Arm® Neoverse™ CMN-700 System Address Map Programming Application Note

Non-confidential

Document Issue: 1

Document ID: 107864

Release information

Document history

Issue	Date	Confidentiality	Change
1	01 August 2024	Non-Confidential	First issue

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.
(PRE-1121-V1.0)

Feedback

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

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

Arm periodically provides updates and corrections to its technical content and Frequently Asked Questions (FAQs) on Arm Developer.

Inclusive language commitment

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

To report offensive language in this document, email **terms@arm.com**.

Contents

1. Introduction

To generate coherent traffic, you must configure the System Address Map (SAM) during Arm® CoreLink™ CMN-700 Coherent Mesh Network initialization. There are two SAM components which are required for coherent traffic: the RN SAM and the HN SAM. This Application Note provides guidance for programming the SAM configuration registers.

1.1. Intended audience

This Application Note is for RTL designers who want to understand how their system memory map relates to the SAM registers in their CMN-700 configuration. It is also for software designers who want to understand how their software routes instructions to each part of the hardware through the SAM configuration.

1.2. Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

This document uses the following terms and abbreviations.

Terms and abbreviations

Term	Meaning
SAM	System Address Map
RN	Request Node
HN	Home Node
SN	Subordinate Node
SCG	System Cache Group
HTG	Hashed Target Group
SLC	System Level Cache
SF	Snoop Filter
CAL	Component Aggregation Layer

Term	Meaning
CCG	CXL Gateway
RA	Requesting Agent
SBSX	CHI to AXI or ACE-Lite bridge
AXU	AXI5 Utility Bus
DSU	Arm® DynamIQ™ Shared Unit
XP	Crosspoint
MXP	Mesh Crosspoint

Typographical conventions

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace` **`bold`**	Language keywords when used outside example code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
Note	An important piece of information that needs your attention.

Convention	Use
Tip	A useful tip that might make it easier, better, or faster to perform a task.
Remember	A reminder of something important that relates to the information you are reading.

1.3. Useful resources

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

- Arm Non-Confidential documents are available on developer.arm.com/documentation. Each document link in the tables below provides direct access to the online version of the document.
- Arm Confidential documents are available to licensees only through the product package.

Arm products	Document ID	Confidentiality
Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual	102309	Confidential
Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual	102308	Non-Confidential
Arm® Socrates™ User Guide	101399	Non-Confidential

Note

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.
Adobe PDF reader products can be downloaded at **http://www.adobe.com**.

2. CMN-700 SAM fundamentals

The CMN-700 SAM maps system memory addresses to target CMN-700 nodes. It is used to determine which nodes to send requests to as they travel across the interconnect and in some cases, to define the ordering requirements for transactions to some memory regions.

There are three parts to the CMN-700 SAM: the RN SAM, HN-F SAM, and HN-I SAM. These parts map a system address from a Request Node (RN) to a Home Node (HN), and then to either a Subordinate Node (SN) or an external downstream completer. For more information about each part of the SAM, see the following sections:

- RN SAM
- HN-F SAM
- HN-I SAM

The properties of the SAM are defined by the SAM configuration registers. These registers are distributed in the different components of your CMN-700 configuration. The RN SAM configuration register blocks are children of the MXPs that the RNs in your configuration are connected to. The HN-F SAM and HN-I SAM configuration registers are in the HN-Fs or HN-Is. For more information, see Finding the SAM configuration registers.

2.1. RN SAM

When a request enters the interconnect at a RN, the RN SAM maps the system address for the request to a target ID. This target ID corresponds to a single HN in the configuration, which is the first CMN-700 target for the request.

In the RN SAM, you can divide the full system address space into separate regions. The RN SAM has three types of memory region, which are used for different purposes:

- Generic Interrupt Controller (GIC) memory region
- Non-hashed memory region
- Hashed Target Group (HTG) memory region

For more information about the types of memory regions and their purposes, see the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

The number of each type of memory region is configuration dependent. The RTL configuration determines the number of register resources that are available for programming the RN SAM. For example, if the RTL is configured to have 16 non-hashed regions, only 16 sets of configuration registers are present to configure the properties of these non-hashed regions.

The target of an HTG memory region in the RN SAM is often a System Cache Group (SCG). An SCG is a group of Fully Coherent Home Nodes (HN-Fs). Individual addresses in a hashed memory region are hashed, or distributed, across the different HN-Fs in an SCG.

2.2. HN-F SAM

When a request reaches an HN-F, the HN-F SAM maps the request to the next target ID. This target ID corresponds to a single SN in the configuration, which is the final CMN-700 target for the request.

The HN-F SAM has two different ways to map requests to a target SN:

- Address-range based mapping
- Default hashed region

For more information about these options, see the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

Like the RN SAM, various aspects of the HN-F SAM functionality are configuration dependent. The RTL configuration is configured determines the number and type of register resources that you can use to configure the HN-F SAM.

2.3. HN-I SAM

When a request reaches an HN-I or an HN-I variant, the HN-I SAM ensures that transactions to certain regions in the address space of the HN-I are kept in order.

The completer downstream of an HN-I can be either:

- A peripheral with a memory-mapped I/O space, such as a UART or GPIO
- Physical memory, such as SRAM or flash memory

Depending on the requirements of the downstream completer, you can program the HN-I SAM to configure the properties and ordering requirements of different regions that the HN-I services.

The HN-I SAM lets you partition the address region that an HN-I services into separate address regions. You can assign each HN-I SAM address region its own properties, for example the memory type that it uses.

The HN-I SAM also lets you partition each address region into multiple order regions. All accesses to the addresses in an order region are kept in order.

3. Location of the CMN-700 SAM configuration registers

The SAM configuration registers are distributed throughout the CMN-700 configuration space. Therefore, to program the SAM configuration registers, you must know their location in the CMN-700 configuration space.

Each configuration register in CMN-700 has an offset from the base address of the configuration space. These offsets determine where to send configuration requests during initialization. To find the offsets of the SAM registers, you must know how the CMN-700 configuration space is organized. For more information, see Organization of the CMN-700 configuration register space.

You can determine the structure of the CMN-700 configuration space for your configuration in two ways:

- Automatically, using the software discovery mechanism
- Manually, using the rendered IP-XACT file for your configuration

For more information, see Organization of System Address Map configuration registers.

3.1. CMN-700 configuration register space structure

All the CMN-700 configuration registers are contained in a single region of the memory map. This region is divided into subregions for each programmable node in the configuration.

The size of the CMN-700 configuration register memory region depends on the size of the mesh. For mesh configurations with X and Y dimensions of eight or less, the region is 256MB. For mesh configurations with X and Y dimensions of 9 or more, the region is 1GB. The region must be size-aligned, so a 256MB configuration register region must be 256MB-aligned in the memory map.

Each configurable CMN-700 node has a separate 64KB subregion in this memory map, and this subregion contains all the configuration registers for that node. By finding the offset of the configurable node for each component, you can determine the offset for each configuration register.

To determine the offset of a specific SAM configuration register, you must know three pieces of information:

- The base address of the CMN-700 configuration register space. This address is also known as PERIPHBASE.
- The base address for the subregion of a node that contains a SAM, as an offset from PERIPHBASE. For example, this could be the base address of an HN-F.
- The offset of the individual register from the subregion base address

Combining these three values gives an address for a specific configuration register.

3.2. SAM configuration register organization

The organization of the SAM registers in the CMN-700 configuration space varies for the different parts of the SAM.

The RN SAM registers for each RN-I or RN-F node are in the subregion for the MXP that they are connected to, not the subregion for the RN-I or RN-F nodes themselves. The HN-I SAM and HN-F SAM registers are in the same memory subregion as the HN-I or HN-F node they belong to.

Each RN SAM configuration node only contains RN SAM registers, so all the registers in that node are part of the RN SAM configuration. By contrast, the HN-I SAM and HN-F SAM registers are amongst other HN configuration registers that are not related to the SAM configuration. Therefore, the offsets for the HN SAM registers start partway through the HN configuration node. For more information on the structure of the different HN configuration nodes, see the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

As CMN-700 configuration register space structure states, to determine the address for each SAM register, you require the PERIPHBASE value, the base address of each configuration node, and the offset of each SAM register from the configuration node base address. There are two ways to find these values: using the software discovery flow or by taking the values from the IP-XACT file that is rendered during RTL configuration. The following sections describe these methods.

3.2.1. Use the software discovery mechanism to find offsets of the SAM configuration registers

To determine the base address of each node in the configuration automatically, you can use the software discovery mechanism. When you know the base address of each node, you can use it to find the exact address of each SAM register in the configuration.

About this task

The discovery mechanism allows CMN-700 initialization without prior knowledge of the mesh topology. Discovery enables the design of generic software that can be used for any CMN-700 configuration. Through the discovery mechanism, software traverses a discovery tree to determine the location of the configuration registers for each node in the mesh. During this process, the software builds a structure containing the following information for each node or device in the configuration:

- Node type
- Node ID
- Logical ID
- Base address of the configuration register subregion for the node

The discovery process starts at PERIPHBASE. CMN-700 has an input strap signal, CFGM_PERIPHBASE, and bits [MSB:28] of this signal define the value of PERIPHBASE during initialization. To set PERIPHBASE, you must drive the required base address on those signal bits. To determine the Most Significant Bit (MSB) of this signal, use the formula REQ_ADDR_WIDTH –

1. For example, if REQ_ADDR_WIDTH is set to 48, then to set the value of PERIPHBASE, drive CFGM_PERIPHBASE[47:28] to point to the PERIPHBASE address in your memory region.

The first set of registers at PERIPHBASE are the Configuration Manager (CFGM) registers, which are in the HN-D node. Before initialization, the RN SAM has a default configuration with a single non-hashed region and the node ID of the HN-D as its target. When the default configuration is active, the RN SAM routes all configuration requests to the HN-D.

The CFGM registers contain pointers to the base address for each Crosspoint (XP) in the configuration. The XP registers then contain pointers to each node that is connected to their device ports. The device nodes contain pointers to any configuration nodes that they contain, and so on, creating a tree structure.

For more information about the discovery process, see the *Discovery* section of the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

Procedure

1. Read the value of the por_cfgm_node_info register at address PERIPHBASE.
2. Read the value of the por_cfgm_child_info register at address PERIPHBASE + `0x80`. This register value provides the number of XPs in the configuration.
3. Read the value of the por_cfgm_child_pointer_0 register at address PERIPHBASE + por_cfgm_child_info.child_ptr_offset. This register value provides the base address of the first XP, labelled XP 0, as an offset from PERIPHBASE.
4. Read the value of the por_mxp_node_info register at the XP 0 base address.
5. Read the value of the por_mxp_child_info register at address XP 0 base address + `0x80`. This register value provides the number of devices attached to XP 0.
6. Read the value of the por_mxp_child_pointer_0 register at address XP 0 base address + por_mxp_child_info.child_ptr_offset. This register value provides the base address of the first device, device 0, as an offset from PERIPHBASE. For example, consider the scenario where device 0 is an RN-I, labelled RN-I 0.
7. Read the value of the por_rni_node_info register at the RN-I 0 base address.
8. Read the value of the por_rni_child_info register at address RN-I 0 base address + `0x80`. This register value provides the number of devices attached to RN-I 0, which is 0. The RN-I is the tip of this branch of the discovery tree because it has no child nodes.
9. Repeat steps 6-8 for each child node and sub-node attached to XP 0. For example, the pointer to the next node on XP 0 is provided by the por_mxp_child_pointer_1 register.
10. Repeat steps 3-9 for each XP in the configuration. For example, the pointer to the next XP in the configuration, XP 1, is provided by the por_cfgm_child_pointer_1 register.

Next steps

You can use the offsets for each node that you determined with the software discovery mechanism to find the specific register offset for each SAM configuration register.

3.2.2. SAM configuration register offsets in the IP-XACT file

When you render the CMN-700 RTL after configuring the interconnect in Arm® Socrates™, the rendering flow outputs an IP-XACT file. You can use the details in this file to determine the base address for each node and register in the interconnect configuration, and therefore the address of each SAM register.

Socrates renders the IP-XACT in the same place as the rendered RTL in your Socrates workspace. For example, the path to the IP-XACT file might be `<Socrates_workspace_path>/<project_path>/logical/cmn700/ipxact/cmn700_1.xml`.

In the IP-XACT file, the base address of each node is expressed as an offset value. The register offsets in the IP-XACT file are calculated according to the value of the **Peripheral Base Address** parameter. If the parameter is set to 0, the register address offset values in the IP-XACT file are expressed as a value relative to PERIPHBASE. If the parameter is set to a non-zero value, then the offsets in the IP-XACT file are expressed as the combined value of the **Peripheral Base Address** parameter and the node base address offset.

The Socrates **Peripheral Base Address** parameter only affects the IP-XACT file. It does not affect the rendered RTL.

The following example shows an excerpt of an IP-XACT file with the details for a single RN SAM node. In this example, the **Peripheral Base Address** parameter is set to 0, so the offsets are relative to PERIPHBASE.

```
<spirit:register>

      <spirit:name>por_rnsam_node_info_u_rnfeesam_nid2</spirit:name>

      <spirit:displayName>por_rnsam_node_info for instance:
u_rnfeesam_nid2</spirit:displayName>

      <spirit:description>Provides component identification
information.</spirit:description>

      <spirit:addressOffset>0x1A0000</spirit:addressOffset>

      <spirit:size>64</spirit:size>

      <spirit:access>read-only</spirit:access>

      <spirit:reset>

            <spirit:value>0x2000f</spirit:value>
```

For more information about configuring Arm IP in Socrates, see the Arm® Socrates™ User Guide.

4. Overall SAM programming requirements

Various requirements apply to the process of programming the SAM. If you do not follow these requirements, the behavior of CMN-700 can be UNPREDICTABLE.

There are specific requirements that apply to RNs in the CMN-700 configuration during SAM programming, to ensure that only configuration traffic enters the network during this boot stage. For more information, see Requirements for RNs during SAM programming.

There is a minimum level of RN SAM programming that is required for CMN-700 to function correctly. For more information, see Minimum RN SAM programming requirements.

There are also different SAM programming requirements for I/O-type memory and DDR-type memory. For more information, see Programming requirements for specific memory types.

4.1. Requirements for RNs during SAM programming

To set up the SAM, a single requester must program all configuration nodes that contain SAM registers.

Specifically, the requester must program:

- All the RN SAM configuration nodes for each RN in the configuration
- All the HN-F SAM and HN-I SAM registers in each HN-F and HN-I configuration node in the configuration
- For Coherent Multichip Link (CML) configurations, the RA SAM registers in each CXRA

The requester configuring the SAM can be an RN-F, a requester connected to an RN-I, or a System Control Processor (SCP).

A CMN-700 configuration typically contains various requesters of different types. These requesters include RN-Fs, AXI or ACE-Lite requesters attached to RN-I or RN-D nodes, and, if CMN-700 is in CML Symmetric Multiprocessing (SMP) mode, CCGs too. During SAM programming, we recommend that all requesters other than the one that configures the SAM are prevented from issuing requests into the interconnect. This constraint should remain until the rnsam_status.use_default_node bit is cleared to 0. If this recommendation is not followed, the default behavior of the SAM remains active. All RN SAM blocks route requests to the HN-D node before the use_default_node bit is cleared to 0. Therefore, any non-configuration traffic from the other requesters is also routed to the HN-D. This behavior could lead to various issues. The HN-D does not manage coherency, so requests sent to the HN-D would not be kept coherent. Also, any writes sent downstream of the HN-D node would not reach the DRAM and any reads sent downstream would receive old or unexpected values.

For AXI or ACE-Lite requesters connected to RN-Is or RN-Ds, the preferred method to prevent these requesters from issuing requests is to set the RNID_SAM_STALL_DIS_Sx_NIDy inputs on the

RN-I or RN-D AXI or ACE-Lite interfaces LOW. If these signals are LOW coming out of reset, the RN-I or RN-D node keeps its AxREADY signals LOW until the RN SAM rnsam_status.nstall_req bit is set to 1. Therefore, requests from those interfaces are prevented from entering the interconnect. When programming is complete, the rnsam_status.nstall_req bit is set to 1, allowing the AxREADY signals to go HIGH.

Caution

Arm recommends programming the HN-F SAM before programming the RN SAM. Configuring the HN-F SAM before configuring the RN SAM ensures the intended SN-F/SBSX nodes are programmed into the HN-F SAM before completing the RN SAM programming, and it guarantees that initial SLC misses will have valid SN targets.

4.2. Minimum RN SAM programming requirements

For your CMN-700 implementation to function correctly, there is a minimum set of requirements for RN SAM programming. This programming ensures that requesters can target the HN-F nodes in the configuration.

The minimum RN SAM programming requirements are:

- Assign all HN-F nodes to SCG0, where the address range for SCG0 spans the entire address space, from address 0 to address $2^{REQ\_ADDR\_WIDTH}$.
- Define one non-hashed region that targets the HN-D node. This region must encompass all the PERIPHBASE ranges, such as the Configuration Manager (CFGM), DynamIQ Shared Unit (DSU), Dynamic Memory Controller (DMC), and MXP_AXU.

4.3. Programming requirements for specific memory types

Each CMN-700 SAM address region must be designated as either I/O memory or DDR memory and there are some further requirements for each memory type.

The I/O memory type corresponds to memory-mapped I/O devices or physical memory such as SRAM or flash memory. The following CMN-700 node types can service regions marked as I/O memory:

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

Each of these nodes also has HN-I SAM registers. Address regions serviced by HN-Is or HN-I variants can be designated as Physical memory or Peripheral memory by programming these attributes in the HN-I SAM registers. Physical memory follows Normal memory guarantees and Peripheral memory follows Device memory guarantees. By default, each address region is configured as Peripheral memory. All transaction ordering at an HN-I or HN-I variant is based on target address, meaning that the MemAttr value is ignored.

Any configuration register transactions targeting the HN-D must be Device-type, so you must configure an address region that targets the HN-D as Peripheral memory.

The DDR memory type corresponds to DRAM memory. The following CMN-700 node types can service regions marked as DDR memory:

- HN-F
- SN-F
- SBSX

All HN-F accesses are treated as Normal memory accesses, meaning that the MemAttr value is ignored. Any memory managed by the HN-F and accessed by either SN-F or SBSX must follow guarantees of the Normal memory type.

5. HN-I SAM programming

Each HN-I has a set of HN-I SAM registers, which you use to configure the memory properties and ordering requirements for the HN-I and its downstream completer. The HN-I SAM functionality is optional, so you do not have to program this part of the SAM for CMN-700 to function.

At reset, address region 0 of the HN-I SAM is set up automatically to cover the entire HN-I SAM address space. Therefore, any transaction that the RN SAM routes to an HN-I can be sent downstream, even if you do not program the HN-I SAM registers.

The por_hni_sam_addrregion{0-3}_cfg registers contain controls that you can use to configure the different HN-I SAM address regions. The {0-3} notation in the names of the registers correspond to the address region numbers. For example, the por_hni_sam_addrregion2_cfg register controls the properties for address region 2.

There are various settings in each of these registers to configure the following properties for each address region:

- The address range for the address region. For more information, see HN-I SAM address region range configuration.
- The memory type for the address region. For more information, see HN-I SAM address region memory type configuration.
- The order region size for the address region. For more information, see HN-I SAM order region size configuration.
- Optional Point-of-Serialization (PoS) settings for the address region. For more information, see Optional PoS settings for the HN-I SAM address regions.

There are also some specific sets of requirements for the HN-I SAM when the HN-I connects to a PCIe device. For more information, see HN-I SAM programming requirements for PCIe device connection.

5.1. HN-I SAM address region range configuration

At reset, the range of HN-I SAM address region 0 is set to cover the whole HN-I SAM address space and the range of this region cannot be changed. To divide the address space into multiple regions with various ordering requirements, use HN-I SAM address regions 1-3.

To set up the address ranges for address regions 1-3, program the following register fields in the por_hni_sam_addrregion{1-3}_cfg registers:

addr_region_size

Specifies the size of the address region. The sizes of regions 1-3 are programmable in 2^n x 4KB size.

base_addr

Specifies the base address of the address region

valid

Specifies whether the programmed values for the address region are valid

Warning

Address regions 1-3 must not overlap. If this requirement is not followed, then ordering might not be maintained and the behavior of CMN-700 can be UNPREDICTABLE.

5.2. HN-I SAM address region memory type configuration

You can configure the memory type of each address region as either Peripheral and Physical memory by setting the por_hni_sam_addrregion{0-3}_cfg.physical_mem_en bit.

The por_hni_sam_addrregion{0-3}_cfg.physical_mem_en bit value has the following encodings:

0
: Peripheral memory. The address region uses the ordering guarantees of Device-type memory.

1
: Physical memory. The address region uses the ordering guarantees of Normal-type memory. In other words, all transactions targeting this address region are issued to the downstream completer with unique AxID values. Any order region settings are ignored.

The reset value of this bit is 0. Therefore, by default, the HN-I SAM sets the entire address space, covered by address region 0, as Peripheral memory. In other words, at reset, the whole HN-I SAM address space uses the ordering guarantees of Device-type memory.

5.3. HN-I SAM order region size configuration

You can divide each HN-I SAM address region further into equally-sized order regions by setting the por_hni_sam_addrregion{0-3}_cfg.order_region_size register field.

Order regions help to guarantee the ordering requirements for transactions that target the same region. All transactions that target the same order region are issued downstream with the same AxID value. Also, the HN-I will order reads vs. writes to the same order region by waiting for completion responses before issuing the next request.

The HN-I SAM address regions support order region sizes that are multiples of 4KB, from a minimum of 4KB up to a maximum of the full size of the address region. The following equation determines the size of the order regions in an address region:

Order region size = 2^n x 4KB, where n = the value of por_hni_sam_addrregion{0-3}_cfg.order_region_size.

5.4. Optional PoS settings for the HN-I SAM address regions

The HN-I SAM address regions support optional settings for configuring the HN-I Point-of-Serialization (PoS) behavior. These settings constrain or restrict the behavior of the HN-I when issuing certain types of transactions.

The por_hni_sam_addrregion{0-3}_cfg registers have the following fields to enable or disable specific PoS behavior:

pos_early_wr_comp_en
: Enables early write acknowledgement for the address region, which helps to improve write performance.

pos_early_rdack_en
: Enables early read receipt from HN-I in the address region, which helps to improve ordered read performance.

ser_devne_wr
: If enabled, HN-I serializes all Device-nGnRnE writes targeting the address region.

ser_all_wr
: If enabled, HN-I serializes all writes targeting the address region.

5.5. HN-I SAM programming requirements for PCIe device connection

If the HN-I connects to a PCIe device, then to allow CMN-700 to communicate with the endpoint, there are PCIe ordering requirements that apply to the HN-I. To ensure the HN-I follows these requirements, you must configure the HN-I SAM address region in a specific way.

If a single HN-I SAM address region contains the configuration address space for a PCIe Root Complex (RC), then the HN-I must ensure that all PCIe configuration writes complete in order. The required HN-I SAM address region settings to ensure this ordering behavior depend on the type of PCIe endpoint:

- If the endpoint is a Device-nGnRnE endpoint, set por_hni_sam_addrregion{0-3}_cfg.ser_devne_wr for the corresponding address region to 1.
- If the endpoint is a different type of PCIe endpoint, set por_hni_sam_addrregionN_cfg.ser_all_wr for the corresponding address region to 1.

If an HN-I SAM address region contains the memory range for a PCIe endpoint, then the HN-I must not issue early write completions back to the RN. It also must pass the PoS responsibility to the PCIe system. To achieve this behavior, set por_hni_sam_addrregion{03}_cfg.pos_early_wr_comp_en for the corresponding address region to 0.

6. HN-F SAM programming

Each HN-F has a set of HN-F SAM registers, which you use to configure the target SNs for all requests that the HN-F receives.

Each HN-F services a specific region of your overall address map. The HN-F SAM maps all the served addresses to a target SN. Each HN-F SAM specifies a default hashed region to provide a default mapping plus multiple optional address ranges for more specific range-based mapping.

The settings that the HN-F SAM uses to define address ranges are configuration dependent. For more information, see HN-F SAM address range definition methods.

You can use range-based mapping to specify extra address regions and either map each region to a single SN or hash all requests to that region across a group of SNs. The default hashed region then provides the mapping for any addresses that are outside of the range-based mapping regions. The default hashed region is also either directly mapped to a single SN or hashed across a group of SNs. For more information about the default hashed region and optional range-based mapping regions, see HN-F SAM default hashed region and HN-F SAM range-based mapping.

The HN-F SAM provides options to indicate the configuration of each target SN, to ensure that the correct request types are sent to the target. For more information about these options, see Configuration of HN-F SAM target SN properties.

There are specific requirements that you must be aware of when programming the HN-F SAM. For more information, see HN-F SAM programming requirements.

There is a specific procedure to follow when programming the HN-F SAM. For more information, see Program the HN-F SAM.

6.1. HN-F SAM address range definition methods

You can choose whether the CMN-700 HN-F SAM address ranges are defined using either the base address and the region size or a start address and end address. You select one of the two choices when you configure the CMN-700 RTL in Socrates.

The CMN-700 `HNSAM_RCOMP_EN` configuration parameter configures how the interconnect defines HN-F SAM address ranges. This parameter supports the following values:

FALSE

The region size is defined by specifying a base address for the region and the size of the region. The size of the region is fixed from 64MB to 4PB, in power-of-two increments. The base address must be aligned to the range size.

TRUE

The region size is defined by specifying a start address and end address for the region. The minimum size must be specified by setting `HNSAM_RCOMP_LSB` parameter to a value between 20-26, which correspond to a minimum size of 1MB to 64MB in power-of-two increments. The start address and the end address of the region must be aligned with the minimum size.

6.2. HN-F SAM default hashed region

The HN-F SAM default hashed region provides a default target for requests that do not match a configured HN-F SAM address range.

To set up the default hashed region in the HN-F SAM, you must define the properties of the address region.

You must also define the SN or SNs that are the target of the default hashed region. If requests to the default hashed region are hashed across multiple SNs, you must also configure the properties of the hash function. For more information, see HN-F SAM default hashed region SN mapping options.

Use the cmn_hns_sam_cfg{1,2}_def_hashed_region registers to set up the HN-F SAM default hashed region address range.

By default there is no address range programmed for the default hashed region. Any request that does not match a non-hashed or HTG address range will result in a Decode Error (DECERR). This Decode Error will not be signalled in the DAT/RSP.RespErr, since the RespErr will indicate OK. For more information about the Decode Error, see sections 3.8.5 and 5.2.5 in the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

If the Decode Error functionality is not required, then the address range for the default hashed region should be configured to cover the entire address space.

The cmn_hns_sam_cfg{1,2}_def_hashed_region registers control the properties of the default address region. The cmn_hns_sam_cfg1_def_hashed_region register has the following fields for configuration:

valid
: Specifies whether the default address region programming is valid

size
: Specifies the size of the default address region if the `HNSAM_RCOMP_EN` parameter is set to `FALSE`.

base_addr
: Specifies the start address of the default region

The cmn_hns_sam_cfg2_def_hashed_region register.end_addr field is also configurable. This field specifies the end address of the default region if the `HNSAM_RCOMP_EN` parameter is set to `TRUE` in your CMN-700 configuration.

6.2.1. HN-F SAM default hashed region SN mapping options

There are various ways that you can map SN targets to the CMN-700 HN-F SAM default hashed region. The final mapping configuration depends on the number of SNs that are mapped to the HN-F.

The default hashed region supports three different ways to map the default address region to SN targets:

Direct mapping

Requests to the whole default hashed region are mapped to a single SN.

Power-of-two hashing

Requests to the default hashed region are hashed across 2, 4, or 8 SNs. A power-of-two hashing algorithm is used to determine the target for a specific address.

Non-power-of-two hashing

Requests to the default hashed region are hashed across 3, 5, or 6 SNs. A non-power-of-two hashing algorithm is used to determine the target for a specific address.

The following registers are used to configure SN target node IDs for the HN-F and the associated SN mapping:

- cmn_hns_sam_control
- cmn_hns_sam_control2
- cmn_hns_sam_6sn_nodeid

6.2.1.1. Direct mapping configuration

When using direct mapping, the whole default hashed region address space for an HN-F is mapped to one SN. To set up direct mapped mode, program the node ID for the relevant SN target into the cmn_hns_sam_control.hn_cfg_sn0_nodeid field.

6.2.1.2. Power-of-two hashing configuration

In power-of-two hashing, the HN-F SAM hashes addresses in the default hashed region across 2, 4, or 8 SNs. The programming requirements for power-of-two hashing depend on whether the HN-F SAM is hashing across 2, 4, or 8 SNs.

To map the default hashed region to 2 SNs, use the following settings:

- Program the node IDs for the two SNs into the cmn_hns_sam_control.hn_cfg_sn{0,1}_nodeid register fields.
- Set the cmn_hns_sam_control2.hn_cfg_two_sn_en register field to 1.

To map the default hashed region to 4 SNs, use the following settings:

- Program the node IDs for the first 3 SNs into the cmn_hns_sam_control.hn_cfg_sn{0-2}_nodeid register fields.
- Program the node ID for the remaining SN into the cmn_hns_sam_6sn_nodeid.hn_cfg_sn3_nodeid register field.
- Set the cmn_hns_sam_control2.hn_cfg_four_sn_en register field to 1.

To map the default hashed region to 8 SNs, use the following settings:

- Program the node IDs for the first 3 SNs into the cmn_hns_sam_control.hn_cfg_sn{0-2}_nodeid register fields.
- Program the node IDs for the remaining 5 SNs in the cmn_hns_sam_6sn_nodeid.hn_cfg_sn{3-7}_nodeid register fields.
- Set the cmn_hns_sam_control2.hn_cfg_eight_sn_en register field to 1.

6.2.1.3. Non-power-of-two hashing configuration

In non-power-of-two hashing, the HN-F SAM hashes addresses in the default hashed region across 3, 5, or 6 SNs. The programming requirements for non-power-of-two hashing depend on whether the HN-F SAM is hashing across 3, 5, or 6 SNs.

The non-power-of-two hash function uses a selection of address bits as inputs to the hash function. These address bits are a configurable group of nine address bits in the range of address bits [21:8] and extra upper address bits configured as top address bits. You can use one of the following groups of address bits:

- [16:8], which is the default option
- [17:9]
- [18:10]
- [19:11]
- [20:12]
- [21:13]

To map the default hashed region to 3 SNs, use the following settings:

- Program the node IDs for the SNs into the cmn_hns_sam_control.hn_cfg_sn{0-2}_nodeid register fields.
- Set the cmn_hns_sam_control.hn_cfg_three_sn_en register field to 1.

To map the default hashed region to 5 SNs, use the following settings:

- Program the node IDs for the first 3 SNs into the cmn_hns_sam_control.hn_cfg_sn{0-2}_nodeid register fields.
- Program the node ID for the remaining 2 SNs into the cmn_hns_sam_6sn_nodeid.hn_cfg_sn{3,4}_nodeid register fields.
- Set the cmn_hns_sam_control.hn_cfg_five_sn_en register field to 1.

To map the default hashed region to 6 SNs, use the following settings:

- Program the node IDs for the first 3 SNs into the cmn_hns_sam_control.hn_cfg_sn{0-2}_nodeid register fields.
- Program the node ID for the remaining 3 SNs into the cmn_hns_sam_6sn_nodeid.hn_cfg_sn{3-5}_nodeid register fields.
- Set the cmn_hns_sam_control.hn_cfg_six_sn_en register field to 1.

You must also configure the address bits that are used in the hash function. To do so, program the following register fields:

- cmn_hns_sam_control.hn_cfg_sam_top_address_bit{0-2} to set the top address bit used in the hash function.
- cmn_hns_sam_control.hn_cfg_sam_inv_top_address_bit to invert the top address bit for the hashed mode, if required. The top address bit depends on the hashed mode that is used:
    - For 3-SN hashing, the top address bit is top_address_bit1.
    - For 5-SN and 6-SN hashing, the top address bit is top_address_bit2.
- cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel to select the group of address bits that are used in the hash function

The appropriate top_address_bit and inv_top_address_bit selection depends on the hashed mode used and the expected DRAM for each SN port. For guidance on how to set the top_address_bit and inv_top_address_bit, see the *HN-F to SN-F memory striping in HN-F SAM* section of the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

6.3. HN-F SAM range-based mapping

The CMN-700 HN-F SAM supports range-based mapping, which maps a specific SN or group of SNs to a specific range of system addresses.

If an address range is mapped to a single SN, it is called a non-hashed region. Non-hashed regions are useful when a partition of memory from the global DRAM is mapped explicitly to an individual SN, for example, an on-chip SRAM.

If an address range is mapped to a group of SNs, the group is called an HTG.

To handle address range overlaps between address ranges of different types, the HN-F SAM implements a prioritization system to determine the selected targets. The different range types are prioritized in the following way, from highest priority to lowest priority:

1. Range-based mapping, non-hashed address region
2. Range-based mapping, HTG region
3. Default hashed region, generating either direct-mapped target ID or hashed target ID

For example, if an incoming address is in both a range-based non-hashed address region and a range-based HTG region, the non-hashed address region target has the higher priority. Therefore, the request is sent to the non-hashed address region target ID. If an incoming address does not fall into any of the range-based mapping regions, then the HN-F SAM uses the default hashed region target selection scheme to generate the target ID.

6.3.1. HN-F SAM non-hashed region configuration

To configure a non-hashed address region in the HN-F SAM, there are various programming requirements. There are also build-time configuration options which affect the non-hashed regions that are available to program.

The number of non-hashed regions available to program is configuration dependent. Up to 64 memory regions can be created by configuring the `HNSAM_NUM_NONHASH` configuration parameter.

Each non-hashed region has a register to program the range and node ID that is assigned to the region. The cmn_hns_sam_memregion{0,1} registers control the settings for non-hashed regions 0 and 1. The cmn_hns_sam_nonhash_cfg1_memregion{2-63} control the settings for non-hashed regions 2-63. These registers have the following fields for configuration:

valid
: Specifies whether the default region is valid

base_addr
: Specifies the start address of the region

size
: Specifies the size of the region if the `HNSAM_RCOMP_EN` configuration parameter is set to `False`.

range{0-63}_nodeid
: Specifies the SN target node ID for the address range

The cmn_hns_sam_memregion{0,1}_end_addr registers each have an end_addr field. Also, the cmn_hns_sam_nonhash_cfg1_memregion{2-63} registers each have an hns_nonhash_region_end_addr{2-63} field. These register fields specify the end address of the address range if the `HNSAM_RCOMP_EN` configuration parameter is set to `True`.

6.3.2. HN-F SAM HTG configuration

To configure a range-based HTG region in the HN-F SAM, there are various programming requirements. There are also build-time configuration options which affect the HTG regions that are available to program.

The number of HTG regions available to program is configuration dependent. Up to 16 HTG regions can be created by configuring the `HNSAM_NUM_HTG` configuration parameter. An HTG can only contain 2, 3, 4, 5, 6, or 8 SNs.

Range-based HTG mapping works similarly to the mapping for the default hashed region, except when the HN-F SAM hashes across 3, 5, or 6 SNs. For 3-SN, 5-SN, and 6-SN hashing in HTG regions, the output of the hash function provides an offset in the SN target ID table from the base index for the HTG.

Each HTG region has three registers which contain general properties for the region, such as the address range the region covers and the properties for the hash function:

- cmn_hns_sam_htg_cfg1_memregion{0-15}
- cmn_hns_sam_htg_cfg2_memregion{0-15}
- cmn_hns_sam_htg_cfg3_memregion{0-15}

The cmn_hns_sam_cfg1_memregion{0-15} registers have the following fields for programming:

htg_region_valid{0-15}
: Specifies whether the HTG region is valid

htg_region_size{0-15}
: Specifies the size of the region if the `HNSAM_RCOMP_EN` configuration parameter is set to `False`

htg_region_base_addr{0-15}
: Specifies the start address of the HTG region

The cmn_hns_sam_htg_cfg2_memregion{0-15} registers have an htg_region_end_addr{0-15} field. These register fields specify the end address of the address range if the `HNSAM_RCOMP_EN` configuration parameter is set to `True`.

The cmn_hns_sam_htg_cfg3_memregion{0-15} registers have the following fields for programming:

htg{0-15}_hash_addr_bits_sel
: Specifies the group of hashed address bits for 3-SN, 5-SN, and 6-SN hashing

htg{0-15}_sn_mode
: Specifies the mapping mode for the HTG

htg{0-15}_inv_top_address_bit
: Inverts the top address bit for the hash function. For 3-SN hashing, this bit is top_address_bit1. For 5-SN and 6-SN hashing, this bit is top_address_bit2.

htg{0-15}_top_address_bit{0-2}
: Specifies the top address bit for 3-SN, 5-SN, and 6-SN hashing

Also, the node IDs for all the SN targets for all range-based HTGs must be specified in the cmn_hns_sam_htg_sn_nodeid_reg{0-15} registers. These registers form an SN target ID table which the HN-F SAM uses to map target SNs to each HTG region. Each register contains 4 fields for SN target node IDs. How the entries are mapped to the HTG region depends on the sn_mode value. The number of targets is based on the sn_mode setting for the number of SNs used to stripe.

6.4. Configuration of HN-F SAM target SN properties

Each SN target in the HN-F SAM has a set of programmable properties. The properties are used to specify features that the target SN supports, so the HN-F is aware of and can send the transaction types that the downstream SN supports.

The registers used to program the SN properties depend on whether you are programming the properties for the targets of the default hashed region, a range-based non-hashed region, or a range-based HTG region.

The registers that control the target SN properties differ for each HN-F SAM address region type:

Default hashed region

The following registers contain the settings for SN 0-5:

- cmn_hns_sam_sn_properties
- cmn_hns_sam_sn_properties1

The cmn_hns_sam_sn_properties2 register contains the settings for SN 6 and 7.

Range-based non-hashed region

The following registers contain the settings for non-hashed regions 0 and 1:

- cmn_hns_sam_sn_properties
- cmn_hns_sam_sn_properties1

The cmn_hns_sam_nonhash_cfg2_memregion{2-63} registers contain the settings for non-hashed regions 2-63.

Range-based HTG region

The cmn_hns_sam_htg_sn_attr{0-15} registers contain the settings for each SN in the SN target ID table.

Each SN has the following properties which can be programmed:

Data width

For SBSX node SN targets: Specifies whether the target SN supports a data width of 256 bits or 128 bits.

For SN-F and CCG SN targets it must be 256 bits.

This property is controlled by the corresponding sn<n>_128b or range<n>_sn_128b register field. A value of 0 corresponds to 256 bits and a value of 1 corresponds to 128 bits.

CMO propagation support

Specifies whether the HN-F can propagate Cache Maintenance Operations (CMOs) to the target SN.

This property is controlled by the corresponding sn<n>_cmo_prop_en or range<n>_sn_cmo_prop_en register field. A value of 0 indicates that CMO propagation is disabled and a value of 1 indicates that CMO propagation is enabled.

Persistent CMO propagation support

Specifies whether the HN-F can propagate persistent CMOs, such as CleanSharedPersist operations, to the target SN.

This property is controlled by the corresponding sn<n>_pcmo_prop_dis or range<n>_pcmo_prop_dis register field. A value of 0 indicates that persistent CMO propagation is enabled and a value of 1 indicates that persistent CMO propagation is disabled.

Persistent CMO conversion support

Specifies whether the HN-F can convert persistent CMO operations with a separate Persist to regular persistent CMO operations. For example, if this property is supported, the HN-F converts CleanSharedPersistSep operations to CleanSharedPersist operations before sending the transaction to the SN.

This property is controlled by the corresponding sn<n>_ pcmosep_conv_to_pcmo or range<n>_pcmosep_conv_to_pcmo register field. A value of 0 indicates that conversion is disabled and a value of 1 indicates that conversion is enabled.

CHI-C version support

Specifies that the downstream SN supports CHI-C.

This property is controlled by the corresponding sn<n>_is_chic or range<n>_sn_is_chic register field. A value of 0 indicates that the target does not support CHI-C and a value of 1 indicates that the target supports CHI-C.

This field must be set to 0 if the target is an SBSX node.

Return data interleaving control

Specifies whether the SN guarantees that return data is not interleaved.

This property is controlled by the corresponding sn<n>_nointlvdata_guaranteed or range1_sn_nointlvdata_guaranteed register field. A value of 0 indicates that the SN might return interleaved data and a value of 1 indicates that the SN guarantees that return data is not interleaved.

If the target is an SBSX with a data width of 128 bits, this field must be set to 1 to use the Direct Memory Transfer (DMT) flow.

CHI-E version support

Specifies that the downstream SN supports CHI-E.

This property is controlled by the corresponding sn<n>_is_chie or range<n>_sn_is_chie register field. A value of 0 indicates that the target does not support CHI-E and a value of 1 indicates that the target supports CHI-E.

This field is enabled by default if the target is an SBSX node, although CHI.C/CHI.D can be indicated to restrict transaction types sent. For more information about these options, see section 5.2.4 of the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

SN grouping

Specifies the group that the SN belongs to. This setting is mainly used for Completer Busy (CBusy) indication.

This property is controlled by the corresponding sn<n>_group or range<n>_sn_group register field. A value of 0 indicates that the target belongs to group A and a value of 1 indicates that the target belongs to group B.

Metadata termination flow enabled

Specifies whether the HN-F implements the metadata termination flow for this target.

This property is controlled by the corresponding sn<n>_metadata_dis or range<n>_sn_metadata_dis register field. A value of 0 indicates that the HN-F does not use the metadata termination flow and a value of 1 indicates that the HN-F uses the metadata termination flow for this target.

6.5. HN-F SAM programming requirements

There are various requirements that you must follow when programming the HN-F SAM. These requirements ensure that the HN-F SAM functions correctly and efficiently and uses all its resources.

You must ensure that all address ranges in your address map have an SN target in the HN-F SAM. If an access targets an address range with no SN target defined in the HN-F SAM, a memory address Decode Error (DECERR) occurs.

For more information on DECERR responses, see the HN-F error handling section of the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

There are some requirements that apply to the HN-F SAM configuration when you use certain RN SAM configurations. For more information, see HN-F SAM programming requirements for RN SAM non power-of-two hashing or hierarchical hashing use case.

There are also some requirements for systems with multiple SNs where address bit removal is applied to retain a contiguous address space. For more information, see Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

If the preceding requirements are followed, and the HN-F SAM settings are consistent with the RN SAM, there are no further requirements for HN-F SAM programming. Each HN-F SAM can be programmed differently and can be mapped to a different SN.

6.5.1. HN-F SAM programming requirements for RN SAM non power-of-two hashing or hierarchical hashing use case

If an RN SAM SCG in your configuration uses either non-power-of-two hashing or hierarchical hashing, there are further requirements for the HN-F SAM programming for the corresponding HN-Fs. These requirements ensure that the HN-F SAM presents each SN with a uniform set of addresses.

If an SCG uses non-power-of-two hashing or hierarchical hashing to hash requests across HN-Fs, the following rules apply to the HN-F SAM for those HN-Fs:

- Per-HN-F direct SN mapping is not supported.
- For SCGs using non-power-of-two hashing, all HN-F nodes in the SCG must map to the same SN or group of SNs.
- For SCGs using hierarchical hashing, all HN-F nodes in either the same cluster or the overall SCG must access the same SN or group of SNs.

- If a cluster or SCG targets multiple SNs, each HN-F within it must:
    - Use the same hashing method
    - Use the same hashing attributes and address ranges
    - Populate the SN TgtID table in the same order
- If a cluster or SCG targets a single SN for all addresses:
    - The default hashed region must cover all addresses for that cluster or SCG
    - The SN0 TgtID value must be the same for each HN-F

For more information about non-power-of-two hashing and hierarchical hashing in RN SAM SCGs, see RN SAM non-hashed region configuration and Hierarchical hashing configuration for an SCG.

6.6. Program the HN-F SAM

Use this procedure to program the HN-F SAM registers according to your system requirements.

At reset, all HN-F SAM registers in the HN-F nodes are unconfigured. You must program the HN-F SAM registers in each HN-F, and each HN-F must be assigned at least one target SN.

Procedure

1. Program the HN-F SAM default hashed region registers for each HN-F according to the requirements for your use case:
    - cmn_hns_sam_control
    - cmn_hns_sam_control2
    - cmn_hns_sam_6sn_nodeid, if required for your use case.

   This step configures the SN mapping for the default hashed region, the properties for any hashing algorithm used, and the HN-F to SN mapping.

   If the HN-F SAM default hashed region is directly mapped to an SN, only program the SN0 target ID and its corresponding attributes. If the default hashed region hashes requests across multiple SN targets, program all SN target IDs, the attributes for each SN, the hashing properties, and any top address bits as required.

   For more information about the options for the HN-F SAM default hashed region, see HN-F SAM default hashed region.
2. Program the HN-F SAM range-based mapping registers for any HN-F that uses range-based mapping according to the requirements for your use case:
    - cmn_hns_sam_memregion{0,1}
    - cmn_hns_sam_nonhash_cfg1_memregion{2-63}

   This step configures the address region properties for the range-based non-hashed regions in the HN-F SAM and maps the address region to an SN.

   For more information about the options for the range-based non-hashed region, see HN-F SAM non-hashed region configuration.

3. Program the HN-F SAM range-based mapping registers for any HN-F that uses range-based mapping according to the requirements for your use case:

    - cmn_hns_sam_htg_cfg[1-3]_memregion[0-15]
    - cmn_hns_sam_htg_sn_nodeid_reg[0-15]

    This step configures the address region properties for the range-based hashed regions in the HN-F SAM, it selects the SN-mode for the HTG region, and it maps the HTG address region to the group of SNs. For more information about the options for the range-based hashed region, see HN-F SAM HTG configuration.

4. Program the SN property registers for all SNs according to the requirements for your use case:

    - cmn_hns_sam_sn_properties
    - cmn_hns_sam_sn_properties1
    - cmn_hns_sam_sn_properties2
    - cmn_hns_sam_nonhash_cfg2_memregion{2-63}, if the corresponding SN property fields were not already programmed in the preceding step.
    - cmn_hns_sam_htg_sn_attr{0-15}

    For more information about the different SN properties, see Configuration of HN-F SAM target SN properties.

7. RN SAM programming

Each RN in your configuration has a set of RN SAM configuration registers, which you use to configure the HN target for different system address regions.

The RN SAM configuration registers for CMN-700 RNs are in the XP that the RN is attached to. In other words, the SAM for all CMN-700 RNs is external to the RN. These RNs are also known as ESAM-type RNs. Therefore, the REQ.TgtID value that any RN-Fs generate is overwritten, or remapped, by the RN SAM in the XP when the REQ flit enters the mesh.

There are specific requirements that you must be aware of when programming the RN SAM. For more information, see RN SAM programming requirements.

The settings that the RN SAM uses to define address ranges are configuration dependent. For more information, see RN SAM address range definition methods.

The RN SAM implements three types of address region for mapping system addresses to a downstream target, the GIC memory region, non-hashed regions, and HTG regions. For more information about configuring these regions, see the following sections:

- RN SAM GIC region configuration
- RN SAM non-hashed region configuration
- RN SAM HTG configuration

The RN SAM implements other types of address regions to QoS override for requests to certain address regions and PrefetchTgt functionality. For more information, see RN SAM QoS override address region configuration and RN SAM PrefetchTgt functionality configuration.

When the RN SAM logic is programmed, there are some register settings that must be set to enable the RN SAM logic. For more information, see RN SAM programming complete indication.

For reference, we provide a procedure for reprogramming the RN SAM after the boot process has completed. For more information, see Reprogram the RN SAM after boot.

7.1. RN SAM programming requirements

All RN SAM instances in a CMN-700 configuration must be programmed consistently. In other words, the same HN target ID must be generated for a specific request address, regardless of which RN in the system issues the request.

Consistent RN SAM programming is required because in CMN-700, all HNs are the PoS, and HN-Fs are also the Point of Coherency (PoC). For ordering and coherency to be maintained, each system address or address range must be serviced by exactly one HN in CMN-700. Therefore, when presented with the same address, the output of all RN SAMs in the system must be the same.

7.2. RN SAM address range definition methods

Like the HN-F SAM, you can choose whether the CMN-700 RN SAM address ranges are defined using either the base address and the region size or a start address and end address. You select one of the two choices when you configure the CMN-700 RTL in Socrates.

The CMN-700 `RNSAM_HTG_RCOMP_EN` and `RNSAM_NONHASH_RCOMP_EN` configuration parameters configure how the interconnect defines RN SAM hashed and non-hashed address ranges respectively. These parameters support the following values:

FALSE

The region size is defined by specifying a base address for the region and the size of the region. The size of the region is fixed from 64MB to 4PB, in power-of-two increments. The base address must be aligned to the range size.

TRUE

The region size is defined by specifying a start address and end address for the region. The minimum size must be specified by setting `RNSAM_{NONHASH/HTG}_RCOMP_LSB` parameter to a value between 20-26, which correspond to a minimum size of 1MB to 64MB in power-of-two increments. The start address and the end address of the region must be aligned with the minimum size.

7.3. RN SAM GIC region configuration

In the CMN-700 RN SAM, the GIC memory region can be used to select a GIC-related address range and send all requests in that region to a single target. The GIC region has its own set of configuration registers in the RN SAM configuration space.

The GIC region has the highest priority in the RN SAM. Therefore, if an address matches both the GIC region and another RN SAM address region, the GIC target is chosen. Using the GIC region lets a GIC-targeted address region to be carved out of other hashed or non-hashed regions. The GIC region can overlap with hashed and non-hashed regions.

To set up the GIC region in the RN SAM, software must program the gic_mem_region_reg register. This register has the following configurable fields:

gic_region_valid

Specifies whether the GIC region is valid

gic_region_size

Specifies the size of the GIC region

gic_region_base_addr

Specifies the base address of the GIC region

gic_region_nodeid

Specifies the node ID of the HN-I that is connected to the GIC in your configuration

7.4. RN SAM non-hashed region configuration

In the CMN-700 RN SAM, non-hashed regions have a single target, to which all requests in that address range are sent. Each non-hashed region has its own set of configuration registers in the RN SAM configuration space.

The I/O space of the system memory map, which is typically serviced by HN-I, HN-D, HN-P, HN-V, and HN-T nodes, is intended to be the target of non-hashed regions. However, single HN-Fs can also be assigned as target nodes for a non-hashed region. If an HN-F is assigned as a target, then the address range would be backed by an SN-F or SBSX node. A RA node can also be assigned as a target for a non-hashed region for access to remote address ranges.

The number of non-hashed regions in the RN SAM is configuration dependent, according to the value of the `RNSAM_NUM_NONHASH_REGION` build-time configuration parameter. The CMN-700 RN SAM supports up to 64 non-hashed regions.

At least one non-hashed region must be specified in the RN SAM for the PERIPHBASE address range. The target of this region must be the HN-D. The PERIPHBASE address range contains 4 address ranges with separate functions:

- CFGM_PERIPHBASE range, which contains the CMN-700 configuration register addresses
- DSU_PERIPHBASE range, which contains the RN-F AXI5 Utility Bus (AXU) addresses
- DMC_PERIPHBASE range, which contains the SN-F AXU addresses
- MXP_AXU_PERIPHBASE, which contains the XP AXU addresses

The size of each of these address ranges determines the overall size of the PERIPHBASE address range. The base address values of these ranges are determined by CMN-700 configuration input signals.

The CFGM_PERIPHBASE range size depends on the size of the mesh:

- For meshes with X and Y dimensions that are less than or equal to 8, the CFGM_PERIPHBASE range spans 256MB.
- For meshes with X or Y dimensions that are greater than 8, the CFGM_PERIPHBASE range spans 1GB.

The DSU_PERIPHBASE, DMC_PERIPHBASE, and MXP_AXU_PERIPHBASE ranges can be calculated using the following formulas:

- DSU_PERIPHBASE = number of RN-F AXUs × 1MB, rounded up to the next power-of-two value, with a maximum range of 256MB.
- DMC_PERIPHBASE = number of SN-F AXUs × 16MB, rounded up to the next power-of-two value, with a maximum range of 256MB.
- MXP_AXU_PERIPHBASE = number of XPs × 1MB, rounded up to the next power-of-two value, with a maximum range of 256MB.

To set up non-hashed regions in the RN SAM, software must program the following registers:

- non_hash_mem_region_reg{0-63}

- non_hash_mem_region_cfg2_reg{0-63}
- non_hash_tgt_nodeid{0-15}

The non_hash_mem_region_reg{0-63} registers have the following configurable fields:

region{0-63}_valid
: Specifies if the region is valid

region{0-63}_target_type
: Specifies the target node type

region{0-63}_secure
: Specifies the security attribute for the region

region{0-63}_base_addr
: Specifies the start address of the region

region{0-63}_size
: Specifies the size of the memory region if the `RNSAM_RCOMP_NONHASH_EN` configuration parameter is set to `False`

The non_hash_mem_region_cfg2_reg{0-63} registers each have a nonhash_region{0-63}_end_addr field. This field specifies the end address of the address range if the `RNSAM_RCOMP_NONHASH_EN` configuration parameter is set to `True`.

To assign node ID targets to RN SAM non-hashed regions, software must program the non_hash_tgt_nodeid{0-15} registers. Each of these registers contains 4 fields to assign the node ID for 4 non-hashed regions. Which non-hashed regions a specific register controls is determined by its index value, from 0-15. The following formula determines which non-hashed regions each register controls:

- First region = 4 × index
- Last region = (4 × index) + 3

For example, the non_hash_tgt_nodeid0 register index is 0. Therefore, it contains the nodeid_{0-3} fields, which assign node IDs for non-hashed regions 0-3. Similarly, the non_hash_tgt_nodeid9 register index is 9. Therefore, it contains the nodeid_{36-39} fields, which assign node IDs for non-hashed regions 36-39.

7.5. RN SAM HTG configuration

In the CMN-700 RN SAM, HTGs distribute accesses in an address range across multiple HN-F, HN-P, or RA nodes. Each HTG has its own set of configuration registers in the RN SAM configuration space.

The number of HTGs in the RN SAM is configuration dependent, based on the value of the `RNSAM_NUM_HTG` configuration parameter. CMN-700 supports up to 32 HTGs in the RN SAM.

CMN-700 supports two types of HTG in the RN SAM:

- SCGs, which distribute requests across HN-F targets according to the address of the request.
- AxID-based hashing HTGs, which distribute requests across HN-F targets according to the request AxID value.

To define the HTG region, there are various properties that must be programmed in the configuration registers:

- The address ranges for the region. For more information, see RN SAM HTG address region configuration.
- The HN-F target IDs for all SCG-type HTGs. For more information, see RN SAM SCG HN-F target ID table configuration.
- The hashing properties for all SCG-type HTGs. For more information, see RN SAM SCG hashing methods.
- The properties for all HTGs that use AxID-based hashing. For more information, see AxID-based hashing HTGs.

7.5.1. RN SAM HTG address region configuration

There are various registers for configuring the RN SAM HTG address regions, depending on which region you are configuring.

The sys_cache_grp_region{0-3} registers contain the settings for HTGs 0-3, which are always configured as SCGs in CMN-700. These registers have the following fields for configuration:

region{0-3}_valid
: Specifies whether the region is valid

region{0-3}_target_type
: Specifies the target node type

region{0-3}_secure
: Specifies the security attribute for the region

region{0-3}_base_addr
: Specifies the start address of the region

region{0-3}_size
: Specifies the size of the memory region if the `RNSAM_RCOMP_HASH_EN` configuration parameter is set to `False`

The equivalent settings for HTGs 4-31 are in the hashed_tgt_grp_cfg1_region{4-31} registers.

The hashed_tgt_grp_cfg2_region{0-31} registers each have a region{0-31}_end_addr field. These fields specify the end address of the address range if the `RNSAM_RCOMP_HASH_EN` configuration parameter is set to `True`.

Each HTG also has an optional secondary address region that you can specify to match incoming addresses against. The address region settings for these regions are in the following registers:

- sys_cache_grp_secondary_reg{0-3}
- hashed_target_grp_secondary_cfg1_reg{4-31}
- hashed_target_grp_secondary_cfg2_reg{0-31}

7.5.2. RN SAM SCG HN-F target ID table configuration

To map HTGs to HN-F targets, the RN SAM implements a target ID table which contains the node IDs for all HTG targets in the configuration. The sys_cache_grp_hn_nodeid_reg{0-15} and hashed_target_grp_hnf_nodeid_reg{16-31} registers form the HTG HN-F target ID table.

Each of these registers contains fields to specify four HN-F node IDs, for a total of 128 HN-Fs. The HN-Fs are assigned to each HTG using the following register fields:

- sys_cache_group_hn_count.scg{0-3}_num_hnf
- sys_cache_group_hn_count.htg{4-7}_num_hn
- hashed_target_group_hn_count_reg{1-3}.htg{8-31}_num_hn

Each HTG is assigned HN-Fs from the target ID table registers sequentially. For example, if SCG0 hashes requests across 3 HN-Fs, the node IDs for these HN-Fs are provided by the sys_cache_grp_hn_nodeid_reg0.nodeid_{0-2} register fields. Then, if SCG 1 hashes requests across 8 HN-Fs, the node IDs for these HN-Fs are provided by the next register fields in the sequence, which would firstly be sys_cache_grp_hn_nodeid_reg0.nodeid_3, then sys_cache_grp_hn_nodeid1.nodeid{4-7}, and then sys_cache_grp_hn_nodeid2.nodeid{8-10}.

To select the HN-F target ID table as the target for an HTG, set the corresponding hashed_target_grp_hash_cntl_reg{0-31}.htg{0-31}_tgtid_sel field to 0.

7.5.3. RN SAM SCG hashing methods

SCGs distribute accesses to an address range across multiple HN-Fs. In the RN SAM, HTGs 0-3 are always defined as an SCG. Each SCG supports various ways of hashing across targets, depending on the number of HN-Fs and CAL targets in the SCG.

SCGs support the following hashing methods:

Power-of-two hashing
: The number of HN-Fs that an SCG distributes requests across is a power-of-two. This method is the default for an SCG.

Non-power-of-two hashing
: The number of HN-Fs that an SCG distributes requests across is not a power-of-two.

Hierarchical hashing
: HN-Fs are grouped into clusters and requests are hashed hierarchically. Firstly, requests are hashed across clusters of HN-Fs. Then, requests are hashed across the HN-Fs in a cluster.

The following figure shows an example hierarchical hashing mode configuration for an SCG.

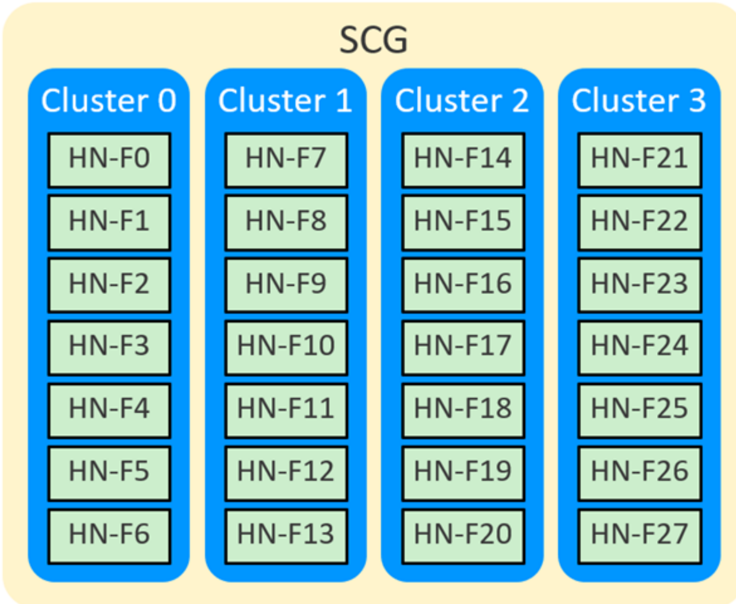

Figure 7-1 Hierarchical hashing configuration

There are also some other SCG configuration scenarios which have programming options or requirements.

If all the HN-Fs in an SCG are connected to CMN-700 through Component Aggregation Layer (CAL) components, then you can enable CAL mode for an SCG. If CAL mode is enabled, only the node ID for the lower HN-F in the CAL pair must be programmed in the target ID table for the SCG. CMN-700 ensures that the request reaches the correct HN-F in the CAL pair. Therefore, CAL mode simplifies the SCG setup requirements.

You can also configure an SCG to have a single HN-F target. There are two ways to assign a single HN target to an SCG. You can either program a single HN-F node ID into the SCG target ID table, or you can enable non-hashed mode for an SCG.

7.5.3.1. Power-of-two hashing configuration for an SCG

Power-of-two hashing is the default hashing method for RN SAM SCGs. Therefore, there are only a few actions required to enable power-of-two hashing.

For SCGs that use power-of-two hashing mode, software must ensure that the number of HN-F targets defined in the configuration registers is a power-of-two. To do this, it must program the following register fields as required for the use case:

- sys_cache_group_hn_count.scg{0-3}_num_hnf
- sys_cache_group_hn_count.htg{4-7}_num_hnf
- hashed_target_group_hn_count_reg{1-3}.htg{8-31}_num_hn

Software must also ensure that the target type of each SCG is set as an HN-F.

7.5.3.2. Non-power-of-two hashing configuration for an SCG

To configure an SCG to use non-power-of-two hashing, there are build-time configuration option requirements and programming requirements.

To use non-power-of-two hashing for the SCGs in your configuration, the CMN-700 `RNSAM_NP2_EN` configuration parameter must have been set to `1` when the interconnect was configured in Socrates. If the configuration parameter is set, then software can enable non-power-of-two hashing for any of the SCGs in the configuration.

To enable non-power-of-two hashing, software must program the hashed_target_grp_hash_cntl_reg{0-31}.htg_region{0-31}_nonpowerof2_hash_en field to 1 for each SCG that has a number of HN-F targets that is not a power-of-two.

7.5.3.3. Hierarchical hashing configuration for an SCG

To configure an SCG to use hierarchical hashing, there are build-time configuration option requirements and programming requirements.

When using hierarchical hashing, you divide the HN-Fs in an SCG across 2, 4, 8, 16, or 32 clusters. Each cluster supports a maximum of 32 HN-Fs and the number of HN-Fs can be a non-power-of-two. In an SCG, each cluster must contain the same number of HN-F nodes.

You can configure the number of clusters and the number of HN-Fs in each cluster at boot time for each SCG.

To use hierarchical hashing mode for the SCGs in your configuration, the CMN-700 `RNSAM_HIER_HASH_EN` configuration parameter must have been set to `1` when the interconnect was configured in Socrates.

To enable hierarchical hashing for an SCG, software must program the hashed_target_grp_hash_cntl_reg{0-31} register for the corresponding SCG. These registers have the following configurable fields:

htg_region{0-31}_hierarchical_hash_en
: Specifies whether hierarchical hashing is enabled

htg_region{0-31}_hier_enable_address_striping
: Specifies the number of address bits to remove when the address is hashed at the second level of hierarchy. This address bit removal is also known as address shuttering. The appropriate value of this field varies according to the number of clusters used in the first level of hierarchy, as set by the hier_hash_clusters field in this register.

htg_region{0-31}_hier_hash_clusters
: Specifies the number of clusters of HN-Fs

htg_region{0-31}_hier_hash_nodes
: Specifies the number of target HN-Fs in each cluster

htg_region{0-31}_hier_cluster_mask	Specifies the interleave granularity across clusters. This must be set to either 64B or 4KB granularity.

In hierarchical hashing, each physical memory address that is hashed at the first level of hierarchy is sent to a single cluster within the SCG. All the HN-Fs within the same cluster or all the HN-Fs within the same SCG must be configured to target either the same SN or set of SN targets using the same hashing configuration. When using hierarchical hashing, direct SN mapping to a single HN-F is not supported. Also, hierarchical hashing can cause unused sets in the HN-F SLC and SF in some use cases. For information on how to mitigate unused sets, see the *HN-F SLC and SF flexible addressing* section of the Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual.

7.5.3.4. CAL mode configuration

To configure CAL mode for an SCG, there are specific register programming requirements. There are also some further considerations that you must be aware of when an SCG uses CAL mode.

To use CAL mode for an SCG, all the HN-F nodes in an SCG must connect through a CAL. Also, for a single CAL, both HN-F nodes attached to a CAL must be in the same SCG.

To enable CAL mode for each SCG, software must set the corresponding register fields for each SCG to 1:

- sys_cache_grp_cal_mode_reg.scg{0-3}_hnf_cal_mode_en
- hashed_target_grp_cal_mode_reg{1-7}.htg{4-31}_hn_cal_mode_en

It must also specify the CAL type using the following register fields:

- sys_cache_grp_cal_mode_reg.scg{0-3}_hnf_cal_type
- hashed_target_grp_cal_mode_reg{1-7}.htg{4-31}_hn_cal_type

Also, the following register fields must be set to half the number of HN-F nodes in the SCG:

- sys_cache_group_hn_count.scg{0-3}_num_hnf
- sys_cache_group_hn_count.htg{4-7}_num_hnf
- hashed_target_group_hn_count_reg{1-3}.htg{8-31}_num_hn

CAL mode also has an option which you can use to modify whether the MSB or the Least Significant Bit (LSB) in the CAL target ID is used. This option can be configured by programming sys_cache_grp_cal_mode_reg.scg{0-3}_hnf_cal_bit_override or hashed_target_grp_cal_mode_reg{1-7}.htg{4-31}_hn_cal_bit_override. This option is only available if the SCG uses power-of-two hashing. If either non-power-of-two hashing or hierarchical hashing are used for the SCG, only the LSB bit can be selected as a NodeID modification bit.

Note

In Socrates, if you configure all HN-Fs to connect to CMN-700 through a CAL, Socrates automatically assumes that CAL mode is used. When Socrates renders the RTL for such a configuration, it only renders the number of registers required based on the number of HN-F CAL instances in the mesh (i.e. half the number of HN-F nodes). If CAL mode is not required in such a configuration, you must increase the number of hashed targets using `RNSAM_NUM_ADD_HASHED_TGT` to render enough registers for all the HN-Fs.

7.5.3.5. Non-hashed mode configuration for an SCG

To configure an SCG in non-hashed mode, there are specific register programming requirements. SCG0 cannot be configured to use non-hashed mode.

There are two benefits to using non-hashed mode for an SCG. Firstly, there are separate registers for the node ID target in non-hashed mode, so the single HN target does not use an SCG target ID table entry. Also, you can specify HN-I or an RA nodes as the SCG target in non-hashed mode, which is not possible in hashed mode.

When you configure an SCG as non-hashed, the RN SAM still uses the hashed TgtID selection priority.

To assign target node IDs for SCGs in non-hashed mode, software must program the sys_cache_grp_nonhash_nodeid and hashed_target_grp_nonhash_nodeid_reg{1-6} following registers, which are dedicated registers for non-hashed mode:

- sys_cache_grp_nonhash_nodeid.scg{1-3}_nodeid
- sys_cache_grp_nonhash_nodeid.htg{4,5}_nodeid
- hashed_target_grp_nonhash_nodeid_reg{1-6}.htg{6-35}_nodeid

7.5.4. AxID-based hashing HTGs

HTGs with AxID-based hashing are mainly used to hash requests from PCIe RN-Is and RN-Ds, which have por_{rni,rnd}_cfg_ctl.pcie_mstr_present set to 1, across multiple HN-P nodes. This method uses power-of-two hashing against the RN-I port and ACE-Lite AxID for request distribution.

An HTG that uses AxID-based hashing supports hashing requests over 1, 2, 4, 8, 16 or 32 HN-Ps, using the full AxID of the request and AXI port identifier. The hash algorithm calculates a pointer in the HN-P ID table in the RN SAM.

To enable AxID-based hashing within the HTG, software must program the hashed_target_grp_hash_cntl_reg{0-31}.htg_region{0-31}_axid_hash_en register field for the HTG to 1. Also, the node ID for all target HN-Ps must be programmed in the hashed_target_grp_hnp_nodeid_reg{0-15}.nodeid_{0-63} register fields.

Also, the hashed_target_grp_hash_cntl_reg{0-31}.htg{0-31}_tgtid_sel register field must be set to 1 to enable targets to be selected from the HN-P target ID table.

Only HTGs 4-31 can be configured to use AxID-based hashing because HTGs 0-3 are always SCGs in a CMN-700 configuration.

7.6. RN SAM QoS override address region configuration

The CMN-700 RN SAM supports up to 8 QoS override address regions, which override the incoming REQ.QoS value for requests to that region. Each QoS override address region has its own set of configuration registers in the RN SAM configuration space.

The RN SAM override regions are independent from each other and the override occurs downstream of the XP, RN-I, and RN-D QoS regulators.

Like the other regions, the QoS override can be defined using either the base address and the region size or a start address and end address. However, the method that the RN SAM uses depends on the `HNSAM_RCOMP_EN` configuration parameter.

QoS override only applies to addresses matching hashed or non-hashed regions. The QoS override is ignored in the following circumstances:

- The transaction address matches a programmed RN SAM GIC region.
- The transaction is a DVM operation.
- The RN SAM target for the transaction address is the default node ID.

There is no requirement for a QoS override region to align with hashed or non-hashed regions in the RN SAM. QoS override regions can span hashed and non-hashed region boundaries. However, a QoS override region must not overlap with another QoS override region.

The sam_qos_mem_region_reg{0-15} and sam_qos_mem_region_cfg2_reg{0-15} registers control the settings for the QoS override regions. The sam_qos_mem_region_reg{0-15} registers have the following configurable fields:

size
: Specifies the size of the QoS override region if the `HNSAM_RCOMP_EN` configuration parameter is set to `FALSE`

base_addr
: Specifies the start address of the region

qos
: Specifies the QoS override value

qos_override
: Specifies whether QoS override is enabled

valid
: Specifies whether the QoS override region is valid

The sam_qos_mem_region_cfg2_reg{0-15} registers each have an end_addr field, which specify the end address of the region if the `HNSAM_RCOMP_EN` configuration parameter is set to `TRUE`.

7.7. RN SAM PrefetchTgt functionality configuration

The RN SAM PrefetchTgt functionality lets an RN directly send Prefetch requests to the target SN, bypassing the HN. To configure the RN SAM PrefetchTgt functionality, there are build-time configuration option requirements and programming requirements.

The PrefetchTgt functionality uses Prefetch regions, which are memory regions that the RN SAM uses to derive the target ID for Prefetch requests. Prefetch regions are defined using configuration registers, but whether and how these configuration registers are rendered in the CMN-700 RTL depends on your configuration.

The `RNSAM_PREFETCH_EN` configuration parameter enables the PrefetchTgt functionality in your configuration. If this parameter is set to `TRUE`, the following configuration parameters take effect, which control how the Prefetch region registers are rendered:

`RNSAM_PFTGT_NUM_SCG`

Specifies the number of SCGs in the RN SAM that support Prefetch requests. Supports values from 0-8.

`RNSAM_PFTGT_NUM_NONHASH_PSCG`

Specifies the number of non-hashed regions in the SCG that support Prefetch requests. Supports values from 0-64.

`RNSAM_PFTGT_NUM_HTG_PSCG`

Specifies the number of HTGs in the SCG that support Prefetch. Supports values from 0-8.

The following tables show the registers that you must program to set up the PrefetchTgt functionality and the Prefetch regions. Any HN-F SAM-related information must be programmed consistently with the corresponding value in the HN-F SAM.

For direct-mapped or address-striped SN-F targets, these are the registers and the corresponding fields that are used to program the PrefetchTarget functionality in the RN SAM:

Table 7-1 RN SAM registers for Prefetch region programming to direct-mapped or address striped SN-F targets

RN SAM register	Field	Description	Related HN-F SAM register fields
sys_cache_grp_sn_attr, sys_cache_grp_sn_attr1	inv_top_address_bit_sys_cache_grp[0-7]	Inverts the top address bit for the Prefetch region. The top address bit is top_address_bit1 for 3-SN mode and top_address_bit2 for 6-SN mode. Top address bit inversion is only used when the address map does not have unique address bit combinations.	cmn_hns_sam_control. hn_cfg_sam_top_address_bit
	sn_hash_addr_bits_sel_sys_cache_grp[0-7]	Selects the SN hash address for the Prefetch region	cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel
	sn_mode_sys_cache_grp[0-7]	Specifies SN selection mode used in HN-F in an SCG for the Prefetch region	cmn_hns_sam_control, cmn_hns_sam_control2 (bits that enable any SN mode)
sys_cache_grp_sn_sam_cfg[0-3]	scg[0-3]_top_address_bit[0-2]	Specifies the top address bit for 3-SN, 5-SN, or 6-SN mode configuration for the Prefetch region	cmn_hns_sam_control. hn_cfg_sam_top_address_bit[0-2]
sys_cache_grp_sn_nodeid_reg0-31	sn_nodeid	Specifies the default hashed target SN node ID	cmn_hns_sam_control. hn_cfg_sn[0-3]_nodeid cmn_hns_sam_6sn_nodeid. hn_cfg_sn[4-7]_nodeid

For Non-hashed target regions in the HN-F, these are the registers and the corresponding fields that are used to program the PrefetchTarget functionality in the RN SAM:

Table 7-2 RN SAM registers for Prefetch region programming to Non-hashed target regions

RN SAM register	Field	Description	Related HN-F SAM register fields
sam_scg[0-7]_prefetch_nonhashed_ mem_region_cfg1_reg[[0-511]%64	scg[0-7]_prefetch_nonhash_reg#{index %64}_region_valid	Specifies whether the non-hashed Prefetch region is valid	-
	scg[0-7]_prefetch_nonhash_reg#{index %64}_qos_override_enable	Specifies whether QoS override is enabled for the non-hashed Prefetch region	-
	scg [0-7]_prefetch_nonhash_reg#{index %64}_qos_value	Specifies the QoS override value used for the non-hashed Prefetch region	-
	scg [0-7]_ prefetch_nonhash_reg#{index %64}_base_addr	Specifies the start address of the non-hashed Prefetch region	cmn_hns_sam_memregion[0-1].base_addr, cmn_hns_sam_nonhash_cfg1_memregion[2-7].base_addr
	scg [0-7]_ prefetch_nonhash_reg#{index %64}_size	Specifies the size of the non-hashed Prefetch region when the `RNSAM_RCOMP_NONHASH_EN` configuration parameter is set to `FALSE`	cmn_hns_sam_memregion[0-1].size, cmn_hns_sam_nonhash_cfg1_memregion[2-7].size
sam_scg[0-7]_prefetch_nonhashed_ mem_region_cfg2_reg[0-511]%64	scg#{index/64}_prefetch_nonhash_reg#{index %64}_tgtid	Specifies the SN target ID used for the non-hashed Prefetch region	cmn_hns_sam_memregion[0-1]_end_addr.range[0-1]_nodeid, cmn_hns_sam_nonhash_cfg1_memregion[2-7].range[2-7]_nodeid
	scg#{index/64}_prefetch_nonhash_reg#{index %64}_end_addr	Specifies the end address of the non-hashed Prefetch region when the `RNSAM_RCOMP_NONHASH_EN` configuration parameter is set to `TRUE`	cmn_hns_sam_memregion[0-1]_end_addr.end_addr cmn_hns_sam_nonhash_cfg1_memregion[2-63]

For HTG regions in the HN-F, these are the registers and the corresponding fields that are used to program the PrefetchTarget functionality in the RN SAM:

Table 7-3 RN SAM registers for Prefetch region programming to HTG regions

RN SAM register	Field	Description	Related HN-F SAM register fields
sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8	scg#{index/64}_prefetch_hashed_reg#{index %64}_region_valid	Specifies whether the hashed Prefetch region is valid	-
	scg#{index/64}_prefetch_hashed_reg#{index %64}_base_addr	Specifies the start address of the hashed Prefetch region	cmn_hns_sam_htg_cfg1_memregion[0-7].base_addr
	scg#{index/64}_prefetch_hashed_reg#{index %64}_size	Specifies the size of the region when the `RNSAM_RCOMP_HASH_EN` configuration parameter is set to `FALSE`	cmn_hns_sam_htg_cfg1_memregion[0-7].size
sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8	scg#{index/64}_prefetch_nonhash_reg#{index %64}_end_addr	Specifies the end address of the hashed Prefetch region when the `RNSAM_RCOMP_HASH_EN` configuration parameter is set to `TRUE`	cmn_hns_sam_htg_cfg2_memregion[0-7].end_addr
sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8	scg#{index/8}_prefetch_hashed_reg#{index %8}_top_address_bit[0-2]	Specifies the top address bit[0-2] of the Prefetch HTG region	cmn_hns_sam_htg_cfg2_memregion[0-7]. htg[0-7]_top_address_bit[0-2]
	scg#{index/8}_prefetch_hashed_reg#{index %8}_inv_top_address_bit	Inverts the top address bit for the HTG Prefetch region. The top address bit is top_address_bit1 for 3-SN mode and top_address_bit2 for 5-SN and 6-SN modes.	cmn_hns_sam_htg_cfg2_memregion[0-7]. htg[0-7]_inv_top_address_bit
	scg#{index/8}_prefetch_hashed_reg#{index %8}_sn_mode	Specifies the SN mode for the HTG Prefetch region	cmn_hns_sam_htg_cfg2_memregion[0-7]. htg[0-7]_sn_mode
	scg#{index/8}_prefetch_hashed_reg#{index %8}_sn_hash_addr_bits_sel	Specifies the SN hashed address bits for the HTG Prefetch region	cmn_hns_sam_htg_cfg2_memregion[0-7]. htg[0-7]_hash_addr_bits_sel

RN SAM register	Field	Description	Related HN-F SAM register fields
sys_cache_grp_hashed_regions_sn_nodeid_reg0-15	sn_nodeid	Specifies the target SN node ID for the HTG Prefetch region	cmn_hns_sam_htg_sn_nodeid_reg[0-15].sn_nodeid

Note

The non-hashed Prefetch regions in the RN SAM provide QoS override ability for all request types. To enable this feature, it is required that:

- The `RNSAM_PREFETCH_EN` configuration parameter is set to `TRUE`.
- The `RNSAM_PFTGT_NUM_SCG` configuration parameter is set to a supported value that is greater than `0`.
- The `RNSAM_PFTGT_NUM_NONHASH_PSCG` configuration parameter is set to a supported value that is greater than `0`.

You must also ensure that RNSAM_NUM_QOS_REGIONS is 0, because this feature cannot be used at the same time as the QoS override region.

7.8. RN SAM programming complete indication

The final required programming step for an individual RN SAM block is to clear the rnsam_status.use_default_node register field to 0 and set the rnsam_status.nstall_req register field to 1. This action enables the RN SAM logic and removes the default RN SAM behavior.

If the rnsam_status.use_default_node register field value remains at 1, the programmed RN SAM logic is not enabled. All requests receive the default node ID, pointing to the HN-D node, as the TgtID, regardless of the SAM programming.

Setting the rnsam_status.nstall_req register field has the following effects:

- Overrides the RNID_SAM_STALL_DIS_Sx_NIDy input signals on RN-I and RN-D nodes
- Unstalls requests received at these nodes, allowing them to be processed

7.9. Reprogram the RN SAM after boot

CMN-700 does not support reprogramming the RN SAM after boot, so this procedure is provided for reference only. Use this procedure with caution.

Warning

Arm has not validated this procedure, so we cannot guarantee that the RN SAM functionality works as intended after reprogramming. If you use this procedure, the behavior of CMN-700 is likely to be UNPREDICTABLE.

Prerequisites

Before reprogramming the RN SAM, all outstanding requests from any RNs in the system must have completed.

About this task

The RN SAM can either be reprogrammed by an RN in the mesh, or by a requester attached to the HN-D APB interface. If you use an RN in the mesh, you must ensure that only this RN can issue transactions into the mesh during reprogramming. The RN must also only access configuration registers and devices downstream of the HN-D node. It must not access any other locations during reprogramming.

After reprogramming, all physical addresses must map to the same SNs as before reprogramming. However, the mapping of physical addresses to HN-Fs can be changed.

Procedure

1. Quiesce all requesters so that no transactions are issued into the mesh, except for those required to reprogram the RN SAM.
2. Set the rnsam_status.use_default_node bit to 1 for the RN doing the reprogramming. This step is only required if you are using an RN to reprogram the RN SAM.

   This step ensures that the RN targets the RN SAM default node ID. The default node ID must always be the HN-D node ID and it cannot be changed.
3. Initiate an Address-Based Flush (ABF) CleanInvalid sequence in each HN-F for the entire coherent DRAM address space.

   This step flushes all RN-F caches and the CMN-700 System Level Cache (SLC). The ABF sequence must complete at every HN-F node before starting the next step.
4. Reprogram all the RN SAM instances in the configuration.
5. Allow RNs to resume requests into the mesh.

8. Example RN SAM and HN-F SAM programming

To show how to map a system memory map to specific SAM register settings, we provide an example memory map and the corresponding RN SAM and HN-F SAM configuration register programming for that memory map.

The following figure shows an example memory map with 1024GB addressable size based on a 40-bit proposed address map.

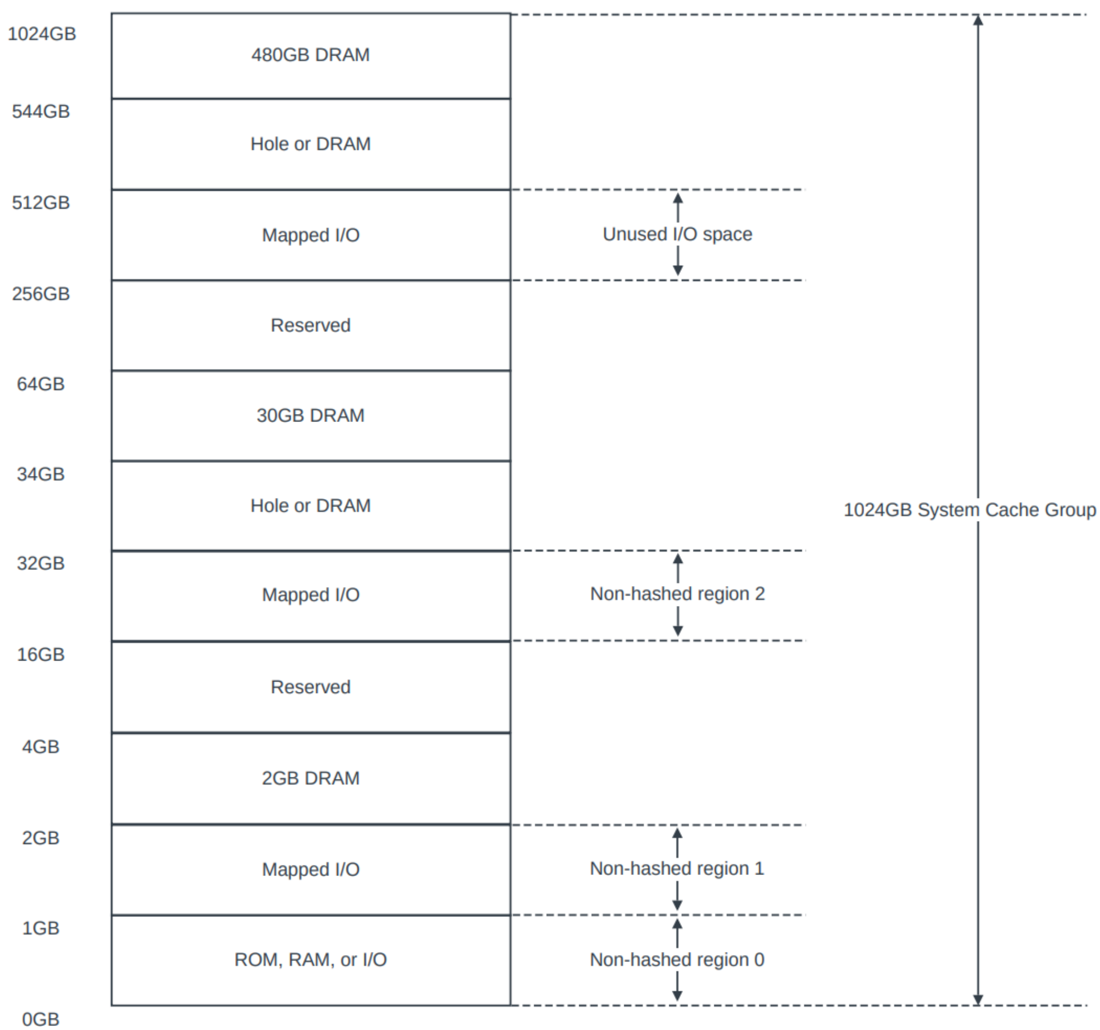

Figure 8-1 Example memory map

The memory map has three separate DRAM regions in the address ranges from 2-4GB, 34-64GB, and 544-1024GB. It also has four I/O regions, which are mapped to specific targets. It is assumed that the I/O region at the 256-512GB address range is unused and no requests are sent to this address.

It is assumed that there are eight HN-Fs in the system and all HN-Fs are in SCG0.

HN-F SAM programming

Like the RN SAM, the HN-F SAM must also be programmed so that it can select the correct SN-F target IDs. All HN-F SAMs within SCG 0 must have the same programming, including the attributes of each SN-F. The following table shows the HN-F SAM registers and required values to configure the example SCG for each HN-F SAM.

Table 8-1 HN-F SAM control register and programmed values

Register	Field	Value	Description
cmn_hns_sam_control	hn_cfg_sn0_nodeid	<sn0_node_id>	Node ID of SN-F 0
	hn_cfg_sn1_nodeid	<sn1_node_id>	Node ID of SN-F 1
	hn_cfg_sn2_nodeid	<sn2_node_id>	Node ID of SN-F 2
	hn_cfg_three_sn_en	1	Enable 3-SN mode
	hn_cfg_sam_top_address_bit0	39	Bit 39 of address
	hn_cfg_sam_top_address_bit1	36	Bit 36 of address
	hn_cfg_sam_inv_top_address_bit	1	Invert top address bit

RN SAM programming

To program this memory map into the RN SAM, we can divide the programming into several steps:

1. Map the full 1024GB memory map to the SCG. We recommend this mapping because DRAM regions are non-contiguous and the entire DRAM space is assigned to one SCG.
2. Carve out each non-hashed region from the full 1024GB memory map as shown in the preceding figure. Assign each non-hashed region to an individual non-hashed target. The non-hashed regions can overlap with the hashed SCG region.
3. When the RN SAM programming is complete, turn on the region-based target ID selection by disabling the default mode of the RN SAM.

The following table shows the RN SAM registers and required values to configure the example SCG for each RN SAM.

Table 8-2 Example RN SAM SCG registers and programmed values

Register	Field	Value	Description
sys_cache_grp_region0	region0_base_addr	`0x0_0000_0000`	Base address [47:16]
	region0_size	`0b0001110`	1024GB size
	region0_target_type	`0b000`	HN-F target type
	region0_valid	1	SCG region 0 is valid

Register	Field	Value	Description
sys_cache_grp_hn_nodeid_reg0	nodeid_0	<hnf0_node_id>	Physical node IDs of the HN-Fs in the system from HN-F 0 to HN-F 7
	nodeid_1	<hnf1_node_id>	
	nodeid_2	<hnf2_node_id>	
	nodeid_3	<hnf3_node_id>	
sys_cache_grp_hn_nodeid_reg1	nodeid_4	<hnf4_node_id>	
	nodeid_5	<hnf5_node_id>	
	nodeid_6	<hnf6_node_id>	
	nodeid_7	<hnf7_node_id>	
sys_cache_group_hn_count	scg0_num_hnf	0x08	Total of eight HN-Fs in this SCG

The following table shows the RN SAM registers and required values to configure the example non-hashed regions for each RN SAM.

Table 8-3 RN SAM non-hashed regions registers and programmed values

Register	Field	Value	Description
non_hash_mem_region_reg0	region0_base_address	0x0_0000_0000	1GB from [47:16] 0x0000_0000
	region0_size	0b0000100	1GB size
	region0_target_type	0b001	HN-I target type
	region0_valid	1	Non-hashed region 0 is valid
non_hash_mem_region_reg1	region1_base_address	0x0_0000_4000	1GB region from 0x4000_0000
	region1_size	0b0000100	1GB size
	region1_target_type	0b001	HN-I target type
	region1_valid	1	Non-hashed region 1 is valid
non_hash_mem_region_reg2	region2_base_address	0x0_0004_0000	16GB region from 0x4_0000_0000
	region2_size	0b0001000	16GB size
	region2_target_type	0b001	HN-I target type
	region2_valid	1	Non-hashed region 2 is valid
non_hash_tgt_nodeid0	nodeid_0	<hni0_node_id>	Node ID of HN-I 0 corresponding to non-hashed region 0
	nodeid_1	<hni1_node_id>	Node ID of HN-I 1 corresponding to non-hashed region 1

Register	Field	Value	Description
	nodeid_2	<hni2_node_id>	Node ID of HN-I 2 corresponding to non-hashed region 2

When the RN SAM block configuration is complete, the rnsam_status register must be programmed. The following table shows the rnsam_status register programmed values.

Table 8-4 RN SAM status register and programmed values

Register	Field	Value	Description
rnsam_status	nstall_req	1	Unstall any operations that depend on SAM programming.
	use_default_node	0	Disable default mode and use the programmed ranges for new incoming addresses.