_s\_mpamcfg\_pri

MPAM priority partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1400

##### Type

RW

##### Reset value

See individual bit resets

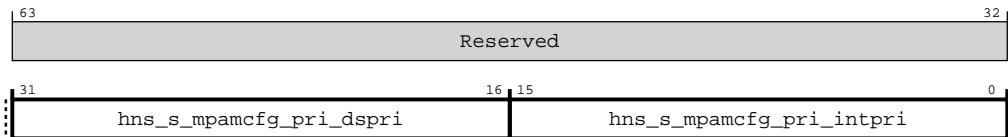
##### Usage constraints

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 8-691: cmn\_hns\_s\_mpamcfg\_pri**



**Table 8-699: 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	0x0
[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	0x0

### 8.3.11.24 cmn\_hns\_s\_mpamcfg\_mbw\_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1500

##### Type

RW

##### Reset value

See individual bit resets

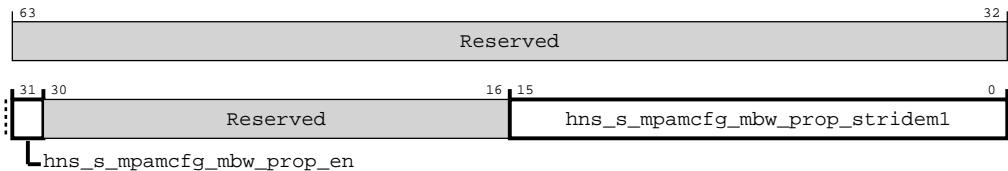
##### Usage constraints

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 8-692: cmn\_hns\_s\_mpamcfg\_mbw\_prop**



**Table 8-700: cmn\_hns\_s\_mpamcfg\_mbw\_prop attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_s_mpamcfg_mbw_prop_en	<b>0</b> The selected partition is not regulated by proportional stride bandwidth partitioning. <b>1</b> The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	0x0
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_s_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	0x0

### 8.3.11.25 cmn\_hns\_s\_mpamcfg\_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1600

##### Type

RW

##### Reset value

See individual bit resets

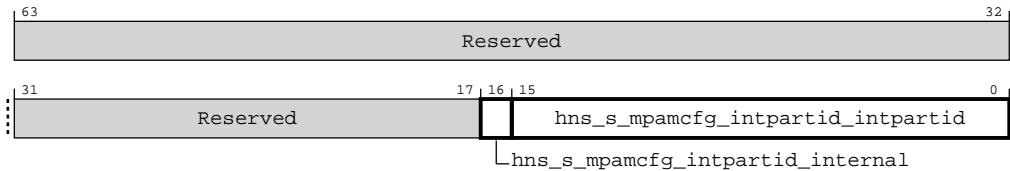
##### Usage constraints

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 8-693: cmn\_hns\_s\_mpamcfg\_intpartid**



**Table 8-701: 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	0x0
[15:0]	hns_s_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	0x0

## 8.3.11.26 cmn\_hns\_s\_msmon\_cfg\_mon\_sel

Memory system performance monitor selection register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x1800

#### Type

RW

#### Reset value

See individual bit resets

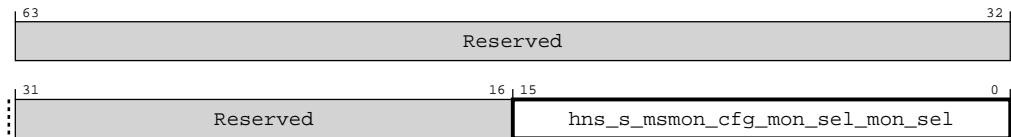
#### Usage constraints

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 8-694: cmn\_hns\_s\_msmon\_cfg\_mon\_sel**



**Table 8-702: 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	0x0

### 8.3.11.27 cmn\_hns\_s\_msmon\_capt\_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1808

##### Type

RW

##### Reset value

See individual bit resets

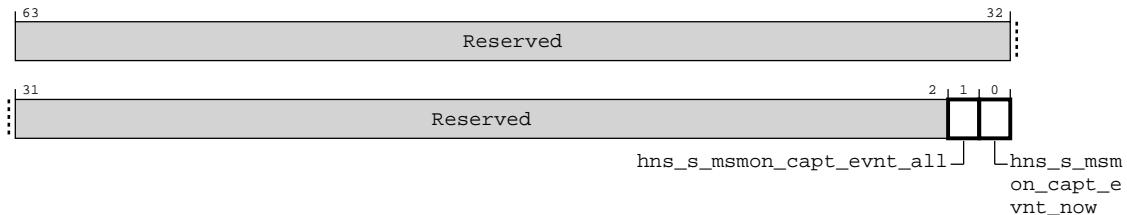
##### Usage constraints

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 8-695: cmn\_hns\_s\_msmon\_capt\_evnt**



**Table 8-703: cmn\_hns\_s\_msmon\_capt\_evnt attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	hns_s_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version if NOW is written as 1, signal a capture event to non-secure monitors in this memory system component with CAPT_EVNT = 7. 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	0x0
[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	0x0

### 8.3.11.28 cmn\_hns\_s\_msmon\_cfg\_csu\_flt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1810

##### Type

RW

### Reset value

See individual bit resets

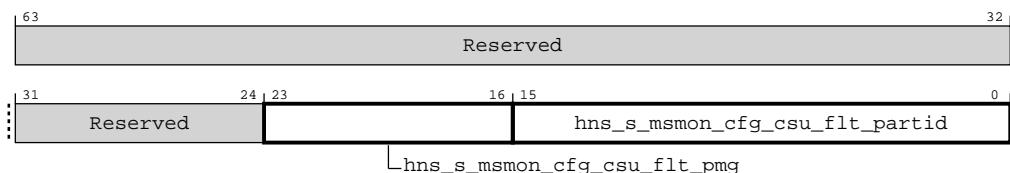
### Usage constraints

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 8-696: cmn\_hns\_s\_msmon\_cfg\_csu\_flt**



**Table 8-704: cmn\_hns\_s\_msmon\_cfg\_csu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_s_msmon_cfg_csu_flt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	0x0
[15:0]	hns_s_msmon_cfg_csu_flt_partid	Configures the cache storage usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	0x0

### 8.3.11.29 cmn\_hns\_s\_msmon\_cfg\_csu\_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1818

##### Type

RW

## Reset value

See individual bit resets

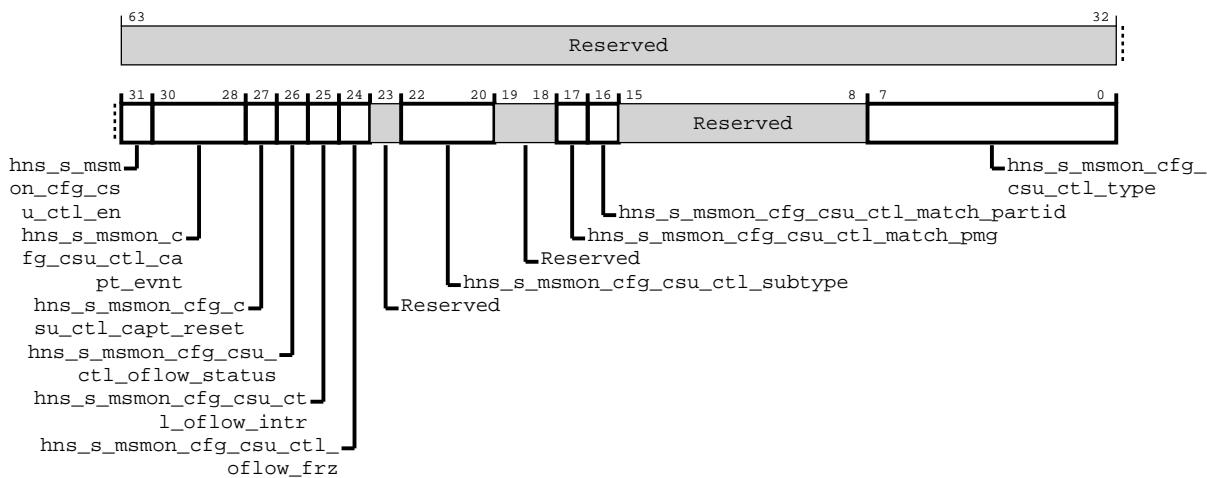
## Usage constraints

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 8-697: cmn\_hns\_s\_msmon\_cfg\_csu\_ctl**



**Table 8-705: 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	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_s_msmon_cfg_csu_ctl_capt_evnt	Select the event that triggers capture from the following:	RW	0x0
		<p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>		
[27]	hns_s_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	0x0
[26]	hns_s_msmon_cfg_csu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[25]	hns_s_msmon_cfg_csu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_s_msmon_cfg_csu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[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	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_s_msmon_cfg_csu_ctl_match_pmg	<p><b>0</b> Monitor storage used by all PMG values.</p> <p><b>1</b> Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0
[16]	hns_s_msmon_cfg_csu_ctl_match_partid	<p><b>0</b> Monitor storage used by all PARTIDs.</p> <p><b>1</b> Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_s_msmon_cfg_csu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	0x43

### 8.3.11.30 cmn\_hns\_s\_msmon\_cfg\_mbwu\_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1820

## Type

RW

## Reset value

See individual bit resets

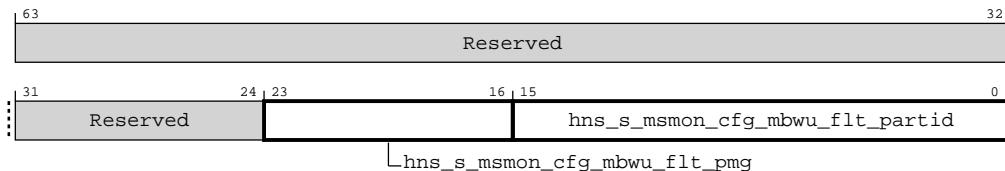
## Usage constraints

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 8-698: cmn\_hns\_s\_msmon\_cfg\_mbwu\_flt**



**Table 8-706: cmn\_hns\_s\_msmon\_cfg\_mbwu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_s_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0
[15:0]	hns_s_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0

## 8.3.11.31 cmn\_hns\_s\_msmon\_cfg\_mbwu\_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.  
This register is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

## Address offset

0x1828

## Type

RW

## Reset value

See individual bit resets

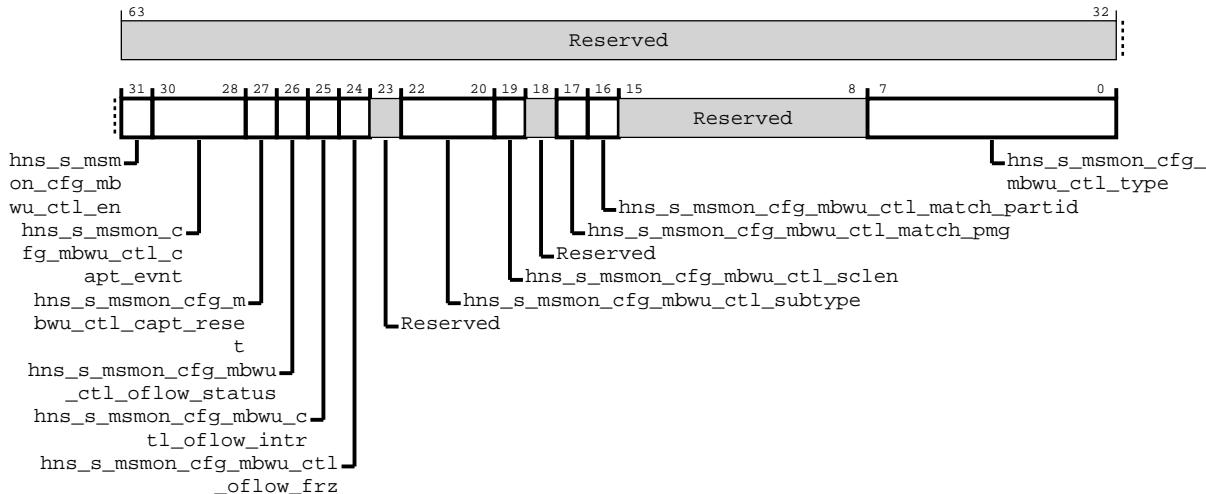
## Usage constraints

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 8-699: cmn\_hns\_s\_msmon\_cfg\_mbwu\_ctl**



**Table 8-707: 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	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_s_msmon_cfg_mbwu_ctl_capt_evnt	<p>Select the event that triggers capture from the following:</p> <p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>	RW	0x0

Bits	Name	Description	Type	Reset
[27]	hns_s_msmon_cfg_mbwu_ctl_capt_reset	<p><b>0</b> Monitor is not reset on capture.</p> <p><b>1</b> Monitor is reset on capture.</p>	RW	0x0
[26]	hns_s_msmon_cfg_mbwu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0
[25]	hns_s_msmon_cfg_mbwu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_s_msmon_cfg_mbwu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[23]	Reserved	Reserved	RO	
[22:20]	hns_s_msmon_cfg_mbwu_ctl_subtype	<p>A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports:</p> <p><b>0</b> Do not count any bandwidth.</p> <p><b>1</b> Count bandwidth used by memory reads</p> <p><b>2</b> Count bandwidth used by memory writes</p> <p><b>3</b> 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 <b>UNPREDICTABLE</b>.</p>	RW	0x0
[19]	hns_s_msmon_cfg_mbwu_ctl_sclen	<p><b>0</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance.</p> <p><b>1</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.</p>	RW	0x0
[18]	Reserved	Reserved	RO	
[17]	hns_s_msmon_cfg_mbwu_ctl_match_pmg	<p><b>0</b> Monitor bandwidth used by all PMG values.</p> <p><b>1</b> Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[16]	hns_s_msmon_cfg_mbwu_ctl_match_partid	<p><b>0</b> Monitor bandwidth used by all PARTIDs.</p> <p><b>1</b> Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_s_msmon_cfg_mbwu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	0x42

### 8.3.11.32 cmn\_hns\_s\_msmon\_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1840

##### Type

RW

##### Reset value

See individual bit resets

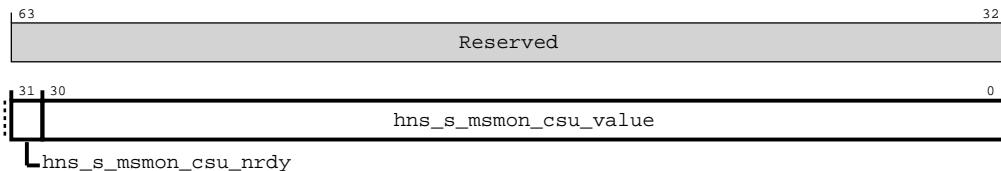
##### Usage constraints

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 8-700: cmn\_hns\_s\_msmon\_csu**



**Table 8-708: 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	0x0
[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	0x0

### 8.3.11.33 cmn\_hns\_s\_msmon\_csu\_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1848

##### Type

RW

##### Reset value

See individual bit resets

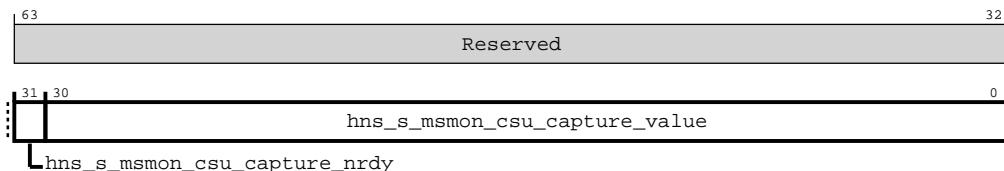
##### Usage constraints

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 8-701: cmn\_hns\_s\_msmon\_csu\_capture**



**Table 8-709: 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	0x0
[30:0]	hns_s_msmon_csu_capture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	0x0

### 8.3.11.34 cmn\_hns\_s\_msmon\_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1860

##### Type

RW

##### Reset value

See individual bit resets

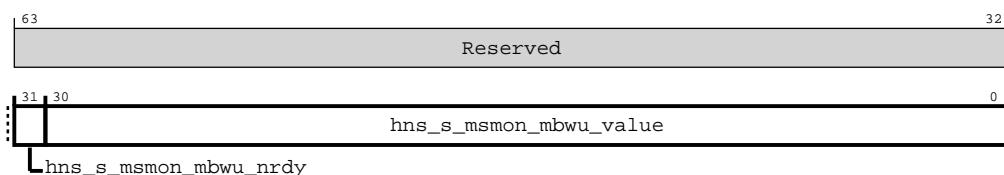
##### Usage constraints

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 8-702: cmn\_hns\_s\_msmon\_mbwu



**Table 8-710: 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	0x0
[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	0x0

### 8.3.11.35 cmn\_hns\_s\_msmon\_mbwu\_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1868

##### Type

RW

##### Reset value

See individual bit resets

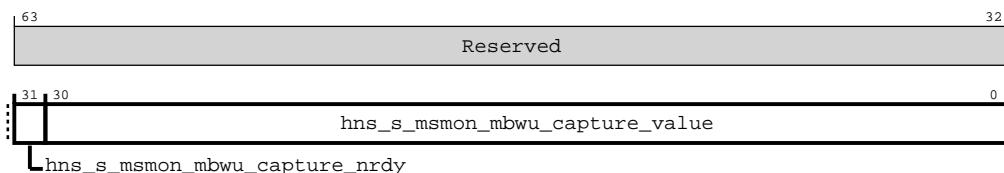
##### Usage constraints

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 8-703: cmn\_hns\_s\_msmon\_mbwu\_capture**



**Table 8-711: 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	0x0
[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	0x0

### 8.3.11.36 cmn\_hns\_s\_mpamcfg\_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x2000

##### Type

RW

##### Reset value

See individual bit resets

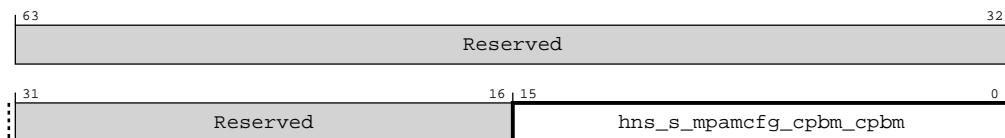
##### Usage constraints

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 8-704: cmn\_hns\_s\_mpamcfg\_cpbm**



**Table 8-712: cmn\_hns\_s\_mpamcfg\_cpbm attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_s_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL.	RW	0xFFFF

### 8.3.11.37 cmn\_hns\_rt\_mpam\_idr

MPAM features ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8000

##### Type

RO

##### Reset value

See individual bit resets

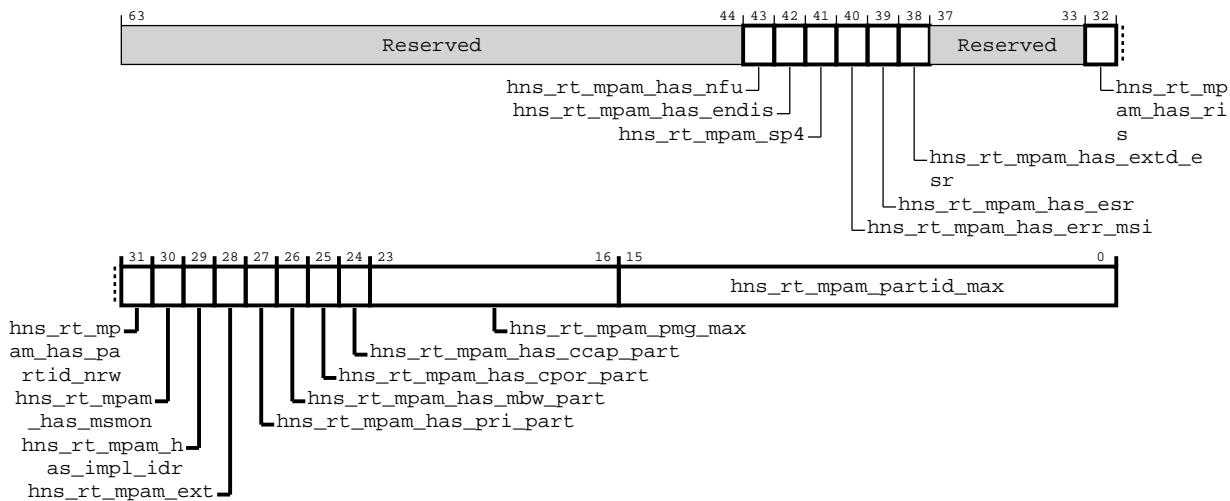
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-705: cmn\_hns\_rt\_mpam\_idr**



**Table 8-713: cmn\_hns\_rt\_mpam\_idr attributes**

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_rt_mpam_has_nfu	<p><b>0</b> HN-F does not support no future use field</p> <p><b>1</b> HN-F supports no future use field</p>	RO	0b0
[42]	hns_rt_mpam_has_endis	<p><b>0</b> HN-F does not support PARTID enable and disable functionality</p> <p><b>1</b> HN-F supports PARTID enable and disable functionality</p>	RO	0b0
[41]	hns_rt_mpam_sp4	<p><b>0</b> HN-F supports two PARTID spaces</p> <p><b>1</b> HN-F supports four PARTID spaces</p>	RO	0b1
[40]	hns_rt_mpam_has_err_msi	<p><b>0</b> HN-F does not support MSI writes to signal MPAM error interrupt</p> <p><b>1</b> HN-F supports MSI writes to signal MPAM error interrupt</p>	RO	0b0
[39]	hns_rt_mpam_has_esr	<p><b>0</b> HN-F does not support MPAM error handling</p> <p><b>1</b> HN-F supports MPAM error handling</p>	RO	0b1
[38]	hns_rt_mpam_has_extd_esr	<p><b>0</b> MPAMF_ESR is 32 bits</p> <p><b>1</b> MPAMF_ESR is 64 bits</p>	RO	0b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_rt_mpam_has_ris	<p><b>0</b> HN-F does not support MPAM resource instance selector</p> <p><b>1</b> HN-F supports MPAM resource instance selector</p>	RO	0b0
[31]	hns_rt_mpam_has_partid_nrw	<p><b>0</b> HN-F does not support MPAM PARTID Narrowing</p> <p><b>1</b> HN-F supports MPAM PARTID Narrowing</p>	RO	0x0
[30]	hns_rt_mpam_has_msmon	<p><b>0</b> MPAM performance monitoring is not supported</p> <p><b>1</b> MPAM performance monitoring is supported</p>	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[29]	hns_rt_mpam_has_impl_idr	<p><b>0</b> MPAM implementation specific partitioning features not supported</p> <p><b>1</b> MPAM implementation specific partitioning features supported</p>	RO	0x0
[28]	hns_rt_mpam_ext	<p><b>0</b> HN-F has no defined bits in [63:32]</p> <p><b>1</b> HN-F has bits defined in [63:32]</p>	RO	0b1
[27]	hns_rt_mpam_has_pri_part	<p><b>0</b> MPAM priority partitioning is not supported</p> <p><b>1</b> MPAM priority partitioning is supported</p>	RO	0x0
[26]	hns_rt_mpam_has_mbw_part	<p><b>0</b> MPAM memory bandwidth partitioning is not supported</p> <p><b>1</b> MPAM memory bandwidth partitioning is supported</p>	RO	0x0
[25]	hns_rt_mpam_has_cpor_part	<p><b>0</b> MPAM cache portion partitioning is not supported</p> <p><b>1</b> MPAM cache portion partitioning is supported</p>	RO	Configuration dependent
[24]	hns_rt_mpam_has_ccap_part	<p><b>0</b> MPAM cache maximum capacity partitioning is not supported</p> <p><b>1</b> MPAM cache maximum capacity partitioning is supported</p>	RO	Configuration dependent
[23:16]	hns_rt_mpam_pmg_max	Maximum value of root PMG supported by this HN-F	RO	0x2
[15:0]	hns_rt_mpam_partid_max	Maximum value of root PARTID supported by this HN-F	RO	0x1

### 8.3.11.38 cmn\_hns\_rt\_mpam\_iidr

MPAM Implementation ID register. This is a shared register for S and NS

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8018

## Type

RO

## Reset value

See individual bit resets

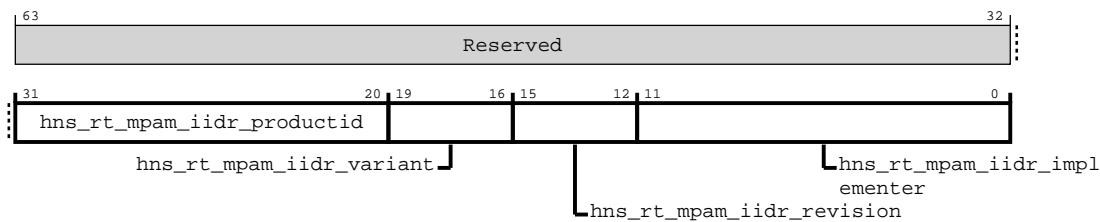
## Usage constraints

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 8-706: cmn\_hns\_rt\_mpam\_iidr**



**Table 8-714: cmn\_hns\_rt\_mpam\_iidr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_rt_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	0x3e
[19:16]	hns_rt_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	Configuration dependent
[15:12]	hns_rt_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	0b0000
[11:0]	hns_rt_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	0x43B

## 8.3.11.39 cmn\_hns\_rt\_mpam\_aids

MPAM architecture ID register. This is a shared register for S and NS

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

## Address offset

0x8020

## Type

RO

## Reset value

See individual bit resets

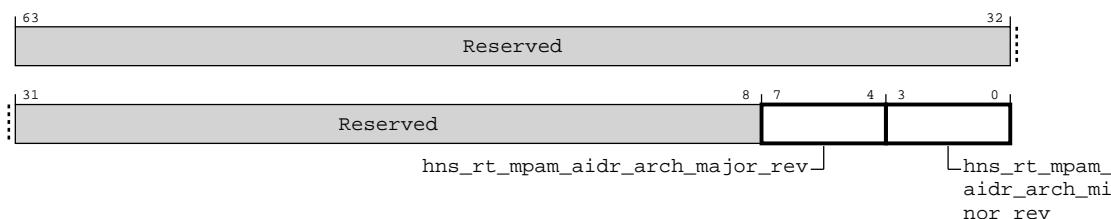
## Usage constraints

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 8-707: cmn\_hns\_rt\_mpam\_aidr**



**Table 8-715: cmn\_hns\_rt\_mpam\_aidr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_rt_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	0b0001
[3:0]	hns_rt_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	0b0001

## 8.3.11.40 cmn\_hns\_rt\_mpam\_impl\_idr

MPAM Implementation defined partitioning feature ID register. This is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8028

### Type

RO

### Reset value

See individual bit resets

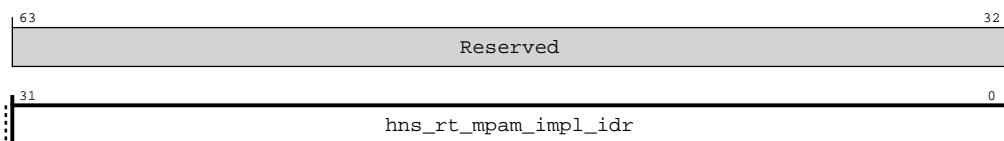
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-708: cmn\_hns\_rt\_mpam\_impl\_idr**



**Table 8-716: 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	0x00000000

### 8.3.11.41 cmn\_hns\_rt\_mpam\_cpor\_idr

MPAM cache portion partitioning ID register. This is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8030

#### Type

RO

#### Reset value

See individual bit resets

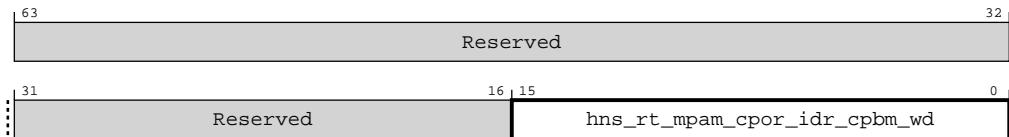
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-709: cmn\_hns\_rt\_mpam\_cpor\_idr**



**Table 8-717: 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	0x10

## 8.3.11.42 cmn\_hns\_rt\_mpam\_ccap\_idr

MPAM cache capacity partitioning ID register. This is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8038

### Type

RO

### Reset value

See individual bit resets

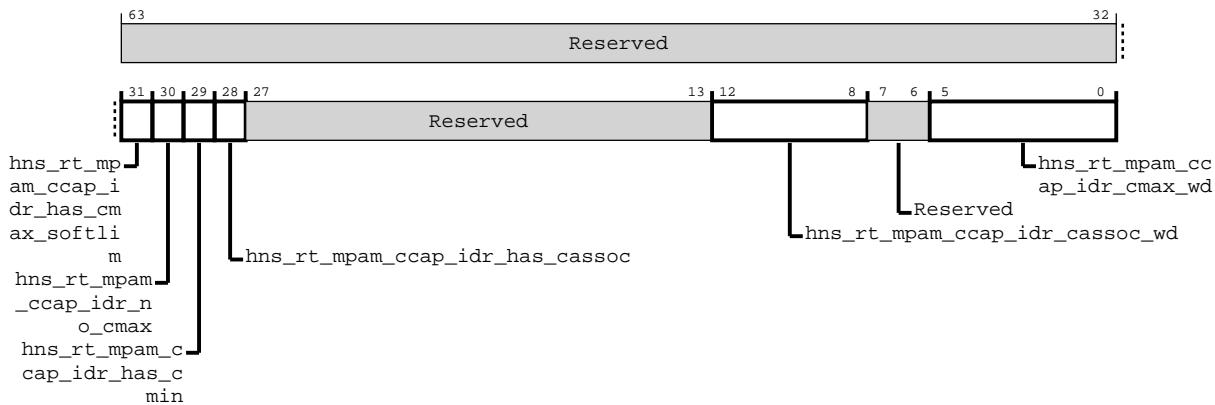
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-710: cmn\_hns\_rt\_mpam\_ccap\_idr**



**Table 8-718: cmn\_hns\_rt\_mpam\_ccap\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_mpam_ccap_idr_has_cmax_softlim	<p><b>0</b> HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit</p> <p><b>1</b> HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit</p>	RO	0x0
[30]	hns_rt_mpam_ccap_idr_no_cmax	<p><b>0</b> HN-F support MPAMCFG_CMAX</p> <p><b>1</b> HN-F doesn't support MPAMCFG_CMAX</p>	RO	0x0
[29]	hns_rt_mpam_ccap_idr_has_cmin	<p><b>0</b> HN-F does not support MPAMCFG_CMIN</p> <p><b>1</b> HN-F supports MPAMCFG_CMIN</p>	RO	0x0
[28]	hns_rt_mpam_ccap_idr_has_cassoc	<p><b>0</b> HN-F does not support MPAMCFG_CASSOC</p> <p><b>1</b> HN-F supports MPAMCFG_CASSOC</p>	RO	0x0
[27:13]	Reserved	Reserved	RO	-
[12:8]	hns_rt_mpam_ccap_idr_cassoc_wd	Number of fractional bits implemented in the cache associativity partitioning.	RO	0x0
[7:6]	Reserved	Reserved	RO	-
[5:0]	hns_rt_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	0x7

### 8.3.11.43 cmn\_hns\_rt\_mpam\_mbw\_idr

MPAM Memory Bandwidth partitioning ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8040

##### Type

RO

##### Reset value

See individual bit resets

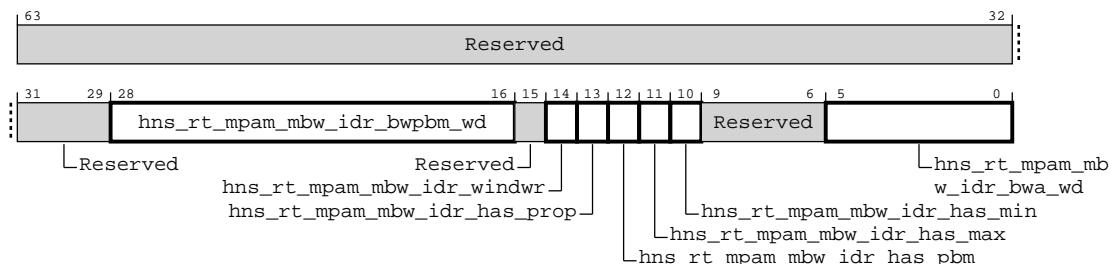
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-711: cmn\_hns\_rt\_mpam\_mbw\_idr**



**Table 8-719: 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	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14]	hns_rt_mpam_mbw_idr_windwr	<p><b>0</b> The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed.</p> <p><b>1</b> The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.</p>	RO	0x0
[13]	hns_rt_mpam_mbw_idr_has_prop	<p><b>0</b> There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register</p> <p><b>1</b> MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.</p>	RO	0x0
[12]	hns_rt_mpam_mbw_idr_has_pbm	<p><b>0</b> There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register</p> <p><b>1</b> MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.</p>	RO	0x0
[11]	hns_rt_mpam_mbw_idr_has_max	<p><b>0</b> There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register</p> <p><b>1</b> MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[10]	hns_rt_mpam_mbw_idr_has_min	<p><b>0</b> There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register</p> <p><b>1</b> MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.</p>	RO	0x0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_rt_mpam_mbw_idr_bwa_wd	<p>Number of implemented bits in bandwidth allocation</p> <p><b>fields</b></p> <p>MIN, MAX, and STRIDE. Value must be between 1 to 16</p>	RO	0b0000

### 8.3.11.44 cmn\_hns\_rt\_mpam\_pri\_idr

MPAM Priority partitioning ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8048

### Type

RO

### Reset value

See individual bit resets

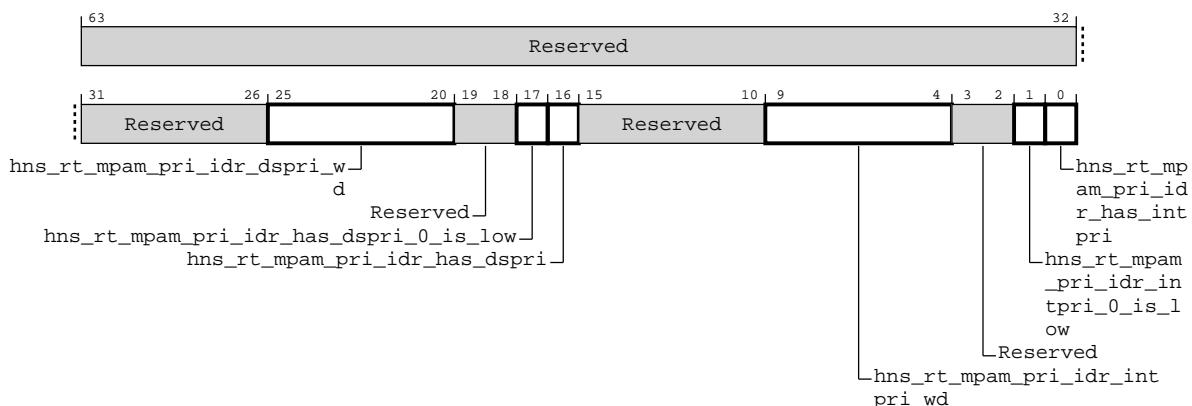
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-712: cmn\_hns\_rt\_mpam\_pri\_idr**



**Table 8-720: 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	0x0
[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	0x0
[16:15]	Reserved			
[10:9]	Reserved			
[4:3]	Reserved			
[2]	hns_rt_mpam_pri_idr_in_tpri_0_is_low			
[1]	hns_rt_mpam_pri_idr_int_pri_wd			
[0]	Reserved			

Bits	Name	Description	Type	Reset
[16]	hns_rt_mpam_pri_idr_has_dspri	<p><b>0</b> This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports downstream priority and has an DSPRI field.</p>	RO	0x0
[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	0x0
[3:2]	Reserved	Reserved	RO	
[1]	hns_rt_mpam_pri_idr_intpri_0_is_low	<p><b>0</b> In the INTPRI field, a value of 0 means highest priority.</p> <p><b>1</b> In the INTPRI field, a value of 0 means lowest priority.</p>	RO	0x0
[0]	hns_rt_mpam_pri_idr_has_intpri	<p><b>0</b> This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI.</p> <p><b>1</b> This memory system component supports internal priority and has an INTPRI field.</p>	RO	0x0

### 8.3.11.45 cmn\_hns\_rt\_mpam\_partid\_nrw\_idr

MPAM PARTID narrowing ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8050

##### Type

RO

##### Reset value

See individual bit resets

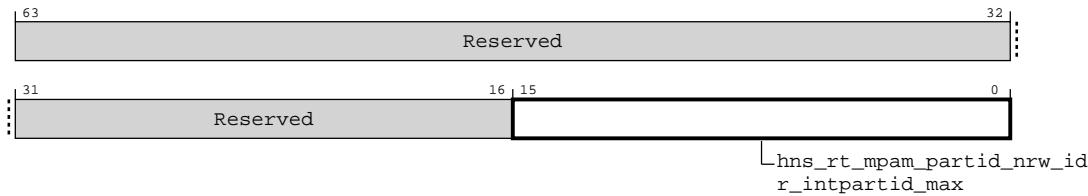
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-713: cmn\_hns\_rt\_mpam\_partid\_nrw\_idr**



**Table 8-721: 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	0x00

### 8.3.11.46 cmn\_hns\_rt\_mpam\_msmon\_idr

MPAM performance monitoring ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8080

##### Type

RO

##### Reset value

See individual bit resets

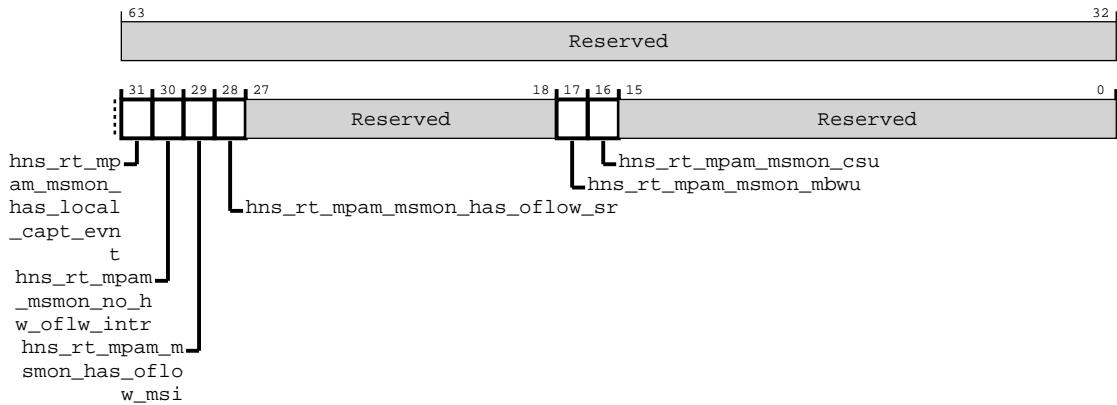
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-714: cmn\_hns\_rt\_mpam\_msmon\_idr**



**Table 8-722: cmn\_hns\_rt\_mpam\_msmon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_mpam_msmon_has_local_capt_evnt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	0x1
[30]	hns_rt_mpam_msmon_no_hw_oflw_intr	0 HNF has hardwired MPAM overflow interrupt 1 HNF doesn't have hardwired MPAM overflow interrupt	RO	0b1
[29]	hns_rt_mpam_msmon_has_oflow_msi	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	0b0
[28]	hns_rt_mpam_msmon_has_oflow_sr	0 HNF doesn't have overflow status register 1 HNF has overflow status register	RO	0b0
[27:18]	Reserved	Reserved	RO	-
[17]	hns_rt_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	0x0
[16]	hns_rt_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

### 8.3.11.47 cmn\_hns\_rt\_mpam\_csumon\_idr

MPAM cache storage usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8088

##### Type

RO

##### Reset value

See individual bit resets

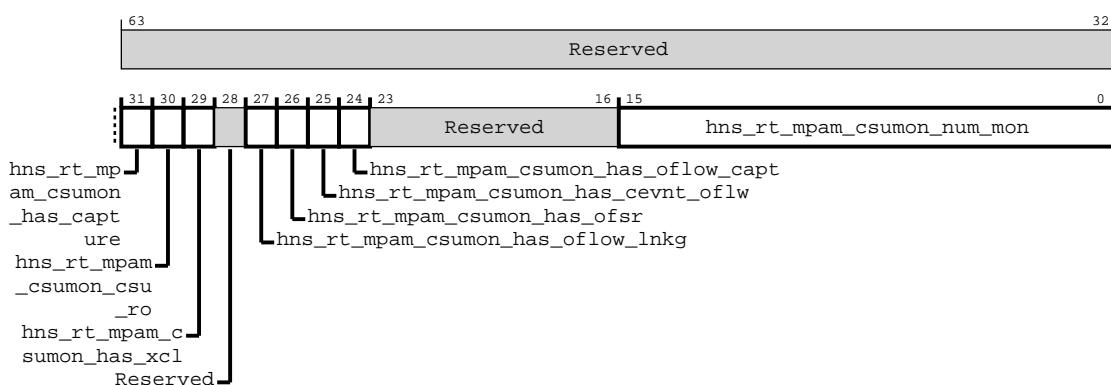
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-715: cmn\_hns\_rt\_mpam\_csumon\_idr**



**Table 8-723: cmn\_hns\_rt\_mpam\_csumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_rt_mpam_csumon_has_capture	<p><b>0</b> MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature.</p> <p><b>1</b> This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.</p>	RO	0x1
[30]	hns_rt_mpam_csumon_csu_ro	<p><b>0</b> MSMON_CSU is read/write.</p> <p><b>1</b> MSMON_CSU is read-only.</p>	RO	0b0
[29]	hns_rt_mpam_csumon_has_xcl	<p><b>0</b> MSMON_CFG_CSU_FLT does not implement the XCL field</p> <p><b>1</b> MSMON_CFG_CSU_FLT implements the XCL field</p>	RO	0b0
[28]	Reserved	Reserved	RO	-
[27]	hns_rt_mpam_csumon_has_oflow_lnk	<p><b>0</b> HNF doesn't support CSU overflow linkage</p> <p><b>1</b> HNF supports CSU overflow linkage</p>	RO	0b0
[26]	hns_rt_mpam_csumon_has_ofsr	<p><b>0</b> MSMON_CSU_OFSR register is not implemented</p> <p><b>1</b> MSMON_CSU_OFSR register is implemented</p>	RO	0b0
[25]	hns_rt_mpam_csumon_has_cevnt_oflw	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW</p>	RO	0b0
[24]	hns_rt_mpam_csumon_has_oflow_capt	<p><b>0</b> HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT</p> <p><b>1</b> HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT</p>	RO	0b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	0x1

### 8.3.11.48 cmn\_hns\_rt\_mpam\_mbwumon\_idr

MPAM memory bandwidth usage monitor ID register. This is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8090

### Type

RO

### Reset value

See individual bit resets

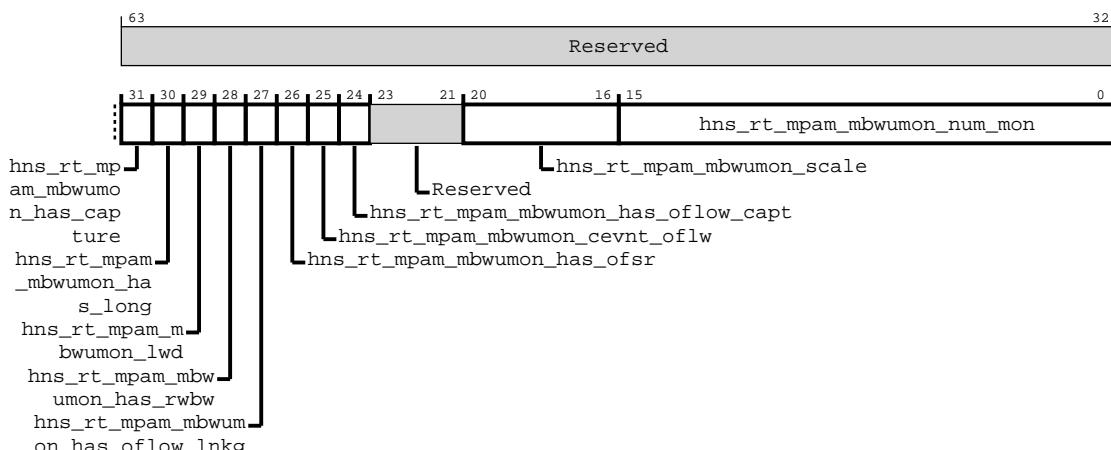
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-716: cmn\_hns\_rt\_mpam\_mbwumon\_idr**



**Table 8-724: cmn\_hns\_rt\_mpam\_mbwumon\_idr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rt_mpam_mbwumon_has_capture	<p><b>0</b></p> <p>MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature.</p> <p><b>1</b></p> <p>This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.</p>	RO	0x0

Bits	Name	Description	Type	Reset
[30]	hns_rt_mpam_mbwumon_has_long	<p><b>0</b> MSMON_MBWU_L is not implemented.</p> <p><b>1</b> MSMON_MBWU_L is implemented.</p>	RO	0b0
[29]	hns_rt_mpam_mbwumon_lwd	<p><b>0</b> MSMON_MBWU_L has 44-bit VALUE field in bits [43:0].</p> <p><b>1</b> MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].</p>	RO	0b0
[28]	hns_rt_mpam_mbwumon_has_rwbw	<p><b>0</b> Read/write bandwidth selection is not implemented.</p> <p><b>1</b> Read/write bandwidth selection is implemented.</p>	RO	0b0
[27]	hns_rt_mpam_mbwumon_has_oflow_lnkg	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.</p>	RO	0b0
[26]	hns_rt_mpam_mbwumon_has_ofsr	<p><b>0</b> MSMON_MBWU_OFSR register is not implemented</p> <p><b>1</b> MSMON_MBWU_OFSR register is implemented</p>	RO	0b0
[25]	hns_rt_mpam_mbwumon_cevnt_oflw	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.</p>	RO	0b0
[24]	hns_rt_mpam_mbwumon_has_oflow_capt	<p><b>0</b> Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p> <p><b>1</b> Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.</p>	RO	0b0
[23:21]	Reserved	Reserved	RO	
[20:16]	hns_rt_mpam_mbwumon_scale	Scalling of MSMON_MBWU.VALUE in bits.	RO	0x0
[15:0]	hns_rt_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	0x0

### 8.3.11.49 cmn\_hns\_rt\_mpam\_ecr

MPAM Error Control Register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x80F0

### Type

RW

### Reset value

See individual bit resets

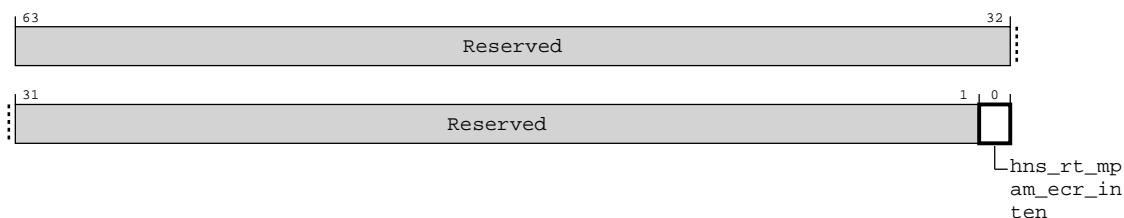
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-717: cmn\_hns\_rt\_mpam\_ecr**



**Table 8-725: 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	0x0

## 8.3.11.50 cmn\_hns\_rt\_mpam\_esr

MPAM Error Status Register. This register is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x80F8

### Type

RW

### Reset value

See individual bit resets

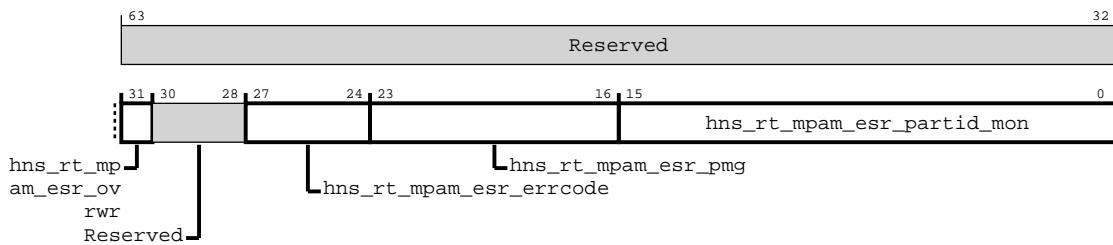
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-718: cmn\_hns\_rt\_mpam\_esr**



**Table 8-726: 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	0x0
[30:28]	Reserved	Reserved	RO	
[27:24]	hns_rt_mpam_esr_errcode	Error code	RW	0x0
[23:16]	hns_rt_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	0x0
[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	0x0

### 8.3.11.51 cmn\_hns\_rt\_mpamcfg\_part\_sel

MPAM partition configuration selection register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8100

### Type

RW

### Reset value

See individual bit resets

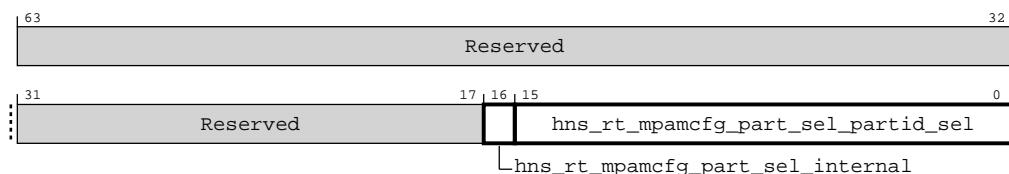
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-719: cmn\_hns\_rt\_mpamcfg\_part\_sel**



**Table 8-727: 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	0x0
[15:0]	hns_rt_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	0x0

## 8.3.11.52 cmn\_hns\_rt\_mpamcfg\_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8108

### Type

RW

### Reset value

See individual bit resets

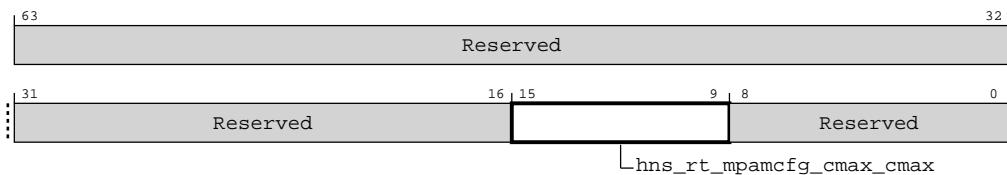
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-720: cmn\_hns\_rt\_mpamcfg\_cmax**



**Table 8-728: 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	0b1111111
[8:0]	Reserved	Reserved	RO	-

### 8.3.11.53 cmn\_hns\_rt\_mpamcfg\_mbw\_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8200

### Type

RW

### Reset value

See individual bit resets

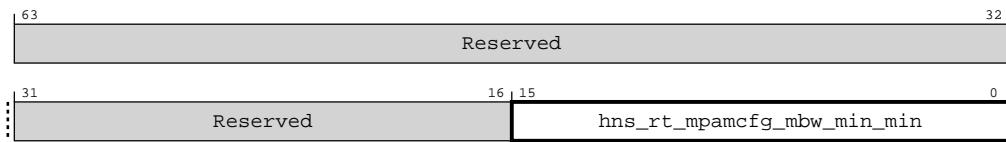
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-721: cmn\_hns\_rt\_mpamcfg\_mbw\_min**



**Table 8-729: 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	0x0

### 8.3.11.54 cmn\_hns\_rt\_mpamcfg\_mbw\_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8208

#### Type

RW

#### Reset value

See individual bit resets

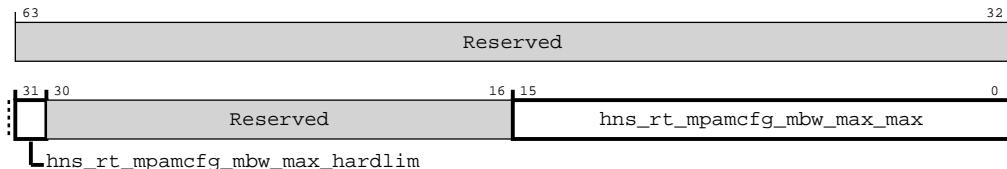
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-722: cmn\_hns\_rt\_mpamcfg\_mbw\_max**



**Table 8-730: cmn\_hns\_rt\_mpamcfg\_mbw\_max attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rt_mpamcfg_mbw_max_hardlim	<p><b>0</b> When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth.</p> <p><b>1</b> When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.</p>	RW	0x0
[30:16]	Reserved	Reserved	RO	
[15:0]	hns_rt_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	0x0

## 8.3.11.55 cmn\_hns\_rt\_mpamcfg\_mbw\_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8220

#### Type

RW

### Reset value

See individual bit resets

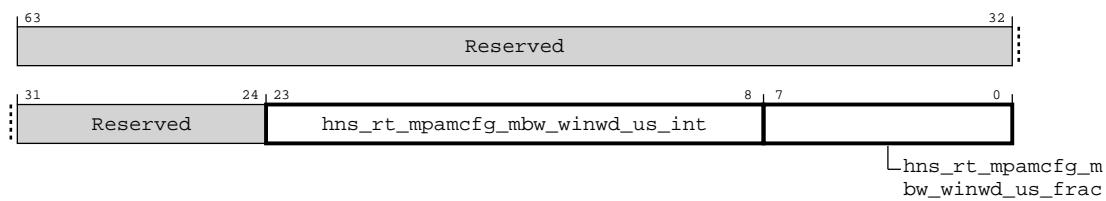
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-723: cmn\_hns\_rt\_mpamcfg\_mbw\_winwd**



**Table 8-731: 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	0x0
[7:0]	hns_rt_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	0x0

### 8.3.11.56 cmn\_hns\_rt\_mpamcfg\_pri

MPAM priority partitioning configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8400

#### Type

RW

#### Reset value

See individual bit resets

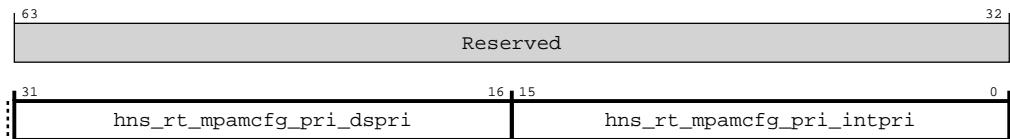
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-724: cmn\_hns\_rt\_mpamcfg\_pri**



**Table 8-732: 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	0x0
[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	0x0

## 8.3.11.57 cmn\_hns\_rt\_mpamcfg\_mbw\_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8500

#### Type

RW

#### Reset value

See individual bit resets

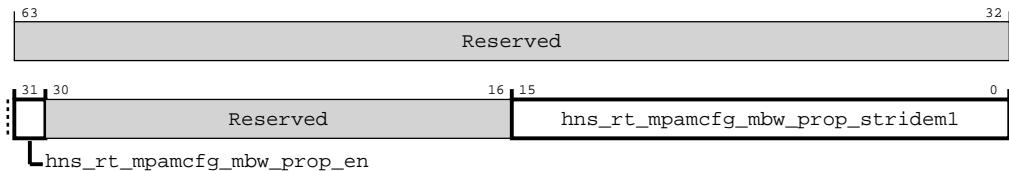
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-725: cmn\_hns\_rt\_mpamcfg\_mbw\_prop**



**Table 8-733: cmn\_hns\_rt\_mpamcfg\_mbw\_prop attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	hns_rt_mpamcfg_mbw_prop_en	<b>0</b> The selected partition is not regulated by proportional stride bandwidth partitioning. <b>1</b> The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	0x0
[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	0x0

### 8.3.11.58 cmn\_hns\_rt\_mpamcfg\_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8600

##### Type

RW

##### Reset value

See individual bit resets

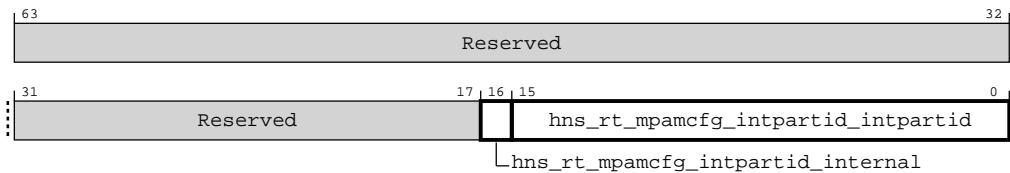
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-726: cmn\_hns\_rt\_mpamcfg\_intpartid**



**Table 8-734: 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	0x0
[15:0]	hns_rt_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	0x0

## 8.3.11.59 cmn\_hns\_rt\_msmon\_cfg\_mon\_sel

Memory system performance monitor selection register. This register is banked separately

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x8800

#### Type

RW

#### Reset value

See individual bit resets

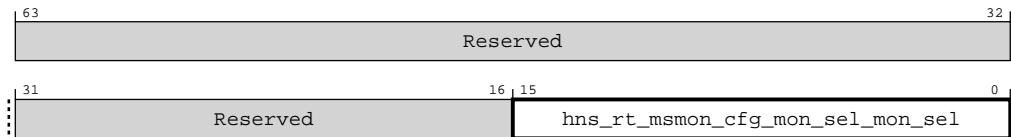
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-727: cmn\_hns\_rt\_msmon\_cfg\_mon\_sel**



**Table 8-735: 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	0x0

### 8.3.11.60 cmn\_hns\_rt\_msmon\_capt\_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8808

##### Type

RW

##### Reset value

See individual bit resets

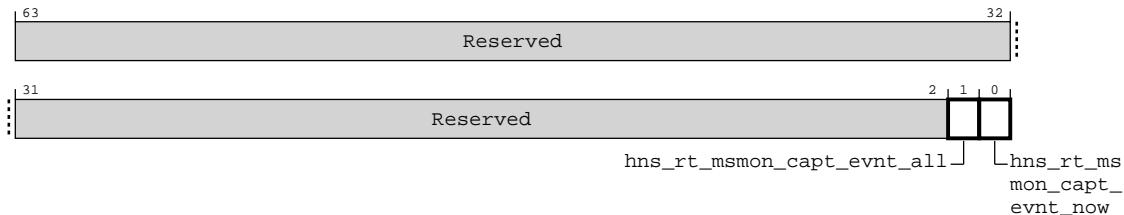
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-728: cmn\_hns\_rt\_msmon\_capt\_evnt**



**Table 8-736: cmn\_hns\_rt\_msmon\_capt\_evnt attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[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	0x0
[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	0x0

### 8.3.11.61 cmn\_hns\_rt\_msmon\_cfg\_csu\_flg

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8810

##### Type

RW

### Reset value

See individual bit resets

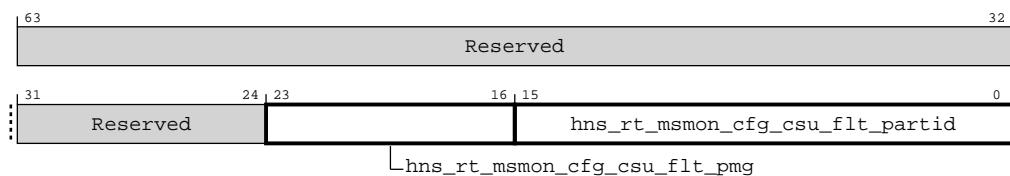
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-729: cmn\_hns\_rt\_msmon\_cfg\_csu\_flt**



**Table 8-737: 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	0x0
[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	0x0

### 8.3.11.62 cmn\_hns\_rt\_msmon\_cfg\_csu\_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8818

##### Type

RW

### Reset value

See individual bit resets

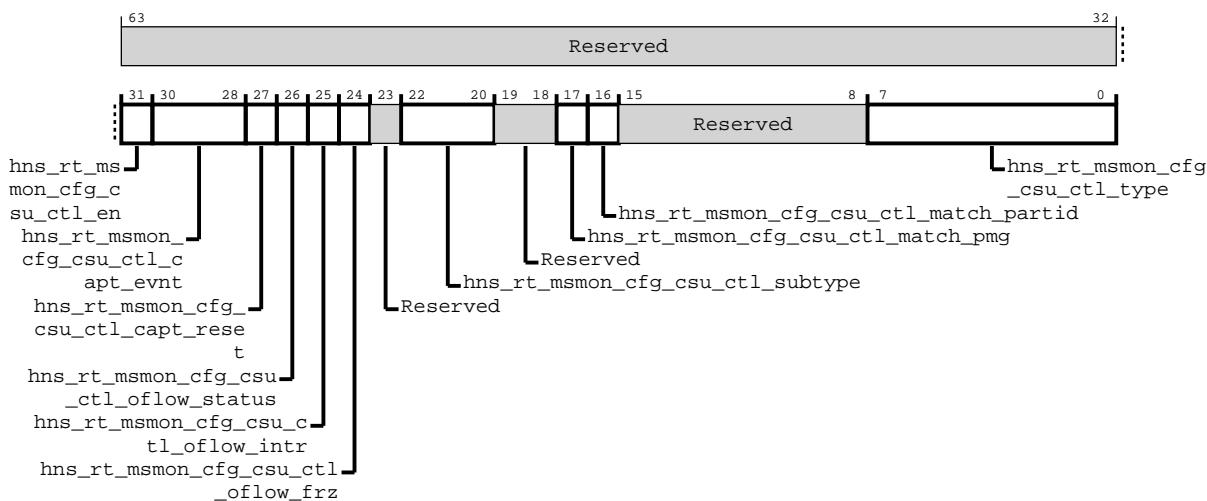
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-730: cmn\_hns\_rt\_msmon\_cfg\_csu\_ctl**



**Table 8-738: 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	0x0
[30:28]	hns_rt_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	0x0
[27]	hns_rt_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	0x0
[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	0x0

Bits	Name	Description	Type	Reset
[25]	hns_rt_msmon_cfg_csu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_rt_msmon_cfg_csu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[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	0x0
[19:18]	Reserved	Reserved	RO	
[17]	hns_rt_msmon_cfg_csu_ctl_match_pmg	<p><b>0</b> Monitor storage used by all PMG values.</p> <p><b>1</b> Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0
[16]	hns_rt_msmon_cfg_csu_ctl_match_partid	<p><b>0</b> Monitor storage used by all PARTIDs.</p> <p><b>1</b> Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_rt_msmon_cfg_csu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	0x43

### 8.3.11.63 cmn\_hns\_rt\_msmon\_cfg\_mbwu\_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register.  
This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8820

## Type

RW

## Reset value

See individual bit resets

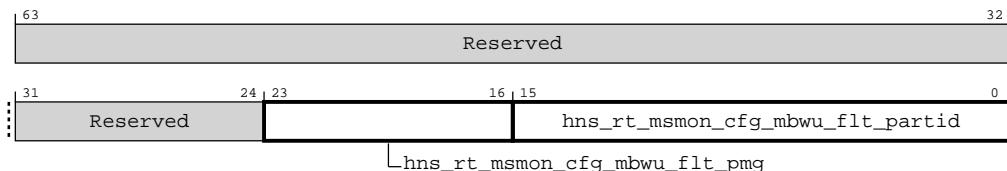
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-731: cmn\_hns\_rt\_msmon\_cfg\_mbwu\_flt**



**Table 8-739: cmn\_hns\_rt\_msmon\_cfg\_mbwu\_flt attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	
[23:16]	hns_rt_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0
[15:0]	hns_rt_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	0x0

## 8.3.11.64 cmn\_hns\_rt\_msmon\_cfg\_mbwu\_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.  
This register is banked separately

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x8828

### Type

RW

### Reset value

See individual bit resets

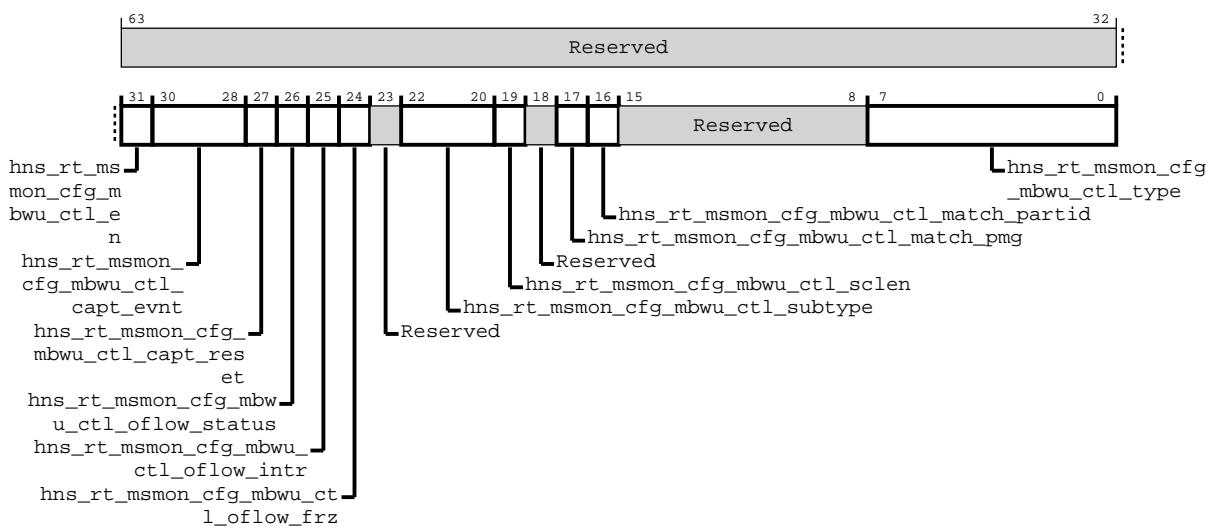
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-732: cmn\_hns\_rt\_msmon\_cfg\_mbwu\_ctl**



**Table 8-740: 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	<p><b>0</b> The monitor is disabled and must not collect any information.</p> <p><b>1</b> The monitor is enabled to collect information according to its configuration.</p>	RW	0x0
[30:28]	hns_rt_msmon_cfg_mbwu_ctl_capt_evnt	<p>Select the event that triggers capture from the following:</p> <p><b>0</b> No capture event is triggered.</p> <p><b>1</b> External capture event 1 (optional but recommended)</p>	RW	0x0

Bits	Name	Description	Type	Reset
[27]	hns_rt_msmon_cfg_mbwu_ctl_capt_reset	<p><b>0</b> Monitor is not reset on capture.</p> <p><b>1</b> Monitor is reset on capture.</p>	RW	0x0
[26]	hns_rt_msmon_cfg_mbwu_ctl_oflow_status	<p><b>0</b> No overflow has occurred.</p> <p><b>1</b> At least one overflow has occurred since this bit was last written.</p>	RW	0x0
[25]	hns_rt_msmon_cfg_mbwu_ctl_oflow_intr	<p><b>0</b> No interrupt.</p> <p><b>1</b> On overflow, an implementation-specific interrupt is signalled.</p>	RW	0x0
[24]	hns_rt_msmon_cfg_mbwu_ctl_oflow_frz	<p><b>0</b> Monitor count wraps on overflow.</p> <p><b>1</b> Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.</p>	RW	0x0
[23]	Reserved	Reserved	RO	
[22:20]	hns_rt_msmon_cfg_mbwu_ctl_subtype	<p>A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports:</p> <p><b>0</b> Do not count any bandwidth.</p> <p><b>1</b> Count bandwidth used by memory reads</p> <p><b>2</b> Count bandwidth used by memory writes</p> <p><b>3</b> 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 <b>UNPREDICTABLE</b>.</p>	RW	0x0
[19]	hns_rt_msmon_cfg_mbwu_ctl_sclen	<p><b>0</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance.</p> <p><b>1</b> MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.</p>	RW	0x0
[18]	Reserved	Reserved	RO	
[17]	hns_rt_msmon_cfg_mbwu_ctl_match_pmg	<p><b>0</b> Monitor bandwidth used by all PMG values.</p> <p><b>1</b> Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.</p>	RW	0x0

Bits	Name	Description	Type	Reset
[16]	hns_rt_msmon_cfg_mbwu_ctl_match_partid	<p><b>0</b> Monitor bandwidth used by all PARTIDs.</p> <p><b>1</b> Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.</p>	RW	0x0
[15:8]	Reserved	Reserved	RO	
[7:0]	hns_rt_msmon_cfg_mbwu_ctl_type	<b>Read-only</b> Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	0x42

### 8.3.11.65 cmn\_hns\_rt\_msmon\_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8840

##### Type

RW

##### Reset value

See individual bit resets

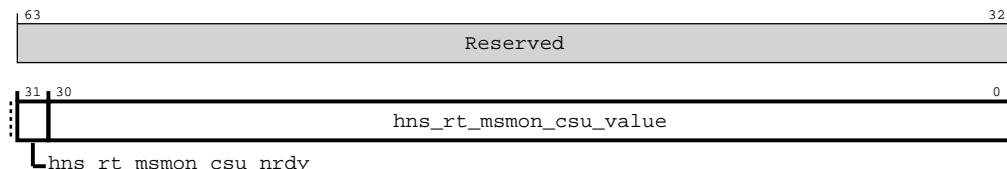
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-733: cmn\_hns\_rt\_msmon\_csu**



**Table 8-741: 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	0x0
[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	0x0

### 8.3.11.66 cmn\_hns\_rt\_msmon\_csu\_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8848

##### Type

RW

##### Reset value

See individual bit resets

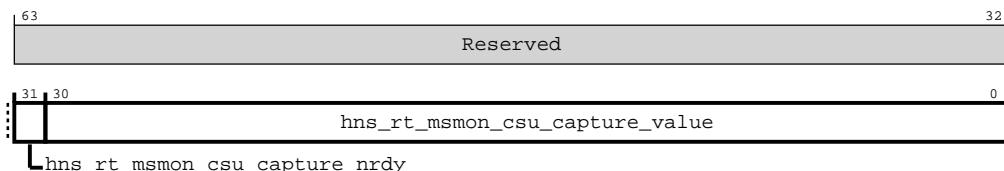
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-734: cmn\_hns\_rt\_msmon\_csu\_capture**



**Table 8-742: 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	0x0
[30:0]	hns_rt_msmon_csu_capture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	0x0

### 8.3.11.67 cmn\_hns\_rt\_msmon\_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8860

##### Type

RW

##### Reset value

See individual bit resets

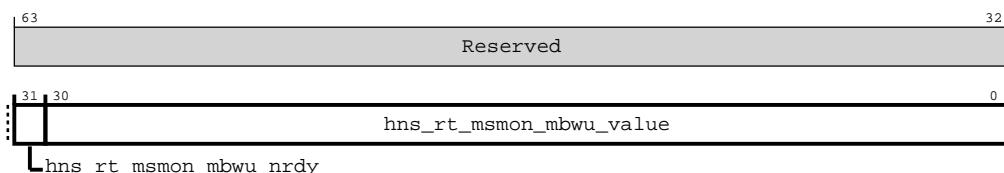
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-735: cmn\_hns\_rt\_msmon\_mbwu**



**Table 8-743: 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	0x0
[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	0x0

### 8.3.11.68 cmn\_hns\_rt\_msmon\_mbwu\_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8868

##### Type

RW

##### Reset value

See individual bit resets

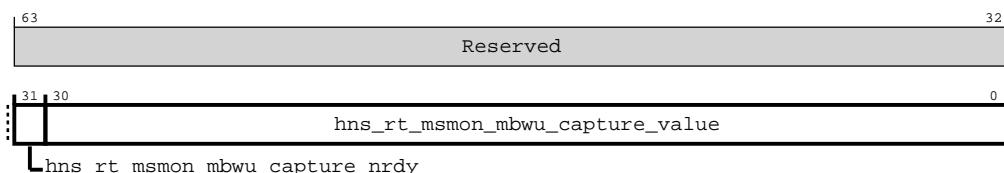
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-736: cmn\_hns\_rt\_msmon\_mbwu\_capture**



**Table 8-744: 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	0x0
[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	0x0

### 8.3.11.69 cmn\_hns\_rt\_mpamcfg\_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x9000

##### Type

RW

##### Reset value

See individual bit resets

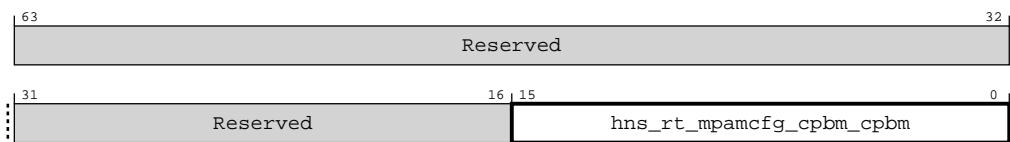
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

#### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-737: cmn\_hns\_rt\_mpamcfg\_cpbm**



**Table 8-745: cmn\_hns\_rt\_mpamcfg\_cpbm attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_rt_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL.  <b>NOTE</b> CPBM can not be all zeros for any PARTID.	RW	0xFFFF

### 8.3.12 HN-I register summary

The following table describes the registers for the relevant component.

**Table 8-746: por\_hni\_cfg register summary**

Offset	Name	Type	Description
0x0	por_hni_node_info	RO	<a href="#">por_hni_node_info</a>
0x80	por_hni_child_info	RO	<a href="#">por_hni_child_info</a>
0x980	por_hni_rcr	RW	<a href="#">por_hni_rcr</a>
0x988	por_hni_scr	RW	<a href="#">por_hni_scr</a>
0x900	por_hni_unit_info	RO	<a href="#">por_hni_unit_info</a>
0x908	por_hni_unit_info_1	RO	<a href="#">por_hni_unit_info_1</a>
0xC00	por_hni_sam_addrregion0_cfg	RW	<a href="#">por_hni_sam_addrregion0_cfg</a>
0xC08	por_hni_sam_addrregion1_cfg	RW	<a href="#">por_hni_sam_addrregion1_cfg</a>
0xC10	por_hni_sam_addrregion2_cfg	RW	<a href="#">por_hni_sam_addrregion2_cfg</a>
0xC18	por_hni_sam_addrregion3_cfg	RW	<a href="#">por_hni_sam_addrregion3_cfg</a>
0xC20	por_hnp_multi_mesh_chn_ctrl	RW	<a href="#">por_hnp_multi_mesh_chn_ctrl</a>
0xC28 : 0xCA0	por_hnp_multi_mesh_chn_sel_0-15	RW	<a href="#">por_hnp_multi_mesh_chn_sel_0-15</a>
0xA00	por_hni_cfg_ctl	RW	<a href="#">por_hni_cfg_ctl</a>
0xA08	por_hni_aux_ctl	RW	<a href="#">por_hni_aux_ctl</a>
0xA10	por_hni_datasource_ctl	RW	<a href="#">por_hni_datasource_ctl</a>
0xE000	por_hni_errfr	RO	<a href="#">por_hni_errfr</a>
0xE008	por_hni_errctlr	RW	<a href="#">por_hni_errctlr</a>
0xE010	por_hni_errstatus	W1C	<a href="#">por_hni_errstatus</a>
0xE018	por_hni_erraddr	RW	<a href="#">por_hni_erraddr</a>
0xE020	por_hni_errmisc1	RW	<a href="#">por_hni_errmisc1</a>
0xE800	por_hni_errpfgf	RO	<a href="#">por_hni_errpfgf</a>
0xE808	por_hni_errpfctl	RW	<a href="#">por_hni_errpfctl</a>
0xE810	por_hni_errpfcdn	RW	<a href="#">por_hni_errpfcdn</a>
0xE040	por_hni_errfr_NS	RO	<a href="#">por_hni_errfr_NS</a>
0xE048	por_hni_errctlr_NS	RW	<a href="#">por_hni_errctlr_NS</a>
0xE050	por_hni_errstatus_NS	W1C	<a href="#">por_hni_errstatus_NS</a>
0xE058	por_hni_erraddr_NS	RW	<a href="#">por_hni_erraddr_NS</a>
0xE068	por_hni_errmisc1_NS	RW	<a href="#">por_hni_errmisc1_NS</a>
0xE840	por_hni_errpfgf_NS	RO	<a href="#">por_hni_errpfgf_NS</a>

Offset	Name	Type	Description
0xE848	por_hni_errpfctl_NS	RW	<a href="#">por_hni_errpfctl_NS</a>
0xE850	por_hni_errpfcdn_NS	RW	<a href="#">por_hni_errpfcdn_NS</a>
0xED00	por_hni_errcapctl	RW	<a href="#">por_hni_errcapctl</a>
0xEE00	por_hni_errgsr	RO	<a href="#">por_hni_errgsr</a>
0xEE10	por_hni_errildr	RO	<a href="#">por_hni_errildr</a>
0xEFA8	por_hni_errdevaff	RO	<a href="#">por_hni_errdevaff</a>
0xEF88	por_hni_errdevarch	RO	<a href="#">por_hni_errdevarch</a>
0xEFC8	por_hni_errdevid	RO	<a href="#">por_hni_errdevid</a>
0xEF00	por_hni_errpidr45	RO	<a href="#">por_hni_errpidr45</a>
0xEFE0	por_hni_errpidr01	RO	<a href="#">por_hni_errpidr01</a>
0xEFE8	por_hni_errpidr23	RO	<a href="#">por_hni_errpidr23</a>
0EFF0	por_hni_errcidr01	RO	<a href="#">por_hni_errcidr01</a>
0EFF8	por_hni_errcidr23	RO	<a href="#">por_hni_errcidr23</a>
0xD900	por_hni_pmu_event_sel	RW	<a href="#">por_hni_pmu_event_sel</a>
0xD908	por_hnp_pmu_event_sel	RW	<a href="#">por_hnp_pmu_event_sel</a>

### 8.3.12.1 por\_hni\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

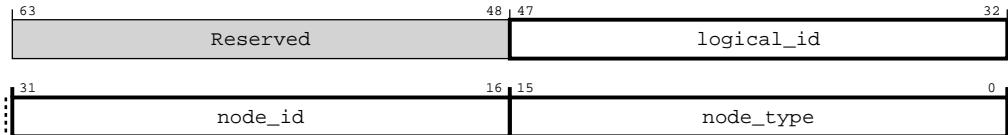
##### Usage constraints

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 8-738: por\_hni\_node\_info**



**Table 8-747: por\_hni\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	Configuration dependent

### 8.3.12.2 por\_hni\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

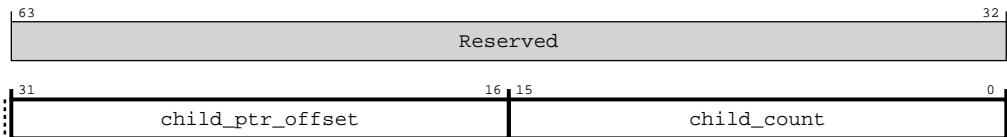
##### Usage constraints

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 8-739: por\_hni\_child\_info**



**Table 8-748: por\_hni\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.12.3 por\_hni\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

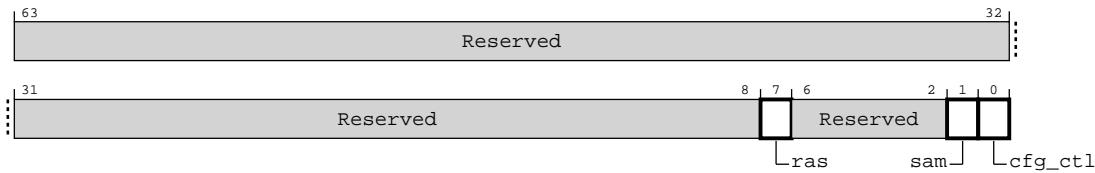
##### Usage constraints

This register is owned in the Root space and is 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 8-740: por\_hni\_rsr**



**Table 8-749: por\_hni\_rsr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allow Root override of the RAS registers	RW	0b0
[6:2]	Reserved	Reserved	RO	
[1]	sam	Allows Root override of the SAM registers	RW	0b0
[0]	cfg_ctl	Allows Root override of the configuration control registers	RW	0b0

### 8.3.12.4 por\_hni\_ssr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

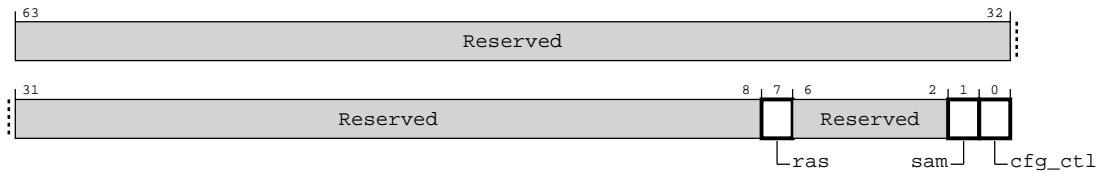
#### Usage constraints

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 8-741: por\_hni\_scr**



**Table 8-750: por\_hni\_scr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allows Secure override of the RAS registers	RW	0b0
[6:2]	Reserved	Reserved	RO	
[1]	sam	Allows Secure override of the SAM registers	RW	0b0
[0]	cfg_ctl	Allows Secure override of the configuration control registers	RW	0b0

### 8.3.12.5 por\_hni\_unit\_info

Provides component identification information for HN-I.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

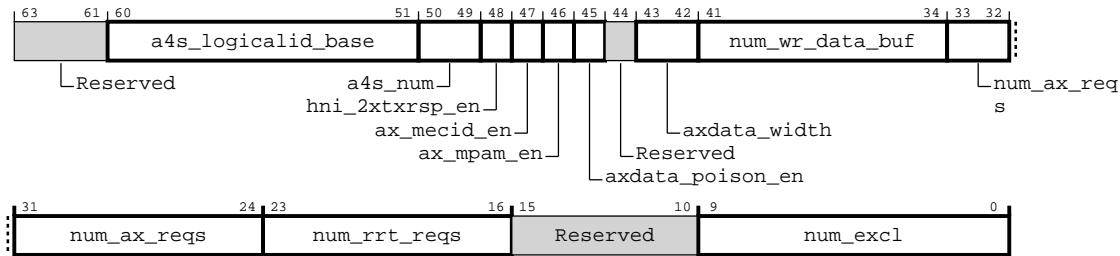
#### Usage constraints

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 8-742: por\_hni\_unit\_info**



**Table 8-751: por\_hni\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:51]	a4s_logicalid_base	AXI4Stream interfaces logical ID base	RO	0x0
[50:49]	a4s_num	Number of AXI4Stream interfaces present	RO	Configuration dependent
[48]	hnิ_2txrsp_en	Enables 2 CHI TXRSP channels. Only allowed on HN-P instances.	RO	0x0
[47]	ax_mecid_en	MECID enable on ACE-Lite/AXI4 interface  <b>0b0</b> Not enabled  <b>0b1</b> Enabled	RO	Configuration dependent
[46]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface  <b>0b0</b> Not enabled  <b>0b1</b> Enabled	RO	Configuration dependent
[45]	axdata_poison_en	Data poison support on ACE-Lite/AXI4 interface  <b>0b0</b> Not supported  <b>0b1</b> Supported	RO	Configuration dependent
[44]	Reserved	Reserved	RO	-
[43:42]	axdata_width	Data width on ACE-Lite/AXI4 interface  <b>0b00</b> 128 bits  <b>0b01</b> 256 bits  <b>0b10</b> 512 bits	RO	0x0
[41:34]	num_wr_data_buf	Number of write data buffers in HN-I	RO	Configuration dependent
[33:24]	num_ax_reqs	Maximum number of outstanding ACE-Lite/AXI4 requests	RO	Configuration dependent
[23:16]	num_rrt_reqs	Number of CHI RRT request tracker entries in HN-I.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:10]	Reserved	Reserved	RO	-
[9:0]	num_excl	Number of exclusive monitors in HN-I	RO	Configuration dependent

### 8.3.12.6 por\_hni\_unit\_info\_1

Provides component identification information for HN-I.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x908

##### Type

RO

##### Reset value

See individual bit resets

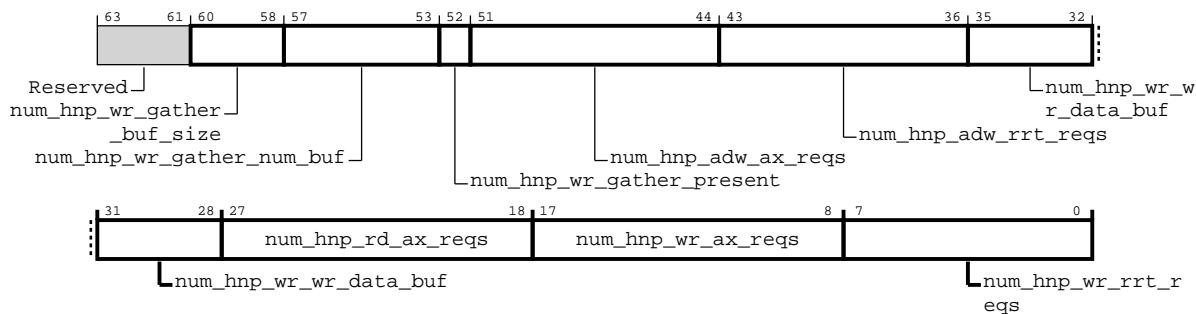
##### Usage constraints

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 8-743: por\_hni\_unit\_info\_1**



**Table 8-752: por\_hni\_unit\_info\_1 attributes**

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:58]	num_hnp_wr_gather_buf_size	Size of HN-P Write Gather buffers.  <b>0x0</b> 256 bytes  <b>0x1</b> 512 bytes	RO	Configuration dependent
[57:53]	num_hnp_wr_gather_num_buf	Number of HN-P Write Gather buffers.	RO	Configuration dependent
[52]	num_hnp_wr_gather_present	Indicates the HN-P Write Gather buffer is present.	RO	Configuration dependent
[51:44]	num_hnp_adw_ax_reqs	Maximum number of outstanding Atomic/Deferrable Write ACE-Lite/AXI4 requests. HN-P only.	RO	0x20
[43:36]	num_hnp_adw_rrt_reqs	Number of Atomic/Deferrable Write CHI RRT request tracker entries. HN-P only.	RO	Configuration dependent
[35:28]	num_hnp_wr_wr_data_buf	Number of P2P write data buffers in HN-I. HN-P only.	RO	Configuration dependent
[27:18]	num_hnp_rd_ax_reqs	Maximum number of outstanding P2P Read ACE-Lite/AXI4 requests. HN-P only.	RO	Configuration dependent
[17:8]	num_hnp_wr_ax_reqs	Maximum number of outstanding P2P Write ACE-Lite/AXI4 requests. HN-P only.	RO	Configuration dependent
[7:0]	num_hnp_wr_rrt_reqs	Number of P2P Write CHI RRT request tracker entries. HN-P only.	RO	Configuration dependent

### 8.3.12.7 por\_hni\_sam\_addrregion0\_cfg

Configures Address Region 0.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.sam

## Secure group override

por\_hni\_scr.sam

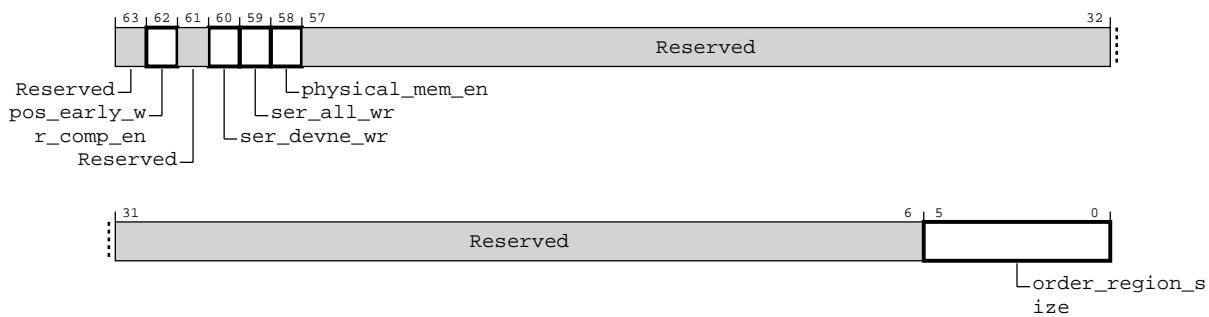
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.sam bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.sam bit and por\_hni\_rcr.sam 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 8-744: por\_hni\_sam\_addrregion0\_cfg**



**Table 8-753: por\_hni\_sam\_addrregion0\_cfg attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 0; used to improve write performance	RW	0b1
[61]	Reserved	Reserved	RO	-
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 0	RW	0b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 0	RW	0b0
[58]	physical_mem_en	Address Region 0 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	0b0
[57:6]	Reserved	Reserved	RO	-
[5:0]	order_region_size	<n>; used to calculate Order Region 0 size within Address Region 0 ( $2^n \times 4KB$ )	RW	0b111111

## 8.3.12.8 por\_hni\_sam\_addrregion1\_cfg

Configures Address Region 1.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xC08

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_hni\_rcr.sam

### Secure group override

por\_hni\_scr.sam

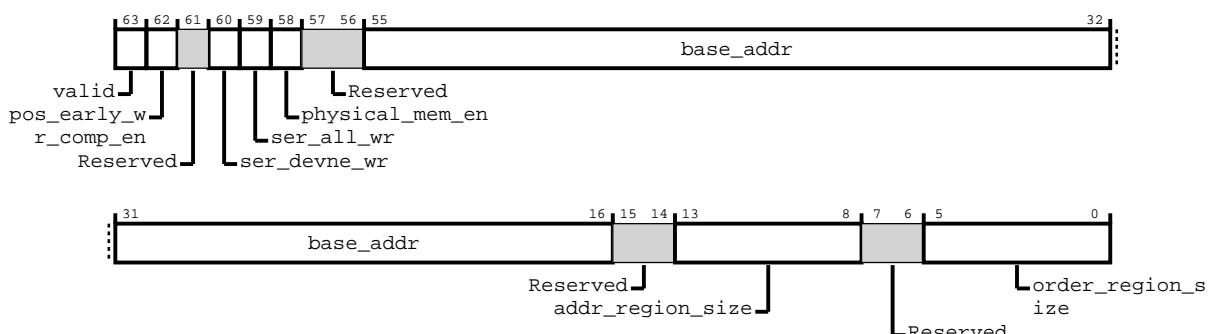
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.sam bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.sam bit and por\_hni\_rcr.sam 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 8-745: por\_hni\_sam\_addrregion1\_cfg**



**Table 8-754: por\_hni\_sam\_addrregion1\_cfg attributes**

Bits	Name	Description	Type	Reset
[63]	valid	Address Region 1 fields are programmed and valid	RW	0x0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 1; used to improve write performance	RW	0b1
[61]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 1	RW	0b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 1	RW	0b0
[58]	physical_mem_en	Address Region 1 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	0b0
[57:56]	Reserved	Reserved	RO	
[55:16]	base_addr	Address Region 1 base address; [address width-1:12] <b>CONSTRAINT</b> Must be an integer multiple of the Address Region 1 size.	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	addr_region_size	<n>; used to calculate Address Region 1 size ( $2^n \times 4KB$ ) <b>CONSTRAINT</b> <n> must be configured so that the Address Region 1 size is less than or equal to $2^{(address\ width)}$ .	RW	0x0
[7:6]	Reserved	Reserved	RO	
[5:0]	order_region_size	<n>; used to calculate Order Region 1 size within Address Region 1 ( $2^n \times 4KB$ )	RW	0x0

### 8.3.12.9 por\_hni\_sam\_addrregion2\_cfg

Configures Address Region 2.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC10

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.sam

##### Secure group override

por\_hni\_scr.sam

#### Usage constraints

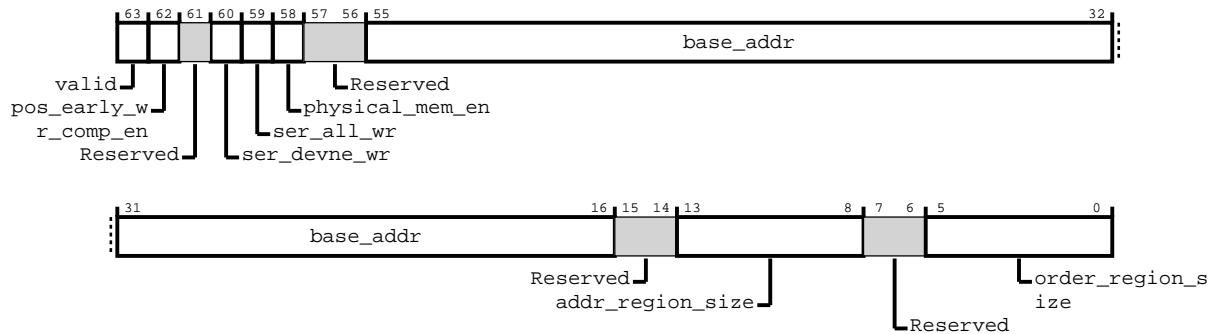
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.sam bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.sam bit and por\_hni\_rcr.sam bit are set, Non-

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 8-746: por\_hni\_sam\_addrregion2\_cfg**



**Table 8-755: por\_hni\_sam\_addrregion2\_cfg attributes**

Bits	Name	Description	Type	Reset
[63]	valid	Address Region 2 fields are programmed and valid	RW	0x0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 2; used to improve write performance	RW	0b1
[61]	Reserved	Reserved	RO	
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 2	RW	0b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 2	RW	0b0
[58]	physical_mem_en	Address Region 2 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	0b0
[57:56]	Reserved	Reserved	RO	
[55:16]	base_addr	Address Region 2 base address; [address width-1:12] <b>CONSTRAINT</b> Must be an integer multiple of the Address Region 2 size	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	addr_region_size	<n>; used to calculate Address Region 2 size ( $2^n \times 4KB$ ) <b>CONSTRAINT</b> <n> must be configured so that the Address Region 2 size is less than or equal to $2^{(address\ width)}$ .	RW	0x0
[7:6]	Reserved	Reserved	RO	
[5:0]	order_region_size	<n>; used to calculate Order Region 2 size within Address Region 2 ( $2^n \times 4KB$ )	RW	0x0

### 8.3.12.10 por\_hni\_sam\_addrregion3\_cfg

Configures Address Region 3.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xC18

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.sam

##### Secure group override

por\_hni\_scr.sam

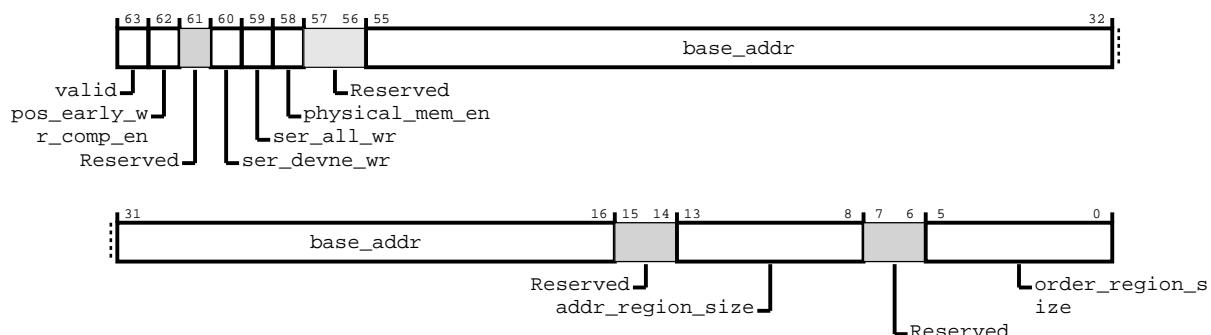
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.sam bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.sam bit and por\_hni\_rcr.sam 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 8-747: por\_hni\_sam\_addrregion3\_cfg**



**Table 8-756: por\_hni\_sam\_addrregion3\_cfg attributes**

Bits	Name	Description	Type	Reset
[63]	valid	Fields of Address Region 3 are programmed and valid	RW	0x0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 3; used to improve write performance	RW	0b1
[61]	Reserved	Reserved	RO	
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 3	RW	0b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 3	RW	0b0
[58]	physical_mem_en	Address Region 3 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	0b0
[57:56]	Reserved	Reserved	RO	
[55:16]	base_addr	Address Region 3 base address; [address width-1:12] <b>CONSTRAINT</b> Must be an integer multiple of the Address Region 3 size	RW	0x0
[15:14]	Reserved	Reserved	RO	
[13:8]	addr_region_size	<n>; used to calculate Address Region 3 size ( $2^n \times 4KB$ ) <b>CONSTRAINT</b> <n> must be configured so that the Address Region 3 size is less than or equal to $2^{(address\ width)}$ .	RW	0x0
[7:6]	Reserved	Reserved	RO	
[5:0]	order_region_size	<n>; used to calculate Order Region 3 size within Address Region 3 ( $2^n \times 4KB$ )	RW	0x0

### 8.3.12.11 por\_hnp\_multi\_mesh\_chn\_ctrl

Functions as the control register for Target based channel selection in Multi-Mesh Channel structure.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC20

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.cfg\_ctl

## Secure group override

por\_hni\_scr.cfg\_ctl

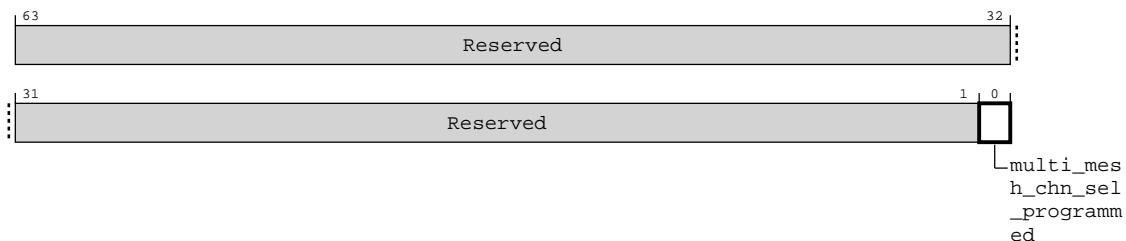
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.cfg\_ctl bit and por\_hni\_rcr.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 8-748: por\_hnp\_multi\_mesh\_chn\_ctrl**



**Table 8-757: por\_hnp\_multi\_mesh\_chn\_ctrl attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	multi_mesh_chn_sel_programmed	Indicates that multi CHI VC channel configured for all the targets specified in the channel select registers.	RW	0b0

### 8.3.12.12 por\_hnp\_multi\_mesh\_chn\_sel\_0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Functions as the CHI VC channel select per Target register in Multi-Mesh Channel structure.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xC28 + #8\*index}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_hni\_rcr.cfg\_ctl

## Secure group override

por\_hni\_scr.cfg\_ctl

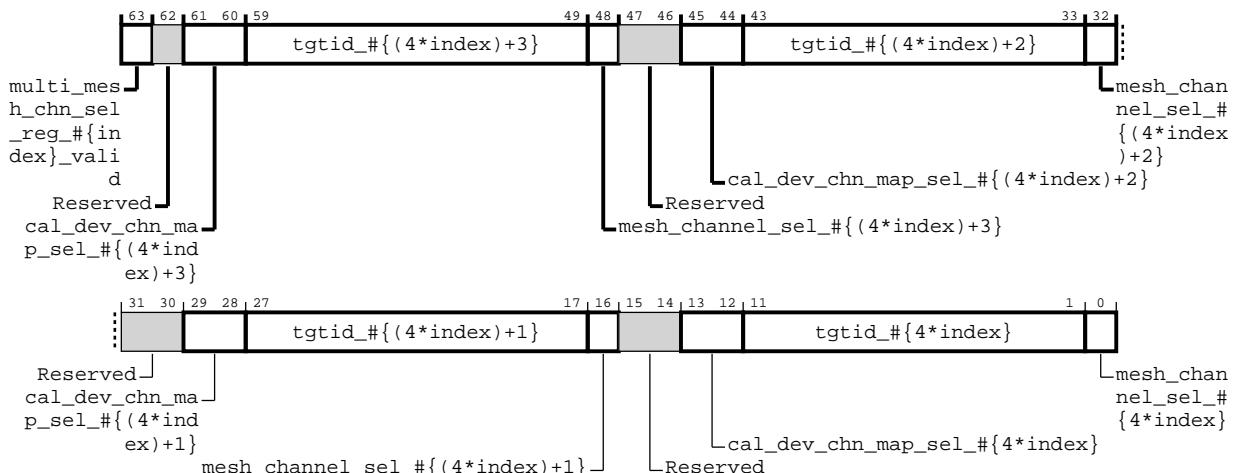
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.cfg\_ctl bit and por\_hni\_rcr.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 8-749: por\_hnp\_multi\_mesh\_chn\_sel\_0-15**



**Table 8-758: por\_hnp\_multi\_mesh\_chn\_sel\_0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	multi_mesh_chn_sel_reg_{index}_valid	Indicates that multi mesh CHI VC channel configured for the targets specified in this register.	RW	0b0
[62]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[61:60]	cal_dev_chn_map_sel_{(4*index)+3}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field)</p> <p><b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b> Reserved,</p> <p><b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[59:49]	tgtid_{(4*index)+3}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[48]	mesh_channel_sel_{(4*index)+3}	<p>CHI VC channel select:</p> <p>1 - CHI VC channel 1 is selected</p> <p>0 - CHI VC channel 0 is selected</p>	RW	0b0
[47:46]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[45:44]	cal_dev_chn_map_sel_{(4*index)+2}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field)</p> <p><b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b> Reserved,</p> <p><b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[43:33]	tgtid_{(4*index)+2}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[32]	mesh_channel_sel_{(4*index)+2}	<p>CHI VC channel select</p> <p>1 - CHI VC channel 1 is selected</p> <p>0 - CHI VC channel 0 is selected</p>	RW	0b0
[31:30]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[29:28]	cal_dev_chn_map_sel_#{(4*index)+1}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field)</p> <p><b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b> Reserved,</p> <p><b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[27:17]	tgtid_#{(4*index)+1}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[16]	mesh_channel_sel_#{(4*index)+1}	<p>CHI VC channel select</p> <p>1 - CHI VC channel 1 is selected</p> <p>0 - CHI VC channel 0 is selected</p>	RW	0b0
[15:14]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[13:12]	cal_dev_chn_map_sel_{4*index}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field)</p> <p><b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b> Reserved,</p> <p><b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[11:1]	tgtid_{4*index}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[0]	mesh_channel_sel_{4*index}	<p>CHI VC channel select</p> <p>1 - CHI VC channel 1 is selected</p> <p>0 - CHI VC channel 0 is selected</p>	RW	0b0

### 8.3.12.13 por\_hni\_cfg\_ctl

Functions as the configuration control register for HN-I.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_hni\_rcr.cfg\_ctl

### Secure group override

por\_hni\_scr.cfg\_ctl

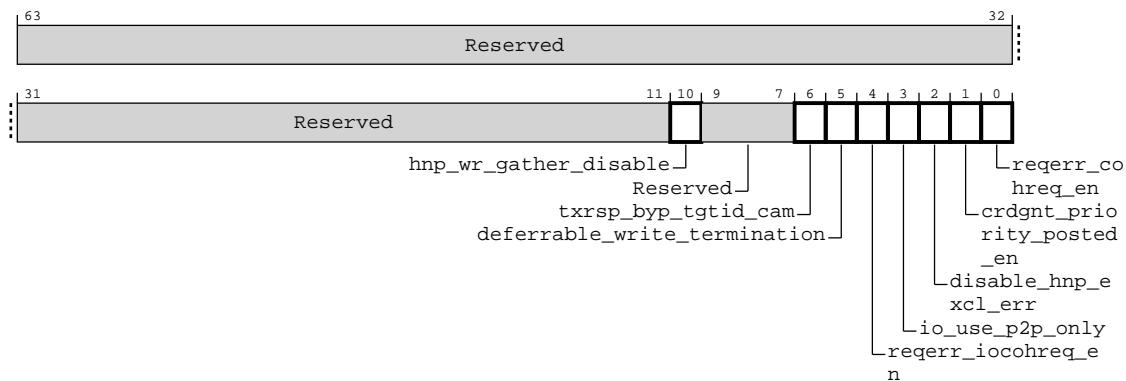
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.cfg\_ctl bit and por\_hni\_rcr.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 8-750: por\_hni\_cfg\_ctl**



**Table 8-759: por\_hni\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	hnp_wr_gather_disable	Disable the write gathering feature.	RW	0x0
[9:7]	Reserved	Reserved	RO	
[6]	txrsp_byp_tgtid_cam	Bypass the Tgtid CAM for finding the tx rsp port to be used.	RW	0x0
[5]	deferrable_write_termination	Enables termination of CHI Deferrable Writes when the ACE Lite/AXI Subordinate does not support receiving Deferrable Writes.	RW	0b0

Bits	Name	Description	Type	Reset
[4]	reqerr_iocohreq_en	<p>Enables sending of NDE response error and logging of RAS error information for the following I/O coherent request types</p> <ol style="list-style-type: none"> <li>1. I/O Coherent ReadOnce (excluding ReadOnceMakeInvalid and ReadOnceCleanInvalid - these are covered by the reqerr_cohreq_en bit)</li> <li>2. I/O Coherent WriteUnique*</li> </ol> <p>This bit only applies to P2P requests from an RNI/RND. All other coherent request types are covered by the reqerr_cohreq_en bit.</p> <p><b>Note</b></p> <p>if reqerr_cohreq_en is disabled, this bit has no effect and all coherent request types will be accepted without error.</p>	RW	0b0
[3]	io_use_p2p_only	<p>All I/O traffic (not just PCIe) from RNI/RND uses only the P2P slices. CPU traffic continues to use the NP2P slice. When enabled, the following apply</p> <ul style="list-style-type: none"> <li>• No support for DWT</li> <li>• No external write combining supported</li> <li>• No P2P write burst support</li> <li>• HNP is not the PoS, so CHI COMP will only be issued after receiving AXI BRESP</li> <li>• No transaction ordering among reads, no ordering among writes, no ordering across the two, no OWO/ExpCompAck support</li> <li>• Source-based ordering is required (ie, source must wait for COMP of previous request before issuing dependent request)</li> <li>• ExpCompAck read requests will continue to route to the Non-P2P slice for handling</li> <li>• All AWID and ARID will be unique</li> </ul>	RW	0b0
[2]	disable_hnp_excl_err	Disables sending NDE and Error logging on ReadNoSnp and WriteNoSnp Exclusives	RW	0b0
[1]	crdgnt_priority_posted_en	Enables High priority Credit Grant responses to Posted requests	RW	0b0
[0]	reqerr_cohreq_en	<p>Enables sending of NDE response error and logging of RAS error information for the following coherent request types</p> <ol style="list-style-type: none"> <li>1. Fully Coherent Read (excluding I/O coherent ReadOnce from RNI/D)</li> <li>2. Coherent Dataless (CleanUnique/MakeUnique/Evict/StashOnce/CMO)</li> <li>3. Fully Coherent/CopyBack Write (excluding I/O coherent WriteUnique* from RNI/D)</li> </ol> <p>The P2P I/O coherent types from RNI/D are covered by the reqerr_iocohreq_en bit. Atomic* requests will always generate an NDE response error and logging of RAS error information.</p>	RW	0b1

### 8.3.12.14 por\_hni\_aux\_ctl

Functions as the auxiliary control register for HN-I.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA08

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_hni\_rcr.cfg\_ctl

### Secure group override

por\_hni\_scr.cfg\_ctl

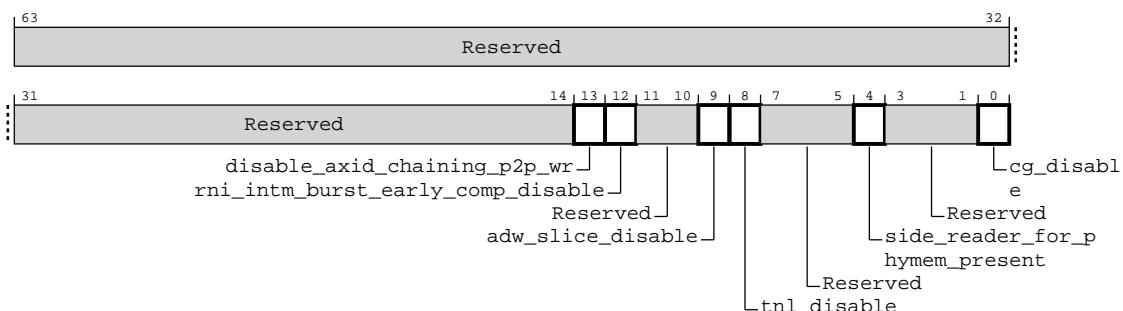
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.cfg\_ctl bit and por\_hni\_rcr.cfg\_ctl bit are set, Non-secure and Realm accesses to this register are permitted. This register can be modified only with prior written permission from Arm.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-751: por\_hni\_aux\_ctl**



**Table 8-760: por\_hni\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13]	disable_axid_chaining_p2p_wr	Disables AXID based chaining of PCIe writes in P2P Write slice. HNP only	RW	0b0
[12]	rni_intm_burst_early_comp_disable	Disables Early COMP to RNI for non-last burst writes	RW	0b0
[11:10]	Reserved	Reserved	RO	-
[9]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[8]	tnl_disable	Disables RNI-HNI Tunneling in HNI. por_rni_aux_ctl.dis_hni_wr_stream must be set before setting this bit	RW	0b0
[7:5]	Reserved	Reserved	RO	-
[4]	side_reader_for_phymem_present	Enables side reader in physical memory range	RW	0b0
[3:1]	Reserved	Reserved	RO	-
[0]	cg_disable	Disables HN-I architectural clock gates	RW	0b0

### 8.3.12.15 por\_hni\_datasource\_ctl

Functions as the DataSource control register to determine how to drive the CHI DAT.DataSource field.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA10

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.cfg\_ctl

##### Secure group override

por\_hni\_scr.cfg\_ctl

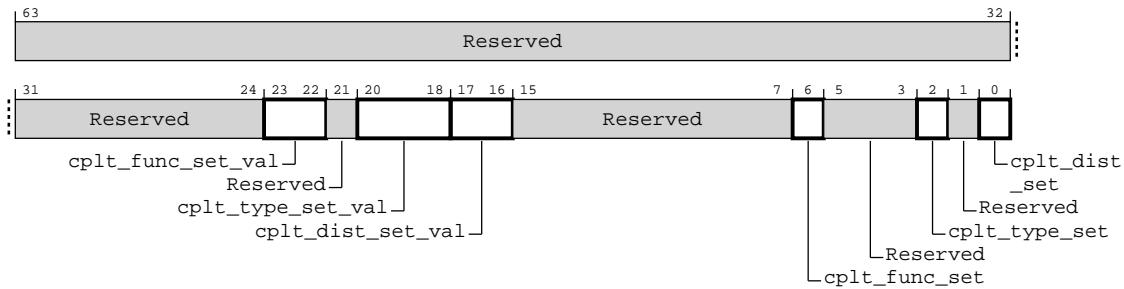
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.cfg\_ctl bit and por\_hni\_rcr.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 8-752: por\_hni\_datasource\_ctl**



**Table 8-761: por\_hni\_datasource\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:22]	cplt_func_set_val	Completer Functional value to use when cplt_func_set is set to 1. Supported values are:  0b00 - Default/PrefetchTgt not useful, 0b01 - PrefetchTgt useful Prefetch useful 0b01 is only supported when cplt_type_set_val is non-zero.	RW	0b00
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val	Completer Type value to use when cplt_type_set is set to 1. Supported values are:  0b000 - Default Completer Type, 0b001 - DRAM Completer Type, 0b011 - High Bandwidth Memory (HBM) Completer Type	RW	0b000
[17:16]	cplt_dist_set_val	Completer Distance value to use when cplt_dist_set is set to 1.	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	cplt_func_set	Completer Functional Set 0b0 - Reserved, 0b1 - Set Completer Functional field to cplt_func_set_val when sending CompData	RW	0b1
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	Completer Type Set 0b0 - Reserved, 0b1 - Set Completer Type field to cplt_type_set_val when sending CompData	RW	0b1
[1]	Reserved	Reserved	RO	-
[0]	cplt_dist_set	Completer Distance Set 0b0 - Reserved, 0b1 - Set Completer Distance field to cplt_dist_set_val when sending CompData	RW	0b1

### 8.3.12.16 por\_hni\_errfr

Error Feature Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xE000

## Type

RO

## Reset value

See individual bit resets

## Root group override

por\_hni\_rcr.ras

## Secure group override

por\_hni\_scr.ras

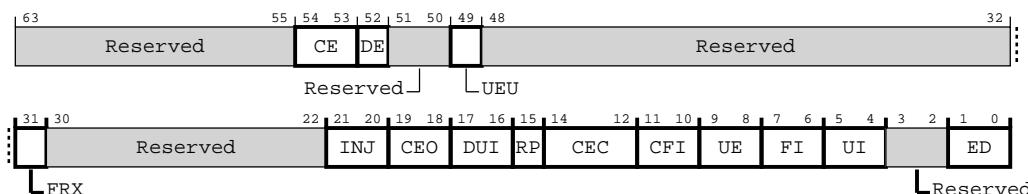
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-753: por\_hni\_errfr**



**Table 8-762: por\_hni\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording <b>0b00</b> Corrected Error not supported	RO	0b00
[52]	DE	Deferred Error recording <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	FRX	Feature Register Extension <b>0b1</b> por_hni_errfr[63:48] are defined by the architecture	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension <b>0b01</b> The node implements the RAS Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error Overwrite <b>0b00</b> Reserved, <b>RES0</b>	RO	0b00
[17:16]	DUI	Error Recovery Interrupt for Deferred Errors control <b>0b10</b> Control for enabling error recovery interrupts on deferred errors is supported and controlled using por_hni_errctlr.DUI	RO	0b10
[15]	RP	Repeat Corrected Error Counter <b>0b0</b> Reserved, <b>RES0</b>	RO	0b0
[14:12]	CEC	Corrected Error Counter <b>0b000</b> Does not implement the standard Corrected error counter model	RO	0b000
[11:10]	CFI	Fault Handling Interrupt for Corrected Errors control <b>0b00</b> Does not support the control for enabling fault handling interrupts on corrected errors	RO	0b00
[9:8]	UE	In-band error response <b>0b01</b> In-band error response (External Abort) is supported and always enabled	RO	0b01
[7:6]	FI	Fault Handling Interrupt <b>0b10</b> Fault handling interrupt is supported and controllable using por_hni_errctlr.FI	RO	0b10
[5:4]	UI	Error Recovery Interrupt for Uncorrected Errors <b>0b10</b> Error handling interrupt is supported and controllable using por_hni_errctlr.UI	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging <b>0b10</b> Error reporting and logging is controllable using por_hni_errctlr.ED	RO	0b10

### 8.3.12.17 por\_hni\_errctlr

Error Control Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE008

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

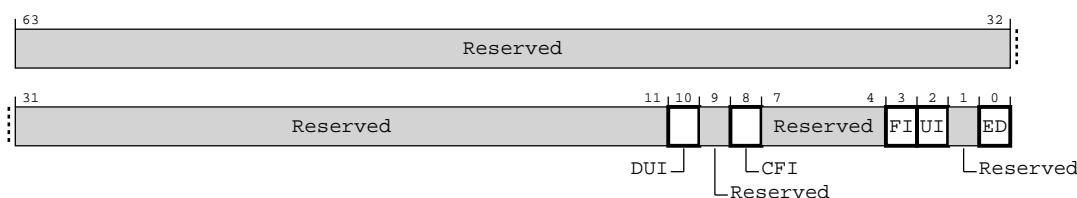
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-754: por\_hni\_errctlr**



**Table 8-763: por\_hni\_errctlr attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Error recovery interrupt for Deferred Errors enable	RW	0b0

Bits	Name	Description	Type	Reset
[9]	Reserved	Reserved	RO	
[8]	CFI	Fault handling interrupt for Corrected Errors enable	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Fault handling interrupt enable	RW	0b0
[2]	UI	Uncorrected error recovery interrupt enable	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Error reporting and logging enable	RW	0b0

### 8.3.12.18 por\_hni\_errstatus

Error Status Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE010

##### Type

W1C

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

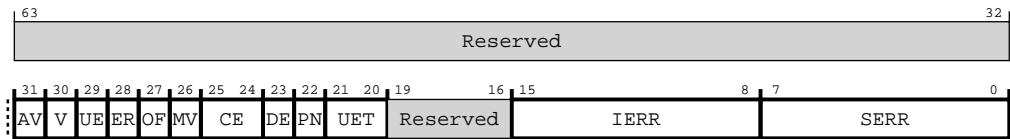
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-755: por\_hni\_errstatus**



**Table 8-764: por\_hni\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address Valid <b>0b0</b> por_hni_erraddr not valid <b>0b1</b> por_hni_erraddr contains an address associated with the highest priority error recorded by this node	W1C	0b0
[30]	V	Status Register Valid <b>0b0</b> por_hni_errstatus not valid <b>0b1</b> por_hni_errstatus valid. At least one error has been recorded	W1C	0b0
[29]	UE	Uncorrected Error <b>0b0</b> No uncorrected errors detected <b>0b1</b> At least one detected error was not corrected and not deferred	W1C	0b0
[28]	ER	Error Reported <b>0b0</b> No in-band error response signaled <b>0b1</b> In-band error response was signaled to the Requester	W1C	0b0
[27]	OF	Overflow <b>0b0</b> Since last cleared to zero, no error syndrome has been discarded <b>0b1</b> Since last cleared to zero, at least one error syndrome has been discarded	W1C	0b0
[26]	MV	Miscellaneous Register Valid <b>0b0</b> por_hni_errmisc register not valid <b>0b1</b> por_hni_errmisc register contains additional information for an error recorded by this node	W1C	0b0

Bits	Name	Description	Type	Reset
[25:24]	CE	Corrected Error  <b>0b00</b> No errors were corrected	W1C	0b0
[23]	DE	Deferred Error  <b>0b0</b> No errors were deferred  <b>0b1</b> At least one error was not corrected and is deferred	W1C	0b0
[22]	PN	Poison  <b>0b0</b> Uncorrected or Deferred error recorded because a corrupt value was detected  <b>0b1</b> Uncorrected or Deferred error recorded because a poison value was detected	W1C	0b0
[21:20]	UET	Uncorrected Error Type  <b>0b00</b> Invalid  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code  <b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_hni_errmisc1.ERRSRC for error type details.	W1C	0b0

### 8.3.12.19 por\_hni\_erraddr

Error Address Register for Root NOTE: BRESP and poison errors (0x8-0xA in por\_hni\_errmisc1.ERRSRC) do not record anything in por\_hni\_erraddr.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xE018

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_hni\_rcr.ras

## Secure group override

por\_hni\_scr.ras

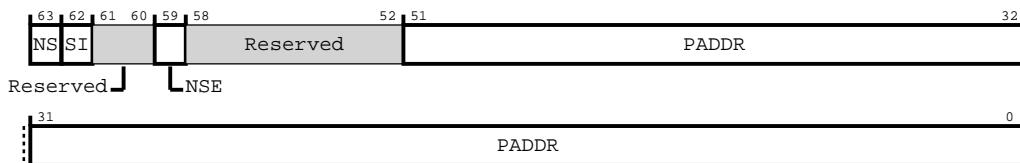
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-756: por\_hni\_erraddr**



**Table 8-765: por\_hni\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Non-Secure attribute (NS, also PAS[0])	RW	0b0
[62]	SI	PAS Incorrect  0b0  PAS[1:0] is correct  0b1  PAS[1:0] might not be correct	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Non-Secure Extension attribute (NSE, also PAS[1])	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	PADDR	Physical Address	RW	0b0

### 8.3.12.20 por\_hni\_errmisc1

Error Miscellaneous Register 1

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE020

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

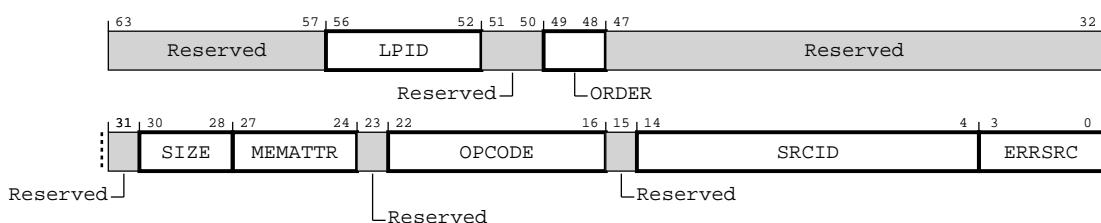
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-757: por\_hni\_errmisc1**



**Table 8-766: por\_hni\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	
[56:52]	LPID	CHI Logic Processor ID (LPID) field	RW	0b0

Bits	Name	Description	Type	Reset
[51:50]	Reserved	Reserved	RO	
[49:48]	ORDER	CHI Order field	RW	0b0
[47:31]	Reserved	Reserved	RO	
[30:28]	SIZE	CHI Size field	RW	0b0
[27:24]	MEMATTR	CHI Memory Attributes field	RW	0b0
[23]	Reserved	Reserved	RO	
[22:16]	OPCODE	CHI Opcode field	RW	0b0
[15]	Reserved	Reserved	RO	
[14:4]	SRCID	CHI Source ID field	RW	0b0
[3:0]	ERRSRC	Error source <b>0b0000</b> Coherent read <b>0b0001</b> Coherent write <b>0b0010</b> Coherent dataless (CleanUnique/MakeUnique/Evict/StashOnce/CMO) <b>0b0011</b> Atomic <b>0b0100</b> Illegal configuration read <b>0b0101</b> Illegal configuration write or dataless <b>0b0110</b> Configuration write data partial byte enable error <b>0b0111</b> Configuration write data parity error or poison error <b>0b1000</b> BRESP error <b>0b1001</b> Poison error <b>0b1010</b> BRESP error and poison error <b>0b1011</b> Unsupported Exclusive access (HN-P only) <b>NOTE</b> BRESP and poison errors (0x8-0xA) are Uncorrectable Errors (UE). SRCID and ERRSRC are the only valid fields in por_hni_errmisc1. All other error types are Deferred Errors (DE) and all fields are valid.	RW	0b0

### 8.3.12.21 por\_hni\_errpfgf

Pseudo-fault Generation Feature Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE800

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

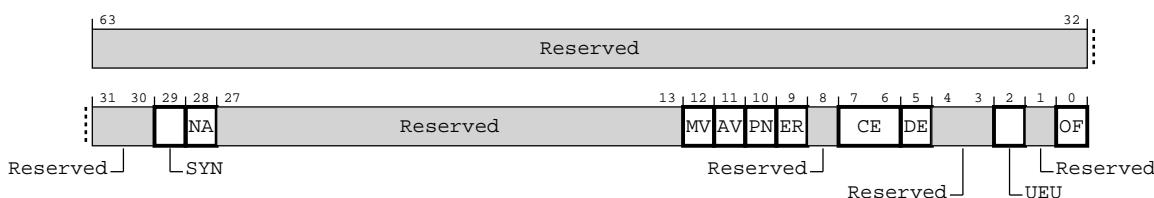
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-758: por\_hni\_errpfgf**



**Table 8-767: por\_hni\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[29]	SYN	Fault syndrome injection  <b>0b1</b> Fault injection does not update por_hni_errstatus.SERR	RO	0b1
[28]	NA	No access required  <b>0b1</b> The component fakes detection of the error spontaneously in the fault injection state	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome  <b>0b1</b> The node does not update any fields in por_hni_errmisc1_NS and sets por_hni_errstatus.MV to por_hni_errpfctl.MV	RO	0b1
[11]	AV	Address syndrome  <b>0b1</b> The node does not update por_hni_erraddr_NS and sets por_hni_errstatus.AV to por_hni_errpfctl.AV	RO	0b1
[10]	PN	Poison flag  <b>0b1</b> When an injected error is recorded, por_hni_errstatus.PN is set to por_hni_errpfctl.PN	RO	0b1
[9]	ER	Error reported flag  <b>0b1</b> When an injected error is recorded, por_hni_errstatus.ER is set to por_hni_errpfctl.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation  <b>0b00</b> The fault generation feature of the node cannot generate this type of error	RO	0b00
[5]	DE	Deferred Error generation  <b>0b1</b> The fault generation feature of the node allows generation of this type of error	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Unrecoverable Error generation  <b>0b1</b> The fault generation feature of the node allows generation of this type of error	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag  <b>0b1</b> When an injected error is recorded, por_hni_errstatus.OF is set to por_hni_errpfctl.OF	RO	0b1

### 8.3.12.22 por\_hni\_errpfctl

Pseudo-fault Generation Control Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE808

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

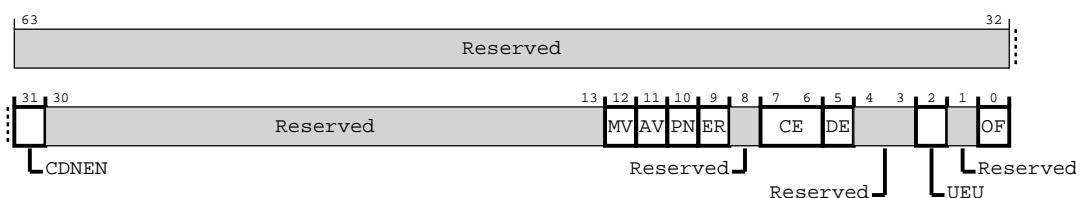
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-759: por\_hni\_errpfctl**



**Table 8-768: por\_hni\_errpfctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	CDNEN	<p>Countdown Enable</p> <p><b>0b0</b> The Error Generation Counter is disabled</p> <p><b>0b1</b> The Error Generation Counter is enabled</p>	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	<p>Miscellaneous syndrome</p> <p><b>0b0</b> por_hni_errstatus.MV is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus.MV is set to 0b1 when an injected error is recorded</p>	RW	0b0
[11]	AV	<p>Address syndrome</p> <p><b>0b0</b> por_hni_ERRSTATUS.AV is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_ERRSTATUS.AV is set to 0b1 when an injected error is recorded</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> por_hni_errstatus.PN is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus.PN is set to 0b1 when an injected error is recorded</p>	RW	0b0
[9]	ER	<p>Error Reported flag</p> <p><b>0b0</b> por_hni_errstatus.ER is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus.ER is set to 0b1 when an injected error is recorded</p>	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	<p>Corrected Error generation enable</p> <p><b>0b00</b> No error of this type will be generated</p>	RW	0b00
[5]	DE	<p>Deferred Error generation enable</p> <p><b>0b0</b> No error of this type will be generated</p> <p><b>0b1</b> An error of this type might be generated when the Error Generation Counter decrements to zero</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Unrecoverable Error generation enable</p> <p><b>0b0</b> No error of this type will be generated</p> <p><b>0b1</b> An error of this type might be generated when the Error Generation Counter decrements to zero</p>	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	OF	<p>Overflow flag</p> <p><b>0b0</b> por_hni_errstatus.OF is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus.OF is set to 0b1 when an injected error is recorded</p>	RW	0b0

### 8.3.12.23 por\_hni\_errfgcdn

Pseudo-fault Generation Countdown Register for Root

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE810

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_hni\_rcr.ras

##### Secure group override

por\_hni\_scr.ras

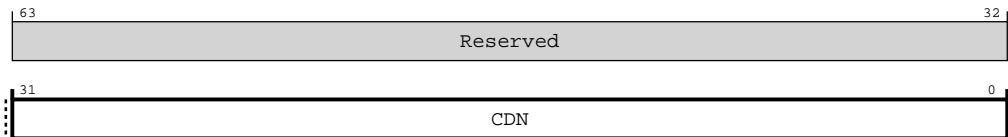
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-760: por\_hni\_errpfgcdn**



**Table 8-769: por\_hni\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.12.24 por\_hni\_errfr\_NS

Error Feature Register for Non-secure

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE040

##### Type

RO

##### Reset value

See individual bit resets

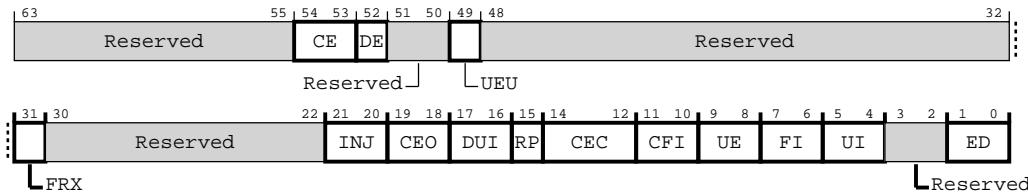
##### Usage constraints

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 8-761: por\_hni\_errfr\_NS**



**Table 8-770: por\_hni\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording <b>0b00</b> Corrected Error not supported	RO	0b00
[52]	DE	Deferred Error recording <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension <b>0b1</b> por_hni_errfr_NS[63:48] are defined by the architecture	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension <b>0b01</b> The node implements the RAS Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error Overwrite <b>0b00</b> Reserved, <b>RES0</b>	RO	0b00
[17:16]	DUI	Error Recovery Interrupt for Deferred Errors control <b>0b10</b> Control for enabling error recovery interrupts on deferred errors is supported and controlled using por_hni_errctrlr.DUI	RO	0b10
[15]	RP	Repeat Corrected Error Counter <b>0b0</b> Reserved, <b>RES0</b>	RO	0b0
[14:12]	CEC	Corrected Error Counter <b>0b000</b> Does not implement the standard Corrected error counter model	RO	0b000

Bits	Name	Description	Type	Reset
[11:10]	CFI	Fault Handling Interrupt for Corrected Errors control  <b>0b00</b> Does not support the control for enabling fault handling interrupts on corrected errors	RO	0b00
[9:8]	UE	In-band error response  <b>0b01</b> In-band error response (External Abort) is supported and always enabled	RO	0b01
[7:6]	FI	Fault Handling Interrupt  <b>0b10</b> Fault handling interrupt is supported and controllable using por_hni_errctlr_NS.FI	RO	0b10
[5:4]	UI	Error Recovery Interrupt for Uncorrected Errors  <b>0b10</b> Error handling interrupt is supported and controllable using por_hni_errctlr_NS.UI	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging  <b>0b10</b> Error reporting and logging is controllable using por_hni_errctlr_NS.ED	RO	0b10

### 8.3.12.25 por\_hni\_errctlr\_NS

Error Control Register for Non-secure

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE048

##### Type

RW

##### Reset value

See individual bit resets

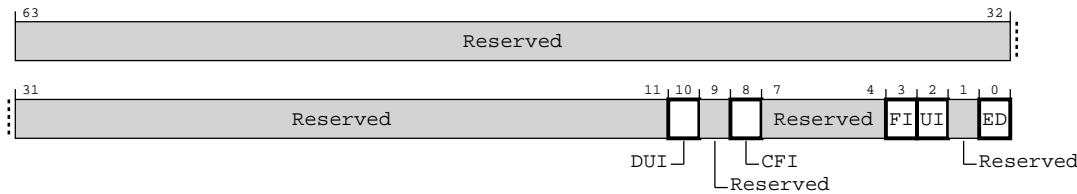
##### Usage constraints

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 8-762: por\_hni\_errctlr\_NS**



**Table 8-771: por\_hni\_errctlr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Error recovery interrupt for Deferred Errors enable	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Fault handling interrupt for Corrected Errors enable	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Fault handling interrupt enable	RW	0b0
[2]	UI	Uncorrected error recovery interrupt enable	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Error reporting and logging enable	RW	0b0

### 8.3.12.26 por\_hni\_errstatus\_NS

Error Status Register for Non-secure

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

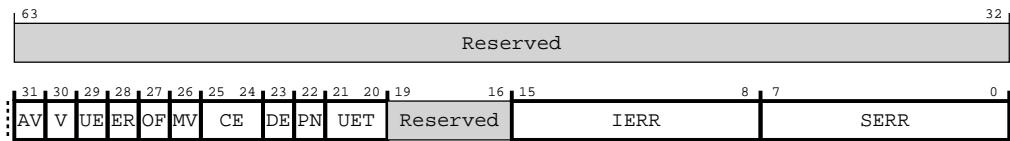
##### Usage constraints

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 8-763: por\_hni\_errstatus\_NS**



**Table 8-772: por\_hni\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address Valid <b>0b0</b> por_hni_erraddr_NS not valid <b>0b1</b> por_hni_erraddr_NS contains an address associated with the highest priority error recorded by this node	W1C	0b0
[30]	V	Status Register Valid <b>0b0</b> por_hni_errstatus_NS not valid <b>0b1</b> por_hni_errstatus_NS valid. At least one error has been recorded	W1C	0b0
[29]	UE	Uncorrected Error <b>0b0</b> No uncorrected errors detected <b>0b1</b> At least one detected error was not corrected and not deferred	W1C	0b0
[28]	ER	Error Reported <b>0b0</b> No in-band error response signaled <b>0b1</b> In-band error response was signaled to the Requester	W1C	0b0
[27]	OF	Overflow <b>0b0</b> Since last cleared to zero, no error syndrome has been discarded <b>0b1</b> Since last cleared to zero, at least one error syndrome has been discarded	W1C	0b0

Bits	Name	Description	Type	Reset
[26]	MV	Miscellaneous Register Valid  <b>0b0</b> por_hni_errmisc_NS register not valid  <b>0b1</b> por_hni_errmisc_NS register contains additional information for an error recorded by this node	W1C	0b0
[25:24]	CE	Corrected Error  <b>0b00</b> No errors were corrected	W1C	0b0
[23]	DE	Deferred Error  <b>0b0</b> No errors were deferred  <b>0b1</b> At least one error was not corrected and is deferred	W1C	0b0
[22]	PN	Poison  <b>0b0</b> Uncorrected or Deferred error recorded because a corrupt value was detected  <b>0b1</b> Uncorrected or Deferred error recorded because a poison value was detected	W1C	0b0
[21:20]	UET	Uncorrected Error Type  <b>0b00</b> Invalid  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code  <b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_hni_errmisc1_NS.ERRSRC for error type details.	W1C	0b0

### 8.3.12.27 por\_hni\_erraddr\_NS

Error Address Register for Non-secure

#### Configurations

This register is available in all configurations.

## Attributes

## Width

64

## Address offset

0xE058

## Type

RW

### Reset value

See individual bit resets

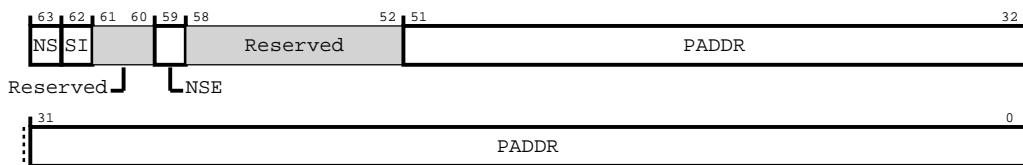
## Usage constraints

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 8-764: por\_hni\_erraddr\_NS**



**Table 8-773: por\_hni\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Non-Secure attribute (NS, also PAS[0])	RW	0b0
[62]	SI	PAS Incorrect  <b>0b0</b> PAS[1:0] is correct  <b>0b1</b> PAS[1:0] might not be correct	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Non-Secure Extension attribute (NSE, also PAS[1])	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	PADDR	Physical Address	RW	0b0

### 8.3.12.28 por\_hni\_errmisc1\_NS

Error Miscellaneous Register 1

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE068

##### Type

RW

##### Reset value

See individual bit resets

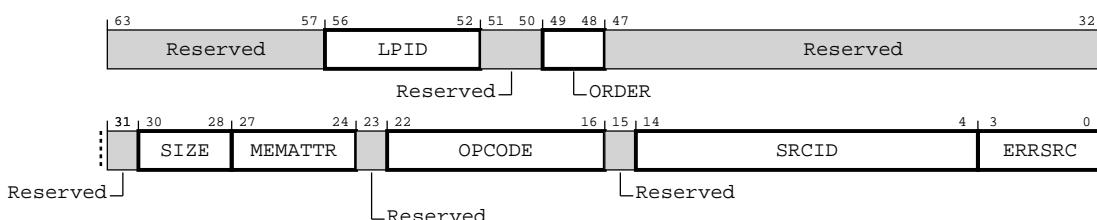
##### Usage constraints

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 8-765: por\_hni\_errmisc1\_NS**



**Table 8-774: por\_hni\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	
[56:52]	LPID	CHI Logic Processor ID (LPID) field	RW	0b0
[51:50]	Reserved	Reserved	RO	
[49:48]	ORDER	CHI Order field	RW	0b0
[47:31]	Reserved	Reserved	RO	
[30:28]	SIZE	CHI Size field	RW	0b0
[27:24]	MEMATTR	CHI Memory Attributes field	RW	0b0
[23]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[22:16]	OPCODE	CHI Opcode field	RW	0b0
[15]	Reserved	Reserved	RO	
[14:4]	SRCID	CHI Source ID field	RW	0b0
[3:0]	ERRSRC	Error source <b>0b0000</b> Coherent read <b>0b0001</b> Coherent write <b>0b0010</b> Coherent dataless (CleanUnique/MakeUnique/Evict/StashOnce/CMO) <b>0b0011</b> Atomic <b>0b0100</b> Illegal configuration read <b>0b0101</b> Illegal configuration write or dataless <b>0b0110</b> Configuration write data partial byte enable error <b>0b0111</b> Configuration write data parity error or poison error <b>0b1000</b> BRESP error <b>0b1001</b> Poison error <b>0b1010</b> BRESP error and poison error <b>0b1011</b> Unsupported Exclusive access (HN-P only) <b>NOTE</b> BRESP and poison errors (0x8-0xA) are Uncorrectable Errors (UE). SRCID and ERRSRC are the only valid fields in por_hni_errmisc1_NS. All other error types are Deferred Errors (DE) and all fields are valid.	RW	0b0

### 8.3.12.29 por\_hni\_errpfgf\_NS

Pseudo-fault Generation Feature Register for Non-secure

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE840

### Type

RO

### Reset value

See individual bit resets

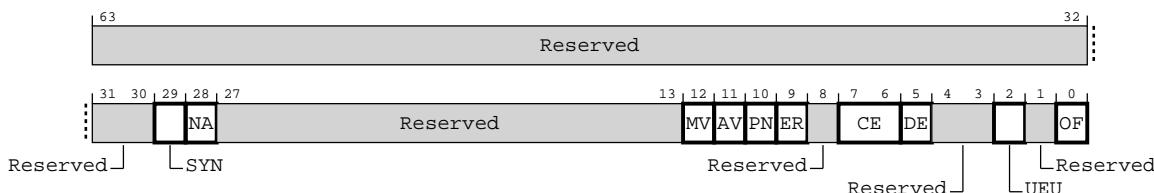
### Usage constraints

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 8-766: por\_hni\_errpfgf\_NS**



**Table 8-775: por\_hni\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection <b>0b1</b> Fault injection does not update por_hni_errstatus_NS.SERR	RO	0b1
[28]	NA	No access required <b>0b1</b> The component fakes detection of the error spontaneously in the fault injection state	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome <b>0b1</b> The node does not update any fields in por_hni_errmisc1_NS and sets por_hni_errstatus_NS.MV to por_hni_errpfgctl_NS.MV	RO	0b1
[11]	AV	Address syndrome <b>0b1</b> The node does not update por_hni_erraddr_NS and sets por_hni_errstatus_NS.AV to por_hni_errpfgctl_NS.AV	RO	0b1

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag <b>0b1</b> When an injected error is recorded, por_hni_errstatus_NS.PN is set to por_hni_errpfctl_NS.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> When an injected error is recorded, por_hni_errstatus_NS.ER is set to por_hni_errpfctl_NS.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation <b>0b00</b> The fault generation feature of the node cannot generate this type of error	RO	0b00
[5]	DE	Deferred Error generation <b>0b1</b> The fault generation feature of the node allows generation of this type of error	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Unrecoverable Error generation <b>0b1</b> The fault generation feature of the node allows generation of this type of error	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag <b>0b1</b> When an injected error is recorded, por_hni_errstatus_NS.OF is set to por_hni_errpfctl_NS.OF	RO	0b1

### 8.3.12.30 por\_hni\_errpfctl\_NS

Pseudo-fault Generation Control Register for Non-secure

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE848

##### Type

RW

##### Reset value

See individual bit resets

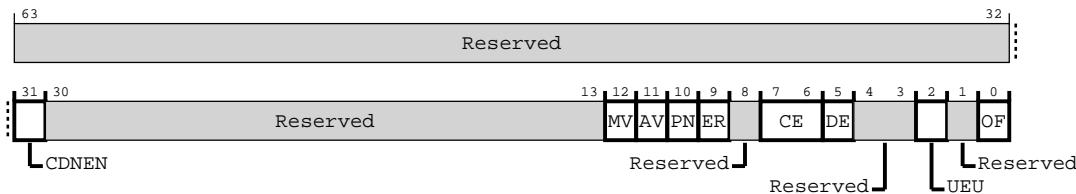
## Usage constraints

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 8-767: por\_hni\_errpfctl\_NS**



**Table 8-776: por\_hni\_errpfctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	<p>Countdown Enable</p> <p><b>0b0</b> The Error Generation Counter is disabled</p> <p><b>0b1</b> The Error Generation Counter is enabled</p>	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	<p>Miscellaneous syndrome</p> <p><b>0b0</b> por_hni_errstatus_NS.MV is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus_NS.MV is set to 0b1 when an injected error is recorded</p>	RW	0b0
[11]	AV	<p>Address syndrome</p> <p><b>0b0</b> por_hni_ERRSTATUS_NS.AV is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_ERRSTATUS_NS.AV is set to 0b1 when an injected error is recorded</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> por_hni_errstatus_NS.PN is set to 0b0 when an injected error is recorded</p> <p><b>0b1</b> por_hni_errstatus_NS.PN is set to 0b1 when an injected error is recorded</p>	RW	0b0

Bits	Name	Description	Type	Reset
[9]	ER	Error Reported flag  <b>0b0</b> por_hni_errstatus_NS.ER is set to 0b0 when an injected error is recorded  <b>0b1</b> por_hni_errstatus_NS.ER is set to 0b1 when an injected error is recorded	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation enable  <b>0b00</b> No error of this type will be generated	RW	0b00
[5]	DE	Deferred Error generation enable  <b>0b0</b> No error of this type will be generated  <b>0b1</b> An error of this type might be generated when the Error Generation Counter decrements to zero	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Unrecoverable Error generation enable  <b>0b0</b> No error of this type will be generated  <b>0b1</b> An error of this type might be generated when the Error Generation Counter decrements to zero	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag  <b>0b0</b> por_hni_errstatus_NS.OF is set to 0b0 when an injected error is recorded  <b>0b1</b> por_hni_errstatus_NS.OF is set to 0b1 when an injected error is recorded	RW	0b0

### 8.3.12.31 por\_hni\_errfgcdn\_NS

Pseudo-fault Generation Countdown Register for Non-secure

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE850

##### Type

RW

### Reset value

See individual bit resets

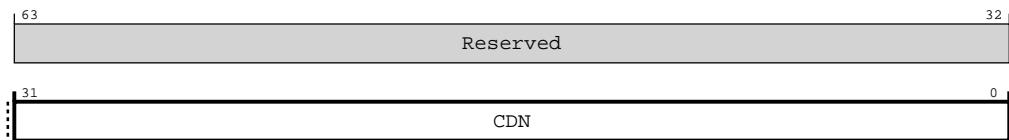
### Usage constraints

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 8-768: por\_hni\_errpfgcdn\_NS**



**Table 8-777: por\_hni\_errpfgcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.12.32 por\_hni\_errcapctl

Error Capture Control Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xED00

#### Type

RW

#### Reset value

See individual bit resets

#### Root group override

por\_hni\_rcr.ras

#### Secure group override

por\_hni\_scr.ras

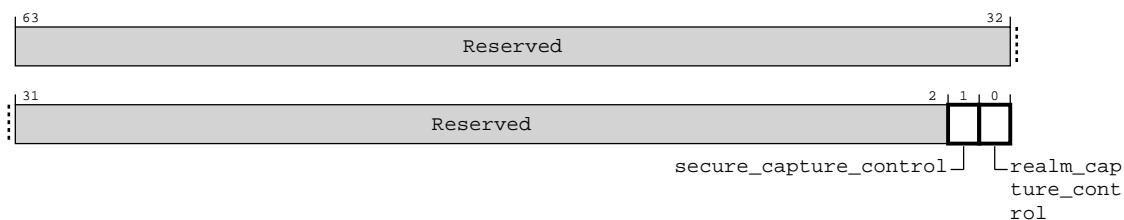
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_hni\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_hni\_scr.ras bit and por\_hni\_rcr.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 8-769: por\_hni\_errcapctl**



**Table 8-778: por\_hni\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control <b>0b0</b> Transaction with Secure PAS captured in Root error record <b>0b1</b> Transaction with Secure PAS captured in Non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control <b>0b0</b> Transaction with Realm PAS captured in Root error record <b>0b1</b> Transaction with Realm PAS captured in Non-secure error record	RW	0b0

### 8.3.12.33 por\_hni\_errgsr

Error Group Status Register

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEE00

### Type

RO

### Reset value

See individual bit resets

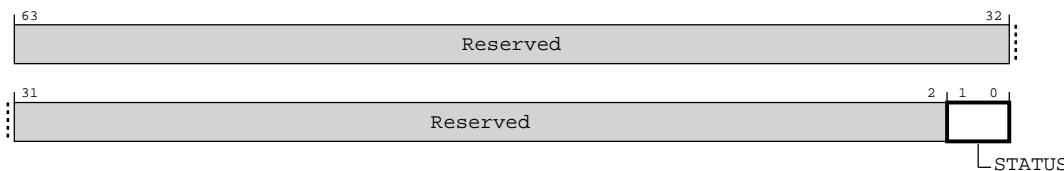
### Usage constraints

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 8-770: por\_hni\_errgsr**



**Table 8-779: por\_hni\_errgsr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	STATUS	Read-only copy of {por_hni_errstatus_NS.V, por_hni_errstatus.V}	RO	0b0

### 8.3.12.34 por\_hni\_errildr

Implementation Identification Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEE10

#### Type

RO

### Reset value

See individual bit resets

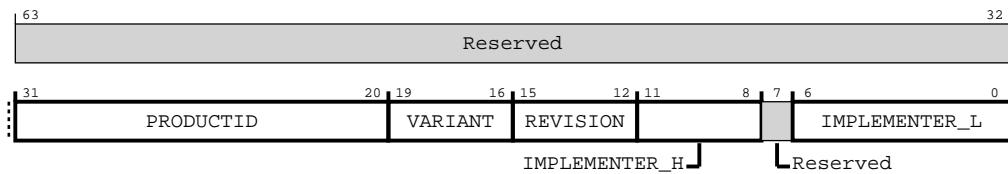
### Usage constraints

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 8-771: por\_hni\_errildr**



**Table 8-780: por\_hni\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	PRODUCTID	Product Part number	RO	0x0
[19:16]	VARIANT	Component major revision	RO	0x0
[15:12]	REVISION	Component minor revision	RO	0x0
[11:8]	IMPLEMENTER_H	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	IMPLEMENTER_L	Implementer[6:0]	RO	0x3B

### 8.3.12.35 por\_hni\_errdevaff

Device Affinity Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFA8

##### Type

RO

### Reset value

See individual bit resets

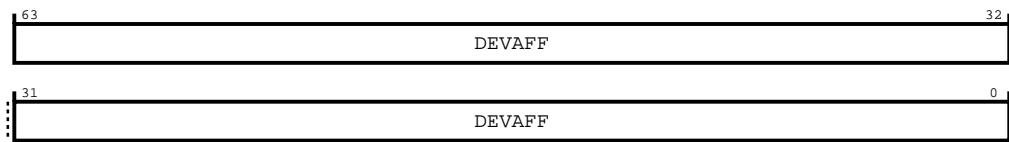
### Usage constraints

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 8-772: por\_hni\_errdevaff**



**Table 8-781: por\_hni\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	DEVAFF	Device affinity register	RO	0b0

### 8.3.12.36 por\_hni\_errdevarch

Device Architecture Register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEF8

#### Type

RO

### Reset value

See individual bit resets

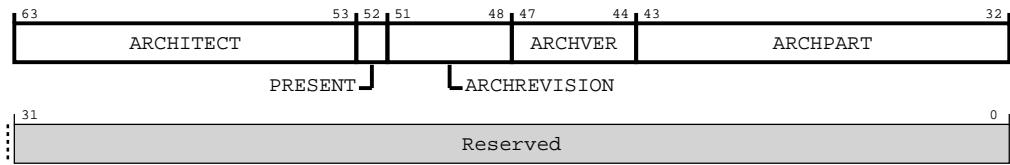
### Usage constraints

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 8-773: por\_hni\_errdevarch**



**Table 8-782: por\_hni\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	ARCHITECT	Architect <b>0x23B</b> JEP106 continuation code 0x4, ID code 0x3B. Arm Limited.	RO	0x23B
[52]	PRESENT	DEVARC Present <b>0b1</b> Device Architecture information present	RO	0b1
[51:48]	ARCHREVISION	Architecture revision <b>0x1</b> RAS System Architecture v1.1	RO	0b1
[47:44]	ARCHVER	Architecture Version <b>0x0</b> RAS System Architecture v1	RO	0x0
[43:32]	ARCHPART	Architecture Part <b>0xA00</b> RAS System Architecture	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

### 8.3.12.37 por\_hni\_errdevid

Device Configuration Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFC8

### Type

RO

### Reset value

See individual bit resets

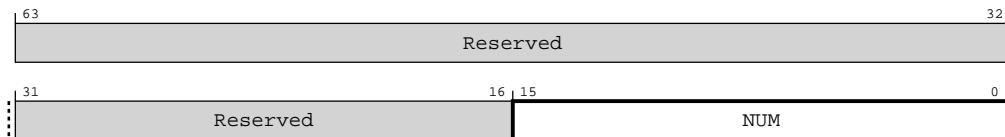
### Usage constraints

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 8-774: por\_hni\_errdevid**



**Table 8-783: por\_hni\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	NUM	Number of error records	RO	0x2

### 8.3.12.38 por\_hni\_errpidr45

Peripheral Identification Register 4 and 5

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEFDO

#### Type

RO

#### Reset value

See individual bit resets

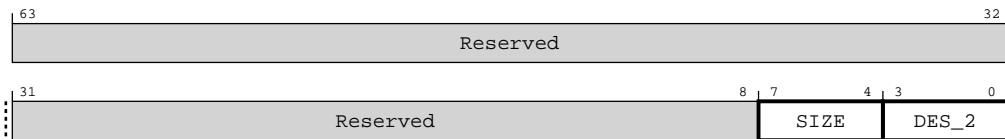
## Usage constraints

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 8-775: por\_hni\_errpidr45**



**Table 8-784: por\_hni\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	SIZE	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	DES_2	Designer bit[10:7]	RO	0x4

## 8.3.12.39 por\_hni\_errpidr01

Peripheral Identification Register 0 and 1

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEFE0

#### Type

RO

#### Reset value

See individual bit resets

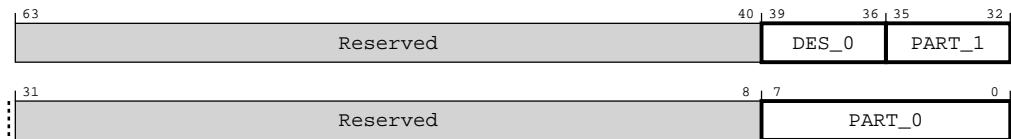
## Usage constraints

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 8-776: por\_hni\_errpidr01**



**Table 8-785: por\_hni\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	DES_0	Designer bit[3:0]	RO	0xb
[35:32]	PART_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	PART_0	Product ID Part 0	RO	0x0

## 8.3.12.40 por\_hni\_errpidr23

Peripheral Identification Register 2 and 3

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEFE8

#### Type

RO

#### Reset value

See individual bit resets

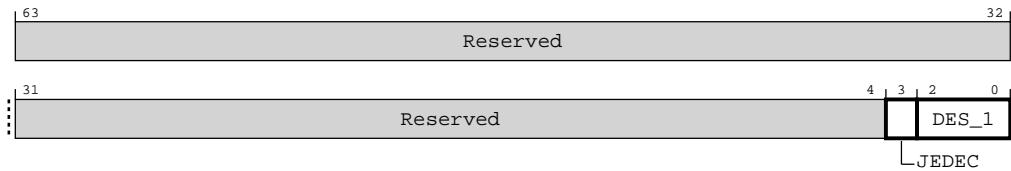
#### Usage constraints

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 8-777: por\_hni\_errpidr23**



**Table 8-786: por\_hni\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	JEDEC	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	DES_1	Designer bit[6:4]	RO	0x3

### 8.3.12.41 por\_hni\_errcidr01

Component Identification Register 0 and 1

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFF0

##### Type

RO

##### Reset value

See individual bit resets

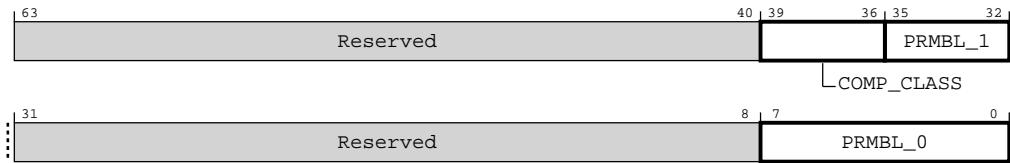
##### Usage constraints

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 8-778: por\_hni\_errcidr01**



**Table 8-787: por\_hni\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	COMP_CLASS	Component Class	RO	0xF
[35:32]	PRMBL_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	PRMBL_0	PRMBL_0	RO	0xD

### 8.3.12.42 por\_hni\_errcidr23

Component Identification Register 2 and 3

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFF8

##### Type

RO

##### Reset value

See individual bit resets

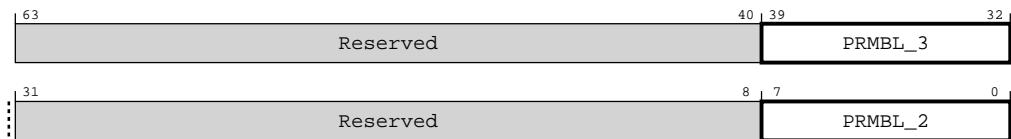
#### Usage constraints

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 8-779: por\_hni\_errcidr23**



**Table 8-788: por\_hni\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	PRMBL_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	PRMBL_2	PRMBL_2	RO	0x5

### 8.3.12.43 por\_hni\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

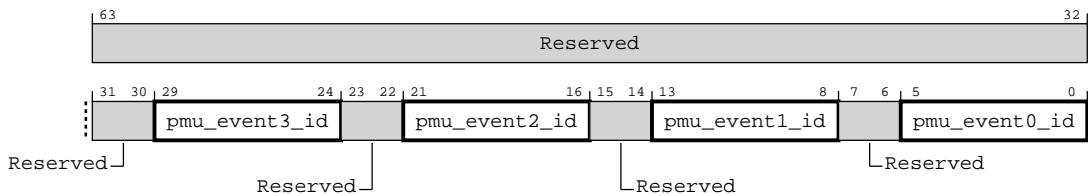
##### Usage constraints

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 8-780: por\_hni\_pmu\_event\_sel**



**Table 8-789: por\_hni\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29:24]	pmu_event3_id	HN-I PMU Event 3 select; see pmu_event0_id for encodings	RW	0b0
[23:22]	Reserved	Reserved	RO	
[21:16]	pmu_event2_id	HN-I PMU Event 2 select; see pmu_event0_id for encodings	RW	0b0
[15:14]	Reserved	Reserved	RO	
[13:8]	pmu_event1_id	HN-I PMU Event 1 select; see pmu_event0_id for encodings	RW	0b0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>HN-I PMU Event 0 select</p> <p><b>0x00</b> No event</p> <p><b>0x20</b> RRT read occupancy count overflow</p> <p><b>0x21</b> RRT write occupancy count overflow</p> <p><b>0x22</b> RDT read occupancy count overflow</p> <p><b>0x23</b> RDT write occupancy count overflow</p> <p><b>0x24</b> WDB occupancy count overflow</p> <p><b>0x25</b> RRT read allocation</p> <p><b>0x26</b> RRT write allocation</p> <p><b>0x27</b> RDT read allocation</p> <p><b>0x28</b> RDT write allocation</p> <p><b>0x29</b> WDB allocation</p> <p><b>0x2A</b> RETRYACK TXRSP flit sent</p> <p><b>0x2B</b> ARVALID set without ARREADY event</p> <p><b>0x2C</b> ARREADY set without ARVALID event</p> <p><b>0x2D</b> AWVALID set without AWREADY event</p> <p><b>0x2E</b> AWREADY set without AWVALID event</p> <p><b>0x2F</b> WVALID set without WREADY event</p> <p><b>0x30</b> TXDAT stall (TXDAT valid but no link credit available)</p> <p><b>0x31</b> Non-PCIe serialization event</p> <p><b>0x32</b> PCIe serialization event</p> <p><b>NOTE</b> All other encodings are reserved.</p>	RW	0b0

### 8.3.12.44 por\_hnp\_pmu\_event\_sel

Specifies the PMU event to be counted. HNP only

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xD908

##### Type

RW

##### Reset value

See individual bit resets

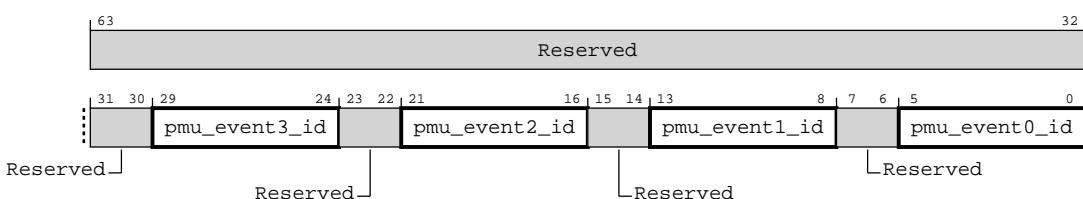
##### Usage constraints

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 8-781: por\_hnp\_pmu\_event\_sel**



**Table 8-790: por\_hnp\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29:24]	pmu_event3_id	P2P Slice PMU Event 3 select; see pmu_event0_id for encodings"	RW	0b0
[23:22]	Reserved	Reserved	RO	
[21:16]	pmu_event2_id	P2P Slice PMU Event 2 select; see pmu_event0_id for encodings	RW	0b0
[15:14]	Reserved	Reserved	RO	
[13:8]	pmu_event1_id	P2P Slice PMU Event 1 select; see pmu_event0_id for encodings	RW	0b0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>P2P Slice PMU Event 0 select</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> P2P Write Slice RRT write occupancy count overflow</p> <p><b>0x02</b> P2P Write Slice RDT write occupancy count overflow</p> <p><b>0x03</b> P2P Write Slice WDB occupancy count overflow</p> <p><b>0x04</b> P2P Write Slice RRT write allocation</p> <p><b>0x05</b> P2P Write Slice RDT write allocation</p> <p><b>0x06</b> P2P Write Slice WDB allocation</p> <p><b>0x07</b> P2P Write Slice AWVALID set without AWREADY event</p> <p><b>0x08</b> P2P Write Slice AWREADY set without AWVALID event</p> <p><b>0x09</b> P2P Write Slice WVALID set without WREADY event</p> <p><b>0x11</b> P2P Read Slice RRT read occupancy count overflow</p> <p><b>0x12</b> P2P Read Slice RDT read occupancy count overflow</p> <p><b>0x13</b> P2P Read Slice RRT read allocation</p> <p><b>0x14</b> P2P Read Slice RDT read allocation</p> <p><b>0x15</b> P2P Read Slice ARVALID set without ARREADY event</p> <p><b>0x16</b> P2P Read Slice ARREADY set without ARVALID event</p> <p><b>NOTE</b> All other encodings are reserved.</p>	RW	0b0

### 8.3.13 MTU register summary

The following table describes the registers for the relevant component.

**Table 8-791: por\_mtu\_cfg register summary**

Offset	Name	Type	Description
0x0	por_mtu_node_info	RO	por_mtu_node_info
0x80	por_mtu_child_info	RO	por_mtu_child_info
0x980	por_mtu_rcr	RW	por_mtu_rcr
0x988	por_mtu_scr	RW	por_mtu_scr
0x900	por_mtu_unit_info	RO	por_mtu_unit_info
0xA00	por_mtu_cfg_ctl	RW	por_mtu_cfg_ctl
0xA08	por_mtu_aux_ctl	RW	por_mtu_aux_ctl
0xA30	por_mtu_tc_flush_pr	RW	por_mtu_tc_flush_pr
0xA38	por_mtu_tc_flush_sr	RO	por_mtu_tc_flush_sr
0xA40	por_mtu_tag_addr_ctl	RW	por_mtu_tag_addr_ctl
0xA48	por_mtu_tag_addr_base	RW	por_mtu_tag_addr_base
0xA50 : 0xA60	por_mtu_tag_addr_shutter0-2	RW	por_mtu_tag_addr_shutter0-2
0xE000	por_mtu_errfr	RO	por_mtu_errfr
0xE008	por_mtu_errctlr	RW	por_mtu_errctlr
0xE010	por_mtu_errstatus	W1C	por_mtu_errstatus
0xE018	por_mtu_erraddr	RW	por_mtu_erraddr
0xE020	por_mtu_errmisc0	RW	por_mtu_errmisc0
0xE028	por_mtu_errmisc1	RW	por_mtu_errmisc1
0xE800	por_mtu_errpfgf	RO	por_mtu_errpfgf
0xE808	por_mtu_errpfgctl	RW	por_mtu_errpfgctl
0xE810	por_mtu_errpfgcdn	RW	por_mtu_errpfgcdn
0xE040	por_mtu_errfr_NS	RO	por_mtu_errfr_NS
0xE048	por_mtu_errctlr_NS	RW	por_mtu_errctlr_NS
0xE050	por_mtu_errstatus_NS	W1C	por_mtu_errstatus_NS
0xE058	por_mtu_erraddr_NS	RW	por_mtu_erraddr_NS
0xE060	por_mtu_errmisc0_NS	RW	por_mtu_errmisc0_NS
0xE068	por_mtu_errmisc1_NS	RW	por_mtu_errmisc1_NS
0xE840	por_mtu_errpfgf_NS	RO	por_mtu_errpfgf_NS
0xE848	por_mtu_errpfgctl_NS	RW	por_mtu_errpfgctl_NS
0xE850	por_mtu_errpfgcdn_NS	RW	por_mtu_errpfgcdn_NS
0xED00	por_mtu_errcapctl	RW	por_mtu_errcapctl
0xEE00	por_mtu_errgsr	RO	por_mtu_errgsr
0xEE10	por_mtu_erriidr	RO	por_mtu_erriidr
0xEF88	por_mtu_errdevaff	RO	por_mtu_errdevaff
0xEF8B	por_mtu_errdevarch	RO	por_mtu_errdevarch
0xEFC8	por_mtu_errdevid	RO	por_mtu_errdevid

Offset	Name	Type	Description
0xEF00	por_mtu_errpidr45	RO	<a href="#">por_mtu_errpidr45</a>
0xEF00	por_mtu_errpidr01	RO	<a href="#">por_mtu_errpidr01</a>
0xEF00	por_mtu_errpidr23	RO	<a href="#">por_mtu_errpidr23</a>
0xEFF0	por_mtu_errcidr01	RO	<a href="#">por_mtu_errcidr01</a>
0xEFF8	por_mtu_errcidr23	RO	<a href="#">por_mtu_errcidr23</a>
0xE030	por_mtu_err_inj	RW	<a href="#">por_mtu_err_inj</a>
0xB80	por_mtu_cfg_tc_dbgrd	WO	<a href="#">por_mtu_cfg_tc_dbgrd</a>
0xB88	por_mtu_tc_cache_access_tc_ctl	RO	<a href="#">por_mtu_tc_cache_access_tc_ctl</a>
0xB98	por_mtu_tc_cache_access_tc_data	RO	<a href="#">por_mtu_tc_cache_access_tc_data</a>
0xD900	por_mtu_pmu_event_sel	RW	<a href="#">por_mtu_pmu_event_sel</a>

### 8.3.13.1 por\_mtu\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

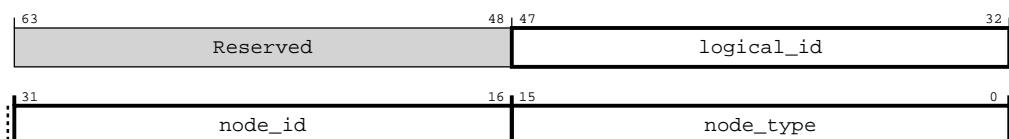
##### Usage constraints

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 8-782: por\_mtu\_node\_info**



**Table 8-792: por\_mtu\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0010

### 8.3.13.2 por\_mtu\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

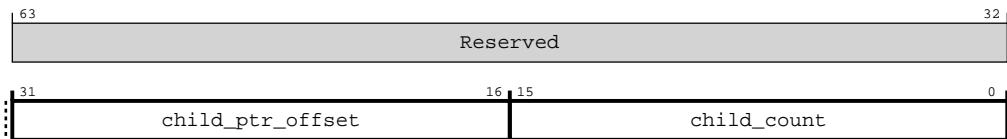
##### Usage constraints

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 8-783: por\_mtu\_child\_info**



**Table 8-793: por\_mtu\_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	0x0

Bits	Name	Description	Type	Reset
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.13.3 por\_mtu\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

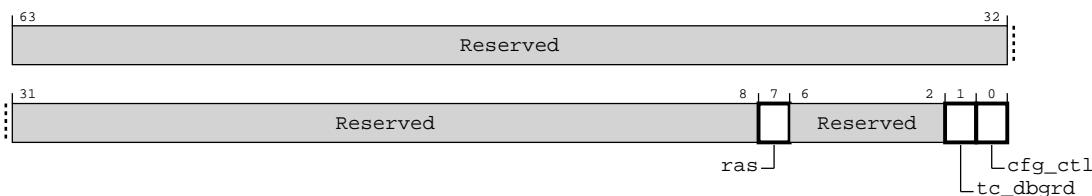
##### Usage constraints

This register is owned in the Root space and is 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 8-784: por\_mtu\_rcr**



**Table 8-794: por\_mtu\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allow Root override of the RAS registers	RW	0b0
[6:2]	Reserved	Reserved	RO	
[1]	tc_dbgrd	Allows Root override of the secure debug register (por_mtu_cfg_tc_dgbrd)	RW	0b0
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.13.4 por\_mtu\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

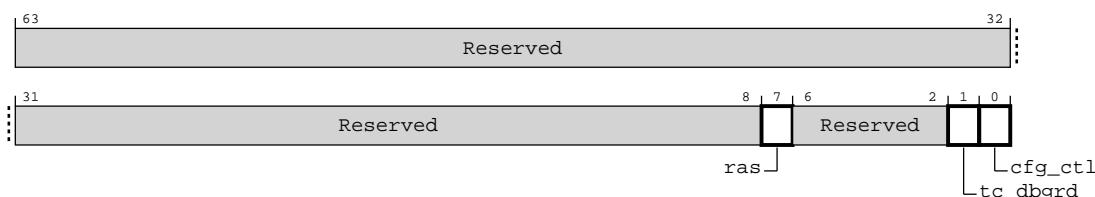
#### Usage constraints

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 8-785: por\_mtu\_scr**



**Table 8-795: por\_mtu\_scr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allow Secure override of the RAS registers	RW	0b0
[6:2]	Reserved	Reserved	RO	
[1]	tc_dbgrd	Allows Secure override of the secure debug register (por_mtu_cfg_tc_dbgrd)	RW	0b0
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.13.5 por\_mtu\_unit\_info

Provides component identification information for MTU.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

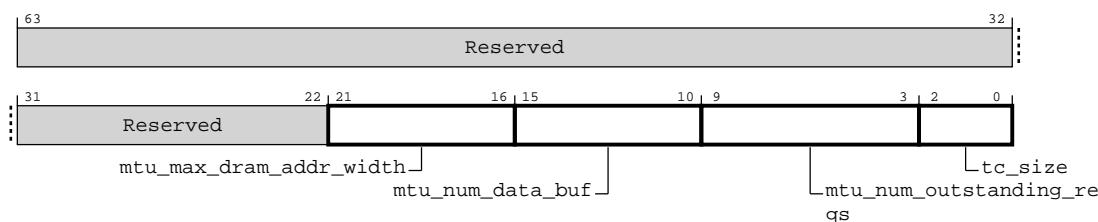
##### Usage constraints

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 8-786: por\_mtu\_unit\_info**



**Table 8-796: por\_mtu\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:16]	mtu_max_dram_addr_width	DRAM Addr width	RO	Configuration dependent
[15:10]	mtu_num_data_buf	Number of data buffers in MTU	RO	Configuration dependent
[9:3]	mtu_num_outstanding_reqs	Maximum number of outstanding AXI requests from MTU	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[2:0]	tc_size	TC size <b>0b000</b> No TC <b>0b001</b> 128KB <b>0b010</b> 256KB <b>0b011</b> 512KB <b>0b100</b> 1MB <b>0b101</b> 2MB <b>0b110</b> 32KB, <b>0b111</b> 64KB	RO	Configuration dependent

### 8.3.13.6 por\_mtu\_cfg\_ctl

Functions as the configuration control register for MTU.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.cfg\_ctl

##### Secure group override

por\_mtu\_scr.cfg\_ctl

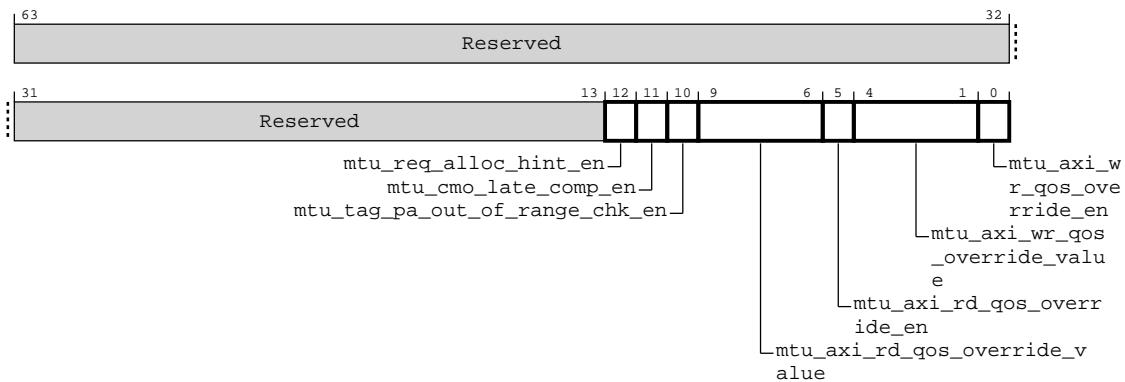
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.cfg\_ctl bit and por\_mtu\_rsr.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 8-787: por\_mtu\_cfg\_ctl**



**Table 8-797: por\_mtu\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12]	mtu_req_alloc_hint_en	Request Alloc Hint is used to determine TC allocation for TC Miss.  <b>Note</b> If this bit is clear, Tags are always allocated in TC. During no_tc_mode, and no_fill_mode; Tags are never allocated.	RW	0b0
[11]	mtu_cmo_late_comp_en	Enables CMO completion only when data is flushed out of TC to Memory	RW	0b0
[10]	mtu_tag_pa_out_of_range_chk_en	Enables Tag Address Out of Range checking.	RW	0b0
[9:6]	mtu_axi_rd_qos_override_value	QoS Override value to be used for all AXI Read requests generated by MTU. This value is used only when axi_rd_qos_override_en is set.	RW	0b0000
[5]	mtu_axi_rd_qos_override_en	Enables QoS override for all AXI Read requests.	RW	0b0
[4:1]	mtu_axi_wr_qos_override_value	QoS Override value to be used for all AXI Write requests generated by MTU. This value is used only when axi_wr_qos_override_en is set.	RW	0b0000
[0]	mtu_axi_wr_qos_override_en	Enables QoS override for all AXI Write requests.	RW	0b0

### 8.3.13.7 por\_mtu\_aux\_ctl

Functions as the auxiliary control register for the MTU

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

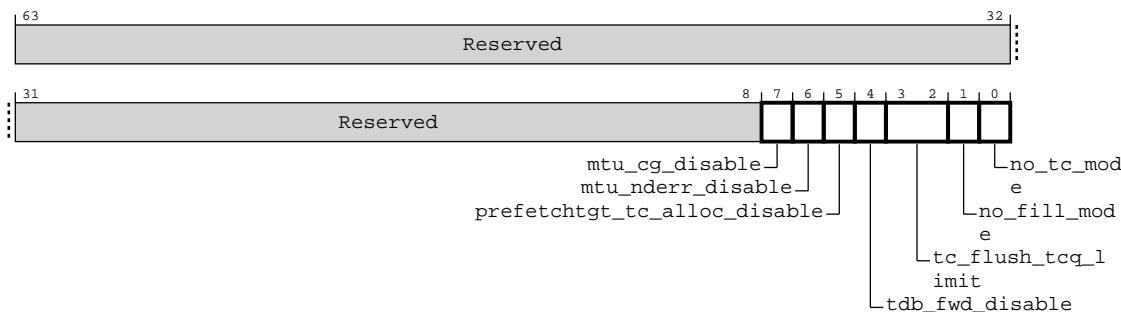
#### Usage constraints

This register is owned in the Root space and is 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 8-788: por\_mtu\_aux\_ctl**



**Table 8-798: por\_mtu\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	mtu_cg_disable	Disables MTU architectural clock gating.	RW	0b0
[6]	mtu_nderr_disable	Disable NDERR from MTU to RN for any Tag errors.	RW	0b0
[5]	prefetchtgt_tc_alloc_disable	Do not allocate PrefetchTgt requests in TC.	RW	0b0
[4]	tdb_fwd_disable	Disable Tag Data Buffer forwarding from a request to a dependent child request.	RW	0b0

Bits	Name	Description	Type	Reset
[3:2]	tc_flush_tcq_limit	<p>Controls number of TC Flush requests allowed to occupy TCQ entries.</p> <p><b>0b00</b> TC Flush Requests allowed to take one TCQ entry.</p> <p><b>0b01</b> TC Flush Requests allowed to take 25% of TCQ entries.</p> <p><b>0b10</b> TC Flush Requests allowed to take 50% of TCQ entries.</p> <p><b>0b11</b> TC Flush Requests allowed to take All TCQ entries.</p>	RW	0b10
[1]	no_fill_mode	Enables No Fill Mode for Tag Cache. When set, no new lines would be allocated in Tag Cache.	RW	0b0
[0]	no_tc_mode	Enables No TC Mode; disables MTU Tag Cache when set.	RW	0b0

### 8.3.13.8 por\_mtu\_tc\_flush\_pr

Functions as Tag Cache Flush Policy Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA30

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.cfg\_ctl

##### Secure group override

por\_mtu\_scr.cfg\_ctl

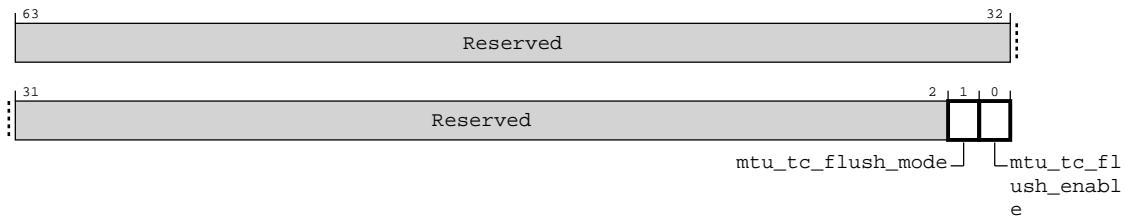
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.cfg\_ctl bit and por\_mtu\_rcr.cfg\_ctl bit are set, Non-secure and Realm accesses to this register are permitted. This register can be modified only with prior written permission from Arm.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-789: por\_mtu\_tc\_flush\_pr**



**Table 8-799: por\_mtu\_tc\_flush\_pr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	mtu_tc_flush_mode	Tag Cache Flush Mode <b>0b0</b> Clean Invalid. WB dirty data and invalidate local copy in TC. <b>0b1</b> Clean Shared. WB dirty data and keep clean copy in TC.	RW	0b0
[0]	mtu_tc_flush_enable	Start Tag Cache Flush	RW	0b0

### 8.3.13.9 por\_mtu\_tc\_flush\_sr

Functions as Tag Cache Flush Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA38

##### Type

RO

##### Reset value

See individual bit resets

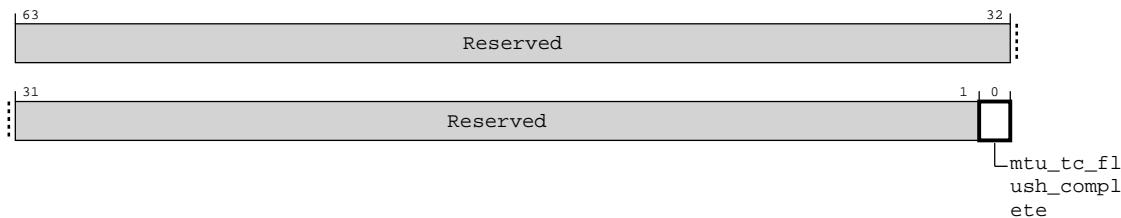
## Usage constraints

This register is owned in the Non-secure space and is accessible using Non-secure, Secure, Root, and Realm transactions. This register can be modified only with prior written permission from Arm.

## Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-790: por\_mtu\_tc\_flush\_sr**



**Table 8-800: por\_mtu\_tc\_flush\_sr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	mtu_tc_flush_complete	Tag Cache Flush Complete	RO	0b0

## 8.3.13.10 por\_mtu\_tag\_addr\_ctl

PA to DA address conversion control

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA40

#### Type

RW

#### Reset value

See individual bit resets

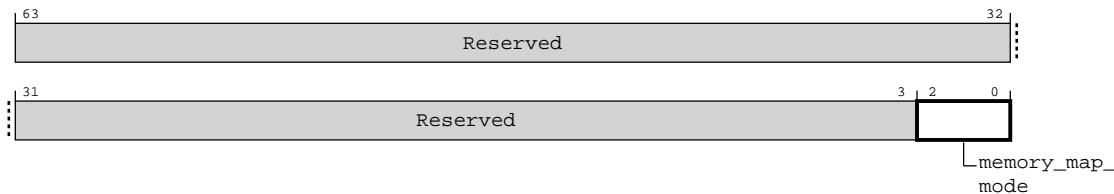
### Usage constraints

This register is owned in the Root space and is 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 8-791: por\_mtu\_tag\_addr\_ctl**



**Table 8-801: por\_mtu\_tag\_addr\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2:0]	memory_map_mode	Memory map mode used for translating data physical address to data DRAM address <b>0b000</b> Pass-through <b>0b001</b> PDD <b>0b010</b> Infra <b>0b011</b> Infra with 2 sockets <b>0b100</b> Map Type 0	RW	0b0

### 8.3.13.11 por\_mtu\_tag\_addr\_base

Physical address of tag base

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA48

##### Type

RW

### Reset value

See individual bit resets

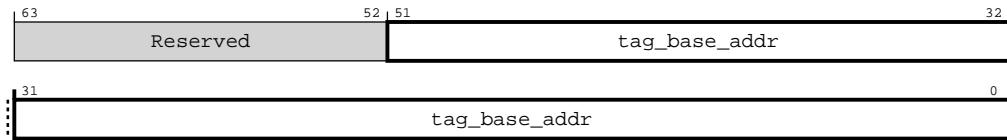
### Usage constraints

This register is owned in the Root space and is 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 8-792: por\_mtu\_tag\_addr\_base**



**Table 8-802: por\_mtu\_tag\_addr\_base attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:0]	tag_base_addr	52-bit Physical address for tag base	RW	0b0

### 8.3.13.12 por\_mtu\_tag\_addr\_shutter0-2

There are 3 iterations of this register. The index ranges from 0 to 2. shutter value to generate DRAM address from physical address

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA50 + #{8\*index}

#### Type

RW

#### Reset value

See individual bit resets

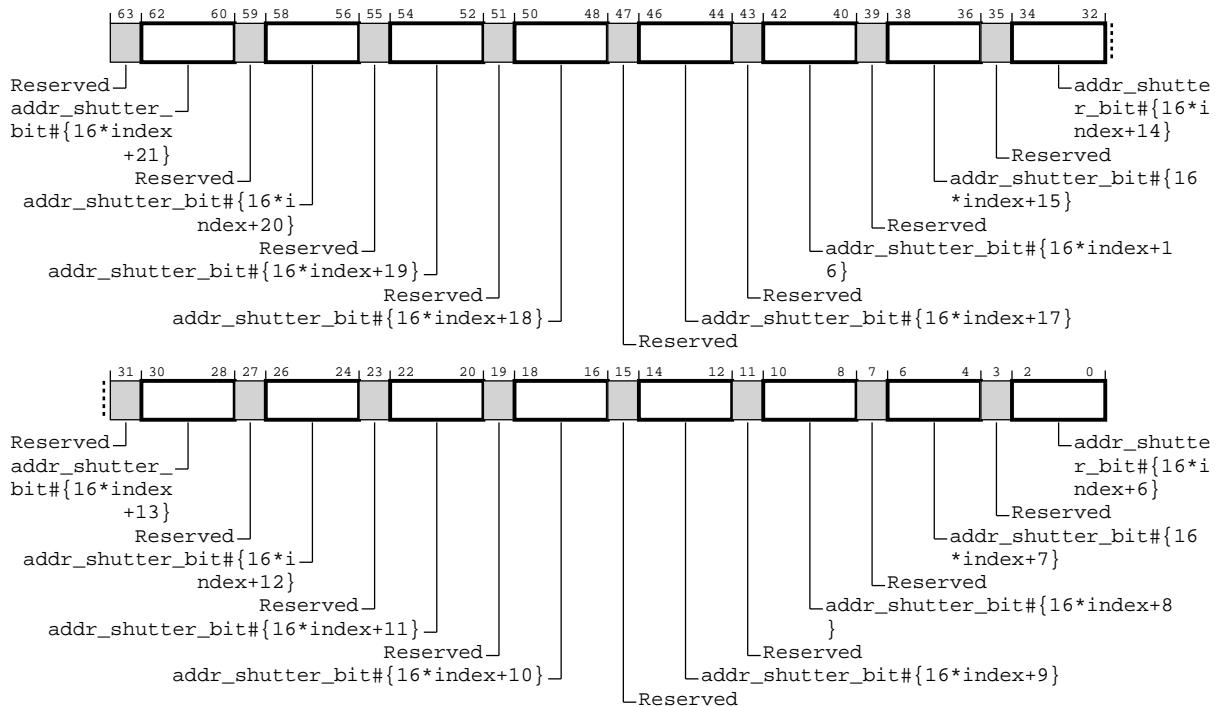
#### Usage constraints

This register is owned in the Root space and is 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 8-793: por\_mtu\_tag\_addr\_shutter0-2**



**Table 8-803: por\_mtu\_tag\_addr\_shutter0-2 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[62:60]	addr_shutter_bit#{16*index+21}	Program to specify how shuttered address bit #{16*index+21} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[59]	Reserved	Reserved	RO	
[58:56]	addr_shutter_bit#{16*index+20}	Program to specify how shuttered address bit #{16*index+20} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[55]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[54:52]	addr_shutter_bit#{16*index+19}	Program to specify how shuttered address bit #{16*index+19} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[51]	Reserved	Reserved	RO	
[50:48]	addr_shutter_bit#{16*index+18}	Program to specify how shuttered address bit #{16*index+18} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[47]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[46:44]	addr_shutter_bit#{16*index+17}	Program to specify how shuttered address bit #{16*index+17} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[43]	Reserved	Reserved	RO	
[42:40]	addr_shutter_bit#{16*index+16}	Program to specify how shuttered address bit #{16*index+16} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[39]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[38:36]	addr_shutter_bit#{16*index+15}	Program to specify how shuttered address bit #{16*index+15} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[35]	Reserved	Reserved	RO	
[34:32]	addr_shutter_bit#{16*index+14}	Program to specify how shuttered address bit #{16*index+14} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[31]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[30:28]	addr_shutter_bit#{16*index+13}	Program to specify how shuttered address bit #{16*index+13} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[27]	Reserved	Reserved	RO	
[26:24]	addr_shutter_bit#{16*index+12}	Program to specify how shuttered address bit #{16*index+12} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[23]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[22:20]	addr_shutter_bit#{16*index+11}	Program to specify how shuttered address bit #{16*index+11} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[19]	Reserved	Reserved	RO	
[18:16]	addr_shutter_bit#{16*index+10}	Program to specify how shuttered address bit #{16*index+10} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[15]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[14:12]	addr_shutter_bit#{16*index+9}	Program to specify how shuttered address bit #{16*index+9} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[11]	Reserved	Reserved	RO	
[10:8]	addr_shutter_bit#{16*index+8}	Program to specify how shuttered address bit #{16*index+8} should be driven from post-translation address  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5 <b>0b110</b> shift_6 <b>0b111</b> shift_7	RW	0b0
[7]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[6:4]	addr_shutter_bit#{16*index+7}	<p>Program to specify how shuttered address bit #{16*index+7} should be driven from post-translation address</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p> <p><b>0b110</b> shift_6</p> <p><b>0b111</b> shift_7</p>	RW	0b0
[3]	Reserved	Reserved	RO	
[2:0]	addr_shutter_bit#{16*index+6}	<p>Program to specify how shuttered address bit #{16*index+6} should be driven from post-translation address</p> <p><b>0b000</b> pass-through</p> <p><b>0b001</b> shift_1</p> <p><b>0b010</b> shift_2</p> <p><b>0b011</b> shift_3</p> <p><b>0b100</b> shift_4</p> <p><b>0b101</b> shift_5</p> <p><b>0b110</b> shift_6</p> <p><b>0b111</b> shift_7</p>	RW	0b0

### 8.3.13.13 por\_mtu\_errfr

Functions as the Root PAS error feature register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE000

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.ras

##### Secure group override

por\_mtu\_scr.ras

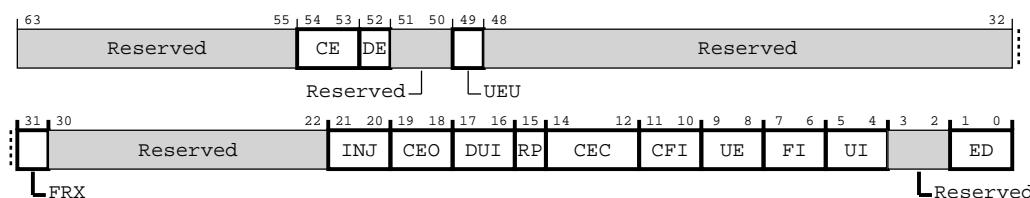
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-794: por\_mtu\_errfr**



**Table 8-804: por\_mtu\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[54:53]	CE	Corrected Error recording  <b>0b00</b> Corrected Error not supported  <b>0b10</b> Non-specific Corrected Error supported	RO	0b10
[52]	DE	Deferred Error recording  <b>0b0</b> Deferred Error not supported	RO	0b0
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension  <b>0b1</b> por_mtu_errfr[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension  <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite  <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors	RO	0b00
[15]	RP	Repeat counter (valid only when por_mtu_errfr.CEC == 0b100.)  <b>0b0</b> Invalid  <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in por_mtu_errmisco	RO	0b1
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter in por_mtu_errmisco	RO	0b100

Bits	Name	Description	Type	Reset
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_mtu_errctlr.CFI.	RO	0b10
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt for Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_mtu_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_mtu_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_mtu_errctlr.ED	RO	0b10

### 8.3.13.14 por\_mtu\_errctlr

Functions as the Root PAS 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

0xE008

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

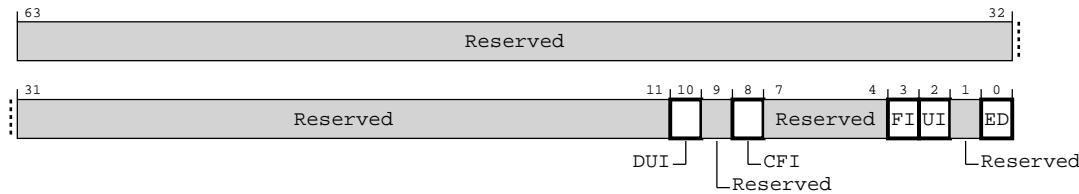
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-795: por\_mtu\_errctlr**



**Table 8-805: por\_mtu\_errctlr attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Enables error recovery interrupt for deferred error as specified in por_mtu_errfr.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_mtu_errfr.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected errors as specified in por_mtu_errfr.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_mtu_errfr.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_mtu_errfr.ED	RW	0b0

### 8.3.13.15 por\_mtu\_errstatus

Functions as the Root PAS 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

0xE010

### Type

W1C

### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

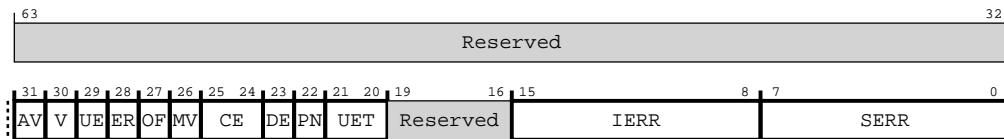
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-796: por\_mtu\_errstatus**



**Table 8-806: por\_mtu\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_mtu_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0

Bits	Name	Description	Type	Reset
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_mtu_errmisc<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison <b>0b1</b> Uncorrected error recorded because a poison value was consumed <b>0b0</b> Other cases	W1C	0b0

Bits	Name	Description	Type	Reset
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> CMN implementation defined error. Refer to por_mtu_errmisc1.ERRSRC for error type details.  <b>0x06</b> ECC error on L3 data  <b>0x07</b> ECC error on L3/SF Tag  <b>0x0A</b> Producer write data was poisoned but ACE-Lite does not support poisoned data  <b>0x0D</b> Illegal address  <b>0x1A</b> Parity error	W1C	0b0

### 8.3.13.16 por\_mtu\_erraddr

Functions as the Root PAS error address register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE018

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-797: por\_mtu\_erraddr**



**Table 8-807: por\_mtu\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Secure status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address for Data Request (Data PA). Same PA received from HN.  <b>Note</b> For Errors on TC evictions, this represents address that caused eviction.  <b>Note</b> Address is undefined if errmisc.OPTYPE is Other. For most part, these transactions are set/way based and not PA based.	RW	0b0

### 8.3.13.17 por\_mtu\_errmisco

Functions as the Root PAS miscellaneous error register 0. Contains information about corrected errors.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE020

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.ras

##### Secure group override

por\_mtu\_scr.ras

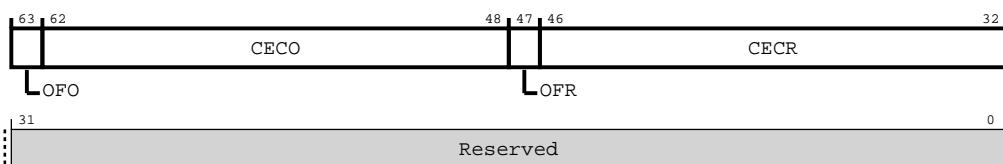
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-798: por\_mtu\_errmisco**



**Table 8-808: por\_mtu\_errmisco attributes**

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	0b0
[62:48]	CECO	Corrected ECC error count	RW	0b0

Bits	Name	Description	Type	Reset
[47]	OFR	Corrected error counter overflow	RW	0b0
[46:32]	CECR	Corrected ECC error count	RW	0b0
[31:0]	Reserved	Reserved	RO	

### 8.3.13.18 por\_mtu\_errmisc1

Functions as the Root PAS miscellaneous error register 1. Contains additional information about errors.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE028

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.ras

##### Secure group override

por\_mtu\_scr.ras

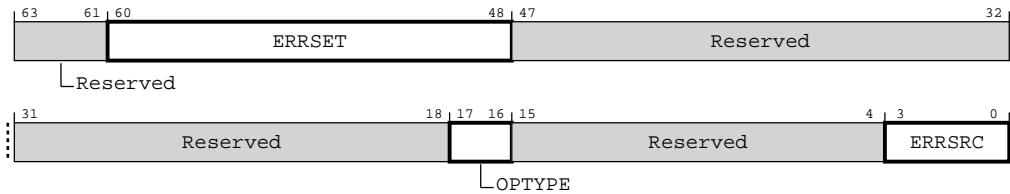
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-799: por\_mtu\_errmisc1**



**Table 8-809: por\_mtu\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	TC set address for ECC Single bit error	RW	0b0
[47:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error opcode type	RW	0b00
		0b00 Read Type (RD_NO_SNP, PrefetchTgt)		
		0b01 Write (WR_NO_SNP)		
		0b10 CMO, WR+CMO		
		0b11 Other		
[15:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	<p>Error source</p> <p><b>0b0001</b> Data single-bit ECC</p> <p><b>0b0010</b> Data double-bit ECC</p> <p><b>0b0011</b> Single-bit ECC overflow</p> <p><b>0b0101</b> Control single-bit ECC</p> <p><b>0b0110</b> Control double-bit ECC</p> <p><b>0b1000</b> AXI AR Subordinate Error</p> <p><b>0b1001</b> AXI AR Decode Error</p> <p><b>0b1010</b> AXI AR Poison Error</p> <p><b>0b1011</b> AXI AR Datachk Error</p> <p><b>0b1100</b> AXI W Subordinate Error</p> <p><b>0b1101</b> AXI W Decode Error</p> <p><b>0b1110</b> PA out of range Error</p>	RW	0b0000

### 8.3.13.19 por\_mtu\_errcfg

Functions as the Root PAS Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE800

##### Type

RO

### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

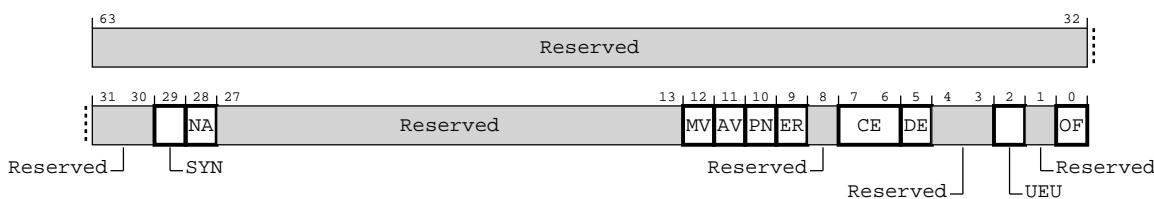
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-800: por\_mtu\_errpfgf**



**Table 8-810: por\_mtu\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update por_mtu_errstatus.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b1</b> Fault injection update por_mtu_errstatus.MV to por_mtu_errpfgctl.MV	RO	0b1
[11]	AV	Address syndrome. <b>0b1</b> Fault injection update por_mtu_errstatus.AV to por_mtu_errpfgctl.AV	RO	0b1
[10]	PN	Poison flag <b>0b1</b> Fault injection update por_mtu_errstatus.PN to por_mtu_errpfgctl.PN	RO	0b1

Bits	Name	Description	Type	Reset
[9]	ER	Error reported flag  <b>0b1</b> Fault injection update por_mtu_errstatus.ER to por_mtu_errfgctl.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> No Corrected error generation.  <b>0b01</b> Non specific Corrected error injection. If por_mtu_errfgctl.CE == 1, update por_mtu_errstatus.CE to 0b10; else, update por_mtu_errstatus.CE to 0b00	RO	0b01
[5]	DE	Deferred error generation.  <b>0b0</b> Deferred error generation not supported	RO	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b1</b> If por_mtu_errfgctl.UEU == 1, update por_mtu_errstatus.UE to 0b1 and por_mtu_errstatus.UET = 0b01; else, update por_mtu_errstatus.UE to 0b0 and por_mtu_errstatus.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b1</b> Fault injection update por_mtu_errstatus.OF to por_mtu_errfgctl.OF	RO	0b1

### 8.3.13.20 por\_mtu\_errfgctl

Functions as the Root PAS Pseudo-fault Generation Control Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE808

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.ras

## Secure group override

por\_mtu\_scr.ras

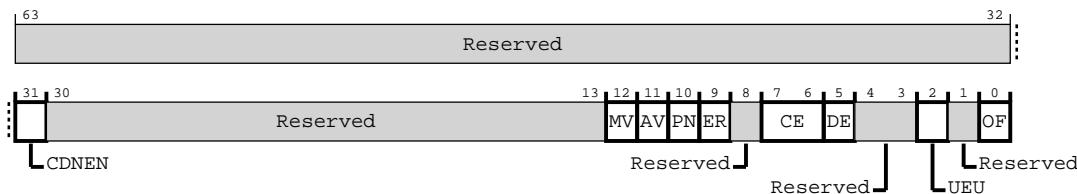
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-801: por\_mtu\_errpfctl**



**Table 8-811: por\_mtu\_errpfctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable.  <b>0b0</b> Countdown disabled.  <b>0b1</b> Error generation counter is set to por_mtu_errpfctl.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b0</b> Fault injection update por_mtu_errstatus.MV to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus.MV to 0b1	RW	0b0
[11]	AV	Address syndrome.  <b>0b0</b> Fault injection update por_mtu_errstatus.AV to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus.AV to 0b1	RW	0b0

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag  <b>0b0</b> Fault injection update por_mtu_errstatus.PN to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus.PN to 0b1	RW	0b0
[9]	ER	Error reported flag  <b>0b0</b> Fault injection update por_mtu_errstatus.ER to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> Non Corrected error is injected. Fault injection update por_mtu_errstatus.CE to 0b00  <b>0b01</b> Non specific Corrected error injection. Fault injection update por_mtu_errstatus.CE to 0b10	RW	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> Deferred error generation not supported	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update por_mtu_errstatus.UE to 0b0 and por_mtu_errstatus.UET = 0b00  <b>0b1</b> Fault injection update por_mtu_errstatus.UE to 0b1 and por_mtu_errstatus.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update por_mtu_errstatus.OF to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus.OF to 0b1	RW	0b0

### 8.3.13.21 por\_mtu\_errcfgcdn

Functions as the Root PAS Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE810

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

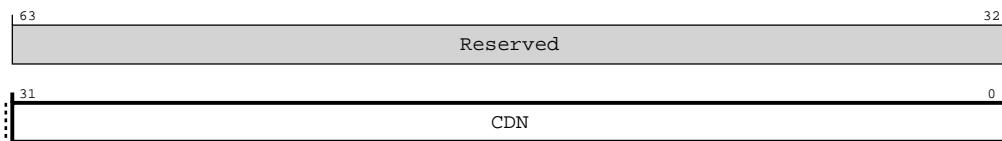
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-802: por\_mtu\_errpfgcdn**



**Table 8-812: por\_mtu\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

## 8.3.13.22 por\_mtu\_errfr\_NS

Functions as the Non-secure PAS error feature register.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE040

### Type

RO

### Reset value

See individual bit resets

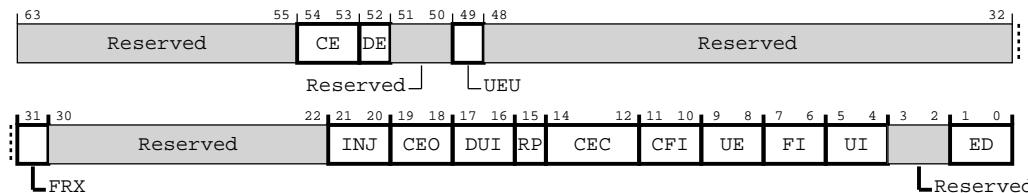
### Usage constraints

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 8-803: por\_mtu\_errfr\_NS**



**Table 8-813: por\_mtu\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording  <b>0b00</b> Corrected Error not supported  <b>0b10</b> Non-specific Corrected Error supported	RO	0b10
[52]	DE	Deferred Error recording  <b>0b0</b> Deferred Error not supported	RO	0b0
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1

Bits	Name	Description	Type	Reset
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension <b>0b1</b> por_mtu_errfr_NS[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors	RO	0b00
[15]	RP	Repeat counter (valid only when por_mtu_errfr_NS.CEC == 0b100.) <b>0b0</b> Invalid <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in por_mtu_errmisco_NS	RO	0b1
[14:12]	CEC	Standard corrected error counter <b>0b000</b> Does not implement standard error counter model <b>0b100</b> Implements a 16-bit Corrected error counter in por_mtu_errmisco_NS	RO	0b100
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_mtu_errctlr_NS.CFI.	RO	0b10
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt for Deferred and Uncorrected errors control <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_mtu_errctlr_NS.FI.	RO	0b10

Bits	Name	Description	Type	Reset
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_mtu_errctlr_NS.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_mtu_errctlr_NS.ED	RO	0b10

### 8.3.13.23 por\_mtu\_errctlr\_NS

Functions as the Non-secure PAS 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

0xE048

##### Type

RW

##### Reset value

See individual bit resets

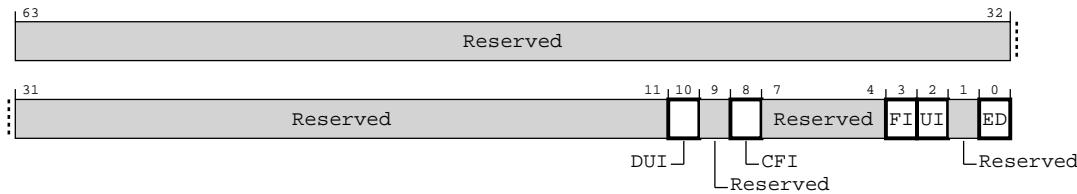
##### Usage constraints

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 8-804: por\_mtu\_errctlr\_NS**



**Table 8-814: por\_mtu\_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 por_mtu_errfr_NS.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_mtu_errfr_NS.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected errors as specified in por_mtu_errfr_NS.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_mtu_errfr_NS.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_mtu_errfr_NS.ED	RW	0b0

### 8.3.13.24 por\_mtu\_errstatus\_NS

Functions as the Non-secure PAS 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

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

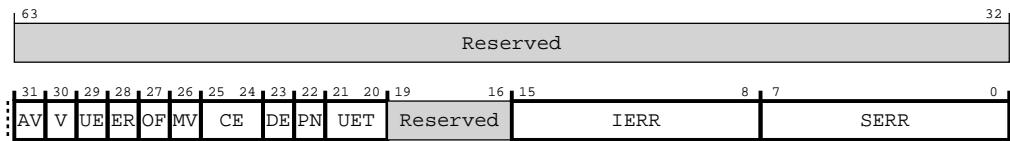
##### Usage constraints

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 8-805: por\_mtu\_errstatus\_NS**



**Table 8-815: por\_mtu\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_mtu_erraddr_NS contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0

Bits	Name	Description	Type	Reset
[26]	MV	por_mtu_errmisc<01>_NS valid  <b>0b1</b> Miscellaneous registers are valid  <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors  <b>0b10</b> At least one corrected error recorded  <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors  <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison  <b>0b1</b> Uncorrected error recorded because a poison value was consumed  <b>0b0</b> Other cases	W1C	0b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0

Bits	Name	Description	Type	Reset
[7:0]	SERR	<p>Architecturally-defined primary error code.</p> <p><b>0x00</b> No error</p> <p><b>0x01</b> CMN implementation defined error. Refer to por_mtu_errmisc1_NS.ERRSRC for error type details.</p> <p><b>0x06</b> ECC error on L3 data</p> <p><b>0x07</b> ECC error on L3/SF Tag</p> <p><b>0x0A</b> Producer write data was poisoned but ACE-Lite does not support poisoned data</p> <p><b>0x0D</b> Illegal address</p> <p><b>0x1A</b> Parity error</p>	W1C	0b0

### 8.3.13.25 por\_mtu\_erraddr\_NS

Functions as the Non-secure PAS error address register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE058

##### Type

RW

##### Reset value

See individual bit resets

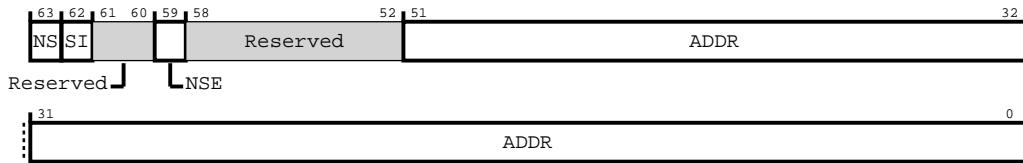
##### Usage constraints

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 8-806: por\_mtu\_erraddr\_NS**



**Table 8-816: por\_mtu\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction  <b>0b1</b> Non-secure transaction  <b>0b0</b> Secure transaction	RW	0b0
[62]	SI	{NSE,NS} valid  <b>0b0</b> PAS field is valid  <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address for Data Request (Data PA). Same PA received from HN.  <b>Note</b> For Errors on TC evictions, this represents address that caused eviction.  <b>Note</b> Address is undefined if errmisc.OPTYPE is Other. For most part, these transactions are set/way based and not PA based.	RW	0b0

### 8.3.13.26 por\_mtu\_errmisc0\_NS

Functions as the Non-secure PAS miscellaneous error register 0. Contains information about corrected errors.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

### Address offset

0xE060

### Type

RW

### Reset value

See individual bit resets

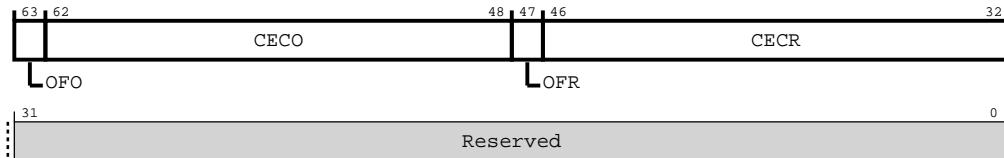
### Usage constraints

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 8-807: por\_mtu\_errmisc0\_NS**



**Table 8-817: por\_mtu\_errmisc0\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	0b0
[62:48]	CECO	Corrected ECC error count	RW	0b0
[47]	OFR	Corrected error counter overflow	RW	0b0
[46:32]	CECR	Corrected ECC error count	RW	0b0
[31:0]	Reserved	Reserved	RO	

### 8.3.13.27 por\_mtu\_errmisc1\_NS

Functions as the Non-secure PAS miscellaneous error register 1. Contains additional information about errors.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

### Address offset

0xE068

## Type

RW

## Reset value

See individual bit resets

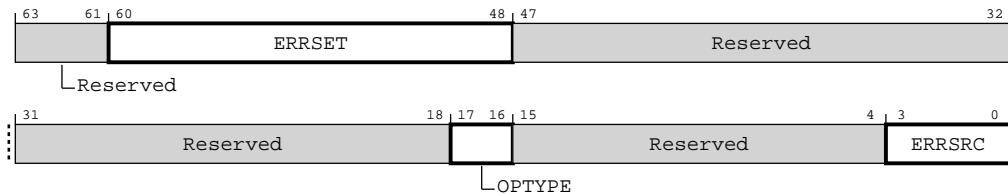
## Usage constraints

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 8-808: por\_mtu\_errmisc1\_NS**



**Table 8-818: por\_mtu\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	TC set address for ECC Single bit error	RW	0b0
[47:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error opcode type <b>0b00</b> Read Type (RD_NO_SNP, PrefetchTgt) <b>0b01</b> Write (WR_NO_SNP) <b>0b10</b> CMO, WR+CMO <b>0b11</b> Other op types	RW	0b00
[15:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	<p>Error source</p> <p><b>0b0001</b> Data single-bit ECC</p> <p><b>0b0010</b> Data double-bit ECC</p> <p><b>0b0011</b> Single-bit ECC overflow</p> <p><b>0b0101</b> Control single-bit ECC</p> <p><b>0b0110</b> Control double-bit ECC</p> <p><b>0b1000</b> AXI AR Subordinate Error</p> <p><b>0b1001</b> AXI AR Decode Error</p> <p><b>0b1010</b> AXI AR Poison Error</p> <p><b>0b1011</b> AXI AR Datachk Error</p> <p><b>0b1100</b> AXI W Subordinate Error</p> <p><b>0b1101</b> AXI W Decode Error</p> <p><b>0b1110</b> PA out of range Error</p>	RW	0b0000

### 8.3.13.28 por\_mtu\_errcfg\_NS

Functions as the Non-secure PAS Pseudo-fault Generation Feature Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE840

##### Type

RO

## Reset value

See individual bit resets

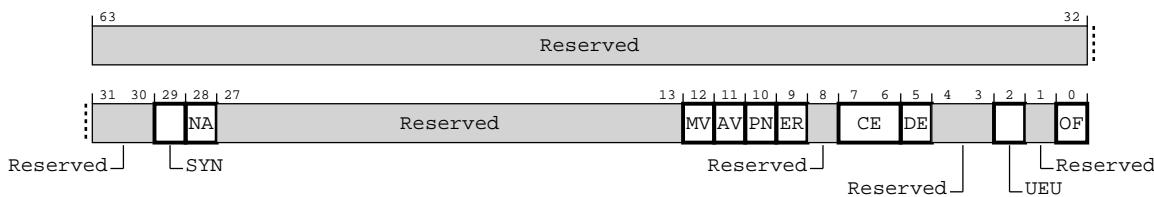
## Usage constraints

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 8-809: por\_mtu\_errpfgf\_NS**



**Table 8-819: por\_mtu\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update por_mtu_errstatus_NS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b1</b> Fault injection update por_mtu_errstatus_NS.MV to por_mtu_errpfgctl_NS.MV	RO	0b1
[11]	AV	Address syndrome. <b>0b1</b> Fault injection update por_mtu_errstatus.AV to por_mtu_errpfgctl_NS.AV	RO	0b1
[10]	PN	Poison flag <b>0b1</b> Fault injection update por_mtu_errstatus_NS.PN to por_mtu_errpfgctl_NS.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> Fault injection update por_mtu_errstatus_NS.ER to por_mtu_errpfgctl_NS.ER	RO	0b1
[8]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> No Corrected error generation.</p> <p><b>0b01</b> Non specific Corrected error injection. If por_mtu_errpfctl_NS.CE == 1, update por_mtu_errstatus_NS.CE to 0b10; else, update por_mtu_errstatus_NS.CE to 0b00</p>	RO	0b01
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Deferred error generation not supported</p>	RO	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b1</b> If por_mtu_errpfctl_NS.UEU == 1, update por_mtu_errstatus_NS.UET to 0b1 and por_mtu_errstatus_NS.UET = 0b01; else, update por_mtu_errstatus_NS.UET to 0b0 and por_mtu_errstatus_NS.UET = 0b00</p>	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	<p>Overflow flag.</p> <p><b>0b1</b> Fault injection update por_mtu_errstatus_NS.OF to por_mtu_errpfctl_NS.OF</p>	RO	0b1

### 8.3.13.29 por\_mtu\_errpfctl\_NS

Functions as the Non-secure PAS Pseudo-fault Generation Control Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE848

##### Type

RW

##### Reset value

See individual bit resets

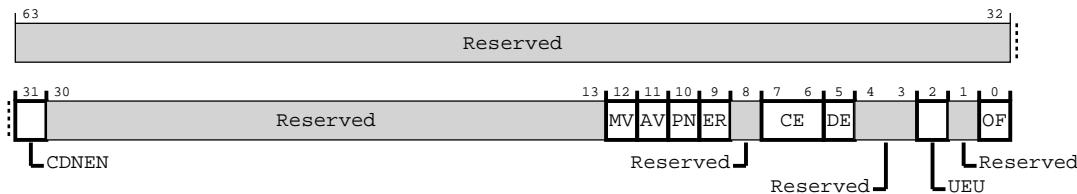
##### Usage constraints

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 8-810: por\_mtu\_errpfctl\_NS**



**Table 8-820: por\_mtu\_errpfctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable.  <b>0b0</b> Countdown disabled.  <b>0b1</b> Error generation counter is set to por_mtu_errpfctl_NS.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.MV to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.MV to 0b1	RW	0b0
[11]	AV	Address syndrome.  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.AV to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.AV to 0b1	RW	0b0
[10]	PN	Poison flag  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.PN to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.PN to 0b1	RW	0b0
[9]	ER	Error reported flag  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.ER to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[7:6]	CE	Corrected Error generation.  <b>0b00</b> Non Corrected error is injected. Fault injection update por_mtu_errstatus_NS.CE to 0b00  <b>0b01</b> Non specific Corrected error injection. Fault injection update por_mtu_errstatus_NS.CE to 0b10	RW	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> Deferred error generation not supported	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.UE to 0b0 and por_mtu_errstatus_NS.UET = 0b00  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.UE to 0b1 and por_mtu_errstatus_NS.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update por_mtu_errstatus_NS.OF to 0b0  <b>0b1</b> Fault injection update por_mtu_errstatus_NS.OF to 0b1	RW	0b0

### 8.3.13.30 por\_mtu\_errfgcdn\_NS

Functions as the Non-secure PAS Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE850

##### Type

RW

##### Reset value

See individual bit resets

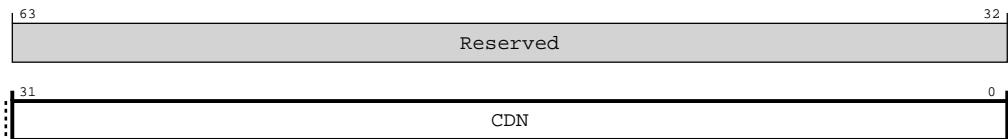
#### Usage constraints

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 8-811: por\_mtu\_errpfcdn\_NS**



**Table 8-821: por\_mtu\_errpfcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.13.31 por\_mtu\_errcapctl

Functions as the error Capture Control Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xED00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.ras

##### Secure group override

por\_mtu\_scr.ras

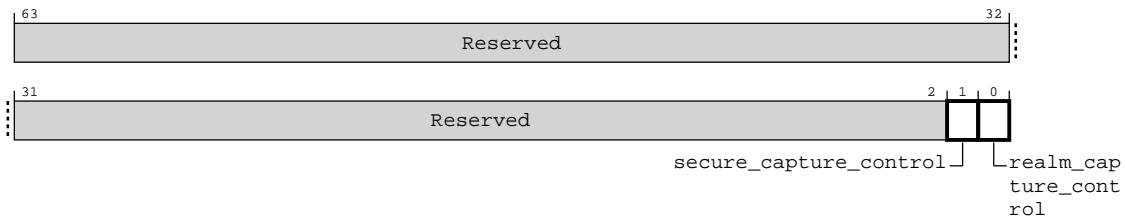
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-812: por\_mtu\_errcapctl**



**Table 8-822: por\_mtu\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control  <b>0b0</b> Transaction with secure PAS captured in root error record  <b>0b1</b> Transaction with secure PAS captured in non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control  <b>0b0</b> Transaction with realm PAS captured in root error record  <b>0b1</b> Transaction with realm PAS captured in non-secure error record	RW	0b0

### 8.3.13.32 por\_mtu\_errgsr

Functions as Error Group Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEE00

##### Type

RO

### Reset value

See individual bit resets

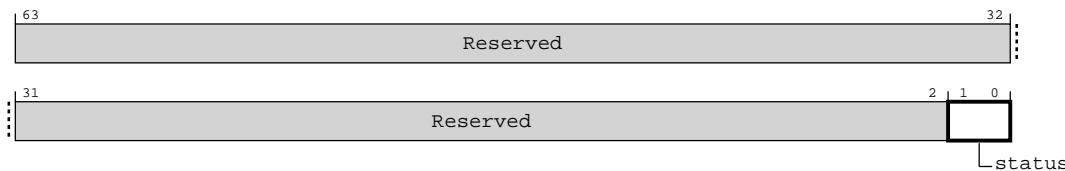
### Usage constraints

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 8-813: por\_mtu\_errgsr**



**Table 8-823: por\_mtu\_errgsr attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1:0]	status	Read-only copy of ERR<n>STATUS.V	RO	0b0

### 8.3.13.33 por\_mtu\_errildr

Functions as the implementation identification register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEEE10

#### Type

RO

### Reset value

See individual bit resets

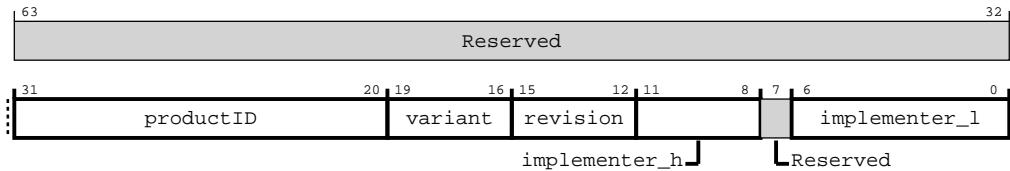
### Usage constraints

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 8-814: por\_mtu\_errildr**



**Table 8-824: por\_mtu\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.13.34 por\_mtu\_errdevaff

Functions as the device affinity register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFA8

##### Type

RO

##### Reset value

See individual bit resets

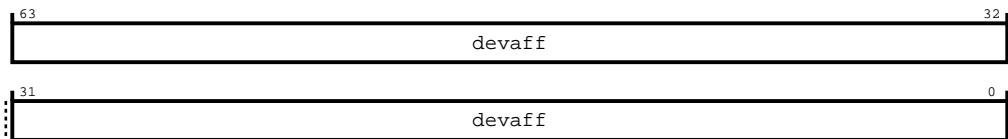
#### Usage constraints

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 8-815: por\_mtu\_errdevaff**



**Table 8-825: por\_mtu\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.13.35 por\_mtu\_errdevarch

Functions as the device architecture register.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xEF8B

### Type

RO

### Reset value

See individual bit resets

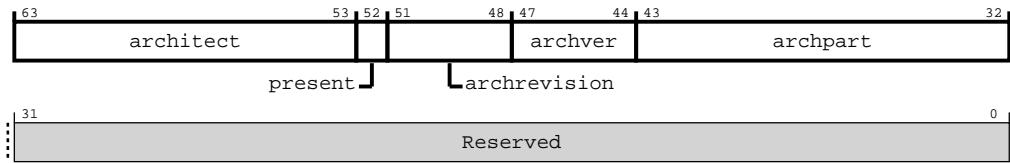
### Usage constraints

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 8-816: por\_mtu\_errdevarch**



**Table 8-826: por\_mtu\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

### 8.3.13.36 por\_mtu\_errdevid

Functions as the device configuration register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFC8

##### Type

RO

##### Reset value

See individual bit resets

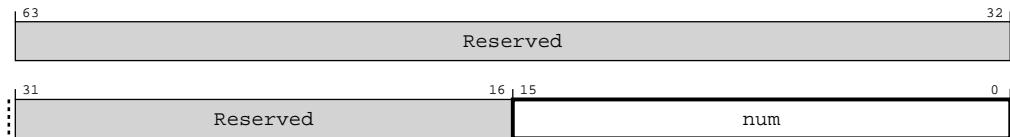
##### Usage constraints

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 8-817: por\_mtu\_errdevid**



**Table 8-827: por\_mtu\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

### 8.3.13.37 por\_mtu\_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

0xEFDO

##### Type

RO

##### Reset value

See individual bit resets

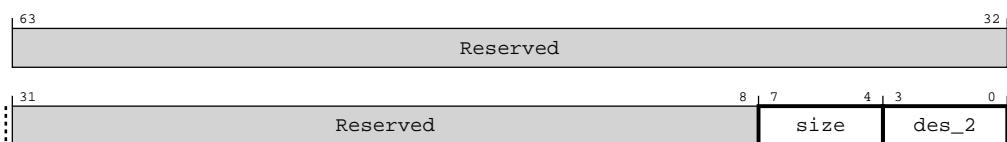
##### Usage constraints

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 8-818: por\_mtu\_errpidr45**



**Table 8-828: por\_mtu\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

### 8.3.13.38 por\_mtu\_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

0xEFE0

##### Type

RO

##### Reset value

See individual bit resets

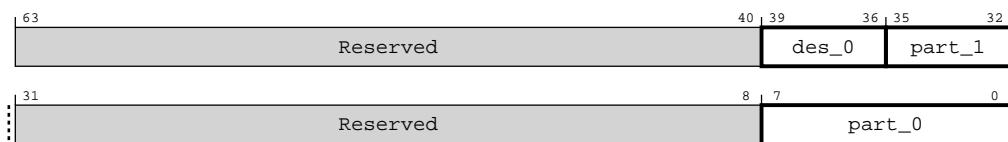
##### Usage constraints

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 8-819: por\_mtu\_errpidr01**



**Table 8-829: por\_mtu\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

### 8.3.13.39 por\_mtu\_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

0xEFE8

##### Type

RO

##### Reset value

See individual bit resets

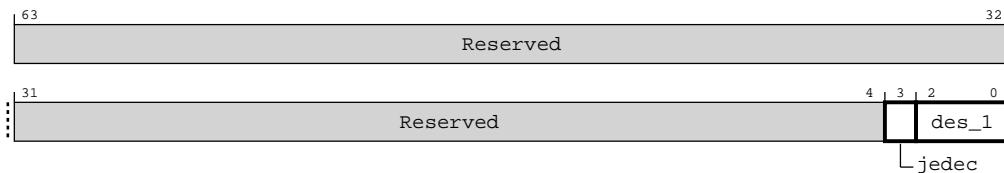
##### Usage constraints

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 8-820: por\_mtu\_errpidr23**



**Table 8-830: por\_mtu\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

### 8.3.13.40 por\_mtu\_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

0xEFF0

##### Type

RO

##### Reset value

See individual bit resets

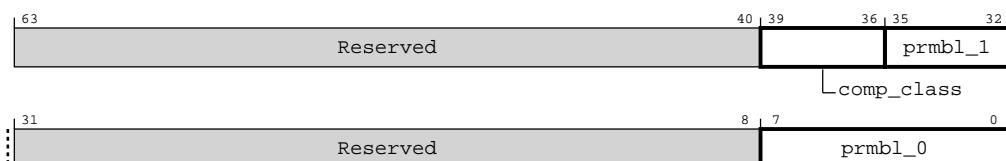
##### Usage constraints

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 8-821: por\_mtu\_errcidr01**



**Table 8-831: por\_mtu\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmb1_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_0	PRMBL_0	RO	0xD

### 8.3.13.41 por\_mtu\_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

0xEFF8

##### Type

RO

##### Reset value

See individual bit resets

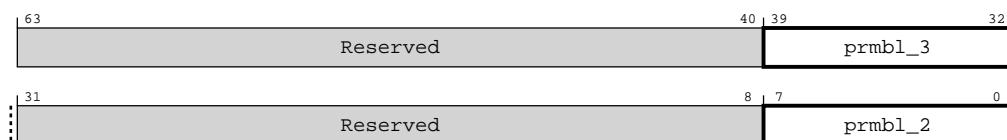
#### Usage constraints

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 8-822: por\_mtu\_errcidr23**



**Table 8-832: por\_mtu\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.13.42 por\_mtu\_err\_inj

Enables error injection and setup. When enabled for a given PA and PAS (NSE/NS), MTU returns an error interrupt which emulates a TC double-bit data ECC error. This feature enables software

to test the error handler. The error is reported for cacheable read access with Tag Op Transfer for which TC hit. No error is reported for cacheable read access for which TC miss.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE030

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mtu\_rcr.ras

### Secure group override

por\_mtu\_scr.ras

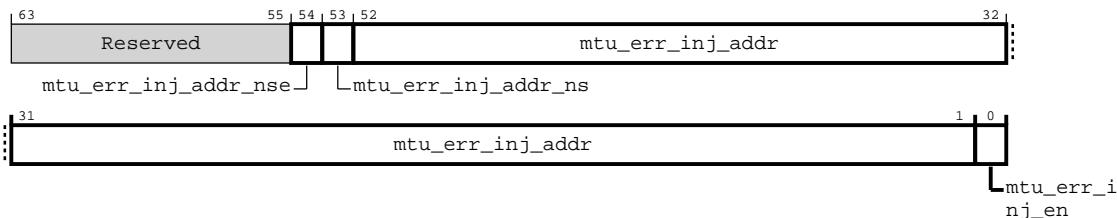
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.ras bit and por\_mtu\_rcr.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 8-823: por\_mtu\_err\_inj**



**Table 8-833: por\_mtu\_err\_inj attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	
[54]	mtu_err_inj_addr_nse	Address NSE used to match for error injection	RW	0b0
[53]	mtu_err_inj_addr_ns	Address NS used to match for error injection	RW	0b0

Bits	Name	Description	Type	Reset
[52:1]	mtu_err_inj_addr	Physical Address used to match for error injection	RW	0b0
[0]	mtu_err_inj_en	Enables error injection and report	RW	0b0

### 8.3.13.43 por\_mtu\_cfg\_tc\_dbgrd

Controls access modes for TC data and TC Control debug read.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xB80

##### Type

WO

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.tc\_dbgrd

##### Secure group override

por\_mtu\_scr.tc\_dbgrd

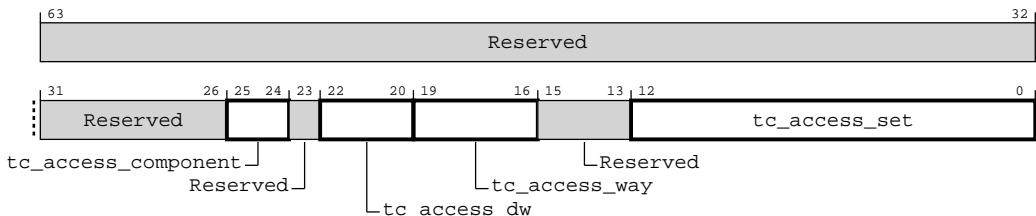
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.tc\_dbgrd bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.tc\_dbgrd bit and por\_mtu\_rcr.tc\_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 8-824: por\_mtu\_cfg\_tc\_dbgrd**



**Table 8-834: por\_mtu\_cfg\_tc\_dbgrd attributes**

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	
[25:24]	tc_access_component	Specifies TC Data/Control array debug read <b>0b01</b> TC data read <b>0b10</b> TC control read	WO	0b10
[23]	Reserved	Reserved	RO	
[22:20]	tc_access_dw	64-bit chunk address for TC data debug read access	WO	0x0
[19:16]	tc_access_way	Way address for TC debug read access	WO	0x0
[15:13]	Reserved	Reserved	RO	
[12:0]	tc_access_set	Set address for TC debug read access	WO	0x0

### 8.3.13.44 por\_mtu\_tc\_cache\_access\_tc\_ctl

Contains TC Control debug read data bits

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xB88

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.tc\_dbgrd

##### Secure group override

por\_mtu\_scr.tc\_dbgrd

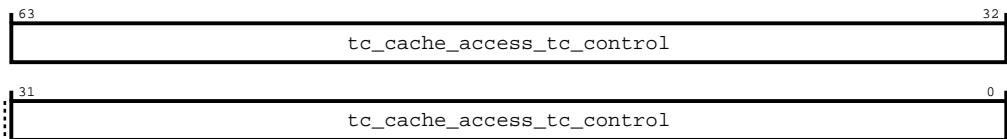
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.tc\_dbgrd bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.tc\_dbgrd bit and por\_mtu\_rcr.tc\_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 8-825: por\_mtu\_tc\_cache\_access\_tc\_ctl**



**Table 8-835: por\_mtu\_tc\_cache\_access\_tc\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:0]	tc_cache_access_tc_control	TC Control debug read data	RO	0x0

### 8.3.13.45 por\_mtu\_tc\_cache\_access\_tc\_data

Contains TC data RAM debug read data.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xB98

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_mtu\_rcr.tc\_dbgrd

##### Secure group override

por\_mtu\_scr.tc\_dbgrd

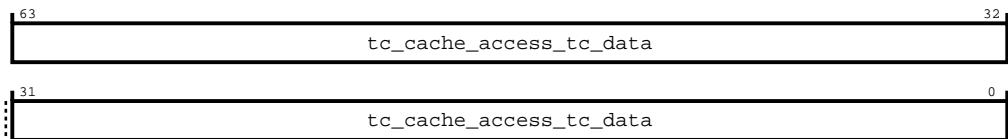
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mtu\_scr.tc\_dbgrd bit is set, Secure accesses to this register are permitted. If both the por\_mtu\_scr.tc\_dbgrd bit and por\_mtu\_rcr.tc\_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 8-826: por\_mtu\_tc\_cache\_access\_tc\_data**



**Table 8-836: por\_mtu\_tc\_cache\_access\_tc\_data attributes**

Bits	Name	Description	Type	Reset
[63:0]	tc_cache_access_tc_data	TC data RAM debug read data	RO	0x0

### 8.3.13.46 por\_mtu\_pmu\_event\_sel

Specifies the PMU event to be counted.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xD900

### Type

RW

### Reset value

See individual bit resets

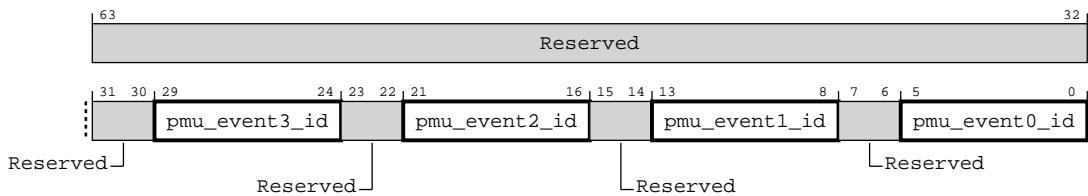
### Usage constraints

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 8-827: por\_mtu\_pmu\_event\_sel**



**Table 8-837: por\_mtu\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29:24]	pmu_event3_id	MTU PMU Event 3 select; see pmu_event0_id for encodings	RW	0b0
[23:22]	Reserved	Reserved	RO	
[21:16]	pmu_event2_id	MTU PMU Event 2 select; see pmu_event0_id for encodings	RW	0b0
[15:14]	Reserved	Reserved	RO	
[13:8]	pmu_event1_id	MTU PMU Event 1 select; see pmu_event0_id for encodings	RW	0b0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>MTU PMU Event 0 select</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> PMU_MTU_TC_LOOKUP_EVENT; Count total cache lookup requests.</p> <p><b>0x02</b> PMU_MTU_TC_FILL_EVENT; Count total number of tag cache allocation (Dirty or Clean) requests.</p> <p><b>0x03</b> PMU_MTU_TC_MISS_EVENT; Count total cache miss responses.</p> <p><b>0x04</b> PMU_MTU_TDB_FORWARD_EVENT; Count total number of requests that got TDB forwarded data.</p> <p><b>0x05</b> PMU_MTU_TCQ_HAZARD_EVENT; Count number of incoming requests hazarding against pending TCQ requests.</p> <p><b>0x06</b> PMU_MTU_TCQ_RD_ALLOC_EVENT; Count number of read requests allocated in TCQ. This includes Read &amp; Write_Match.</p> <p><b>0x07</b> PMU_MTU_TCQ_WR_ALLOC_EVENT; Count number of write requests allocated in TCQ. This includes Write_Update.</p> <p><b>0x08</b> PMU_MTU_TCQ_CMO_ALLOC_EVENT; Count number of CMO requests allocated in TCQ. This includes CMOs &amp; wr+CMO.</p> <p><b>0x09</b> PMU_MTU_AXI_RD_REQ_EVENT; Count number of read requests sent out on AXI.</p> <p><b>0x0A</b> PMU_MTU_AXI_WR_REQ_EVENT; Count number of write requests sent out on AXI.</p> <p><b>0x0B</b> PMU_MTU_TCQ_OCCUPANCY_CNT_OVERFLOW_EVENT; TCQ tracker occupancy count overflow.</p> <p><b>0x0C</b> PMU_MTU_TDB_OCCUPANCY_CNT_OVERFLOW_EVENT; TDB occupancy count overflow.</p> <p><b>NOTE</b> All other encodings are reserved.</p>	RW	0b0

### 8.3.14 MXP register summary

The following table describes the registers for the relevant component.

**Table 8-838: por\_mxp\_cfg register summary**

Offset	Name	Type	Description
0x0	por_mxp_node_info	RO	<a href="#">por_mxp_node_info</a>
0x8 : 0x30	por_mxp_device_port_connect_info_p0-5	RO	<a href="#">por_mxp_device_port_connect_info_p0-5</a>
0x38	por_mxp_mesh_port_connect_info_east	RO	<a href="#">por_mxp_mesh_port_connect_info_east</a>
0x40	por_mxp_mesh_port_connect_info_north	RO	<a href="#">por_mxp_mesh_port_connect_info_north</a>
0x48 : 0x70	por_mxp_device_port_connect_ldid_info_p0-5	RO	<a href="#">por_mxp_device_port_connect_ldid_info_p0-5</a>
0x80	por_mxp_child_info	RO	<a href="#">por_mxp_child_info</a>
0x100 : 0x1F8	por_mxp_child_pointer_0-31	RO	<a href="#">por_mxp_child_pointer_0-31</a>
0x900 : 0x950	por_mxp_p0-5_info	RO	<a href="#">por_mxp_p0-5_info</a>
0x908 : 0x958	por_mxp_p0-5_info_1	RO	<a href="#">por_mxp_p0-5_info_1</a>
0x960	por_dtm_unit_info	RO	<a href="#">por_dtm_unit_info</a>
0x968 : 0x978	por_dtm_unit_info_dt1-3	RO	<a href="#">por_dtm_unit_info_dt1-3</a>
0x980	por_mxp_scr	RW	<a href="#">por_mxp_scr</a>
0x988	por_mxp_rcr	RW	<a href="#">por_mxp_rcr</a>
0xA00	por_mxp_aux_ctl	RW	<a href="#">por_mxp_aux_ctl</a>
0xA08	por_mxp_device_port_ctl	RW	<a href="#">por_mxp_device_port_ctl</a>
0xA10 : 0xA38	por_mxp_p0-5_mpam_override	RW	<a href="#">por_mxp_p0-5_mpam_override</a>
0xA40 : 0xA68	por_mxp_p0-5_ldid_override	RW	<a href="#">por_mxp_p0-5_ldid_override</a>
0xA70	por_mxp_device_port_disable	RW	<a href="#">por_mxp_device_port_disable</a>
0xA78	por_mxp_cfg_ctl	RW	<a href="#">por_mxp_cfg_ctl</a>
0xA80 : 0xB60	por_mxp_p0-5_qos_control	RW	<a href="#">por_mxp_p0-5_qos_control</a>
0xA88 : 0xB68	por_mxp_p0-5_qos_lat_tgt	RW	<a href="#">por_mxp_p0-5_qos_lat_tgt</a>
0xA90 : 0xB70	por_mxp_p0-5_qos_lat_scale	RW	<a href="#">por_mxp_p0-5_qos_lat_scale</a>
0xA98 : 0xB78	por_mxp_p0-5_qos_lat_range	RW	<a href="#">por_mxp_p0-5_qos_lat_range</a>
0xA000	por_mxp_pmu_event_sel	RW	<a href="#">por_mxp_pmu_event_sel</a>
0xE000	por_mxp_errfr	RO	<a href="#">por_mxp_errfr</a>
0xE008	por_mxp_errctlr	RW	<a href="#">por_mxp_errctlr</a>
0xE010	por_mxp_errstatus	W1C	<a href="#">por_mxp_errstatus</a>
0xE018	por_mxp_erraddr	RW	<a href="#">por_mxp_erraddr</a>
0xE028	por_mxp_errmisc1	RW	<a href="#">por_mxp_errmisc1</a>
0xE930 : 0xE968	por_mxp_p0-5_byte_par_err_inj	WO	<a href="#">por_mxp_p0-5_byte_par_err_inj</a>
0xE800	por_mxp_errpfgf	RO	<a href="#">por_mxp_errpfgf</a>
0xE808	por_mxp_errpfgctl	RW	<a href="#">por_mxp_errpfgctl</a>
0xE810	por_mxp_errpfgcdn	RW	<a href="#">por_mxp_errpfgcdn</a>
0xE040	por_mxp_errfr_NS	RO	<a href="#">por_mxp_errfr_NS</a>
0xE048	por_mxp_errctlr_NS	RW	<a href="#">por_mxp_errctlr_NS</a>
0xE050	por_mxp_errstatus_NS	W1C	<a href="#">por_mxp_errstatus_NS</a>

Offset	Name	Type	Description
0xE058	por_mxp_erraddr_NS	RW	<a href="#">por_mxp_erraddr_NS</a>
0xE068	por_mxp_errmisc1_NS	RW	<a href="#">por_mxp_errmisc1_NS</a>
0xE840	por_mxp_errpfgf_NS	RO	<a href="#">por_mxp_errpfgf_NS</a>
0xE848	por_mxp_errpfctl_NS	RW	<a href="#">por_mxp_errpfctl_NS</a>
0xE850	por_mxp_errpgcdn_NS	RW	<a href="#">por_mxp_errpgcdn_NS</a>
0xED00	por_mxp_errcapctl	RW	<a href="#">por_mxp_errcapctl</a>
0xEE00	por_mxp_errgsr	RO	<a href="#">por_mxp_errgsr</a>
0xEE10	por_mxp_eriidr	RO	<a href="#">por_mxp_eriidr</a>
0xEFA8	por_mxp_errdevaff	RO	<a href="#">por_mxp_errdevaff</a>
0xEF88	por_mxp_errdevarch	RO	<a href="#">por_mxp_errdevarch</a>
0xEFC8	por_mxp_errdevid	RO	<a href="#">por_mxp_errdevid</a>
0xEF00	por_mxp_errpidr45	RO	<a href="#">por_mxp_errpidr45</a>
0xEFE0	por_mxp_errpidr01	RO	<a href="#">por_mxp_errpidr01</a>
0xEFE8	por_mxp_errpidr23	RO	<a href="#">por_mxp_errpidr23</a>
0xEFF0	por_mxp_errcidr01	RO	<a href="#">por_mxp_errcidr01</a>
0xEFF8	por_mxp_errcidr23	RO	<a href="#">por_mxp_errcidr23</a>
0x1000 : 0x1070	por_mxp_p0-5_syscoreq_ctl	RW	<a href="#">por_mxp_p0-5_syscoreq_ctl</a>
0x1008 : 0x1078	por_mxp_p0-5_syscoack_status	RO	<a href="#">por_mxp_p0-5_syscoack_status</a>
0xA100	por_dtm_control	RW	<a href="#">por_dtm_control</a>
0xA118	por_dtm_fifo_entry_ready	W1C	<a href="#">por_dtm_fifo_entry_ready</a>
0xA120 : 0xA180	por_dtm_fifo_entry0-3_0	RO	<a href="#">por_dtm_fifo_entry0-3_0</a>
0xA128 : 0xA188	por_dtm_fifo_entry0-3_1	RO	<a href="#">por_dtm_fifo_entry0-3_1</a>
0xA130 : 0xA190	por_dtm_fifo_entry0-3_2	RO	<a href="#">por_dtm_fifo_entry0-3_2</a>
0xA138 : 0xA198	por_dtm_fifo_entry0-3_3	RO	<a href="#">por_dtm_fifo_entry0-3_3</a>
0xA1A0 : 0xA1E8	por_dtm_wp0-3_config	RW	<a href="#">por_dtm_wp0-3_config</a>
0xA1A8 : 0xA1F0	por_dtm_wp0-3_val	RW	<a href="#">por_dtm_wp0-3_val</a>
0xA1B0 : 0xA1F8	por_dtm_wp0-3_mask	RW	<a href="#">por_dtm_wp0-3_mask</a>
0xA200	por_dtm_pmsicr	RW	<a href="#">por_dtm_pmsicr</a>
0xA208	por_dtm_pmsirr	RW	<a href="#">por_dtm_pmsirr</a>
0xA210	por_dtm_pmu_config	RW	<a href="#">por_dtm_pmu_config</a>
0xA220	por_dtm_pmevcnt	RW	<a href="#">por_dtm_pmevcnt</a>
0xA240	por_dtm_pmevcntsr	RW	<a href="#">por_dtm_pmevcntsr</a>
0xA300 : 0xA700	por_dtm_control_dt1-3	RW	<a href="#">por_dtm_control_dt1-3</a>
0xA318 : 0xA718	por_dtm_fifo_entry_ready_dt1-3	W1C	<a href="#">por_dtm_fifo_entry_ready_dt1-3</a>
0xA320 : 0xA900	por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1	RO	<a href="#">por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1</a>
0xA328 : 0xA908	por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1	RO	<a href="#">por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1</a>
0xA330 : 0xA910	por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1	RO	<a href="#">por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1</a>
0xA338 : 0xA918	por_dtm_fifo_entry0-11%4_3_dt(0-11/4)+1	RO	<a href="#">por_dtm_fifo_entry0-11%4_3_dt(0-11/4)+1</a>
0xA3A0 : 0xA968	por_dtm_wp0-11%4_config_dt(0-11/4)+1	RW	<a href="#">por_dtm_wp0-11%4_config_dt(0-11/4)+1</a>
0xA3A8 : 0xA970	por_dtm_wp0-11%4_val_dt(0-11/4)+1	RW	<a href="#">por_dtm_wp0-11%4_val_dt(0-11/4)+1</a>

Offset	Name	Type	Description
0xA3B0 : 0xA978	por_dtm_wp0-11%4_mask_dt(0-11/4)+1	RW	<a href="#">por_dtm_wp0-11%4_mask_dt(0-11/4)+1</a>
0xA400 : 0xA800	por_dtm_pmsicr_dt1-3	RW	<a href="#">por_dtm_pmsicr_dt1-3</a>
0xA408 : 0xA808	por_dtm_pmsirr_dt1-3	RW	<a href="#">por_dtm_pmsirr_dt1-3</a>
0xA410 : 0xA810	por_dtm_pmu_config_dt1-3	RW	<a href="#">por_dtm_pmu_config_dt1-3</a>
0xA420 : 0xA820	por_dtm_pmevcnt_dt1-3	RW	<a href="#">por_dtm_pmevcnt_dt1-3</a>
0xA440 : 0xA840	por_dtm_pmevcntsr_dt1-3	RW	<a href="#">por_dtm_pmevcntsr_dt1-3</a>
0xC00 : 0xC78	por_mxp_multi_mesh_chn_sel_0-15	RW	<a href="#">por_mxp_multi_mesh_chn_sel_0-15</a>
0xC00 + 0x80	por_mxp_multi_mesh_chn_ctrl	RW	<a href="#">por_mxp_multi_mesh_chn_ctrl</a>
0xC90 : 0xCC8	por_mxp_xy_override_sel_0-7	RW	<a href="#">por_mxp_xy_override_sel_0-7</a>
0xCD0 : 0xDB0	por_mxp_p0-5_pa2setaddr_slc	RW	<a href="#">por_mxp_p0-5_pa2setaddr_slc</a>
0xCD8 : 0xDB8	por_mxp_p0-5_pa2setaddr_sf	RW	<a href="#">por_mxp_p0-5_pa2setaddr_sf</a>
0xCE0 : 0xDC0	por_mxp_p0-5_pa2setaddr_flex_slc	RW	<a href="#">por_mxp_p0-5_pa2setaddr_flex_slc</a>
0xCE8 : 0xDC8	por_mxp_p0-5_pa2setaddr_flex_sf	RW	<a href="#">por_mxp_p0-5_pa2setaddr_flex_sf</a>
0xE00 : 0xE38	por_mxp_p0-5_datasource_ctl	RW	<a href="#">por_mxp_p0-5_datasource_ctl</a>

### 8.3.14.1 por\_mxp\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

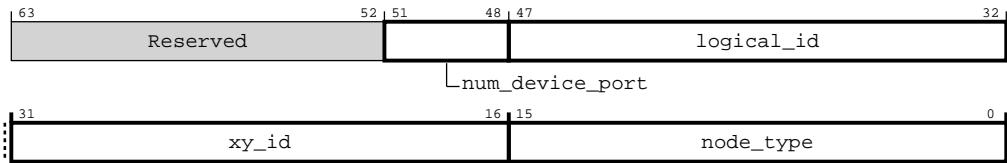
##### Usage constraints

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 8-828: por\_mxp\_node\_info**



**Table 8-839: por\_mxp\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:48]	num_device_port	Number of device ports attached to the MXP. Mesh config = (1x1)? Max. of 6 Device Ports are supported : Max. of 4 Device Ports are supported	RO	Configuration dependent
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	xy_id	Identifies (X,Y) location of XP within the mesh  <b>NOTE</b> The (X,Y) location is specified following the node ID format as defined in Node ID mapping section, with the bottom 3 bits, corresponding to port ID and device ID, set to 0. Bits 31:11 must always be set to 0. The range of bits representing the (X,Y) location varies for different node ID formats.	RO	0x0000
[15:0]	node_type	CMN node type identifier	RO	0x0006

### 8.3.14.2 por\_mxp\_device\_port\_connect\_info\_p0-5

There are 6 iterations of this register. The index ranges from 0 to 5. Contains device port connection information for port #{{index}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x8 + #{{8\*index}}

##### Type

RO

##### Reset value

See individual bit resets

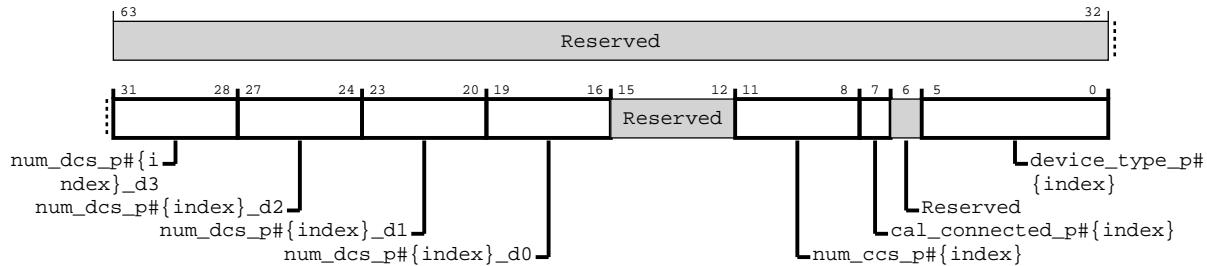
##### Usage constraints

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 8-829: por\_mxp\_device\_port\_connect\_info\_p0-5**



**Table 8-840: por\_mxp\_device\_port\_connect\_info\_p0-5 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	num_dcs_p#{index}_d3	Number of device credited slices connected to port #{index} device 3 <b>Allowed values</b> 0-4	RO	Configuration dependent
[27:24]	num_dcs_p#{index}_d2	Number of device credited slices connected to port #{index} device 2 <b>Allowed values</b> 0-4	RO	Configuration dependent
[23:20]	num_dcs_p#{index}_d1	Number of device credited slices connected to port #{index} device 1 <b>Allowed values</b> 0-4	RO	Configuration dependent
[19:16]	num_dcs_p#{index}_d0	Number of device credited slices connected to port #{index} device 0 <b>Allowed values</b> 0-4	RO	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:8]	num_ccs_p#{index}	Number of CAL credited slices connected to port #{index} <b>Allowed values</b> 0-2	RO	Configuration dependent
[7]	cal_connected_p#{index}	When set, CAL is connected on port #{index} <b>Allowed values</b> 0-1	RO	Configuration dependent
[6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	device_type_p#{index}	Connected device type <b>0b000000</b> Reserved <b>0b000001</b> RN-I <b>0b000010</b> RN-D <b>0b000011</b> Reserved <b>0b000100</b> RN-F_CHIB <b>0b000101</b> RN-F_CHIB_ESAM <b>0b001000</b> HN-T <b>0b001001</b> HN-I <b>0b001010</b> HN-D <b>0b001011</b> HN-P <b>0b001100</b> SN-F_CHIC <b>0b001101</b> SBSX <b>0b001110</b> HN-F <b>0b001111</b> SN-F_CHIE <b>0b010000</b> SN-F_CHID <b>0b010001</b> CXHA <b>0b010010</b> CXRA	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[5:0]	device_type_p#{index}	Connected device type <b>0b010100</b> RN-F_CHID <b>0b010101</b> RN-F_CHID_ESAM <b>0b010110</b> RN-F_CHIC <b>0b010111</b> RN-F_CHIC_ESAM <b>0b011000</b> RN-F_CHIE <b>0b011001</b> RN-F_CHIE_ESAM <b>0b011010</b> HN-S <b>0b011011</b> Reserved <b>0b011100</b> MTSX <b>0b011101</b> HN-V <b>0b011110</b> CCG <b>0b100000</b> RN-F_CHIF <b>0b100001</b> RN-F_CHIF_ESAM <b>0b100010</b> SN-F_CHIF <b>0b100100</b> RN-F_CHIG_ESAM <b>0b100101</b> SN-F_CHIG <b>0b100110 - 6'111111</b> Reserved	RO	Configuration dependent

### 8.3.14.3 por\_mxp\_mesh\_port\_connect\_info\_east

Contains port connection information for East port.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x38

### Type

RO

### Reset value

See individual bit resets

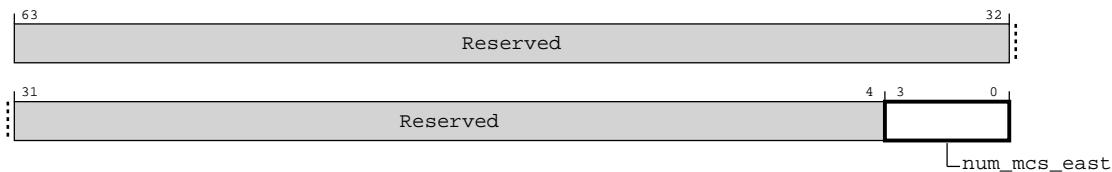
### Usage constraints

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 8-830: por\_mxp\_mesh\_port\_connect\_info\_east**



**Table 8-841: por\_mxp\_mesh\_port\_connect\_info\_east attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_east	Number of mesh credited slices connected to East port <b>Allowed values</b> 0-4	RO	Configuration dependent

## 8.3.14.4 por\_mxp\_mesh\_port\_connect\_info\_north

Contains port connection information for North port.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x40

### Type

RO

### Reset value

See individual bit resets

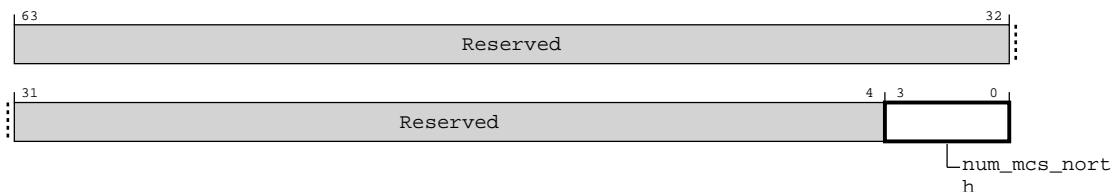
### Usage constraints

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 8-831: por\_mxp\_mesh\_port\_connect\_info\_north**



**Table 8-842: por\_mxp\_mesh\_port\_connect\_info\_north attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_north	Number of mesh credited slices connected to North port  <b>Allowed values</b> 0-4	RO	Configuration dependent

### 8.3.14.5 por\_mxp\_device\_port\_connect\_ldid\_info\_p0-5

There are 6 iterations of this register. The index ranges from 0 to 5. Contains LDID information for devices connected to port #{{index}}. Valid only for RN-Fs

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

$0x48 + \#(8 * \text{index})$

### Type

RO

### Reset value

See individual bit resets

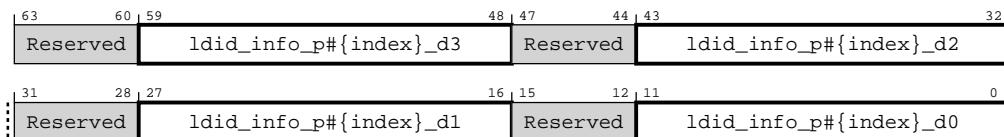
### Usage constraints

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 8-832: por\_mxp\_device\_port\_connect\_ldid\_info\_p0-5**



**Table 8-843: por\_mxp\_device\_port\_connect\_ldid\_info\_p0-5 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	ldid_info_p#{index}_d3	LDID value of the device connected to port P#{index}_D3	RO	Configuration dependent
[47:44]	Reserved	Reserved	RO	-
[43:32]	ldid_info_p#{index}_d2	LDID value of the device connected to port P#{index}_D2	RO	Configuration dependent
[31:28]	Reserved	Reserved	RO	-
[27:16]	ldid_info_p#{index}_d1	LDID value of the device connected to port P#{index}_D1	RO	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:0]	ldid_info_p#{index}_d0	LDID value of the device connected to port P#{index}_D0	RO	Configuration dependent

### 8.3.14.6 por\_mxp\_child\_info

Provides component child identification information.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

### Address offset

0x80

### Type

RO

### Reset value

See individual bit resets

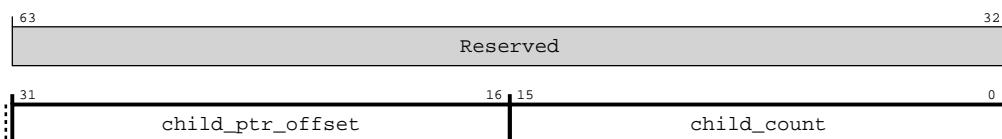
### Usage constraints

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 8-833: por\_mxp\_child\_info**



**Table 8-844: por\_mxp\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x100
[15:0]	child_count	Number of child nodes; used in discovery process	RO	Configuration dependent

### 8.3.14.7 por\_mxp\_child\_pointer\_0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Contains base address of the configuration subordinate for child #{{index}}.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0x100 + #{8\*index}

## Type

RO

## Reset value

See individual bit resets

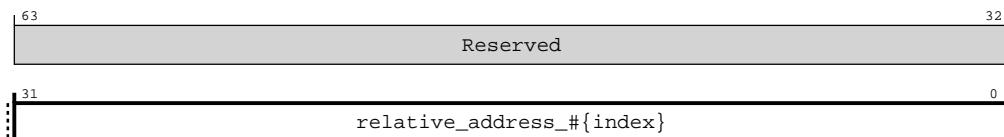
## Usage constraints

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 8-834: por\_mxp\_child\_pointer\_0-31**



**Table 8-845: por\_mxp\_child\_pointer\_0-31 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	relative_address_{index}	<p><b>Bit[31]</b> External or internal child node</p> <p><b>0b1</b> Indicates this child pointer points to a configuration node that is external to CMN</p> <p><b>0b0</b> Indicates this child pointer points to a configuration node that is internal to CMN</p> <p><b>Bits[30]</b> Child node owned by an isolated device.</p> <p><b>see</b> <a href="#">por_mxp_device_port_disable</a></p> <p><b>0b1</b> Indicates this child points is isolated. Thus, the child node cannot be read or accessed.</p> <p><b>0b0</b> Indicates this child points is not isolated. Thus, the child node is:</p> <p><b>Bits[29:0]</b> Child node address offset relative to PERIPHBASE</p>	RO	0b0

### 8.3.14.8 por\_mxp\_p0-5\_info

There are 6 iterations of this register. The index ranges from 0 to 5. Provides component identification information for XP port #*{index}*. NOTE: There will be max. of 8 MXP Port Info registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port Info register will be at the next 8 byte address boundary. Each successive MXP Port Info register will be named with the suffix. For example por\_mxp\_p<0:7>\_info

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x900 + #*{16\*index}*

##### Type

RO

##### Reset value

See individual bit resets

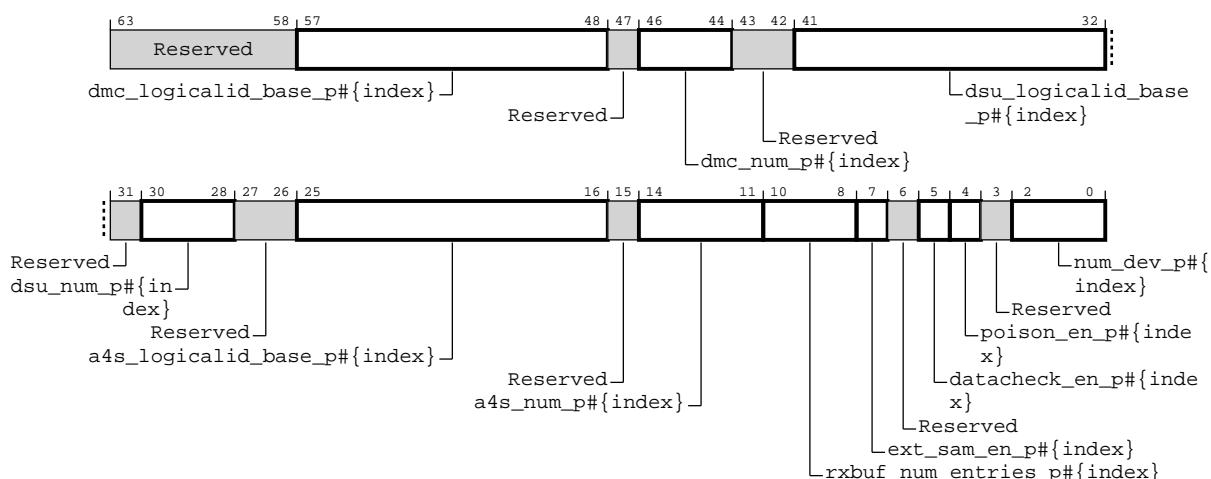
##### Usage constraints

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 8-835: por\_mxp\_p0-5\_info**



**Table 8-846: por\_mxp\_p0-5\_info attributes**

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	dmc_logicalid_base_p#{index}	DMC AXIU interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[47]	Reserved	Reserved	RO	-
[46:44]	dmc_num_p#{index}	Total number of SN-F AXIU interfaces at this port (0 to 4)	RO	Configuration dependent
[43:42]	Reserved	Reserved	RO	-
[41:32]	dsu_logicalid_base_p#{index}	DSU AXIU interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[31]	Reserved	Reserved	RO	-
[30:28]	dsu_num_p#{index}	Total number of RN-F AXIU interfaces at this port (0 to 4)	RO	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	a4s_logicalid_base_p#{index}	AXI4Stream interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[15]	Reserved	Reserved	RO	-
[14:11]	a4s_num_p#{index}	Total number of RN-F AXI4Stream interfaces at this port (0 to 4)	RO	Configuration dependent
[10:8]	rdbuf_num_entries_p#{index}	Number of input buffers at this port (2 to 4)	RO	Configuration dependent
[7]	ext_sam_en_p#{index}	ESAM enable	RO	Configuration dependent
[6]	Reserved	Reserved	RO	-
[5]	datacheck_en_p#{index}	Datacheck enable	RO	Configuration dependent
[4]	poison_en_p#{index}	Poison enable	RO	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2:0]	num_dev_p#{index}	Number of devices connected to this port (0 to 4)	RO	Configuration dependent

### 8.3.14.9 por\_mxp\_p0-5\_info\_1

There are 6 iterations of this register. The index ranges from 0 to 5. Provides component identification information for XP port #{index}. NOTE: There will be max. of 8 MXP Port Info registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port Info register will be at the next 8 byte address boundary. Each successive MXP Port Info register will be named with the suffix. For example por\_mxp\_p<0:7>\_info\_1

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x908 + #{16\*index}

##### Type

RO

### Reset value

See individual bit resets

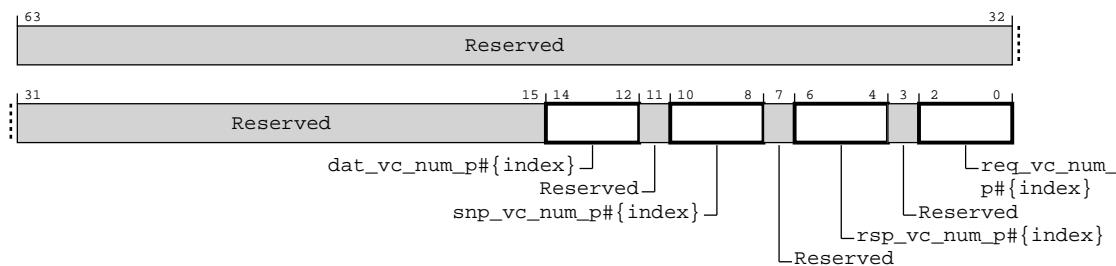
### Usage constraints

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 8-836: por\_mxp\_p0-5\_info\_1**



**Table 8-847: por\_mxp\_p0-5\_info\_1 attributes**

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	dat_vc_num_p#{index}	Number of replicated channels on DAT VC at this port	RO	Configuration dependent
[11]	Reserved	Reserved	RO	-
[10:8]	snp_vc_num_p#{index}	Number of replicated channels on SNP VC at this port	RO	Configuration dependent
[7]	Reserved	Reserved	RO	-
[6:4]	rsp_vc_num_p#{index}	Number of replicated channels on RSP VC at this port	RO	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2:0]	req_vc_num_p#{index}	Number of replicated channels on REQ VC at this port	RO	Configuration dependent

### 8.3.14.10 por\_dtm\_unit\_info

Provides component identification information for XP port 0 and 1.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x960

## Type

RO

## Reset value

See individual bit resets

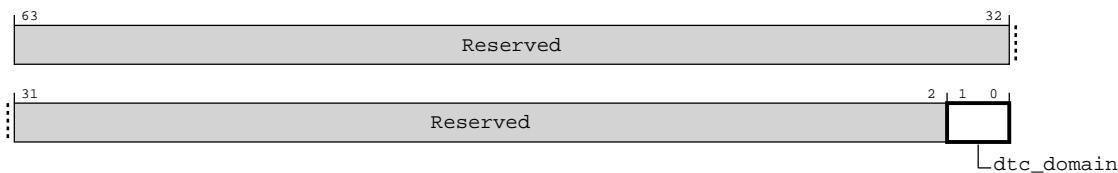
## Usage constraints

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 8-837: por\_dtm\_unit\_info**



**Table 8-848: por\_dtm\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	dtc_domain	DTC domain number associated with this DTM	RO	Configuration dependent

### 8.3.14.11 por\_dtm\_unit\_info\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Provides component identification information for XP ports #{{2\*index}} and #{{(2\*index)+1}}.



There are a maximum of 4 DTM Unit Info registers, based on `MXP_MULTIPLE_DTM_EN` and `MXP_NUM_DEV_PORT` value. Each successive DTM Unit Info register will be at the next 8-byte address boundary and named with the suffix corresponding to the DT register number.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0x968 + #{8\*(index-1)}

### Type

RO

### Reset value

See individual bit resets

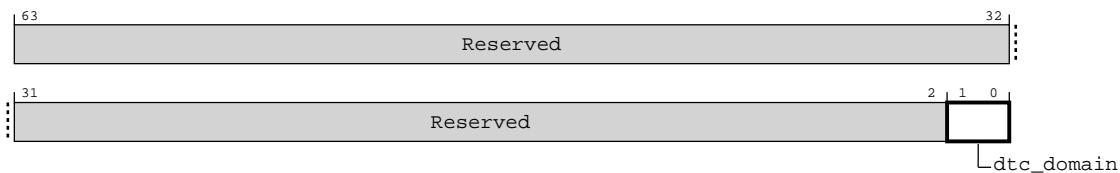
### Usage constraints

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 8-838: por\_dtm\_unit\_info\_dt1-3**



**Table 8-849: por\_dtm\_unit\_info\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	dtc_domain	DTC domain number associated with this DTM	RO	Configuration dependent

### 8.3.14.12 por\_mxp\_scr

Secure register access override.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x980

#### Type

RW

### Reset value

See individual bit resets

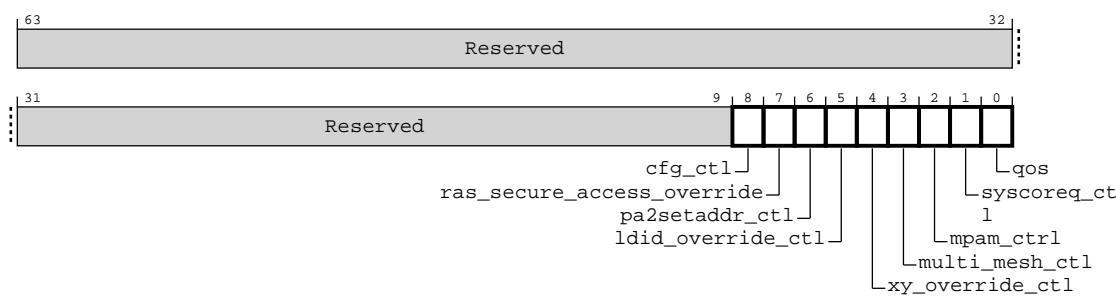
### Usage constraints

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 8-839: por\_mxp\_scr**



**Table 8-850: por\_mxp\_scr attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0
[7]	ras_secure_access_override	Allow Secure override of the to RAS registers	RW	0b0
[6]	pa2setaddr_ctl	Allows Secure override of the PA to SETADDR control registers	RW	0b0
[5]	ldid_override_ctl	Allows Secure override of the LDID override registers	RW	0b0
[4]	xy_override_ctl	Allows Secure override of the XY override registers	RW	0b0
[3]	multi_mesh_ctl	Allows Secure override of the Multi Mesh control registers	RW	0b0
[2]	mpam_ctrl	Allows Secure override of the CHI port MPAM override registers	RW	0b0
[1]	syscoreq_ctl	Allows Secure override of the syscoreq_ctl registers	RW	0b0
[0]	qos	Allows Secure override of the QoS registers	RW	0b0

### 8.3.14.13 por\_mxp\_rcr

Root register access override.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x988

### Type

RW

### Reset value

See individual bit resets

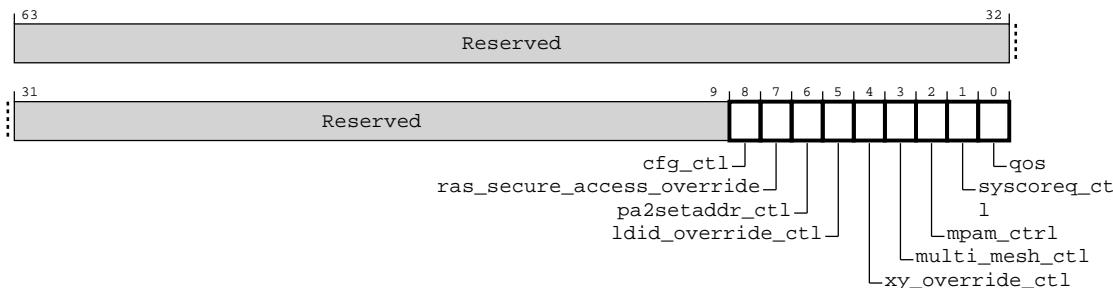
### Usage constraints

This register is owned in the Root space and is 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 8-840: por\_mxp\_rcr**



**Table 8-851: por\_mxp\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	
[8]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0
[7]	ras_secure_access_override	Allows Root override of the RAS registers	RW	0b0
[6]	pa2setaddr_ctl	Allows Root override of the PA to SETADDR control registers	RW	0b0
[5]	ldid_override_ctl	Allows Root override of the LDID override registers	RW	0b0
[4]	xy_override_ctl	Allows Root override of the XY override registers	RW	0b0
[3]	multi_mesh_ctl	Allows Root override of the Multi Mesh control registers	RW	0b0
[2]	mpam_ctrl	Allows Root override of the CHI port MPAM override registers	RW	0b0
[1]	syscoreq_ctl	Allows Root override of the syscoreq_ctl registers	RW	0b0
[0]	qos	Allows Root override of the QoS registers	RW	0b0

### 8.3.14.14 por\_mxp\_aux\_ctl

Functions as the auxiliary control register for XP.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

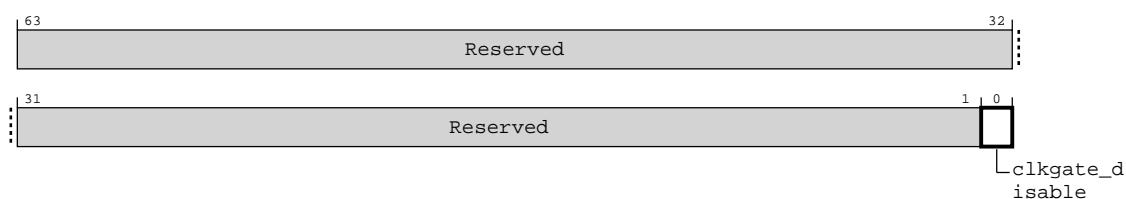
#### Usage constraints

This register is owned in the Root space and is 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 8-841: por\_mxp\_aux\_ctl**



**Table 8-852: por\_mxp\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	clkgate_disable	Disables clock gating when set	RW	0b0

### 8.3.14.15 por\_mxp\_device\_port\_ctl

Functions as the control register for XP device ports.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

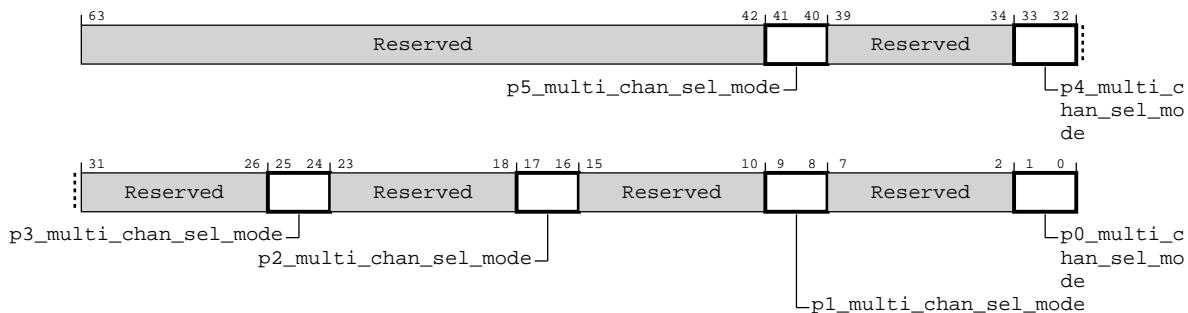
##### Usage constraints

This register is owned in the Root space and is 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 8-842: por\_mxp\_device\_port\_ctl**



**Table 8-853: por\_mxp\_device\_port\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[41:40]	p5_multi_chan_sel_mode	<p>Selects the mode/scheme for channel selection in multi channel mesh structure</p> <p><b>0x0</b> Enable channel mapping based on TGTID scheme</p> <p><b>0x1</b> Enable channel mapping based on dynamic credit availability scheme</p> <p><b>0x2</b> Enable channel mapping based on direct connect scheme</p> <p><b>0x3</b> Reserved</p>	RW	0b0
[39:34]	Reserved	Reserved	RO	
[33:32]	p4_multi_chan_sel_mode	<p>Selects the mode/scheme for channel selection in multi channel mesh structure</p> <p><b>0x0</b> Enable channel mapping based on TGTID scheme</p> <p><b>0x1</b> Enable channel mapping based on dynamic credit availability scheme</p> <p><b>0x2</b> Enable channel mapping based on direct connect scheme</p> <p><b>0x3</b> Reserved</p>	RW	0b0
[31:26]	Reserved	Reserved	RO	
[25:24]	p3_multi_chan_sel_mode	<p>Selects the mode/scheme for channel selection in multi channel mesh structure</p> <p><b>0x0</b> Enable channel mapping based on TGTID scheme</p> <p><b>0x1</b> Enable channel mapping based on dynamic credit availability scheme</p> <p><b>0x2</b> Enable channel mapping based on direct connect scheme</p> <p><b>0x3</b> Reserved</p>	RW	0b0
[23:18]	Reserved	Reserved	RO	
[17:16]	p2_multi_chan_sel_mode	<p>Selects the mode/scheme for channel selection in multi channel mesh structure</p> <p><b>0x0</b> Enable channel mapping based on TGTID scheme</p> <p><b>0x1</b> Enable channel mapping based on dynamic credit availability scheme</p> <p><b>0x2</b> Enable channel mapping based on direct connect scheme</p> <p><b>0x3</b> Reserved</p>	RW	0b0
[15:10]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[9:8]	p1_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure  <b>0x0</b> Enable channel mapping based on TGTID scheme  <b>0x1</b> Enable channel mapping based on dynamic credit availability scheme  <b>0x2</b> Enable channel mapping based on direct connect scheme  <b>0x3</b> Reserved	RW	0b0
[7:2]	Reserved	Reserved	RO	
[1:0]	p0_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure  <b>0x0</b> Enable channel mapping based on TGTID scheme  <b>0x1</b> Enable channel mapping based on dynamic credit availability scheme  <b>0x2</b> Enable channel mapping based on direct connect scheme  <b>0x3</b> Reserved	RW	0b0

### 8.3.14.16 por\_mxp\_p0-5\_mpam\_override

There are 6 iterations of this register. The index ranges from 0 to 5. Controls MPAM fields for devices connected to port #{{index}}. Valid only if the devices doesn't support MPAM.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA10 + #{{8\*index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.mpam\_ctrl

## Secure group override

por\_mxp\_scr.mpam\_ctrl

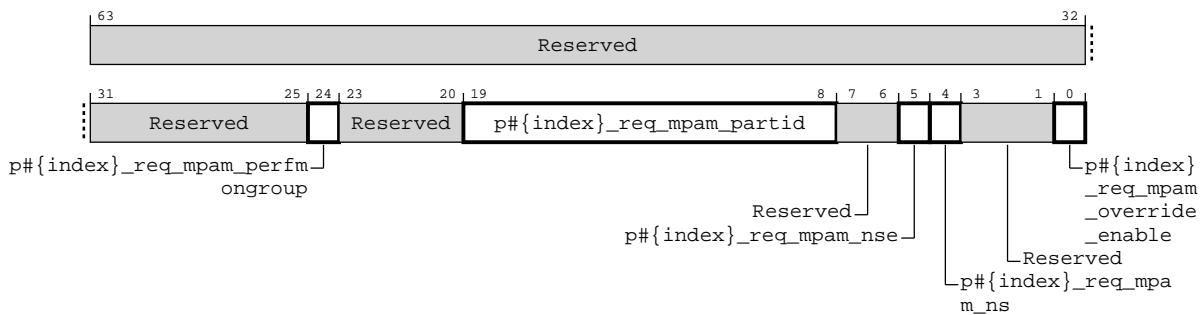
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.mpam\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.mpam\_ctrl bit and por\_mxp\_rsr.mpam\_ctrl 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 8-843: por\_mxp\_p0-5\_mpam\_override**



**Table 8-854: por\_mxp\_p0-5\_mpam\_override attributes**

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	
[24]	p#{index}_req_mpam_perfmongroup	MPAM.PerfMonGroup sub-field that overrides the REQ channel MPAM.PerfMonGroup when p#{index}_req_mpam_override_enable is set	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	p#{index}_req_mpam_partid	MPAM.PartID sub-field that overrides the REQ channel MPAM.PartID when p#{index}_req_mpam_override_enable is set	RW	0b0
[7:6]	Reserved	Reserved	RO	
[5]	p#{index}_req_mpam_nse	MPAM.NSE sub-field that overrides the REQ channel MPAM.NSE when p#{index}_req_mpam_override_enable is set	RW	0b0
[4]	p#{index}_req_mpam_ns	MPAM.NS sub-field that overrides the REQ channel MPAM.NS when p#{index}_req_mpam_override_enable is set	RW	0b0
[3:1]	Reserved	Reserved	RO	
[0]	p#{index}_req_mpam_override_enable	P#{index} DEV MPAM Override Enable on REQ Channel:  1 - Drive the MPAM fields on REQ channel with the values from this register  0 - Override of MPAM fields in REQ channel is disabled	RW	0b0

### 8.3.14.17 por\_mxp\_p0-5\_ldid\_override

There are 6 iterations of this register. The index ranges from 0 to 5. Controls LDID fields in REQ FLIT for devices connected to port #{{index}}. Valid only if POR\_MXP\_RNF\_CLUSTER\_EN\_PARAM is 1.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA40 + #{{8\*index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.ldid\_override\_ctl

##### Secure group override

por\_mxp\_scr.ldid\_override\_ctl

#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ldid\_override\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ldid\_override\_ctl bit and por\_mxp\_rcr.ldid\_override\_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 8-844: por\_mxp\_p0-5\_ldid\_override**



**Table 8-855: por\_mxp\_p0-5\_ldid\_override attributes**

Bits	Name	Description	Type	Reset
[63]	p#{index}_req_ldid_override_enable	P#{index} DEV LDID Override Enable on REQ Channel:  1 - Drive the LDID fields on REQ channel with the values from this register  0 - Override of LDID fields in REQ channel is disabled	RW	0b0
[62:60]	Reserved	Reserved	RO	
[59:48]	p#{index}_d3_req_ldid_field	LDID value that overrides the P#{index}_D3 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	0b0
[47:44]	Reserved	Reserved	RO	
[43:32]	p#{index}_d2_req_ldid_field	LDID value that overrides the P#{index}_D2 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	0b0
[31:28]	Reserved	Reserved	RO	
[27:16]	p#{index}_d1_req_ldid_field	LDID value that overrides the P#{index}_D1 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	0b0
[15:12]	Reserved	Reserved	RO	
[11:0]	p#{index}_d0_req_ldid_field	LDID value that overrides the P#{index}_D0 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	0b0

### 8.3.14.18 por\_mxp\_device\_port\_disable

Controls whether a device port attached to the SMXP is disabled

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

0xA70

### Type

RW

### Reset value

See individual bit resets

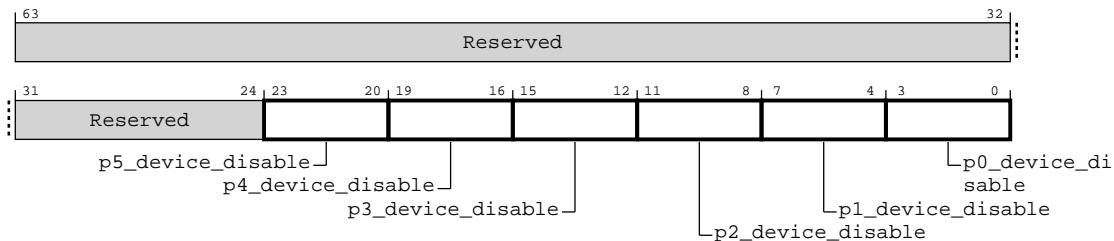
### Usage constraints

This register is owned in the Root space and is 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 8-845: por\_mxp\_device\_port\_disable**



**Table 8-856: por\_mxp\_device\_port\_disable attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	p5_device_disable	Port 5 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111

Bits	Name	Description	Type	Reset
[19:16]	p4_device_disable	Port 4 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111
[15:12]	p3_device_disable	Port 3 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111
[11:8]	p2_device_disable	Port 2 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111
[7:4]	p1_device_disable	Port 1 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111

Bits	Name	Description	Type	Reset
[3:0]	p0_device_disable	Port 0 device disabled <b>0b0001</b> Disable device 0 <b>0b0010</b> Disable device 1 <b>0b0100</b> Disable device 2 <b>0b1000</b> Disable device 3	RW	0b1111

### 8.3.14.19 por\_mxp\_cfg\_ctl

Functions as the configuration control register for MXP.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA78

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.cfg\_ctl

##### Secure group override

por\_mxp\_scr.cfg\_ctl

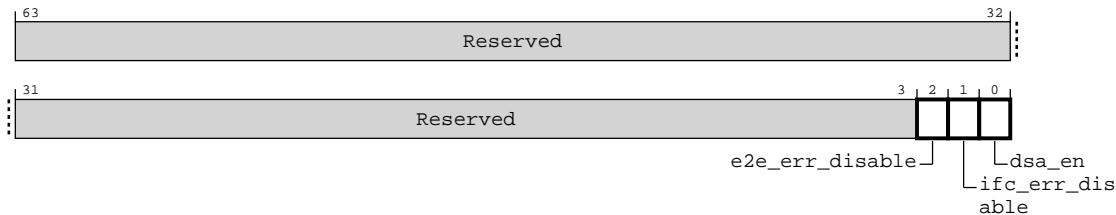
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.cfg\_ctl bit and por\_mxp\_rcr.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 8-846: por\_mxp\_cfg\_ctl**



**Table 8-857: por\_mxp\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	e2e_err_disable	Disable FUSA IFC ERR generation in MXP	RW	0b0
[1]	ifc_err_disable	Disable FUSA IFC ERR generation in MXP	RW	0b0
[0]	dsa_en	Enables Direct Subordinate Access feature to SBSX/MTSX. Requires POR_DSA_EN_PARAM to be set, otherwise this config bit has no effect.	RW	0b0

### 8.3.14.20 por\_mxp\_p0-5\_qos\_control

There are 6 iterations of this register. The index ranges from 0 to 5. Controls QoS settings for devices connected to port #{{index}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA80 + #{{32\*index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.qos

##### Secure group override

por\_mxp\_scr.qos

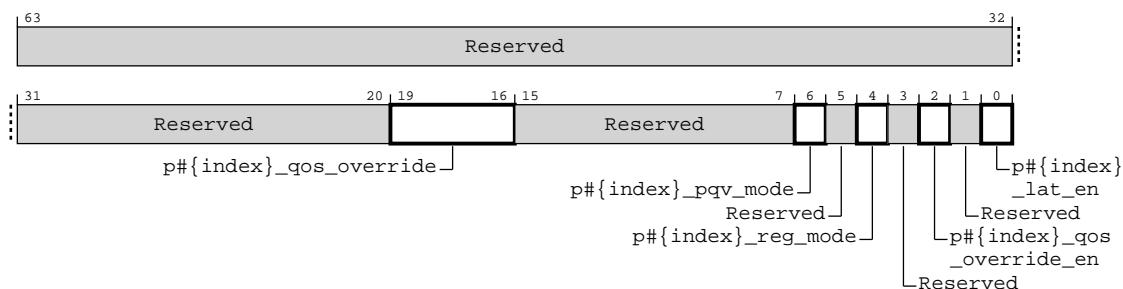
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.qos bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.qos bit and por\_mxp\_rcr.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 8-847: por\_mxp\_p0-5\_qos\_control**



**Table 8-858: por\_mxp\_p0-5\_qos\_control attributes**

Bits	Name	Description	Type	Reset
[63:20]	Reserved	Reserved	RO	-
[19:16]	p#[index]_qos_override	QoS override value for port #{index}	RW	0b0000
[15:7]	Reserved	Reserved	RO	-
[6]	p#[index]_pqv_mode	Configures the QoS regulator mode during period mode <b>0b0</b> Normal mode; QoS value is stable when the manager is idle <b>0b1</b> Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	0b0
[5]	Reserved	Reserved	RO	-
[4]	p#[index]_reg_mode	Configures the QoS regulator mode <b>0b0</b> Latency mode <b>0b1</b> Period mode; used for bandwidth regulation	RW	0b0
[3]	Reserved	Reserved	RO	-
[2]	p#[index]_qos_override_en	Enables port #{index} QoS override; when set, allows QoS value on inbound transactions to be overridden	RW	0b0
[1]	Reserved	Reserved	RO	-
[0]	p#[index]_lat_en	Enables port #{index} QoS regulation when set	RW	0b0

### 8.3.14.21 por\_mxp\_p0-5\_qos\_lat\_tgt

There are 6 iterations of this register. The index ranges from 0 to 5. Controls QoS target latency/period (in cycles) for regulation of devices connected to port #{{index}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA88 + #{{32\*index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.qos

##### Secure group override

por\_mxp\_scr.qos

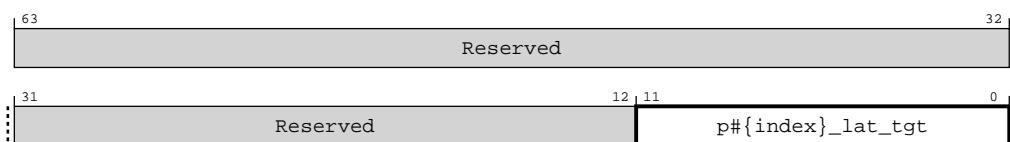
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.qos bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.qos bit and por\_mxp\_rcr.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 8-848: por\_mxp\_p0-5\_qos\_lat\_tgt**



**Table 8-859: por\_mxp\_p0-5\_qos\_lat\_tgt attributes**

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:0]	p#{index}_lat_tgt	Port #{{index}} transaction target latency/period; a value of 0 corresponds to no regulation	RW	0x000

### 8.3.14.22 por\_mxp\_p0-5\_qos\_lat\_scale

There are 6 iterations of this register. The index ranges from 0 to 5. Controls the QoS target scale factor for devices connected to port #*{index}*. The scale factor is represented in powers of two from the range  $2^{-3}$  to  $2^{-10}$ .

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA90 + #*{32\*index}*

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.qos

##### Secure group override

por\_mxp\_scr.qos

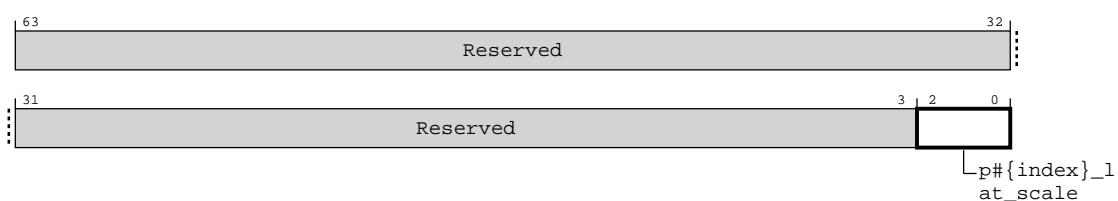
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.qos bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.qos bit and por\_mxp\_rcr.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 8-849: por\_mxp\_p0-5\_qos\_lat\_scale**



**Table 8-860: por\_mxp\_p0-5\_qos\_lat\_scale attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2:0]	p#{index}_lat_scale	Port 0 QoS scale factor <b>0b000</b> $2^{-3}$ <b>0b001</b> $2^{-4}$ <b>0b010</b> $2^{-5}$ <b>0b011</b> $2^{-6}$ <b>0b100</b> $2^{-7}$ <b>0b101</b> $2^{-8}$ <b>0b110</b> $2^{-9}$ <b>0b111</b> $2^{-10}$	RW	0x0

### 8.3.14.23 por\_mxp\_p0-5\_qos\_lat\_range

There are 6 iterations of this register. The index ranges from 0 to 5. Controls the minimum and maximum QoS values generated by the QoS regulator for devices connected to port #{index}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA98 + #{32\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.qos

## Secure group override

por\_mxp\_scr.qos

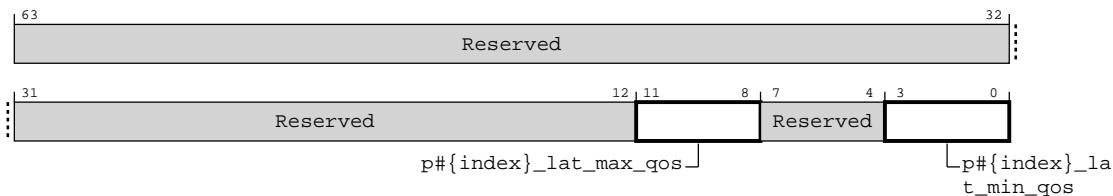
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.qos bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.qos bit and por\_mxp\_rcr.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 8-850: por\_mxp\_p0-5\_qos\_lat\_range**



**Table 8-861: por\_mxp\_p0-5\_qos\_lat\_range attributes**

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	
[11:8]	p#{index}_lat_max_qos	Port #{index} QoS maximum value	RW	0x0
[7:4]	Reserved	Reserved	RO	
[3:0]	p#{index}_lat_min_qos	Port #{index} QoS minimum value	RW	0x0

### 8.3.14.24 por\_mxp\_pmu\_event\_sel

Specifies the PMU event to be counted.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA000

### Type

RW

## Reset value

See individual bit resets

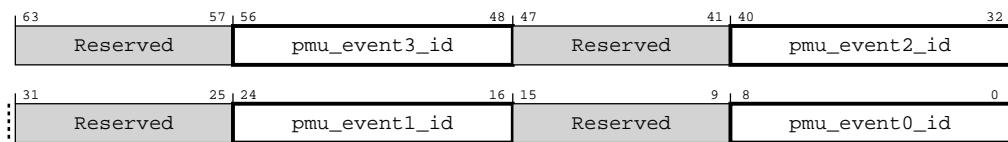
## Usage constraints

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 8-851: por\_mxp\_pmu\_event\_sel**



**Table 8-862: por\_mxp\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	
[56:48]	pmu_event3_id	XP PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[47:41]	Reserved	Reserved	RO	
[40:32]	pmu_event2_id	XP PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[31:25]	Reserved	Reserved	RO	
[24:16]	pmu_event1_id	XP PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0
[15:9]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[8:0]	pmu_event0_id	<p>XP PMU Event 0 ID Bits</p> <p><b>[8:5]</b> PC</p> <p><b>0b0000</b> REQ; REQ channel when POR_REQ_VC_NUM_PARAM = 1 ; REQ Sub-channel 1: when POR_REQ_VC_NUM_PARAM &gt; 1</p> <p><b>0b0001</b> RSP; RSP channel when POR_RSP_VC_NUM_PARAM = 1 ; RSP Sub-channel 1: when POR_RSP_VC_NUM_PARAM &gt; 1</p> <p><b>0b0010</b> SNP; SNP channel when POR_SNP_VC_NUM_PARAM = 1 ; SNP Sub-channel 1 : when POR_SNP_VC_NUM_PARAM &gt; 1</p> <p><b>0b0011</b> DAT; DAT channel when POR_DAT_VC_NUM_PARAM = 1 ; DAT Sub-channel 1: when POR_DAT_VC_NUM_PARAM &gt; 1</p> <p><b>0b0100</b> PUB</p> <p><b>0b0101</b> RSP2; RSP Sub-channel 2: Applicable when POR_RSP_VC_NUM_PARAM &gt; 1</p> <p><b>0b0110</b> DAT2; DAT Sub-channel 2: Applicable when POR_DAT_VC_NUM_PARAM &gt; 1</p> <p><b>0b0111</b> REQ2; REQ Sub-channel 2: Applicable when POR_REQ_VC_NUM_PARAM &gt; 1</p> <p><b>0b1000</b> SNP2; SNP Sub-channel 2 : Applicable when POR_SNP_VC_NUM_PARAM &gt; 1 Bits</p>	RW	0b0

Bits	Name	Description	Type	Reset
[8:0]	pmu_event0_id	<p>XP PMU Event 0 ID Bits</p> <p><b>[4:2]</b> Interface</p> <p><b>0b000</b> East when NUM_XP &gt; 1 ; Device port 0 when NUM_XP == 1 (Single XP config)</p> <p><b>0b001</b> West when NUM_XP &gt; 1 ; Device port 1 when NUM_XP == 1 (Single XP config)</p> <p><b>0b010</b> North when NUM_XP &gt; 1 ; Device port 2 when NUM_XP == 1 (Single XP config)</p> <p><b>0b011</b> South when NUM_XP &gt; 1 ; Device port 3 when NUM_XP == 1 (Single XP config)</p> <p><b>0b100</b> Device port 0 when NUM_XP &gt; 1 ; Device port 4 when NUM_XP == 1 (Single XP config)</p> <p><b>0b101</b> Device port 1 when NUM_XP &gt; 1 ; Device port 5 when NUM_XP == 1 (Single XP config)</p> <p><b>0b110</b> Device port 2 when NUM_XP &gt; 1 ; No Selection when NUM_XP == 1 (Single XP config)</p> <p><b>0b111</b> Device port 3 when NUM_XP &gt; 1 ; No Selection when NUM_XP == 1 (Single XP config) Bits</p> <p><b>[1:0]</b> Event specifier</p> <p><b>0b00</b> No event</p> <p><b>0b01</b> TX flit valid; signaled when a flit is successfully transmitted</p> <p><b>0b10</b> TX flit stall; signaled when flit transmission is stalled and waiting on credits</p> <p><b>0b11</b> Partial DAT flit; signaled when 128-bit DAT flits could not be merged into a 256-bit DAT flit; only applicable on the DAT PC on RN-F CHIA and RN-F CHIA ESAM ports</p>	RW	0b0

### 8.3.14.25 por\_mxp\_errfr

Functions as the error feature register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xE000

## Type

RO

## Reset value

See individual bit resets

## Root group override

`por_mxp_rcr.ras_secure_access_override`

## Secure group override

`por_mxp_scr.ras_secure_access_override`

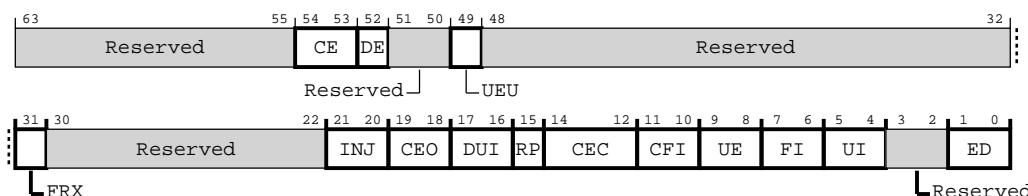
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.ras_secure_access_override` bit and `por_mxp_rcr.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 8-852: por\_mxp\_errfr**



**Table 8-863: por\_mxp\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording  0b00 Corrected Error not supported  0b10 Non-specific Corrected Error supported	RO	0b00
[52]	DE	Deferred Error recording  0b0 Deferred Error not supported  0b1 Deffered Error supported	RO	0b1

Bits	Name	Description	Type	Reset
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording <b>0b0</b> Unrecoverable Error not supported <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. <b>0b1</b> por_mxp_errfr[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite. <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_mxp_errctlr.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when por_mxp_errfr.CEC != 0b000.) <b>0b0</b> Invalid; <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in por_mxp_errmisco	RO	0b0
[14:12]	CEC	Standard corrected error counter <b>0b000</b> Does not implement standard error counter model <b>0b100</b> Implements a 16-bit Corrected error counter in por_mxp_errmisco	RO	0b000
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_mxp_errctlr.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01

Bits	Name	Description	Type	Reset
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_mxp_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_mxp_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_mxp_errctlr.ED	RO	0b10

### 8.3.14.26 por\_mxp\_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE008

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

##### Secure group override

por\_mxp\_scr.ras\_secure\_access\_override

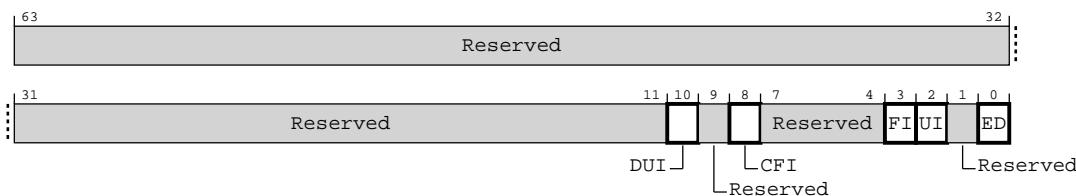
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ras\_secure\_access\_override bit and por\_mxp\_rcr.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 8-853: por\_mxp\_errctlr**



**Table 8-864: por\_mxp\_errctlr attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Enables error recovery interrupt for deferred error as specified in por_mxp_errfr.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_mxp_errfr.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_mxp_errfr.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_mxp_errfr.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_mxp_errfr.ED	RW	0b0

### 8.3.14.27 por\_mxp\_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

## Address offset

0xE010

## Type

W1C

## Reset value

See individual bit resets

## Root group override

`por_mxp_rcr.ras_secure_access_override`

## Secure group override

`por_mxp_scr.ras_secure_access_override`

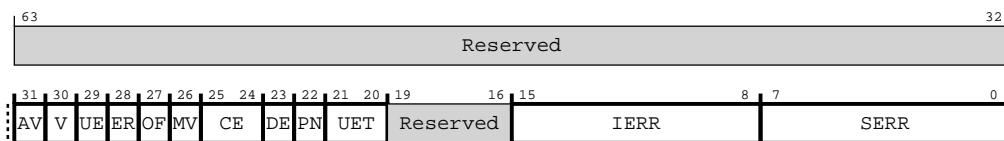
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.ras_secure_access_override` bit and `por_mxp_rcr.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 8-854: por\_mxp\_errstatus**



**Table 8-865: por\_mxp\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> Address is valid <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0

Bits	Name	Description	Type	Reset
[29]	UE	<p>Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear</p> <p><b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate</p> <p><b>0b0</b> No uncorrected errors detected</p>	W1C	0b0
[28]	ER	<p>Error Reported</p> <p><b>0b1</b> In-band error response signaled to the Requester</p> <p><b>0b0</b> No in-band error response signaled</p>	W1C	0b0
[27]	OF	<p>Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear</p> <p><b>0b1</b> More than one error detected</p> <p><b>0b0</b> None or only one error detected as described by UE/DE/CE fields</p>	W1C	0b0
[26]	MV	<p>por_mxp_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear</p> <p><b>0b1</b> Miscellaneous registers are valid</p> <p><b>0b0</b> Miscellaneous registers are not valid</p>	W1C	0b0
[25:24]	CE	<p>Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear</p> <p><b>0b10</b> At least one corrected error recorded</p> <p><b>0b00</b> No corrected errors recorded</p>	W1C	0b00
[23]	DE	<p>Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear</p> <p><b>0b1</b> At least one error is not corrected and is deferred</p> <p><b>0b0</b> No errors deferred</p>	W1C	0b0
[22]	PN	<p>Poison</p> <p><b>0b1</b> Uncorrected error recorded because a poison value was consumed</p> <p><b>0b0</b> Other cases</p>	W1C	0b0

Bits	Name	Description	Type	Reset
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code. bit  <b>0</b> Partner implementation defined error bit  <b>1</b> FUSA IFC_ERR error bit  <b>2</b> FUSA E2E_ERR error  <b>Other bits</b> reserved	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_mxp_errmisc1.ERRSRC for error type details.	W1C	0b0

### 8.3.14.28 por\_mxp\_erraddr

Contains the error record address.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE018

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

## Secure group override

`por_mxp_scr.ras_secure_access_override`

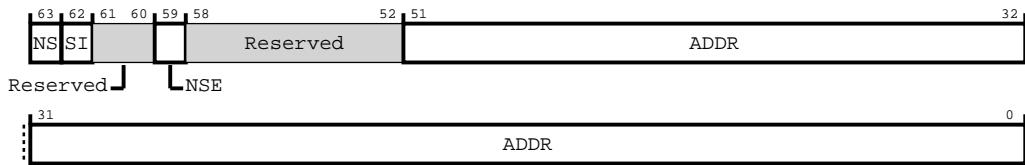
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.ras_secure_access_override` bit and `por_mxp_rcr.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 8-855: por\_mxp\_erraddr**



**Table 8-866: por\_mxp\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

## 8.3.14.29 por\_mxp\_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

0xE028

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

### Secure group override

por\_mxp\_scr.ras\_secure\_access\_override

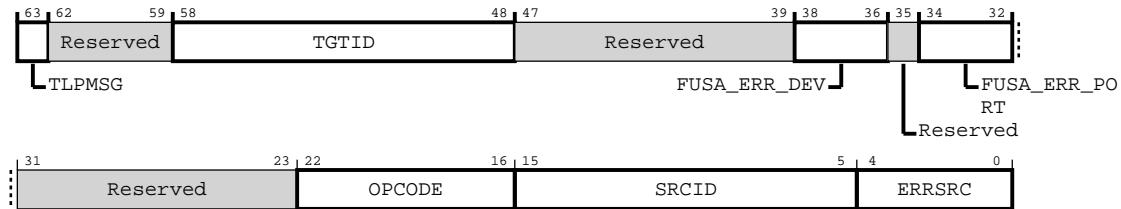
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ras\_secure\_access\_override bit and por\_mxp\_rcr.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 8-856: por\_mxp\_errmisc1**



**Table 8-867: por\_mxp\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63]	TLPMSG	Error flit TLPMSG Status	RW	0b0
[62:59]	Reserved	Reserved	RO	
[58:48]	TGTID	Error flit target ID	RW	0b0
[47:39]	Reserved	Reserved	RO	
[38:36]	FUSA_ERR_DEV	FUSA error device	RW	0b0
[35]	Reserved	Reserved	RO	
[34:32]	FUSA_ERR_PORT	FUSA error port	RW	0b0

Bits	Name	Description	Type	Reset
[31:23]	Reserved	Reserved	RO	
[22:16]	OPCODE	Error flit opcode	RW	0b0
[15:5]	SRCID	Error flit source ID	RW	0b0
[4:0]	ERRSRC	Error source for Mesh With 8 ports Bit[3:2]: Transaction type <b>0b00</b> REQ <b>0b01</b> RSP <b>0b10</b> SNP <b>0b11</b> DAT  Bits[4,1:0]: Port <b>0b000</b> Port 0 <b>0b001</b> Port 1 <b>0b010</b> Port 2 <b>0b011</b> Port 3 <b>0b100</b> Port 4 <b>0b101</b> Port 5 <b>0b110</b> Port 6 <b>0b111</b> Port 7	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh With Replicated Channels Bits[4:2]: Transaction type</p> <p><b>0b000</b> REQ</p> <p><b>0b001</b> RSP</p> <p><b>0b010</b> SNP</p> <p><b>0b011</b> DAT</p> <p><b>0b100</b> REQ2</p> <p><b>0b101</b> RSP2</p> <p><b>0b110</b> SNP2</p> <p><b>0b111</b> DAT2</p> <p>Bit[1:0]: Port</p> <p><b>0b00</b> Port 0</p> <p><b>0b01</b> Port 1</p> <p><b>0b10</b> Port 2</p> <p><b>0b11</b> Port 3</p>	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh Without Replicated Channels Bits[4:2]: Transaction type</p> <p><b>0b000</b> REQ</p> <p><b>0b001</b> RSP</p> <p><b>0b010</b> SNP</p> <p><b>0b011</b> DAT</p> <p><b>0b100</b> Reserved</p> <p><b>0b101</b> Reserved</p> <p><b>0b110</b> Reserved</p> <p><b>0b111</b> Reserved</p> <p>Bit[1:0]: Port</p> <p><b>0b00</b> Port 0</p> <p><b>0b01</b> Port 1</p> <p><b>0b10</b> Port 2</p> <p><b>0b11</b> Port 3</p>	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh With Single MXP (1x1 Mesh) Bits [4:3]: Transaction type</p> <p><b>0b00</b> REQ</p> <p><b>0b01</b> RSP</p> <p><b>0b10</b> SNP</p> <p><b>0b11</b> DAT</p> <p>Bits[2:0]: Port</p> <p><b>0b000</b> Port 0</p> <p><b>0b001</b> Port 1</p> <p><b>0b010</b> Port 2</p> <p><b>0b011</b> Port 3</p> <p><b>0b100</b> Port 4</p> <p><b>0b101</b> Port 5</p>	RW	0b0

### 8.3.14.30 por\_mxp\_p0-5\_byte\_par\_err\_inj

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the byte parity error injection register for XP port #{{index}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE930 + #{8\*index}

##### Type

WO

### Reset value

See individual bit resets

### Root group override

`por_mxp_rcr.ras_secure_access_override`

### Secure group override

`por_mxp_scr.ras_secure_access_override`

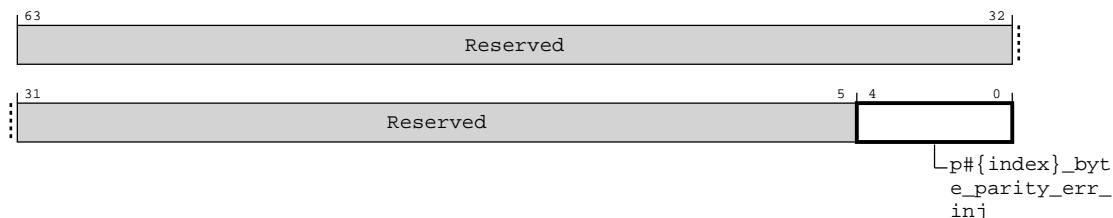
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.ras_secure_access_override` bit and `por_mxp_rcr.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 8-857: por\_mxp\_p0-5\_byte\_par\_err\_inj**



**Table 8-868: por\_mxp\_p0-5\_byte\_par\_err\_inj attributes**

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	
[4:0]	<code>p#{index}_byte_parity_err_inj</code>	<p>Specifies a byte lane; once this register is written, a byte parity error is injected in the specified byte lane on the next DAT flit upload</p> <p><b>NOTE</b></p> <p>Only applicable if an RN-F is attached to port #<code>{index}</code>. Byte parity error is only injected if the RN-F is configured to not support Datacheck.</p>	WO	0x00

### 8.3.14.31 por\_mxp\_errpfg

Functions as the Pseudo-fault Generation Feature Register.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE800

### Type

RO

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

### Secure group override

por\_mxp\_scr.ras\_secure\_access\_override

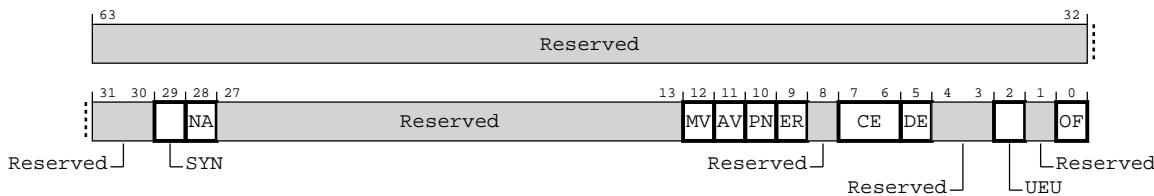
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ras\_secure\_access\_override bit and por\_mxp\_rcr.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 8-858: por\_mxp\_errpfgf**



**Table 8-869: por\_mxp\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update ERRSTATUS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	Miscellaneous syndrome. <b>0b1</b> Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	0b1
[11]	AV	Address syndrome. <b>0b1</b> Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	0b1
[10]	PN	Poison flag <b>0b1</b> Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation. <b>0b00</b> No Corrected error generation. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 0b10; else, update ERRSTATUS.CE to 0b00	RO	0b00
[5]	DE	Deferred error generation. <b>0b0</b> No Deferred error generation. <b>0b1</b> Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation. <b>0b1</b> If ERRPFGCTL.UEU == 1, update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01; else, update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag. <b>0b1</b> Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	0b1

### 8.3.14.32 por\_mxp\_errpfgctl

Functions as the Pseudo-fault Generation Control Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

## Address offset

0xE808

## Type

RW

## Reset value

See individual bit resets

## Root group override

`por_mxp_rcr.ras_secure_access_override`

## Secure group override

`por_mxp_scr.ras_secure_access_override`

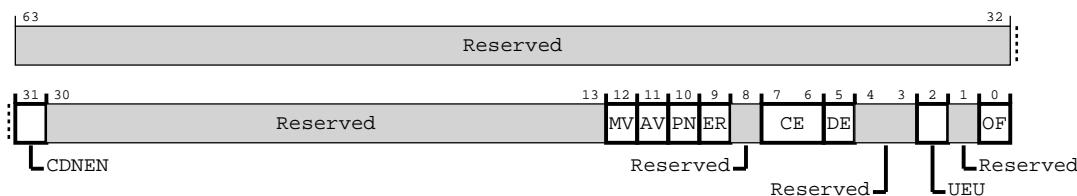
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.ras_secure_access_override` bit and `por_mxp_rcr.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 8-859: por\_mxp\_errpfgctl**



**Table 8-870: por\_mxp\_errpfgctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable.  0b0 Countdown disabled.  0b1 Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12]	MV	<p>Miscellaneous syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.MV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.MV to 0b1</p>	RW	0b0
[11]	AV	<p>Address syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.AV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.AV to 0b1</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.PN to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.PN to 0b1</p>	RW	0b0
[9]	ER	<p>Error reported flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.ER to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.ER to 0b1</p>	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 0b00</p> <p><b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 0b10</p>	RW	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.DE to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.DE to 0b1</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00</p> <p><b>0b1</b> Fault injection update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01</p>	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	OF	<p>Overflow flag.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.OF to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.OF to 0b1</p>	RW	0b0

### 8.3.14.33 por\_mxp\_errpfgcdn

Functions as the Pseudo-fault Generation Countdown Register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE810

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

##### Secure group override

por\_mxp\_scr.ras\_secure\_access\_override

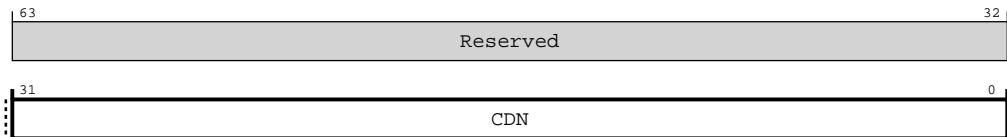
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ras\_secure\_access\_override bit and por\_mxp\_rcr.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 8-860: por\_mxp\_errpfgcdn**



**Table 8-871: por\_mxp\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.14.34 por\_mxp\_errfr\_NS

Functions as the non-secure error feature register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE040

##### Type

RO

##### Reset value

See individual bit resets

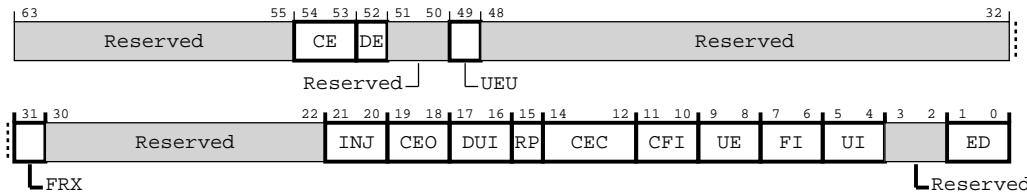
##### Usage constraints

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 8-861: por\_mxp\_errfr\_NS**



**Table 8-872: por\_mxp\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording  <b>0b00</b> Corrected Error not supported  <b>0b10</b> Non-specific Corrected Error supported	RO	0b00
[52]	DE	Deferred Error recording  <b>0b0</b> Deferred Error not supported  <b>0b1</b> Deffered Error supported	RO	0b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension.  <b>0b1</b> por_mxp_errfr_NS[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension.  <b>0b1</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite.  <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00

Bits	Name	Description	Type	Reset
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_mxp_errctlr.DUI.	RO	0b10
[15]	RP	Repeat counter (valid only when por_mxp_errfr_NS.CEC != 0b000.)  <b>0b1</b> Implements a first (repeat) counter and a second (other) counter in por_mxp_errmisc0	RO	0b0
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter in por_mxp_errmisc0	RO	0b000
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_mxp_errctlr.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_mxp_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_mxp_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_mxp_errctlr.ED	RO	0b10

### 8.3.14.35 por\_mxp\_errctlr\_NS

Functions as the non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE048

##### Type

RW

##### Reset value

See individual bit resets

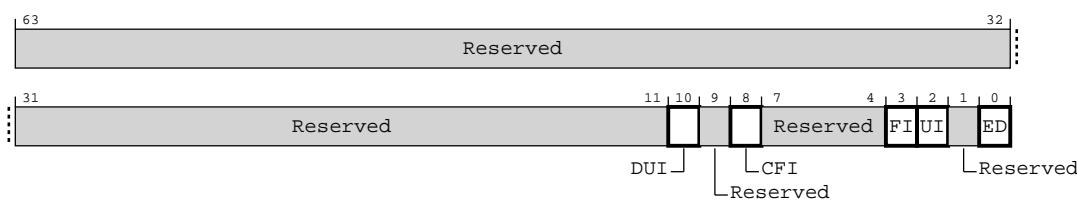
##### Usage constraints

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 8-862: por\_mxp\_errctlr\_NS**



**Table 8-873: por\_mxp\_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 por_mxp_errfr_NS.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_mxp_errfr_NS.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_mxp_errfr_NS.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_mxp_errfr_NS.UI	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	ED	Enables error detection as specified in por_mxp_errfr_NS.ED	RW	0b0

### 8.3.14.36 por\_mxp\_errstatus\_NS

Functions as the non-secure error status register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

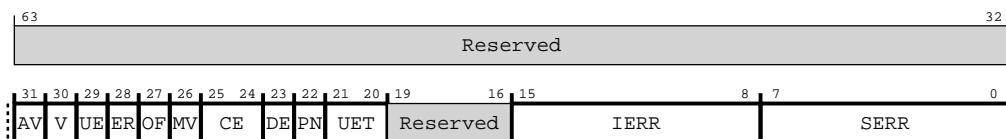
#### Usage constraints

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 8-863: por\_mxp\_errstatus\_NS**



**Table 8-874: por\_mxp\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> Address is valid <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear <b>0b1</b> More than one error detected <b>0b0</b> Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_mxp_errmisc_NS valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear <b>0b1</b> At least one transient corrected error recorded <b>0b0</b> No corrected errors recorded	W1C	0b00

Bits	Name	Description	Type	Reset
[23]	DE	<p>Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear</p> <p><b>0b1</b> At least one error is not corrected and is deferred</p> <p><b>0b0</b> No errors deferred</p>	W1C	0b0
[22]	PN	<p>Poison</p> <p><b>0b1</b> Uncorrected error recorded because a poison value was consumed</p> <p><b>0b0</b> Other cases</p>	W1C	0b0
[21:20]	UET	<p>Uncorrected Error Type, valid only when UE != 0</p> <p><b>0b01</b> Uncorrected error, Unrecoverable error (UEU).</p> <p><b>0b00</b> Invalid</p>	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	<p>Implementation-defined primary error code.</p> <p><b>0x00</b> No error</p> <p><b>0x01</b> Partner implementation defined error</p>	W1C	0b0
[7:0]	SERR	<p>Architecturally-defined primary error code.</p> <p><b>0x00</b> No error</p> <p><b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_mxp_errmisc1.ERRSRC for error type details.</p>	W1C	0b0

### 8.3.14.37 por\_mxp\_erraddr\_NS

Contains the non-secure error record address.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xE058

### 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 8-864: por\_mxp\_erraddr\_NS**



**Table 8-875: por\_mxp\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.14.38 por\_mxp\_errmisc1\_NS

Functions as the non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xE068

### Type

RW

### Reset value

See individual bit resets

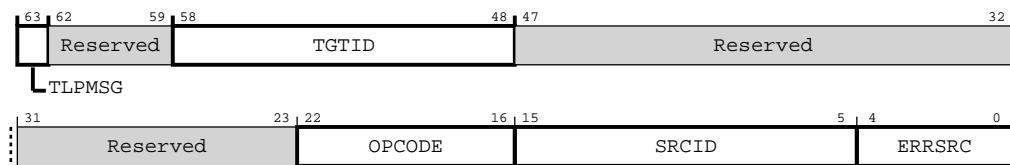
### Usage constraints

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 8-865: por\_mxp\_errmisc1\_NS**



**Table 8-876: por\_mxp\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	TLPMSG	Error flit TLPMSG Status	RW	0b0
[62:59]	Reserved	Reserved	RO	
[58:48]	TGTID	Error flit target ID	RW	0b0
[47:23]	Reserved	Reserved	RO	
[22:16]	OPCODE	Error flit opcode	RW	0b0
[15:5]	SRCID	Error flit source ID	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh With 8 ports Bit[3:2]: Transaction type</p> <p><b>0b00</b> REQ</p> <p><b>0b01</b> RSP</p> <p><b>0b10</b> SNP</p> <p><b>0b11</b> DAT</p> <p>Bits[4,1:0]: Port</p> <p><b>0b000</b> Port 0</p> <p><b>0b001</b> Port 1</p> <p><b>0b010</b> Port 2</p> <p><b>0b011</b> Port 3</p> <p><b>0b100</b> Port 4</p> <p><b>0b101</b> Port 5</p> <p><b>0b110</b> Port 6</p> <p><b>0b111</b> Port 7</p>	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh With Replicated Channels Bits[4:2]: Transaction type</p> <p><b>0b000</b> REQ</p> <p><b>0b001</b> RSP</p> <p><b>0b010</b> SNP</p> <p><b>0b011</b> DAT</p> <p><b>0b100</b> REQ2</p> <p><b>0b101</b> RSP2</p> <p><b>0b110</b> SNP2</p> <p><b>0b111</b> DAT2</p> <p>Bit[1:0]: Port</p> <p><b>0b00</b> Port 0</p> <p><b>0b01</b> Port 1</p> <p><b>0b10</b> Port 2</p> <p><b>0b11</b> Port 3</p>	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh Without Replicated Channels Bits[4:2]: Transaction type</p> <p><b>0b000</b> REQ</p> <p><b>0b001</b> RSP</p> <p><b>0b010</b> SNP</p> <p><b>0b011</b> DAT</p> <p><b>0b100</b> Reserved</p> <p><b>0b101</b> Reserved</p> <p><b>0b110</b> Reserved</p> <p><b>0b111</b> Reserved</p> <p>Bit[1:0]: Port</p> <p><b>0b00</b> Port 0</p> <p><b>0b01</b> Port 1</p> <p><b>0b10</b> Port 2</p> <p><b>0b11</b> Port 3</p>	RW	0b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	<p>Error source for Mesh With Single MXP (1x1 Mesh) Bits [4:3]: Transaction type</p> <p><b>0b00</b> REQ</p> <p><b>0b01</b> RSP</p> <p><b>0b10</b> SNP</p> <p><b>0b11</b> DAT</p> <p>Bits[2:0]: Port</p> <p><b>0b000</b> Port 0</p> <p><b>0b001</b> Port 1</p> <p><b>0b010</b> Port 2</p> <p><b>0b011</b> Port 3</p> <p><b>0b100</b> Port 4</p> <p><b>0b101</b> Port 5</p>	RW	0b0

### 8.3.14.39 por\_mxp\_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

0xE840

##### Type

RO

##### Reset value

See individual bit resets

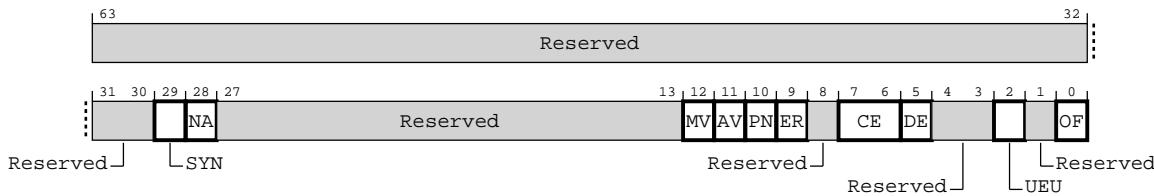
## Usage constraints

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 8-866: por\_mxp\_errpfgf\_NS**



**Table 8-877: por\_mxp\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection. <b>0b1</b> Fault injection does not update ERRSTATUS_NS.SERR	RO	0b1
[28]	NA	No access required. <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to ERRPFGCTL_NS.MV	RO	0b1
[11]	AV	Address syndrome. <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to ERRPFGCTL_NS.AV	RO	0b1
[10]	PN	Poison flag <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to ERRPFGCTL_NS.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to ERRPFGCTL_NS.ER	RO	0b1
[8]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> No Corrected error generation.</p> <p><b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL_NS.CE == 1, update ERRSTATUS_NS.CE to 0b10; else, update ERRSTATUS_NS.CE to 0b00</p>	RO	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> No Deferred error generation.</p> <p><b>0b1</b> Fault injection update ERRSTATUS_NS.DE to ERRPFGCTL_NS.DE</p>	RO	0b1
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b1</b> If ERRPFGCTL_NS.UEU == 1, update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01; else, update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00</p>	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	<p>Overflow flag.</p> <p><b>0b1</b> Fault injection update ERRSTATUS_NS.OF to ERRPFGCTL_NS.OF</p>	RO	0b1

### 8.3.14.40 por\_mxp\_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

0xE848

##### Type

RW

##### Reset value

See individual bit resets

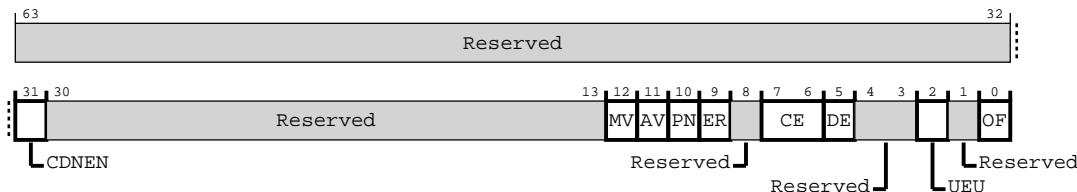
##### Usage constraints

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 8-867: por\_mxp\_errpfctl\_NS**



**Table 8-878: por\_mxp\_errpfctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDENEN	Countdown Enable. <b>0b0</b> Countdown disabled. <b>0b1</b> Error generation counter is set to ERRPFGCDN_NS.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.MV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to 0b1	RW	0b0
[11]	AV	Address syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.AV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to 0b1	RW	0b0
[10]	PN	Poison flag <b>0b0</b> Fault injection update ERRSTATUS_NS.PN to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to 0b1	RW	0b0
[9]	ER	Error reported flag <b>0b0</b> Fault injection update ERRSTATUS_NS.ER to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS_NS.CE to 0b00</p> <p><b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS_NS.CE to 0b10</p>	RW	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS_NS.DE to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS_NS.DE to 0b1</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	<p>Uncorrected error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00</p> <p><b>0b1</b> Fault injection update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01</p>	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	<p>Overflow flag.</p> <p><b>0b0</b> Fault injection update ERRSTATUS_NS.OF to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS_NS.OF to 0b1</p>	RW	0b0

### 8.3.14.41 por\_mxp\_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

0xE850

##### Type

RW

##### Reset value

See individual bit resets

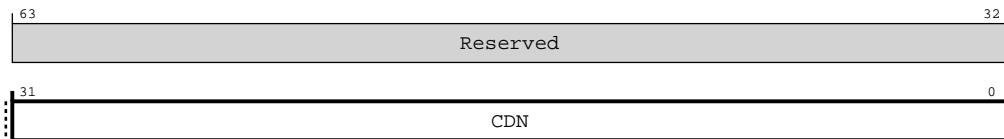
## Usage constraints

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 8-868: por\_mxp\_errpfgcdn\_NS**



**Table 8-879: por\_mxp\_errpfgcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.14.42 por\_mxp\_errcapctl

Functions as the error Capture Control Register

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xED00

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.ras\_secure\_access\_override

### Secure group override

por\_mxp\_scr.ras\_secure\_access\_override

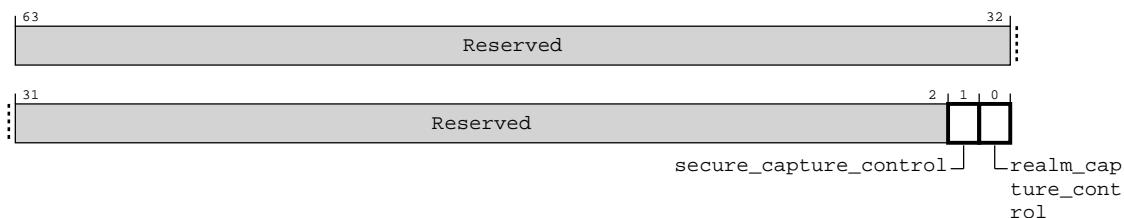
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.ras\_secure\_access\_override bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.ras\_secure\_access\_override bit and por\_mxp\_rcr.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 8-869: por\_mxp\_errcapctl**



**Table 8-880: por\_mxp\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control  <b>0b0</b> Transaction with secure PAS captured in root error record  <b>0b1</b> Transaction with secure PAS captured in non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control  <b>0b0</b> Transaction with realm PAS captured in root error record  <b>0b1</b> Transaction with realm PAS captured in non-secure error record	RW	0b0

### 8.3.14.43 por\_mxp\_errgsr

Functions as Error Group Status Register

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

### Address offset

0xEE00

### Type

RO

### Reset value

See individual bit resets

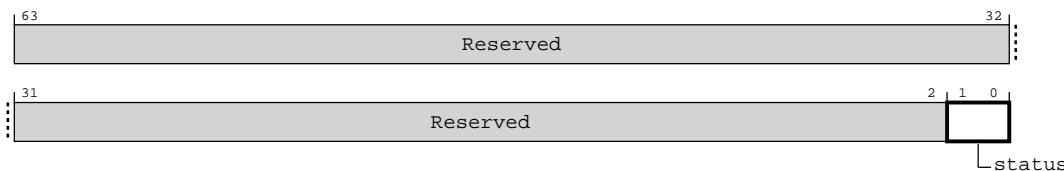
### Usage constraints

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 8-870: por\_mxp\_errgsr**



**Table 8-881: por\_mxp\_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	0b0

### 8.3.14.44 por\_mxp\_errildr

Functions as the implementation identification register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEE10

#### Type

RO

### Reset value

See individual bit resets

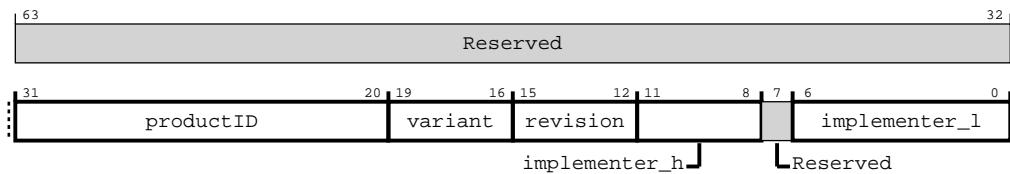
### Usage constraints

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 8-871: por\_mxp\_errildr**



**Table 8-882: por\_mxp\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.14.45 por\_mxp\_errdevaff

Functions as the device affinity register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xEFA8

##### Type

RO

### Reset value

See individual bit resets

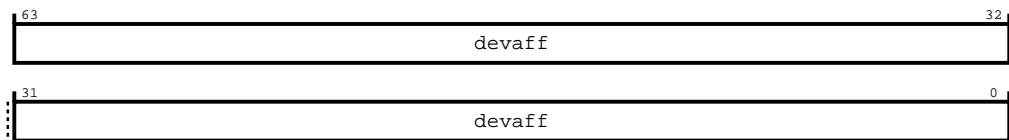
### Usage constraints

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 8-872: por\_mxp\_errdevaff**



**Table 8-883: por\_mxp\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.14.46 por\_mxp\_errdevarch

Functions as the device architecture register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEF8

#### Type

RO

### Reset value

See individual bit resets

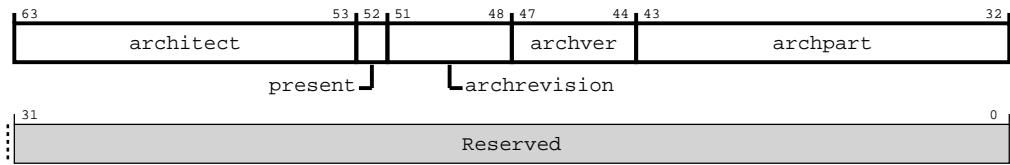
### Usage constraints

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 8-873: por\_mxp\_errdevarch**



**Table 8-884: por\_mxp\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

## 8.3.14.47 por\_mxp\_errdevid

Functions as the device configuration register

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xEFC8

#### Type

RO

#### Reset value

See individual bit resets

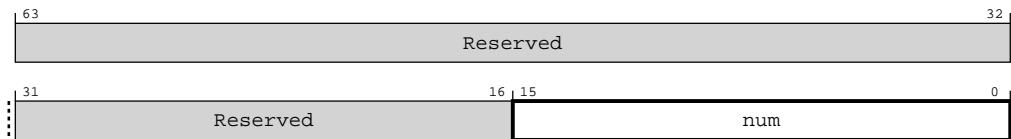
#### Usage constraints

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 8-874: por\_mxp\_errdevid**



**Table 8-885: por\_mxp\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

## 8.3.14.48 por\_mxp\_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

0xEFDO

#### Type

RO

#### Reset value

See individual bit resets

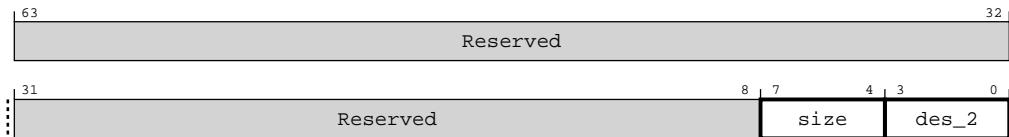
#### Usage constraints

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 8-875: por\_mxp\_errpidr45**



**Table 8-886: por\_mxp\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

### 8.3.14.49 por\_mxp\_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

0xEFEO

##### Type

RO

##### Reset value

See individual bit resets

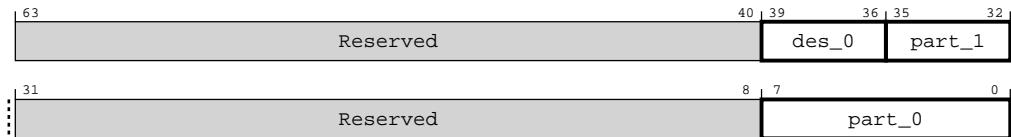
##### Usage constraints

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 8-876: por\_mxp\_errpidr01**



**Table 8-887: por\_mxp\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

### 8.3.14.50 por\_mxp\_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

0xEFE8

##### Type

RO

##### Reset value

See individual bit resets

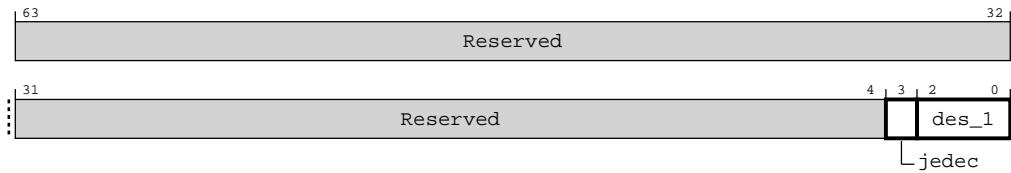
##### Usage constraints

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 8-877: por\_mxp\_errpidr23**



**Table 8-888: por\_mxp\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

### 8.3.14.51 por\_mxp\_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

0xEFF0

##### Type

RO

##### Reset value

See individual bit resets

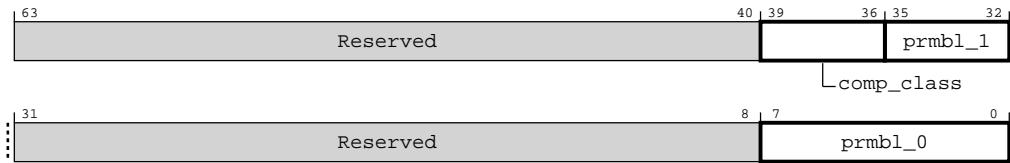
##### Usage constraints

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 8-878: por\_mxp\_errcidr01**



**Table 8-889: por\_mxp\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmbl_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmbl_0	PRMBL_0	RO	0xD

### 8.3.14.52 por\_mxp\_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

0xEFF8

##### Type

RO

##### Reset value

See individual bit resets

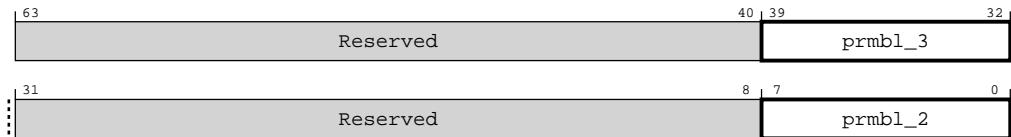
#### Usage constraints

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 8-879: por\_mxp\_errcidr23**



**Table 8-890: por\_mxp\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.14.53 por\_mxp\_p0-5\_syscoreq\_ctl

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the port #{{index}} snoop and DVM domain control register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por\_mxp\_p#{{index}}\_syscoack\_status. NOTE: Only valid on RN-F ports.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1000 + #{{16\*index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.syscoreq\_ctl

##### Secure group override

por\_mxp\_scr.syscoreq\_ctl

#### Usage constraints

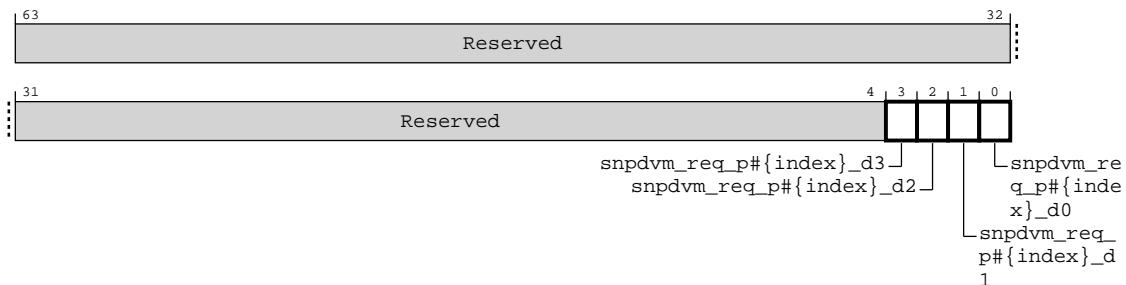
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.syscoreq\_ctl bit is set, Secure

accesses to this register are permitted. If both the por\_mxp\_scr.syscoreq\_ctl bit and por\_mxp\_rcr.syscoreq\_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.

**Figure 8-880: por\_mxp\_p0-5\_syscoreq\_ctl**



**Table 8-891: por\_mxp\_p0-5\_syscoreq\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	snpdvm_req_p#{index}_d3	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 3 on port #{index}	RW	0b0
[2]	snpdvm_req_p#{index}_d2	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 2 on port #{index}	RW	0b0
[1]	snpdvm_req_p#{index}_d1	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 1 on port #{index}	RW	0b0
[0]	snpdvm_req_p#{index}_d0	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 0 on port #{index}	RW	0b0

### 8.3.14.54 por\_mxp\_p0-5\_syscoack\_status

There are 8 iterations of this register. The index ranges from 0 to 5. Functions as the port #{index} snoop and DVM domain status register. Provides a software alternative to hardware SYSREQ/SYSACK handshake. Works with por\_mxp\_p#{index}\_syscoreq\_ctl. NOTE: Only valid on RN-F ports.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

`0x1008 + #{16*index}`

### Type

RO

### Reset value

See individual bit resets

### Root group override

`por_mxp_rcr.syscoreq_ctl`

### Secure group override

`por_mxp_scr.syscoreq_ctl`

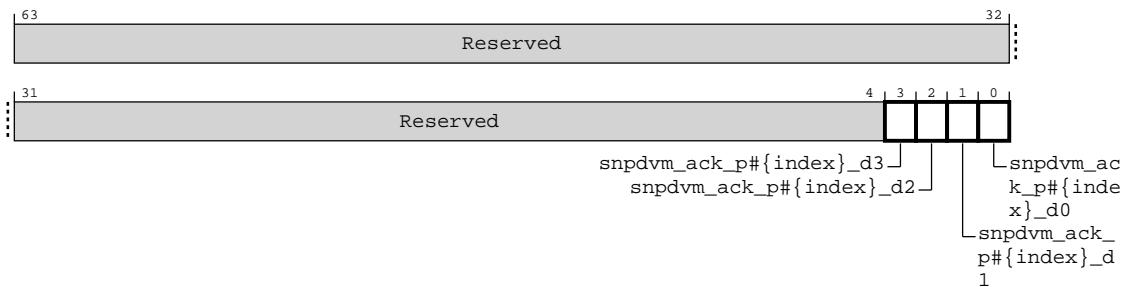
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.syscoreq_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.syscoreq_ctl` bit and `por_mxp_rcr.syscoreq_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.

**Figure 8-881: por\_mxp\_p0-5\_syscoack\_status**



**Table 8-892: por\_mxp\_p0-7\_syscoack\_status attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	snpdvm_ack_p#{index}_d3	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 3 on port #{{index}}	RO	0b0
[2]	snpdvm_ack_p#{index}_d2	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 2 on port #{{index}}	RO	0b0
[1]	snpdvm_ack_p#{index}_d1	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 1 on port #{{index}}	RO	0b0
[0]	snpdvm_ack_p#{index}_d0	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 0 on port #{{index}}	RO	0b0

### 8.3.14.55 por\_dtm\_control

Functions as the DTM control register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA100

##### Type

RW

##### Reset value

See individual bit resets

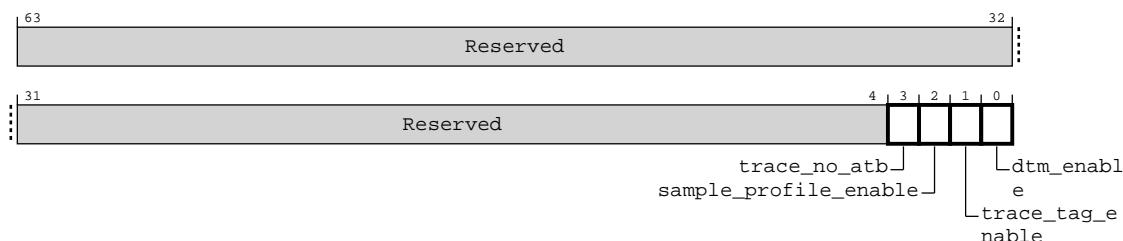
##### Usage constraints

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 8-882: por\_dtm\_control**



**Table 8-893: por\_dtm\_control attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	trace_no_atb	When set, trace packet is not delivered out of ATB, and FIFO entry holds the first trace packet.  <b>NOTE</b> if any MXP has this bit set, ATB protocol will not be functional.	RW	0b0
[2]	sample_profile_enable	Enables sample profile function	RW	0b0

Bits	Name	Description	Type	Reset
[1]	trace_tag_enable	Watchpoint trace tag enable  <b>0b1</b> Trace tag enabled  <b>0b0</b> No trace tag	RW	0b0
[0]	dtm_enable	Enables debug watchpoint and PMU function; prior to writing this bit, all other DT configuration registers must be programmed; once this bit is set, other DT configuration registers must not be modified	RW	0b0

### 8.3.14.56 por\_dtm\_fifo\_entry\_ready

Controls status of DTM FIFO entries.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA118

##### Type

W1C

##### Reset value

See individual bit resets

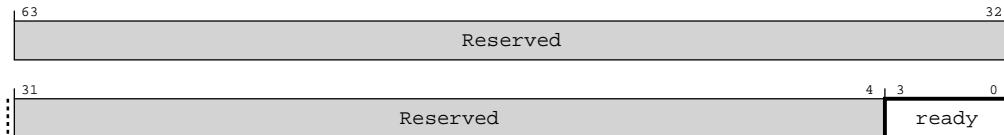
#### Usage constraints

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 8-883: por\_dtm\_fifo\_entry\_ready**



**Table 8-894: por\_dtm\_fifo\_entry\_ready attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3:0]	ready	<p>Indicates which DTM FIFO entries are ready; write a 1 to clear</p> <p><b>Bit[3]</b> Entry 3 ready when set</p> <p><b>Bit[2]</b> Entry 2 ready when set</p> <p><b>Bit[1]</b> Entry 1 ready when set</p> <p><b>Bit[0]</b> Entry 0 ready when set</p>	W1C	0b0

### 8.3.14.57 por\_dtm\_fifo\_entry0-3\_0

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{{index}} data.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA120 + #{32\*index}

##### 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 8-884: por\_dtm\_fifo\_entry0-3\_0**



**Table 8-895: por\_dtm\_fifo\_entry0-3\_0 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data0	Entry data bit vector 63:0	RO	0b0

### 8.3.14.58 por\_dtm\_fifo\_entry0-3\_1

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{{index}} data.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA128 + #{32\*index}

##### 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 8-885: por\_dtm\_fifo\_entry0-3\_1**



**Table 8-896: por\_dtm\_fifo\_entry0-3\_1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data1	Entry data bit vector 127:64	RO	0b0

### 8.3.14.59 por\_dtm\_fifo\_entry0-3\_2

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{{index}} data.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA130 + #{32\*index}

##### Type

RO

##### Reset value

See individual bit resets

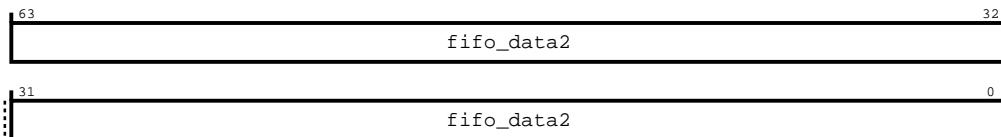
#### Usage constraints

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 8-886: por\_dtm\_fifo\_entry0-3\_2**



**Table 8-897: por\_dtm\_fifo\_entry0-3\_2 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data2	Entry data bit vector 191:128	RO	0b0

### 8.3.14.60 por\_dtm\_fifo\_entry0-3\_3

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{{index}} data.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA138 + #{32\*index}

##### Type

RO

##### Reset value

See individual bit resets

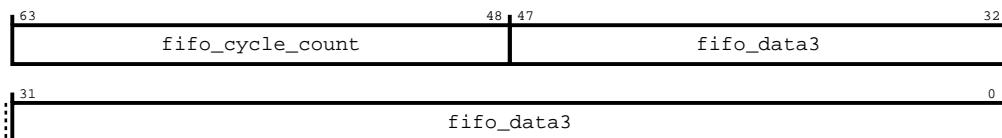
#### Usage constraints

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 8-887: por\_dtm\_fifo\_entry0-3\_3**



**Table 8-898: por\_dtm\_fifo\_entry0-3\_3 attributes**

Bits	Name	Description	Type	Reset
[63:48]	fifo_cycle_count	Entry cycle count bit vector 15:0	RO	0b0
[47:0]	fifo_data3	Entry data bit vector 239:192	RO	0b0

### 8.3.14.61 por\_dtm\_wp0-3\_config

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{{index}}.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA1A0 + #{24\*index}

##### Type

RW

##### Reset value

See individual bit resets

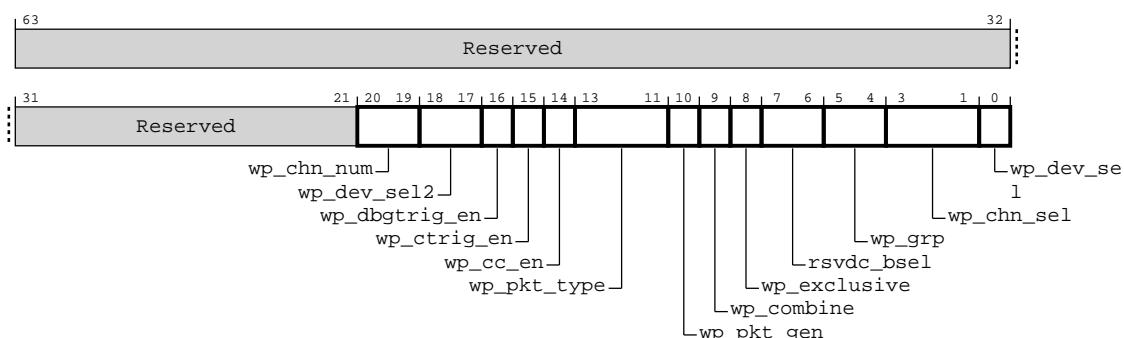
#### Usage constraints

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 8-888: por\_dtm\_wp0-3\_config**



**Table 8-899: por\_dtm\_wp0-3\_config attributes**

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[20:19]	wp_chn_num	VC number index for replicated channels in specified SMXP	RW	0b0
[18:17]	wp_dev_sel2	Upper bits for device port selection in specified SMXP	RW	0b0
[16]	wp_dbgtrig_en	Enables watchpoint debug trigger packet generation	RW	0b0

Bits	Name	Description	Type	Reset
[15]	wp_ctrig_en	Enables watchpoint cross trigger packet generation	RW	0b0
[14]	wp_cc_en	Enables inclusion of cycle count in watchpoint track packet generation	RW	0b0
[13:11]	wp_pkt_type	Trace packet type <b>0b000</b> TXNID (up to X18) <b>0b001</b> TXNID + opcode (up to X9) <b>0b010</b> TXNID + opcode + source ID + target ID (up to X4) <b>0b011</b> Reserved <b>0b100</b> Control flit <b>0b101</b> DAT flit DATA [127:0] <b>0b110</b> DAT flit DATA [255:128] <b>0b111</b> Reserved	RW	0b000
[10]	wp_pkt_gen	Enables watchpoint trace packet generation	RW	0b0
[9]	wp_combine	Enables combination of watchpoints #{{index}} and #{{index+1}}	RW	0b0
[8]	wp_exclusive	Watchpoint mode <b>0b0</b> Regular mode <b>0b1</b> Exclusive mode	RW	0b0
[7:6]	rsvdc_bsel	Byte select of RSVDC in trace packet <b>0x0</b> Select RSVDC[7:0] <b>0x1</b> Select RSVDC[15:8] <b>0x2</b> Select RSVDC[23:16] <b>0x3</b> Select RSVDC[31:24]	RW	0b0

Bits	Name	Description	Type	Reset
[5:4]	wp_grp	<p>Watchpoint register format group</p> <p><b>0x0</b> Select primary group</p> <p><b>0x1</b> Select secondary group</p> <p><b>0x2</b> Select tertiary group</p> <p><b>0x3</b> Reserved</p>	RW	0b0
[3:1]	wp_chn_sel	<p>VC selection</p> <p><b>0b000</b> Select REQ VC</p> <p><b>0b001</b> Select RSP VC</p> <p><b>0b010</b> Select SNP VC</p> <p><b>0b011</b> Select DATA VC</p> <p><b>NOTE</b> All other values are reserved.</p>	RW	0b000
[0]	wp_dev_sel	<p>Device port selection in specified SMXP</p> <p><b>0b0</b> Select device port 0</p> <p><b>0b1</b> Select device port 1</p>	RW	0b0

### 8.3.14.62 por\_dtm\_wp0-3\_val

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{{index}} comparison value.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA1A8 + #{{24 \* index}}

##### Type

RW

### Reset value

See individual bit resets

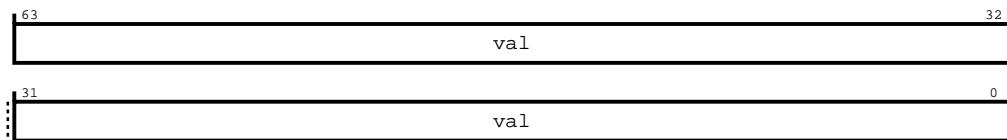
### Usage constraints

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 8-889: por\_dtm\_wp0-3\_val**



**Table 8-900: por\_dtm\_wp0-3\_val attributes**

Bits	Name	Description	Type	Reset
[63:0]	val	Refer to DTM watchpoint section for details	RW	0b0

### 8.3.14.63 por\_dtm\_wp0-3\_mask

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{{index}} comparison mask.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA1B0 + #{{24\*index}}

#### Type

RW

#### Reset value

See individual bit resets

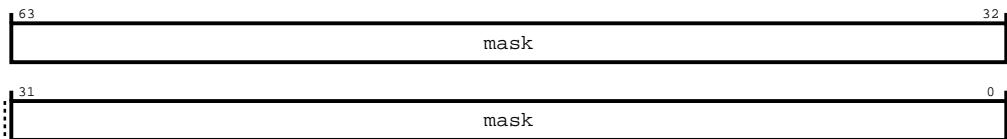
#### Usage constraints

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 8-890: por\_dtm\_wp0-3\_mask**



**Table 8-901: por\_dtm\_wp0-3\_mask attributes**

Bits	Name	Description	Type	Reset
[63:0]	mask	Refer to DTM watchpoint section for details	RW	0b0

## 8.3.14.64 por\_dtm\_pmsicr

Functions as the sampling interval counter register.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA200

#### Type

RW

#### Reset value

See individual bit resets

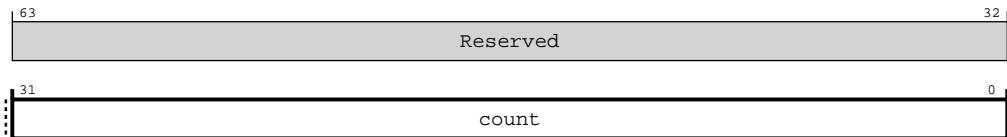
#### Usage constraints

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 8-891: por\_dtm\_pmsicr**



**Table 8-902: por\_dtm\_pmsicr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	count	Current value of sample counter	RW	0b0

### 8.3.14.65 por\_dtm\_pmsirr

Functions as the sampling interval reload register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA208

##### Type

RW

##### Reset value

See individual bit resets

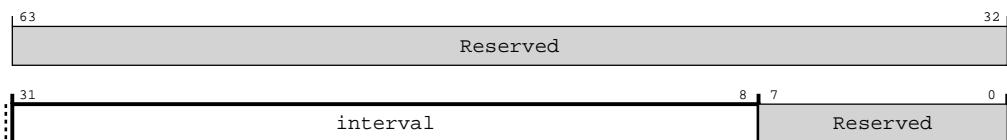
##### Usage constraints

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 8-892: por\_dtm\_pmsirr**



**Table 8-903: por\_dtm\_pmsirr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:8]	interval	Sampling interval to be reloaded	RW	0b0
[7:0]	Reserved	Reserved	RO	

### 8.3.14.66 por\_dtm\_pmu\_config

Configures the DTM PMU.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA210

##### Type

RW

##### Reset value

See individual bit resets

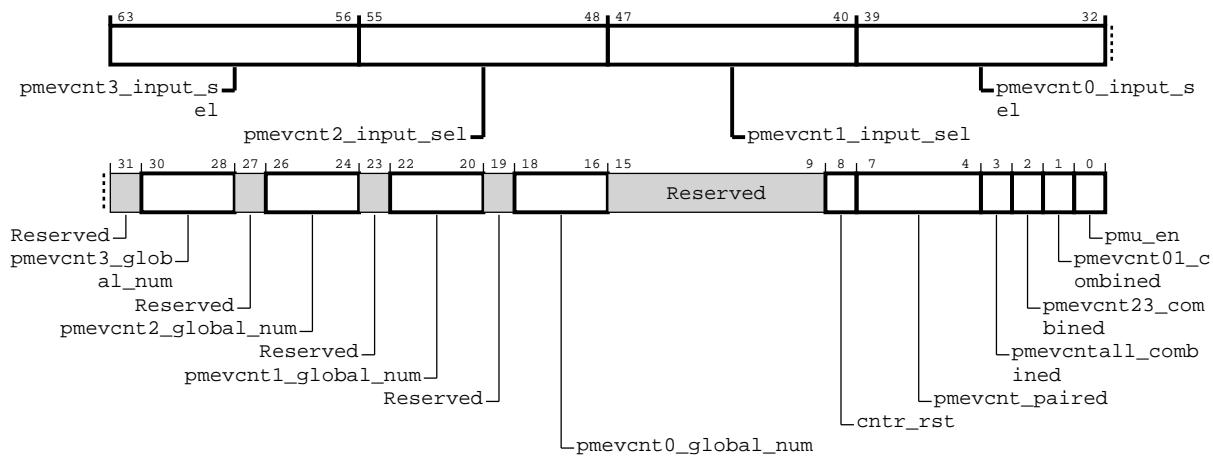
##### Usage constraints

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 8-893: por\_dtm\_pmu\_config**



**Table 8-904: por\_dtm\_pmu\_config attributes**

Bits	Name	Description	Type	Reset
[63:56]	pmevcnt3_input_sel	Source to be counted in PMU counter 3; see pmevcnt0_input_sel for encodings	RW	0b0
[55:48]	pmevcnt2_input_sel	Source to be counted in PMU counter 2; see pmevcnt0_input_sel for encodings	RW	0b0
[47:40]	pmevcnt1_input_sel	Source to be counted in PMU counter 1; see pmevcnt0_input_sel for encodings	RW	0b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	<p>Source to be counted in PMU counter</p> <p><b>0</b> Port2, Port3, Port4 and Port5 encodings are applicable when (MXP_NUM_DEV_PORT_PARAM &gt; 2 and MXP_MULTIPLE_DTM_EN_PARAM = 0)</p> <p><b>0x00</b> Watchpoint 0</p> <p><b>0x01</b> Watchpoint 1</p> <p><b>0x02</b> Watchpoint 2</p> <p><b>0x03</b> Watchpoint 3</p> <p><b>0x04</b> XP PMU Event 0</p> <p><b>0x05</b> XP PMU Event 1</p> <p><b>0x06</b> XP PMU Event 2</p> <p><b>0x07</b> XP PMU Event 3</p> <p><b>0x10</b> Port 0 Device 0 PMU Event 0</p> <p><b>0x11</b> Port 0 Device 0 PMU Event 1</p> <p><b>0x12</b> Port 0 Device 0 PMU Event 2</p> <p><b>0x13</b> Port 0 Device 0 PMU Event 3</p> <p><b>0x14</b> Port 0 Device 1 PMU Event 0</p> <p><b>0x15</b> Port 0 Device 1 PMU Event 1</p> <p><b>0x16</b> Port 0 Device 1 PMU Event 2</p> <p><b>0x17</b> Port 0 Device 1 PMU Event 3</p> <p><b>0x18</b> Port 0 Device 2 PMU Event 0</p> <p><b>0x19</b> Port 0 Device 2 PMU Event 1</p> <p><b>0x1A</b> Port 0 Device 2 PMU Event 2</p>	RW	0b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	<p>Source to be counted in PMU counter</p> <p><b>0x1B</b> Port 0 Device 2 PMU Event 3</p> <p><b>0x1C</b> Port 0 Device 3 PMU Event 0</p> <p><b>0x1D</b> Port 0 Device 3 PMU Event 1</p> <p><b>0x1E</b> Port 0 Device 3 PMU Event 2</p> <p><b>0x1F</b> Port 0 Device 3 PMU Event 3</p> <p><b>0x20</b> Port 1 Device 0 PMU Event 0</p> <p><b>0x21</b> Port 1 Device 0 PMU Event 1</p> <p><b>0x22</b> Port 1 Device 0 PMU Event 2</p> <p><b>0x23</b> Port 1 Device 0 PMU Event 3</p> <p><b>0x24</b> Port 1 Device 1 PMU Event 0</p> <p><b>0x25</b> Port 1 Device 1 PMU Event 1</p> <p><b>0x26</b> Port 1 Device 1 PMU Event 2</p> <p><b>0x27</b> Port 1 Device 1 PMU Event 3</p> <p><b>0x28</b> Port 1 Device 2 PMU Event 0</p> <p><b>0x29</b> Port 1 Device 2 PMU Event 1</p> <p><b>0x2A</b> Port 1 Device 2 PMU Event 2</p> <p><b>0x2B</b> Port 1 Device 2 PMU Event 3</p> <p><b>0x2C</b> Port 1 Device 3 PMU Event 0</p> <p><b>0x2D</b> Port 1 Device 3 PMU Event 1</p> <p><b>0x2E</b> Port 1 Device 3 PMU Event 2</p> <p><b>0x2F</b> Port 1 Device 3 PMU Event 3</p> <p><b>0x30</b> Port 2 Device 0 PMU Event 0</p>	RW	0b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	Source to be counted in PMU counter	RW	0b0
		<b>0x31</b> Port 2 Device 0 PMU Event 1		
		<b>0x32</b> Port 2 Device 0 PMU Event 2		
		<b>0x33</b> Port 2 Device 0 PMU Event 3		
		<b>0x34</b> Port 2 Device 1 PMU Event 0		
		<b>0x35</b> Port 2 Device 1 PMU Event 1		
		<b>0x36</b> Port 2 Device 1 PMU Event 2		
		<b>0x37</b> Port 2 Device 1 PMU Event 3		
		<b>0x38</b> Port 2 Device 2 PMU Event 0		
		<b>0x39</b> Port 2 Device 2 PMU Event 1		
		<b>0x3A</b> Port 2 Device 2 PMU Event 2		
		<b>0x3B</b> Port 2 Device 2 PMU Event 3		
		<b>0x3C</b> Port 2 Device 3 PMU Event 0		
		<b>0x3D</b> Port 2 Device 3 PMU Event 1		
		<b>0x3E</b> Port 2 Device 3 PMU Event 2		
		<b>0x3F</b> Port 2 Device 3 PMU Event 3		
		<b>0x40</b> Port 3 Device 0 PMU Event 0		
		<b>0x41</b> Port 3 Device 0 PMU Event 1		
		<b>0x42</b> Port 3 Device 0 PMU Event 2		
		<b>0x43</b> Port 3 Device 0 PMU Event 3		
		<b>0x44</b> Port 3 Device 1 PMU Event 0		
		<b>0x45</b> Port 3 Device 1 PMU Event 1		
		<b>0x46</b> Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved. Non-Confidential Port 3 Device 1 PMU Event 2		
		<b>0x47</b> Port 3 Device 1 PMU Event 3		
		<b>0x48</b>		

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	Source to be counted in PMU counter	RW	0b0
	<b>0x51</b>	Port 4 Device 0 PMU Event 1		
	<b>0x52</b>	Port 4 Device 0 PMU Event 2		
	<b>0x53</b>	Port 4 Device 0 PMU Event 3		
	<b>0x54</b>	Port 4 Device 1 PMU Event 0		
	<b>0x55</b>	Port 4 Device 1 PMU Event 1		
	<b>0x56</b>	Port 4 Device 1 PMU Event 2		
	<b>0x57</b>	Port 4 Device 1 PMU Event 3		
	<b>0x58</b>	Port 4 Device 2 PMU Event 0		
	<b>0x59</b>	Port 4 Device 2 PMU Event 1		
	<b>0x5A</b>	Port 4 Device 2 PMU Event 2		
	<b>0x5B</b>	Port 4 Device 2 PMU Event 3		
	<b>0x5C</b>	Port 4 Device 3 PMU Event 0		
	<b>0x5D</b>	Port 4 Device 3 PMU Event 1		
	<b>0x5E</b>	Port 4 Device 3 PMU Event 2		
	<b>0x5F</b>	Port 4 Device 3 PMU Event 3		
	<b>0x60</b>	Port 5 Device 0 PMU Event 0		
	<b>0x61</b>	Port 5 Device 0 PMU Event 1		
	<b>0x62</b>	Port 5 Device 0 PMU Event 2		
	<b>0x63</b>	Port 5 Device 0 PMU Event 3		
	<b>0x64</b>	Port 5 Device 1 PMU Event 0		
	<b>0x65</b>	Port 5 Device 1 PMU Event 1		
	<b>0x66</b>	Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved. Non-Confidential Port 5 Device 1 PMU Event 2		
	<b>0x67</b>	Port 5 Device 1 PMU Event 3		
	<b>0x68</b>			

Bits	Name	Description	Type	Reset
[31]	Reserved	Reserved	RO	
[30:28]	pmevcnt3_global_num	Global counter to pair with PMU counter 3; see pmevcnt0_global_num for encodings	RW	0b0
[27]	Reserved	Reserved	RO	
[26:24]	pmevcnt2_global_num	Global counter to pair with PMU counter 2; see pmevcnt0_global_num for encodings	RW	0b0
[23]	Reserved	Reserved	RO	
[22:20]	pmevcnt1_global_num	Global counter to pair with PMU counter 1; see pmevcnt0_global_num for encodings	RW	0b0
[19]	Reserved	Reserved	RO	
[18:16]	pmevcnt0_global_num	Global counter to pair with PMU counter 0	RW	0b0
		<b>0b000</b> Global PMU event counter A		
		<b>0b001</b> Global PMU event counter B		
		<b>0b010</b> Global PMU event counter C		
		<b>0b011</b> Global PMU event counter D		
		<b>0b100</b> Global PMU event counter E		
		<b>0b101</b> Global PMU event counter F		
		<b>0b110</b> Global PMU event counter G		
		<b>0b111</b> Global PMU event counter H		
[15:9]	Reserved	Reserved	RO	
[8]	cntr_RST	Enables clearing of live counters upon assertion of snapshot	RW	0b0
[7:4]	pmevcnt_paired	PMU local counter paired with global counter	RW	0b0
[3]	pmevcntall_combined	Enables combination of all PMU counters (0, 1, 2, 3)	RW	0b0
		<b>NOTE</b> When set, pmevcnt01_combined and pmevcnt23_combined have no effect.		
[2]	pmevcnt23_combined	Enables combination of PMU counters 2 and 3	RW	0b0
[1]	pmevcnt01_combined	Enables combination of PMU counters 0 and 1	RW	0b0
[0]	pmu_en	DTM PMU enable	RW	0b0
		<b>NOTE</b> All other fields in this register are valid only if this bit is set.		

### 8.3.14.67 por\_dtm\_pmevcnt

Contains all PMU event counters (0, 1, 2, 3).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA220

##### Type

RW

##### Reset value

See individual bit resets

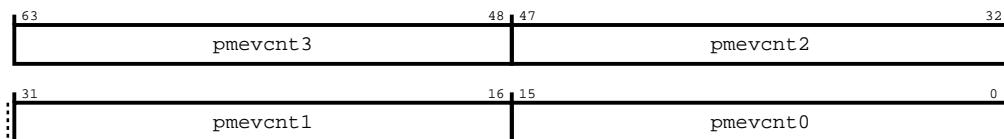
##### Usage constraints

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 8-894: por\_dtm\_pmevcnt**



**Table 8-905: por\_dtm\_pmevcnt attributes**

Bits	Name	Description	Type	Reset
[63:48]	pmevcnt3	PMU event counter 3	RW	0x0000
[47:32]	pmevcnt2	PMU event counter 2	RW	0x0000
[31:16]	pmevcnt1	PMU event counter 1	RW	0x0000
[15:0]	pmevcnt0	PMU event counter 0	RW	0x0000

### 8.3.14.68 por\_dtm\_pmevcntsr

Functions as the PMU event counter shadow register for all counters (0, 1, 2, 3).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA240

##### Type

RW

##### Reset value

See individual bit resets

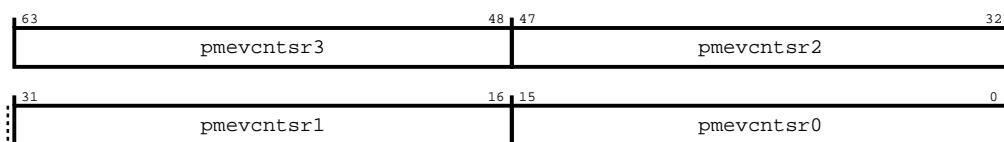
##### Usage constraints

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 8-895: por\_dtm\_pmevcntsr**



**Table 8-906: por\_dtm\_pmevcntsr attributes**

Bits	Name	Description	Type	Reset
[63:48]	pmevcntsr3	PMU event counter 3 shadow register	RW	0x0000
[47:32]	pmevcntsr2	PMU event counter 2 shadow register	RW	0x0000
[31:16]	pmevcntsr1	PMU event counter 1 shadow register	RW	0x0000
[15:0]	pmevcntsr0	PMU event counter 0 shadow register	RW	0x0000

### 8.3.14.69 por\_dtm\_control\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the DTM control register.



There are a maximum of 4 DTM registers, based on `MXP_MULTIPLE_DTM_EN` and `MXP_NUM_DEV_PORT` value. Each successive DTM register will be at the next 512-byte address boundary and is named with the suffix corresponding to the DT register number.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

`0xA100 + #{512*index}`

### Type

RW

### Reset value

See individual bit resets

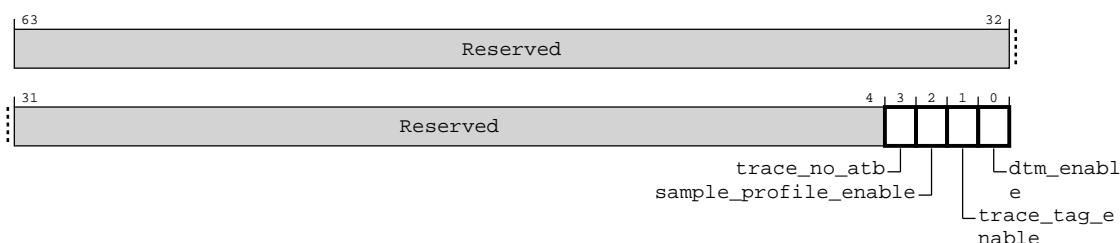
### Usage constraints

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 8-896: por\_dtm\_control\_dt1-3**



**Table 8-907: por\_dtm\_control\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	trace_no_atb	When set, trace packet is not delivered out of ATB, and FIFO entry holds the first trace packet	RW	0b0
[2]	sample_profile_enable	Enables sample profile function	RW	0b0
[1]	trace_tag_enable	Watchpoint trace tag enable  <b>0b1</b> Trace tag enabled  <b>0b0</b> No trace tag	RW	0b0
[0]	dtm_enable	Enables debug watchpoint and PMU function; prior to writing this bit, all other DT configuration registers must be programmed; once this bit is set, other DT configuration registers must not be modified	RW	0b0

### 8.3.14.70 por\_dtm\_fifo\_entry\_ready\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Controls status of DTM FIFO entries.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA118 + #{512\*index}

##### Type

W1C

##### Reset value

See individual bit resets

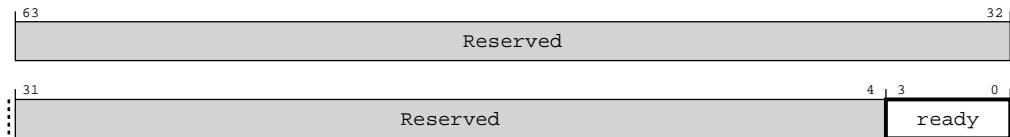
##### Usage constraints

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 8-897: por\_dtm\_fifo\_entry\_ready\_dt1-3**



**Table 8-908: por\_dtm\_fifo\_entry\_ready\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3:0]	ready	Indicates which DTM FIFO entries are ready; write a 1 to clear <b>Bit[3]</b> Entry 3 ready when set <b>Bit[2]</b> Entry 2 ready when set <b>Bit[1]</b> Entry 1 ready when set <b>Bit[0]</b> Entry 0 ready when set	W1C	0b0

### 8.3.14.71 por\_dtm\_fifo\_entry0-11%4\_0\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{{index%4}} data.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA120 + #{{32\*(index%4)}} + #{{512\*((index/4)+1)}}

##### 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 8-898: por\_dtm\_fifo\_entry0-11%4\_0\_dt(0-11/4)+1**



**Table 8-909: por\_dtm\_fifo\_entry0-11%4\_0\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data0	Entry data bit vector 63:0	RO	0b0

## 8.3.14.72 por\_dtm\_fifo\_entry0-11%4\_1\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{{index%4}} data.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xA128 + #{{32(index%4)}} + #{{512((index/4)+1)}}

#### 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 8-899: por\_dtm\_fifo\_entry0-11%4\_1\_dt(0-11/4)+1**



**Table 8-910: por\_dtm\_fifo\_entry0-11%4\_1\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data1	Entry data bit vector 127:64	RO	0b0

### 8.3.14.73 por\_dtm\_fifo\_entry0-11%4\_2\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{{index%4}} data.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA130 + #{{32(index%4)}} + #{{512((index/4)+1)}}

##### 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 8-900: por\_dtm\_fifo\_entry0-11%4\_2\_dt(0-11/4)+1**



**Table 8-911: por\_dtm\_fifo\_entry0-11%4\_2\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	fifo_data2	Entry data bit vector 191:128	RO	0b0

### 8.3.14.74 por\_dtm\_fifo\_entry0-11%4\_3\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{{index%4}} data.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA138 + #{{32(index%4)}} + #{{512((index/4)+1)}}

##### 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 8-901: por\_dtm\_fifo\_entry0-11%4\_3\_dt(0-11/4)+1**



**Table 8-912: por\_dtm\_fifo\_entry0-11%4\_3\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:48]	fifo_cycle_count	Entry cycle count bit vector 15:0	RO	0b0
[47:0]	fifo_data3	Entry data bit vector 239:192	RO	0b0

### 8.3.14.75 por\_dtm\_wp0-11%4\_config\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{{index%4}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA1A0 + #{{24(index%4)}} + #{{512((index/4)+1)}}

##### Type

RW

##### Reset value

See individual bit resets

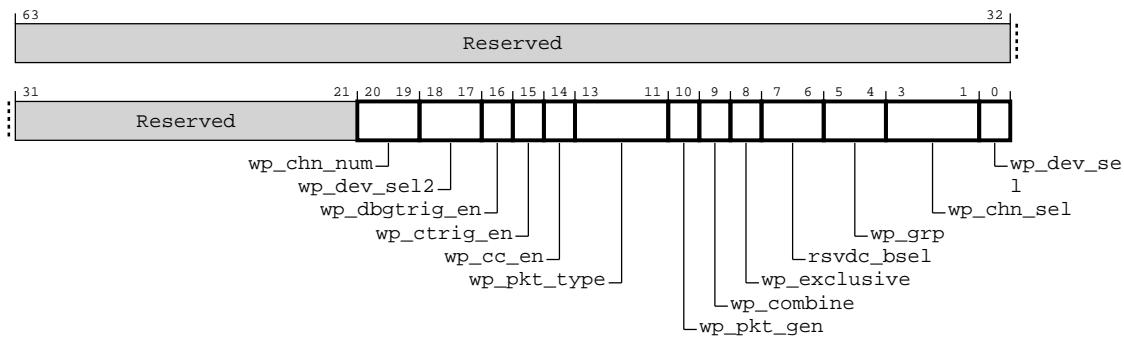
##### Usage constraints

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 8-902: por\_dtm\_wp0-11%4\_config\_dt(0-11/4)+1**



**Table 8-913: por\_dtm\_wp0-11%4\_config\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[20:19]	wp_chn_num	VC number index for replicated channels in specified SMXP	RW	0b0
[18:17]	wp_dev_sel2	Upper bits for device port selection in specified SMXP	RW	0b0
[16]	wp_dbgtrig_en	Enables watchpoint debug trigger packet generation	RW	0b0
[15]	wp_ctrig_en	Enables watchpoint cross trigger packet generation	RW	0b0
[14]	wp_cc_en	Enables inclusion of cycle count in watchpoint track packet generation	RW	0b0
[13:11]	wp_pkt_type	Trace packet type  <b>0b000</b> TXNID (up to X18)  <b>0b001</b> TXNID + opcode (up to X9)  <b>0b010</b> TXNID + opcode + source ID + target ID (up to X4)  <b>0b011</b> Reserved  <b>0b100</b> Control flit  <b>0b101</b> DAT flit DATA [127:0]  <b>0b110</b> DAT flit DATA [255:128]  <b>0b111</b> Reserved	RW	0b000
[10]	wp_pkt_gen	Enables watchpoint trace packet generation	RW	0b0
[9]	wp_combine	Enables combination of watchpoints #{{index%4}} and #{{(index%4)+1}}	RW	0b0

Bits	Name	Description	Type	Reset
[8]	wp_exclusive	Watchpoint mode <b>0b0</b> Regular mode <b>0b1</b> Exclusive mode	RW	0b0
[7:6]	rsvdc_bsel	Byte select of RSVDC in trace packet <b>0x0</b> Select RSVDC[7:0] <b>0x1</b> Select RSVDC[15:8] <b>0x2</b> Select RSVDC[23:16] <b>0x3</b> Select RSVDC[31:24]	RW	0b0
[5:4]	wp_grp	Watchpoint register format group <b>0x0</b> Select primary group <b>0x1</b> Select secondary group <b>0x2</b> Select tertiary group <b>0x3</b> Reserved	RW	0b0
[3:1]	wp_chn_sel	VC selection <b>0b000</b> Select REQ VC <b>0b001</b> Select RSP VC <b>0b010</b> Select SNP VC <b>0b011</b> Select DATA VC <b>NOTE</b> All other values are reserved.	RW	0b000
[0]	wp_dev_sel	Device port selection in specified SMXP <b>0b0</b> Select device port 0 <b>0b1</b> Select device port 1	RW	0b0

### 8.3.14.76 por\_dtm\_wp0-11%4\_val\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{{index%4}} comparison value.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$0xA1A8 + \{24(index \% 4)\} + \{512((index / 4) + 1)\}$

##### Type

RW

##### Reset value

See individual bit resets

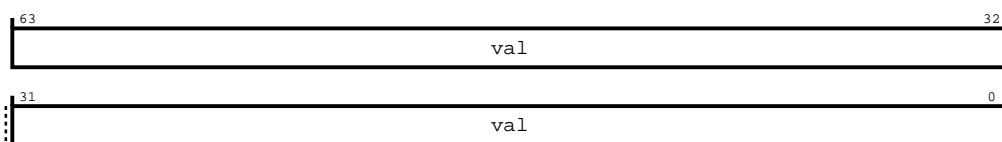
##### Usage constraints

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 8-903: por\_dtm\_wp0-11%4\_val\_dt(0-11/4)+1**



**Table 8-914: por\_dtm\_wp0-11%4\_val\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	val	Refer to DTM watchpoint section for details	RW	0b0

### 8.3.14.77 por\_dtm\_wp0-11%4\_mask\_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{{index%4}} comparison mask.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$0xA1B0 + \{24(index \% 4)\} + \{512((index / 4) + 1)\}$

##### Type

RW

##### Reset value

See individual bit resets

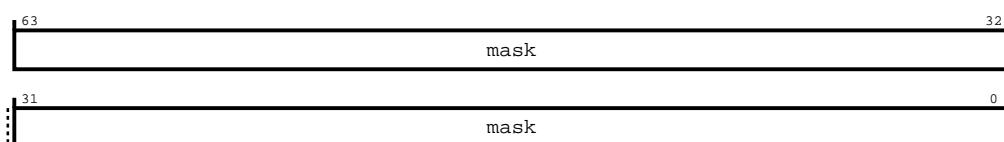
##### Usage constraints

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 8-904: por\_dtm\_wp0-11%4\_mask\_dt(0-11/4)+1**



**Table 8-915: por\_dtm\_wp0-11%4\_mask\_dt(0-11/4)+1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	mask	Refer to DTM watchpoint section for details	RW	0b0

### 8.3.14.78 por\_dtm\_pmsicr\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the sampling interval counter register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA200 + #{512\*index}

##### Type

RW

##### Reset value

See individual bit resets

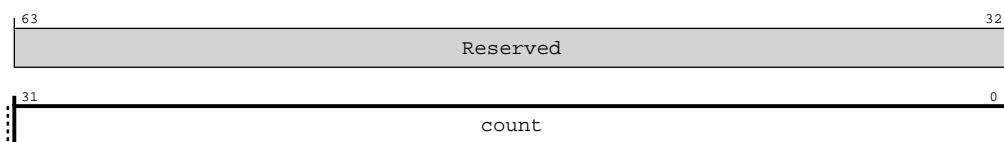
##### Usage constraints

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 8-905: por\_dtm\_pmsicr\_dt1-3**



**Table 8-916: por\_dtm\_pmsicr\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	count	Current value of sample counter	RW	0b0

### 8.3.14.79 por\_dtm\_pmsirr\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the sampling interval reload register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA208 + #{512\*index}

##### Type

RW

##### Reset value

See individual bit resets

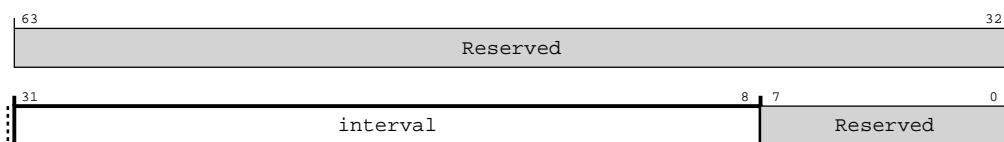
##### Usage constraints

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 8-906: por\_dtm\_pmsirr\_dt1-3**



**Table 8-917: por\_dtm\_pmsirr\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:8]	interval	Sampling interval to be reloaded	RW	0b0
[7:0]	Reserved	Reserved	RO	

### 8.3.14.80 por\_dtm\_pmu\_config\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Configures the DTM PMU.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA210 + #{512\*index}

##### Type

RW

##### Reset value

See individual bit resets

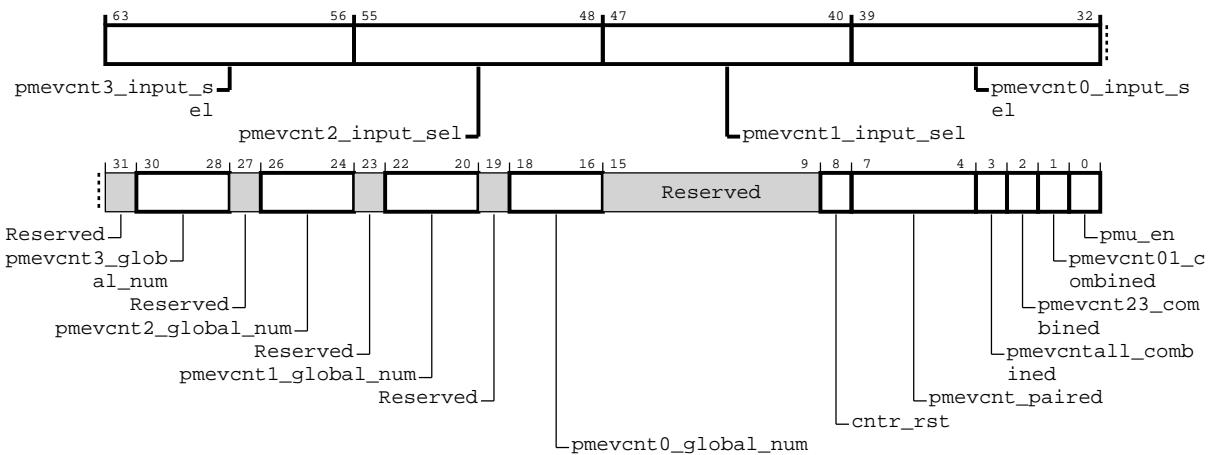
##### Usage constraints

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 8-907: por\_dtm\_pmu\_config\_dt1-3**



**Table 8-918: por\_dtm\_pmu\_config\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:56]	pmevcnt3_input_sel	Source to be counted in PMU counter 3; see pmevcnt0_input_sel for encodings	RW	0b0

Bits	Name	Description	Type	Reset
[55:48]	pmevcnt2_input_sel	Source to be counted in PMU counter 2; see pmevcnt0_input_sel for encodings	RW	0b0
[47:40]	pmevcnt1_input_sel	Source to be counted in PMU counter 1; see pmevcnt0_input_sel for encodings	RW	0b0
[39:32]	pmevcnt0_input_sel	Source to be counted in PMU counter	RW	0b0
	<b>0</b>	Supports 2 Ports		
	<b>DT1</b>	P2 and P3,		
	<b>DT2</b>	P4 and P5 when (MXP_NUM_DEV_PORT_PARAM > 2 and MXP_MULTIPLE_DTM_EN_PARAM = 1)		
	<b>0x00</b>	Watchpoint 0		
	<b>0x01</b>	Watchpoint 1		
	<b>0x02</b>	Watchpoint 2		
	<b>0x03</b>	Watchpoint 3		
	<b>0x04</b>	XP PMU Event 0		
	<b>0x05</b>	XP PMU Event 1		
	<b>0x06</b>	XP PMU Event 2		
	<b>0x07</b>	XP PMU Event 3		
	<b>0x10</b>	Port 0 Device 0 PMU Event 0		
	<b>0x11</b>	Port 0 Device 0 PMU Event 1		
	<b>0x12</b>	Port 0 Device 0 PMU Event 2		
	<b>0x13</b>	Port 0 Device 0 PMU Event 3		
	<b>0x14</b>	Port 0 Device 1 PMU Event 0		
	<b>0x15</b>	Port 0 Device 1 PMU Event 1		
	<b>0x16</b>	Port 0 Device 1 PMU Event 2		
	<b>0x17</b>	Port 0 Device 1 PMU Event 3		

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	<p>Source to be counted in PMU counter</p> <p><b>0x18</b> Port 0 Device 2 PMU Event 0</p> <p><b>0x19</b> Port 0 Device 2 PMU Event 1</p> <p><b>0x1A</b> Port 0 Device 2 PMU Event 2</p> <p><b>0x1B</b> Port 0 Device 2 PMU Event 3</p> <p><b>0x1C</b> Port 0 Device 3 PMU Event 0</p> <p><b>0x1D</b> Port 0 Device 3 PMU Event 1</p> <p><b>0x1E</b> Port 0 Device 3 PMU Event 2</p> <p><b>0x1F</b> Port 0 Device 3 PMU Event 3</p> <p><b>0x20</b> Port 1 Device 0 PMU Event 0</p> <p><b>0x21</b> Port 1 Device 0 PMU Event 1</p> <p><b>0x22</b> Port 1 Device 0 PMU Event 2</p> <p><b>0x23</b> Port 1 Device 0 PMU Event 3</p> <p><b>0x24</b> Port 1 Device 1 PMU Event 0</p> <p><b>0x25</b> Port 1 Device 1 PMU Event 1</p> <p><b>0x26</b> Port 1 Device 1 PMU Event 2</p> <p><b>0x27</b> Port 1 Device 1 PMU Event 3</p> <p><b>0x28</b> Port 1 Device 2 PMU Event 0</p> <p><b>0x29</b> Port 1 Device 2 PMU Event 1</p> <p><b>0x2A</b> Port 1 Device 2 PMU Event 2</p> <p><b>0x2B</b> Port 1 Device 2 PMU Event 3</p> <p><b>0x2C</b> Port 1 Device 3 PMU Event 0</p> <p><b>0x2D</b> Port 1 Device 3 PMU Event 1</p> <p><b>0x2E</b> Port 1 Device 3 PMU Event 2</p> <p><b>0x2F</b></p>	RW	0b0

Bits	Name	Description	Type	Reset
[31]	Reserved	Reserved	RO	
[30:28]	pmevcnt3_global_num	Global counter to pair with PMU counter 3; see pmevcnt0_global_num for encodings	RW	0b0
[27]	Reserved	Reserved	RO	
[26:24]	pmevcnt2_global_num	Global counter to pair with PMU counter 2; see pmevcnt0_global_num for encodings	RW	0b0
[23]	Reserved	Reserved	RO	
[22:20]	pmevcnt1_global_num	Global counter to pair with PMU counter 1; see pmevcnt0_global_num for encodings	RW	0b0
[19]	Reserved	Reserved	RO	
[18:16]	pmevcnt0_global_num	Global counter to pair with PMU counter 0	RW	0b0
		<b>0b000</b> Global PMU event counter A		
		<b>0b001</b> Global PMU event counter B		
		<b>0b010</b> Global PMU event counter C		
		<b>0b011</b> Global PMU event counter D		
		<b>0b100</b> Global PMU event counter E		
		<b>0b101</b> Global PMU event counter F		
		<b>0b110</b> Global PMU event counter G		
		<b>0b111</b> Global PMU event counter H		
[15:9]	Reserved	Reserved	RO	
[8]	cntr_RST	Enables clearing of live counters upon assertion of snapshot	RW	0b0
[7:4]	pmevcnt_paired	PMU local counter paired with global counter	RW	0b0
[3]	pmevcntall_combined	Enables combination of all PMU counters (0, 1, 2, 3)	RW	0b0
		<b>NOTE</b> When set, pmevcnt01_combined and pmevcnt23_combined have no effect.		
[2]	pmevcnt23_combined	Enables combination of PMU counters 2 and 3	RW	0b0
[1]	pmevcnt01_combined	Enables combination of PMU counters 0 and 1	RW	0b0
[0]	pmu_en	DTM PMU enable	RW	0b0
		<b>NOTE</b> All other fields in this register are valid only if this bit is set.		

### 8.3.14.81 por\_dtm\_pmevcnt\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Contains all PMU event counters (0, 1, 2, 3).

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA220 + #{512\*index}

##### Type

RW

##### Reset value

See individual bit resets

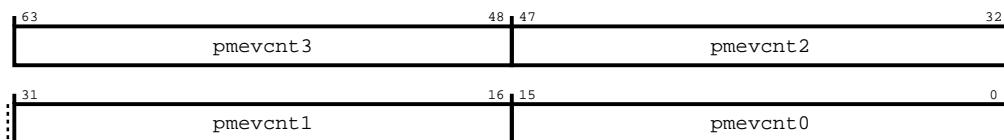
##### Usage constraints

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 8-908: por\_dtm\_pmevcnt\_dt1-3**



**Table 8-919: por\_dtm\_pmevcnt\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:48]	pmevcnt3	PMU event counter 3	RW	0x0000
[47:32]	pmevcnt2	PMU event counter 2	RW	0x0000
[31:16]	pmevcnt1	PMU event counter 1	RW	0x0000
[15:0]	pmevcnt0	PMU event counter 0	RW	0x0000

### 8.3.14.82 por\_dtm\_pmevcntsr\_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the PMU event counter shadow register for all counters (0, 1, 2, 3).

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA240 + #{512\*index}

##### Type

RW

##### Reset value

See individual bit resets

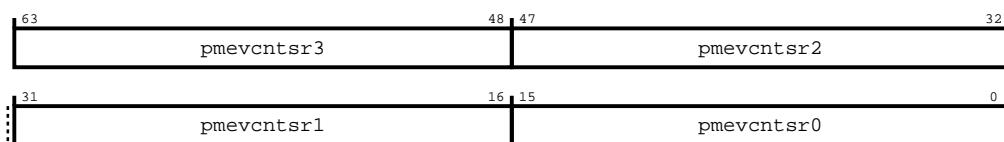
##### Usage constraints

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 8-909: por\_dtm\_pmevcntsr\_dt1-3**



**Table 8-920: por\_dtm\_pmevcntsr\_dt1-3 attributes**

Bits	Name	Description	Type	Reset
[63:48]	pmevcntsr3	PMU event counter 3 shadow register	RW	0x0000
[47:32]	pmevcntsr2	PMU event counter 2 shadow register	RW	0x0000
[31:16]	pmevcntsr1	PMU event counter 1 shadow register	RW	0x0000
[15:0]	pmevcntsr0	PMU event counter 0 shadow register	RW	0x0000

### 8.3.14.83 por\_mxp\_multi\_mesh\_chn\_sel\_0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Functions as the CHI VC channel select per Target register in Multi-Mesh Channel structure.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.multi\_mesh\_ctl

##### Secure group override

por\_mxp\_scr.multi\_mesh\_ctl

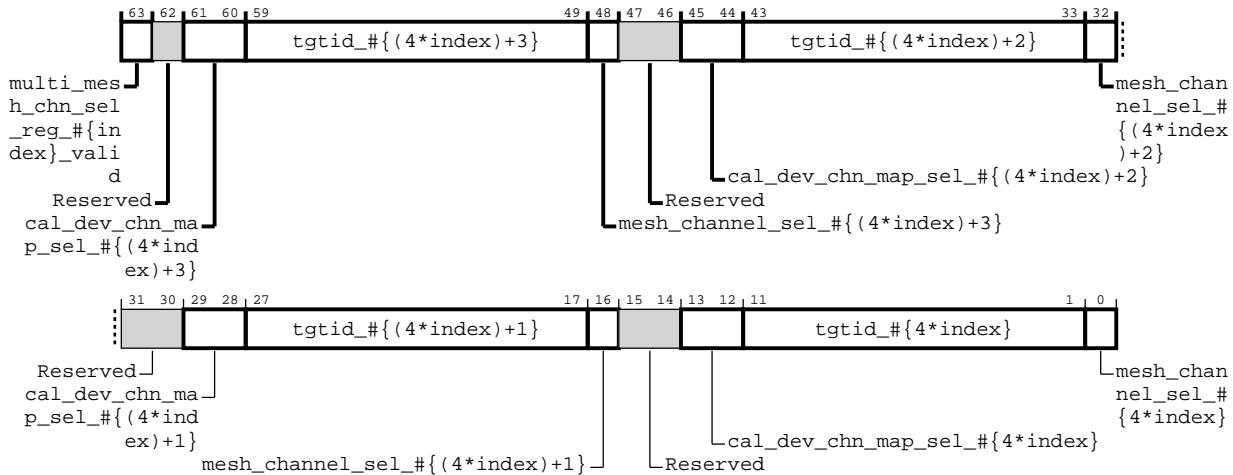
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.multi\_mesh\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.multi\_mesh\_ctl bit and por\_mxp\_rcr.multi\_mesh\_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 8-910: por\_mxp\_multi\_mesh\_chn\_sel\_0-15**



**Table 8-921: por\_mxp\_multi\_mesh\_chn\_sel\_0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	multi_mesh_chn_sel_reg_{index}_valid	Indicates that multi mesh CHI VC channel configured for the targets specified in this register.	RW	0b0
[62]	Reserved	Reserved	RO	
[61:60]	cal_dev_chn_map_sel_{(4*index)+3}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field): <b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), <ul style="list-style-type: none"> <li>CAL2: DEV0, DEV1 are mapped to same channel,</li> </ul> <b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <b>0b10</b> Reserved, <b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[59:49]	tgtid_{(4*index)+3}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0

Bits	Name	Description	Type	Reset
[48]	mesh_channel_sel_#{(4*index)+3}	CHI VC channel <b>select</b> 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[47:46]	Reserved	Reserved	RO	
[45:44]	cal_dev_chn_map_sel_#{(4*index)+2}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):  <b>0b00</b> CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), <ul style="list-style-type: none"><li>CAL2:DEV0, DEV1 are mapped to same channel,</li></ul> <b>0b01</b> CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, <ul style="list-style-type: none"><li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li></ul> <b>0b10</b> Reserved,  <b>0b11</b> Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, <ul style="list-style-type: none"><li>CAL2: DEV0,DEV1 can be mapped to different channel,</li><li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li></ul>	RW	0b0
[43:33]	tgtid_#{(4*index)+2}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[32]	mesh_channel_sel_#{(4*index)+2}	CHI VC channel <b>select</b> 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[31:30]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[29:28]	cal_dev_chn_map_sel_#{(4*index)+1}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b>            CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>• CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b>            CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>• CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b>            Reserved,</p> <p><b>0b11</b>            Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>• CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>• CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[27:17]	tgtid_#{(4*index)+1}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[16]	mesh_channel_sel_#{(4*index)+1}	CHI VC channel <b>select</b> 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	0b0
[15:14]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[13:12]	cal_dev_chn_map_sel_{4*index}	<p>Channel Map select for target devices behind CAL (associated with the corresponding tgtid field):</p> <p><b>0b00</b>            CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with &gt; 2 device ports per XP),</p> <ul style="list-style-type: none"> <li>CAL2:DEV0, DEV1 are mapped to same channel,</li> </ul> <p><b>0b01</b>            CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel,</li> </ul> <p><b>0b10</b>            Reserved,</p> <p><b>0b11</b>            Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below,</p> <ul style="list-style-type: none"> <li>CAL2: DEV0,DEV1 can be mapped to different channel,</li> <li>CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel</li> </ul>	RW	0b0
[11:1]	tgtid_{4*index}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	0b0
[0]	mesh_channel_sel_{4*index}	<p>CHI VC channel</p> <p><b>select</b>            1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected</p>	RW	0b0

### 8.3.14.84 por\_mxp\_multi\_mesh\_chn\_ctrl

Functions as the control register for Target based channel selection in Multi-Mesh Channel structure.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xC00 + 0x80

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.multi\_mesh\_ctl

### Secure group override

por\_mxp\_scr.multi\_mesh\_ctl

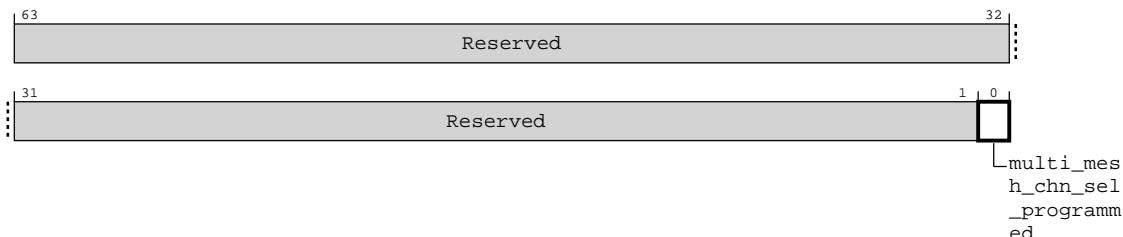
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.multi\_mesh\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.multi\_mesh\_ctl bit and por\_mxp\_rcr.multi\_mesh\_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 8-911: por\_mxp\_multi\_mesh\_chn\_ctrl**



**Table 8-922: por\_mxp\_multi\_mesh\_chn\_ctrl attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	multi_mesh_chn_sel_programmed	Indicates that multi CHI VC channel configured for all the targets specified in the channel select registers.	RW	0b0

### 8.3.14.85 por\_mxp\_xy\_override\_sel\_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Functions as SRC-TGT pair whose X-Y route path can be overridden (for Non-XY Route feature per Source-Target pair) per XP.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xC90 + #{8\*index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.xy\_override\_ctl

### Secure group override

por\_mxp\_scr.xy\_override\_ctl

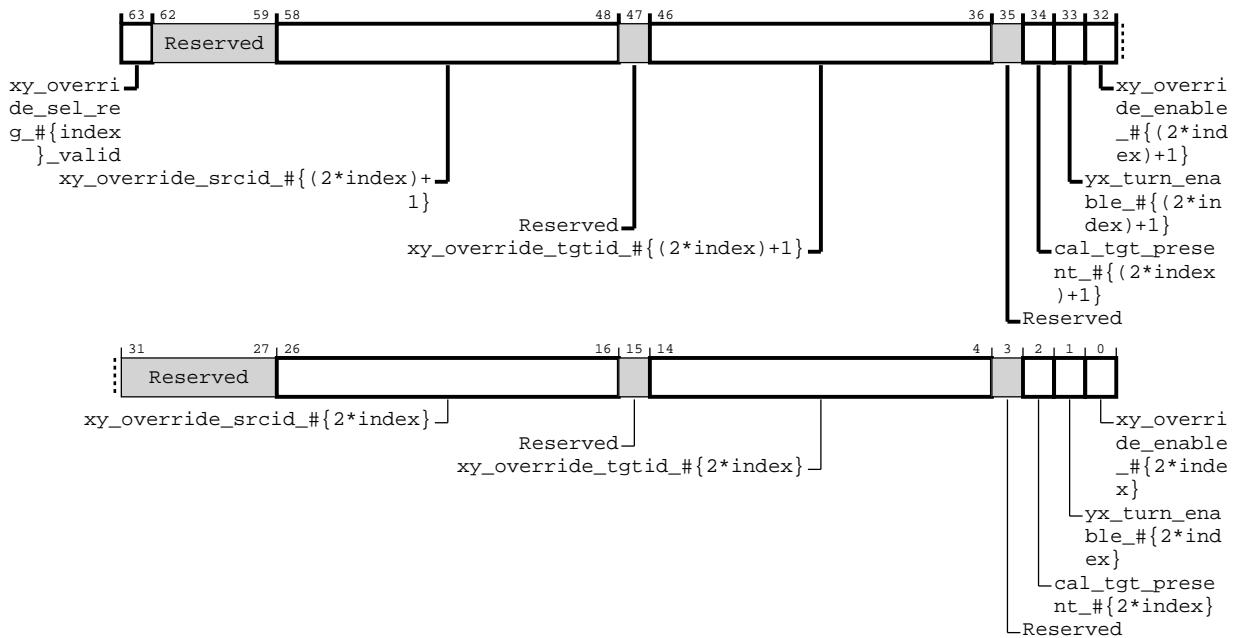
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.xy\_override\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.xy\_override\_ctl bit and por\_mxp\_rcr.xy\_override\_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 8-912: por\_mxp\_xy\_override\_sel\_0-7**



**Table 8-923: por\_mxp\_xy\_override\_sel\_0-7 attributes**

Bits	Name	Description	Type	Reset
[63]	xy_override_sel_reg_{index}_valid	Indicates that Source-Target pairs whose X-Y route path can be overridden are configured in this register.	RW	0b0
[62:59]	Reserved	Reserved	RO	-
[58:48]	xy_override_srcid_{\{2*index\}+1}	11-bit Source ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overridden.	RW	0b0
[47]	Reserved	Reserved	RO	-
[46:36]	xy_override_tgtid_{\{2*index\}+1}	11-bit Target ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overridden.	RW	0b0
[35]	Reserved	Reserved	RO	-
[34]	cal_tgt_present_{\{2*index\}+1}	CAL TGT Presence Indication for XY Route Override of all devices behind CAL  1 - CAL4 TGT Present for XY Route Override of all devices behind CAL  0 - CAL2 TGT or no CAL TGT Present for XY Route Override of all devices behind CAL.	RW	0b0
[33]	yx_turn_enable_{\{2*index\}+1}	Y-X Turn Enable  1 - Y-X Turn enabled for associated Source-Target Pair,  0 - Y-X Turn disabled	RW	0b0

Bits	Name	Description	Type	Reset
[32]	xy_override_enable_{2*index}+1	X-Y Route Override Enable  1 - X-Y Route override enabled for associated Source-Target Pair  0 - X-Y Route override disabled	RW	0b0
[31:27]	Reserved	Reserved	RO	-
[26:16]	xy_override_srcid_{2*index}	11-bit Source ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overridden.	RW	0b0
[15]	Reserved	Reserved	RO	-
[14:4]	xy_override_tgtid_{2*index}	11-bit Target ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overridden.	RW	0b0
[3]	Reserved	Reserved	RO	-
[2]	cal_tgt_present_{2*index}	CAL TGT Presence Indication for XY Route Override of all devices behind CAL  1 - CAL4 TGT Present for XY Route Override of all devices behind CAL,  0 - CAL2 TGT or no CAL TGT Present for XY Route Override of all devices behind CAL.	RW	0b0
[1]	yx_turn_enable_{2*index}	Y-X Turn Enable  1 - Y-X Turn enabled for associated Source-Target Pair  0 - Y-X Turn disabled	RW	0b0
[0]	xy_override_enable_{2*index}	X-Y Route Override Enable  1 - X-Y Route override enabled for associated Source-Target Pair  0 - X-Y Route override disabled	RW	0b0

### 8.3.14.86 por\_mxp\_p0-5\_pa2setaddr\_slc

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the control register of PA to SetAddr and vice versa conversion for HNF-SLC on XP port #{index}. NOTE: There will be max. of 6 MXP Port registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

## Address offset

$0x\text{CD}0 + \#\{32*\text{index}\}$

## Type

RW

## Reset value

See individual bit resets

## Root group override

`por_mxp_rcr.pa2setaddr_ctl`

## Secure group override

`por_mxp_scr.pa2setaddr_ctl`

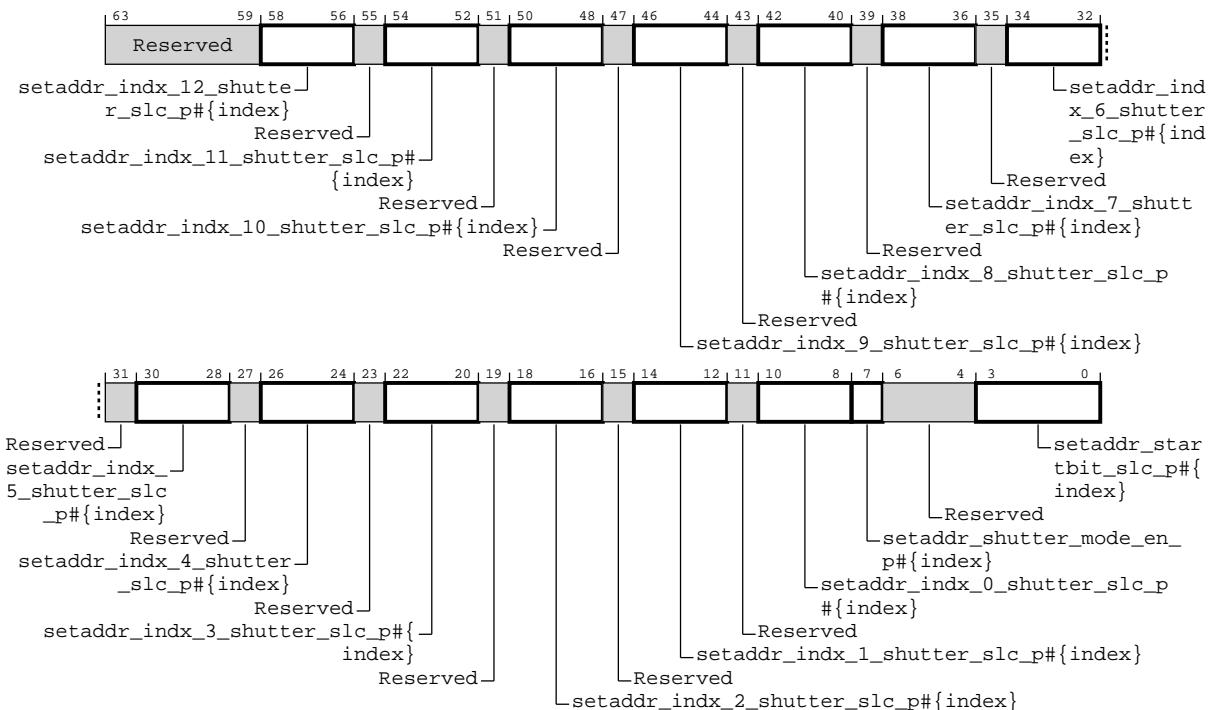
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.pa2setaddr_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.pa2setaddr_ctl` bit and `por_mxp_rcr.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 8-913: por\_mxp\_p0-5\_pa2setaddr\_slc**



**Table 8-924: por\_mxp\_p0-5\_pa2setaddr\_slc attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_idx_12_shutter_slc_p#[index]	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_idx_11_shutter_slc_p#[index]	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[51]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[50:48]	setaddr_idx_10_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_idx_9_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[43]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[42:40]	setaddr_idx_8_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_idx_7_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[35]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[34:32]	setaddr_idx_6_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_idx_5_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[27]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[26:24]	setaddr_idx_4_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_idx_3_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	setaddr_idx_2_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_idx_1_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_slc <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[11]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[10:8]	setaddr_idx_0_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_slc  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[7]	setaddr_shutter_mode_en_p#{index}	Enables address shuttering mode for SLC as programmed by setaddr_idx_X_shutter registers	RW	0b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_slc_p#{index}	<b>SLC</b> SetAddr starting bit for SLC in HNF connected to port p#{index}  <b>0b0110</b> Setaddr starts from PA[6] <b>0b0111</b> Setaddr starts from PA[7] <b>0b1000</b> Setaddr starts from PA[8] <b>0b1001</b> Setaddr starts from PA[9] <b>0b1010</b> Setaddr starts from PA[10] <b>0b1011</b> Setaddr starts from PA[11] <b>0b1100</b> Setaddr starts from PA[12]	RW	0b0110

### 8.3.14.87 por\_mxp\_p0-5\_pa2setaddr\_sf

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the control register of PA to Set/TagAddr and vice versa conversion for HNF-SF on XP port #{{index}}. NOTE:

There will be max. of 6 MXP Port registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xCD8 + #{32\*index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.pa2setaddr\_ctl

### Secure group override

por\_mxp\_scr.pa2setaddr\_ctl

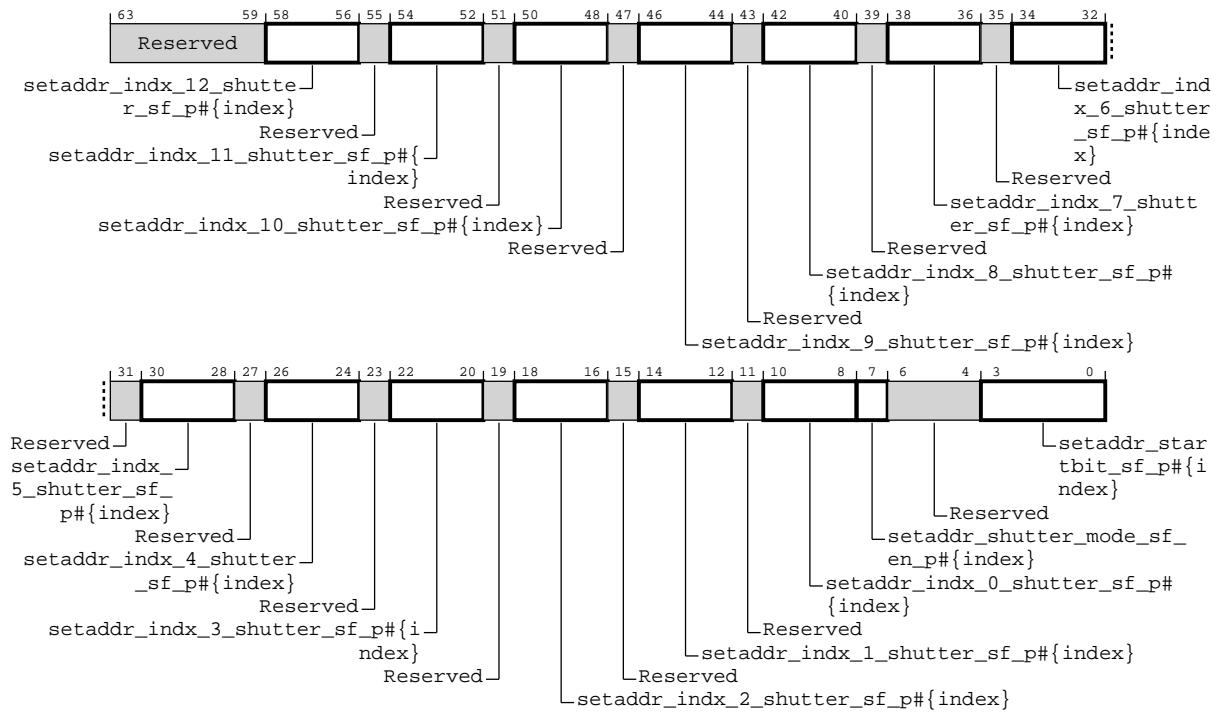
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.pa2setaddr\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.pa2setaddr\_ctl bit and por\_mxp\_rcr.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 8-914: por\_mxp\_p0-5\_pa2setaddr\_sf**



**Table 8-925: por\_mxp\_p0-5\_pa2setaddr\_sf attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_idx_12_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_sf  0b000 shift_1 0b001 shift_2 0b010 shift_3 0b011 shift_4 0b100 shift_5	RW	0b00
[55]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[54:52]	setaddr_idx_11_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_idx_10_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[47]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[46:44]	setaddr_idx_9_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_idx_8_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:36]	setaddr_idx_7_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_idx_6_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[31]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[30:28]	setaddr_idx_5_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_idx_4_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[23]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[22:20]	setaddr_idx_3_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[19]	Reserved	Reserved	RO	-
[18:16]	setaddr_idx_2_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:12]	setaddr_idx_1_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_idx_0_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_sf  <b>0b000</b> pass-through <b>0b001</b> shift_1 <b>0b010</b> shift_2 <b>0b011</b> shift_3 <b>0b100</b> shift_4 <b>0b101</b> shift_5	RW	0b0
[7]	setaddr_shutter_mode_sf_en_p#{index}	Enables address shuttering mode for SF as programmed by setaddr_idx_X_shutter_sf registers	RW	0b0
[6:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3:0]	setaddr_startbit_sf_p#{index}	<p><b>SF</b> SetAddr starting bit for SF in HNF connected to port p#{index}</p> <p><b>0b0110</b> Setaddr starts from PA[6]</p> <p><b>0b0111</b> Setaddr starts from PA[7]</p> <p><b>0b1000</b> Setaddr starts from PA[8]</p> <p><b>0b1001</b> Setaddr starts from PA[9]</p> <p><b>0b1010</b> Setaddr starts from PA[10]</p> <p><b>0b1011</b> Setaddr starts from PA[11]</p> <p><b>0b1100</b> Setaddr starts from PA[12]</p>	RW	0b0110

### 8.3.14.88 por\_mxp\_p0-5\_pa2setaddr\_flex\_slc

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the SLC control register of PA to Set/TagAddr and vice versa conversion for HNF (flexible) on XP port #{index}.

NOTE: There will be max. of 6 MXP Port registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xCE0 + #{32\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_mxp\_rcr.pa2setaddr\_ctl

##### Secure group override

por\_mxp\_scr.pa2setaddr\_ctl

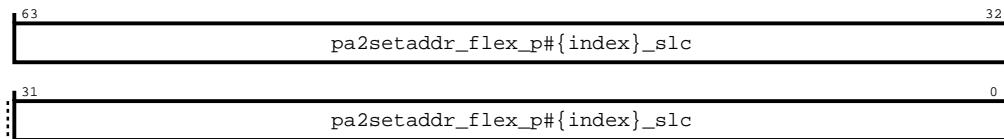
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.pa2setaddr\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.pa2setaddr\_ctl bit and por\_mxp\_rcr.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 8-915: por\_mxp\_p0-5\_pa2setaddr\_flex\_slc**



**Table 8-926: por\_mxp\_p0-5\_pa2setaddr\_flex\_slc attributes**

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_p#{index}_slc	<b>FLEXIBLE</b> PA to SET/TAG ADDR and vice versa conversion config field for HNF connected to port p#{index}	RW	0b0

## 8.3.14.89 por\_mxp\_p0-5\_pa2setaddr\_flex\_sf

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the SF control register of PA to Set/TagAddr and vice versa conversion for HNF (flexible) on XP port #{index}.

NOTE: There will be max. of 6 MXP Port registers based on MXP\_NUM\_DEV\_PORT\_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xCE8 + #{32\*index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_mxp_rcr.pa2setaddr_ctl`

### Secure group override

`por_mxp_scr.pa2setaddr_ctl`

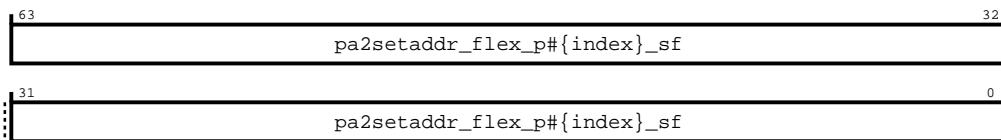
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_mxp_scr.pa2setaddr_ctl` bit is set, Secure accesses to this register are permitted. If both the `por_mxp_scr.pa2setaddr_ctl` bit and `por_mxp_rcr.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 8-916: por\_mxp\_p0-5\_pa2setaddr\_flex\_sf**



**Table 8-927: por\_mxp\_p0-5\_pa2setaddr\_flex\_sf attributes**

Bits	Name	Description	Type	Reset
[63:0]	<code>pa2setaddr_flex_p#{index}_sf</code>	<b>FLEXIBLE</b> PA to SET/TAG ADDR conversion and vice versa config field for HNF connected to port p#{index}	RW	0b0

### 8.3.14.90 por\_mxp\_p0-5\_datasource\_ctl

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the DataSource control register to determine how to drive the CHI DAT.DataSource field. Valid only for Legacy CHI RN-F and SN-F devices.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

### Address offset

0xE00 + #{8\*index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_mxp\_rcr.cfg\_ctl

### Secure group override

por\_mxp\_scr.cfg\_ctl

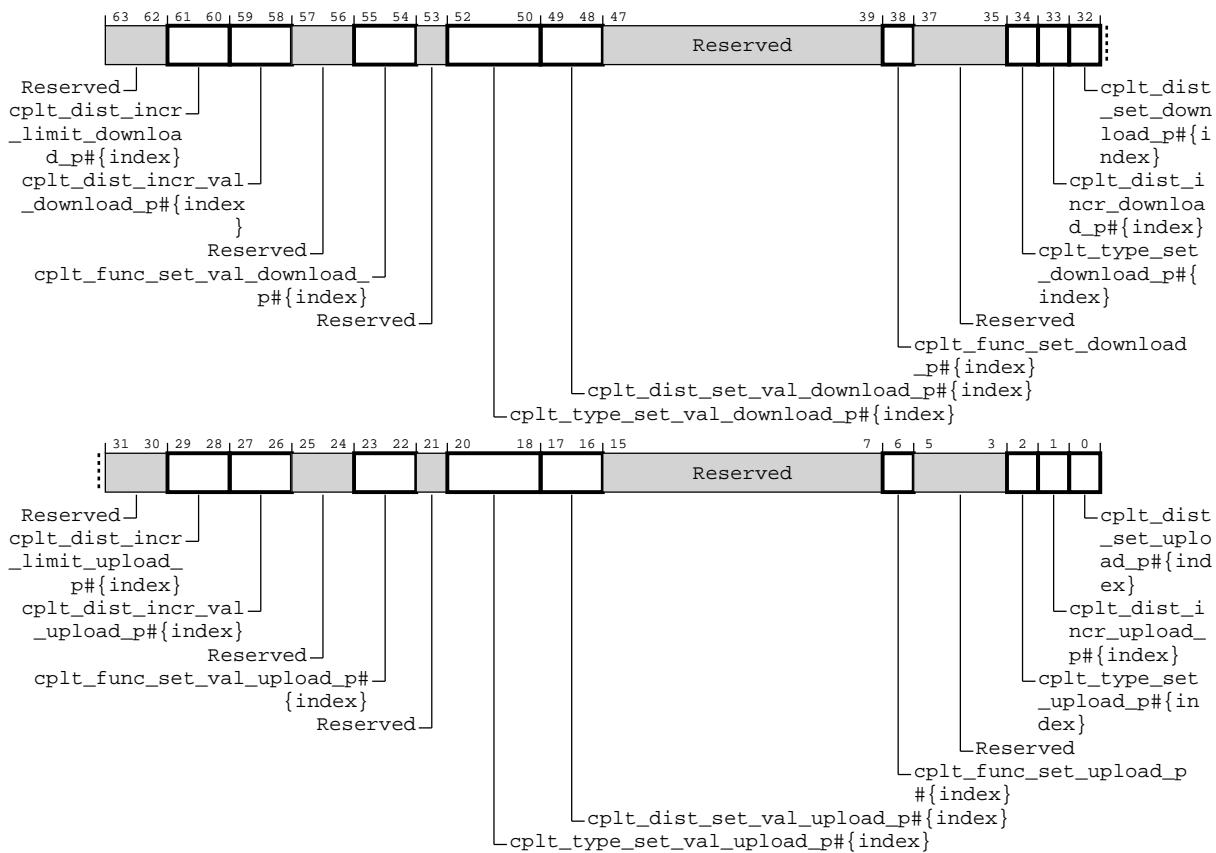
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_mxp\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_mxp\_scr.cfg\_ctl bit and por\_mxp\_rcr.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 8-917: por\_mxp\_p0-5\_datasource\_ctl**



**Table 8-928: por\_mxp\_p0-5\_datasource\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:60]	cplt_dist_incr_limit_download_p#{index}	Completer Increment limit to use when cplt_dist_incr is set to 1 and Completer Distance is incremented. This field is ONLY used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b11
[59:58]	cplt_dist_incr_val_download_p#{index}	Completer Increment value to use when cplt_dist_incr is set to 1. This field is ONLY used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b01
[57:56]	Reserved	Reserved	RO	-
[55:54]	cplt_func_set_val_download_p#{index}	Functional value to use when func_set is set to 1. This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b00
[53]	Reserved	Reserved	RO	-
[52:50]	cplt_type_set_val_download_p#{index}	Completer Type value to use when cplt_type_set is set to 1. This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b100

Bits	Name	Description	Type	Reset
[49:48]	cplt_dist_set_val_download_p#{index}	Completer Distance value to use when cplt_dist_set is set to 1. This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b00
[47:39]	Reserved	Reserved	RO	-
[38]	cplt_func_set_download_p#{index}	Completer Functional Set 0b0 - Functional Set override is not supported for a download 0b1 - Reserved This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b0
[37:35]	Reserved	Reserved	RO	-
[34]	cplt_type_set_download_p#{index}	Completer Type Set 0b0 - Completer Type Set override is not supported for a download 0b1 - Reserved This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b0
[33]	cplt_dist_incr_download_p#{index}	Completer Distance Increment 0b0 - Use the incoming datasource value from the CHIG mesh to do the direct mapping conversion 0b1 - Increment Completer Distance by programmed cplt_dist_incrvalue upto cplt_dist_inrlimit to override the incoming datasource value from the CHIG mesh to do the direct mapping conversion This field is ONLY used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b1
[32]	cplt_dist_set_download_p#{index}	Completer Distance Set 0b0 - Completer Distance Set override is not supported for a download 0b1 - Reserved This field is NOT used for a download on an RN-F device and datasource mapping for download on an SN-F device is not supported.	RW	0b0
[31:30]	Reserved	Reserved	RO	-
[29:28]	cplt_dist_incr_limit_upload_p#{index}	Completer Increment limit to use when cplt_dist_incr is set to 1 and Completer Distance is incremented. This field is NOT used for an upload from either an SN-F or an RN-F device.	RW	0b11
[27:26]	cplt_dist_incr_val_upload_p#{index}	Completer Increment value to use when cplt_dist_incr is set to 1. This field is NOT used for an upload from either an SN-F or an RN-F device.	RW	0b01
[25:24]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[23:22]	cplt_func_set_val_upload_p#{index}	<p>Completer Functional value to use when cplt_func_set is set to 1. Supported values for memory group completer type (applicable only for SN-F device type) are:</p> <p><b>0b00</b> Memory PrefetchTgt not useful</p> <p><b>0b01</b> Memory PrefetchTgt useful</p> <p>Supported values for cachegroup completer type (applicable only for RN-F device type) are:</p> <p><b>0b00</b> Cache group</p> <p><b>0b01</b> Cache group unused prefetch Memory PrefetchTgt useful or cache group unused prefetch</p> <p>0b01 are only supported when cplt_type_set_val is non-zero. For default completer type, functional should always be 00.</p>	RW	Configuration dependent
[21]	Reserved	Reserved	RO	-
[20:18]	cplt_type_set_val_upload_p#{index}	<p>Completer Type value to use when cplt_type_set is set to 1. Supported values are:</p> <p><b>0b000</b> Default Completer Type</p> <p><b>0b001</b> DRAM Completer Type</p> <p><b>0b011</b> High Bandwidth Memory (HBM) Completer Type</p> <p><b>0b100</b> Cache Snoop Hit (applicable only for RN-F device type)</p> <p><b>0b101</b> Interconnect Cache (applicable only for RN-F device type)</p> <p><b>0b110</b> Cache Group 2 (applicable only for RN-F device type)</p>	RW	Configuration dependent
[17:16]	cplt_dist_set_val_upload_p#{index}	Completer Distance value to use when cplt_dist_set is set to 1.	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	cplt_func_set_upload_p#{index}	Completer Functional Set 0b0 - Use the unused prefetch info for cache groups (applicable only for RN-F device type) 0b0 - Use the memory prefetch useful/not useful info (applicable only for SN-F device type) 0b1 - Set Completer Functional field to cplt_func_set_val	RW	0b1
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set_upload_p#{index}	Completer Type Set 0b0 - Reserved 0b1 - Set Completer Type field to cplt_type_set_val	RW	0b1

Bits	Name	Description	Type	Reset
[1]	cplt_dist_incr_upload_p#{index}	Completer Distance Increment 0b0 - Completer Distance Increment override is not supported for an upload 0b1 - Reserved This field is NOT used for an upload from either an SN-F or an RN-F device.	RW	0b0
[0]	cplt_dist_set_upload_p#{index}	Completer Distance Set 0b0 - Reserved 0b1 - Set Completer Distance field to cplt_dist_set_val	RW	0b1

### 8.3.15 RN-D register summary

The following table describes the registers for the relevant component.

**Table 8-929: por\_rnd\_cfg register summary**

Offset	Name	Type	Description
0x0	por_rnd_node_info	RO	<a href="#">por_rnd_node_info</a>
0x80	por_rnd_child_info	RO	<a href="#">por_rnd_child_info</a>
0x980	por_rnd_rcr	RW	<a href="#">por_rnd_rcr</a>
0x988	por_rnd_scr	RW	<a href="#">por_rnd_scr</a>
0x900	por_rnd_unit_info	RO	<a href="#">por_rnd_unit_info</a>
0x908	por_rnd_unit_info2	RO	<a href="#">por_rnd_unit_info2</a>
0xA00	por_rnd_cfg_ctl	RW	<a href="#">por_rnd_cfg_ctl</a>
0xA08	por_rnd_cfg_ctl2	RW	<a href="#">por_rnd_cfg_ctl2</a>
0xA10	por_rnd_aux_ctl	RW	<a href="#">por_rnd_aux_ctl</a>
0xA18 : 0xA28	por_rnd_s0-2_port_control	RW	<a href="#">por_rnd_s0-2_port_control</a>
0xA30 : 0xA40	por_rnd_s0-2_mpam_control	RW	<a href="#">por_rnd_s0-2_mpam_control</a>
0xA80 : 0xAC0	por_rnd_s0-2_qos_control	RW	<a href="#">por_rnd_s0-2_qos_control</a>
0xA88 : 0xAC8	por_rnd_s0-2_qos_lat_tgt	RW	<a href="#">por_rnd_s0-2_qos_lat_tgt</a>
0xA90 : 0xAD0	por_rnd_s0-2_qos_lat_scale	RW	<a href="#">por_rnd_s0-2_qos_lat_scale</a>
0xA98 : 0xAD8	por_rnd_s0-2_qos_lat_range	RW	<a href="#">por_rnd_s0-2_qos_lat_range</a>
0xD900	por_rnd_pmu_event_sel	RW	<a href="#">por_rnd_pmu_event_sel</a>
0x1900	por_rnd_syscoreq_ctl	RW	<a href="#">por_rnd_syscoreq_ctl</a>
0x1908	por_rnd_syscoack_status	RO	<a href="#">por_rnd_syscoack_status</a>

#### 8.3.15.1 por\_rnd\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x0

### Type

RO

### Reset value

See individual bit resets

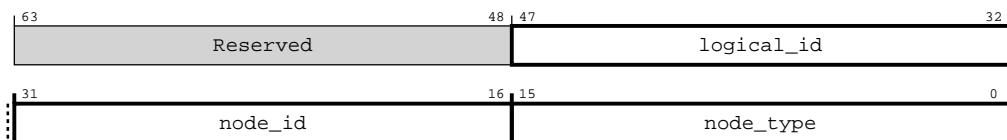
### Usage constraints

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 8-918: por\_rnd\_node\_info**



**Table 8-930: por\_rnd\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x000D

## 8.3.15.2 por\_rnd\_child\_info

Provides component child identification information.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x80

### Type

RO

### Reset value

See individual bit resets

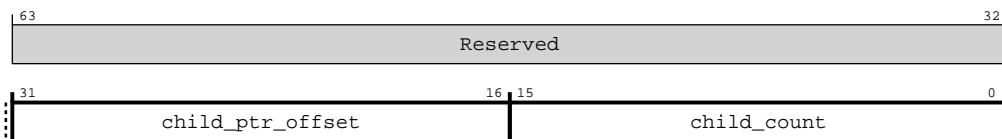
### Usage constraints

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 8-919: por\_rnd\_child\_info**



**Table 8-931: por\_rnd\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0x0

### 8.3.15.3 por\_rnd\_rcr

Root register access override.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x980

#### Type

RW

### Reset value

See individual bit resets

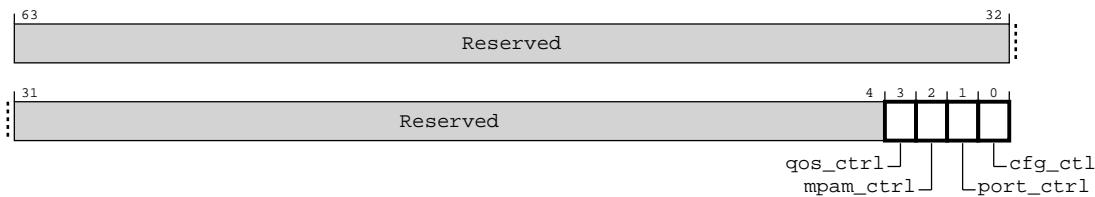
### Usage constraints

This register is owned in the Root space and is 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 8-920: por\_rnd\_rcr**



**Table 8-932: por\_rnd\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	qos_ctrl	Allows Root override of the QoS control registers	RW	0b0
[2]	mpam_ctrl	Allows Root override of the AXI port MPAM override register	RW	0b0
[1]	port_ctrl	Allows Root override of the AXI port control registers	RW	0b0
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.15.4 por\_rnd\_scr

Secure register access override.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x988

#### Type

RW

#### Reset value

See individual bit resets

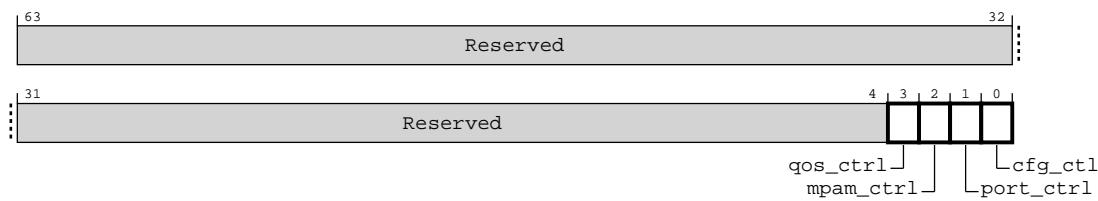
## Usage constraints

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 8-921: por\_rnd\_scr**



**Table 8-933: por\_rnd\_scr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	qos_ctrl	Allows Secure override of the QoS control registers	RW	0b0
[2]	mpam_ctrl	Allows Secure override of the AXI port MPAM override register	RW	0b0
[1]	port_ctrl	Allows Secure override of the AXI port control registers	RW	0b0
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.15.5 por\_rnd\_unit\_info

Provides component identification information for RN-D.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x900

### Type

RO

### Reset value

See individual bit resets

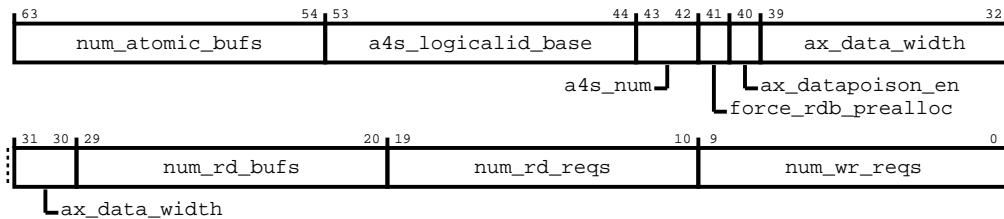
## Usage constraints

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 8-922: por\_rnd\_unit\_info**



**Table 8-934: por\_rnd\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:54]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[53:44]	a4s_logicalid_base	AXI4Stream interfaces logical ID base	0x0	Configuration dependent
[43:42]	a4s_num	Number of AXI4Stream interfaces present	RO	Configuration dependent
[41]	force_rdb_prealloc	Force read data buffer preallocation  <b>0b1</b> yes  <b>0b0</b> no	RO	Configuration dependent
[40]	ax_datapoison_en	Data Poison enable on ACE-Lite/AXI4 interface  <b>0b1</b> Enabled  <b>0b0</b> Not enabled	RO	Configuration dependent
[39:30]	ax_data_width	AXI interface data width in bits	RO	Configuration dependent
[29:20]	num_rd_bufs	Number of read data buffers	RO	Configuration dependent
[19:10]	num_rd_reqs	Number of outstanding read requests	RO	Configuration dependent
[9:0]	num_wr_reqs	Number of outstanding write requests	RO	Configuration dependent

### 8.3.15.6 por\_rnd\_unit\_info2

Provides additional component identification information for RN-D.

## Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x908

### Type

RO

### Reset value

See individual bit resets

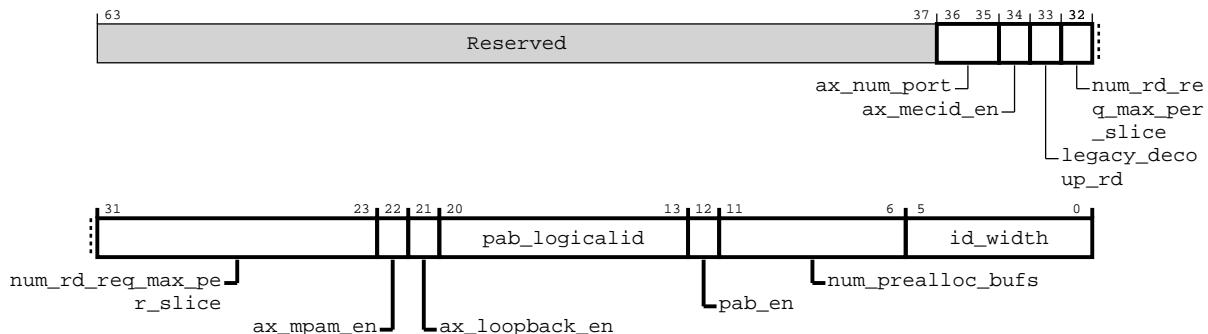
### Usage constraints

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 8-923: por\_rnd\_unit\_info2**



**Table 8-935: por\_rnd\_unit\_info2 attributes**

Bits	Name	Description	Type	Reset
[63:37]	Reserved	Reserved	RO	-
[36:35]	ax_num_port	Number of ACE-Lite/AXI4 interfaces present	RO	Configuration dependent
[34]	ax_mecid_en	MECID enable on ACE-Lite/AXI4 interface  0b1 Enabled  0b0 Not enabled	RO	Configuration dependent
[33]	legacy_decoupled_rd	Legacy decoupled read mode, no read burst propagation	RO	Configuration dependent
[32:23]	num_rd_req_max_per_slice	Number of read request entires per slice	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[22]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface  <b>0b1</b> Enabled  <b>0b0</b> Not enabled	RO	Configuration dependent
[21]	ax_loopback_en	LoopBack enable on ACE-Lite/AXI4 interface  <b>0b1</b> Enabled  <b>0b0</b> Not enabled	RO	Configuration dependent
[20:13]	pab_logicalid	PUB AUB bridge Logical ID	RO	0x10
[12]	pab_en	PUB AUB bridge enable  <b>0b1</b> Enabled  <b>0b0</b> Not enabled	RO	0x0
[11:6]	num_prealloc_bufs	Number of Pre-allocated Read Data Buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite subordinate ports	RO	Configuration dependent

### 8.3.15.7 por\_rnd\_cfg\_ctl

Functions as the configuration control register. Specifies the current mode.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnd\_rcr.cfg\_ctl

##### Secure group override

por\_rnd\_scr.cfg\_ctl

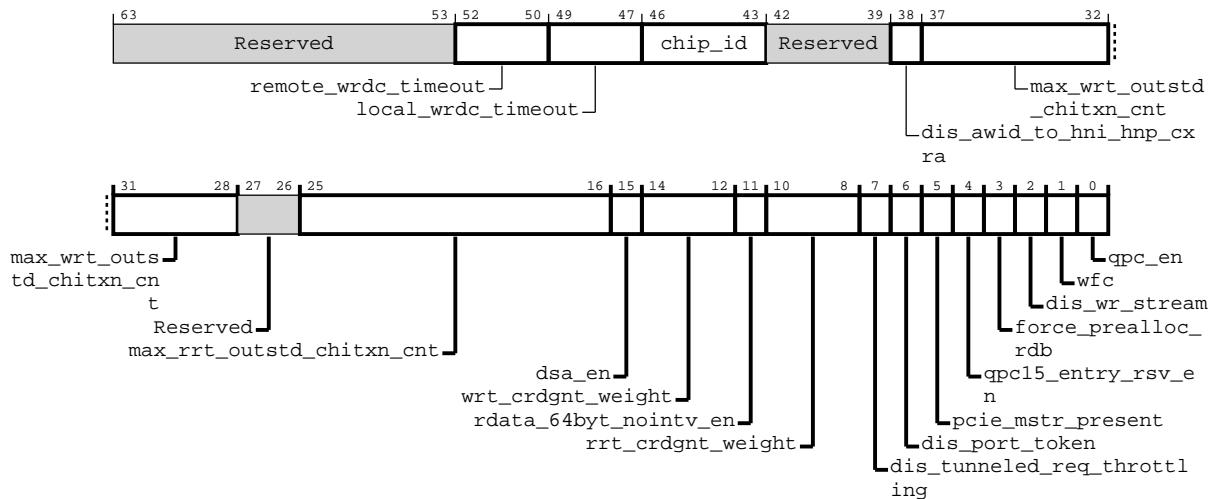
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.cfg\_ctl bit and por\_rnd\_rcr.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 8-924: por\_rnd\_cfg\_ctl**



**Table 8-936: por\_rnd\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52:50]	remote_wrdc_timeout	Configurable write data cancel timeout value for remote traffic.  0b000 ~2000 cycles 0b001 ~4000 cycles 0b010 ~8000 cycles 0b011 ~16000 cycles (default) 0b100 ~32000 cycles (all other values reserved)	RW	0b011

Bits	Name	Description	Type	Reset
[49:47]	local_wrdc_timeout	Configurable write data cancel timeout value for local traffic. 0b000 : ~2000 cycles <b>0b001</b> ~4000 cycles <b>0b010</b> ~8000 cycles <b>0b011</b> ~16000 cycles (default) <b>0b100</b> ~32000 cycles (all other values rsvd)	RW	0b010
[46:43]	chip_id	Configurable ChipID for this RNX instance. Must be correctly set for proper handling of remote traffic to HNI/HNP. Only supports values 0..3. Two MSB's is reserved.	RW	0b0000
[42:39]	Reserved	Reserved	RO	-
[38]	dis_awid_to_hni_hnp_cxra	If set, disables compressed AWID to HNI, HNP and CXRA, also disables compressed AWID based ordering. Set this bit if uniq-ID write performance is needed.	RW	0b0
[37:28]	max_wrt_outstd_chitxn_cnt	Maximum number of outstanding writes allowed on CHI-side	RW	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	max_rrt_outstd_chitxn_cnt	Maximum number of outstanding reads allowed on CHI-side	RW	Configuration dependent
[15]	dsa_en	Enables Direct Subordinate Access feature to SBSX/MTSX. When setting this bit, dis_compack_on_writes must be set to 1, because dsa mode does not work with compack flow. Requires POR_DSA_EN_PARAM to be set, otherwise this config bit has no effect.	RW	0b0
[14:12]	wrt_crdgnt_weight	Determines weight of credit grant allocated to retried writes in presence of pending retried reads	RW	0b001
[11]	rdata_64byt_nointv_en	Enables no interleaving property on normal memory read data within 64B granule when set	RW	0b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant allocated to retried reads in presence of pending retried writes	RW	0b100
[7]	dis_tunneled_req_throttling	Disables retry based throttling of tunneled write requests	RW	0b0
[6]	dis_port_token	If set, disables per port reservation in the tracker(rd and wr)	RW	0b1
[5]	pcie_mstr_present	Indicates PCIe requester/manager is present; must be set if PCIe requester/manager is present upstream of RN-I or RN-D	RW	0b0
[4]	qpc15_entry_rsv_en	Enables QPC15 entry reservation <b>0b1</b> Reserves tracker entry for QoS15 requests <b>0b0</b> Does not reserve tracker entry for QoS15 requests <b>NOTE</b> Only valid and applicable when por_rni_qpc_en is set	RW	0b0
[3]	force_prealloc_rdb	When set, all reads from the RN-I are sent with a preallocated read data buffer	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[2]	dis_wr_stream	Disables streaming of ordered writes when set	RW	0b0
[1]	wfc	When set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	0b0
[0]	qpc_en	When set, enables QPC-based scheduling using two QoS priority classes (QoS15 and non-QoS15)	RW	0b1

### 8.3.15.8 por\_rnd\_cfg\_ctl2

Functions as the configuration control register. Specifies the current mode.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnd\_rcr.cfg\_ctl

##### Secure group override

por\_rnd\_scr.cfg\_ctl

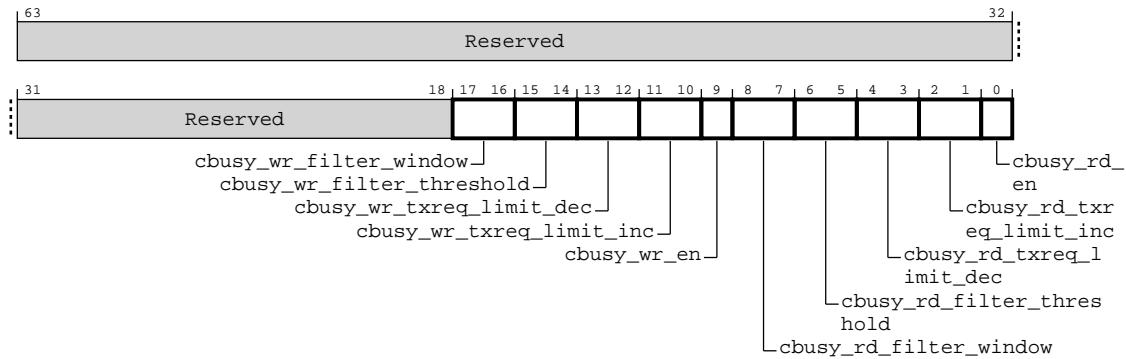
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.cfg\_ctl bit and por\_rnd\_rcr.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 8-925: por\_rnd\_cfg\_ctl2**



**Table 8-937: por\_rnd\_cfg\_ctl2 attributes**

Bits	Name	Description	Type	Reset
[63:18]	Reserved	Reserved	RO	
[17:16]	cbusy_wr_filter_window	<p>Number of CBusy write responses in one sampling window.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> <li><b>0b00</b> 256 (default)</li> <li><b>0b01</b> 64</li> <li><b>0b10</b> 128</li> <li><b>0b11</b> 512</li> </ul>	RW	0b00
[15:14]	cbusy_wr_filter_threshold	<p>Fraction of CBusy write responses in the sampling window necessary to be considered valid sample of that CBusy value.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 1/16 (default)</li> <li><b>0b01</b> 1/32</li> <li><b>0b10</b> 1/8</li> <li><b>0b11</b> 1/4</li> </ul>	RW	0b00

Bits	Name	Description	Type	Reset
[13:12]	cbusy_wr_txreq_limit_dec	<p>Dynamic write TXREQ limit decrement. Controls how quickly the dynamic write TXREQ limit is decreased when CBusy indicates a value of 3.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[11:10]	cbusy_wr_txreq_limit_inc	<p>Dynamic write TXREQ limit decrement. Controls how quickly the dynamic write TXREQ limit is decreased when CBusy indicates a value of 2.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[9]	cbusy_wr_en	Enables CBusy throttling of write txreq within the RNI	RW	0b1
[8:7]	cbusy_rd_filter_window	<p>Number of CBusy read responses in one sampling window.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 256 (default)</li> <li><b>0b01</b> 64</li> <li><b>0b10</b> 128</li> <li><b>0b11</b> 512</li> </ul>	RW	0b00

Bits	Name	Description	Type	Reset
[6:5]	cbusy_rd_filter_threshold	<p>Fraction of CBusy read responses in the sampling window necessary to be considered valid sample of that CBusy value.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 1/16 (default)</li> <li><b>0b01</b> 1/32</li> <li><b>0b10</b> 1/8</li> <li><b>0b11</b> 1/4</li> </ul>	RW	0b00
[4:3]	cbusy_rd_txreq_limit_dec	<p>Dynamic read TXREQ limit decrement. Controls how quickly the dynamic read TXREQ limit is decreased when CBusy indicates a value of 3.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[2:1]	cbusy_rd_txreq_limit_inc	<p>Dynamic read TXREQ limit increment. Controls how quickly the dynamic read TXREQ limit is increased when CBusy indicates values less than 2.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[0]	cbusy_rd_en	Enables CBusy throttling of read txreq within the RNI	RW	0b1

### 8.3.15.9 por\_rnd\_aux\_ctl

Functions as the auxiliary control register for RN-D.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA10

### Type

RW

### Reset value

See individual bit resets

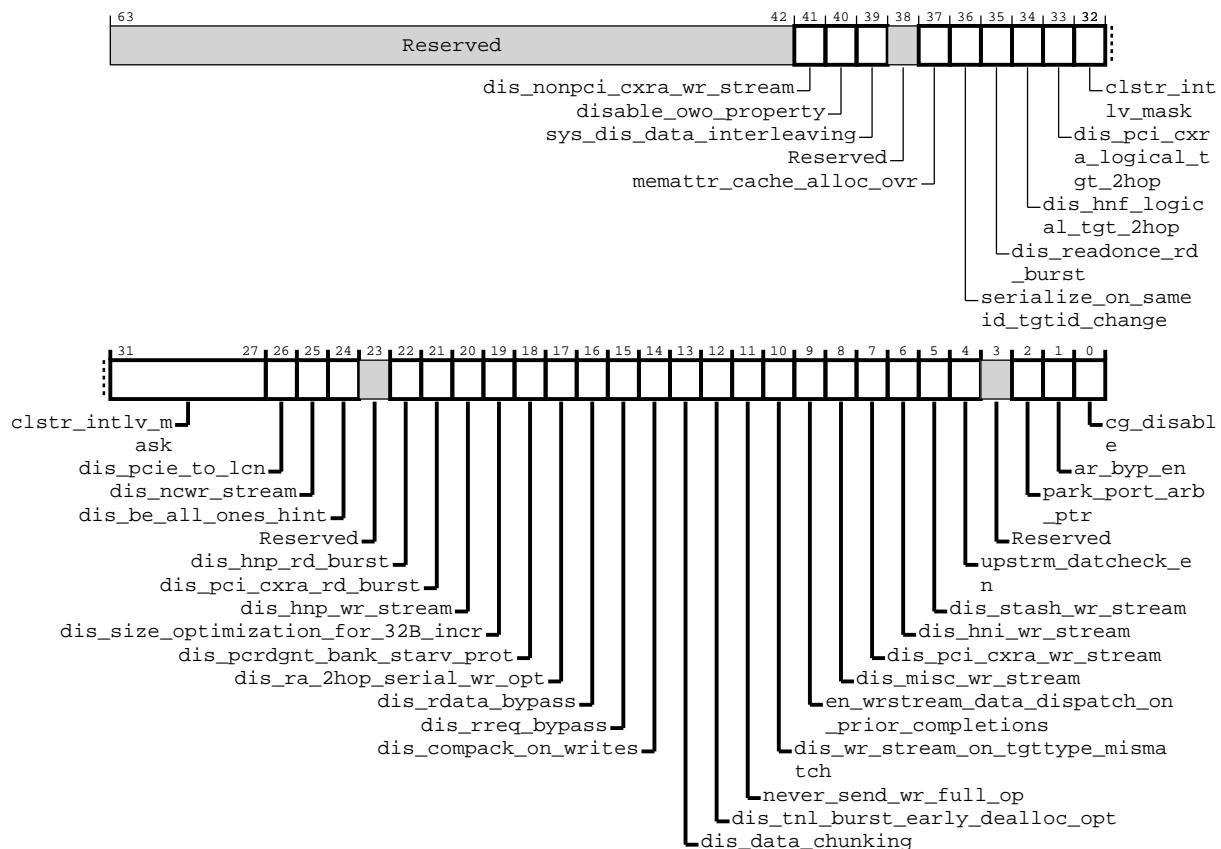
### Usage constraints

This register is owned in the Root space and is 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 8-926: por\_rnd\_aux\_ctl**



**Table 8-938: por\_rnd\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41]	dis_nonpci_cxra_wr_stream	Disables streaming of ordered writes to (non-PCI)-CXRA when set	RW	0b0
[40]	disable_owo_property	When set RN-D AXI interface will behave as if AXI Ordered_Write_Observation property is disabled for all writes.	RW	0b0
[39]	sys_dis_data_interleaving	<p>System optimized disable read DATA interleaving for all ports. Disables all read data interleaving, including atomic read data being returned for all AXI ports. Read burst preservation is enabled same as in normal mode, but this requires certain system level restrictions:</p> <ol style="list-style-type: none"> <li>1. Cannot set SYS_DIS_DATA_INTERLEAVING for multi-chip systems. Support for remote HN-P is a future feature.</li> <li>2. When setting SYS_DIS_DATA_INTERLEAVING for an RN-I/RN-D it is required for that RN-I/RN-D to target only one HN-P with read bursts. It is also required for the path to a single remote HN-P, when remote support is available, to only go through a single CCG and for the ccg_rni to also have SYS_DIS_DATA_INTERLEAVING set.</li> <li>3. The AXI subordinate downstream of HN-P must not interleave read burst data.</li> <li>4. Must have NUM_RD_REQ == NUM_RD_BUF and NUM_RD_REQ &lt;= 256, otherwise this bit has no effect.</li> <li>5. The sum of the maximum number of beats of a cracked burst per each port must fit wholly within an arslice - breaking this rule will result in a hang. Must comprehend DIS_PORT_TOKEN AND QPC15_ENTRY_RSV settings, which will limit number of available entries</li> </ol>	RW	0b0
[38]	Reserved	Reserved	RO	-
[37]	memattr_cache_alloc_ovr	If set, overrides the incoming memattr cacheable and allocate attributes both to 1 for non-device and bufferable requests	RW	0b0
[36]	serialize_on_sameid_tgtid_change	If set, serializes request issue for sameid writes to cxra when tgtid mismatches previous write request.	RW	0b0
[35]	dis_readonce_rd_burst	If set, disables read burst for ReadOnce from AXI.	RW	0b0
[34]	dis_hnf_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical target is pci_cxra and logical target is hnf, otherwise may tunneling/2hop to RA if interleaving granularity settings allow.	RW	0b0
[33]	dis_pci_cxra_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical and logical target is pci_cxra, otherwise tunneling/2hop to RA.	RW	0b0

Bits	Name	Description	Type	Reset
[32:27]	clstr_intlv_mask	Encoded static mask for max interleave granularity supported. When this setting is less than or equal to rnsam's programmed interleave granularity for a write to pci_cxra, tunneling/2hop flow will be used.  <b>0b111111</b> 64B <b>0b111110</b> 128B <b>0b111100</b> 256B <b>0b111000</b> 512B <b>0b110000</b> 1024B <b>0b100000</b> 2048B <b>0b000000</b> 4096B <b>others</b> Reserved	RW	0b000000
[26]	dis_PCIE_to_lcn	If set, all pcie traffic sent directly to HNF/CCG, bypasses LCN. Only has effect when pcie_mstr_present	RW	0b1
[25]	dis_ncwr_stream	Disables streaming of ordered non-cacheable writes when set	RW	0b0
[24]	dis_be_all_ones_hint	If set, disables hint to HNF which signals all BE=1's on writes	RW	0b0
[23]	Reserved	Reserved	RO	-
[22]	dis_hnp_rd_burst	If set, disables read burst to HNP on CHI request flits . Read burst on CHI is supported in either non-decoupled RDB configuration or when decoupled (NUM_RD_BUF < NUM_RD_REQ) and LEGACY_DECOUP_RD = 0 and read data chunking is enabled on AXI.	RW	0b0
[21]	dis_pci_cxra_rd_burst	If set, disables read burst to PCI-CXRA on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	0b0
[20]	dis_hnp_wr_stream	Disables streaming of ordered writes to HNP when set	RW	0b0
[19]	dis_size_optimization_for_32B_incr	If set, disables the size related optimization for a 32B INCR burst (rh-2512). Only applies to writes.	RW	0b0
[18]	dis_pcrgnt_bank_starv_prot	If set, disables across arslice starvation protection	RW	0b0
[17]	dis_ra_2hop_serial_wr_opt	If set, disables 2 hop indication to ra for serialized writes; will indicate 3 hop	RW	0b0
[16]	dis_rdata_bypass	If set, disables read data bypass path	RW	0b0
[15]	dis_req_bypass	If set, disables read request bypass path	RW	0b0
[14]	dis_compack_on_writes	If set, disables comp_ack on streaming writes. WrData is used for ordering writes	RW	0b1

Bits	Name	Description	Type	Reset
[13]	dis_data_chunking	If set, disables the data chunking feature	RW	0b0
[12]	dis_tnl_burst_early_dealloc_opt	If set, disables the optimization related to early deallocation of tunneled writes for intermediate txns of burst	RW	0b0
[11]	never_send_wr_full_op	If set, RNI will never send WR FULL op. All write ops will be of PTL type	RW	0b0
[10]	dis_wr_stream_on_tgtype_mismatch	If set, serializes first write when moving from one tgtype to another	RW	0b0
[9]	en_wrstream_data_dispatch_on_prior_completions	If set, data dispatch for streaming writes waits for completion of all older writes	RW	0b0
[8]	dis_misc_wr_stream	Disables streaming of ordered writes with following attributes when set : Device memory or EWA=0 or Excl=1	RW	0b0
[7]	dis_pci_cxra_wr_stream	Disables streaming of ordered writes to PCI-CXRA when set	RW	0b0
[6]	dis_hni_wr_stream	Disables streaming of ordered writes to HNI when set	RW	0b0
[5]	dis_stash_wr_stream	Disables streaming of ordered WrUniqStash when set	RW	0b0
[4]	upstrm_datcheck_en	Upstream supports Datacheck	RW	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2]	park_port_arb_ptr	Parks the AXI port arbitration pointer for Burst	RW	0b0
[1]	ar_byp_en	AR bypass enable; enables bypass path in the AR pipeline	RW	0b1
[0]	cg_disable	Disables clock gating when set	RW	0b0

### 8.3.15.10 por\_rnd\_s0-2\_port\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface settings.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA18 + #{index}\*8}

##### Type

RW

##### Reset value

See individual bit resets

## Root group override

por\_rnd\_rcr.port\_ctrl

## Secure group override

por\_rnd\_scr.port\_ctrl

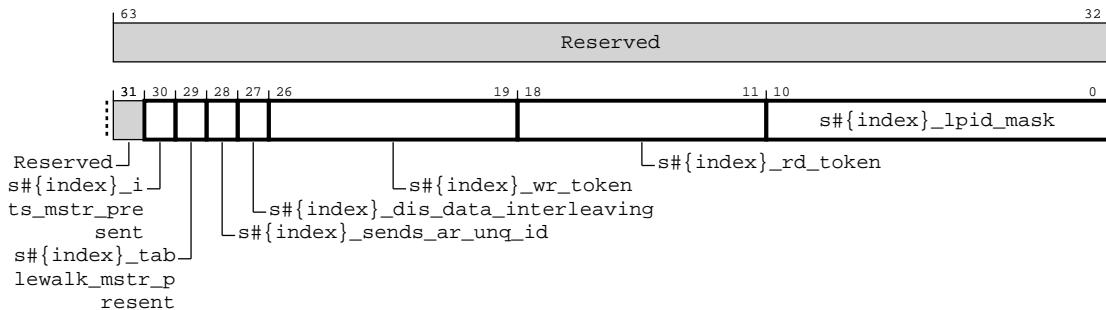
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.port\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.port\_ctrl bit and por\_rnd\_rcr.port\_ctrl 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 8-927: por\_rnd\_s0-2\_port\_control**



**Table 8-939: por\_rnd\_s0-2\_port\_control attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-
[30]	s#{index}_its_mstr_present	Must be set if translation table walk manager present such as TCU or GIC for non-PCIE case. This affects RND AW channel only.	RW	0b0
[29]	s#{index}_tablewalk_mstr_present	Must be set if translation table walk manager present such as TCU or GIC. This affects RND AR channel only.	RW	0b0
[28]	s#{index}_sends_ar_unq_id	If set, indicates AR transactions on Port#{index} are always Unique ID. This bit for a port must be set to 1 to enable Read Burst on the CHI side of RND.	RW	0b0
[27]	s#{index}_dis_data_interleaving	If set, disables read DATA interleaving on RDATAS#{index} channel. This applies only to RDATA generated as a response to requests on AR channel. This does not apply to RDATA generated as a response to Atomic request on AW channel. I.e. RDATA of an Atomic op, on AW channel, may interleave with RDATA of an AR channel request. If setting cfg_ctl.wfc bit along with this bit, aux_ctl.dis_rreq_bypass must also be set, otherwise completion order will be violated.	RW	0b0

Bits	Name	Description	Type	Reset
[26:19]	s#{index}_wr_token	Port S#{index} reserved token count for AW channel This must be less than the number of Wr requests(RNID_NUM_WR_REQ_PARAM) on AW achnnel	RW	8'b0000_0000
[18:11]	s#{index}_rd_token	Port S#{index} reserved token count for AR channel per slice This should be less than the number of Rd requests(RNID_NUM_RD_REQ_PARAM) per slice on AR achnnel	RW	8'b0000_0000
[10:0]	s#{index}_lpid_mask	Port S#{index} LPID mask  <b>LPIID[0]</b> Equal to the result of UnaryOR of BitwiseAND of LPID mask and AXID (LPIID[0] =  (AXID & mask)); specifies which AXID bit is reflected in the LSB of LPID  <b>LPIID[2:1]</b> Equal to port ID[1:0]; the MSB of LPID contains port ID	RW	11'b000_0000_0000

### 8.3.15.11 por\_rnd\_s0-2\_mpam\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface MPAM override values

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA30 + #{index}\*8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnd\_rcr.mpam\_ctrl

##### Secure group override

por\_rnd\_scr.mpam\_ctrl

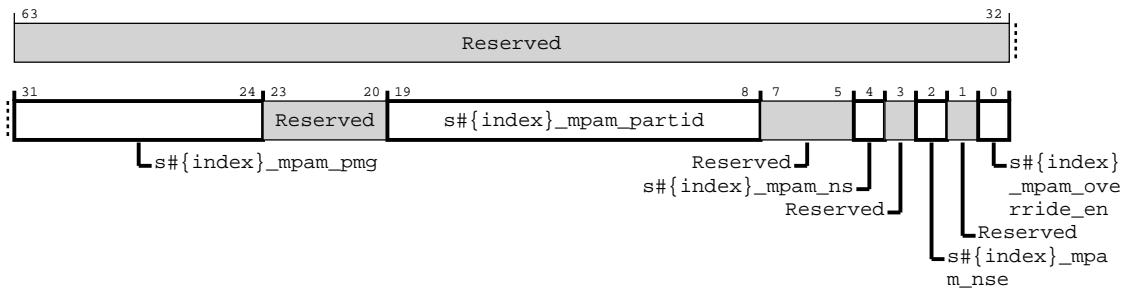
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.mpam\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.mpam\_ctrl bit and por\_rnd\_rcr.mpam\_ctrl 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 8-928: por\_rnd\_s0-2\_mpam\_control**



**Table 8-940: por\_rnd\_s0-2\_mpam\_control attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:24]	s#{index}_mpam_pmg	Port S#{index} MPAM_PMG value	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	s#{index}_mpam_partid	Port S#{index} MPAM_PARTID value	RW	0b0
[7:5]	Reserved	Reserved	RO	
[4]	s#{index}_mpam_ns	Port S#{index} MPAM_NS value	RW	0b0
[3]	Reserved	Reserved	RO	
[2]	s#{index}_mpam_nse	Port S#{index} MPAM_NSE value	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	s#{index}_mpam_override_en	Port S#{index} MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Note that when RNID_AXMPAM_EN_PARAM is set to 0, MPAM override value is always used irrespective of this bit value	RW	0b0

### 8.3.15.12 por\_rnd\_s0-2\_qos\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE subordinate interface.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA80 + #{index}\*32}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rnd\_rcr.qos\_ctrl

## Secure group override

por\_rnd\_scr.qos\_ctrl

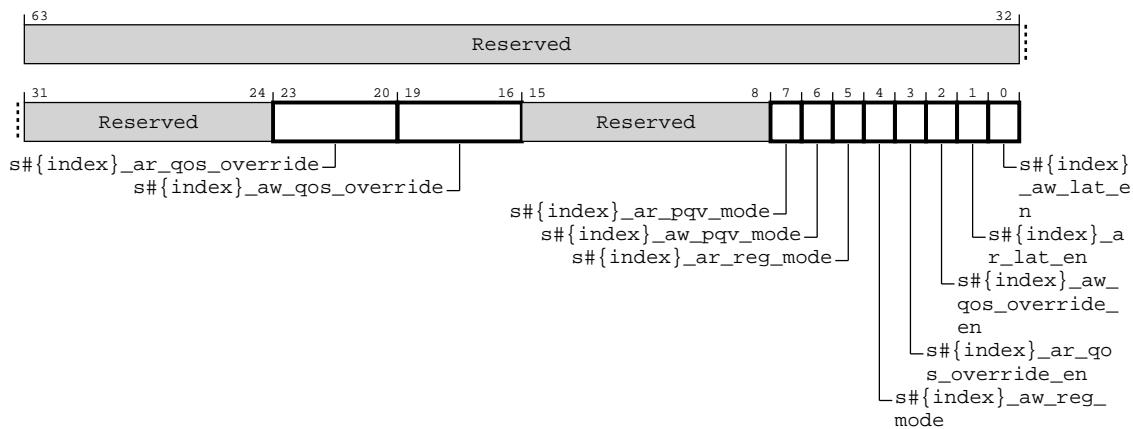
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.qos\_ctrl bit and por\_rnd\_rcr.qos\_ctrl 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 8-929: por\_rnd\_s0-2\_qos\_control**



**Table 8-941: por\_rnd\_s0-2\_qos\_control attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	s#{index}_ar_qos_override	AR QoS override value for port S#{index}	RW	0b0000
[19:16]	s#{index}_aw_qos_override	AW QoS override value for port S#{index}	RW	0b0000
[15:8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7]	s#{index}_ar_pqv_mode	<p>Configures the QoS regulator mode for read transactions during period mode</p> <p><b>0b0</b> Normal mode; QoS value is stable when the manager is idle</p> <p><b>0b1</b> Quiesce high mode; QoS value tends to the maximum value when the manager is idle</p>	RW	0b0
[6]	s#{index}_aw_pqv_mode	<p>Configures the QoS regulator mode for write transactions during period mode</p> <p><b>0b0</b> Normal mode; QoS value is stable when the manager is idle</p> <p><b>0b1</b> Quiesce high mode; QoS value tends to the maximum value when the manager is idle</p>	RW	0b0
[5]	s#{index}_ar_reg_mode	<p>Configures the QoS regulator mode for read transactions</p> <p><b>0b0</b> Latency mode</p> <p><b>0b1</b> Period mode; used for bandwidth regulation</p>	RW	0b0
[4]	s#{index}_aw_reg_mode	<p>Configures the QoS regulator mode for write transactions</p> <p><b>0b0</b> Latency mode</p> <p><b>0b1</b> Period mode; used for bandwidth regulation</p>	RW	0b0
[3]	s#{index}_ar_qos_override_en	Enables port S#{index} AR QoS override; when set, allows QoS value on inbound AR transactions to be overridden	RW	0b0
[2]	s#{index}_aw_qos_override_en	Enables port S#{index} AW QoS override; when set, allows QoS value on inbound AW transactions to be overridden	RW	0b0
[1]	s#{index}_ar_lat_en	Enables port S#{index} AR QoS regulation when set	RW	0b0
[0]	s#{index}_aw_lat_en	Enables port S#{index} AW QoS regulation when set	RW	0b0

### 8.3.15.13 por\_rnd\_s0-2\_qos\_lat\_tgt

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS target latency (in cycles) for regulations of port S#{index} read and write transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA88 + #{index}\*32

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnd\_rcr.qos\_ctrl

### Secure group override

por\_rnd\_scr.qos\_ctrl

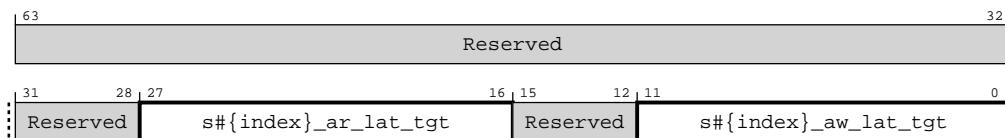
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.qos\_ctrl bit and por\_rnd\_rcr.qos\_ctrl 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 8-930: por\_rnd\_s0-2\_qos\_lat\_tgt**



**Table 8-942: por\_rnd\_s0-2\_qos\_lat\_tgt attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:16]	s#{index}_ar_lat_tgt	Port S#{index} AR channel target latency; a value of 0 corresponds to no regulation	RW	0x000
[15:12]	Reserved	Reserved	RO	-
[11:0]	s#{index}_aw_lat_tgt	Port S#{index} AW channel target latency; a value of 0 corresponds to no regulation	RW	0x000

### 8.3.15.14 por\_rnd\_s0-2\_qos\_lat\_scale

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the QoS target latency scale factor for port S#{index} read and write transactions. This register represents powers of two from the range 2<sup>(-5)</sup> to 2<sup>(-12)</sup>; it is used to match a 16-bit integrator.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA90 + #{index}\*32}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnd\_rcr.qos\_ctrl

### Secure group override

por\_rnd\_scr.qos\_ctrl

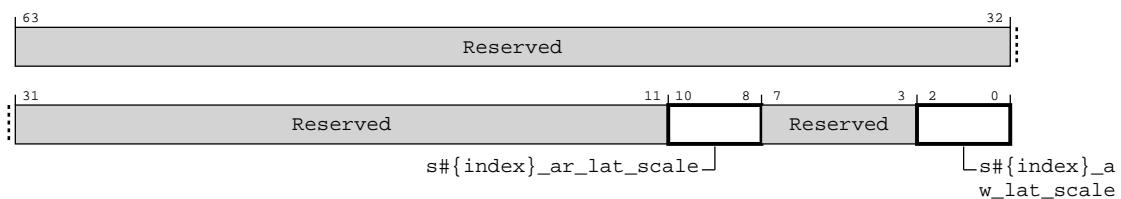
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.qos\_ctrl bit and por\_rnd\_rcr.qos\_ctrl 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 8-931: por\_rnd\_s0-2\_qos\_lat\_scale**



**Table 8-943: por\_rnd\_s0-2\_qos\_lat\_scale attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[10:8]	s#{index}_ar_lat_scale	Port S#{index} AR QoS scale factor <b>0b000</b> $2^{-5}$ <b>0b001</b> $2^{-6}$ <b>0b010</b> $2^{-7}$ <b>0b011</b> $2^{-8}$ <b>0b100</b> $2^{-9}$ <b>0b101</b> $2^{-10}$ <b>0b110</b> $2^{-11}$ <b>0b111</b> $2^{-12}$	RW	0x0
[7:3]	Reserved	Reserved	RO	
[2:0]	s#{index}_aw_lat_scale	Port S#{index} AW QoS scale factor <b>0b000</b> $2^{-5}$ <b>0b001</b> $2^{-6}$ <b>0b010</b> $2^{-7}$ <b>0b011</b> $2^{-8}$ <b>0b100</b> $2^{-9}$ <b>0b101</b> $2^{-10}$ <b>0b110</b> $2^{-11}$ <b>0b111</b> $2^{-12}$	RW	0x0

### 8.3.15.15 por\_rnd\_s0-2\_qos\_lat\_range

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the minimum and maximum QoS values generated by the QoS latency regulator for port S#{index} read and write transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA98 + #{index}\*32}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnd\_rcr.qos\_ctrl

##### Secure group override

por\_rnd\_scr.qos\_ctrl

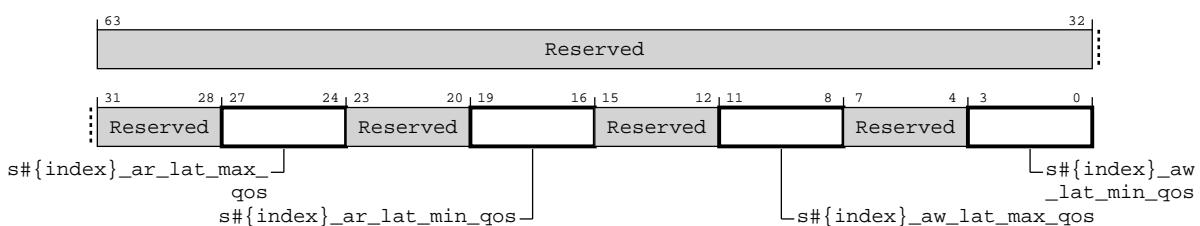
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnd\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rnd\_scr.qos\_ctrl bit and por\_rnd\_rcr.qos\_ctrl 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 8-932: por\_rnd\_s0-2\_qos\_lat\_range**



**Table 8-944: por\_rnd\_s0-2\_qos\_lat\_range attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	
[27:24]	s#[index]_ar_lat_max_qos	Port S#[index] AR QoS maximum value	RW	0x0
[23:20]	Reserved	Reserved	RO	
[19:16]	s#[index]_ar_lat_min_qos	Port S#[index] AR QoS minimum value	RW	0x0
[15:12]	Reserved	Reserved	RO	
[11:8]	s#[index]_aw_lat_max_qos	Port S#[index] AW QoS maximum value	RW	0x0
[7:4]	Reserved	Reserved	RO	
[3:0]	s#[index]_aw_lat_min_qos	Port S#[index] AW QoS minimum value	RW	0x0

### 8.3.15.16 por\_rnd\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

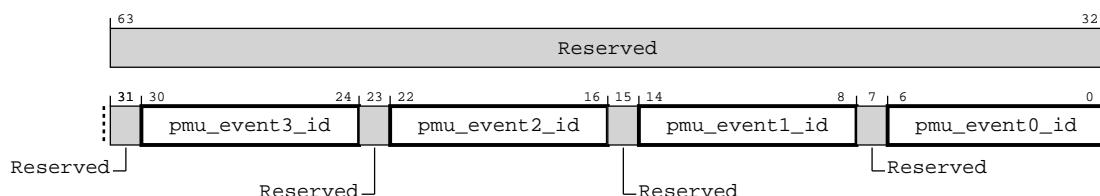
#### Usage constraints

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 8-933: por\_rnd\_pmu\_event\_sel**



**Table 8-945: por\_rnd\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	
[30:24]	pmu_event3_id	RN-D PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[23]	Reserved	Reserved	RO	
[22:16]	pmu_event2_id	RN-D PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[15]	Reserved	Reserved	RO	
[14:8]	pmu_event1_id	RN-D PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0
[7]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-D PMU Event 0 ID</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> Port S0 RDataBeats</p> <p><b>0x02</b> Port S1 RDataBeats</p> <p><b>0x03</b> Port S2 RDataBeats</p> <p><b>0x04</b> RXDAT flits received</p> <p><b>0x05</b> TXDAT flits sent</p> <p><b>0x06</b> Total TXREQ flits sent</p> <p><b>0x07</b> Retried TXREQ flits sent</p> <p><b>0x08</b> RRT occupancy count overflow_slice0</p> <p><b>0x09</b> WRT occupancy count overflow</p> <p><b>0x0A</b> Replayed TXREQ flits</p> <p><b>0x0B</b> WriteCancel sent</p> <p><b>0x0C</b> Port S0 WDataBeats</p> <p><b>0x0D</b> Port S1 WDataBeats</p> <p><b>0x0E</b> Port S2 WDataBeats</p> <p><b>0x0F</b> RRT allocation</p> <p><b>0x10</b> WRT allocation</p> <p><b>0x11</b> PADB occupancy count overflow</p> <p><b>0x12</b> RPDB occupancy count overflow</p> <p><b>0x13</b> RRT occupancy count overflow_slice1</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-D PMU Event 0 ID</p> <p><b>0x14</b> RRT occupancy count overflow_slice2</p> <p><b>0x15</b> RRT occupancy count overflow_slice3</p> <p><b>0x16</b> WRT request throttled</p> <p><b>0x17</b> RNI backpressure CHI LDB full</p> <p><b>0x18</b> RRT normal rd req occupancy count overflow_slice0</p> <p><b>0x19</b> RRT normal rd req occupancy count overflow_slice1</p> <p><b>0x1A</b> RRT normal rd req occupancy count overflow_slice2</p> <p><b>0x1B</b> RRT normal rd req occupancy count overflow_slice3</p> <p><b>0x1C</b> RRT PCIe RD burst req occupancy count overflow_slice0</p> <p><b>0x1D</b> RRT PCIe RD burst req occupancy count overflow_slice1</p> <p><b>0x1E</b> RRT PCIe RD burst req occupancy count overflow_slice2</p> <p><b>0x1F</b> RRT PCIe RD burst req occupancy count overflow_slice3</p> <p><b>0x20</b> RRT PCIe RD burst allocation</p> <p><b>0x21</b> Compressed AWID ordering</p> <p><b>0x22</b> Atomic data buffer allocation</p> <p><b>0x23</b> Atomic data buffer occupancy</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-D PMU Event 0 ID</p> <p><b>0x24</b> RRT occupancy count overflow_slice4</p> <p><b>0x25</b> RRT occupancy count overflow_slice5</p> <p><b>0x26</b> RRT occupancy count overflow_slice6</p> <p><b>0x27</b> RRT occupancy count overflow_slice7</p> <p><b>0x28</b> RRT normal rd req occupancy count overflow_slice4</p> <p><b>0x29</b> RRT normal rd req occupancy count overflow_slice5</p> <p><b>0x2A</b> RRT normal rd req occupancy count overflow_slice6</p> <p><b>0x2B</b> RRT normal rd req occupancy count overflow_slice7</p> <p><b>0x2C</b> RRT PCIe RD burst req occupancy count overflow_slice4</p> <p><b>0x2D</b> RRT PCIe RD burst req occupancy count overflow_slice5</p> <p><b>0x2E</b> RRT PCIe RD burst req occupancy count overflow_slice6</p> <p><b>0x2F</b> RRT PCIe RD burst req occupancy count overflow_slice7</p> <p><b>0x30</b> RRT CBUSY throttled</p> <p><b>0x31</b> RRT CBUSY 3</p> <p><b>0x32</b> RRT CBUSY 2</p> <p><b>0x33</b> RRT CBUSY 1</p> <p><b>0x34</b> RRT CBUSY 0</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-D PMU Event 0 ID</p> <p><b>0x35</b> WRT CBUSY throttled</p> <p><b>0x36</b> WRT CBUSY 3</p> <p><b>0x37</b> WRT CBUSY 2</p> <p><b>0x38</b> WRT CBUSY 1</p> <p><b>0x39</b> WRT CBUSY 0</p> <p><b>0x3A</b> RRT TXREQ dispatched occupancy count overflow_slice0</p> <p><b>0x3B</b> RRT TXREQ dispatched occupancy count overflow_slice1</p> <p><b>0x3C</b> RRT TXREQ dispatched occupancy count overflow_slice2</p> <p><b>0x3D</b> RRT TXREQ dispatched occupancy count overflow_slice3</p> <p><b>0x3E</b> RRT TXREQ dispatched occupancy count overflow_slice4</p> <p><b>0x3F</b> RRT TXREQ dispatched occupancy count overflow_slice5</p> <p><b>0x40</b> RRT TXREQ dispatched occupancy count overflow_slice6</p> <p><b>0x41</b> RRT TXREQ dispatched occupancy count overflow_slice7</p> <p><b>0x42</b> WRT TXREQ dispatched occupancy count overflow</p>	RW	0b0

### 8.3.15.17 por\_rnd\_syscoreq\_ctl

Functions as the RN-D DVM domain control register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por\_rnd\_syscoack\_status.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

### Address offset

0x1900

### Type

RW

### Reset value

See individual bit resets

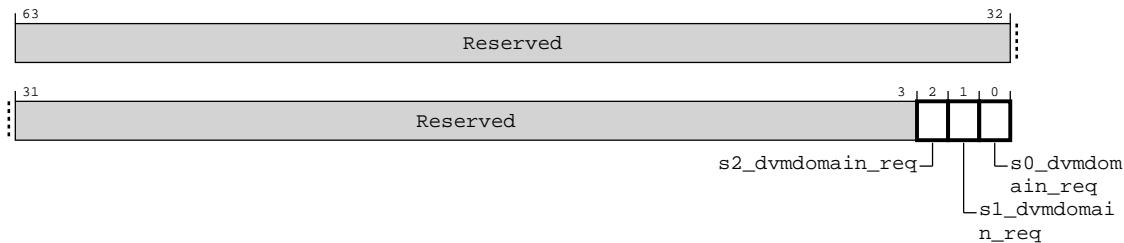
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-934: por\_rnd\_syscoreq\_ctl**



**Table 8-946: por\_rnd\_syscoreq\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	s2_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S2	RW	0b0
[1]	s1_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S1	RW	0b0
[0]	s0_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S0	RW	0b0

### 8.3.15.18 por\_rnd\_syscoack\_status

Functions as the RN-D DVM domain status register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por\_rnd\_syscoreq\_ctl.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x1908

### Type

RO

### Reset value

See individual bit resets

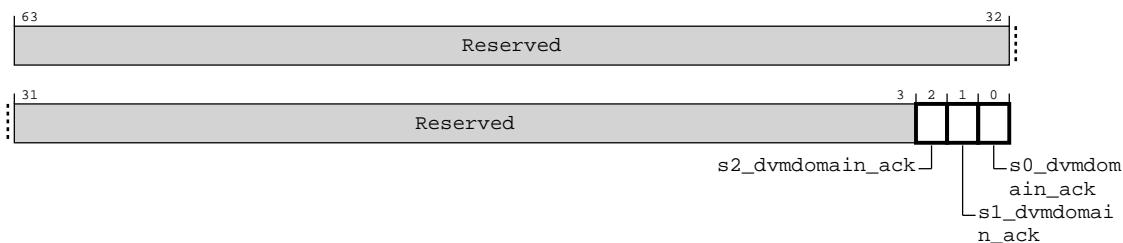
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-935: por\_rnd\_syscoack\_status**



**Table 8-947: por\_rnd\_syscoack\_status attributes**

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	
[2]	s2_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S2	RO	0b0
[1]	s1_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S1	RO	0b0
[0]	s0_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S0	RO	0b0

### 8.3.16 RN-I register summary

The following table describes the registers for the relevant component.

**Table 8-948: por\_rni\_cfg register summary**

Offset	Name	Type	Description
0x0	por_rni_node_info	RO	<a href="#">por_rni_node_info</a>
0x80	por_rni_child_info	RO	<a href="#">por_rni_child_info</a>
0x980	por_rni_rcr	RW	<a href="#">por_rni_rcr</a>
0x988	por_rni_scr	RW	<a href="#">por_rni_scr</a>
0x900	por_rni_unit_info	RO	<a href="#">por_rni_unit_info</a>
0x908	por_rni_unit_info2	RO	<a href="#">por_rni_unit_info2</a>
0xA00	por_rni_cfg_ctl	RW	<a href="#">por_rni_cfg_ctl</a>
0xA08	por_rni_cfg_ctl2	RW	<a href="#">por_rni_cfg_ctl2</a>
0xA10	por_rni_aux_ctl	RW	<a href="#">por_rni_aux_ctl</a>

Offset	Name	Type	Description
0xA18 : 0xA28	por_rni_s0-2_port_control	RW	<a href="#">por_rni_s0-2_port_control</a>
0xA30 : 0xA40	por_rni_s0-2_mpam_control	RW	<a href="#">por_rni_s0-2_mpam_control</a>
0xA80 : 0xAC0	por_rni_s0-2_qos_control	RW	<a href="#">por_rni_s0-2_qos_control</a>
0xA88 : 0xAC8	por_rni_s0-2_qos_lat_tgt	RW	<a href="#">por_rni_s0-2_qos_lat_tgt</a>
0xA90 : 0xAD0	por_rni_s0-2_qos_lat_scale	RW	<a href="#">por_rni_s0-2_qos_lat_scale</a>
0xA98 : 0xAD8	por_rni_s0-2_qos_lat_range	RW	<a href="#">por_rni_s0-2_qos_lat_range</a>
0xD900	por_rni_pmu_event_sel	RW	<a href="#">por_rni_pmu_event_sel</a>

### 8.3.16.1 por\_rni\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

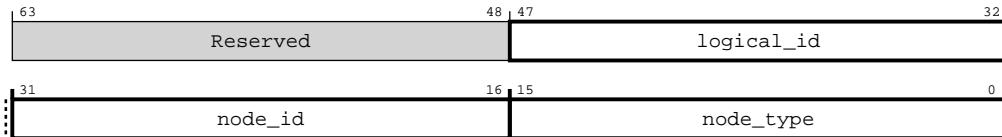
##### Usage constraints

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 8-936: por\_rni\_node\_info**



**Table 8-949: por\_rni\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x000A

### 8.3.16.2 por\_rni\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

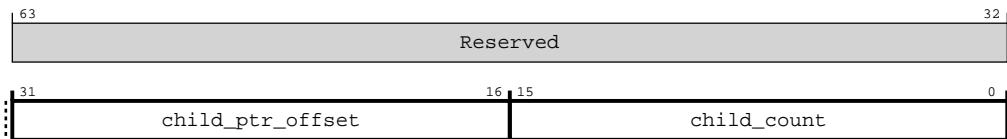
##### Usage constraints

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 8-937: por\_rni\_child\_info**



**Table 8-950: por\_rni\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0

Bits	Name	Description	Type	Reset
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0x0

### 8.3.16.3 por\_rni\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

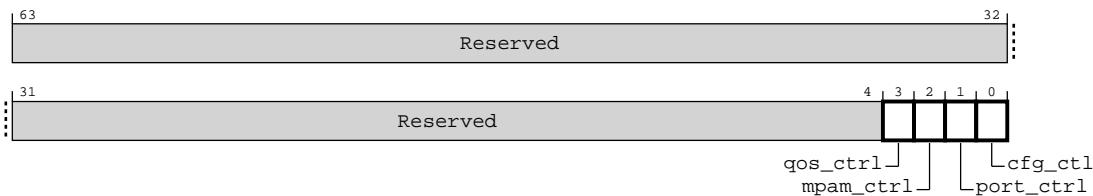
##### Usage constraints

This register is owned in the Root space and is 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 8-938: por\_rni\_rcr**



**Table 8-951: por\_rni\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	qos_ctrl	Allows Root override of the QoS control registers	RW	0b0
[2]	mpam_ctrl	Allows Root override of the AXI port MPAM override register	RW	0b0
[1]	port_ctrl	Allows Root override of the AXI port control registers	RW	0b0
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.16.4 por\_rni\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

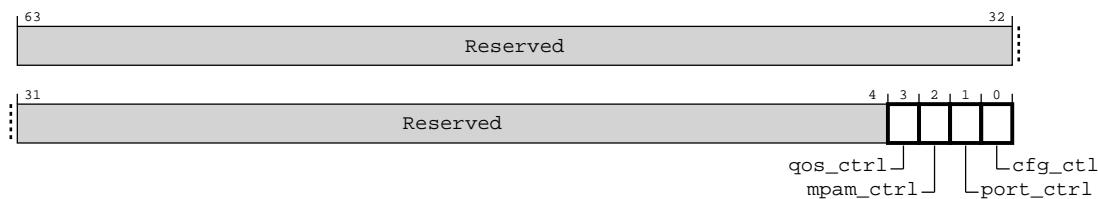
##### Usage constraints

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 8-939: por\_rni\_scr**



**Table 8-952: por\_rni\_scr attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	qos_ctrl	Allows Secure override of the QoS control registers	RW	0b0
[2]	mpam_ctrl	Allows Secure override of the AXI port MPAM override register	RW	0b0
[1]	port_ctrl	Allows Secure override of the AXI port control registers	RW	0b0
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.16.5 por\_rni\_unit\_info

Provides component identification information for RN-I.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

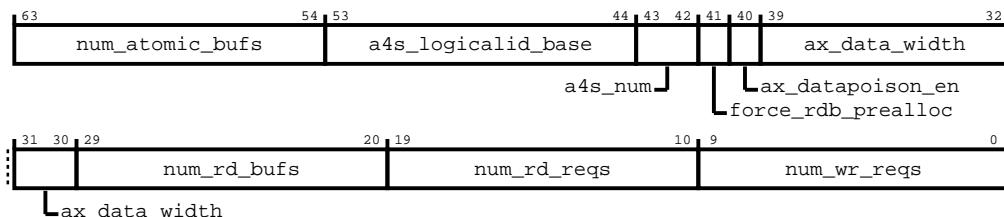
##### Usage constraints

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 8-940: por\_rni\_unit\_info**



**Table 8-953: por\_rni\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:54]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[53:44]	a4s_logicalid_base	AXI4Stream interfaces logical ID base	RO	0x0
[43:42]	a4s_num	Number of AXI4Stream interfaces present	RO	0x0
[41]	force_rdb_prealloc	Force read data buffer preallocation  0b1 yes  0b0 no	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[40]	ax_datapoison_en	Data Poison enable on ACE-Lite/AXI4 interface  <b>0b1</b> Enabled  <b>0b0</b> Not enabled	RO	Configuration dependent
[39:30]	ax_data_width	AXI interface data width in bits	RO	Configuration dependent
[29:20]	num_rd_bufs	Number of read data buffers	RO	Configuration dependent
[19:10]	num_rd_reqs	Number of outstanding read requests	RO	Configuration dependent
[9:0]	num_wr_reqs	Number of outstanding write requests	RO	Configuration dependent

### 8.3.16.6 por\_rni\_unit\_info2

Provides additional component identification information for RN-I.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x908

##### Type

RO

##### Reset value

See individual bit resets

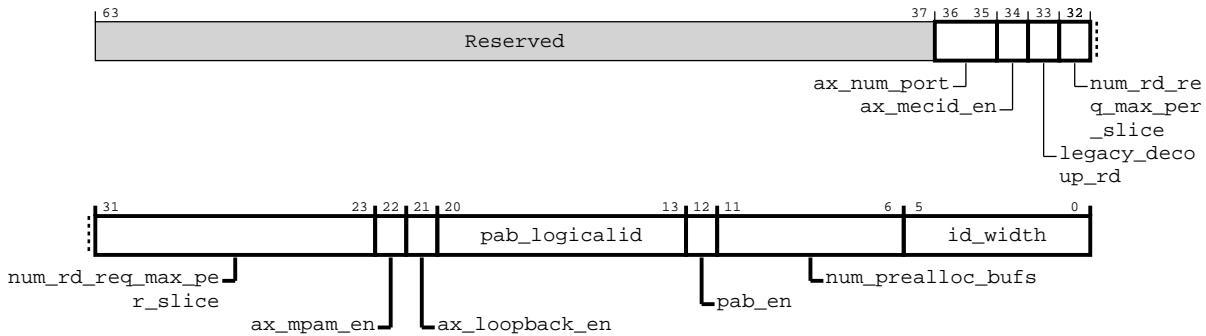
##### Usage constraints

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 8-941: por\_rni\_unit\_info2**



**Table 8-954: por\_rni\_unit\_info2 attributes**

Bits	Name	Description	Type	Reset
[63:37]	Reserved	Reserved	RO	-
[36:35]	ax_num_port	Number of ACE-Lite/AXI4 interfaces present	RO	Configuration dependent
[34]	ax_mecid_en	MECID enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[33]	legacy_decoup_rd	Legacy decoupled read mode, no read burst propagation	RO	Configuration dependent
[32:23]	num_rd_req_max_per_slice	Number of read request entires per slice	RO	Configuration dependent
[22]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[21]	ax_loopback_en	LoopBack enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[20:13]	pab_logicalid	PUB AUB bridge Logical ID	RO	0x0
[12]	pab_en	PUB AUB bridge enable	RO	0x0
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[11:6]	num_prealloc_bufs	Number of Pre-allocated Read Data Buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite subordinate ports	RO	Configuration dependent

### 8.3.16.7 por\_rni\_cfg\_ctl

Functions as the configuration control register. Specifies the current mode.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rni\_rcr.cfg\_ctl

##### Secure group override

por\_rni\_scr.cfg\_ctl

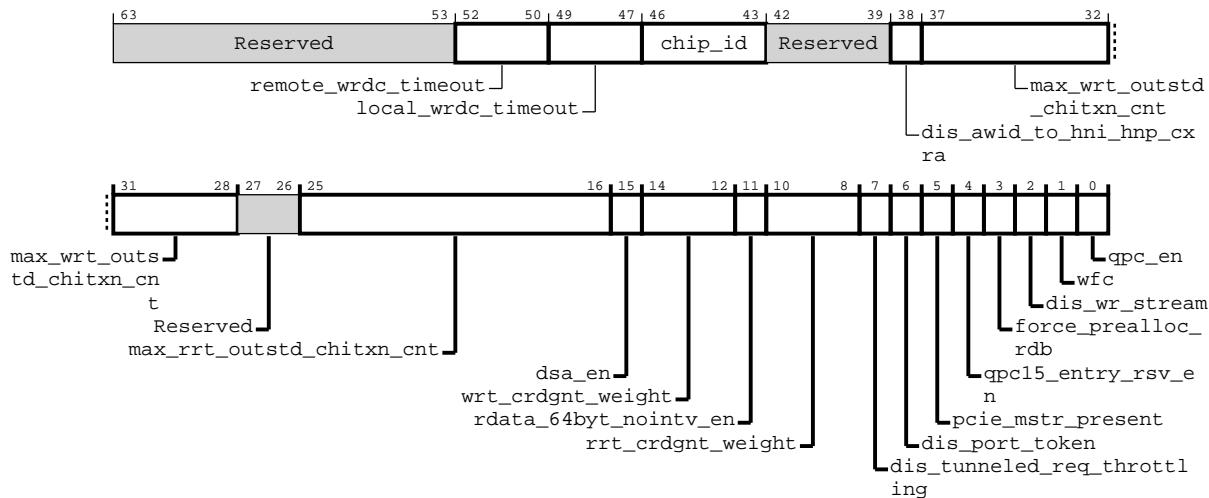
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.cfg\_ctl bit and por\_rni\_rcr.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 8-942: por\_rni\_cfg\_ctl**



**Table 8-955: por\_rni\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52:50]	remote_wrdc_timeout	Configurable write data cancel timeout value for remote traffic. <b>0b000</b> ~2000 cycles <b>0b001</b> ~4000 cycles <b>0b010</b> ~8000 cycles <b>0b011</b> ~16000 cycles (default) <b>0b100</b> ~32000 cycles (all other values rsvd)	RW	0b011
[49:47]	local_wrdc_timeout	Configurable write data cancel timeout value for local traffic. <b>0b000</b> ~2000 cycles <b>0b001</b> ~4000 cycles <b>0b010</b> ~8000 cycles <b>0b011</b> ~16000 cycles (default) <b>0b100</b> ~32000 cycles (all other values rsvd)	RW	0b010

Bits	Name	Description	Type	Reset
[46:43]	chip_id	Configurable ChipID for this RNX instance. Must be correctly set for proper handling of remote traffic to HNI/HNP. Only supports values 0..3. Two MSB's is reserved.	RW	0b0000
[42:39]	Reserved	Reserved	RO	-
[38]	dis_awid_to_hni_hnp_cxra	If set, disables compressed AWID to HNI, HNP and CXRA, also disables compressed AWID based ordering. Set this bit if uniq-ID write performance is needed.	RW	0b0
[37:28]	max_wrt_outstd_chitxn_cnt	Maximum number of outstanding writes allowed on CHI-side	RW	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	max_rrt_outstd_chitxn_cnt	Maximum number of outstanding reads allowed on CHI-side	RW	Configuration dependent
[15]	dsa_en	Enables Direct Subordinate Access feature to SBSX/MTSX. When setting this bit, dis_compack_on_writes must be set to 1, because dsa mode does not work with compack flow. Requires POR_DSA_EN_PARAM to be set, otherwise this config bit has no effect.	RW	0b0
[14:12]	wrt_crdgnt_weight	Determines weight of credit grant allocated to retried writes in presence of pending retried reads	RW	0b001
[11]	rdata_64byt_nointv_en	Enables no interleaving property on normal memory read data within 64B granule when set	RW	0b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant allocated to retried reads in presence of pending retried writes	RW	0b100
[7]	dis_tunneled_req_throttling	Disables retry based throttling of tunneled write requests	RW	0b0
[6]	dis_port_token	If set, disables per port reservation in the tracker(rd and wr)	RW	0b1
[5]	pcie_mstr_present	Indicates PCIe requester/manager is present; must be set if PCIe requester/manager is present upstream of RN-I or RN-D	RW	0b0
[4]	qpc15_entry_rsv_en	Enables QPC15 entry reservation <b>0b1</b> Reserves tracker entry for QoS15 requests <b>0b0</b> Does not reserve tracker entry for QoS15 requests <b>NOTE</b> Only valid and applicable when por_rni_qpc_en is set	RW	0b0
[3]	force_prealloc_rdb	When set, all reads from the RN-I are sent with a preallocated read data buffer	RW	Configuration dependent
[2]	dis_wr_stream	Disables streaming of ordered writes when set	RW	0b0
[1]	wfc	When set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	0b0
[0]	qpc_en	When set, enables QPC-based scheduling using two QoS priority classes (QoS15 and non-QoS15)	RW	0b1

### 8.3.16.8 por\_rni\_cfg\_ctl2

Functions as the configuration control register. Specifies the current mode.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rni\_rcr.cfg\_ctl

##### Secure group override

por\_rni\_scr.cfg\_ctl

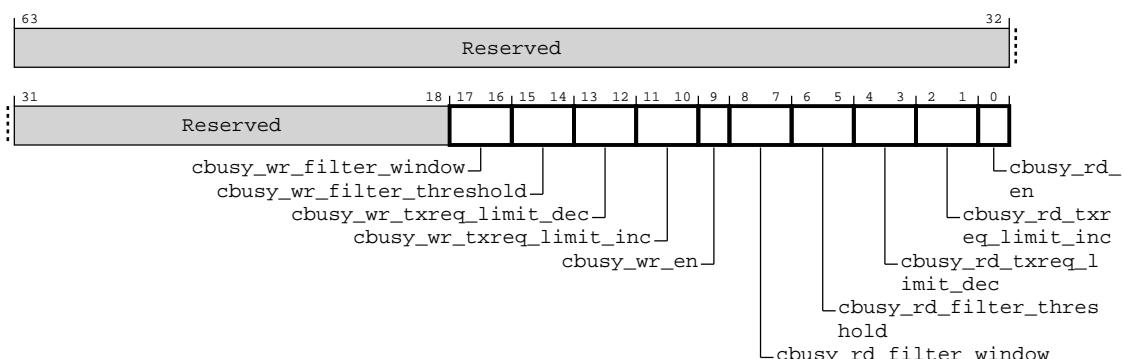
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.cfg\_ctl and por\_rni\_rcr.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 8-943: por\_rni\_cfg\_ctl2**



**Table 8-956: por\_rni\_cfg\_ctl2 attributes**

Bits	Name	Description	Type	Reset
[63:18]	Reserved	Reserved	RO	
[17:16]	cbusy_wr_filter_window	<p>Number of CBusy write responses in one sampling window.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> <li><b>0b00</b> 256 (default)</li> <li><b>0b01</b> 64</li> <li><b>0b10</b> 128</li> <li><b>0b11</b> 512</li> </ul>	RW	0b00
[15:14]	cbusy_wr_filter_threshold	<p>Fraction of CBusy write responses in the sampling window necessary to be considered valid sample of that CBusy value.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 1/16 (default)</li> <li><b>0b01</b> 1/32</li> <li><b>0b10</b> 1/8</li> <li><b>0b11</b> 1/4</li> </ul>	RW	0b00
[13:12]	cbusy_wr_txreq_limit_dec	<p>Dynamic write TXREQ limit decrement. Controls how quickly the dynamic write TXREQ limit is decreased when CBusy indicates a value of 3.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00

Bits	Name	Description	Type	Reset
[11:10]	cbusy_wr_txreq_limit_inc	<p>Dynamic write TXREQ limit decrement. Controls how quickly the dynamic write TXREQ limit is decreased when CBusy indicates a value of 2.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[9]	cbusy_wr_en	Enables CBusy throttling of write txreq within the RNI	RW	0b1
[8:7]	cbusy_rd_filter_window	<p>Number of CBusy read responses in one sampling window.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 256 (default)</li> <li><b>0b01</b> 64</li> <li><b>0b10</b> 128</li> <li><b>0b11</b> 512</li> </ul>	RW	0b00
[6:5]	cbusy_rd_filter_threshold	<p>Fraction of CBusy read responses in the sampling window necessary to be considered valid sample of that CBusy value.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 1/16 (default)</li> <li><b>0b01</b> 1/32</li> <li><b>0b10</b> 1/8</li> <li><b>0b11</b> 1/4</li> </ul>	RW	0b00

Bits	Name	Description	Type	Reset
[4:3]	cbusy_rd_txreq_limit_dec	<p>Dynamic read TXREQ limit decrement. Controls how quickly the dynamic read TXREQ limit is decreased when CBusy indicates a value of 3.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[2:1]	cbusy_rd_txreq_limit_inc	<p>Dynamic read TXREQ limit increment. Controls how quickly the dynamic read TXREQ limit is increased when CBusy indicates values less than 2.</p> <p>The possible values are</p> <ul style="list-style-type: none"> <li><b>0b00</b> 4 (default)</li> <li><b>0b01</b> 2</li> <li><b>0b10</b> 8</li> <li><b>0b11</b> 16</li> </ul>	RW	0b00
[0]	cbusy_rd_en	Enables CBusy throttling of read txreq within the RNI	RW	0b1

### 8.3.16.9 por\_rni\_aux\_ctl

Functions as the auxiliary control register for RN-I.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA10

##### Type

RW

##### Reset value

See individual bit resets

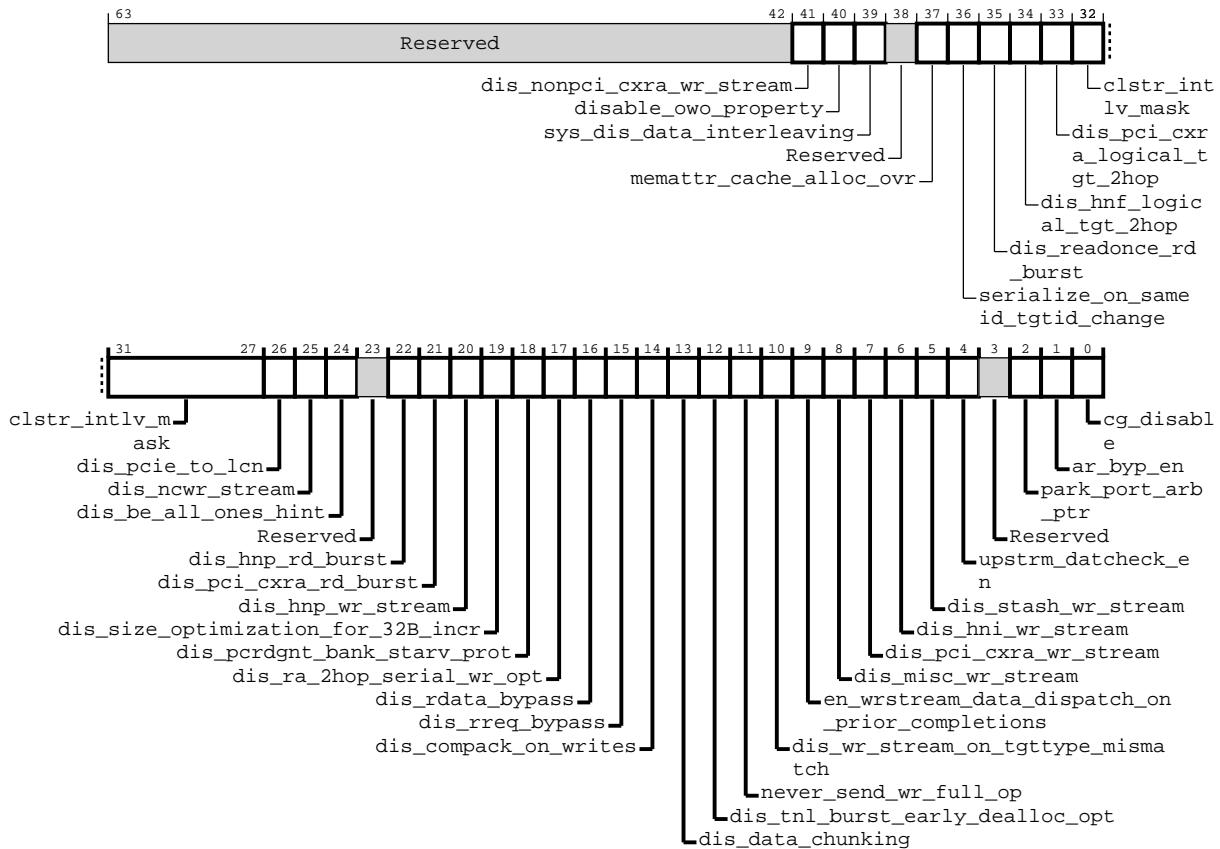
## Usage constraints

This register is owned in the Root space and is 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 8-944: por\_rni\_aux\_ctl**



**Table 8-957: por\_rni\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41]	dis_nonpci_cxra_wr_stream	Disables streaming of ordered writes to (non-PCI)-CXRA when set	RW	0b0
[40]	disable_owo_property	When set RN-D AXI interface will behave as if AXI Ordered_Write_Observation property is disabled for all writes.	RW	0b0

Bits	Name	Description	Type	Reset
[39]	sys_dis_data_interleaving	<p>System optimized disable read DATA interleaving for all ports. Disables all read data interleaving, including atomic read data being returned for all AXI ports. Read burst preservation is enabled same as in normal mode, but this requires certain system level restrictions:</p> <ol style="list-style-type: none"> <li>1. Cannot set SYS_DIS_DATA_INTERLEAVING for multi-chip systems. Support for remote HN-P is a future feature.</li> <li>2. When setting SYS_DIS_DATA_INTERLEAVING for an RN-I/RN-D it is required for that RN-I/RN-D to target only one HN-P with read bursts. It is also required for the path to a single remote HN-P, when remote support is available, to only go through a single CCG and for the ccg_rni to also have SYS_DIS_DATA_INTERLEAVING set.</li> <li>3. The AXI subordinate downstream of HN-P must not interleave read burst data.</li> <li>4. Must have NUM_RD_REQ == NUM_RD_BUF and NUM_RD_REQ &lt;= 256, otherwise this bit has no effect.</li> <li>5. The sum of the maximum number of beats of a cracked burst per each port must fit wholly within an arslice - breaking this rule will result in a hang. Must comprehend DIS_PORT_TOKEN AND QPC15_ENTRY_RSV settings, which will limit number of available entries</li> </ol>	RW	0b0
[38]	Reserved	Reserved	RO	-
[37]	memattr_cache_alloc_ovr	If set, overrides the incoming memattr cacheable and allocate attributes both to 1 for non-device and bufferable requests	RW	0b0
[36]	serialize_on_sameid_tgtid_change	If set, serializes request issue for sameid writes to cxra when tgtid mismatches previous write request.	RW	0b0
[35]	dis_readonce_rd_burst	If set, disables read burst for ReadOnce from AXI.	RW	0b0
[34]	dis_hnf_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical target is pci_cxra and logical target is hnf, otherwise may tunneling/2hop to RA if interleaving granularity settings allow.	RW	0b0
[33]	dis_pci_cxra_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical and logical target is pci_cxra, otherwise tunneling/2hop to RA.	RW	0b0

Bits	Name	Description	Type	Reset
[32:27]	clstr_intlv_mask	Encoded static mask for max interleave granularity supported. When this setting is less than or equal to rnsam's programmed interleave granularity for a write to pci_cxra, tunneling/2hop flow will be used.  <b>0b111111</b> 64B <b>0b111110</b> 128B <b>0b111100</b> 256B <b>0b111000</b> 512B <b>0b110000</b> 1024B <b>0b100000</b> 2048B <b>0b000000</b> 4096B <b>others</b> Reserved	RW	0b000000
[26]	dis_PCIE_to_lcn	If set, all pcie traffic sent directly to HNF/CCG, bypasses LCN. Only has effect when pcie_mstr_present	RW	0b1
[25]	dis_ncwr_stream	Disables streaming of ordered non-cacheable writes when set	RW	0b0
[24]	dis_be_all_ones_hint	If set, disables hint to HNF which signals all BE=1's on writes	RW	0b0
[23]	Reserved	Reserved	RO	-
[22]	dis_hnp_rd_burst	If set, disables read burst to HNP on CHI request flits . Read burst on CHI is supported in either non-decoupled RDB configuration or when decoupled (NUM_RD_BUF < NUM_RD_REQ) and LEGACY_DECOUP_RD = 0 and read data chunking is enabled on AXI.	RW	0b0
[21]	dis_pci_cxra_rd_burst	If set, disables read burst to PCI-CXRA on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	0b0
[20]	dis_hnp_wr_stream	Disables streaming of ordered writes to HNP when set	RW	0b0
[19]	dis_size_optimization_for_32B_incr	If set, disables the size related optimization for a 32B INCR burst (rh-2512). Only applies to writes.	RW	0b0
[18]	dis_pcrgnt_bank_starv_prot	If set, disables across arslice starvation protection	RW	0b0
[17]	dis_ra_2hop_serial_wr_opt	If set, disables 2 hop indication to ra for serialized writes; will indicate 3 hop	RW	0b0
[16]	dis_rdata_bypass	If set, disables read data bypass path	RW	0b0
[15]	dis_req_bypass	If set, disables read request bypass path	RW	0b0
[14]	dis_compack_on_writes	If set, disables comp_ack on streaming writes. WrData is used for ordering writes	RW	0b1

Bits	Name	Description	Type	Reset
[13]	dis_data_chunking	If set, disables the data chunking feature	RW	0b0
[12]	dis_tnl_burst_early_dealloc_opt	If set, disables the optimization related to early deallocation of tunneled writes for intermediate txns of burst	RW	0b0
[11]	never_send_wr_full_op	If set, RNI will never send WR FULL op. All write ops will be of PTL type	RW	0b0
[10]	dis_wr_stream_on_tgtype_mismatch	If set, serializes first write when moving from one tgtype to another	RW	0b0
[9]	en_wrstream_data_dispatch_on_prior_completions	If set, data dispatch for streaming writes waits for completion of all older writes	RW	0b0
[8]	dis_misc_wr_stream	Disables streaming of ordered writes with following attributes when set : Device memory or EWA=0 or Excl=1	RW	0b0
[7]	dis_pci_cxra_wr_stream	Disables streaming of ordered writes to PCI-CXRA when set	RW	0b0
[6]	dis_hni_wr_stream	Disables streaming of ordered writes to HNI when set	RW	0b0
[5]	dis_stash_wr_stream	Disables streaming of ordered WrUniqStash when set	RW	0b0
[4]	upstrm_datcheck_en	Upstream supports Datacheck	RW	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2]	park_port_arb_ptr	Parks the AXI port arbitration pointer for Burst	RW	0b0
[1]	ar_byp_en	AR bypass enable; enables bypass path in the AR pipeline	RW	0b1
[0]	cg_disable	Disables clock gating when set	RW	0b0

### 8.3.16.10 por\_rni\_s0-2\_port\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface settings.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA18 + #{index}\*8}

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_rni\_rcr.port\_ctrl

### Secure group override

por\_rni\_scr.port\_ctrl

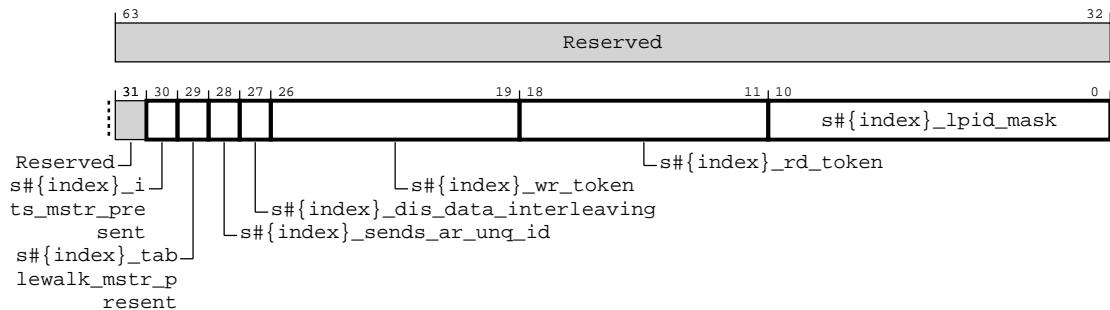
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.port\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.port\_ctrl bit and por\_rni\_rcr.port\_ctrl 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 8-945: por\_rni\_s0-2\_port\_control**



**Table 8-958: por\_rni\_s0-2\_port\_control attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-
[30]	s#{index}_its_mstr_present	Must be set if translation table walk manager present such as TCU or GIC for non-PCIE case. This affects RNI AW channel only.	RW	0b0
[29]	s#{index}_tablewalk_mstr_present	Must be set if translation table walk manager present such as TCU or GIC. This affects RNI AR channel only.	RW	0b0
[28]	s#{index}_sends_ar_unq_id	If set, indicates AR transactions on Port#{index} are always Unique ID. This bit for a port must be set to 1 to enable Read Burst on the CHI side of RNI.	RW	0b0
[27]	s#{index}_dis_data_interleaving	If set, disables read DATA interleaving on RDATAS#{index} channel. This applies only to RDATA generated as a response to requests on AR channel. This does not apply to RDATA generated as a response to Atomic request on AW channel. I.e. RDATA of an Atomic op, on AW channel, may interleave with RDATA of an AR channel request. If setting cfg_ctl.wfc bit along with this bit, aux_ctl.dis_rreq_bypass must also be set, otherwise completion order will be violated.	RW	0b0

Bits	Name	Description	Type	Reset
[26:19]	s#{index}_wr_token	Port S#{index} reserved token count for AW channel This must be less than the number of Wr requests(RNID_NUM_XRT_REQ) on AW achnnel	RW	8'b0000_0000
[18:11]	s#{index}_rd_token	Port S#{index} reserved token count for AR channel per slice This should be less than the number of Rd requests(RNID_NUM_XRT_SLICE_REQ) per slice on AR achnnel	RW	8'b0000_0000
[10:0]	s#{index}_lpid_mask	Port S#{index} LPID mask  <b>LPIID[0]</b> Equal to the result of UnaryOR of BitwiseAND of LPID mask and AXID (LPIID[0] =  (AXID & mask)); specifies which AXID bit is reflected in the LSB of LPID  <b>LPIID[2:1]</b> Equal to port ID[1:0]; the MSB of LPID contains port ID	RW	11'b000_0000_0000

### 8.3.16.11 por\_rni\_s0-2\_mpam\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface MPAM override values

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xA30 + #{index}\*8

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rni\_rcr.mpam\_ctrl

##### Secure group override

por\_rni\_scr.mpam\_ctrl

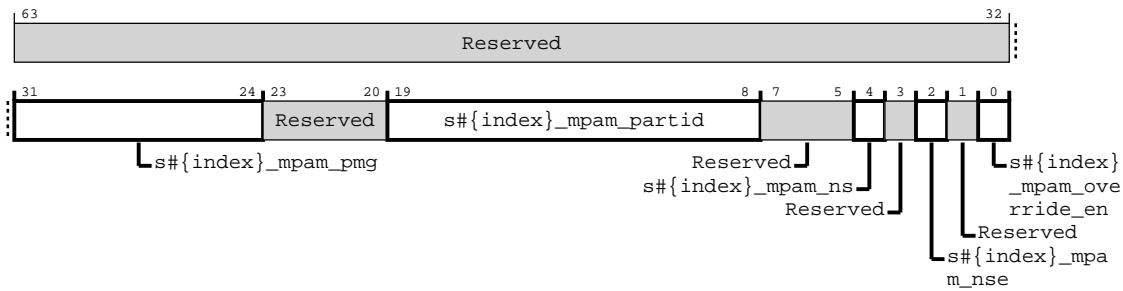
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.mpam\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.mpam\_ctrl bit and por\_rni\_rcr.mpam\_ctrl 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 8-946: por\_rni\_s0-2\_mpam\_control**



**Table 8-959: por\_rni\_s0-2\_mpam\_control attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:24]	s#{index}_mpam_pmg	Port S#{index} MPAM_PMG value	RW	0b0
[23:20]	Reserved	Reserved	RO	
[19:8]	s#{index}_mpam_partid	Port S#{index} MPAM_PARTID value	RW	0b0
[7:5]	Reserved	Reserved	RO	
[4]	s#{index}_mpam_ns	Port S#{index} MPAM_NS value	RW	0b0
[3]	Reserved	Reserved	RO	
[2]	s#{index}_mpam_nse	Port S#{index} MPAM_NSE value	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	s#{index}_mpam_override_en	Port S#{index} MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Note that when RNID_AXMPAM_EN_PARAM is set to 0, MPAM override value is always used irrespective of this bit value	RW	0b0

### 8.3.16.12 por\_rni\_s0-2\_qos\_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE subordinate interface.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA80 + #{index}\*32}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rni\_rcr.qos\_ctrl

## Secure group override

por\_rni\_scr.qos\_ctrl

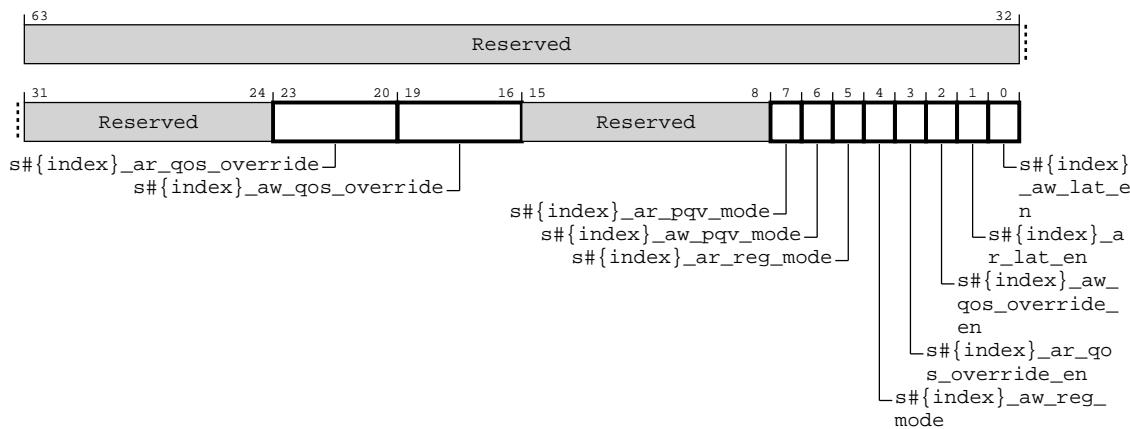
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.qos\_ctrl bit and por\_rni\_rcr.qos\_ctrl 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 8-947: por\_rni\_s0-2\_qos\_control**



**Table 8-960: por\_rni\_s0-2\_qos\_control attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	s#{index}_ar_qos_override	AR QoS override value for port S#{index}	RW	0b0000
[19:16]	s#{index}_aw_qos_override	AW QoS override value for port S#{index}	RW	0b0000
[15:8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7]	s#{index}_ar_pqv_mode	<p>Configures the QoS regulator mode for read transactions during period mode</p> <p><b>0b0</b> Normal mode; QoS value is stable when the manager is idle</p> <p><b>0b1</b> Quiesce high mode; QoS value tends to the maximum value when the manager is idle</p>	RW	0b0
[6]	s#{index}_aw_pqv_mode	<p>Configures the QoS regulator mode for write transactions during period mode</p> <p><b>0b0</b> Normal mode; QoS value is stable when the manager is idle</p> <p><b>0b1</b> Quiesce high mode; QoS value tends to the maximum value when the manager is idle</p>	RW	0b0
[5]	s#{index}_ar_reg_mode	<p>Configures the QoS regulator mode for read transactions</p> <p><b>0b0</b> Latency mode</p> <p><b>0b1</b> Period mode; used for bandwidth regulation</p>	RW	0b0
[4]	s#{index}_aw_reg_mode	<p>Configures the QoS regulator mode for write transactions</p> <p><b>0b0</b> Latency mode</p> <p><b>0b1</b> Period mode; used for bandwidth regulation</p>	RW	0b0
[3]	s#{index}_ar_qos_override_en	Enables port S#{index} AR QoS override; when set, allows QoS value on inbound AR transactions to be overridden	RW	0b0
[2]	s#{index}_aw_qos_override_en	Enables port S#{index} AW QoS override; when set, allows QoS value on inbound AW transactions to be overridden	RW	0b0
[1]	s#{index}_ar_lat_en	Enables port S#{index} AR QoS regulation when set	RW	0b0
[0]	s#{index}_aw_lat_en	Enables port S#{index} AW QoS regulation when set	RW	0b0

### 8.3.16.13 por\_rni\_s0-2\_qos\_lat\_tgt

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS target latency (in cycles) for regulations of port S#{index} read and write transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA88 + #{index}\*32

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rni\_rcr.qos\_ctrl

### Secure group override

por\_rni\_scr.qos\_ctrl

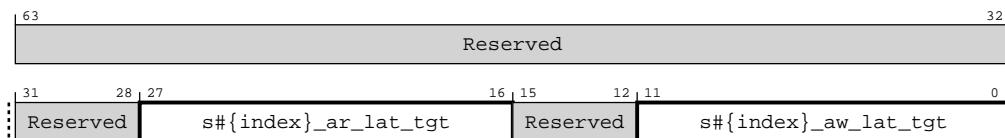
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.qos\_ctrl bit and por\_rni\_rcr.qos\_ctrl 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 8-948: por\_rni\_s0-2\_qos\_lat\_tgt**



**Table 8-961: por\_rni\_s0-2\_qos\_lat\_tgt attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:16]	s#{index}_ar_lat_tgt	Port S#{index} AR channel target latency; a value of 0 corresponds to no regulation	RW	0x000
[15:12]	Reserved	Reserved	RO	-
[11:0]	s#{index}_aw_lat_tgt	Port S#{index} AW channel target latency; a value of 0 corresponds to no regulation	RW	0x000

### 8.3.16.14 por\_rni\_s0-2\_qos\_lat\_scale

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the QoS target latency scale factor for port S#{index} read and write transactions. This register represents powers of two from the range  $2^{-5}$  to  $2^{-12}$ ; it is used to match a 16-bit integrator.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0xA90 + #{index}\*32}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rni\_rcr.qos\_ctrl

### Secure group override

por\_rni\_scr.qos\_ctrl

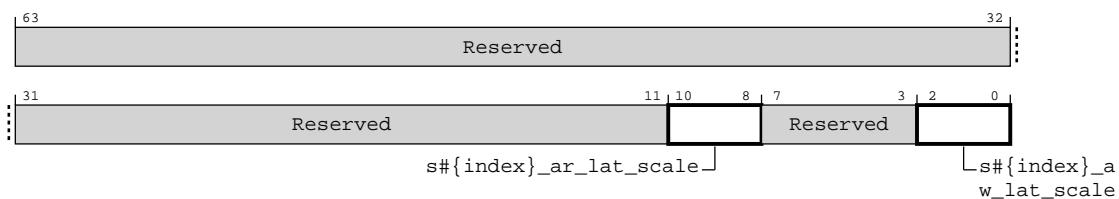
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.qos\_ctrl bit and por\_rni\_rcr.qos\_ctrl 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 8-949: por\_rni\_s0-2\_qos\_lat\_scale**



**Table 8-962: por\_rni\_s0-2\_qos\_lat\_scale attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[10:8]	s#{index}_ar_lat_scale	Port S#{index} AR QoS scale factor <b>0b000</b> $2^{-5}$ <b>0b001</b> $2^{-6}$ <b>0b010</b> $2^{-7}$ <b>0b011</b> $2^{-8}$ <b>0b100</b> $2^{-9}$ <b>0b101</b> $2^{-10}$ <b>0b110</b> $2^{-11}$ <b>0b111</b> $2^{-12}$	RW	0x0
[7:3]	Reserved	Reserved	RO	
[2:0]	s#{index}_aw_lat_scale	Port S#{index} AW QoS scale factor <b>0b000</b> $2^{-5}$ <b>0b001</b> $2^{-6}$ <b>0b010</b> $2^{-7}$ <b>0b011</b> $2^{-8}$ <b>0b100</b> $2^{-9}$ <b>0b101</b> $2^{-10}$ <b>0b110</b> $2^{-11}$ <b>0b111</b> $2^{-12}$	RW	0x0

### 8.3.16.15 por\_rni\_s0-2\_qos\_lat\_range

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the minimum and maximum QoS values generated by the QoS latency regulator for port S#{index} read and write transactions.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA98 + #{index}\*32}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rni\_rcr.qos\_ctrl

##### Secure group override

por\_rni\_scr.qos\_ctrl

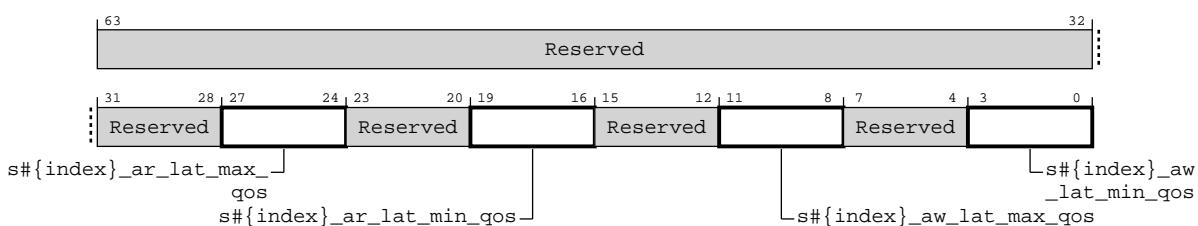
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rni\_scr.qos\_ctrl bit is set, Secure accesses to this register are permitted. If both the por\_rni\_scr.qos\_ctrl bit and por\_rni\_rcr.qos\_ctrl 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 8-950: por\_rni\_s0-2\_qos\_lat\_range**



**Table 8-963: por\_rni\_s0-2\_qos\_lat\_range attributes**

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	
[27:24]	s#[index]_ar_lat_max_qos	Port S#[index] AR QoS maximum value	RW	0x0
[23:20]	Reserved	Reserved	RO	
[19:16]	s#[index]_ar_lat_min_qos	Port S#[index] AR QoS minimum value	RW	0x0
[15:12]	Reserved	Reserved	RO	
[11:8]	s#[index]_aw_lat_max_qos	Port S#[index] AW QoS maximum value	RW	0x0
[7:4]	Reserved	Reserved	RO	
[3:0]	s#[index]_aw_lat_min_qos	Port S#[index] AW QoS minimum value	RW	0x0

### 8.3.16.16 por\_rni\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

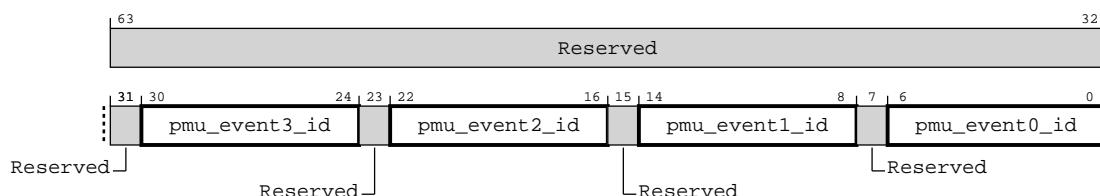
##### Usage constraints

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 8-951: por\_rni\_pmu\_event\_sel**



**Table 8-964: por\_rni\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	
[30:24]	pmu_event3_id	RN-I PMU Event 3 ID; see pmu_event0_id for encodings	RW	0b0
[23]	Reserved	Reserved	RO	
[22:16]	pmu_event2_id	RN-I PMU Event 2 ID; see pmu_event0_id for encodings	RW	0b0
[15]	Reserved	Reserved	RO	
[14:8]	pmu_event1_id	RN-I PMU Event 1 ID; see pmu_event0_id for encodings	RW	0b0
[7]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-I PMU Event 0 ID</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> Port S0 RDataBeats</p> <p><b>0x02</b> Port S1 RDataBeats</p> <p><b>0x03</b> Port S2 RDataBeats</p> <p><b>0x04</b> RXDAT flits received</p> <p><b>0x05</b> TXDAT flits sent</p> <p><b>0x06</b> Total TXREQ flits sent</p> <p><b>0x07</b> Retried TXREQ flits sent</p> <p><b>0x08</b> RRT occupancy count overflow_slice0</p> <p><b>0x09</b> WRT occupancy count overflow</p> <p><b>0x0A</b> Replayed TXREQ flits</p> <p><b>0x0B</b> WriteCancel sent</p> <p><b>0x0C</b> Port S0 WDataBeats</p> <p><b>0x0D</b> Port S1 WDataBeats</p> <p><b>0x0E</b> Port S2 WDataBeats</p> <p><b>0x0F</b> RRT allocation</p> <p><b>0x10</b> WRT allocation</p> <p><b>0x11</b> PADB occupancy count overflow</p> <p><b>0x12</b> RPDB occupancy count overflow</p> <p><b>0x13</b> RRT occupancy count overflow_slice1</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-I PMU Event 0 ID</p> <p><b>0x14</b> RRT occupancy count overflow_slice2</p> <p><b>0x15</b> RRT occupancy count overflow_slice3</p> <p><b>0x16</b> WRT request throttled</p> <p><b>0x17</b> RNI backpressure CHI LDB full</p> <p><b>0x18</b> RRT normal rd req occupancy count overflow_slice0</p> <p><b>0x19</b> RRT normal rd req occupancy count overflow_slice1</p> <p><b>0x1A</b> RRT normal rd req occupancy count overflow_slice2</p> <p><b>0x1B</b> RRT normal rd req occupancy count overflow_slice3</p> <p><b>0x1C</b> RRT PCIe RD burst req occupancy count overflow_slice0</p> <p><b>0x1D</b> RRT PCIe RD burst req occupancy count overflow_slice1</p> <p><b>0x1E</b> RRT PCIe RD burst req occupancy count overflow_slice2</p> <p><b>0x1F</b> RRT PCIe RD burst req occupancy count overflow_slice3</p> <p><b>0x20</b> RRT PCIe RD burst allocation</p> <p><b>0x21</b> Compressed AWID ordering</p> <p><b>0x22</b> Atomic data buffer allocation</p> <p><b>0x23</b> Atomic data buffer occupancy</p> <p><b>0x24</b> RRT occupancy count overflow_slice4</p> <p><b>0x25</b> RRT occupancy count overflow_slice5</p> <p><b>0x26</b> RRT occupancy count overflow_slice6</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	<p>RN-I PMU Event 0 ID</p> <p><b>0x27</b> RRT occupancy count overflow_slice7</p> <p><b>0x28</b> RRT normal rd req occupancy count overflow_slice4</p> <p><b>0x29</b> RRT normal rd req occupancy count overflow_slice5</p> <p><b>0x2A</b> RRT normal rd req occupancy count overflow_slice6</p> <p><b>0x2B</b> RRT normal rd req occupancy count overflow_slice7</p> <p><b>0x2C</b> RRT PCIe RD burst req occupancy count overflow_slice4</p> <p><b>0x2D</b> RRT PCIe RD burst req occupancy count overflow_slice5</p> <p><b>0x2E</b> RRT PCIe RD burst req occupancy count overflow_slice6</p> <p><b>0x2F</b> RRT PCIe RD burst req occupancy count overflow_slice7</p> <p><b>0x30</b> RRT CBUSY throttled</p> <p><b>0x31</b> RRT CBUSY 3</p> <p><b>0x32</b> RRT CBUSY 2</p> <p><b>0x33</b> RRT CBUSY 1</p> <p><b>0x34</b> RRT CBUSY 0</p> <p><b>0x35</b> WRT CBUSY throttled</p> <p><b>0x36</b> WRT CBUSY 3</p> <p><b>0x37</b> WRT CBUSY 2</p> <p><b>0x38</b> WRT CBUSY 1</p> <p><b>0x39</b> WRT CBUSY 0</p>	RW	0b0

Bits	Name	Description	Type	Reset
[6:0]	pmu_event0_id	RN-I PMU Event 0 ID <b>0x3A</b> RRT TXREQ dispatched occupancy count overflow_slice0 <b>0x3B</b> RRT TXREQ dispatched occupancy count overflow_slice1 <b>0x3C</b> RRT TXREQ dispatched occupancy count overflow_slice2 <b>0x3D</b> RRT TXREQ dispatched occupancy count overflow_slice3 <b>0x3E</b> RRT TXREQ dispatched occupancy count overflow_slice4 <b>0x3F</b> RRT TXREQ dispatched occupancy count overflow_slice5 <b>0x40</b> RRT TXREQ dispatched occupancy count overflow_slice6 <b>0x41</b> RRT TXREQ dispatched occupancy count overflow_slice7 <b>0x42</b> WRT TXREQ dispatched occupancy count overflow	RW	0b0

### 8.3.17 RNSAM register summary

The following table describes the registers for the relevant component.

**Table 8-965: por\_rnsam\_cfg register summary**

Offset	Name	Type	Description
0x0	por_rnsam_node_info	RO	<a href="#">por_rnsam_node_info</a>
0x80	por_rnsam_child_info	RO	<a href="#">por_rnsam_child_info</a>
0x980	por_rnsam_scr	RW	<a href="#">por_rnsam_scr</a>
0x988	por_rnsam_rcr	RW	<a href="#">por_rnsam_rcr</a>
0x900	por_rnsam_unit_info	RO	<a href="#">por_rnsam_unit_info</a>
0x908	por_rnsam_unit_info1	RO	<a href="#">por_rnsam_unit_info1</a>
{0-23} 0xC00 : 0xCB8	non_hash_mem_region_reg0-127	RW	<a href="#">non_hash_mem_region_reg0-127</a>
{24-151} 0x20C0 : 0x24B8			

Offset	Name	Type	Description
{0-23} 0xCC0 : 0xD78	non_hash_mem_region_cfg2_reg0-127	RW	<a href="#">non_hash_mem_region_cfg2_reg0-127</a>
{24-151} 0x24C0 : 0x28B8			
{0-15} 0xD80 : 0xDF8	non_hash_tgt_nodeid0-31	RW	<a href="#">non_hash_tgt_nodeid0-31</a>
{16-47} 0x2880 : 0x2978			
0x11A0	cml_port_aggr_mode_ctrl_reg	RW	<a href="#">cml_port_aggr_mode_ctrl_reg</a>
{0-3} 0x11A0 : 0x11B8	cml_port_aggr_mode_ctrl_reg1-12	RW	<a href="#">cml_port_aggr_mode_ctrl_reg1-12</a>
{4-19} 0x2A20 : 0x2A98			
0xE00 : 0xE18	sys_cache_grp_region0-3	RW	<a href="#">sys_cache_grp_region0-3</a>
{0-7} 0xE00 : 0xE38	hashed_tgt_grp_cfg1_region4-31	RW	<a href="#">hashed_tgt_grp_cfg1_region4-31</a>
{8-31} 0x3040 : 0x30F8			
{0-31} 0x3100 : 0x31F8	hashed_tgt_grp_cfg2_region0-31	RW	<a href="#">hashed_tgt_grp_cfg2_region0-31</a>
{0-7} 0xE40 : 0xE78	sys_cache_grp_secondary_reg0-3	RW	<a href="#">sys_cache_grp_secondary_reg0-3</a>
{8-31} 0x3240 : 0x32F8			

Offset	Name	Type	Description
{0-7} 0xE40 : 0xE78	hashed_target_grp_secondary_cfg1_reg4-31	RW	hashed_target_grp_secondary_cfg1_reg4-31
{8-31} 0x3240 : 0x32F8			
{0-31} 0x3300 : 0x33F8	hashed_target_grp_secondary_cfg2_reg0-31	RW	hashed_target_grp_secondary_cfg2_reg0-31
{0-31} 0x3400 : 0x34F8	hashed_target_grp_hash_cntl_reg0-31	RW	hashed_target_grp_hash_cntl_reg0-31
0xEA0	sys_cache_group_hn_count	RW	sys_cache_group_hn_count
{0-1} 0xEA0 : 0xEA8	hashed_target_group_hn_count_reg1-3	RW	hashed_target_group_hn_count_reg1-3
{2-3} 0x3710 : 0x3718			
0xEC0	sys_cache_grp_nonhash_nodeid	RW	sys_cache_grp_nonhash_nodeid
{0-5} 0xEC0 : 0xEE8	hashed_target_grp_nonhash_nodeid_reg1-6	RW	hashed_target_grp_nonhash_nodeid_reg1-6
{6-9} 0x3800 : 0x3818			
{0-31} 0xF00 : 0xFF8	sys_cache_grp_hn_nodeid_reg0-15	RW	sys_cache_grp_hn_nodeid_reg0-15
{0-31} 0xF00 : 0xFF8	hashed_target_grp_hnf_nodeid_reg16-31	RW	hashed_target_grp_hnf_nodeid_reg16-31
{32-63} 0x3600 : 0x36F8			
{0-31} 0x3600 : 0x36F8	hashed_target_grp_hnp_nodeid_reg0-15	RW	hashed_target_grp_hnp_nodeid_reg0-15

Offset	Name	Type	Description
{0-7} 0x1900 : 0x1938	hashed_target_grp_misc_nodeid_reg0-7	RW	hashed_target_grp_misc_nodeid_reg0-7
0x1980	hashed_target_grp_misc_tgts_lcn_bound_cfg_reg0	RW	hashed_target_grp_misc_tgts_lcn_bound_cfg_reg0
{0-7} 0x1800 : 0x1838	hashed_tgt_override_cam_reg0-7	RW	hashed_tgt_override_cam_reg0-7
0x1120	sys_cache_grp_cal_mode_reg	RW	sys_cache_grp_cal_mode_reg
{0-3} 0x1120 : 0x1138	hashed_target_grp_cal_mode_reg1-7	RW	hashed_target_grp_cal_mode_reg1-7
{4-7} 0x37A0 : 0x37B8			
0x1180	sys_cache_grp_hn_cpa_en_reg	RW	sys_cache_grp_hn_cpa_en_reg
{0-1} 0x1180 : 0x1188	hashed_target_grp_hnf_cpa_en_reg1-1	RW	hashed_target_grp_hnf_cpa_en_reg1-1
{2-3} 0x3730 : 0x3738			
{0-15} 0x3900 : 0x3978	hashed_target_grp_cpag_perhnf_reg0-15	RW	hashed_target_grp_cpag_perhnf_reg0-15
0x1190	sys_cache_grp_hn_cpa_grp_reg	RW	sys_cache_grp_hn_cpa_grp_reg
{0-1} 0x1190 : 0x1198	hashed_target_grp_cpa_grp_reg1-7	RW	hashed_target_grp_cpa_grp_reg1-7
{2-9} 0x3750 : 0x3788			
{0-3} 0x37C0 : 0x37D8	hashed_target_grp_hnf_lcn_bound_cfg_reg0-1	RW	hashed_target_grp_hnf_lcn_bound_cfg_reg0-1
{0-1} 0x37E0 : 0x37E8	hashed_target_grp_hnf_target_type_override_cfg_reg0-1	RW	hashed_target_grp_hnf_target_type_override_cfg_reg0-1

Offset	Name	Type	Description
{0-31} 0x3A00 : 0x3AF8	hashed_target_grp_compact_cpag_ctrl0-31	RW	hashed_target_grp_compact_cpag_ctrl0-31
{0-31} 0x3B00 : 0x3BF8	hashed_target_grp_compact_hash_ctrl0-31	RW	hashed_target_grp_compact_hash_ctrl0-31
0xE80	rnsam_hash_addr_mask_reg	RW	rnsam_hash_addr_mask_reg
0xE88	rnsam_hash_axi_id_mask_reg	RW	rnsam_hash_axi_id_mask_reg
0xE90	rnsam_region_cmp_addr_mask_reg	RW	rnsam_region_cmp_addr_mask_reg
{0-5} 0x11C0 : 0x11E8	cml_port_aggr_grp0-31_add_mask	RW	cml_port_aggr_grp0-31_add_mask
{6-37} 0x2B30 : 0x2C28			
{0-7} 0x2B00 : 0x2B38	cml_cpag_base_idx_grp0-3	RW	cml_cpag_base_idx_grp0-3
{0-2} 0x11F0 : 0x1200	cml_port_aggr_grp_reg0-12	RW	cml_port_aggr_grp_reg0-12
{3-14} 0x2C18 : 0x2C70			
0x1208	cml_port_aggr_ctrl_reg	RW	cml_port_aggr_ctrl_reg
{0-15} 0x1208 : 0x1280	cml_port_aggr_ctrl_reg1-6	RW	cml_port_aggr_ctrl_reg1-6
0xEB0	sys_cache_grp_sn_attr	RW	sys_cache_grp_sn_attr
0xEB8	sys_cache_grp_sn_attr1	RW	sys_cache_grp_sn_attr1
{0-7} 0x1140 : 0x1178	sys_cache_grp_sn_sam_cfg0-3	RW	sys_cache_grp_sn_sam_cfg0-3
{0-15} 0x1280 : 0x12F8	sam_qos_mem_region_reg0-15	RW	sam_qos_mem_region_reg0-15

Offset	Name	Type	Description
{0-15} 0x1340 : 0x13B8	sam_qos_mem_region_cfg2_reg0-15	RW	sam_qos_mem_region_cfg2_reg0-15
{0-511} 0x4000 : 0x4FF8	sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg1_reg0-511%64 cfg2_reg0-511%64	RW	sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg1_reg0-511%64 cfg2_reg0-511%64
{0-511} 0x5000 : 0x5FF8	sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg2_reg0-511%64 cfg1_reg0-511%64	RW	sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg2_reg0-511%64 cfg1_reg0-511%64
{0-63} 0x6000 : 0x61F8	sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8	RW	sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8
{0-63} 0x6200 : 0x63F8	sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8	RW	sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8
{0-63} 0x6400 : 0x65F8	sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8	RW	sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8
{0-31} 0x1000 : 0x10F8	sys_cache_grp_sn_nodeid_reg0-31	RW	sys_cache_grp_sn_nodeid_reg0-31
{0-63} 0x1400 : 0x15F8	sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32	RW	sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32
{0-31} 0x6600 : 0x66F8	sys_cache_grp_hashed_regions_sn_nodeid_reg0-31	RW	sys_cache_grp_hashed_regions_sn_nodeid_reg0-31
{0-3} 0x6800 : 0x6818	sys_cache_grp_hashed_regions_cxg_sa_nodeid_reg0-3	RW	sys_cache_grp_hashed_regions_cxg_sa_nodeid_reg0-3
0x1100	rnsam_status	RW	rnsam_status
0x1108	gic_mem_region_reg	RW	gic_mem_region_reg
0x1110	dsu_hni_region_reg	RW	dsu_hni_region_reg
{0-7} 0x1600 : 0x1638	sam_generic_regs0-7	RW	sam_generic_regs0-7

### 8.3.17.1 por\_rnsam\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

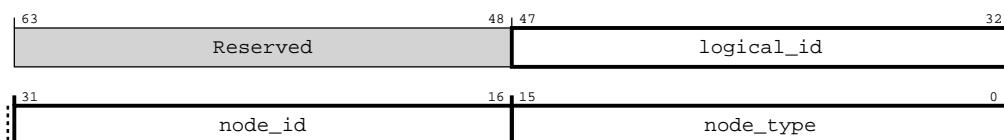
##### Usage constraints

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 8-952: por\_rnsam\_node\_info**



**Table 8-966: por\_rnsam\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID  <b>NOTE</b> RN SAM logical ID is always set to 0b0.	RO	0x0
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x000F

### 8.3.17.2 por\_rnsam\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

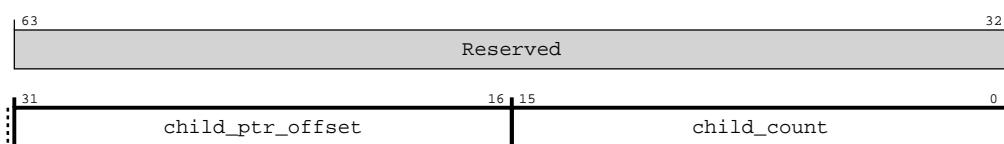
##### Usage constraints

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 8-953: por\_rnsam\_child\_info**



**Table 8-967: por\_rnsam\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.17.3 por\_rnsam\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x980

### Type

RW

### Reset value

See individual bit resets

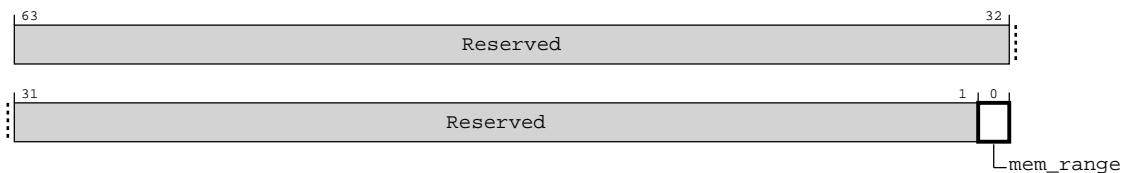
### Usage constraints

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 8-954: por\_rnsam\_scr**



**Table 8-968: por\_rnsam\_scr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	mem_range	Allows Secure override of the memory range registers	RW	0b0

## 8.3.17.4 por\_rnsam\_rcr

Root register access override.

### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x988

### Type

RW

### Reset value

See individual bit resets

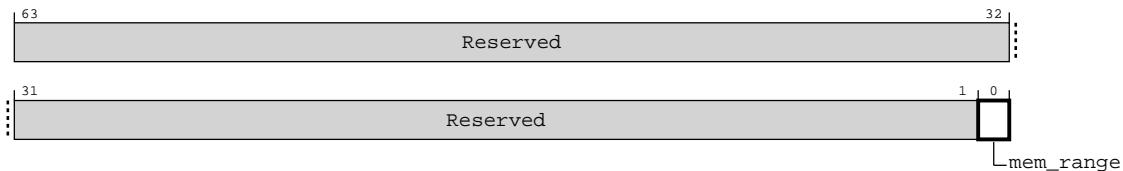
### Usage constraints

This register is owned in the Root space and is 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 8-955: por\_rnsam\_rcr**



**Table 8-969: por\_rnsam\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	mem_range	Allows Root override of the memory range registers	RW	0b0

### 8.3.17.5 por\_rnsam\_unit\_info

Provides component identification information for RN SAM.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0x900

#### Type

RO

## Reset value

See individual bit resets

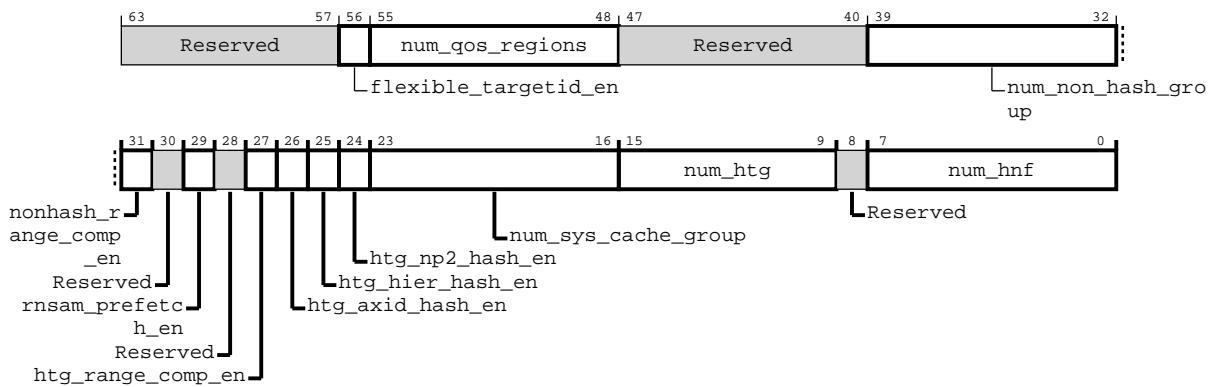
## Usage constraints

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 8-956: por\_rnsam\_unit\_info**



**Table 8-970: por\_rnsam\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56]	flexible_targetid_en	flexible target enable to preserve backward compatibility	RO	0x1
[55:48]	num_qos_regions	Number of QOS regions	RO	0x10
[47:40]	Reserved	Reserved	RO	-
[39:32]	num_non_hash_group	Number of non-hashed groups supported	RO	Configuration dependent
[31]	nonhash_range_comp_en	Define start and end address for each HTG region	RO	Configuration dependent
[30]	Reserved	Reserved	RO	-
[29]	rnsam_prefetch_h_en	RNSAM prefetch enabled	RO	0x1
[28]	Reserved	Reserved	RO	-
[27]	htg_range_comp_en	Define start and end address for each HTG region	RO	Configuration dependent
[26]	htg_axid_hash_en	Enable AXID based hashing scheme	RO	0x1
[25]	htg_hier_hash_en	Enable Hierarchical hashing scheme	RO	Configuration dependent
[24]	htg_np2_hash_en	Enable non-power of two hash scheme	RO	0x1
[23:16]	num_sys_cache_group	Number of system cache groups supported	RO	0x08
[15:9]	num_htg	Number of Hashed target groups	RO	Configuration dependent
[8]	Reserved	Reserved	RO	-
[7:0]	num_hnf	Number of hashed targets supported	RO	Configuration dependent

### 8.3.17.6 por\_rnsam\_unit\_info1

Provides component identification information for RN SAM.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x908

##### Type

RO

##### Reset value

See individual bit resets

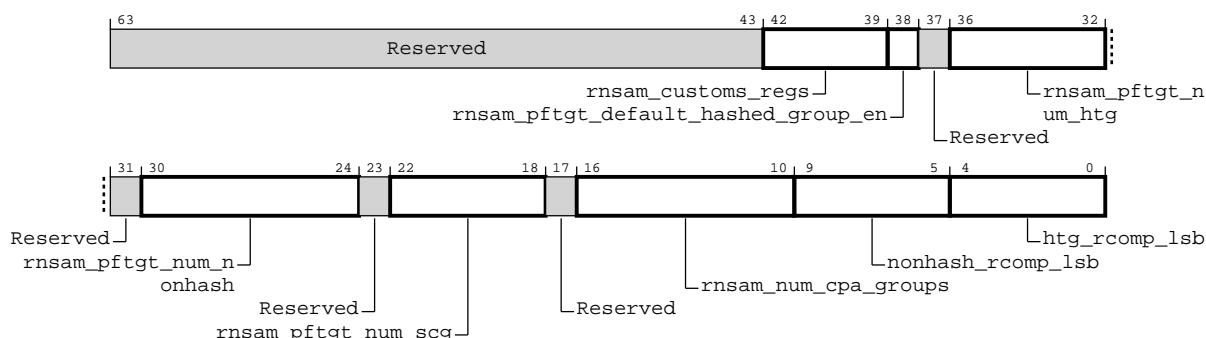
##### Usage constraints

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 8-957: por\_rnsam\_unit\_info1**



**Table 8-971: por\_rnsam\_unit\_info1 attributes**

Bits	Name	Description	Type	Reset
[63:43]	Reserved	Reserved	RO	-
[42:39]	rnsam_customs_regs	Number of customer specific registers for customer implemented logic	RO	Configuration dependent
[38]	rnsam_pftgt_default_hashed_group_en	Enable default hashed group for prefetch transactions. To support backward compatible, set this parameter	RO	0x1

Bits	Name	Description	Type	Reset
[37]	Reserved	Reserved	RO	-
[36:32]	rnsam_pftgt_num_htg	Number of prefetch HTG regions supported per System Cache Group by the RNSAM	RO	Configuration dependent
[31]	Reserved	Reserved	RO	-
[30:24]	rnsam_pftgt_num_nonhash	Number of prefetch non-hashed regions supported per System Cache Group by the RNSAM	RO	Configuration dependent
[23]	Reserved	Reserved	RO	-
[22:18]	rnsam_pftgt_num_scg	Number of system cache groups enabled for prefetch targets	RO	Configuration dependent
[17]	Reserved	Reserved	RO	-
[16:10]	rnsam_num_cpa_groups	Number of CPA groups	RO	Configuration dependent
[9:5]	nonhash_rcomp_lsb	NONHASH RCOMP LSB bit position defining minimum region size	RO	Configuration dependent
[4:0]	htg_rcomp_lsb	HTG RCOMP LSB bit position defining minimum region size	RO	Configuration dependent

### 8.3.17.7 non\_hash\_mem\_region\_reg0-127

There are 128 iterations of this register. The index ranges from 0 to 127. Configures non-hashed memory regions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-23) : 0xC00 + #{8 \* index}

##### index(24-151)

0x2000 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

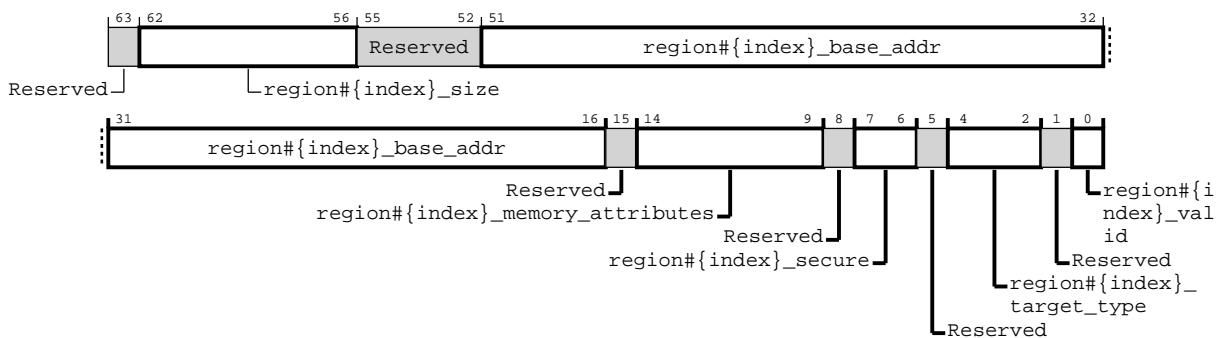
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-958: non\_hash\_mem\_region\_reg0-127**



**Table 8-972: non\_hash\_mem\_region\_reg0-127 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	region#{index}_size	Memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0x0
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_NONHASH_RCOMP_LSB_PARAM	RW	0x0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:9]	region#{index}_memory_attributes	<p>Indicates Memory Attributes, [3:0] directly translates to CHI memory attributes</p> <p>[5] snoop attr</p> <p>[4] 1 - persistent device, 0 - volatile device</p> <p>[3] Allocate</p> <p>[2] Cacheable</p> <p>[1] Device or Normal</p> <p>[0] EWA</p>	RW	0b00000
[8]	Reserved	Reserved	RO	-
[7:6]	region#{index}_secure	<p>Indicates secure type</p> <p><b>0b00</b> Trusted device.</p> <p><b>0b01</b> Trusted device attached memory range only.</p> <p><b>0b10</b> Untrusted device</p> <p><b>0b11</b> Reserved</p>	RW	0b00
[5]	Reserved	Reserved	RO	-
[4:2]	region#{index}_target_type	<p>Indicates node type</p> <p><b>0b000</b> HN-F</p> <p><b>0b001</b> HN-I</p> <p><b>0b010</b> CXRA</p> <p><b>0b011</b> HN-P</p> <p><b>0b100</b> PCI-CXRA</p> <p><b>0b101</b> HN-S</p> <p><b>Others</b> Reserved</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b000

Bits	Name	Description	Type	Reset
[1]	Reserved	Reserved	RO	-
[0]	region#{index}_valid	Memory region #{index} valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.17.8 non\_hash\_mem\_region\_cfg2\_reg0-127

There are 128 iterations of this register. The index ranges from 0 to 127. Configures non-hashed memory region end address

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

###### **index(0-23)**

0xCC0 + #{8 \* index}

###### **index(24-151)**

0x2400 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

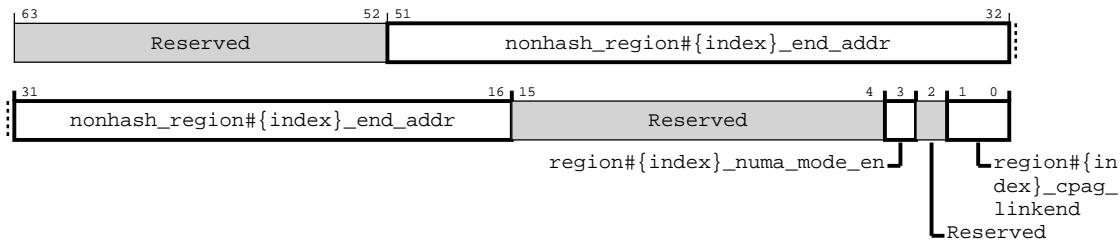
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-959: non\_hash\_mem\_region\_cfg2\_reg0-127**



**Table 8-973: non\_hash\_mem\_region\_cfg2\_reg0-127 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	nonhash_region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_NONHASH_RCOMP_LSB_PARAM	RW	0x0
[15:4]	Reserved	Reserved	RO	
[3]	region#{index}_numa_mode_en	Configure this bit if the non-hashed region is programmed for the remote-chip NUMA	RW	0b0
[2]	Reserved	Reserved	RO	
[1:0]	region#{index}_cpag_linkend	Specifies CPAG Linkend	RW	0x0

### 8.3.17.9 non\_hash\_tgt\_nodeid0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures non-hashed target node IDs #{4index} to #{4index + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15)

0xD80 + #{8 \* index}

##### index(16-47)

0x2800 + #{8 \* index}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

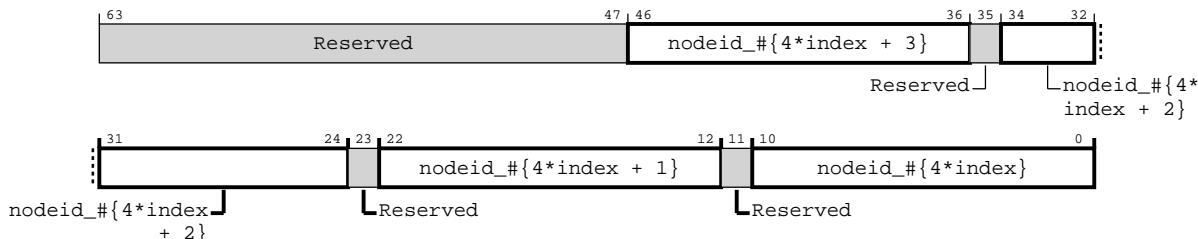
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-960: non\_hash\_tgt\_nodeid0-31**



**Table 8-974: non\_hash\_tgt\_nodeid0-31 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{4*index + 3}	Non-hashed target node ID #{4*index + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{4*index + 2}	Non-hashed target node ID #{4*index + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{4*index + 1}	Non-hashed target node ID #{4*index + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{4*index}	Non-hashed target node ID #{4*index}	RW	0b000000000000

### 8.3.17.10 cml\_port\_aggr\_mode\_ctrl\_reg

Configures the CCIX port aggregation modes for all non-hashed memory regions.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x11A0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

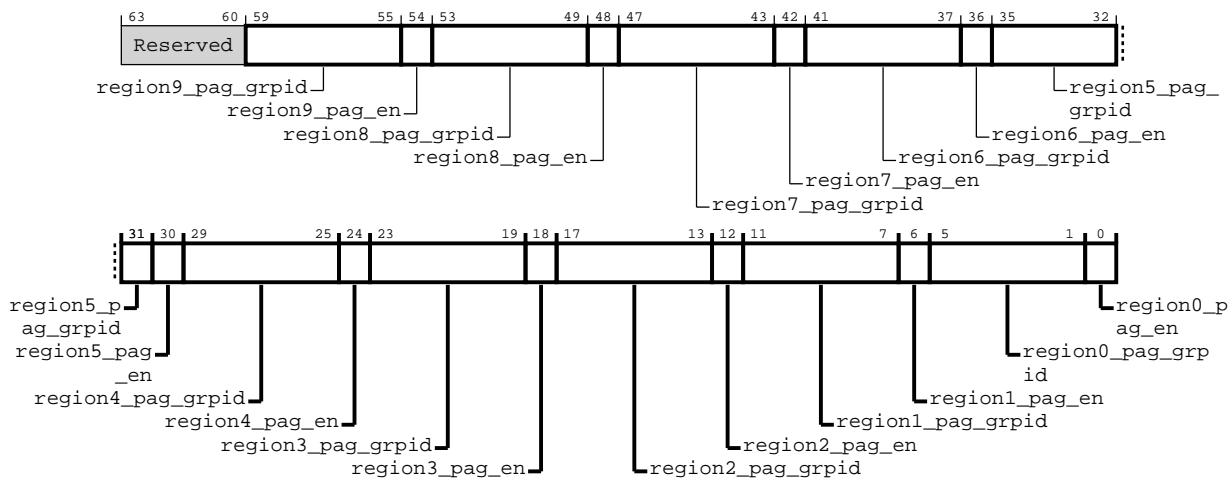
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-961: cml\_port\_aggr\_mode\_ctrl\_reg**



**Table 8-975: cml\_port\_aggr\_mode\_ctrl\_reg attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	
[59:55]	region9_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[54]	region9_pag_en	Enables the CPA mode for non-hashed memory region 9	RW	0b0
[53:49]	region8_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[48]	region8_pag_en	Enables the CPA mode for non-hashed memory region 8	RW	0b0
[47:43]	region7_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[42]	region7_pag_en	Enables the CPA mode for non-hashed memory region 7	RW	0b0
[41:37]	region6_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[36]	region6_pag_en	Enables the CPA mode for non-hashed memory region 6	RW	0b0
[35:31]	region5_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[30]	region5_pag_en	Enables the CPA mode for non-hashed memory region 5	RW	0b0
[29:25]	region4_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[24]	region4_pag_en	Enables the CPA mode for non-hashed memory region 4	RW	0b0
[23:19]	region3_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[18]	region3_pag_en	Enables the CPA mode for non-hashed memory region 3	RW	0b0
[17:13]	region2_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[12]	region2_pag_en	Enables the CPA mode for non-hashed memory region 2	RW	0b0
[11:7]	region1_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[6]	region1_pag_en	Enables the CPA mode for non-hashed memory region 1	RW	0b0
[5:1]	region0_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[0]	region0_pag_en	Enables the CPA mode for non-hashed memory region 0	RW	0b0

### 8.3.17.11 cml\_port\_aggr\_mode\_ctrl\_reg1-12

There are 12 iterations of this register. The index ranges from 1 to 12. Configures the CCIX port aggregation modes for all non-hashed memory regions.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-3) : 0x11A0 + #{8 \* index}

##### index(4-19)

0x2A00 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

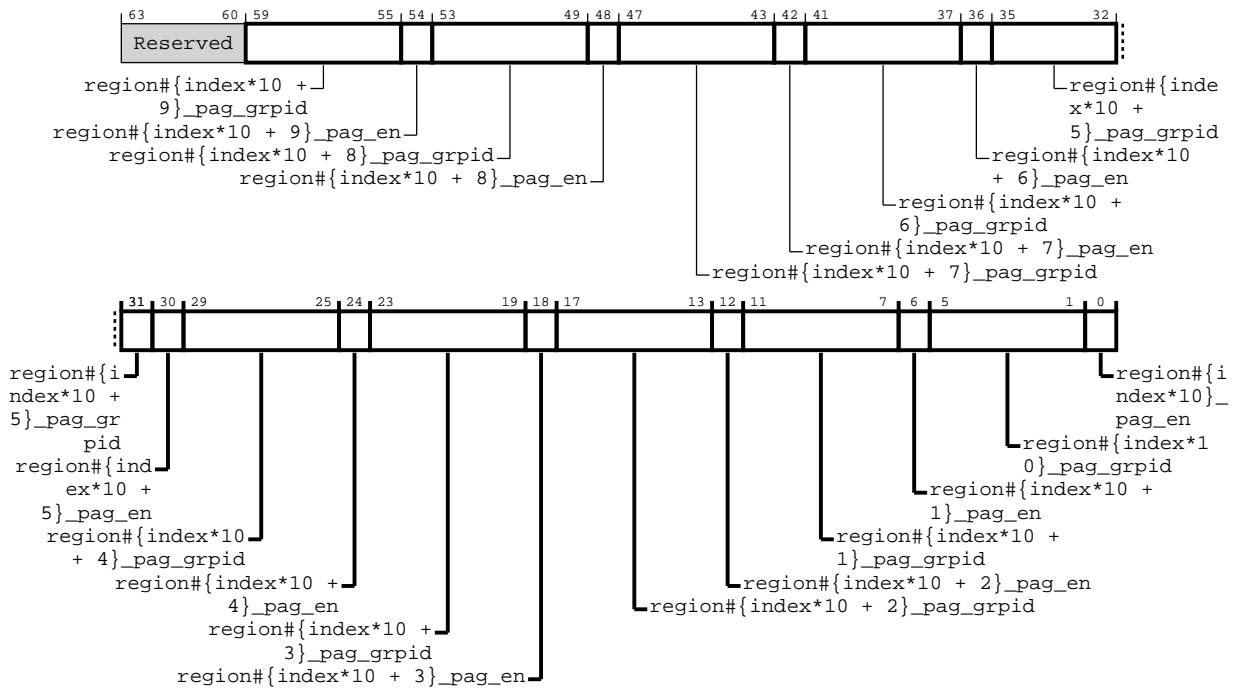
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-962: cml\_port\_aggr\_mode\_ctrl\_reg1-12**



**Table 8-976: cml\_port\_aggr\_mode\_ctrl\_reg1-12 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	
[59:55]	region#{index*10 + 9}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[54]	region#{index*10 + 9}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 9}	RW	0b0
[53:49]	region#{index*10 + 8}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[48]	region#{index*10 + 8}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 8}	RW	0b0
[47:43]	region#{index*10 + 7}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[42]	region#{index*10 + 7}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 7}	RW	0b0
[41:37]	region#{index*10 + 6}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[36]	region#{index*10 + 6}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 6}	RW	0b0
[35:31]	region#{index*10 + 5}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[30]	region#{index*10 + 5}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 5}	RW	0b0
[29:25]	region#{index*10 + 4}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[24]	region#{index*10 + 4}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 4}	RW	0b0
[23:19]	region#{index*10 + 3}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[18]	region#{index*10 + 3}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 3}	RW	0b0
[17:13]	region#{index*10 + 2}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[12]	region#{index*10 + 2}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 2}	RW	0b0
[11:7]	region#{index*10 + 1}_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[6]	region#{index*10 + 1}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 1}	RW	0b0

Bits	Name	Description	Type	Reset
[5:1]	region#[index*10]_pag_grpid	Specifies CCIX port aggregation group ID	RW	0x0
[0]	region#[index*10]_pag_en	Enables the CPA mode for non-hashed memory region #{index*10}	RW	0b0

### 8.3.17.12 sys\_cache\_grp\_region0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures hashed memory regions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE00 + #{8\*index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

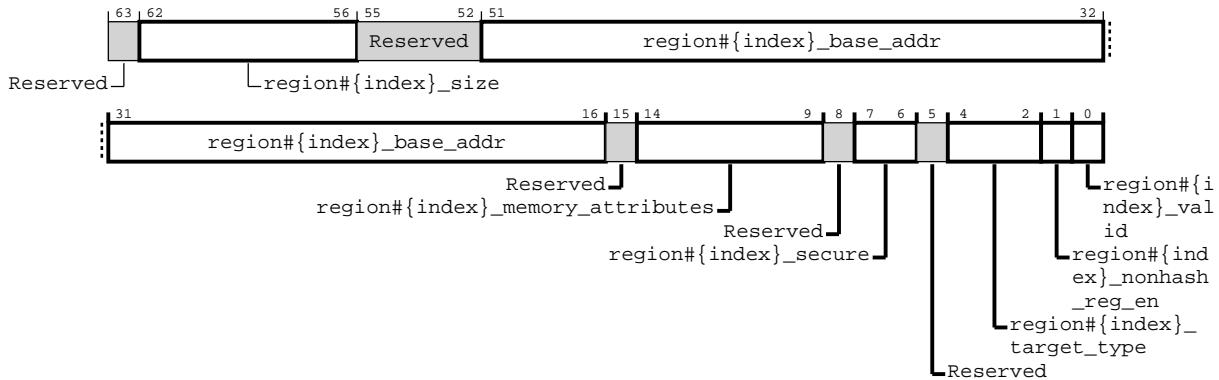
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-963: sys\_cache\_grp\_region0-3**



**Table 8-977: sys\_cache\_grp\_region0-3 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	region#{index}_size	Memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000000
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAMHTGRCOMP_LSB_PARAM	RW	0b00000000000000000000000000000000
[15]	Reserved	Reserved	RO	-
[14:9]	region#{index}_memory_attributes	Indicates Memory Attributes, [3:0] directly translates to CHI memory attributes  [4] 1 - persistent device, 0 - volatile device [3] Allocate [2] Cacheable [1] Device or Normal [0] EWA	RW	0b00000
[8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7:6]	region#{index}_secure	<p>Indicates secure type</p> <p><b>0b00</b> Trusted device.</p> <p><b>0b01</b> Trusted device attached memory range only.</p> <p><b>0b10</b> Untrusted device</p> <p><b>0b11</b> Reserved</p>	RW	0b00
[5]	Reserved	Reserved	RO	-
[4:2]	region#{index}_target_type	<p>Indicates node type</p> <p><b>0b000</b> HN-F</p> <p><b>0b001</b> HN-I</p> <p><b>0b010</b> CXRA</p> <p><b>0b011</b> HN-P</p> <p><b>0b100</b> PCI-CXRA</p> <p><b>0b101</b> HN-S</p> <p><b>Others</b> Reserved</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b000
[1]	region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	0b0
[0]	region#{index}_valid	<p>Memory region #{index} valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.17.13 hashed\_tgt\_grp\_cfg1\_region4-31

There are 28 iterations of this register. The index ranges from 4 to 31. Configures hashed memory regions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-7) : 0xE00 + #{8 \* index}

##### index(8-31)

0x3000 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

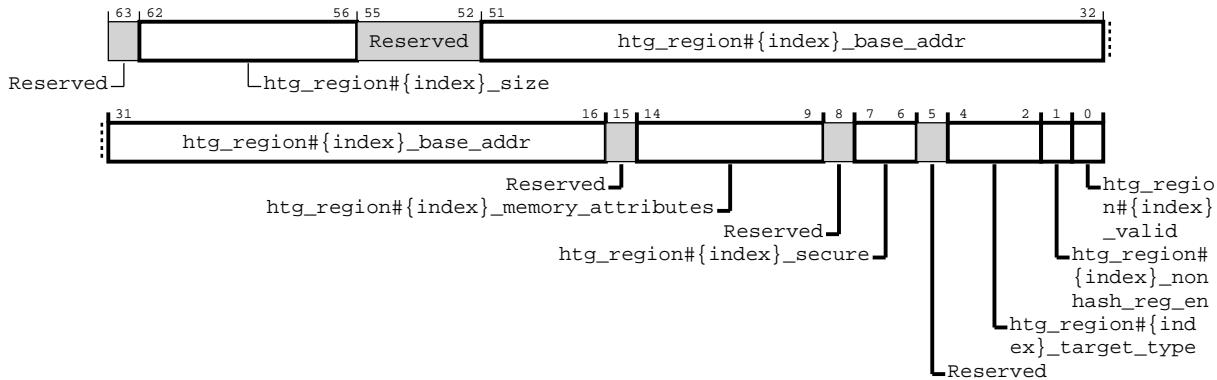
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-964: hashed\_tgt\_grp\_cfg1\_region4-31**



**Table 8-978: hashed\_tgt\_grp\_cfg1\_region4-31 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_region#{index}_size	Memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b0000000
[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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15]	Reserved	Reserved	RO	-
[14:9]	htg_region#{index}_memory_attributes	Indicates Memory Attributes, [3:0] directly translates to CHI memory attributes [4] 1 - persistent device, 0 - volatile device [3] Allocate [2] Cacheable [1] Device or Normal [0] EWA	RW	0b00000
[8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7:6]	htg_region#{index}_secure	<p>Indicates secure type</p> <p><b>0b00</b> Trusted device.</p> <p><b>0b01</b> Trusted device attached memory range only.</p> <p><b>0b10</b> Untrusted device</p> <p><b>0b11</b> Reserved</p>	RW	0b00
[5]	Reserved	Reserved	RO	-
[4:2]	htg_region#{index}_target_type	<p>Indicates node type</p> <p><b>0b000</b> HN-F</p> <p><b>0b001</b> HN-I</p> <p><b>0b010</b> CXRA</p> <p><b>0b011</b> HN-P</p> <p><b>0b100</b> PCI-CXRA</p> <p><b>0b101</b> HN-S</p> <p><b>Others</b> Reserved</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b000
[1]	htg_region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	0b0
[0]	htg_region#{index}_valid	<p>Memory region #{index} valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.17.14 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

$\text{index}(0\text{-}31) : 0x3100 + \#\{8 * \text{index}\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

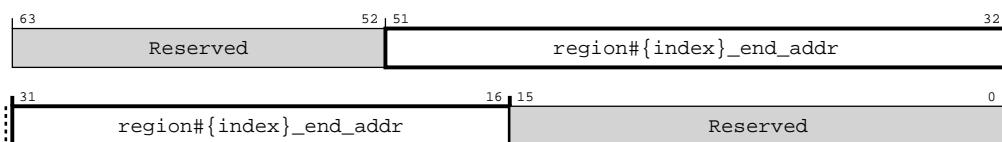
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-965: hashed\_tgt\_grp\_cfg2\_region0-31**



**Table 8-979: hashed\_tgt\_grp\_cfg2\_region0-31 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	0x0
[15:0]	Reserved	Reserved	RO	

### 8.3.17.15 sys\_cache\_grp\_secondary\_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures secondary hashed memory regions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7)  
0xE40 + #{8 \* index}

##### index(8-31)

0x3200 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

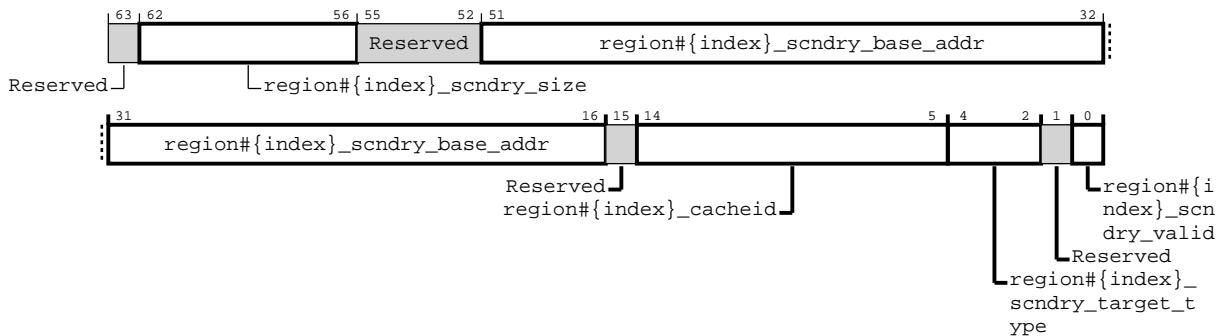
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-966: sys\_cache\_grp\_secondary\_reg0-3**



**Table 8-980: sys\_cache\_grp\_secondary\_reg0-3 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	region#{index}_scndry_size	Secondary memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_scndry_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15]	Reserved	Reserved	RO	-
[14:5]	region#{index}_cacheid	Specifies the coherency domain associated to the HTG#{index}	RW	0x0
[4:2]	region#{index}_scndry_target_type	Indicates node type <b>0b000</b> HN-F <b>0b001</b> HN-I <b>0b010</b> CXRA <b>0b011</b> HN-P <b>0b100</b> PCI-CXRA <b>0b101</b> HN-S <b>Others</b> Reserved <b>CONSTRAINT</b> Only applicable for RN-I	RW	0b000
[1]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[0]	region#{index}_scndry_valid	<p>Secondary memory region #{index} valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.17.16 hashed\_target\_grp\_secondary\_cfg1\_reg4-31

There are 28 iterations of this register. The index ranges from 4 to 31. Configures secondary hashed memory regions

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-7) : 0xE40 + #{8 \* index}

##### index(8-31)

0x3200 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

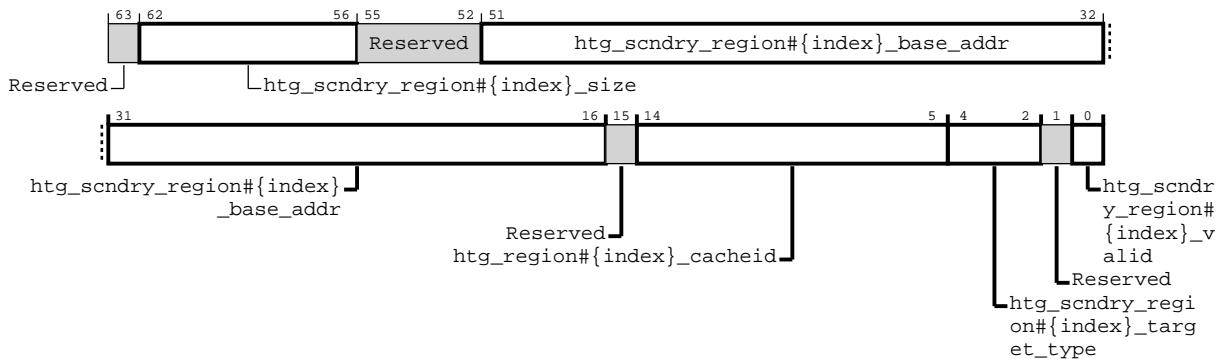
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-967: hashed\_target\_grp\_secondary\_cfg1\_reg4-31**



**Table 8-981: hashed\_target\_grp\_secondary\_cfg1\_reg4-31 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_scndry_region#{index}_size	Secondary memory region #{index} size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[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_RNSAMHTG_RCOMP_LSB_PARAM	RW	0x0
[15]	Reserved	Reserved	RO	-
[14:5]	htg_region#{index}_cacheid	Specifies the coherency domain associated to the HTG#{index}	RW	0x0
[4:2]	htg_scndry_region#{index}_target_type	Indicates node type <b>0b000</b> HN-F <b>0b001</b> HN-I <b>0b010</b> CXRA <b>0b011</b> HN-P <b>0b100</b> PCI-CXRA <b>0b101</b> HN-S <b>Others</b> Reserved <b>CONSTRAINT</b> Only applicable for RN-I	RW	0b000
[1]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[0]	htg_scndry_region#[index]_valid	<p>Secondary memory region #[index] valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.17.17 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3300 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

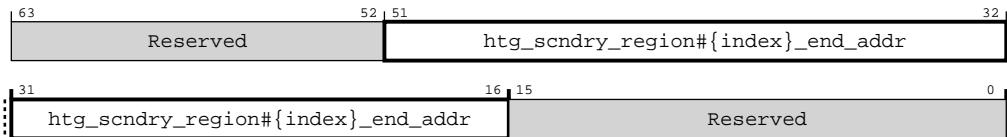
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-968: hashed\_target\_grp\_secondary\_cfg2\_reg0-31**



**Table 8-982: hashed\_target\_grp\_secondary\_cfg2\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAMHTG_RCOMP_LSB_PARAM	RW	0b00000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

### 8.3.17.18 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3400 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

##### Usage constraints

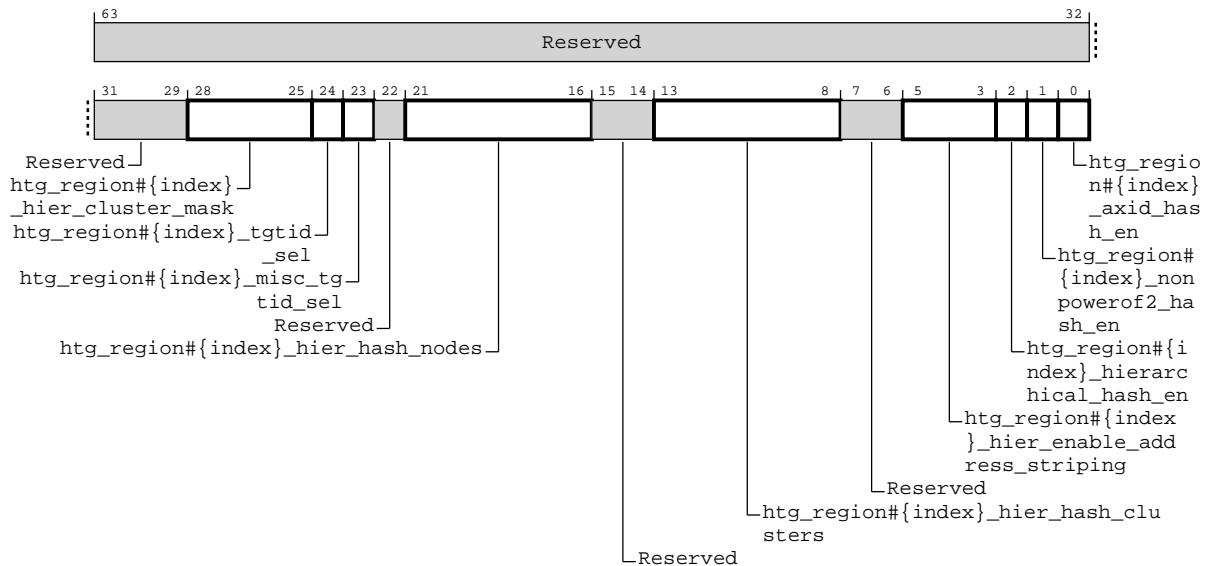
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range bit 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 8-969: hashed\_target\_grp\_hash\_cntl\_reg0-31**



**Table 8-983: 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	<b>Hierarchical hashing</b> Enable cluster masking to achieve different interleave granularity across clusters. <b>0b0000</b> 64 byte interleave granularity across clusters <b>0b0110</b> 4096 byte interleave granularity across clusters <b>0b1111</b> Cluster interleaving disabled <b>others</b> Reserved	RW	0b1111
[24]	htg_region#{index}_tgtid_sel	Select the TgtID's from HNF or HNP trgt tables <b>0b0</b> Default, selects from HNF table <b>0b1</b> selects from HNP table	RW	0b0

Bits	Name	Description	Type	Reset
[23]	htg_region#{index}_misc_tgtid_sel	Select the TgtID's from HNF or Misc tgt tables  <b>0b0</b> Default, selects from HNF table  <b>0b1</b> selects from Misc Tgt table	RW	0b0
[22]	Reserved	Reserved	RO	-
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	0x0
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	0x0
[7:6]	Reserved	Reserved	RO	-
[5:3]	htg_region#{index}_hier_enable_address_striping	Hierarchical  <b>hashing</b> configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask).  <b>0b000</b> no address shuttering <b>0b001</b> one addr bit shuttered (2 clusters) <b>0b010</b> two addr bit shuttered (4 clusters) <b>0b011</b> three addr bit shuttered (8 clusters) <b>0b100</b> four addr bit shuttered (16 clusters) <b>0b101</b> five addr bit shuttered (32 clusters) <b>others</b> Reserved	RW	0b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	0b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	0b0
[0]	htg_region#{index}_axid_hash_en	AXID based Hashing mode enable configure bit	RW	0b0

### 8.3.17.19 sys\_cache\_group\_hn\_count

Indicates number of HN-F/HN-P's in hashed target groups 0 to 7. To Enable 256-way with NO CAL, 512-way with CAL2, 1024-Way with CAL4, we need to program num\_hnf = 0xFF

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEA0

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

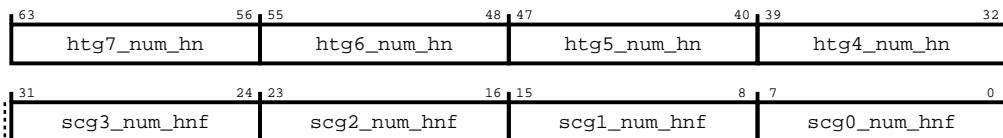
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-970: sys\_cache\_group\_hn\_count**



**Table 8-984: sys\_cache\_group\_hn\_count attributes**

Bits	Name	Description	Type	Reset
[63:56]	htg7_num_hn	HN count for hashed target group 7	RW	0x00
[55:48]	htg6_num_hn	HN count for hashed target group 6	RW	0x00
[47:40]	htg5_num_hn	HN count for hashed target group 5	RW	0x00
[39:32]	htg4_num_hn	HN count for hashed target group 4	RW	0x00
[31:24]	scg3_num_hnf	HN count for hashed target group 3	RW	0x00
[23:16]	scg2_num_hnf	HN count for hashed target group 2	RW	0x00
[15:8]	scg1_num_hnf	HN count for hashed target group 1	RW	0x00
[7:0]	scg0_num_hnf	HN count for hashed target group 0	RW	0x00

### 8.3.17.20 hashed\_target\_group\_hn\_count\_reg1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Indicates number of HN-F/HN-P's in hashed target groups # $\{index8\}$  to # $\{index8 + 7\}$ . To Enable 256-way with NO CAL, 512-way with CAL2, 1024-Way with CAL4, we need to program num\_hnf = 0xFF

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-1) : 0xEA0 + # $\{8 * index\}$

##### index(2-3)

0x3700 + # $\{8 * index\}$

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

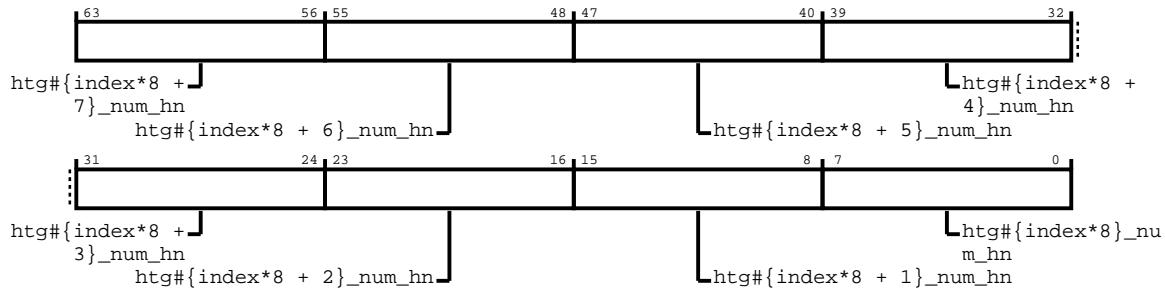
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-971: hashed\_target\_group\_hn\_count\_reg1-3**



**Table 8-985: hashed\_target\_group\_hn\_count\_reg1-3 attributes**

Bits	Name	Description	Type	Reset
[63:56]	htg#{index*8 + 7}_num_hn	HN count for hashed target group 7	RW	0x00
[55:48]	htg#{index*8 + 6}_num_hn	HN count for hashed target group 6	RW	0x00
[47:40]	htg#{index*8 + 5}_num_hn	HN count for hashed target group 5	RW	0x00
[39:32]	htg#{index*8 + 4}_num_hn	HN count for hashed target group 4	RW	0x00
[31:24]	htg#{index*8 + 3}_num_hn	HN count for hashed target group 3	RW	0x00
[23:16]	htg#{index*8 + 2}_num_hn	HN count for hashed target group 2	RW	0x00
[15:8]	htg#{index*8 + 1}_num_hn	HN count for hashed target group 1	RW	0x00
[7:0]	htg#{index*8}_num_hn	HN count for hashed target group 0	RW	0x00

### 8.3.17.21 sys\_cache\_grp\_nonhash\_nodeid

Configures non-hashed node IDs for hashed target groups 1 to 5. NOTE: Only applicable in the non-hashed mode.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEC0

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

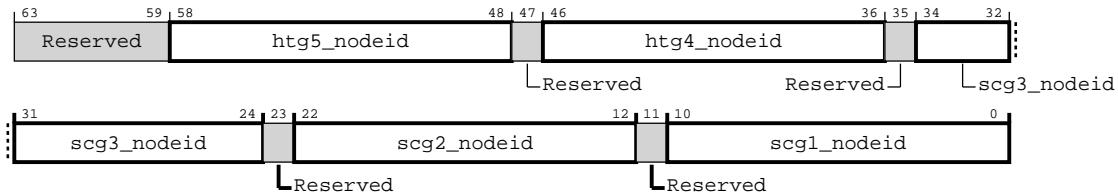
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-972: sys\_cache\_grp\_nonhash\_nodeid**



**Table 8-986: sys\_cache\_grp\_nonhash\_nodeid attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	htg5_nodeid	Non-hashed node ID for Hashed target group 5	RW	0b000000000000
[47]	Reserved	Reserved	RO	-
[46:36]	htg4_nodeid	Non-hashed node ID for Hashed target group 4	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	scg3_nodeid	Non-hashed node ID for Hashed target group 3	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	scg2_nodeid	Non-hashed node ID for Hashed target group 2	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	scg1_nodeid	Non-hashed node ID for Hashed target group 1	RW	0b000000000000

### 8.3.17.22 hashed\_target\_grp\_nonhash\_nodeid\_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures non-hashed node IDs for hashed target groups # $\{index5 + 1\}$  to # $\{index5 + 5\}$ . NOTE: Only applicable in the non-hashed mode.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-5) : 0xEC0 + # $\{8 * index\}$

##### index(6-9)

0x3800 + # $\{8 * (index-6)\}$

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

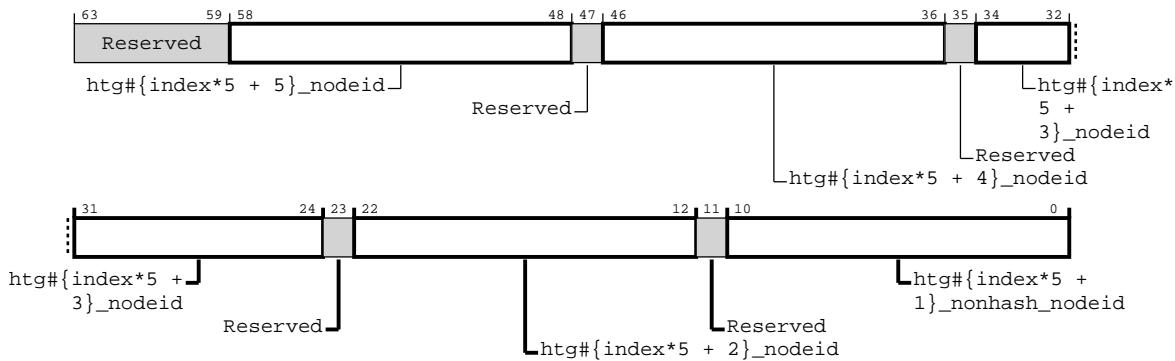
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-973: hashed\_target\_grp\_nonhash\_nodeid\_reg1-6**



**Table 8-987: hashed\_target\_grp\_nonhash\_nodeid\_reg1-6 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	htg#{index*5 + 5}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 5}	RW	0b00000000000000
[47]	Reserved	Reserved	RO	-
[46:36]	htg#{index*5 + 4}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 4}	RW	0b00000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	htg#{index*5 + 3}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 3}	RW	0b00000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	htg#{index*5 + 2}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 2}	RW	0b00000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	htg#{index*5 + 1}_nonhash_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 1}	RW	0b00000000000000

### 8.3.17.23 sys\_cache\_grp\_hn\_nodeid\_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HNF node IDs for hashed target groups. Controls target HNF node IDs #{index4} to #{index4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0xF00 + #8 \* index

##### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_rnsam_rcr.mem_range`

### Secure group override

`por_rnsam_scr.mem_range`

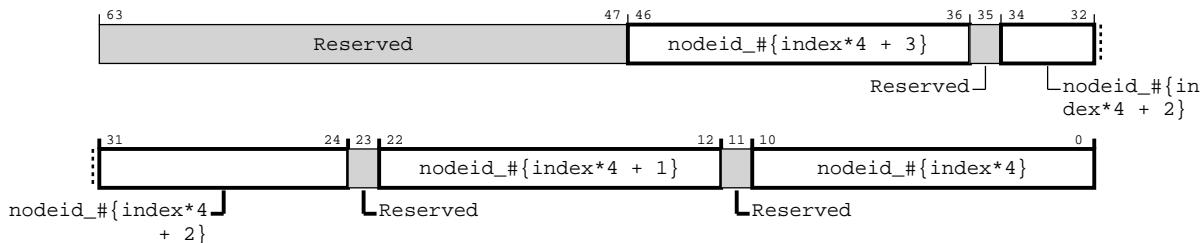
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_rnsam_scr.mem_range` bit is set, Secure accesses to this register are permitted. If both the `por_rnsam_scr.mem_range` bit and `por_rnsam_rcr.mem_range` 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 8-974: sys\_cache\_grp\_hn\_nodeid\_reg0-15**



**Table 8-988: sys\_cache\_grp\_hn\_nodeid\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{index*4 + 3}	HNF target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{index*4 + 2}	HNF target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{index*4 + 1}	HNF target node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	HNF target node ID #{index*4}	RW	0b000000000000

### 8.3.17.24 hashed\_target\_grp\_hnf\_nodeid\_reg16-31

There are 16 iterations of this register. The index ranges from 16 to 31. Configures HNF node IDs for hashed target groups. Controls target HNF node IDs #{{index4}} to #{{index4 + 3}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-31) : 0xF00 + #{8 \* index}

##### index(32-63)

0x3500 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

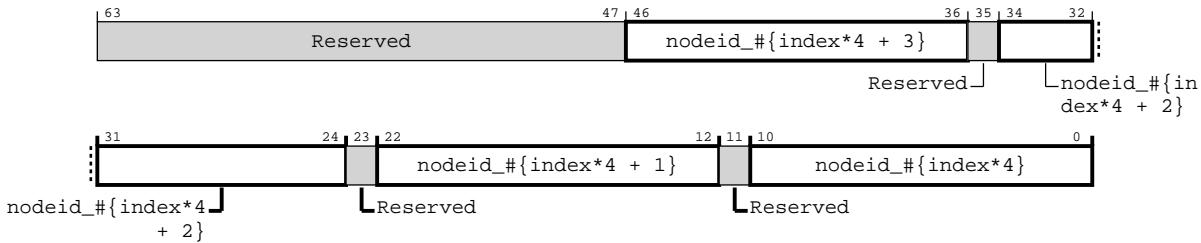
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-975: hashed\_target\_grp\_hnf\_nodeid\_reg16-31**



**Table 8-989: hashed\_target\_grp\_hnf\_nodeid\_reg16-31 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{index*4 + 3}	HNF target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{index*4 + 2}	HNF target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{index*4 + 1}	HNF target node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	HNF target node ID #{index*4}	RW	0b000000000000

### 8.3.17.25 hashed\_target\_grp\_hnp\_nodeid\_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HNP node IDs for hashed target groups. Controls target HNP node IDs #{index4} to #{index4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3600 + #8 \* index

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

`por_rnsam_scr.mem_range`

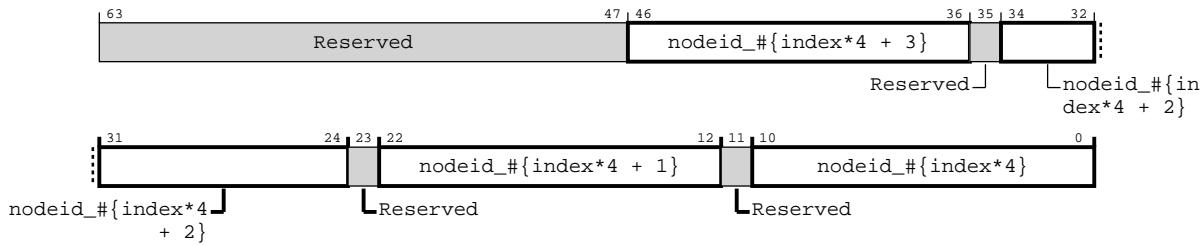
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_rnsam_scr.mem_range` bit is set, Secure accesses to this register are permitted. If both the `por_rnsam_scr.mem_range` bit and `por_rnsam_rcr.mem_range` 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 8-976: hashed\_target\_grp\_hnp\_nodeid\_reg0-15**



**Table 8-990: hashed\_target\_grp\_hnp\_nodeid\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_{index*4 + 3}	HNP target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_{index*4 + 2}	HNP target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_{index*4 + 1}	HNP target node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	HNP target node ID #{index*4}	RW	0b000000000000

## 8.3.17.26 hashed\_target\_grp\_misc\_nodeid\_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures Misc node IDs for hashed target groups. This register is used either for HNF/HNP/CCGs. Controls misc node IDs  $\#\{index4\}$  to  $\#\{index4 + 3\}$ .

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

$\text{index}(0\text{-}7) : 0x1900 + \#\{8 * \text{index}\}$

### Type

RW

### Reset value

See individual bit resets

### Root group override

`por_rnsam_rcr.mem_range`

### Secure group override

`por_rnsam_scr.mem_range`

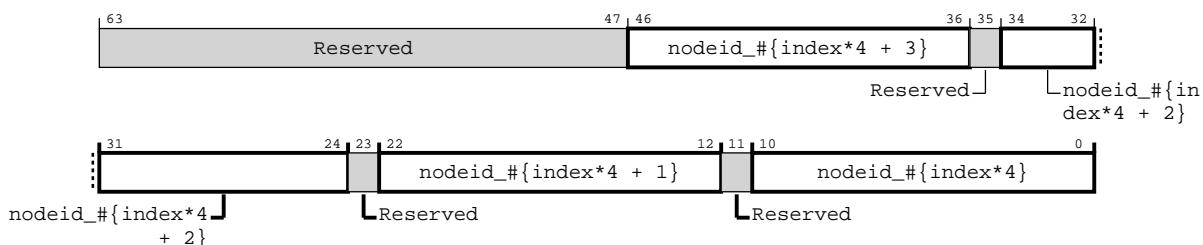
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the `por_rnsam_scr.mem_range` bit is set, Secure accesses to this register are permitted. If both the `por_rnsam_scr.mem_range` bit and `por_rnsam_rcr.mem_range` 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 8-977: hashed\_target\_grp\_misc\_nodeid\_reg0-7**



**Table 8-991: hashed\_target\_grp\_misc\_nodeid\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_#{index*4 + 3}	Misc target node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{index*4 + 2}	Misc target node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{index*4 + 1}	Misc target node ID #{index*4 + 1}	RW	0b000000000000

Bits	Name	Description	Type	Reset
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_{index*4}	Misc target node ID #{index*4}	RW	0b000000000000

### 8.3.17.27 hashed\_target\_grp\_misc\_tgts\_lcn\_bound\_cfg\_reg0

Configures cache lines routed to the HNF as LCN bound or Home bound

**0b0**

cache lines routed to Home bound

**0b1**

cache lines routed to LCN bound

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1980

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

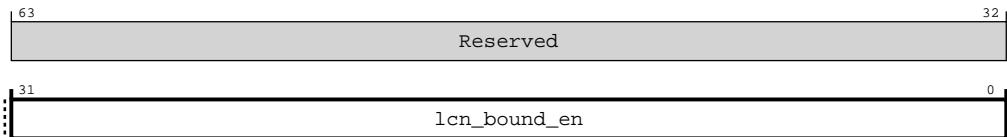
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-978: hashed\_target\_grp\_misc\_tgts\_lcn\_bound\_cfg\_reg0**



**Table 8-992: hashed\_target\_grp\_misc\_tgts\_lcn\_bound\_cfg\_reg0 attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	lcn_bound_en	Marks the Hashed HNF index as a LCN	RW	0x00000000

### 8.3.17.28 hashed\_tgt\_override\_cam\_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures hashed target override CAM structure register. Each entry contains a logical-id and physical-id of the override HN

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7) : 0x1800 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

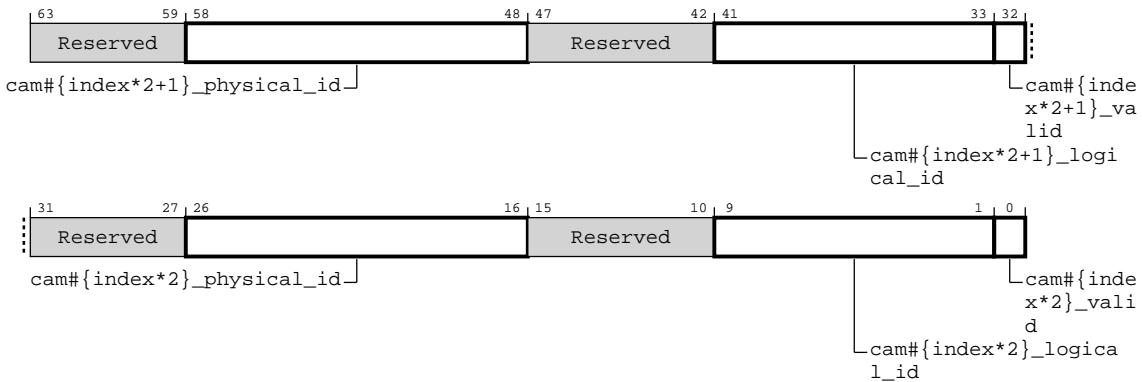
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-979: hashed\_tgt\_override\_cam\_reg0-7**



**Table 8-993: hashed\_tgt\_override\_cam\_reg0-7 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	cam#{index*2+1}_physical_id	CAM structure HN target-id #{index*4+1}	RW	0b000000000000
[47:42]	Reserved	Reserved	RO	-
[41:33]	cam#{index*2+1}_logical_id	CAM structure HN logical-id #{index*4+1}	RW	0b0000000000
[32]	cam#{index*2+1}_valid	CAM structure valid #{index*4+1}	RW	0b0
[31:27]	Reserved	Reserved	RO	-
[26:16]	cam#{index*2}_physical_id	CAM structure HN target-id #{index*4}	RW	0b000000000000
[15:10]	Reserved	Reserved	RO	-
[9:1]	cam#{index*2}_logical_id	CAM structure HN logical-id #{index*4}	RW	0b0000000000
[0]	cam#{index*2}_valid	CAM structure valid #{index*4}	RW	0b0

### 8.3.17.29 sys\_cache\_grp\_cal\_mode\_reg

Configures the HN CAL mode support for all hashed target groups.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1120

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

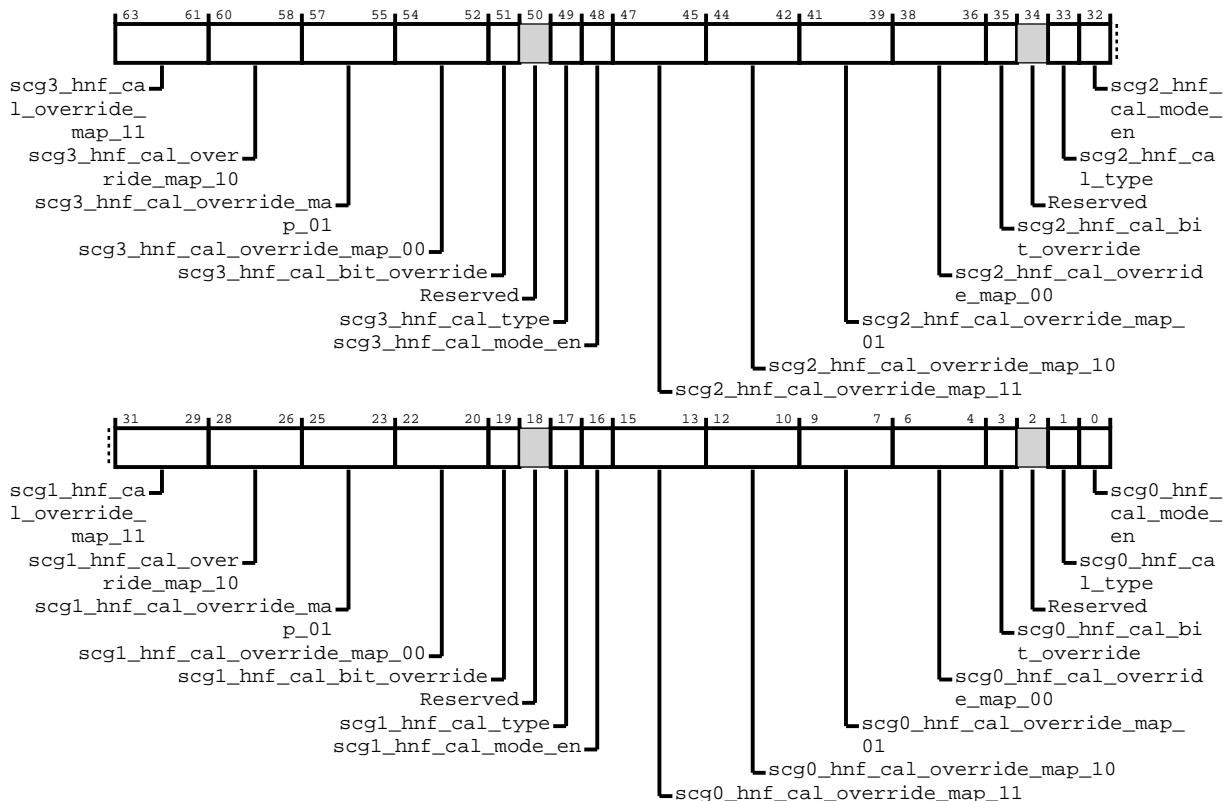
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-980: sys\_cache\_grp\_cal\_mode\_reg**



**Table 8-994: sys\_cache\_grp\_cal\_mode\_reg attributes**

Bits	Name	Description	Type	Reset
[63:61]	scg3_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[60:58]	scg3_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[57:55]	scg3_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[54:52]	scg3_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[51]	scg3_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 3  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[50]	Reserved	Reserved	RO	-
[49]	scg3_hnf_cal_type	Enables type of HN CAL for HTG 3  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[48]	scg3_hnf_cal_mode_en	Enables support for HN CAL for HTG 3	RW	0b0
[47:45]	scg2_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[44:42]	scg2_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[41:39]	scg2_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[38:36]	scg2_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[35]	scg2_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 2  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[34]	Reserved	Reserved	RO	-
[33]	scg2_hnf_cal_type	Enables type of HN CAL for HTG 2  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[32]	scg2_hnf_cal_mode_en	Enables support for HN CAL for HTG 2	RW	0b0
[31:29]	scg1_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[28:26]	scg1_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[25:23]	scg1_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[22:20]	scg1_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000

Bits	Name	Description	Type	Reset
[19]	scg1_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 1  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[18]	Reserved	Reserved	RO	-
[17]	scg1_hnf_cal_type	Enables type of HN CAL for HTG 1  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[16]	scg1_hnf_cal_mode_en	Enables support for HN CAL for HTG 1	RW	0b0
[15:13]	scg0_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[12:10]	scg0_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[9:7]	scg0_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[6:4]	scg0_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[3]	scg0_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 0  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[2]	Reserved	Reserved	RO	-
[1]	scg0_hnf_cal_type	Enables type of HN CAL for HTG 0  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[0]	scg0_hnf_cal_mode_en	Enables support for HN CAL for HTG 0	RW	0b0

### 8.3.17.30 hashed\_target\_grp\_cal\_mode\_reg1-7

There are 7 iterations of this register. The index ranges from 1 to 7. Configures the HN CAL mode support for all hashed target groups.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-3) : 0x1120 + #{8 \* index}

### **index(4-7)**

0x3780 + #{8 \* index}

#### **Type**

RW

#### **Reset value**

See individual bit resets

#### **Root group override**

por\_rnsam\_rcr.mem\_range

#### **Secure group override**

por\_rnsam\_scr.mem\_range

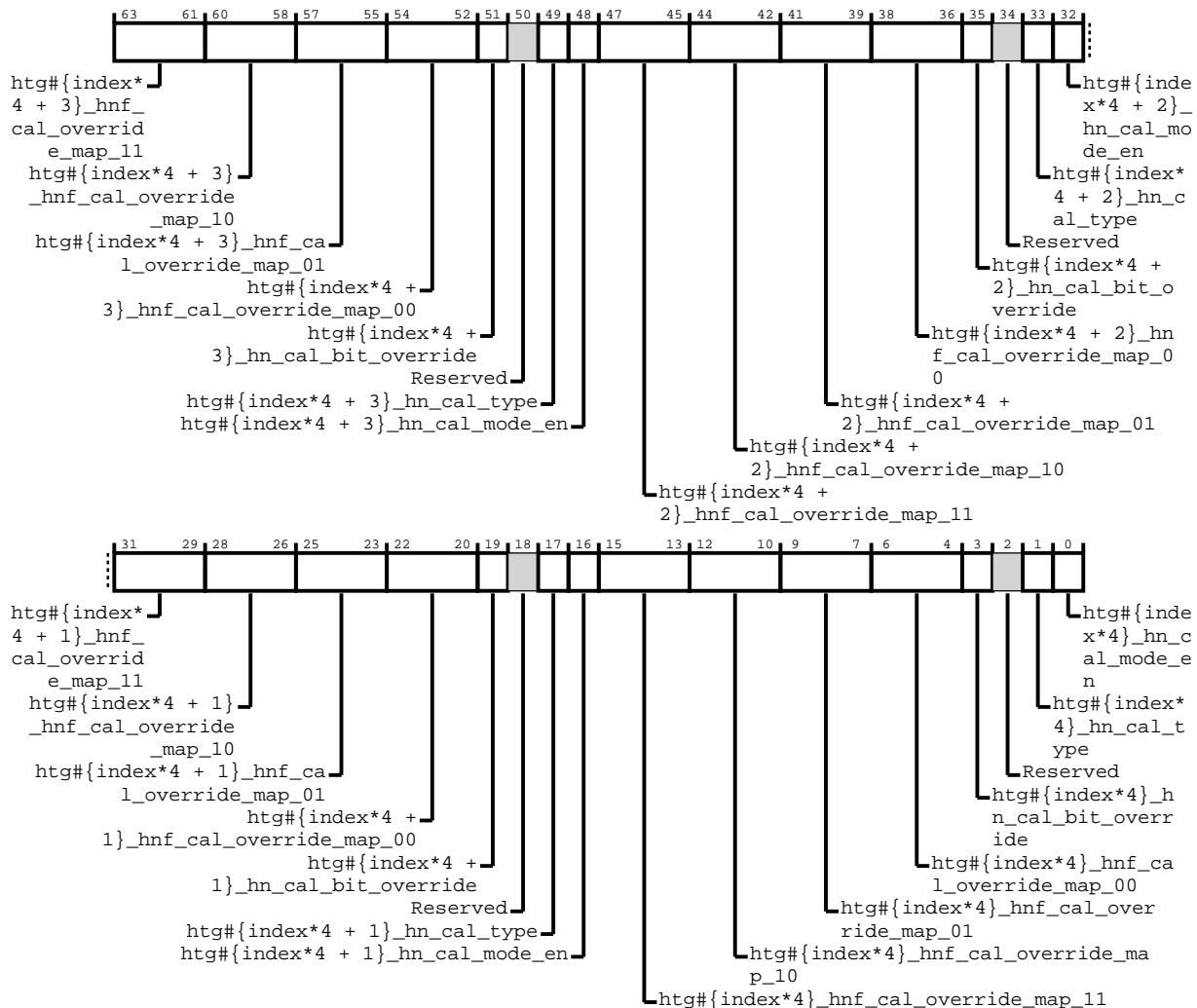
#### **Usage constraints**

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-981: hashed\_target\_grp\_cal\_mode\_reg1-7**



**Table 8-995: hashed\_target\_grp\_cal\_mode\_reg1-7 attributes**

Bits	Name	Description	Type	Reset
[63:61]	<code>htg#[index*4 + 3]_hnf_cal_override_map_11</code>	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[60:58]	<code>htg#[index*4 + 3]_hnf_cal_override_map_10</code>	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[57:55]	<code>htg#[index*4 + 3]_hnf_cal_override_map_01</code>	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[54:52]	<code>htg#[index*4 + 3]_hnf_cal_override_map_00</code>	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000

Bits	Name	Description	Type	Reset
[51]	htg#[index*4 + 3]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 3}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[50]	Reserved	Reserved	RO	-
[49]	htg#[index*4 + 3]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 3}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[48]	htg#[index*4 + 3]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 3}	RW	0b0
[47:45]	htg#[index*4 + 2]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[44:42]	htg#[index*4 + 2]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[41:39]	htg#[index*4 + 2]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[38:36]	htg#[index*4 + 2]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[35]	htg#[index*4 + 2]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 2}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[34]	Reserved	Reserved	RO	-
[33]	htg#[index*4 + 2]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 2}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[32]	htg#[index*4 + 2]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 2}	RW	0b0
[31:29]	htg#[index*4 + 1]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[28:26]	htg#[index*4 + 1]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[25:23]	htg#[index*4 + 1]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[22:20]	htg#[index*4 + 1]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000

Bits	Name	Description	Type	Reset
[19]	htg#[index*4 + 1]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4 + 1}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[18]	Reserved	Reserved	RO	-
[17]	htg#[index*4 + 1]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4 + 1}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[16]	htg#[index*4 + 1]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4 + 1}	RW	0b0
[15:13]	htg#[index*4]_hnf_cal_override_map_11	configuration to map the CAL override 0b11 to the custom 3bit value	RW	0b011
[12:10]	htg#[index*4]_hnf_cal_override_map_10	configuration to map the CAL override 0b10 to the custom 3bit value	RW	0b010
[9:7]	htg#[index*4]_hnf_cal_override_map_01	configuration to map the CAL override 0b01 to the custom 3bit value	RW	0b001
[6:4]	htg#[index*4]_hnf_cal_override_map_00	configuration to map the CAL override 0b00 to the custom 3bit value	RW	0b000
[3]	htg#[index*4]_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{{index}*4}  <b>0b0</b> Hash MSB bit to override Device ID  <b>0b1</b> Hash LSB bit to override Device ID	RW	0b0
[2]	Reserved	Reserved	RO	-
[1]	htg#[index*4]_hn_cal_type	Enables type of HN CAL for HTG #{{index}*4}  <b>0b0</b> CAL2 mode  <b>0b1</b> CAL4 mode	RW	0b0
[0]	htg#[index*4]_hn_cal_mode_en	Enables support for HN CAL for HTG #{{index}*4}	RW	0b0

### 8.3.17.31 sys\_cache\_grp\_hn\_cpa\_en\_reg

Configures CCIX port aggregation mode for hashed HNF node IDs

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

0x1180

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

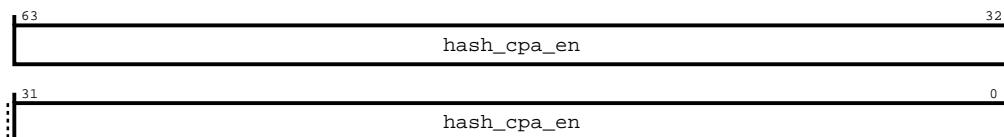
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-982: sys\_cache\_grp\_hn\_cpa\_en\_reg**



**Table 8-996: sys\_cache\_grp\_hn\_cpa\_en\_reg attributes**

Bits	Name	Description	Type	Reset
[63:0]	hash_cpa_en	Enable CPA for each hashed HNF node ID	RW	0x0000000000000000

### 8.3.17.32 hashed\_target\_grp\_hnf\_cpa\_en\_reg1-1

There are 1 iterations of this register. The index ranges from 1 to 1. Configures CCIX port aggregation mode for hashed HNF node IDs

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

Address offset : index(0-1) : 0x1180 + #{8 \* index}

### index(2-3)

0x3720 + #{8 \* index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

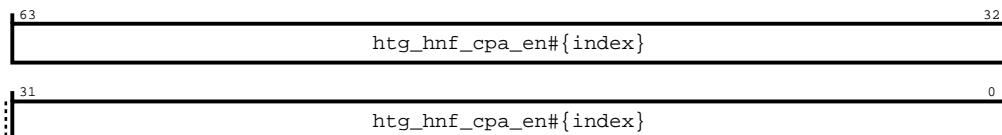
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-983: hashed\_target\_grp\_hnf\_cpa\_en\_reg1-1**



**Table 8-997: hashed\_target\_grp\_hnf\_cpa\_en\_reg1-1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	htg_hnf_cpa_en#{index}	Enable CPA for each hashed HNF node ID	RW	0x0000000000000000

### 8.3.17.33 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15) : 0x3900 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

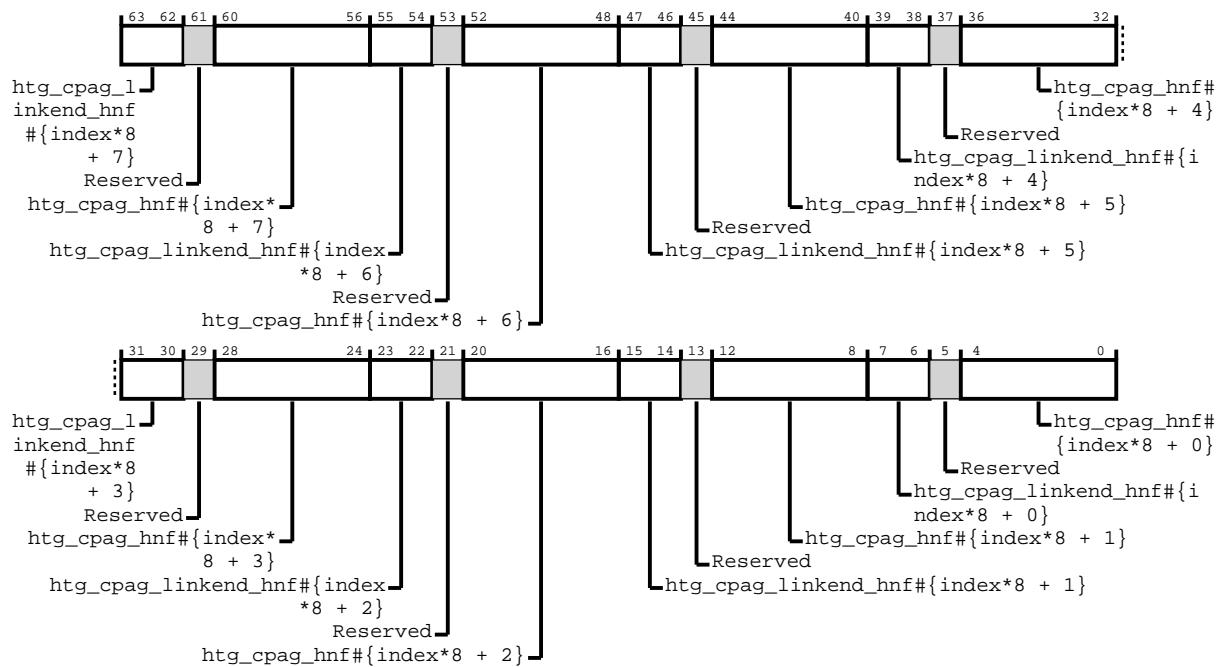
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-984: hashed\_target\_grp\_cpag\_perhnf\_reg0-15**



**Table 8-998: hashed\_target\_grp\_cpag\_perhnf\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63:62]	htg_cpag_linkend_hnf#[index*8 + 7]	CPAG Linkend associated to the HNF#[index*8 + 7]	RW	0b0
[61]	Reserved	Reserved	RO	
[60:56]	htg_cpag_hnf#[index*8 + 7]	CPAG associated to the HNF#[index*8 + 7]	RW	0b0
[55:54]	htg_cpag_linkend_hnf#[index*8 + 6]	CPAG Linkend associated to the HNF#[index*8 + 6]	RW	0b0
[53]	Reserved	Reserved	RO	
[52:48]	htg_cpag_hnf#[index*8 + 6]	CPAG associated to the HNF#[index*8 + 6]	RW	0b0
[47:46]	htg_cpag_linkend_hnf#[index*8 + 5]	CPAG Linkend associated to the HNF#[index*8 + 5]	RW	0b0
[45]	Reserved	Reserved	RO	
[44:40]	htg_cpag_hnf#[index*8 + 5]	CPAG associated to the HNF#[index*8 + 5]	RW	0b0
[39:38]	htg_cpag_linkend_hnf#[index*8 + 4]	CPAG Linkend associated to the HNF#[index*8 + 4]	RW	0b0
[37]	Reserved	Reserved	RO	
[36:32]	htg_cpag_hnf#[index*8 + 4]	CPAG associated to the HNF#[index*8 + 4]	RW	0b0
[31:30]	htg_cpag_linkend_hnf#[index*8 + 3]	CPAG Linkend associated to the HNF#[index*8 + 3]	RW	0b0
[29]	Reserved	Reserved	RO	
[28:24]	htg_cpag_hnf#[index*8 + 3]	CPAG associated to the HNF#[index*8 + 3]	RW	0b0
[23:22]	htg_cpag_linkend_hnf#[index*8 + 2]	CPAG Linkend associated to the HNF#[index*8 + 2]	RW	0b0
[21]	Reserved	Reserved	RO	
[20:16]	htg_cpag_hnf#[index*8 + 2]	CPAG associated to the HNF#[index*8 + 2]	RW	0b0
[15:14]	htg_cpag_linkend_hnf#[index*8 + 1]	CPAG Linkend associated to the HNF#[index*8 + 1]	RW	0b0
[13]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[12:8]	htg_cpag_hnf#[index*8 + 1]	CPAG associated to the HNF#[index*8 + 1]	RW	0b0
[7:6]	htg_cpag_linkend_hnf#[index*8 + 0]	CPAG Linkend associated to the HNF#[index*8 + 0]	RW	0b0
[5]	Reserved	Reserved	RO	
[4:0]	htg_cpag_hnf#[index*8 + 0]	CPAG associated to the HNF#[index*8 + 0]	RW	0b0

### 8.3.17.34 sys\_cache\_grp\_hn\_cpa\_grp\_reg

Configures CCIX port aggregation group ID for each System Cache Group

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1190

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

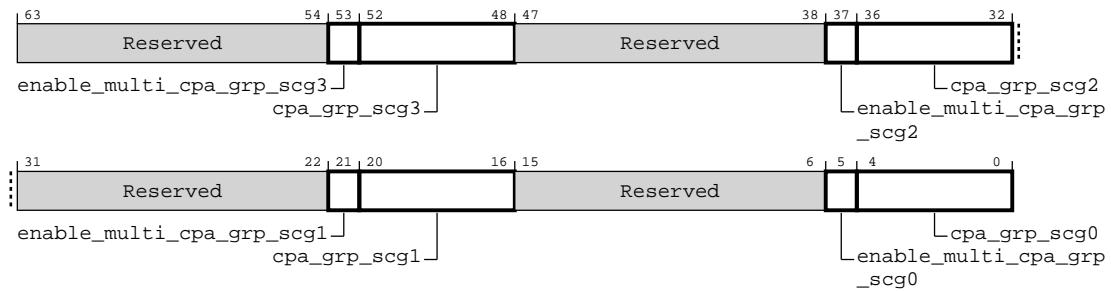
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-985: sys\_cache\_grp\_hn\_cpa\_grp\_reg**



**Table 8-999: sys\_cache\_grp\_hn\_cpa\_grp\_reg attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53]	enable_multi_cpa_grp_scg3	Enables multiple CPA groups to be configured to SCG3  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg  <b>0b1</b> multi_cpag mode : multiple CPA group ID's are configured to SCG3.	RW	0b0
[52:48]	cpa_grp_scg3	Specifies CCIX port aggregation group ID for Hashed Target Group 3	RW	0x0
[47:38]	Reserved	Reserved	RO	
[37]	enable_multi_cpa_grp_scg2	Enables multiple CPA groups to be configured to SCG2  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to SCG2.	RW	0b0
[36:32]	cpa_grp_scg2	Specifies CCIX port aggregation group ID for Hashed target Group 2	RW	0x0
[31:22]	Reserved	Reserved	RO	
[21]	enable_multi_cpa_grp_scg1	Enables multiple CPA groups to be configured to SCG1  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to SCG1.	RW	0b0
[20:16]	cpa_grp_scg1	Specifies CCIX port aggregation group ID for Hashed target Group 1	RW	0x0
[15:6]	Reserved	Reserved	RO	
[5]	enable_multi_cpa_grp_scg0	Enables multiple CPA groups to be configured to SCG0  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to SCG0.	RW	0b0
[4:0]	cpa_grp_scg0	Specifies CCIX port aggregation group ID for hashed target Group 0	RW	0x0

### 8.3.17.35 hashed\_target\_grp\_cpa\_grp\_reg1-7

There are 7 iterations of this register. The index ranges from 1 to 7. Configures CCIX port aggregation group ID for each System Cache Group

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-1) : 0x1190 + #{8 \* index}

##### index(2-9)

0x3740 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

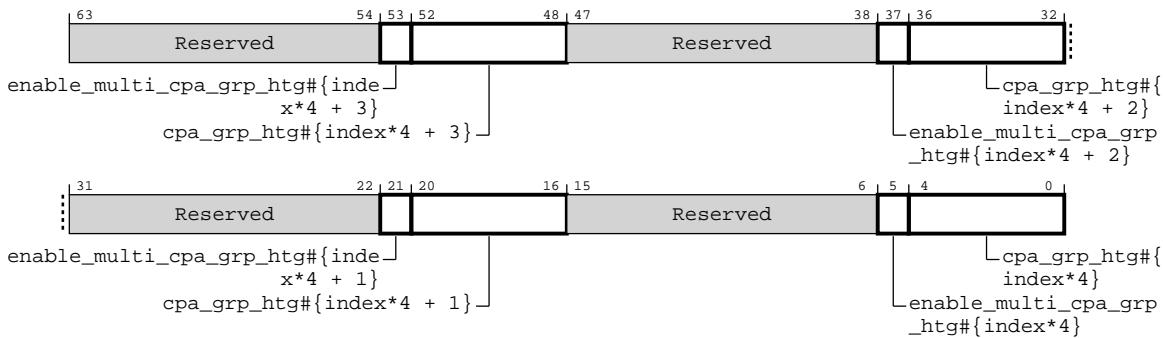
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-986: hashed\_target\_grp\_cpa\_grp\_reg1-7**



**Table 8-1000: hashed\_target\_grp\_cpa\_grp\_reg1-7 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53]	enable_multi_cpa_grp_htg#{index*4 + 3}	Enables multiple CPA groups to be configured to HTG  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	0b0
[52:48]	cpa_grp_htg#{index*4 + 3}	Specifies CCIX port aggregation group ID for Hashed Target Group #{index*4 + 3}	RW	0x0
[47:38]	Reserved	Reserved	RO	
[37]	enable_multi_cpa_grp_htg#{index*4 + 2}	Enables multiple CPA groups to be configured to HTG  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	0b0
[36:32]	cpa_grp_htg#{index*4 + 2}	Specifies CCIX port aggregation group ID for Hashed target Group #{index*4 + 2}	RW	0x0
[31:22]	Reserved	Reserved	RO	
[21]	enable_multi_cpa_grp_htg#{index*4 + 1}	Enables multiple CPA groups to be configured to HTG  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	0b0
[20:16]	cpa_grp_htg#{index*4 + 1}	Specifies CCIX port aggregation group ID for Hashed target Group #{index*4 + 1}	RW	0x0
[15:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5]	enable_multi_cpa_grp_htg#{index*4}	Enables multiple CPA groups to be configured to HTG  <b>0b0</b> Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg  <b>0b1</b> multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	0b0
[4:0]	cpa_grp_htg#{index*4}	Specifies CCIX port aggregation group ID for hashed target Group #{index*4}	RW	0x0

### 8.3.17.36 hashed\_target\_grp\_hnf\_lcn\_bound\_cfg\_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures cache lines routed to the HNF as LCN bound or Home bound 0b0: cache lines routed to Home bound 0b1: cache lines routed to LCN bound

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-3) : 0x37C0 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

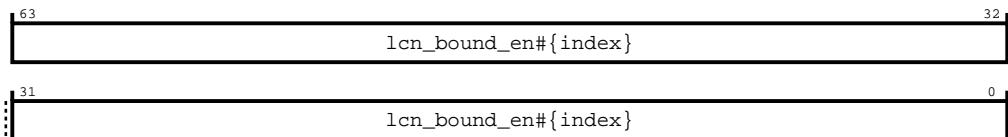
##### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-987: hashed\_target\_grp\_hnf\_lcn\_bound\_cfg\_reg0-1**



**Table 8-1001: hashed\_target\_grp\_hnf\_lcn\_bound\_cfg\_reg0-1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	lcn_bound_en#{index}	Marks the Hashed HNF index as a LCN	RW	0x0000000000000000

### 8.3.17.37 hashed\_target\_grp\_hnf\_target\_type\_override\_cfg\_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures the target type for each targetID, RNI/D uses this information to enable tunneling vs streaming. For POR\_RNSAM\_COMPACT\_HN\_TABLES\_EN\_PARAM == 1, this indication is derived from cpa\_en. This config register is used when CCG TgtIDs are directly loaded into HNF table without specifying CPA\_EN and CPAG. 0b0: HNF target 0b1: CCG target

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-1) : 0x37E0 + #{8 \* index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

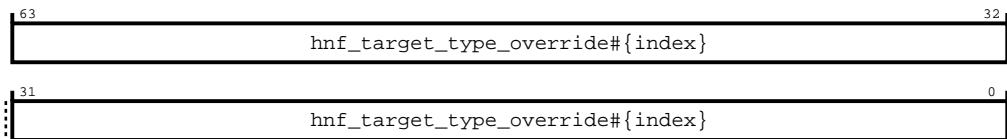
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-988: hashed\_target\_grp\_hnf\_target\_type\_override\_cfg\_reg0-1**



**Table 8-1002: hashed\_target\_grp\_hnf\_target\_type\_override\_cfg\_reg0-1 attributes**

Bits	Name	Description	Type	Reset
[63:0]	hnf_target_type_override#{index}	Overrides the HNF target type with the CCG	RW	0x0000000000000000

## 8.3.17.38 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

index(0-31) : 0x3A00 + #{8 \* index}

#### Type

RW

#### Reset value

See individual bit resets

#### Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

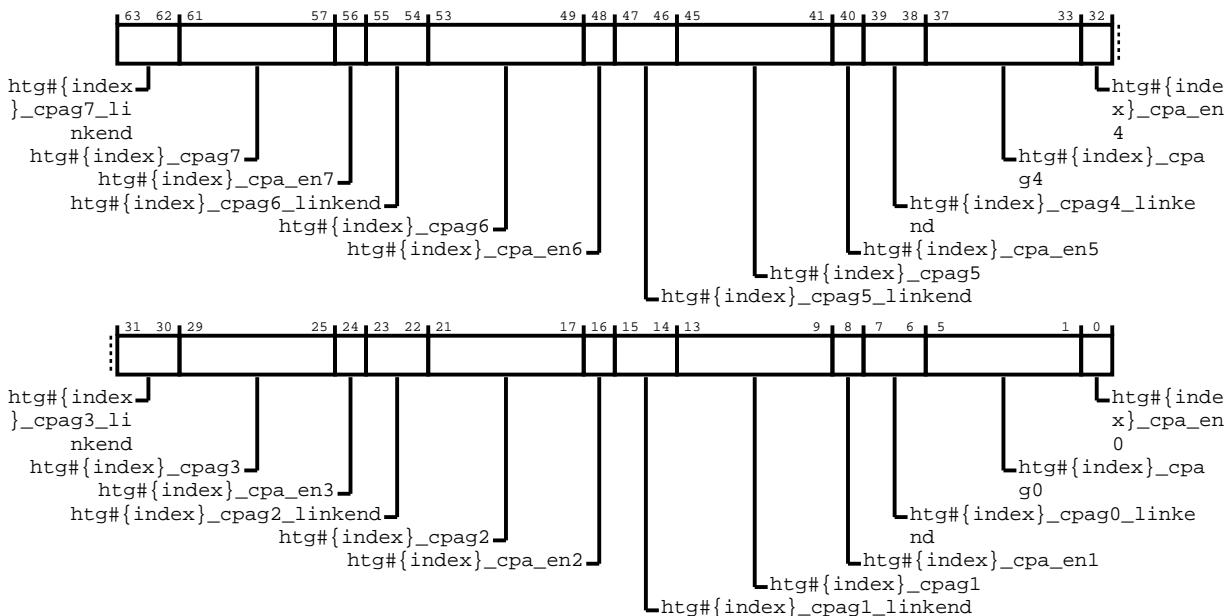
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-989: hashed\_target\_grp\_compact\_cpag\_ctrl0-31**



**Table 8-1003: hashed\_target\_grp\_compact\_cpag\_ctrl0-31 attributes**

Bits	Name	Description	Type	Reset
[63:62]	htg#{index}_cpag7_linkend	cpag id for index7	RW	0b0
[61:57]	htg#{index}_cpag7	cpag id for index7	RW	0b0
[56]	htg#{index}_cpa_en7	cpa enable for index7	RW	0b0
[55:54]	htg#{index}_cpag6_linkend	cpag id for index6	RW	0b0
[53:49]	htg#{index}_cpag6	cpag id for index6	RW	0b0
[48]	htg#{index}_cpa_en6	cpa enable for index6	RW	0b0
[47:46]	htg#{index}_cpag5_linkend	cpag id for index5	RW	0b0
[45:41]	htg#{index}_cpag5	cpag id for index5	RW	0b0
[40]	htg#{index}_cpa_en5	cpa enable for index5	RW	0b0

Bits	Name	Description	Type	Reset
[39:38]	htg#[index]_cpag4_linkend	cpag id for index4	RW	0b0
[37:33]	htg#[index]_cpag4	cpag id for index4	RW	0b0
[32]	htg#[index]_cpa_en4	cpa enable for index4	RW	0b0
[31:30]	htg#[index]_cpag3_linkend	cpag id for index3	RW	0b0
[29:25]	htg#[index]_cpag3	cpag id for index0	RW	0b0
[24]	htg#[index]_cpa_en3	cpa enable for index3	RW	0b0
[23:22]	htg#[index]_cpag2_linkend	cpag id for index2	RW	0b0
[21:17]	htg#[index]_cpag2	cpag id for index2	RW	0b0
[16]	htg#[index]_cpa_en2	cpa enable for index2	RW	0b0
[15:14]	htg#[index]_cpag1_linkend	cpag id for index1	RW	0b0
[13:9]	htg#[index]_cpag1	cpag id for index1	RW	0b0
[8]	htg#[index]_cpa_en1	cpa enable for index1	RW	0b0
[7:6]	htg#[index]_cpag0_linkend	cpag id for index0	RW	0b0
[5:1]	htg#[index]_cpag0	cpag id for index0	RW	0b0
[0]	htg#[index]_cpa_en0	cpa enable for index0	RW	0b0

### 8.3.17.39 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x3B00 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

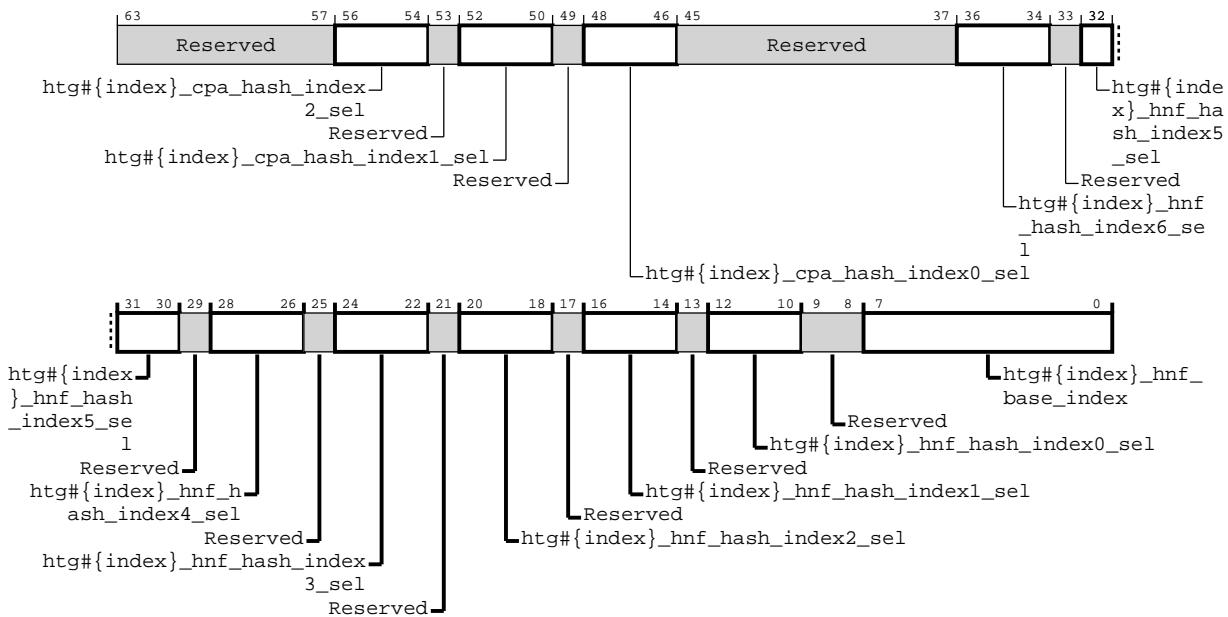
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-990: hashed\_target\_grp\_compact\_hash\_ctrl0-31**



**Table 8-1004: hashed\_target\_grp\_compact\_hash\_ctrl0-31 attributes**

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[56:54]	htg#[index]_cpa_hash_index2_sel	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index2.</p> <p><b>0b001</b> SMP hash index2 + 1.</p> <p><b>0b010</b> SMP hash index2 + 2.</p> <p><b>0b011</b> SMP hash index2 + 3.</p> <p><b>0b100</b> SMP hash index2 + 4.</p> <p><b>0b101</b> SMP hash index2 + 5.</p> <p><b>0b110</b> SMP hash index2 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[53]	Reserved	Reserved	RO	-
[52:50]	htg#[index]_cpa_hash_index1_sel	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index1.</p> <p><b>0b001</b> SMP hash index1 + 1.</p> <p><b>0b010</b> SMP hash index1 + 2.</p> <p><b>0b011</b> SMP hash index1 + 3.</p> <p><b>0b100</b> SMP hash index1 + 4.</p> <p><b>0b101</b> SMP hash index1 + 5.</p> <p><b>0b110</b> SMP hash index1 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[49]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[48:46]	htg#[index]_cpa_hash_index0_sel	<p>configures the CPAG hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index0.</p> <p><b>0b001</b> SMP hash index0 + 1.</p> <p><b>0b010</b> SMP hash index0 + 2.</p> <p><b>0b011</b> SMP hash index0 + 3.</p> <p><b>0b100</b> SMP hash index0 + 4.</p> <p><b>0b101</b> SMP hash index0 + 5.</p> <p><b>0b110</b> SMP hash index0 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[45:37]	Reserved	Reserved	RO	-
[36:34]	htg#[index]_hnf_hash_index6_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index6.</p> <p><b>0b001</b> SMP hash index6 + 1.</p> <p><b>0b010</b> SMP hash index6 + 2.</p> <p><b>0b011</b> SMP hash index6 + 3.</p> <p><b>0b100</b> SMP hash index6 + 4.</p> <p><b>0b101</b> SMP hash index6 + 5.</p> <p><b>0b110</b> SMP hash index6 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[33]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[32:30]	htg#[index]_hnf_hash_index5_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index5.</p> <p><b>0b001</b> SMP hash index5 + 1.</p> <p><b>0b010</b> SMP hash index5 + 2.</p> <p><b>0b011</b> SMP hash index5 + 3.</p> <p><b>0b100</b> SMP hash index5 + 4.</p> <p><b>0b101</b> SMP hash index5 + 5.</p> <p><b>0b110</b> SMP hash index5 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[29]	Reserved	Reserved	RO	-
[28:26]	htg#[index]_hnf_hash_index4_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index4.</p> <p><b>0b001</b> SMP hash index4 + 1.</p> <p><b>0b010</b> SMP hash index4 + 2.</p> <p><b>0b011</b> SMP hash index4 + 3.</p> <p><b>0b100</b> SMP hash index4 + 4.</p> <p><b>0b101</b> SMP hash index4 + 5.</p> <p><b>0b110</b> SMP hash index4 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[25]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[24:22]	htg#[index]_hnf_hash_index3_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index3.</p> <p><b>0b001</b> SMP hash index3 + 1.</p> <p><b>0b010</b> SMP hash index3 + 2.</p> <p><b>0b011</b> SMP hash index3 + 3.</p> <p><b>0b100</b> SMP hash index3 + 4.</p> <p><b>0b101</b> SMP hash index3 + 5.</p> <p><b>0b110</b> SMP hash index3 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[21]	Reserved	Reserved	RO	-
[20:18]	htg#[index]_hnf_hash_index2_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index2.</p> <p><b>0b001</b> SMP hash index2 + 1.</p> <p><b>0b010</b> SMP hash index2 + 2.</p> <p><b>0b011</b> SMP hash index2 + 3.</p> <p><b>0b100</b> SMP hash index2 + 4.</p> <p><b>0b101</b> SMP hash index2 + 5.</p> <p><b>0b110</b> SMP hash index2 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[17]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[16:14]	htg#[index]_hnf_hash_index1_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index1.</p> <p><b>0b001</b> SMP hash index1 + 1.</p> <p><b>0b010</b> SMP hash index1 + 2.</p> <p><b>0b011</b> SMP hash index1 + 3.</p> <p><b>0b100</b> SMP hash index1 + 4.</p> <p><b>0b101</b> SMP hash index1 + 5.</p> <p><b>0b110</b> SMP hash index1 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[13]	Reserved	Reserved	RO	-
[12:10]	htg#[index]_hnf_hash_index0_sel	<p>configures the local hnfs hash selection bits from the total hnfs hash across SMP.</p> <p><b>0b000</b> pass through from the SMP hnf_hash_index0.</p> <p><b>0b001</b> SMP hash index0 + 1.</p> <p><b>0b010</b> SMP hash index0 + 2.</p> <p><b>0b011</b> SMP hash index0 + 3.</p> <p><b>0b100</b> SMP hash index0 + 4.</p> <p><b>0b101</b> SMP hash index0 + 5.</p> <p><b>0b110</b> SMP hash index0 + 6.</p> <p><b>0b111</b> Hardcoded value 0b0</p>	RW	0b0
[9:8]	Reserved	Reserved	RO	-
[7:0]	htg#[index]_hnf_base_index	base index to the HNF target ID table This configuration is applicable for both modes COMPACT_HN_TABLES = 0/1 Programming 0xFF keeps RTL backward compatible	RW	Configuration dependent

### 8.3.17.40 rnsam\_hash\_addr\_mask\_reg

Configures the address mask that is applied before hashing the address bits.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE80

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

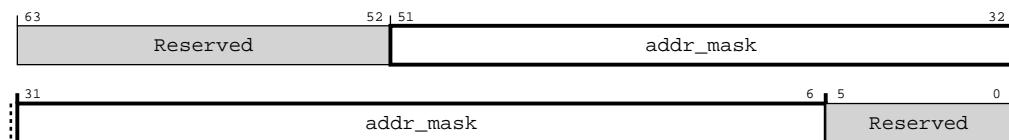
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-991: rnsam\_hash\_addr\_mask\_reg**



**Table 8-1005: rnsam\_hash\_addr\_mask\_reg attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:6]	addr_mask	Address mask applied before hashing	RW	0xFFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	

### 8.3.17.41 rnsam\_hash\_axi\_id\_mask\_reg

Configures the AXI\_ID mask that is applied before hashing the address bits.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE88

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

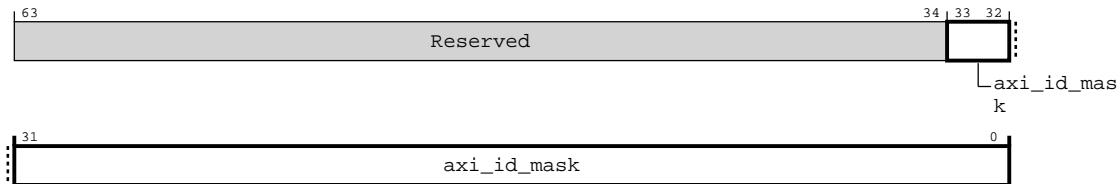
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-992: rnsam\_hash\_axi\_id\_mask\_reg**



**Table 8-1006: rnsam\_hash\_axi\_id\_mask\_reg attributes**

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33:0]	axi_id_mask	AXI_ID mask applied before hashing	RW	0x3FFFFFFF

### 8.3.17.42 rnsam\_region\_cmp\_addr\_mask\_reg

Configures the address mask that is applied before region compare.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE90

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

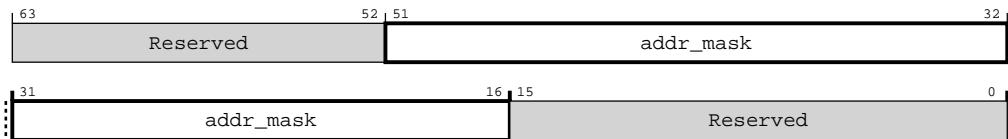
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-993: rnsam\_region\_cmp\_addr\_mask\_reg**



**Table 8-1007: rnsam\_region\_cmp\_addr\_mask\_reg attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region compare	RW	0xFFFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

### 8.3.17.43 cml\_port\_aggr\_grp0-31\_add\_mask

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CCIX port aggregation address mask for group #{{index}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

Address offset : index(0-5) : 0x11C0 + #{8 \* index}

##### index(6-37)

0x2B00 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

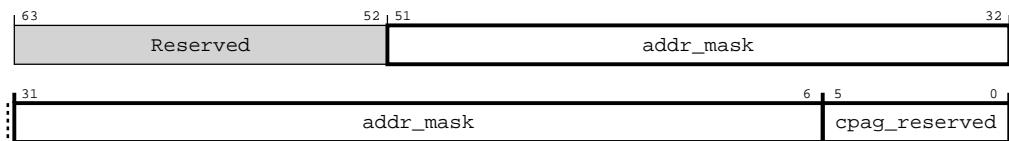
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-994: cml\_port\_aggr\_grp0-31\_add\_mask**



**Table 8-1008: cml\_port\_aggr\_grp0-31\_add\_mask attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address/AXID mask to be applied before hashing <b>CONSTRAINT</b> ADDR MASK is [51:6] <b>CONSTRAINT</b> AXID MASK is [31:0] = [37:6]	RW	0x3FFFFFFFFF
[5:0]	cpag_reserved	reserved bits	RW	0x0

## 8.3.17.44 cml\_cpag\_base\_idx\_grp0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures the CPAG base indexes.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

index(0-7) : 0x2B00 + #8 \* index}

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

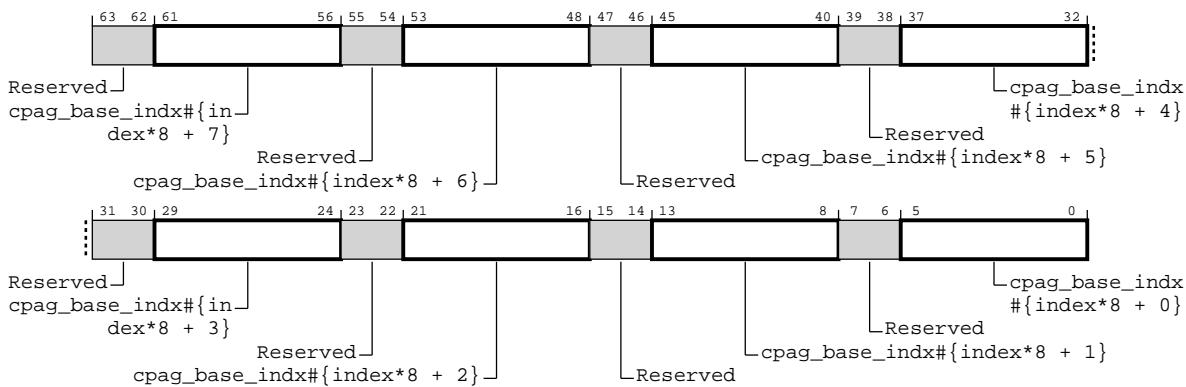
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-995: cml\_cpag\_base\_idx\_grp0-3**



**Table 8-1009: cml\_cpag\_base\_idx\_grp0-3 attributes**

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	
[61:56]	cpag_base_idx#{index*8 + 7}	Configures the CPAG base index #{8*index + 7}	RW	0x3F
[55:54]	Reserved	Reserved	RO	
[53:48]	cpag_base_idx#{index*8 + 6}	Configures the CPAG base index #{8*index + 6}	RW	0x3F
[47:46]	Reserved	Reserved	RO	
[45:40]	cpag_base_idx#{index*8 + 5}	Configures the CPAG base index #{8*index + 5}	RW	0x3F
[39:38]	Reserved	Reserved	RO	
[37:32]	cpag_base_idx#{index*8 + 4}	Configures the CPAG base index #{8*index + 4}	RW	0x3F

Bits	Name	Description	Type	Reset
[31:30]	Reserved	Reserved	RO	
[29:24]	cpag_base_idx#{index*8 + 3}	Configures the CPAG base index #{8*index + 3}	RW	0x3F
[23:22]	Reserved	Reserved	RO	
[21:16]	cpag_base_idx#{index*8 + 2}	Configures the CPAG base index #{8*index + 2}	RW	0x3F
[15:14]	Reserved	Reserved	RO	
[13:8]	cpag_base_idx#{index*8 + 1}	Configures the CPAG base index #{8*index + 1}	RW	0x3F
[7:6]	Reserved	Reserved	RO	
[5:0]	cpag_base_idx#{index*8 + 0}	Configures the CPAG base index #{8*index + 0}	RW	0x3F

### 8.3.17.45 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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

###### index(0-2)

0x11F0 + #{8 \* index}

index(3-14) : 0x2C00 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

##### Usage constraints

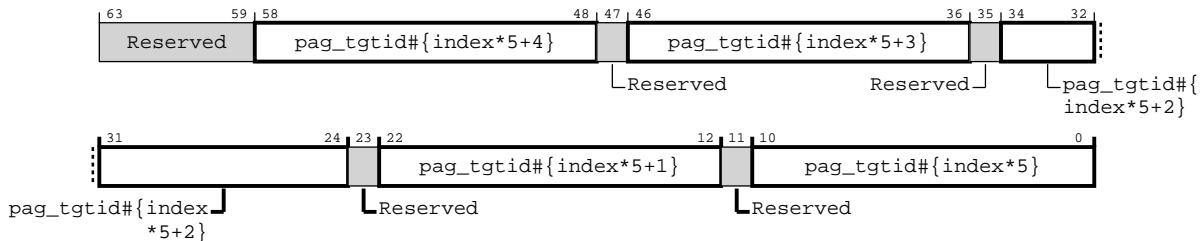
This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range bit 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 8-996: cml\_port\_aggr\_grp\_reg0-12**



**Table 8-1010: cml\_port\_aggr\_grp\_reg0-12 attributes**

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	
[58:48]	pag_tgtid#[index*5+4]	Specifies target ID #{index*5+4} for CPAG	RW	0b0
[47]	Reserved	Reserved	RO	
[46:36]	pag_tgtid#[index*5+3]	Specifies target ID #{index*5+3} for CPAG	RW	0b0
[35]	Reserved	Reserved	RO	
[34:24]	pag_tgtid#[index*5+2]	Specifies target ID #{index*5+2} for CPAG	RW	0b0
[23]	Reserved	Reserved	RO	
[22:12]	pag_tgtid#[index*5+1]	Specifies target ID #{index*5+1} for CPAG	RW	0b0
[11]	Reserved	Reserved	RO	
[10:0]	pag_tgtid#[index*5]	Specifies target ID #{index*5} for CPAG	RW	0b0

### 8.3.17.46 cml\_port\_aggr\_ctrl\_reg

Configures the CCI-X port aggregation port IDs for group 2.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0x1208

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

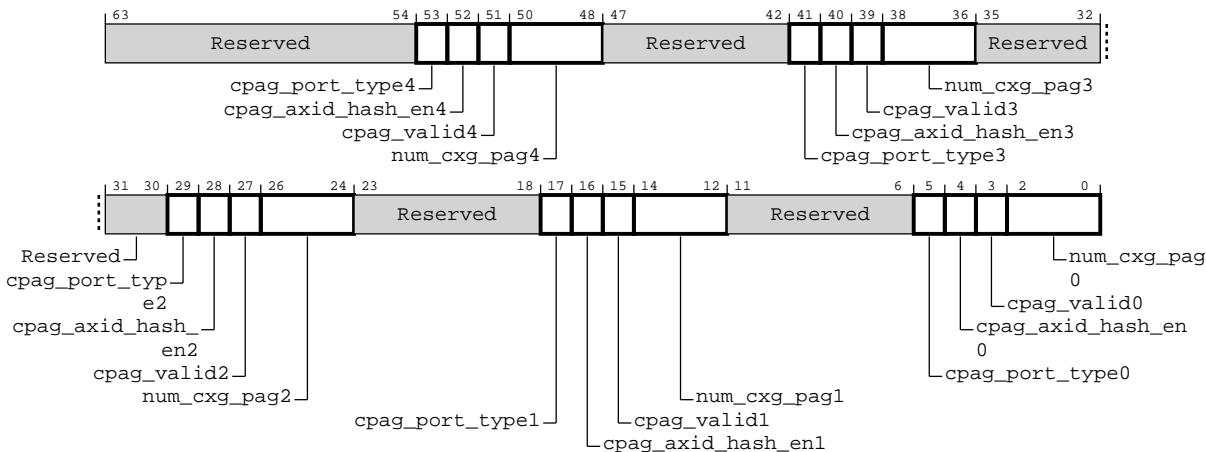
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-997: cml\_port\_aggr\_ctrl\_reg**



**Table 8-1011: cml\_port\_aggr\_ctrl\_reg attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53]	cpag_port_type4	Specifies the port type  <b>0b0</b> CXL port  <b>0b1</b> CML SMP port	RW	0b0

Bits	Name	Description	Type	Reset
[52]	cpag_axid_hash_en4	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[51]	cpag_valid4	Valid programming for CPAG4, Enabled by default (backward compatible)	RW	0b1
[50:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG4 <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[47:42]	Reserved	Reserved	RO	
[41]	cpag_port_type3	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[40]	cpag_axid_hash_en3	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[39]	cpag_valid3	Valid programming for CPAG + 3, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[38:36]	num_cxg_pag3	<p>Specifies the number of CXRAs in CPAG3</p> <p><b>0b000</b> 1 port</p> <p><b>0b001</b> 2 ports</p> <p><b>0b010</b> 4 ports</p> <p><b>0b011</b> 8 ports</p> <p><b>0b100</b> 16 ports</p> <p><b>0b101</b> 32 ports</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b0
[35:30]	Reserved	Reserved	RO	
[29]	cpag_port_type2	<p>Specifies the port type</p> <p><b>0b0</b> CXL port</p> <p><b>0b1</b> CML SMP port</p>	RW	0b0
[28]	cpag_axid_hash_en2	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[27]	cpag_valid2	Valid programming for CPAG + 2, Enabled by default (backward compatible)	RW	0b1
[26:24]	num_cxg_pag2	<p>Specifies the number of CXRAs in CPAG2</p> <p><b>0b000</b> 1 port</p> <p><b>0b001</b> 2 ports</p> <p><b>0b010</b> 4 ports</p> <p><b>0b011</b> 8 ports</p> <p><b>0b100</b> 16 ports</p> <p><b>0b101</b> 32 ports</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[23:18]	Reserved	Reserved	RO	
[17]	cpag_port_type1	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[16]	cpag_axid_hash_en1	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[15]	cpag_valid1	Valid programming for CPAG + 1, Enabled by default (backward compatible)	RW	0b1
[14:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG1 <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[11:6]	Reserved	Reserved	RO	
[5]	cpag_port_type0	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[4]	cpag_axid_hash_en0	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[3]	cpag_valid0	Valid programming for CPAG, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[2:0]	num_cxg_pag0	<p>Specifies the number of CXRAs in CPAG0</p> <p><b>0b000</b> 1 port</p> <p><b>0b001</b> 2 ports</p> <p><b>0b010</b> 4 ports</p> <p><b>0b011</b> 8 ports</p> <p><b>0b100</b> 16 ports</p> <p><b>0b101</b> 32 ports</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b0

### 8.3.17.47 cml\_port\_aggr\_ctrl\_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures the CCIX port aggregation port IDs for group 2.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15) : 0x1208 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

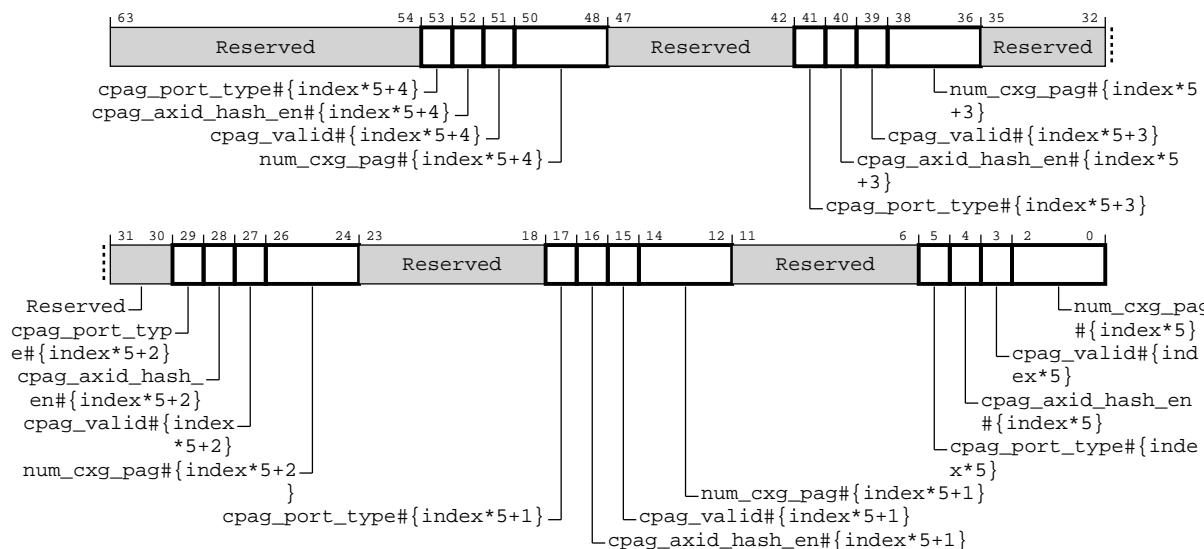
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-998: cml\_port\_aggr\_ctrl\_reg1-6**



**Table 8-1012: cml\_port\_aggr\_ctrl\_reg1-6 attributes**

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	
[53]	cpag_port_type#{index*5+4}	Specifies the port type  <b>0b0</b> CXL port  <b>0b1</b> CML SMP port	RW	0b0
[52]	cpag_axid_hash_en#{index*5+4}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[51]	cpag_valid#{index*5+4}	Valid programming for CPAG #{index*5 + 4}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag#{index*5+4}	Specifies the number of CXRAs in CPAG #{{index}*5 + 4} <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[47:42]	Reserved	Reserved	RO	
[41]	cpag_port_type#{index*5+3}	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[40]	cpag_axid_hash_en#{index*5+3}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{{index}*5 + 3}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[38:36]	num_cxg_pag#{index*5+3}	Specifies the number of CXRAs in CPAG #{{index}*5 + 3} <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[35:30]	Reserved	Reserved	RO	
[29]	cpag_port_type#{index*5+2}	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[28]	cpag_axid_hash_en#{index*5+2}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{{index}*5 + 2}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[26:24]	num_cxg_pag#{index*5+2}	Specifies the number of CXRAs in CPAG #{{index}*5 + 2} <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[23:18]	Reserved	Reserved	RO	
[17]	cpag_port_type#{index*5+1}	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[16]	cpag_axid_hash_en#{index*5+1}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{{index}*5 + 1}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[14:12]	num_cxg_pag#{index*5+1}	Specifies the number of CXRAs in CPAG #{index*5 + 1} <b>0b000</b> 1 port <b>0b001</b> 2 ports <b>0b010</b> 4 ports <b>0b011</b> 8 ports <b>0b100</b> 16 ports <b>0b101</b> 32 ports <b>0b110</b> 3 ports (MOD-3 hash) <b>0b111</b> Reserved	RW	0b0
[11:6]	Reserved	Reserved	RO	
[5]	cpag_port_type#{index*5}	Specifies the port type <b>0b0</b> CXL port <b>0b1</b> CML SMP port	RW	0b0
[4]	cpag_axid_hash_en#{index*5}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	0b0
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	0b1

Bits	Name	Description	Type	Reset
[2:0]	num_cxg_pag#{index*5}	<p>Specifies the number of CXRAs in CPAG #{index*5}</p> <p><b>0b000</b> 1 port</p> <p><b>0b001</b> 2 ports</p> <p><b>0b010</b> 4 ports</p> <p><b>0b011</b> 8 ports</p> <p><b>0b100</b> 16 ports</p> <p><b>0b101</b> 32 ports</p> <p><b>0b110</b> 3 ports (MOD-3 hash)</p> <p><b>0b111</b> Reserved</p>	RW	0b0

### 8.3.17.48 sys\_cache\_grp\_sn\_attr

Configures attributes for SN node IDs for system cache groups.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEB0

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

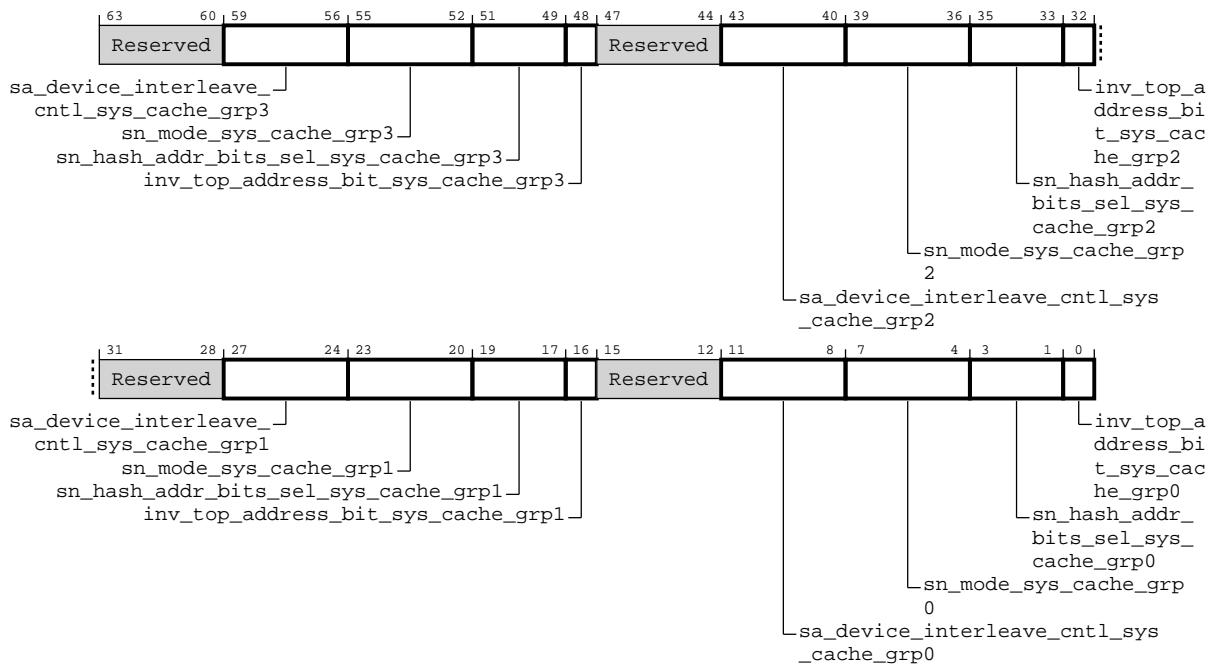
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-999: sys\_cache\_grp\_sn\_attr**



**Table 8-1013: sys\_cache\_grp\_sn\_attr attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[59:56]	sa_device_interleave_cntl_sys_cache_grp3	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[55:52]	sn_mode_sys_cache_grp3	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[51:49]	sn_hash_addr_bits_sel_sys_cache_grp3	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[48]	inv_top_address_bit_sys_cache_grp3	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[47:44]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[43:40]	sa_device_interleave_cntl_sys_cache_grp2	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[39:36]	sn_mode_sys_cache_grp2	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b00

Bits	Name	Description	Type	Reset
[35:33]	sn_hash_addr_bits_sel_sys_cache_grp2	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[32]	inv_top_address_bit_sys_cache_grp2	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[31:28]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[27:24]	sa_device_interleave_cntl_sys_cache_grp1	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[23:20]	sn_mode_sys_cache_grp1	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[19:17]	sn_hash_addr_bits_sel_sys_cache_grp1	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[16]	inv_top_address_bit_sys_cache_grp1	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[15:12]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[11:8]	sa_device_interleave_cntl_sys_cache_grp0	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[7:4]	sn_mode_sys_cache_grp0	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[3:1]	sn_hash_addr_bits_sel_sys_cache_grp0	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[0]	inv_top_address_bit_sys_cache_grp0	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0

### 8.3.17.49 sys\_cache\_grp\_sn\_attr1

Configures attributes for SN node IDs for system cache groups.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEB8

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

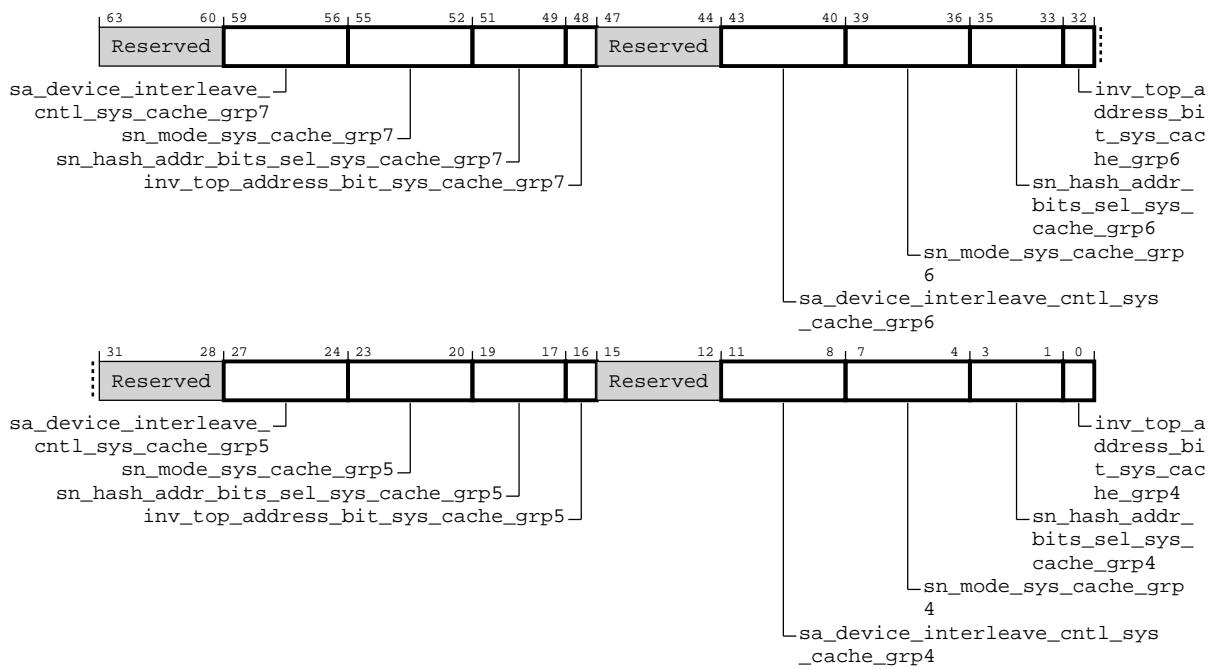
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1000: sys\_cache\_grp\_sn\_attr1**



**Table 8-1014: sys\_cache\_grp\_sn\_attr1 attributes**

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[59:56]	sa_device_interleave_cntl_sys_cache_grp7	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[55:52]	sn_mode_sys_cache_grp7	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[51:49]	sn_hash_addr_bits_sel_sys_cache_grp7	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[48]	inv_top_address_bit_sys_cache_grp7	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[47:44]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[43:40]	sa_device_interleave_cntl_sys_cache_grp6	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[39:36]	sn_mode_sys_cache_grp6	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b00

Bits	Name	Description	Type	Reset
[35:33]	sn_hash_addr_bits_sel_sys_cache_grp6	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[32]	inv_top_address_bit_sys_cache_grp6	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[31:28]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[27:24]	sa_device_interleave_cntl_sys_cache_grp5	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[23:20]	sn_mode_sys_cache_grp5	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[19:17]	sn_hash_addr_bits_sel_sys_cache_grp5	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[16]	inv_top_address_bit_sys_cache_grp5	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[15:12]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[11:8]	sa_device_interleave_cntl_sys_cache_grp4	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[7:4]	sn_mode_sys_cache_grp4	<p>SN selection mode</p> <p><b>0b0000</b> 1-SN mode (SNO)</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> Reserved #CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0

Bits	Name	Description	Type	Reset
[3:1]	sn_hash_addr_bits_sel_sys_cache_grp4	<p>SN hash address select(Valid for 3SN, 5SN, 6SN)</p> <p><b>0b000</b> [16:8] address bits (Default)</p> <p><b>0b001</b> [17:9] address bits</p> <p><b>0b010</b> [18:10] address bits</p> <p><b>0b011</b> [19:11] address bits</p> <p><b>0b100</b> [20:12] address bits</p> <p><b>0b101</b> [21:13] address bits</p> <p><b>0b110</b> [22:14] address bits</p> <p><b>0b111</b> [23:15] address bits</p> <p><b>Others</b> Reserved</p>	RW	0x0
[0]	inv_top_address_bit_sys_cache_grp4	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0

### 8.3.17.50 sys\_cache\_grp\_sn\_sam\_cfg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures top address bits for SN SAM system cache groups #*{index2}* and #*{index2 + 1}*. All top\_address\_bit fields must be between bits 47 and 28. top\_address\_bit2 > top\_address\_bit1 > top\_address\_bit0.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7) : 0x1140 + #*{8 \* index}*

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

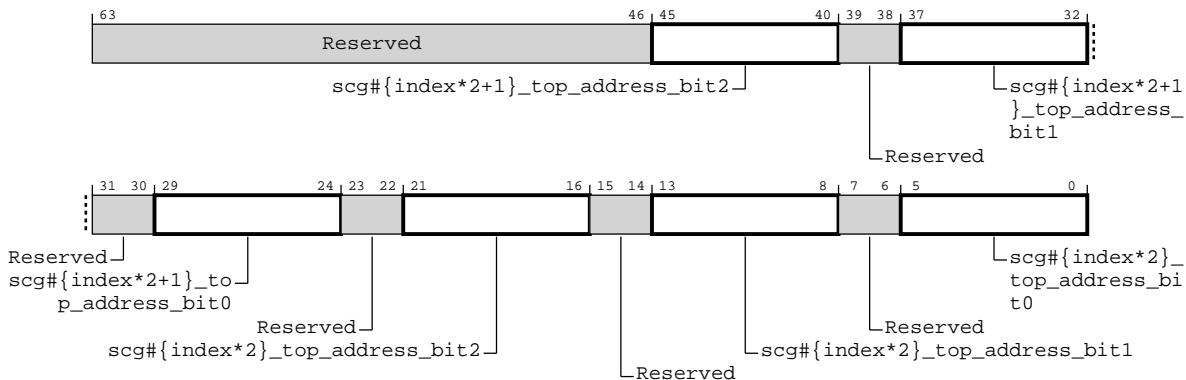
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1001: sys\_cache\_grp\_sn\_sam\_cfg0-3**



**Table 8-1015: sys\_cache\_grp\_sn\_sam\_cfg0-3 attributes**

Bits	Name	Description	Type	Reset
[63:46]	Reserved	Reserved	RO	
[45:40]	scg#{index*2+1}_top_address_bit2	Top address bit 2 for system cache group #{index*2+1}	RW	0x00
[39:38]	Reserved	Reserved	RO	
[37:32]	scg#{index*2+1}_top_address_bit1	Top address bit 1 for system cache group #{index*2+1}	RW	0x00
[31:30]	Reserved	Reserved	RO	
[29:24]	scg#{index*2+1}_top_address_bit0	Top address bit 0 for system cache group #{index*2+1}	RW	0x00
[23:22]	Reserved	Reserved	RO	
[21:16]	scg#{index*2}_top_address_bit2	Top address bit 2 for system cache group #{index*2}	RW	0x00
[15:14]	Reserved	Reserved	RO	
[13:8]	scg#{index*2}_top_address_bit1	Top address bit 1 for system cache group #{index*2}	RW	0x00
[7:6]	Reserved	Reserved	RO	
[5:0]	scg#{index*2}_top_address_bit0	Top address bit 0 for system cache group #{index*2}	RW	0x00

### 8.3.17.51 sam\_qos\_mem\_region\_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the QoS value for memory region #{{index}}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15) : 0x1280 + #{{8 \* index}}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

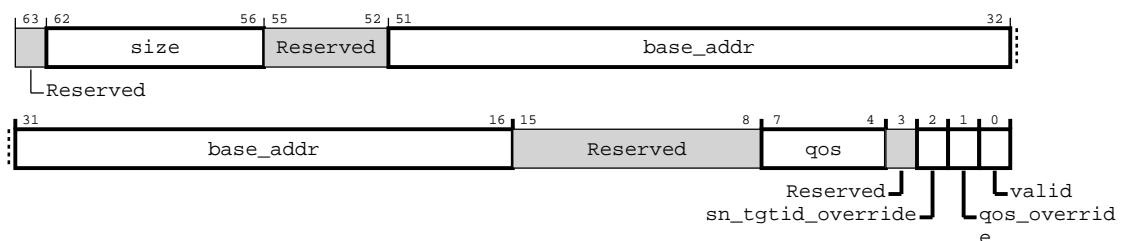
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1002: sam\_qos\_mem\_region\_reg0-15**



**Table 8-1016: sam\_qos\_mem\_region\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	size	Memory region size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:8]	Reserved	Reserved	RO	-
[7:4]	qos	Indicates the QoS value to be used for this region	RW	0b0000
[3]	Reserved	Reserved	RO	-
[2]	sn_tgtid_override	Override the SN targetId for address contained in the region of this register	RW	0b0
[1]	qos_override	QoS Memory region allow override <b>0b0</b> Do not override the QoS value from the QoS regulator <b>0b1</b> Override the QoS value with the programmed value in regionX_qos	RW	0b0
[0]	valid	QoS Memory region valid <b>0b0</b> Not valid <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.17.52 sam\_qos\_mem\_region\_cfg2\_rego-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the QOS memory region #{{index}}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-15) : 0x1340 + #{{8 \* index}}

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

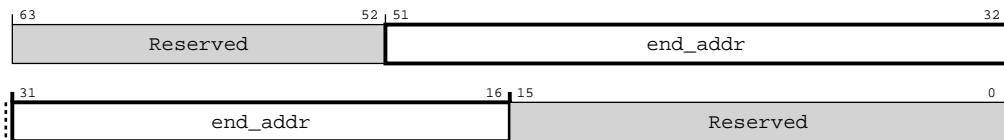
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1003: sam\_qos\_mem\_region\_cfg2\_reg0-15**



**Table 8-1017: sam\_qos\_mem\_region\_cfg2\_reg0-15 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:0]	Reserved	Reserved	RO	

### 8.3.17.53 sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg1\_reg0-511%64

There are 512 iterations of this register. The index ranges from 0 to 511. Configures the prefetch nonhash memory region #{{index}}

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

index(0-511) : 0x4000 + #{8 \* index}

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

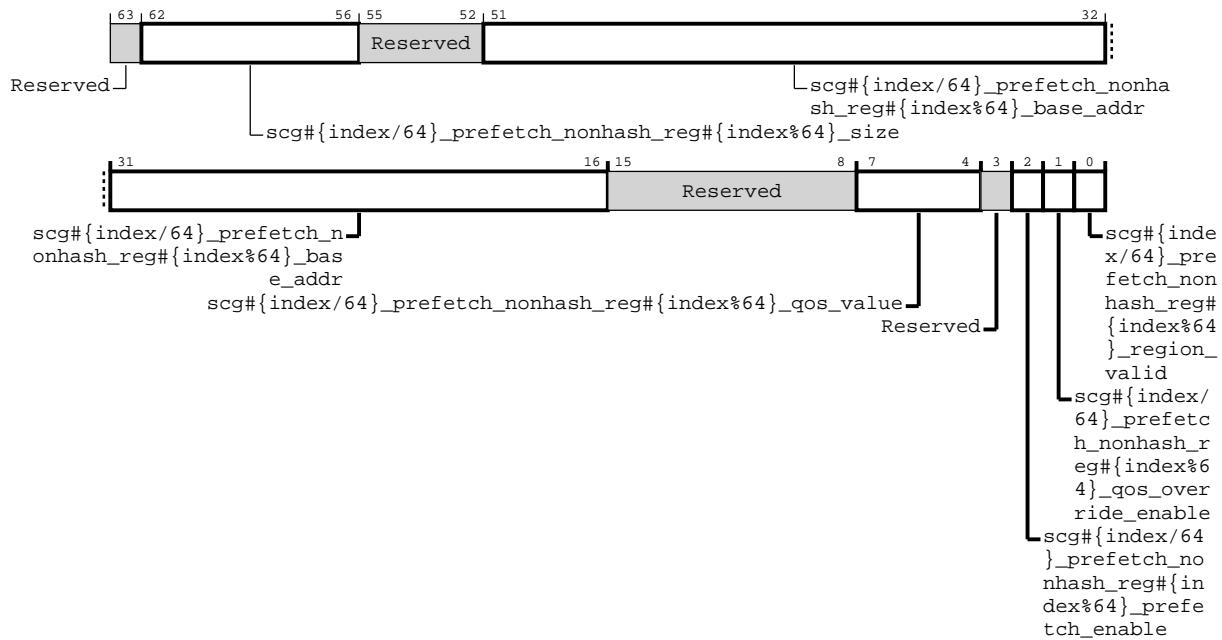
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1004: sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg1\_reg0-511%64**



**Table 8-1018: sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg1\_reg0-511%64 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	scg#[index/64]_prefetch_nonhash_reg#{index%64}_size	Memory region size <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	scg#[index/64]_prefetch_nonhash_reg#{index%64}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:8]	Reserved	Reserved	RO	-
[7:4]	scg#[index/64]_prefetch_nonhash_reg#{index%64}_qos_value	Indicates the QoS value to be used for this region	RW	0b0000
[3]	Reserved	Reserved	RO	-
[2]	scg#[index/64]_prefetch_nonhash_reg#{index%64}_prefetch_enable	Enables the prefetch non-hashed logic (added to enable QOS override only functionality) <b>0b0</b> Disables the prefetch functionality (programmed if only QOS override functionality is intended) <b>0b1</b> Enables the prefetch functionality (Enabled by default)	RW	0b1

Bits	Name	Description	Type	Reset
[1]	scg#[{index/64}_prefetch_nonhash_reg#{index%64}_qos_override_enable	Prefetch nonhash allow QOS override  <b>0b0</b> Do not override the QoS value from the QoS regulator  <b>0b1</b> Override the QoS value with the programmed value in regionX_qos	RW	0b0
[0]	scg#[{index/64}_prefetch_nonhash_reg#{index%64}_region_valid	Prefetch Nonhash region valid  <b>0b0</b> Not valid  <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.17.54 sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg2\_reg0-511%64

There are 512 iterations of this register. The index ranges from 0 to 511. Configures the Prefetch nonhash memory region #{index}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-511) : 0x5000 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

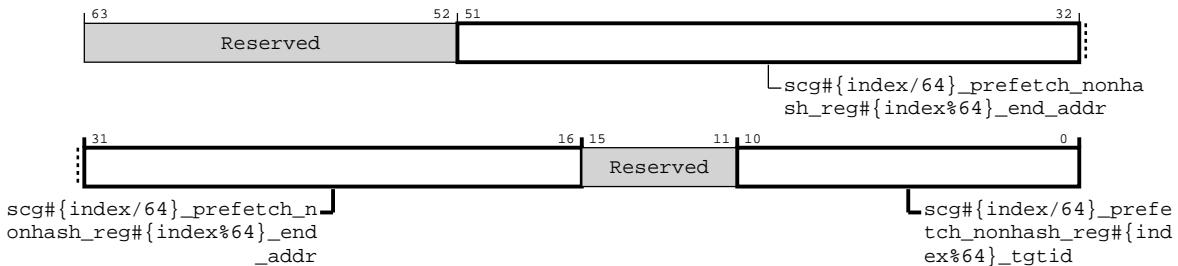
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1005: sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg2\_reg0-511%64**



**Table 8-1019: sam\_scg0-511/64\_prefetch\_nonhashed\_mem\_region\_cfg2\_reg0-511%64 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	scg#[{index/64}_prefetch_nonhash_reg#{index%64}_end_addr]	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:11]	Reserved	Reserved	RO	
[10:0]	scg#[{index/64}_prefetch_nonhash_reg#{index%64}_tgtid]	SN TgtID for the non-hashed region	RW	0x0

### 8.3.17.55 sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg1\_reg0-63%8

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the prefetch hashed memory region #{index}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-63) : 0x6000 + #8 \* index

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

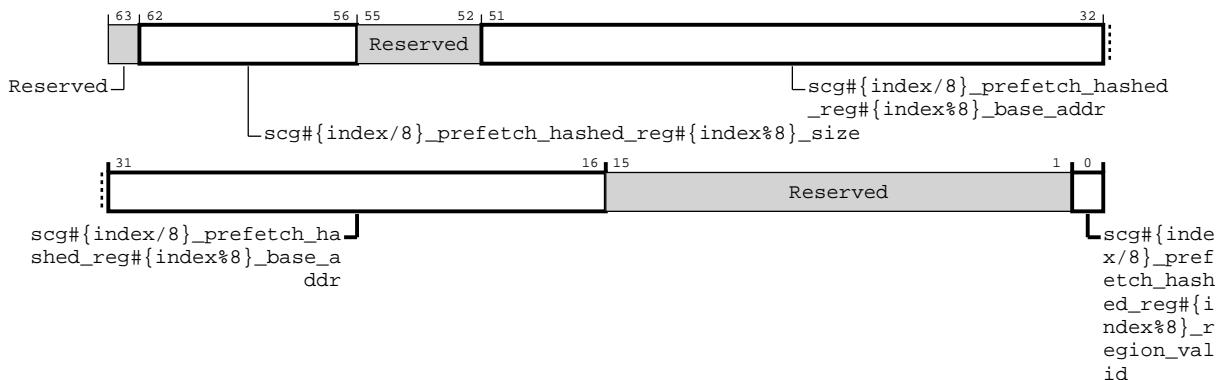
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1006: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg1\_reg0-63%8**



**Table 8-1020: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg1\_reg0-63%8 attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	scg#[index/8]_prefetch_hashed_reg#[index%8]_size	Memory region size  <b>CONSTRAINT</b> Memory region must be a power of two, from minimum size supported to maximum memory size ( $2^{\text{address width}}$ ).	RW	0b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	scg#[index/8]_prefetch_hashed_reg#[index%8]_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:1]	Reserved	Reserved	RO	-
[0]	scg#[index/8]_prefetch_hashed_reg#[index%8]_region_valid	Prefetch Hashed region valid  <b>0b0</b> Not valid  <b>0b1</b> Valid for memory region comparison	RW	0b0

### 8.3.17.56 sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg2\_reg0-63%

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the Prefetch hashed memory region #{index}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-63) : 0x6200 + #8 \* index

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

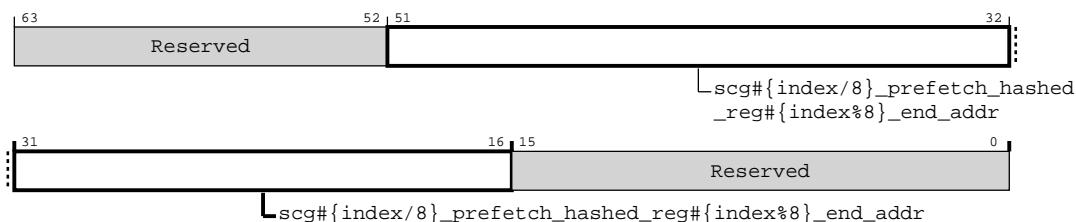
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1007: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg2\_reg0-63%**



**Table 8-1021: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg2\_reg0-63%8 attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:16]	scg#[index/8]_prefetch_hashed_reg#[index %8]_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	0x0
[15:0]	Reserved	Reserved	RO	

### 8.3.17.57 sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg3\_reg0-63%8

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the Prefetch hashed memory region #{index}

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-63) : 0x6400 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

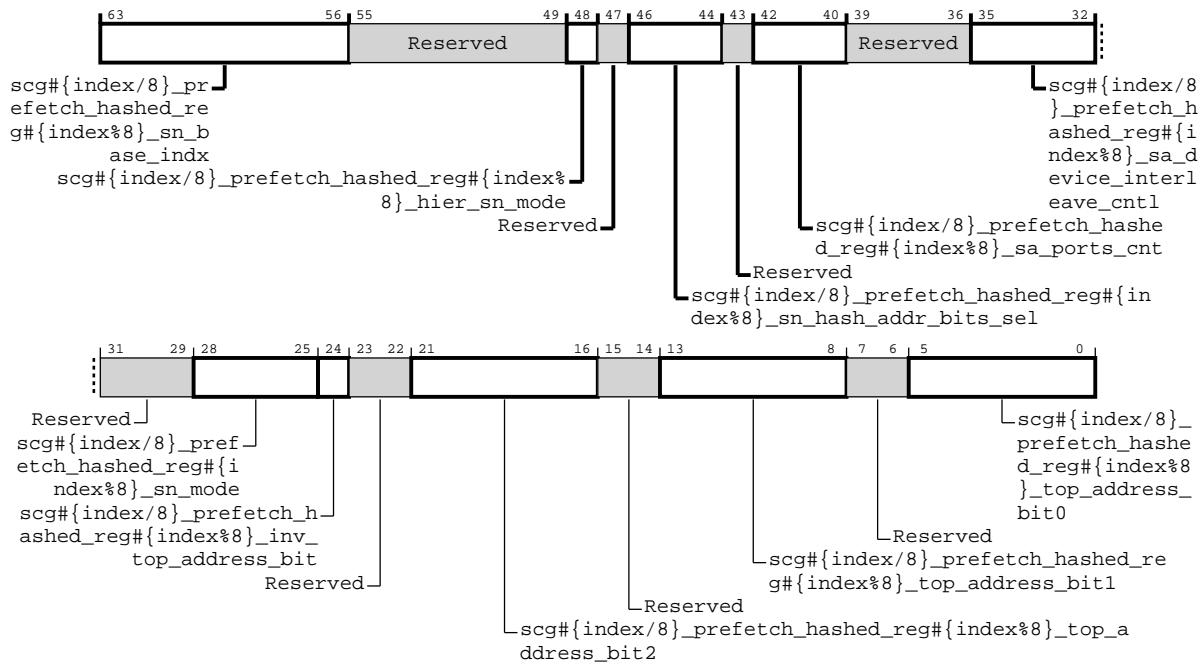
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1008: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg3\_reg0-63%8**



**Table 8-1022: sam\_scg0-63/8\_prefetch\_hashed\_region\_cfg3\_reg0-63%8 attributes**

Bits	Name	Description	Type	Reset
[63:56]	scg#[index/8]_prefetch_hashed_reg#[index%8]_sn_base_idx	Base index for the prefetch SN TgtID table Programming 0xFF keeps RTL backward compatible ## Do not program this register	RW	0xFF
[55:49]	Reserved	Reserved	RO	
[48]	scg#[index/8]_prefetch_hashed_reg#[index%8]_hier_sn_mode	When Hier SN enabled, SN mode is per cluster (Hierarchical hashing enabled). Otherwise, SN mode is defined for all HNFs. Not applicable for CXSA mode.	RW	0x0
[47]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[46:44]	scg#[index/8]_prefetch_hashed_reg#[index%8]_sn_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN) <b>0b000</b> [16:8] address bits (Default) <b>0b001</b> [17:9] address bits <b>0b010</b> [18:10] address bits <b>0b011</b> [19:11] address bits <b>0b100</b> [20:12] address bits <b>0b101</b> [21:13] address bits <b>0b110</b> [22:14] address bits <b>0b111</b> [23:15] address bits <b>Others</b> Reserved	RW	0x0
[43]	Reserved	Reserved	RO	
[42:40]	scg#[index/8]_prefetch_hashed_reg#[index%8]_sa_ports_cnt	Specifies the number of CXSA/CXLSA device aggregated <b>0b000</b> 1 port used <b>0b001</b> 2 ports used <b>0b010</b> 4 ports used <b>0b011</b> 8 ports used <b>0b100</b> 16 ports used <b>0b101</b> 3 ports <b>0b110</b> 6 ports <b>0b111</b> 12 ports	RW	0b0
[39:36]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[35:32]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_sa_device_interleave_crtl	<p>This field controls the interleave size across all aggregated CXSA/CXLSA Devices. This field is used for arithmetic modulo to select the address bits</p> <p><b>0x0</b> 64B Interleaved (CXSA), [52:6] - arithmetic modulo</p> <p><b>0x1</b> 128B Interleaved (CXSA), [52:7] - arithmetic modulo</p> <p><b>0x2</b> 256B Interleaved (CXSA), [52:8] - arithmetic modulo</p> <p><b>0x3</b> 512B Interleaved (CXSA), [52:9] - arithmetic modulo</p> <p><b>0x4</b> 1KB Interleaved (CXSA), [52:10] - arithmetic modulo</p> <p><b>0x5</b> 2KB Interleaved (CXSA), [52:11] - arithmetic modulo</p> <p><b>0x6</b> 4KB Interleaved (CXSA), [52:12] - arithmetic modulo</p> <p><b>0x7</b> 8KB Interleaved (CXSA), [52:13] - arithmetic modulo</p> <p><b>0x8</b> 16KB Interleaved (CXSA), [52:14] - arithmetic modulo</p> <p><b>0x9</b> Reserved (CXSA), [52:15] - arithmetic modulo</p> <p><b>Others</b> Reserved</p>	RW	0b0
[31:29]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[28:25]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_sn_mode	<p>SN selection mode</p> <p><b>0b0000</b> Reserved</p> <p><b>0b0001</b> 3-SN mode (SNO, SN1, SN2)</p> <p><b>0b0010</b> 6-SN mode (SNO, SN1, SN2, SN3, SN4, SN5)</p> <p><b>0b0011</b> 5-SN mode (SNO, SN1, SN2, SN3, SN4)</p> <p><b>0b0100</b> 2-SN mode (SNO, SN1) power of 2 hashing</p> <p><b>0b0101</b> 4-SN mode (SNO, SN1, SN2, SN3) power of 2 hashing</p> <p><b>0b0110</b> 8-SN mode (SNO, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing</p> <p><b>0b0111</b> CXSA/CXLSA aggregated SA selection function</p> <p><b>0b1000</b> 3-SN mode (arithmetic modulo)</p> <p><b>0b1001</b> 6-SN mode (arithmetic modulo)</p> <p><b>Others</b> Reserved</p>	RW	0b0
[24]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_inv_top_address_bit	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	0x0
[23:22]	Reserved	Reserved	RO	
[21:16]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_top_address_bit2	Top address bit 2	RW	0x00
[15:14]	Reserved	Reserved	RO	
[13:8]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_top_address_bit1	Top address bit 1	RW	0x00
[7:6]	Reserved	Reserved	RO	
[5:0]	scg#[{index/8}_prefetch_hashed_reg#{index %8}_top_address_bit0	Top address bit 0	RW	0x00

### 8.3.17.58 sys\_cache\_grp\_sn\_nodeid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed node IDs for system cache groups. Controls target SN node IDs #{{index}4} to #{{index}4 + 3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x1000 + #8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

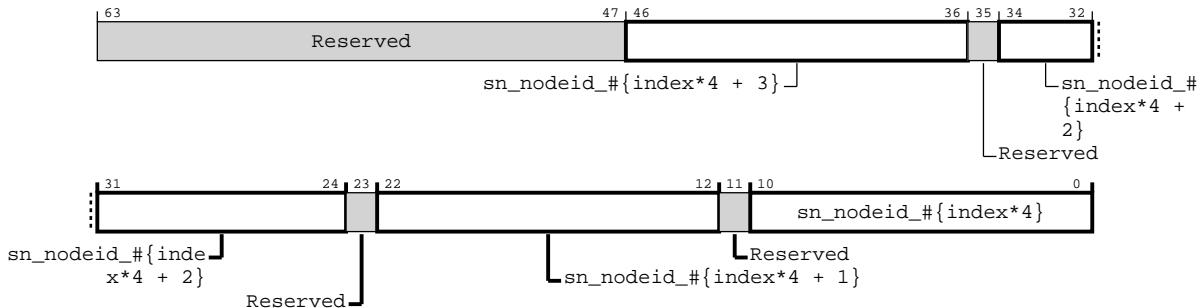
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1009: sys\_cache\_grp\_sn\_nodeid\_reg0-31**



**Table 8-1023: sys\_cache\_grp\_sn\_nodeid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	sn_nodeid_{index*4 + 3}	Default Hashed target SN node ID #{index*4 + 3}	RW	0b00000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	sn_nodeid_{index*4 + 2}	Default Hashed target SN node ID #{index*4 + 2}	RW	0b00000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	sn_nodeid_{index*4 + 1}	Default Hashed target SN node ID #{index*4 + 1}	RW	0b00000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_{index*4}	Default Hashed target SN node ID #{index*4}	RW	0b00000000000000

### 8.3.17.59 sys\_cache\_grp\_region0-63/32\_sn\_nodeid\_reg0-63%32

There are 64 iterations of this register. The index ranges from 0 to 63. Configures node IDs for SCG's Default hashed Region memory. Controls target SN node IDs #{index\*4} to #{index\*4+3}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-63) : 0x1400 + #8 \* index

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

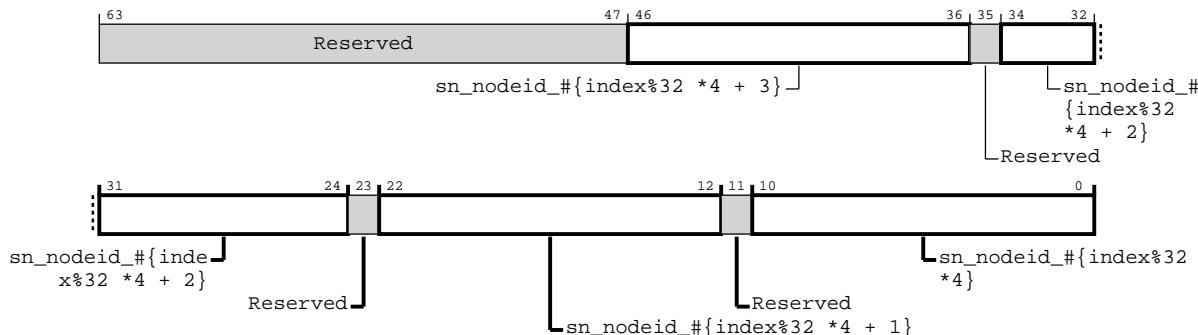
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1010: sys\_cache\_grp\_region0-63/32\_sn\_nodeid\_reg0-63%32**



**Table 8-1024: sys\_cache\_grp\_region0-63/32\_sn\_nodeid\_reg0-63%32 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	sn_nodeid_{index%32 *4 + 3}	Hashed target SN node ID #{index%32 * 4 +3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	sn_nodeid_{index%32 *4 + 2}	Hashed target SN node ID #{index%32 * 4 +2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	sn_nodeid_{index%32 *4 + 1}	Hashed target SN node ID #{index%32 * 4 +1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_{index%32 *4}	Hashed target SN node ID #{index%32 * 4}	RW	0b000000000000

### 8.3.17.60 sys\_cache\_grp\_hashed\_regions\_sn\_nodeid\_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures SN node IDs for SCG's Hashed groups in the HNSAM . Controls target SN node IDs #{{index4}} to #{{index4 + 3}}.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-31) : 0x6600 + #8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

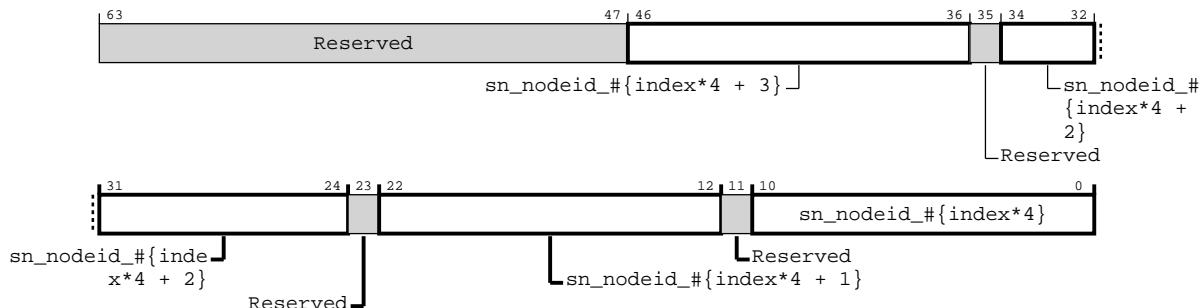
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1011: sys\_cache\_grp\_hashed\_regions\_sn\_nodeid\_reg0-31**



**Table 8-1025: sys\_cache\_grp\_hashed\_regions\_sn\_nodeid\_reg0-31 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	$\text{sn\_nodeid\_#\{index*4 + 3\}}$	Hashed target SN node ID # $\{\text{index*4} + 3\}$	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	$\text{sn\_nodeid\_#\{index*4 + 2\}}$	Hashed target SN node ID # $\{\text{index*4} + 2\}$	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	$\text{sn\_nodeid\_#\{index*4 + 1\}}$	Hashed target SN node ID # $\{\text{index*4} + 1\}$	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	$\text{sn\_nodeid\_#\{index*4\}}$	Hashed target SN node ID # $\{\text{index*4}\}$	RW	0b000000000000

### 8.3.17.61 sys\_cache\_grp\_hashed\_regions\_cxg\_sa\_nodeid\_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures SN node IDs for SCG's Hashed groups in the HNSAM . Controls target SN node IDs # $\{\text{index*4}\}$  to # $\{\text{index*4} + 3\}$ .

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

$\text{index}(0-3) : 0x6800 + \#\{8 * \text{index}\}$

##### Type

RW

##### Reset value

See individual bit resets

## Root group override

por\_rnsam\_rcr.mem\_range

## Secure group override

por\_rnsam\_scr.mem\_range

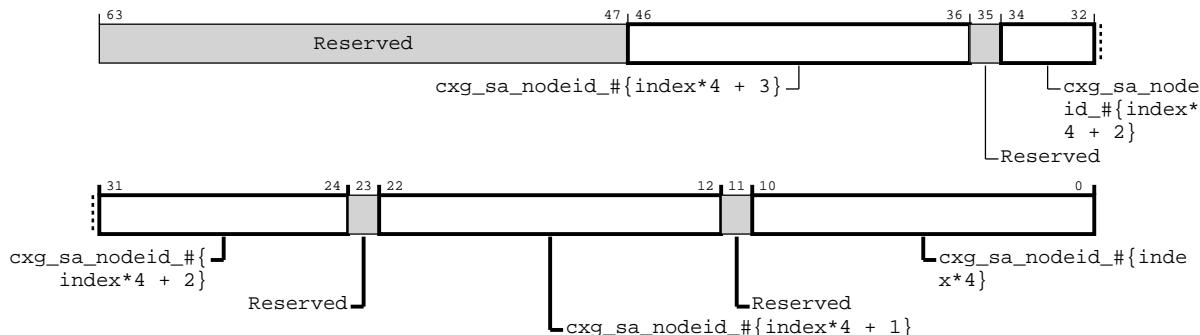
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1012: sys\_cache\_grp\_hashed\_regions\_cxg\_sa\_nodeid\_reg0-3**



**Table 8-1026: sys\_cache\_grp\_hashed\_regions\_cxg\_sa\_nodeid\_reg0-3 attributes**

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	cxg_sa_nodeid_{index*4 + 3}	Hashed target CXG SA node ID #{index*4 + 3}	RW	0b000000000000
[35]	Reserved	Reserved	RO	-
[34:24]	cxg_sa_nodeid_{index*4 + 2}	Hashed target CXG SA node ID #{index*4 + 2}	RW	0b000000000000
[23]	Reserved	Reserved	RO	-
[22:12]	cxg_sa_nodeid_{index*4 + 1}	Hashed target CXG SA node ID #{index*4 + 1}	RW	0b000000000000
[11]	Reserved	Reserved	RO	-
[10:0]	cxg_sa_nodeid_{index*4}	Hashed target CXG SA node ID #{index*4}	RW	0b000000000000

### 8.3.17.62 rnsam\_status

Functions as the default and programming mode status register.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x1100

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

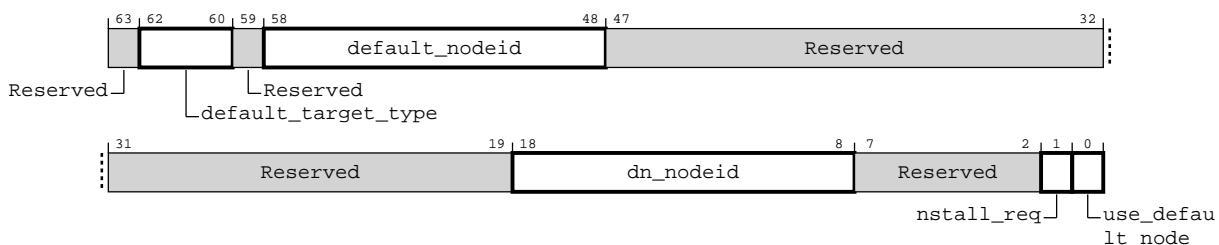
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1013: rnsam\_status**



**Table 8-1027: rnsam\_status attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:60]	default_target_type	Indicates node type <b>0b000</b> HN-F <b>0b001</b> HN-I <b>0b010</b> CXRA <b>0b011</b> HN-P <b>0b100</b> PCI-CXRA <b>0b101</b> HN-S <b>Others</b> Reserved <b>CONSTRAINT</b> Only applicable for RN-I	RW	0b001
[59]	Reserved	Reserved	RO	-
[58:48]	default_nodeid	Default Node ID	RW	0x000
[47:19]	Reserved	Reserved	RO	-
[18:8]	dn_nodeid	DN Node ID for DN operations	RW	0b0
[7:2]	Reserved	Reserved	RO	-
[1]	nstall_req	Indicates RN SAM is programmed and ready <b>0b0</b> STALL requests <b>0b1</b> UNSTALL requests	RW	0b0
[0]	use_default_node	Indicates target ID selection mode <b>0b0</b> Enables RN SAM to hash address bits and generate target ID <b>0b1</b> Uses default target ID	RW	0b1

### 8.3.17.63 gic\_mem\_region\_reg

Configures GIC memory region.

#### Configurations

This register is available in all configurations.

## Attributes

### Width

64

### Address offset

0x1108

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

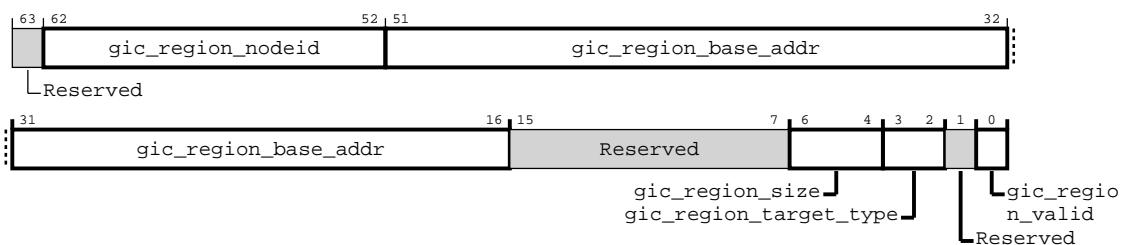
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1014: gic\_mem\_region\_reg**



**Table 8-1028: gic\_mem\_region\_reg attributes**

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:52]	gic_region_nodeid	GIC node ID	RW	0b000000000000
[51:16]	gic_region_base_addr	Base address of the GIC memory region <b>CONSTRAINT</b> Must be an integer multiple of region size	RW	0x0000000000
[15:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6:4]	gic_region_size	<p>GIC memory region size</p> <p><b>0b000</b> 64KB</p> <p><b>0b001</b> 128KB</p> <p><b>0b010</b> 256KB</p> <p><b>0b011</b> 512KB</p> <p><b>CONSTRAINT</b> Memory region must be a power of 2.</p>	RW	0b000
[3:2]	gic_region_target_type	<p>Indicates node type</p> <p><b>0b00</b> HN-F</p> <p><b>0b01</b> HN-I</p> <p><b>0b10</b> CXRA</p> <p><b>0b11</b> HN-P</p> <p><b>CONSTRAINT</b> Only applicable for RN-I</p>	RW	0b00
[1]	Reserved	Reserved	RO	-
[0]	gic_region_valid	<p>Memory region 1 valid</p> <p><b>0b0</b> Not valid</p> <p><b>0b1</b> Valid for memory region comparison</p>	RW	0b0

### 8.3.17.64 dsu\_hni\_region\_reg

Configures DSU\_HNI region.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

### Address offset

0x1110

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_rnsam\_rcr.mem\_range

### Secure group override

por\_rnsam\_scr.mem\_range

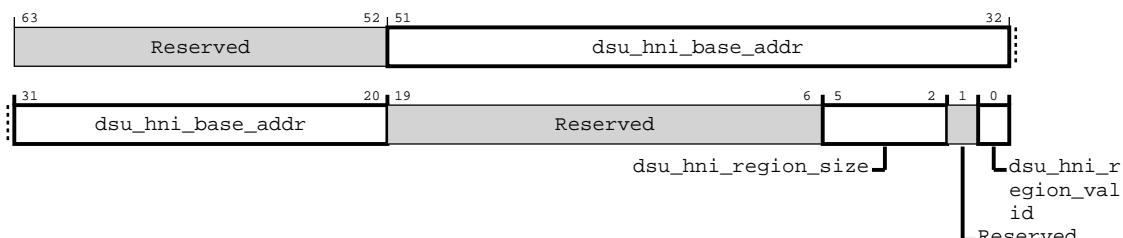
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1015: dsu\_hni\_region\_reg**



**Table 8-1029: dsu\_hni\_region\_reg attributes**

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	
[51:20]	dsu_hni_base_addr	Defines base address of the DSU HNI region	RW	0b0
[19:6]	Reserved	Reserved	RO	
[5:2]	dsu_hni_region_size	Defines the size of the each HNI region. total DSU_HNI region size is defined by num_HNI x size of each HNI region 0x0 - 1MB 0x1 - 2MB 0x2 - 4MB 0x3 - 8MB 0x4 - 16MB 0x5 - 32MB 0x6 - 64MB others - Reserved	RW	0b0
[1]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[0]	dsu_hni_region_valid	DSU_HNI region valid  0b0 DSU HNI region not enabled  0b1 DSU HNI region enabled	RW	0b0

### 8.3.17.65 sam\_generic\_regs0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configuration register for the custom logic

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

index(0-7) : 0x1600 + #{8 \* index}

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_rnsam\_rcr.mem\_range

##### Secure group override

por\_rnsam\_scr.mem\_range

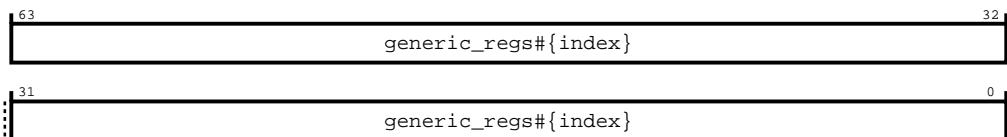
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_rnsam\_scr.mem\_range bit is set, Secure accesses to this register are permitted. If both the por\_rnsam\_scr.mem\_range bit and por\_rnsam\_rcr.mem\_range 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 8-1016: sam\_generic\_regs0-7**



**Table 8-1030: sam\_generic\_regs0-7 attributes**

Bits	Name	Description	Type	Reset
[63:0]	generic_regs#{index}	Configuration register for the custom logic	RW	0x0

### 8.3.18 SBSX register summary

The following table describes the registers for the relevant component.

**Table 8-1031: por\_sbsx\_cfg register summary**

Offset	Name	Type	Description
0x0	por_sbsx_node_info	RO	por_sbsx_node_info
0x80	por_sbsx_child_info	RO	por_sbsx_child_info
0x980	por_sbsx_rcr	RW	por_sbsx_rcr
0x988	por_sbsx_scr	RW	por_sbsx_scr
0x900	por_sbsx_unit_info	RO	por_sbsx_unit_info
0xA00	por_sbsx_cfg_ctl	RW	por_sbsx_cfg_ctl
0xA08	por_sbsx_aux_ctl	RW	por_sbsx_aux_ctl
0xA10	por_sbsx_datasource_ctl	RW	por_sbsx_datasource_ctl
0xA18	por_sbsx_cbusy_limit_ctl	RW	por_sbsx_cbusy_limit_ctl
0xE000	por_sbsx_errfr	RO	por_sbsx_errfr
0xE008	por_sbsx_errctlr	RW	por_sbsx_errctlr
0xE010	por_sbsx_errstatus	W1C	por_sbsx_errstatus
0xE018	por_sbsx_erraddr	RW	por_sbsx_erraddr
0xE028	por_sbsx_errmisc1	RW	por_sbsx_errmisc1
0xE800	por_sbsx_errpfgf	RO	por_sbsx_errpfgf
0xE808	por_sbsx_errpfgctl	RW	por_sbsx_errpfgctl
0xE810	por_sbsx_errpfgcdn	RW	por_sbsx_errpfgcdn
0xE040	por_sbsx_errfr_NS	RO	por_sbsx_errfr_NS
0xE048	por_sbsx_errctlr_NS	RW	por_sbsx_errctlr_NS
0xE050	por_sbsx_errstatus_NS	W1C	por_sbsx_errstatus_NS
0xE058	por_sbsx_erraddr_NS	RW	por_sbsx_erraddr_NS
0xE068	por_sbsx_errmisc1_NS	RW	por_sbsx_errmisc1_NS
0xE840	por_sbsx_errpfgf_NS	RO	por_sbsx_errpfgf_NS
0xE848	por_sbsx_errpfgctl_NS	RW	por_sbsx_errpfgctl_NS

Offset	Name	Type	Description
0xE850	por_sbsx_errpfgcdn_NS	RW	por_sbsx_errpfgcdn_NS
0xED00	por_sbsx_errcapctl	RW	por_sbsx_errcapctl
0xEE00	por_sbsx_errgsr	RO	por_sbsx_errgsr
0xEE10	por_sbsx_erriidr	RO	por_sbsx_erriidr
0xEFA8	por_sbsx_errdevaff	RO	por_sbsx_errdevaff
0xEF88	por_sbsx_errdevarch	RO	por_sbsx_errdevarch
0xEFC8	por_sbsx_errdevid	RO	por_sbsx_errdevid
0xEF00	por_sbsx_errpidr45	RO	por_sbsx_errpidr45
0xEFE0	por_sbsx_errpidr01	RO	por_sbsx_errpidr01
0xEFE8	por_sbsx_errpidr23	RO	por_sbsx_errpidr23
0EFF0	por_sbsx_errcidr01	RO	por_sbsx_errcidr01
0EFF8	por_sbsx_errcidr23	RO	por_sbsx_errcidr23
0xD900	por_sbsx_pmu_event_sel	RW	por_sbsx_pmu_event_sel

### 8.3.18.1 por\_sbsx\_node\_info

Provides component identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x0

##### Type

RO

##### Reset value

See individual bit resets

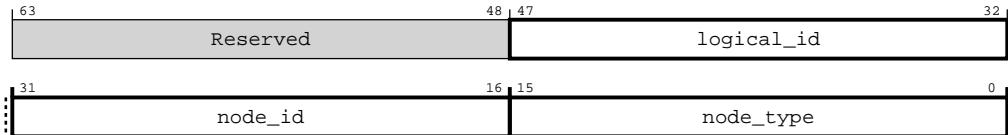
##### Usage constraints

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 8-1017: por\_sbsx\_node\_info**



**Table 8-1032: por\_sbsx\_node\_info attributes**

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	0x00
[31:16]	node_id	Component node ID	RO	0x00
[15:0]	node_type	CMN node type identifier	RO	0x0007

### 8.3.18.2 por\_sbsx\_child\_info

Provides component child identification information.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x80

##### Type

RO

##### Reset value

See individual bit resets

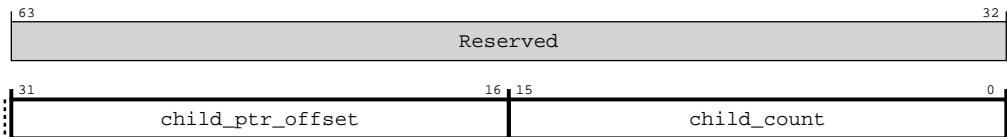
#### Usage constraints

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 8-1018: por\_sbsx\_child\_info**



**Table 8-1033: por\_sbsx\_child\_info attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	0x0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	0b0

### 8.3.18.3 por\_sbsx\_rcr

Root register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x980

##### Type

RW

##### Reset value

See individual bit resets

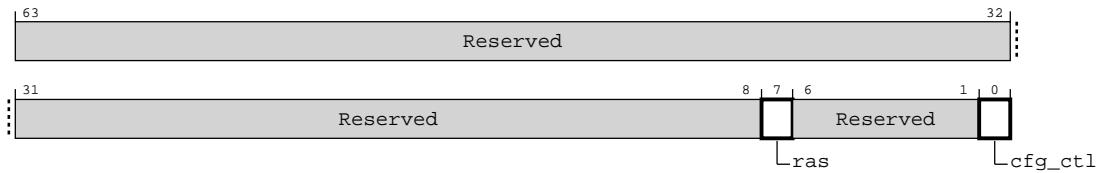
##### Usage constraints

This register is owned in the Root space and is 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 8-1019: por\_sbsx\_rcr**



**Table 8-1034: por\_sbsx\_rcr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allow Root override of the RAS registers	RW	0b0
[6:1]	Reserved	Reserved	RO	
[0]	cfg_ctl	Allows Root override of the configuration control register	RW	0b0

### 8.3.18.4 por\_sbsx\_scr

Secure register access override.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x988

##### Type

RW

##### Reset value

See individual bit resets

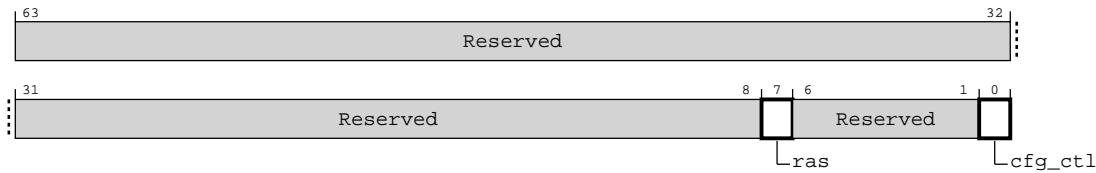
#### Usage constraints

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 8-1020: por\_sbsx\_scr**



**Table 8-1035: por\_sbsx\_scr attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7]	ras	Allow Secure override of the RAS registers	RW	0b0
[6:1]	Reserved	Reserved	RO	
[0]	cfg_ctl	Allows Secure override of the configuration control register	RW	0b0

### 8.3.18.5 por\_sbsx\_unit\_info

Provides component identification information for SBSX.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0x900

##### Type

RO

##### Reset value

See individual bit resets

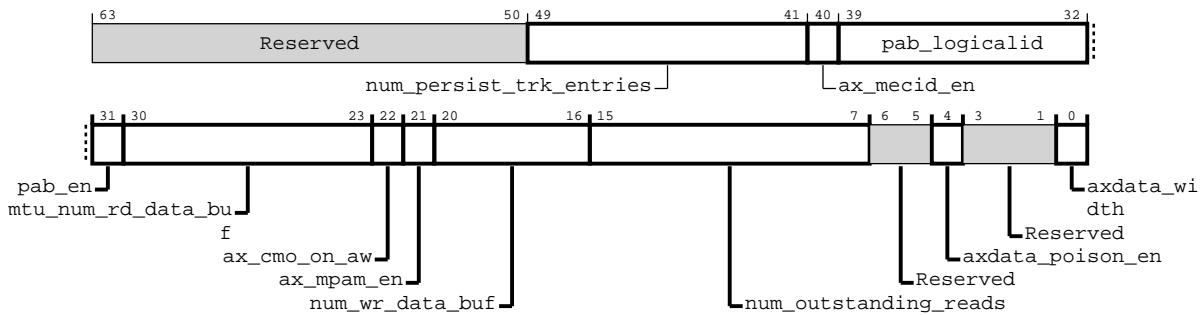
##### Usage constraints

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 8-1021: por\_sbsx\_unit\_info**



**Table 8-1036: por\_sbsx\_unit\_info attributes**

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:41]	num_persist_trk_entries	Number of entries in persist/tagmatch tracker. Size 0 indicates tracker is disabled.	RO	Configuration dependent
[40]	ax_mecid_en	MECID enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[39:32]	pab_logicalid	PUB AUB bridge Logical ID	RO	0x20
[31]	pab_en	PUB AUB bridge enable	RO	0x0
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[30:23]	mtu_num_rd_data_buf	Number of mtu read data buffers in SBSX	RO	Configuration dependent
[22]	ax_cmo_on_aw	Write Channel CMOs enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[21]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface	RO	Configuration dependent
	<b>0b1</b>	Enabled		
	<b>0b0</b>	Not enabled		
[20:16]	num_wr_data_buf	Number of write data buffers in SBSX	RO	Configuration dependent
[15:7]	num_outstanding_reads	Maximum number of outstanding AXI read requests from SBSX	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[6:5]	Reserved	Reserved	RO	-
[4]	axdata_poison_en	Data poison support on ACE-Lite/AXI4 interface  <b>0b0</b> Not supported  <b>0b1</b> Supported	RO	Configuration dependent
[3:1]	Reserved	Reserved	RO	-
[0]	axdata_width	Data width on ACE-Lite/AXI4 interface  <b>0b0</b> 128 bits  <b>0b1</b> 256 bits	RO	0x1

### 8.3.18.6 por\_sbsx\_cfg\_ctl

Functions as the configuration control register for SBSX bridge.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA00

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_sbsx\_rcr.cfg\_ctl

##### Secure group override

por\_sbsx\_scr.cfg\_ctl

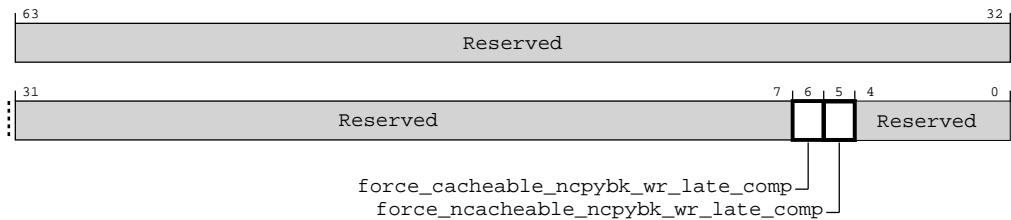
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.cfg\_ctl bit and por\_sbsx\_rcr.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 8-1022: por\_sbsx\_cfg\_ctl**



**Table 8-1037: por\_sbsx\_cfg\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	
[6]	force_cacheable_ncpybk_wr_late_comp	Late Comp for Cacheable Non-CopyBack Writes. Overrides EWA. Not applicable to Writes where DoDWT is set.	RW	0b0
[5]	force_ncacheable_ncpybk_wr_late_comp	Late Comp for Non-cacheable Non-CopyBack Writes. Overrides EWA. Not applicable to Writes where DoDWT is set.	RW	0b0
[4:0]	Reserved	Reserved	RO	

### 8.3.18.7 por\_sbsx\_aux\_ctl

Functions as the auxiliary control register for the SBSX bridge.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA08

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.cfg\_ctl

### Secure group override

por\_sbsx\_scr.cfg\_ctl

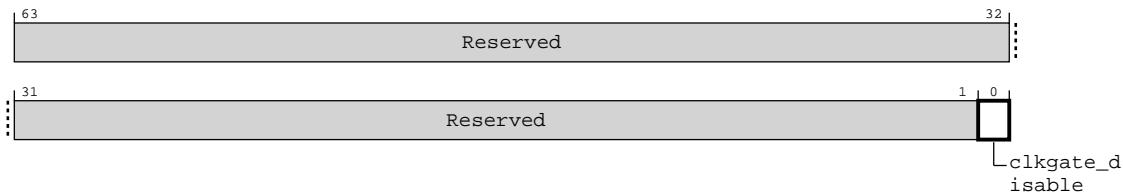
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.cfg\_ctl bit and por\_sbsx\_rcr.cfg\_ctl bit are set, Non-secure and Realm accesses to this register are permitted. This register can be modified only with prior written permission from Arm.

### Bit descriptions

The following image shows the higher register bit assignments.

**Figure 8-1023: por\_sbsx\_aux\_ctl**



**Table 8-1038: por\_sbsx\_aux\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	
[0]	clkgate_disable	Disables internal clock gating in SBSX bridge	RW	0b0

### 8.3.18.8 por\_sbsx\_datasource\_ctl

Functions as the DataSource control register to determine how to drive the CHI DAT.DataSource field.

### Configurations

This register is available in all configurations.

### Attributes

#### Width

64

#### Address offset

0xA10

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_sbsx\_rcr.cfg\_ctl

## Secure group override

por\_sbsx\_scr.cfg\_ctl

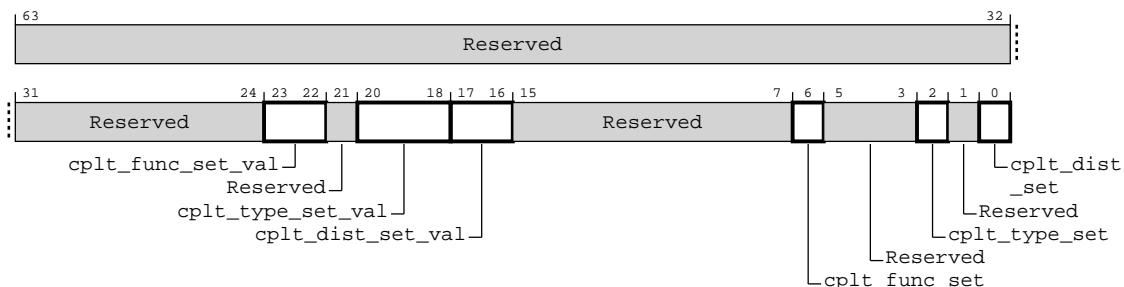
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.cfg\_ctl bit and por\_sbsx\_rcr.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 8-1024: por\_sbsx\_datasource\_ctl**



**Table 8-1039: por\_sbsx\_datasource\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:22]	cplt_func_set_val	Completer Functional value to use when cplt_func_set is set to 1. Supported values are <b>0b00</b> PrefetchTgt not useful <b>0b01</b> PrefetchTgt useful	RW	0b01
[21]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[20:18]	cplt_type_set_val	Completer Type value to use when cplt_type_set is set to 1. Supported values are <b>0b001</b> DRAM Completer Type <b>0b011</b> High Bandwidth Memory (HBM) Completer Type	RW	0b001
[17:16]	cplt_dist_set_val	Completer Distance value to use when cplt_dist_set is set to 1.	RW	0b00
[15:7]	Reserved	Reserved	RO	-
[6]	cplt_func_set	Completer Functional Set <b>0b0</b> Pass Completer Functional field as-is when sending CompData <b>0b1</b> Set Completer Functional field to cplt_func_set_val when sending CompData	RW	0b0
[5:3]	Reserved	Reserved	RO	-
[2]	cplt_type_set	Completer Type Set <b>0b0</b> Reserved <b>0b1</b> Set Completer Type field to cplt_type_set_val when sending CompData	RW	0b1
[1]	Reserved	Reserved	RO	-
[0]	cplt_dist_set	Completer Distance Set <b>0b0</b> Reserved <b>0b1</b> Set Completer Distance field to cplt_dist_set_val when sending CompData	RW	0b1

### 8.3.18.9 por\_sbsx\_cbusy\_limit\_ctl

Cbusy threshold limits for Request Tracker entries.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xA18

##### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.cfg\_ctl

### Secure group override

por\_sbsx\_scr.cfg\_ctl

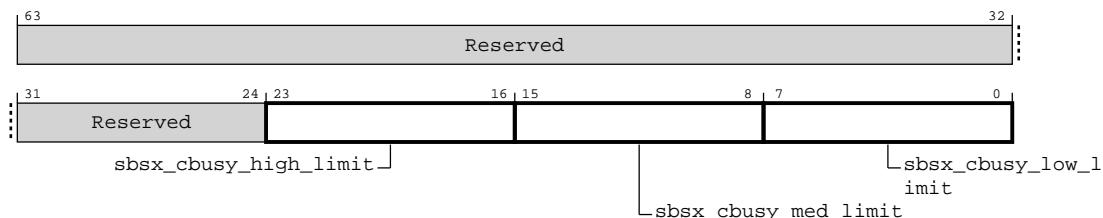
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.cfg\_ctl bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.cfg\_ctl bit and por\_sbsx\_rcr.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 8-1025: por\_sbsx\_cbusy\_limit\_ctl**



**Table 8-1040: por\_sbsx\_cbusy\_limit\_ctl attributes**

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	sbsx_cbusy_high_limit	ReqTracker limit for CBusy High	RW	0x48
[15:8]	sbsx_cbusy_med_limit	ReqTracker limit for CBusy Med	RW	0x30
[7:0]	sbsx_cbusy_low_limit	ReqTracker limit for CBusy Low	RW	0x18

### 8.3.18.10 por\_sbsx\_errfr

Functions as the error feature register.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE000

### Type

RO

### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.ras

### Secure group override

por\_sbsx\_scr.ras

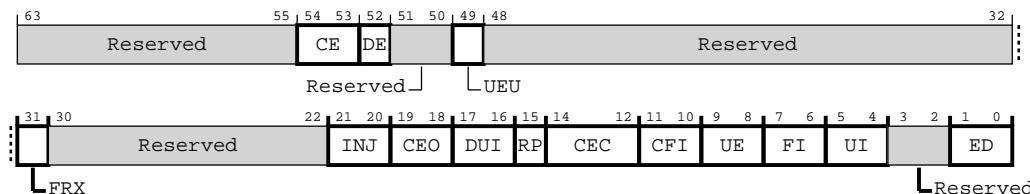
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1026: por\_sbsx\_errfr**



**Table 8-1041: por\_sbsx\_errfr attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording  <b>0b00</b> Corrected Error not supported  <b>0b10</b> Non-specific Corrected Error supported	RO	0b00

Bits	Name	Description	Type	Reset
[52]	DE	Deferred Error recording  <b>0b0</b> Deferred Error not supported  <b>0b1</b> Deffered Error supported	RO	0b0
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording  <b>0b0</b> Unrecoverable Error not supported  <b>0b1</b> Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension.  <b>0b1</b> por_sbsx_errfr[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension.  <b>0b01</b> Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite.  <b>0b00</b> Keep the first Corrected Error syndrome	RO	0b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_sbsx_errctlr.DUI.	RO	0b00
[15]	RP	Repeat counter (valid only when por_sbsx_errfr.CEC != 0b000.)  <b>0b0</b> Invalid;  <b>0b1</b> Implements a first (repeat) counter and a second (other) counter	RO	0b0
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter	RO	0b000

Bits	Name	Description	Type	Reset
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_sbsx_errctlr.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_sbsx_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_sbsx_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_sbsx_errctlr.ED	RO	0b10

### 8.3.18.11 por\_sbsx\_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE008

##### Type

RW

##### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.ras

### Secure group override

por\_sbsx\_scr.ras

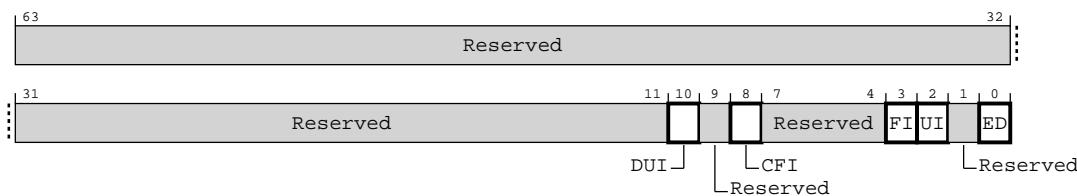
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1027: por\_sbsx\_errctlr**



**Table 8-1042: por\_sbsx\_errctlr attributes**

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	
[10]	DUI	Enables error recovery interrupt for deferred error as specified in por_sbsx_errfr.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_sbsx_errfr.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_sbsx_errfr.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_sbsx_errfr.UI	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_sbsx_errfr.ED	RW	0b0

### 8.3.18.12 por\_sbsx\_errstatus

Functions as the error status register. When V is set, only write exact same value as in the register can clear it.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE010

### Type

W1C

### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.ras

### Secure group override

por\_sbsx\_scr.ras

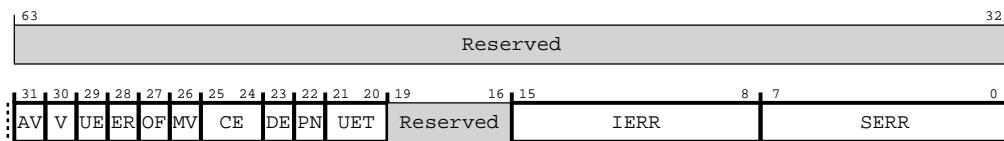
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1028: por\_sbsx\_errstatus**



**Table 8-1043: por\_sbsx\_errstatus attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_sbsx_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0

Bits	Name	Description	Type	Reset
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_sbsx_errmisc<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b1</b> At least one error is not corrected and is deferred <b>0b0</b> No errors deferred	W1C	0b0
[22]	PN	Poison <b>0b1</b> Uncorrected error recorded because a poison value was consumed <b>0b0</b> Other cases	W1C	0b0

Bits	Name	Description	Type	Reset
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_sbsx_errmisc1.ERRSRC for error type details.	W1C	0b0

### 8.3.18.13 por\_sbsx\_erraddr

Contains the error record address.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE018

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_sbsx\_rcr.ras

##### Secure group override

por\_sbsx\_scr.ras

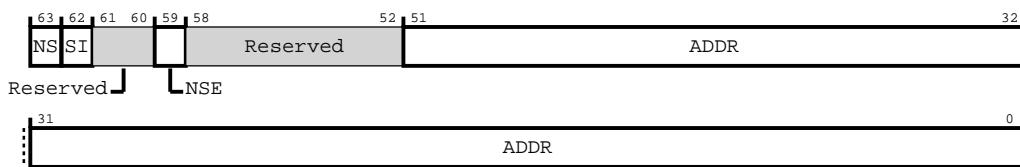
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1029: por\_sbsx\_erraddr**



**Table 8-1044: por\_sbsx\_erraddr attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.18.14 por\_sbsx\_errmisc1

Functions as the miscellaneous error register 1. Contains miscellaneous information about deferred/uncorrected errors.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE028

### Type

RW

### Reset value

See individual bit resets

### Root group override

por\_sbsx\_rcr.ras

### Secure group override

por\_sbsx\_scr.ras

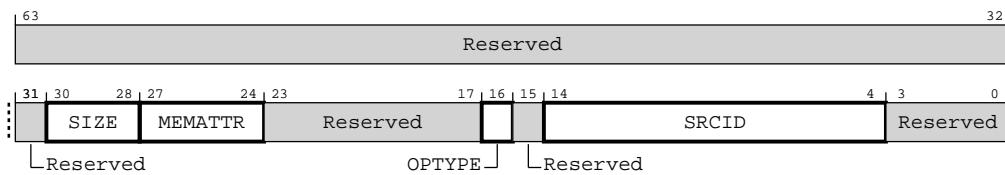
### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1030: por\_sbsx\_errmisc1**



**Table 8-1045: por\_sbsx\_errmisc1 attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	
[30:28]	SIZE	Error transaction size	RW	0b0
[27:24]	MEMATTR	Error memory attributes	RW	0b0
[23:17]	Reserved	Reserved	RO	
[16]	OPTYPE	Error opcode type  <b>0b1</b> WR_NO_SNP_PTL (partial)  <b>0b0</b> WR_NO_SNP_FULL	RW	0b0
[15]	Reserved	Reserved	RO	
[14:4]	SRCID	Error source ID	RW	0b0
[3:0]	Reserved	Reserved	RO	

### 8.3.18.15 por\_sbsx\_errpfgf

Functions as the Pseudo-fault Generation Feature Register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE800

##### Type

RO

##### Reset value

See individual bit resets

##### Root group override

por\_sbsx\_rcr.ras

##### Secure group override

por\_sbsx\_scr.ras

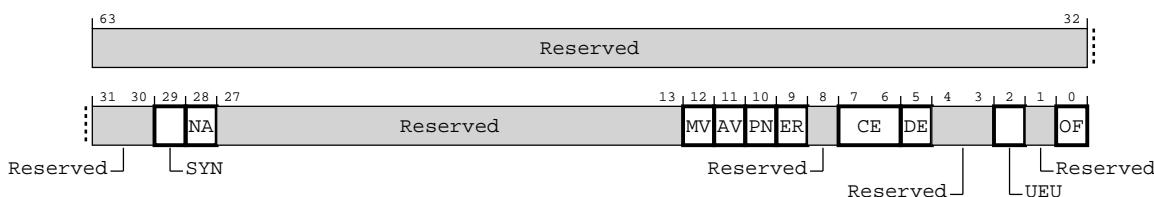
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1031: por\_sbsx\_errpfgf**



**Table 8-1046: por\_sbsx\_errpfgf attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[29]	SYN	Fault syndrome injection.  <b>0b1</b> Fault injection does not update ERRSTATUS.SERR	RO	0b1
[28]	NA	No access required.  <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	0b1
[11]	AV	Address syndrome.  <b>0b1</b> Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	0b1
[10]	PN	Poison flag  <b>0b1</b> Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	0b1
[9]	ER	Error reported flag  <b>0b1</b> Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> No Corrected error generation.  <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 0b10; else, update ERRSTATUS.CE to 0b00	RO	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> No Deferred error generation.  <b>0b1</b> Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b1</b> If ERRPFGCTL.UEU == 1, update ERRSTATUS.UET to 0b1 and ERRSTATUS.UET = 0b01; else, update ERRSTATUS.UET to 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b1</b> Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	0b1

### 8.3.18.16 por\_sbsx\_errpfctl

Functions as the Pseudo-fault Generation Control Register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE808

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_sbsx\_rcr.ras

##### Secure group override

por\_sbsx\_scr.ras

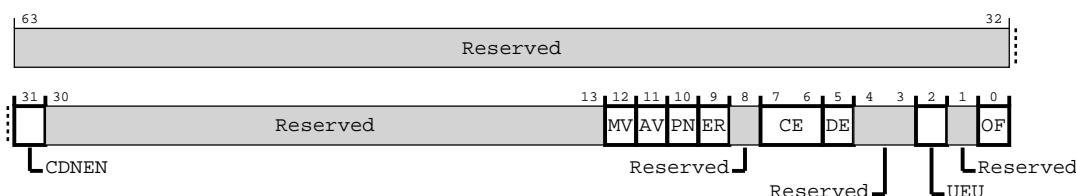
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1032: por\_sbsx\_errpfctl**



**Table 8-1047: por\_sbsx\_errpfctl attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	CDNEN	<p>Countdown Enable.</p> <p><b>0b0</b> Countdown disabled.</p> <p><b>0b1</b> Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.</p>	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	<p>Miscellaneous syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.MV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.MV to 0b1</p>	RW	0b0
[11]	AV	<p>Address syndrome.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.AV to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.AV to 0b1</p>	RW	0b0
[10]	PN	<p>Poison flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.PN to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.PN to 0b1</p>	RW	0b0
[9]	ER	<p>Error reported flag</p> <p><b>0b0</b> Fault injection update ERRSTATUS.ER to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.ER to 0b1</p>	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	<p>Corrected Error generation.</p> <p><b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 0b00</p> <p><b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 0b10</p>	RW	0b00
[5]	DE	<p>Deferred error generation.</p> <p><b>0b0</b> Fault injection update ERRSTATUS.DE to 0b0</p> <p><b>0b1</b> Fault injection update ERRSTATUS.DE to 0b1</p>	RW	0b0
[4:3]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update ERRSTATUS.UE to 0b0 and ERRSTATUS.UET = 0b00  <b>0b1</b> Fault injection update ERRSTATUS.UE to 0b1 and ERRSTATUS.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update ERRSTATUS.OF to 0b0  <b>0b1</b> Fault injection update ERRSTATUS.OF to 0b1	RW	0b0

### 8.3.18.17 por\_sbsx\_errpfgcdn

Functions as the Pseudo-fault Generation Countdown Register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE810

##### Type

RW

##### Reset value

See individual bit resets

##### Root group override

por\_sbsx\_rcr.ras

##### Secure group override

por\_sbsx\_scr.ras

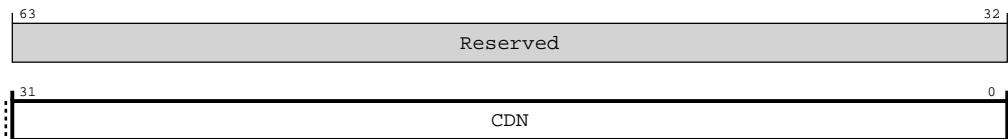
#### Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1033: por\_sbsx\_errpfgcdn**



**Table 8-1048: por\_sbsx\_errpfgcdn attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

### 8.3.18.18 por\_sbsx\_errfr\_NS

Functions as the non-secure error feature register.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE040

### Type

RO

### Reset value

See individual bit resets

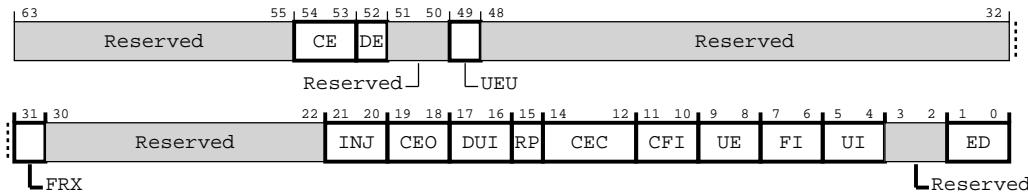
## Usage constraints

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 8-1034: por\_sbsx\_errfr\_NS**



**Table 8-1049: por\_sbsx\_errfr\_NS attributes**

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:52]	CE	Corrected Error recording <b>0b00</b> : Corrected Error not supported <b>0b10</b> : Non-specific Corrected Error supported	RO	0b00
[51:50]	DE	Deferred Error recording <b>0b0</b> : Deferred Error not supported <b>0b1</b> : Deffered Error supported	RO	0b0
[49]	UEU	Unrecoverable Error recording <b>0b0</b> : Unrecoverable Error not supported <b>0b1</b> : Unrecoverable Error supported	RO	0b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. <b>0b1</b> : por_sbsx_errfr_NS[63:48] is architecturally defined	RO	0b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. <b>0b1</b> : Support Common Fault Injection Model Extension	RO	0b01
[19:18]	CEO	Corrected Error overwrite. <b>0b00</b> : Keep the first Corrected Error syndrome	RO	0b00

Bits	Name	Description	Type	Reset
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Deferred errors  <b>0b10</b> Support Error Recovery Interrupt from Deferred errors and controllable using por_sbsx_errctlr.DUI.	RO	0b00
[15]	RP	Repeat counter (valid only when por_sbsx_errfr_NS.CEC != 0b000.)  <b>0b1</b> Implements a first (repeat) counter and a second (other)	RO	0b0
[14:12]	CEC	Standard corrected error counter  <b>0b000</b> Does not implement standard error counter model  <b>0b100</b> Implements a 16-bit Corrected error counter	RO	0b000
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Corrected errors  <b>0b10</b> Support Fault Handling Interrupt on corrected errors and controllable using por_sbsx_errctlr.CFI.	RO	0b00
[9:8]	UE	In-band error response is always on	RO	0b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control  <b>0b00</b> Does not support Fault Handling Interrupt from Deferred and Uncorrected errors  <b>0b10</b> Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using por_sbsx_errctlr.FI.	RO	0b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control  <b>0b00</b> Does not support Error Recovery Interrupt from Uncorrected errors  <b>0b10</b> Support Error Recovery Interrupt on Uncorrected errors and controllable using por_sbsx_errctlr.UI.	RO	0b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control  <b>0b10</b> Error reporting and logging is controllable using por_sbsx_errctlr.ED	RO	0b10

### 8.3.18.19 por\_sbsx\_errctlr\_NS

Functions as the non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE048

##### Type

RW

##### Reset value

See individual bit resets

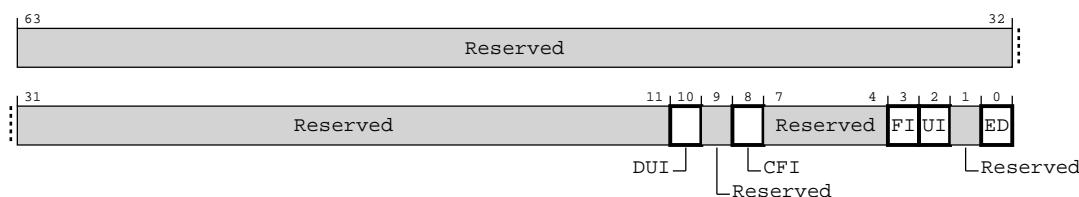
##### Usage constraints

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 8-1035: por\_sbsx\_errctlr\_NS**



**Table 8-1050: por\_sbsx\_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 por_sbsx_errfr_NS.DUI	RW	0b0
[9]	Reserved	Reserved	RO	
[8]	CFI	Enables fault handling interrupt for corrected error as specified in por_sbsx_errfr_NS.CFI	RW	0b0
[7:4]	Reserved	Reserved	RO	
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in por_sbsx_errfr_NS.FI	RW	0b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in por_sbsx_errfr_NS.UI	RW	0b0

Bits	Name	Description	Type	Reset
[1]	Reserved	Reserved	RO	
[0]	ED	Enables error detection as specified in por_sbsx_errfr_NS.ED	RW	0b0

### 8.3.18.20 por\_sbsx\_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

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE050

##### Type

W1C

##### Reset value

See individual bit resets

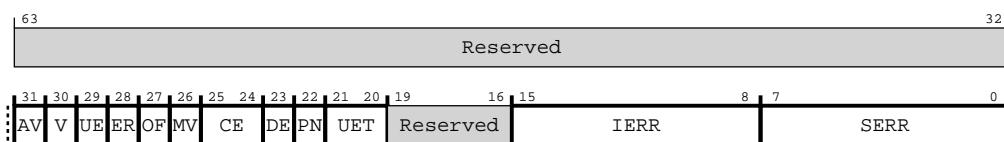
##### Usage constraints

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 8-1036: por\_sbsx\_errstatus\_NS**



**Table 8-1051: por\_sbsx\_errstatus\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[31]	AV	Address register valid <b>0b1</b> Address is valid; por_sbsx_erraddr contains a physical address for that recorded error <b>0b0</b> Address is not valid	W1C	0b0
[30]	V	Status register valid <b>0b1</b> At least one error recorded; register is valid <b>0b0</b> No errors recorded	W1C	0b0
[29]	UE	Uncorrected errors <b>0b1</b> At least one error detected that is not corrected and is not deferred to a subordinate <b>0b0</b> No uncorrected errors detected	W1C	0b0
[28]	ER	Error Reported <b>0b1</b> In-band error response signaled to the Requester <b>0b0</b> No in-band error response signaled	W1C	0b0
[27]	OF	Overflow; asserted when multiple errors are detected <b>0b1</b> More than one error detected <b>0b0</b> None or only one error detected as described by UE/DE/CE fields	W1C	0b0
[26]	MV	por_sbsx_errmisc<01> valid <b>0b1</b> Miscellaneous registers are valid <b>0b0</b> Miscellaneous registers are not valid	W1C	0b0
[25:24]	CE	Corrected errors <b>0b10</b> At least one corrected error recorded <b>0b00</b> No corrected errors recorded	W1C	0b00
[23]	DE	Deferred errors <b>0b1</b> At least one error is not corrected and is deferred <b>0b0</b> No errors deferred	W1C	0b0

Bits	Name	Description	Type	Reset
[22]	PN	Poison  <b>0b1</b> Uncorrected error recorded because a poison value was consumed  <b>0b0</b> Other cases	W1C	0b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0  <b>0b01</b> Uncorrected error, Unrecoverable error (UEU).  <b>0b00</b> Invalid	W1C	0b00
[19:16]	Reserved	Reserved	RO	
[15:8]	IERR	Implementation-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> Partner implementation defined error	W1C	0b0
[7:0]	SERR	Architecturally-defined primary error code.  <b>0x00</b> No error  <b>0x01</b> <b>IMPLEMENTATION DEFINED</b> error. Refer to por_sbsx_errmisc1_NS.ERRSRC for error type details.	W1C	0b0

### 8.3.18.21 por\_sbsx\_erraddr\_NS

Contains the non-secure error record address.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE058

##### Type

RW

##### Reset value

See individual bit resets

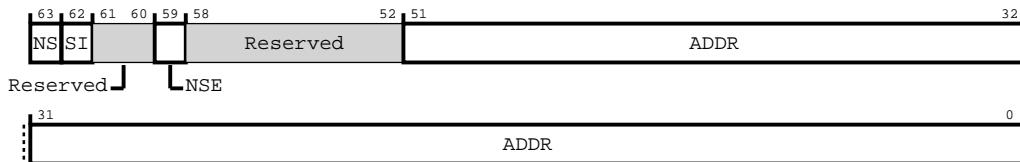
## Usage constraints

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 8-1037: por\_sbsx\_erraddr\_NS**



**Table 8-1052: por\_sbsx\_erraddr\_NS attributes**

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	0b0
[62]	SI	{NSE,NS} valid <b>0b0</b> PAS field is valid <b>0b1</b> PAS field is invalid	RW	0b0
[61:60]	Reserved	Reserved	RO	
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	0b0
[58:52]	Reserved	Reserved	RO	
[51:0]	ADDR	Transaction address	RW	0b0

### 8.3.18.22 por\_sbsx\_errmisc1\_NS

Functions as the non-secure miscellaneous error register 1. Contains miscellaneous information about deferred/uncorrected errors.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE068

## Type

RW

## Reset value

See individual bit resets

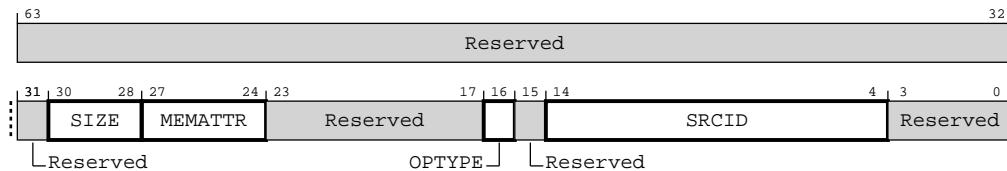
## Usage constraints

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 8-1038: por\_sbsx\_errmisc1\_NS**



**Table 8-1053: por\_sbsx\_errmisc1\_NS attributes**

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	
[30:28]	SIZE	Error transaction size	RW	0b0
[27:24]	MEMATTR	Error memory attributes	RW	0b0
[23:17]	Reserved	Reserved	RO	
[16]	OPTYPE	Error opcode type  <b>0b1</b> WR_NO_SNP_PTL (partial)  <b>0b0</b> WR_NO_SNP_FULL	RW	0b0
[15]	Reserved	Reserved	RO	
[14:4]	SRCID	Error source ID	RW	0b0
[3:0]	Reserved	Reserved	RO	

## 8.3.18.23 por\_sbsx\_errpfg\_NS

Functions as the non-secure Pseudo-fault Generation Feature Register.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE840

### Type

RO

### Reset value

See individual bit resets

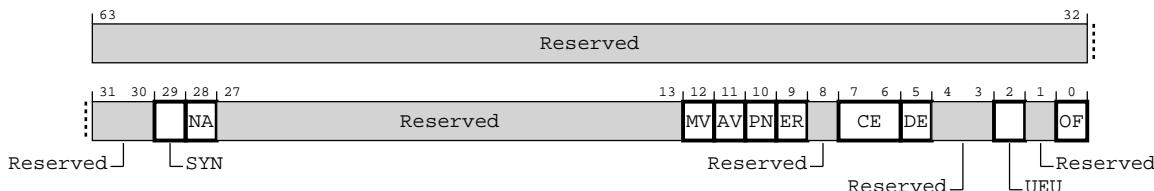
### Usage constraints

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 8-1039: por\_sbsx\_errpfgf\_NS**



**Table 8-1054: por\_sbsx\_errpfgf\_NS attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29]	SYN	Fault syndrome injection.  <b>0b1</b> Fault injection does not update ERRSTATUS_NS.SERR	RO	0b1
[28]	NA	No access required.  <b>0b1</b> Fault injection does not require cfg access	RO	0b1
[27:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome.  <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to ERRPFGCTL_NS.MV	RO	0b1
[11]	AV	Address syndrome.  <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to ERRPFGCTL_NS.AV	RO	0b1

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to ERRPFGCTL_NS.PN	RO	0b1
[9]	ER	Error reported flag <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to ERRPFGCTL_NS.ER	RO	0b1
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation. <b>0b00</b> No Corrected error generation. <b>0b01</b> Non specific Corrected error injection. If ERRPFGCTL_NS.CE == 1, update ERRSTATUS_NS.CE to 0b10; else, update ERRSTATUS_NS.CE to 0b00	RO	0b00
[5]	DE	Deferred error generation. <b>0b0</b> No Deferred error generation. <b>0b1</b> Fault injection update ERRSTATUS_NS.DE to ERRPFGCTL_NS.DE	RO	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation. <b>0b1</b> If ERRPFGCTL_NS.UEU == 1, update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01; else, update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00	RO	0b1
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag. <b>0b1</b> Fault injection update ERRSTATUS_NS.OF to ERRPFGCTL_NS.OF	RO	0b1

### 8.3.18.24 por\_sbsx\_errpfgctl\_NS

Functions as the non-secure Pseudo-fault Generation Control Register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xE848

## Type

RW

## Reset value

See individual bit resets

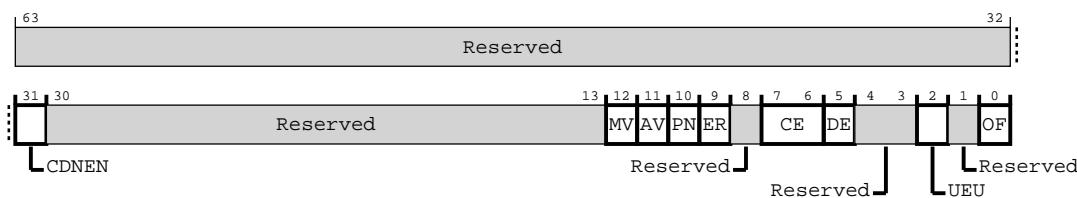
## Usage constraints

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 8-1040: por\_sbsx\_errpfgctl\_NS**



**Table 8-1055: por\_sbsx\_errpfgctl\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31]	CDNEN	Countdown Enable. <b>0b0</b> Countdown disabled. <b>0b1</b> Error generation counter is set to ERRPFGCDN_NS.CDN, and countdown enabled.	RW	0b0
[30:13]	Reserved	Reserved	RO	
[12]	MV	Miscellaneous syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.MV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.MV to 0b1	RW	0b0
[11]	AV	Address syndrome. <b>0b0</b> Fault injection update ERRSTATUS_NS.AV to 0b0 <b>0b1</b> Fault injection update ERRSTATUS_NS.AV to 0b1	RW	0b0

Bits	Name	Description	Type	Reset
[10]	PN	Poison flag  <b>0b0</b> Fault injection update ERRSTATUS_NS.PN to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.PN to 0b1	RW	0b0
[9]	ER	Error reported flag  <b>0b0</b> Fault injection update ERRSTATUS_NS.ER to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.ER to 0b1	RW	0b0
[8]	Reserved	Reserved	RO	
[7:6]	CE	Corrected Error generation.  <b>0b00</b> Non Corrected error is injected. Fault injection update ERRSTATUS_NS.CE to 0b00  <b>0b01</b> Non specific Corrected error injection. Fault injection update ERRSTATUS_NS.CE to 0b10	RW	0b00
[5]	DE	Deferred error generation.  <b>0b0</b> Fault injection update ERRSTATUS_NS.DE to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.DE to 0b1	RW	0b0
[4:3]	Reserved	Reserved	RO	
[2]	UEU	Uncorrected error generation.  <b>0b0</b> Fault injection update ERRSTATUS_NS.UE to 0b0 and ERRSTATUS_NS.UET = 0b00  <b>0b1</b> Fault injection update ERRSTATUS_NS.UE to 0b1 and ERRSTATUS_NS.UET = 0b01	RW	0b0
[1]	Reserved	Reserved	RO	
[0]	OF	Overflow flag.  <b>0b0</b> Fault injection update ERRSTATUS_NS.OF to 0b0  <b>0b1</b> Fault injection update ERRSTATUS_NS.OF to 0b1	RW	0b0

### 8.3.18.25 por\_sbsx\_errpfgcdn\_NS

Functions as the non-secure Pseudo-fault Generation Countdown Register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xE850

### Type

RW

### Reset value

See individual bit resets

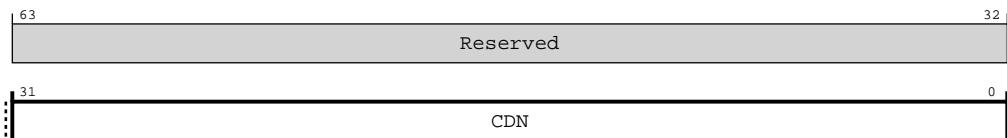
### Usage constraints

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 8-1041: por\_sbsx\_errpfgcdn\_NS**



**Table 8-1056: por\_sbsx\_errpfgcdn\_NS attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:0]	CDN	Countdown value	RW	0b0

## 8.3.18.26 por\_sbsx\_errcapctl

Functions as the error Capture Control Register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xED00

## Type

RW

## Reset value

See individual bit resets

## Root group override

por\_sbsx\_rcr.ras

## Secure group override

por\_sbsx\_scr.ras

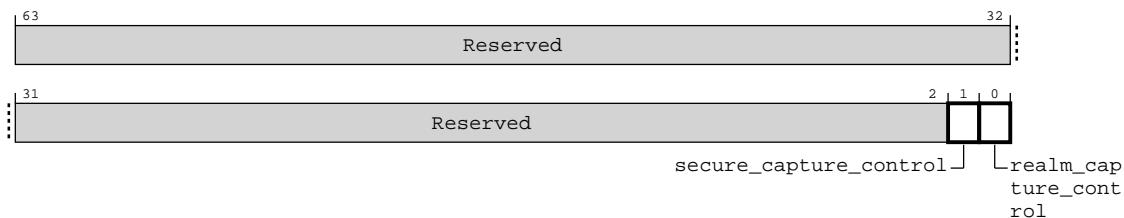
## Usage constraints

This register is owned in the Root space and is accessible using Root transactions, unless the following override options are set: If the por\_sbsx\_scr.ras bit is set, Secure accesses to this register are permitted. If both the por\_sbsx\_scr.ras bit and por\_sbsx\_rcr.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 8-1042: por\_sbsx\_errcapctl**



**Table 8-1057: por\_sbsx\_errcapctl attributes**

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	
[1]	secure_capture_control	Secure Capture Control <b>0b0</b> Transaction with secure PAS captured in root error record <b>0b1</b> Transaction with secure PAS captured in non-secure error record	RW	0b0
[0]	realm_capture_control	Realm Capture Control <b>0b0</b> Transaction with realm PAS captured in root error record <b>0b1</b> Transaction with realm PAS captured in non-secure error record	RW	0b0

### 8.3.18.27 por\_sbsx\_errgsr

Functions as Error Group Status Register

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEE00

##### Type

RO

##### Reset value

See individual bit resets

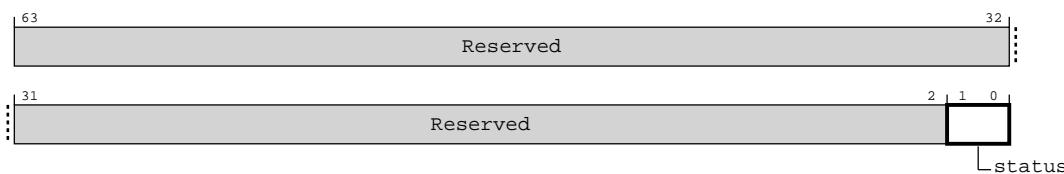
##### Usage constraints

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 8-1043: por\_sbsx\_errgsr**



**Table 8-1058: por\_sbsx\_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	0b0

### 8.3.18.28 por\_sbsx\_errildr

Functions as the implementation identification register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEE10

##### Type

RO

##### Reset value

See individual bit resets

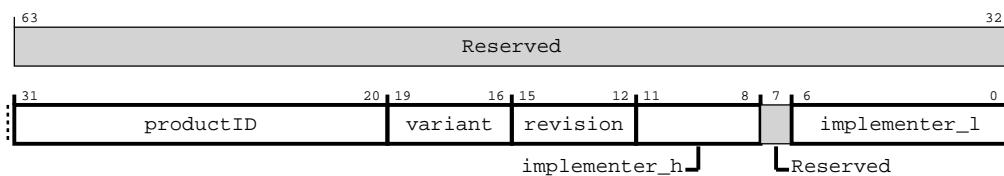
##### Usage constraints

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 8-1044: por\_sbsx\_errildr**



**Table 8-1059: por\_sbsx\_errildr attributes**

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	
[31:20]	productID	Product Part number	RO	0x0
[19:16]	variant	Component major revision	RO	0x0
[15:12]	revision	Component minor revision	RO	0x0
[11:8]	implementer_h	Implementer[10:7]	RO	0x4
[7]	Reserved	Reserved	RO	
[6:0]	implementer_l	Implementer[6:0]	RO	0x3B

### 8.3.18.29 por\_sbsx\_errdevaff

Functions as the device affinity register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

#### Attributes

##### Width

64

##### Address offset

0xEFA8

##### Type

RO

##### Reset value

See individual bit resets

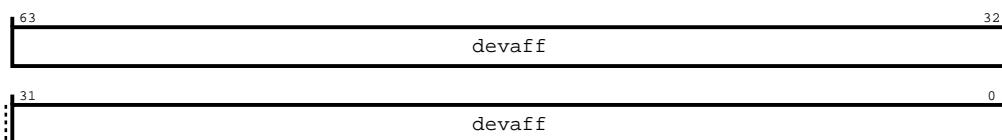
##### Usage constraints

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 8-1045: por\_sbsx\_errdevaff**



**Table 8-1060: por\_sbsx\_errdevaff attributes**

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	0b0

### 8.3.18.30 por\_sbsx\_errdevarch

Functions as the device architecture register.

#### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEF8

### Type

RO

### Reset value

See individual bit resets

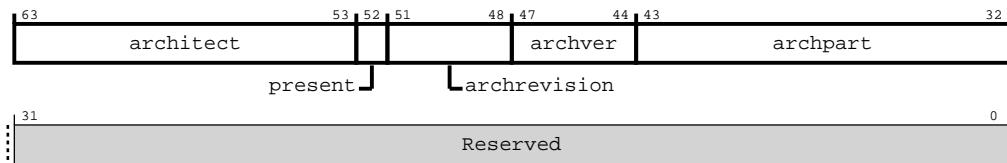
### Usage constraints

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 8-1046: por\_sbsx\_errdevarch**



**Table 8-1061: por\_sbsx\_errdevarch attributes**

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	0x23B
[52]	present	Present	RO	0b1
[51:48]	archrevision	Architecture revision	RO	0b1
[47:44]	archver	Architecture Version	RO	0x0
[43:32]	archpart	Architecture Part	RO	0xA00
[31:0]	Reserved	Reserved	RO	-

### 8.3.18.31 por\_sbsx\_errdevid

Functions as the device configuration register

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEFC8

### Type

RO

### Reset value

See individual bit resets

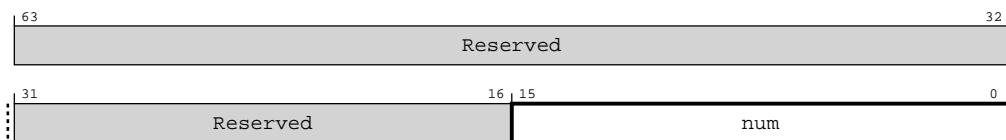
### Usage constraints

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 8-1047: por\_sbsx\_errdevid**



**Table 8-1062: por\_sbsx\_errdevid attributes**

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	
[15:0]	num	Number of error records	RO	0x2

## 8.3.18.32 por\_sbsx\_errpidr45

Functions as the identification register for peripheral ID 4 and peripheral ID 5.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEF0D0

## Type

RO

## Reset value

See individual bit resets

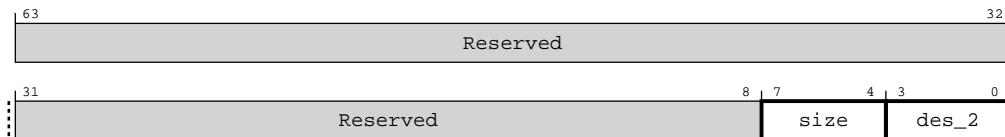
## Usage constraints

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 8-1048: por\_sbsx\_errpidr45**



**Table 8-1063: por\_sbsx\_errpidr45 attributes**

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	
[7:4]	size	Size of the RAS component. 0x0 means 4K block	RO	0x0
[3:0]	des_2	Designer bit[10:7]	RO	0x4

## 8.3.18.33 por\_sbsx\_errpidr01

Functions as the identification register for peripheral ID 0 and peripheral ID 1.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEFE0

### Type

RO

### Reset value

See individual bit resets

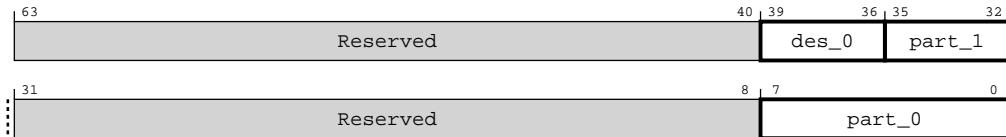
## Usage constraints

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 8-1049: por\_sbsx\_errpidr01**



**Table 8-1064: por\_sbsx\_errpidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	des_0	Designer bit[3:0]	RO	0xb
[35:32]	part_1	Product ID Part 1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	part_0	Product ID Part 0	RO	0x0

### 8.3.18.34 por\_sbsx\_errpidr23

Functions as the identification register for peripheral ID 2 and peripheral ID 3.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEFE8

### Type

RO

### Reset value

See individual bit resets

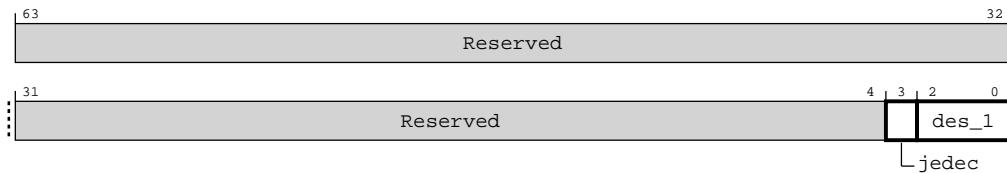
## Usage constraints

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 8-1050: por\_sbsx\_errpidr23**



**Table 8-1065: por\_sbsx\_errpidr23 attributes**

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	0b1
[2:0]	des_1	Designer bit[6:4]	RO	0x3

## 8.3.18.35 por\_sbsx\_errcidr01

Functions as the identification register for component ID 0 and component ID 1.

### Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

### Attributes

#### Width

64

#### Address offset

0xEFF0

#### Type

RO

#### Reset value

See individual bit resets

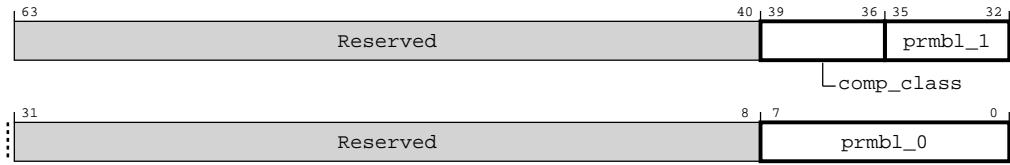
### Usage constraints

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 8-1051: por\_sbsx\_errcidr01**



**Table 8-1066: por\_sbsx\_errcidr01 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:36]	comp_class	Component Class	RO	0xF
[35:32]	prmb1_1	PRMBL_1	RO	0x0
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_0	PRMBL_0	RO	0xD

### 8.3.18.36 por\_sbsx\_errcidr23

Functions as the identification register for component ID 2 and component ID 3.

## Configurations

In certain configurations this register is not present, and the bits are reserved. See the offsets in the IPXact file generated for your configuration to determine if this register is available.

## Attributes

### Width

64

### Address offset

0xEFF8

### Type

RO

### Reset value

See individual bit resets

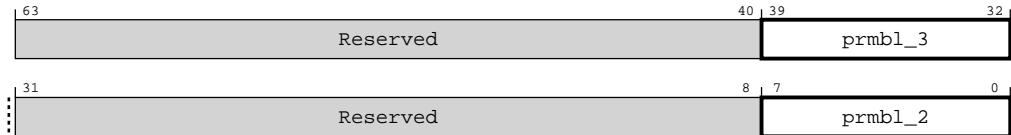
## Usage constraints

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 8-1052: por\_sbsx\_errcidr23**



**Table 8-1067: por\_sbsx\_errcidr23 attributes**

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	
[39:32]	prmb1_3	PRMBL_3	RO	0xB1
[31:8]	Reserved	Reserved	RO	
[7:0]	prmb1_2	PRMBL_2	RO	0x5

### 8.3.18.37 por\_sbsx\_pmu\_event\_sel

Specifies the PMU event to be counted.

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Address offset

0xD900

##### Type

RW

##### Reset value

See individual bit resets

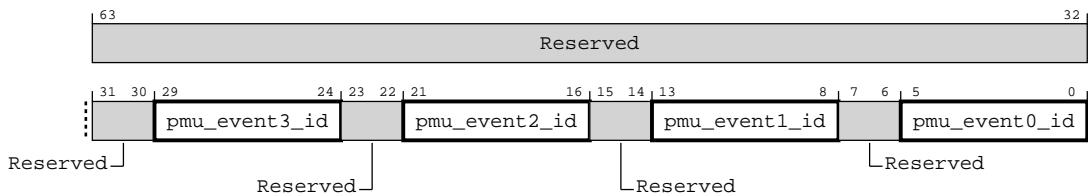
##### Usage constraints

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 8-1053: por\_sbsx\_pmu\_event\_sel**



**Table 8-1068: por\_sbsx\_pmu\_event\_sel attributes**

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	
[29:24]	pmu_event3_id	SBSX PMU Event 3 select; see pmu_event0_id for encodings	RW	0b0
[23:22]	Reserved	Reserved	RO	
[21:16]	pmu_event2_id	SBSX PMU Event 2 select; see pmu_event0_id for encodings	RW	0b0
[15:14]	Reserved	Reserved	RO	
[13:8]	pmu_event1_id	SBSX PMU Event 1 select; see pmu_event0_id for encodings	RW	0b0
[7:6]	Reserved	Reserved	RO	

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>SBSX PMU Event 0 select</p> <p><b>0x00</b> No event</p> <p><b>0x01</b> Read request</p> <p><b>0x02</b> Write request</p> <p><b>0x03</b> CMO request</p> <p><b>0x04</b> RETRYACK TXRSP flit sent</p> <p><b>0x05</b> TXDAT flit seen</p> <p><b>0x06</b> TXRSP flit seen</p> <p><b>0x11</b> Read request tracker occupancy count</p> <p><b>0x12</b> Write request tracker occupancy count</p> <p><b>0x13</b> CMO request tracker occupancy count</p> <p><b>0x14</b> WDB occupancy count</p> <p><b>0x15</b> Read AXI pending tracker occupancy count</p> <p><b>0x16</b> CMO AXI pending tracker occupancy count</p> <p><b>0x17</b> RDB occupancy count. (Only when MTU is enabled)</p> <p><b>0x21</b> ARVALID set without ARREADY</p> <p><b>0x22</b> AWVALID set without AWREADY</p> <p><b>0x23</b> WVALID set without WREADY</p> <p><b>0x24</b> TXDAT stall (TXDAT valid but no link credit available)</p> <p><b>0x25</b> TXRSP stall (TXRSP valid but no link credit available)</p> <p><b>NOTE</b> All other encodings are reserved.</p>	RW	0b0

## 8.4 CMN S3(AE) programming

This section contains CMN S3(AE) programming information.

### 8.4.1 Boot-time programming sequence

You must use a specific boot-time programming sequence to set up CMN S3(AE) correctly. An example sequence is provided, which uses a System Control Processor (SCP) to perform the initial boot configuration.

After reset, the following configuration steps must happen before broad access to CMN S3(AE) components is available:

1. CMN S3(AE) uses a default configuration to access boot flash through the HN-D ACE-Lite requester interface and also the configuration registers.
2. An RN-F, or a requester that is connected to an RN-I, must then access the configuration registers to configure CMN S3(AE). This boot-time configuration must happen before there is broader access to components such as HN-F or SN.



If booting from a device downstream of the HN-D node, in your final RN SAM configuration the boot code region target must not change from the HN-D.

The following example provides more information on the boot process. It assumes an SCP is performing the CMN S3(AE) configuration.

1. The SCP boots, either from local memory or through CMN S3(AE) memory accesses targeting memory behind the HN-D:
  - All other requesters are either held in reset or issue no requests to CMN S3(AE) until the boot programming is complete.
2. If necessary, the SCP discovers the system.
3. The SCP determines the wanted address map and corresponding SAM register values.
4. If necessary, the SCP remaps the configuration register space by completing the following steps:
  - a. It drains all requests in flight by waiting for their responses.
  - b. It issues a single 64-bit store to a PERIPHBASE register behind the HN-D. This register would be in logic that is external to CMN S3(AE) and an update would cause the signal values on the CFGM\_PERIPHBASE input to change.
  - c. It waits for the response for that store.
5. If necessary, the SCP writes to the CMN S3(AE) configuration registers to program the SAM for all HN-Fs.