



# Arm<sup>®</sup> Corstone<sup>™</sup> SSE-310 Example Subsystem

Revision: r0p0

## Technical Reference Manual

### Non-Confidential

Copyright © 2022 Arm Limited (or its affiliates).  
All rights reserved.

### Issue 01

102778\_0000\_01\_en



# Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

## Release information

### Document history

Issue	Date	Confidentiality	Change
0000-01	9 May 2022	Non-Confidential	First EAC release for r0p0

## Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

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

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

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

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

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

## Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

## Product Status

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

## Feedback

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

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

## Inclusive language commitment

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

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

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

# Contents

<b>1 Introduction.....</b>	<b>9</b>
1.1 Product revision status.....	9
1.2 Intended audience.....	9
1.3 Conventions.....	9
1.4 Additional reading.....	11
<b>2 Overview of SSE-310.....</b>	<b>13</b>
2.1 Document-specific conventions.....	14
2.2 Compliance.....	16
2.3 Product documentation.....	16
2.4 Product revisions.....	17
2.5 Arm architecture.....	17
2.6 Trusted Base System Architecture for Armv8-M.....	18
2.7 Interrupt controller architecture.....	19
2.8 Power Policy Unit architecture.....	19
2.9 Advanced Microcontroller Bus architecture.....	19
<b>3 Topology.....</b>	<b>20</b>
3.1 System Block Diagram.....	20
<b>4 Interfaces.....</b>	<b>22</b>
4.1 Input and output clocks.....	23
4.2 Reset inputs and outputs.....	25
4.3 P-Channel and Q-Channel Device interfaces.....	27
4.3.1 Clock Control Q-Channel Device interfaces.....	28
4.3.2 Power Control P-Channel Expansion Device interfaces.....	29
4.3.3 Power Control P-Channel Extension Device interfaces.....	30
4.3.4 Power Control Wakeup Q-Channel Device interfaces.....	30
4.4 Clock Control Q-Channel Control interfaces.....	31
4.5 Expansion Power Control Dependency interface.....	31
4.6 Power Domain ON Status Signals.....	32
4.7 System Timestamp Interface.....	33
4.8 Main Interconnect Expansion interfaces.....	34

4.9 Peripheral Interconnect Expansion interfaces.....	35
4.10 Interrupt interfaces.....	36
4.11 CPU Coprocessor interface.....	37
4.12 TCM Subordinate Interface.....	37
4.13 Debug and Trace Related Interfaces.....	38
4.13.1 Debug Access Interface.....	38
4.13.2 Serial Wire JTAG Interface.....	38
4.13.3 Debug Timestamp Interface.....	39
4.13.4 Cross Trigger Channel Interface.....	39
4.13.5 CPU0 External Peripheral Interface EPPB.....	39
4.13.6 ATB Trace Interfaces.....	40
4.13.7 Trace port interface.....	40
4.13.8 Debug Authentication Interface.....	40
4.13.9 Memory Protection Controller Expansion.....	41
4.13.10 Peripheral Interconnect Peripheral Protection Controller Expansion.....	42
4.13.11 Main Interconnect Peripheral Protection Controller Expansion.....	43
4.13.12 Manager Security Controller Expansion.....	44
4.13.13 Bridge Buffer Error Expansion.....	44
4.13.14 Other Security Expansion Signals.....	45
4.14 Clock configuration interfaces.....	45
4.15 Miscellaneous Signals.....	47
<b>5 Functional Description.....</b>	<b>48</b>
5.1 Clocking infrastructure.....	48
5.2 Reset infrastructure.....	51
5.2.1 Power-on and Cold reset Handling.....	51
5.2.2 CPU Reset Handling.....	52
5.2.3 Warm reset generation and control.....	52
5.2.4 Boot after reset.....	53
5.2.5 NPU reset handling.....	53
5.3 Cortex-M85 processor.....	53
5.3.1 EVENT Interfaces.....	58
5.3.2 Interrupts.....	58
5.4 System Interconnect infrastructure.....	59
5.4.1 ACC_WAIT Control.....	63
5.5 Volatile Memory.....	64

5.6 Timers and Watchdogs.....	64
5.6.1 Timestamp based Timers.....	65
5.6.2 SLOWCLK AON Timers.....	66
5.7 Power Policy Units.....	66
5.8 Peripheral Protection Controllers.....	67
5.9 Manager Security Controller.....	68
5.10 Memory Protection Controllers.....	68
5.11 Debug Infrastructure.....	68
5.11.1 DAP-Lite2.....	69
5.11.2 Debug timestamp generator.....	69
5.11.3 TPIU-M.....	69
5.11.4 MCU debug ROM table.....	70
5.12 System and Security Control.....	70
5.13 Power integration.....	71
5.13.1 Power domain hierarchy and bounded regions.....	71
5.13.2 Power domains.....	74
5.13.3 Power Policy Units.....	76
5.13.4 Bounded Region power modes.....	78
5.13.5 Wake-up sources.....	86
5.13.6 Power Dependency Control.....	87
5.13.7 System power states.....	89
5.14 NPU.....	93
5.14.1 NPU security mapping.....	94
<b>6 Programmer Model.....</b>	<b>97</b>
6.1 System Memory Map Overview.....	97
6.1.1 High level system address map.....	98
6.2 CPU TCM memories.....	104
6.3 Volatile Memory Region.....	105
6.4 Peripheral Region.....	105
6.4.1 Secure Access Configuration Register Block.....	108
6.4.2 Non-secure Access Configuration Register Block.....	123
6.4.3 Timestamp Timers.....	127
6.4.4 Timestamp Watchdogs.....	127
6.4.5 NPU registers.....	128
6.5 Processor Private Region.....	128

6.5.1 CPU0_PWRCTRL Register Block.....	129
6.5.2 CPU0_IDENTITY Register Block.....	130
6.5.3 CPU0_SECCTRL Register Block.....	131
6.6 System Control Peripheral Region.....	132
6.6.1 SYSINFO Register Block.....	133
6.6.2 System Control Register Block.....	137
6.7 CPU Private Peripheral Bus region.....	155
6.7.1 EWIC.....	157
6.7.2 Cortex-M85 TPIU-M registers.....	157
6.8 Debug System Access Region.....	157
6.9 Peripheral Expansion Region.....	157
<b>A Configurable render options.....</b>	<b>159</b>
A.1 Global configuration options.....	159
A.2 Global CPU Configuration Options.....	161
A.3 CPU0 Element Configuration Options.....	163
A.4 NPU0 Element Configuration Options.....	171
A.5 Main and Peripheral interconnect element configuration options.....	172
A.6 Power, Clock and Reset Control Configuration Options.....	177
A.7 Revision Information Configuration Options.....	178
A.8 Expansion Element Configuration Options.....	179
A.9 Rendering Control Configuration Options.....	180
A.10 IP Version Options.....	180
<b>B Revisions.....</b>	<b>187</b>



# 1 Introduction

## 1.1 Product revision status

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

<b><math>r_x</math></b>	Identifies the major revision of the product, for example, $r1$ .
<b><math>p_y</math></b>	Identifies the minor revision or modification status of the product, for example, $p2$ .

## 1.2 Intended audience

This book is written for electronics engineers that are designing, configuring, laying out, validating, and verifying the Arm® SSE-310 Example Subsystem. It is assumed that the primary readers have high technical ability and have experience of Verilog. The documentation does not assume experience of Arm devices or implementation, but it is likely that the primary readers have some prior experience of Arm IP. It is assumed that they are familiar with all necessary implementation tools and compute resources. It is also assumed that the readers have access to all necessary tools.

## 1.3 Conventions

The following subsections describe conventions used in Arm documents.

### Glossary







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

See the Arm® Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

### Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.

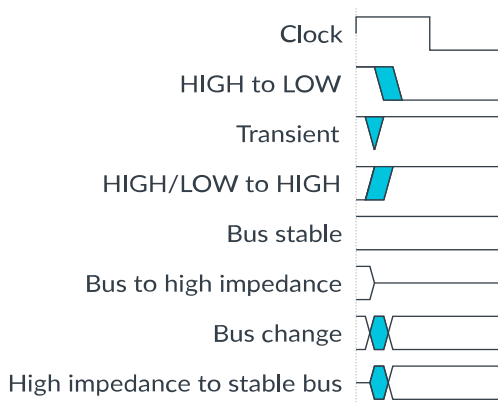
Convention	Use
<b>bold</b>	Interface elements, such as menu names.  Signal names.  Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
<b>monospace bold</b>	Language keywords when used outside example code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example:  <pre>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</pre>
<b>SMALL CAPITALS</b>	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .
 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.
 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.

## Timing diagrams

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

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

**Figure 1-1: Key to timing diagram conventions**



## Signals

The signal conventions are:

### Signal level

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

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

### Lowercase n

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

## 1.4 Additional reading

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

**Table 1-2: Arm publications**

Document Name	Document ID	Licensee only
AMBA® 4 ATB Protocol Specification, ATBv1.0 and ATBv1.1	IHI 0032	No
AMBA® 5 AHB Protocol Specification	IHI 0033	No
AMBA® APB Protocol Specification	IHI 0024	No
AMBA® AXI and ACE Protocol Specification AXI3, AXI4, AXI5, ACE and ACE5	IHI 0022F	No
AMBA® Ethos™-U55 NPU Technical Reference Manual	102420	No
AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces	IHI 0068	No

Document Name	Document ID	Licensee only
Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual	101150	No
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	No
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual	101526	No
Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual	101375	No
Arm® CoreSight™ Architecture Specification v3.0	IHI 0029	No
Arm® CoreSight™ DAP-Lite2 Configuration and Integration Manual	100589	Yes
Arm® CoreSight™ DAP-Lite2 Technical Reference Manual	100572	No
Arm® Coresight™ TPIU-M Technical Reference Manual	102427	No
Arm® Corstone™ Reference Systems Architecture Specification Ma1	102803	No
Arm® Corstone™ SSE-310 Example Subsystem Configuration and Integration Manual	102779	Yes
Arm® Corstone™ SSE-310 Example Subsystem Release Note	107556	Yes
Arm® Cortex®-M85 Processor Integration and Implementation Manual	101925	Yes
Arm® Cortex®-M85 Processor Technical Reference Manual	101924	No
Arm® Cortex®-M85 Processor User Guide Reference Material	101927	No
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074	No
Arm® Ethos™-U55 NPU Configuration and Integration Manual	101887	Yes
Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M	DEN 0083	No
Arm® Power Policy Unit Architecture Specification	DEN 0051	No
Arm®v8-M Architecture Reference Manual	DDI 0553	No



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 Overview of SSE-310

This document specifies the Corstone SSE-310 Internet of Things (IoT) Example Subsystem (SSE-310).

SSE-310 integrates the following key components available from Arm that can be integrated into a larger system:

- Cortex-M85 processor core with optional MVE, FPU, DSP, extensions, caches, TCMs and ETM
- Arm® Ethos™-U55 neural processing unit (NPU)
- SSE-310 supports only one Arm® Cortex-M85 processor
- SSE-310 supports two volatile memory (VM) banks, for example SRAMs
- Memory Protection Controllers (MPC)
- Exclusive Access Monitor (EAM)
- System interconnect
- Implementation Defined Attribution Unit (IDAU)
- CMSDK timers and watchdog timers
- Timestamp based system timers and watchdog timers
- Subsystem controllers for security and general system control
- Power Policy Units, Clock Controller and Low Power Interface interconnect components (PCK-600)

The above components are integrated to implement SSE-310 with the following features:

- TrustZone® aware system with the system segregated into Secure and Non-secure worlds
- Configurability to allow several features within the system to be included or removed
- Power Control infrastructure with several pre-defined voltage and power domains
- Each switchable power domain has a local power policy control, and coordinates with other power domains through a centralized dependency control or power interfaces. Switchable power domains provide the system with autonomous dynamic power control infrastructure that, while being software configurable, aiming to minimize software interaction.
- Clock control infrastructure that supports high level clock control including dynamic clock gating and provides clock request handshakes to clock generators
- Comprehensive reset generation and control
- A CoreSight SoC-based debug infrastructure that supports:
  - A shared Debug Access Port (without example expansion logic)
  - A JTAG/SW debug port (with example expansion logic)
  - Trace Port
  - Cross-triggering

## 2.1 Document-specific conventions

In addition to the Typographic Conventions section, there are document-specific conventions that apply.

### Text

Text between < and > is a label to be replaced with the actual name or value of the item. For example, CPU<n> where “n” is an instance number of either 0 or 1.

Text between { and } indicates the legal values. The allowed values can be either:

- A range indicate by a start and end with a “-” or in-between. For example, {0-3} allows the any value between 0 and 3. One of the numbers can also be a variable. For example, {0-<NUMCPU>} where NUMCPU is a configuration value between 0 to 3.
- A list of discrete values indicated by a comma separate list. For example, {0,1,2,3} allows the value to be any one of those values. One of the discrete values can also be a range. For example, {0, 3-6, 9} is the same as writing {0, 3, 4, 5, 6, 9}.
- A variable which has constraints. For example, {x} where “x” is between 0 and 3. Allows “x” to take any value between 0 and 3.

### Numbers

Numbers are normally written in decimal. Binary numbers are preceded by 0b, and hexadecimal numbers by 0x.

For both binary and hexadecimal numbers, where a bit is represented by the letter “x”, the value is irrelevant. For example, a value expressed as 0b1x can be either 0b11 or 0b10.

### Signals

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

- HIGH for active-HIGH
- LOW for active-LOW

A lowercase ‘n’ at the start or end of a signal name denotes an active-LOW signal.

The value of a multi-bit signal is defined using hexadecimal numbers.

When referring to bits within of a multi-bit signal, the following custom is used:

- **<SIGNAL\_NAME>[BIT\_RANGE]**

Where:

- SIGNAL\_NAME is the name of the signal being referred to.
- BIT\_RANGE is the bit range being referred to. If BIT\_RANGE is omitted, then the text is referring to the whole signal.

For example, when referring to the bit range 31:2 of Debug Access Interface address signal, the text would read:

- **DEBUGPADDR[31:2]**

## Registers

When referring to registers and fields, the following convention is used:

- **<REGISTER\_NAME>.<FIELD\_NAME>[BIT\_RANGE]**

Where:

- REGISTER\_NAME is the short name of the register being referred to.
- FIELD\_NAME is the name of the field within the register being referred to. If the FIELD\_NAME is omitted, then the text is referring to the whole register.
- BIT\_RANGE is the bit range, within the field, being referred to. If the BIT\_RANGE is omitted, then the text is referring to the whole field or to the whole register if FIELD\_NAME is omitted as well. When referring to the bit ranges of a field, the field starts at 0.

For example, when referring to the DESIGNER\_ID field of the SYSVERSION register, bits 1:0, the text would read:

- **SYSVERSION.DESIGNER\_ID[1:0]**

In register bit field description tables, the following abbreviations are used to describe the accessibility of each bit field:

**Table 2-1: Register bit field abbreviations**

Abbreviation	Accessibility	Description
RW	Read and Write Accessible	Unless stated, these bit fields retain the value written to it until reset or powering down.
RO	Read only	These can only be read. Unless stated, any writes to these bit fields are ignored. This is similar to WI.
RAZ	Read as Zero	These always return Zeros for reads. Unless stated, writes proceeds as normal.
WI	Write Ignore	These ignores writes. Unless stated, reads proceeds as normal. This is similar to RO.
<b>RAZWI</b>	Read as Zero Write Ignores	These always return Zeros for read and ignores writes.
RAZWO	Read as Zero Write Only	These can only be written. These always returns Zeros for reads.
WO	Write only	These can only be written. Unless stated, any read from these bit fields returns zeros.
W1S	Write 1 to Set	Each bit when written with one is set to one. Writing zeros to these are ignored.
W1C	Write 1 to Clear	Each bit when written with one is cleared to zero. Writing zeros to these are ignored.
W0S	Write 0 to Set	Each bit when written with zero is set to one. Writing ones to these are ignored.
W0C	Write 0 to Clear	Each bit when written with zero is cleared to zero. Writing ones to these are ignored.

Unless stated otherwise, after a register bit field is modified by a register access, it retains its values until a reset is applied or power is lost.

Register values are defined using actual values that are written in or read from the registers. They are expressed as numbers, either as binary numbers or hexadecimal numbers. Alternatively, for single-bit values, they can be expressed as either 1 or 0, representing 0b1 and 0b0 respectively.

## 2.2 Compliance

The Arm SSE-310 described in this document complies with, or includes components that comply with, the following specifications:

- *Arm®v8-M Architecture Reference Manual*
- *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M*
- *Arm® Power Policy Unit Architecture Specification*
- *AMBA® AXI and ACE Protocol Specification AXI3, AXI4, AXI5, ACE and ACE5*
- *AMBA® 5 AHB Protocol Specification*
- *AMBA® 4 ATB Protocol Specification, ATBv1.0 and ATBv1.1*
- *AMBA® APB Protocol Specification*
- *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces*
- *Arm® Corstone™ Reference Systems Architecture Specification Ma1*

This Technical Reference Manual complements the TRMs for included components, architecture reference manuals, architecture specifications, protocol specifications, and relevant external standards. It does not duplicate information from these sources.

## 2.3 Product documentation

Each SSE-310 document has an intended audience and is associated with specific tasks in the design flow.

These documents do not reproduce SSE-310 architecture and protocol information. For relevant protocol and architectural information that relates to this product, see [Additional reading](#).

The SSE-310 documentation is as follows:

### Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the SSE-310. It is required at all stages of the design flow. The choices that are made in the design flow can mean that some behaviors that the TRM describes are not relevant.

If you are programming the SSE-310 contact:

- The implementer to determine:
  - The build configuration of the implementation



- What integration, if any, was performed before implementing SSE-310
- The integrator to determine the signal configuration of the device that you use

The TRM complements architecture and protocol specifications and relevant external standards. It does not duplicate information from these sources.

## Configuration and Integration Manual

The CIM describes:

- The available build configuration options
- How to configure the Register Transfer Level (RTL) with the build configuration options
- How to integrate SSE-310 into an SoC
- How to implement SSE-310 into your design
- The processes to validate the configured design



The CIM is a confidential book that is only available to licensees.

---

## 2.4 Product revisions

There can be differences in functionality between different product revisions. Arm records these differences in this section.

### r0p0

First release

## 2.5 Arm architecture

The Arm® Cortex-M85 Processor in SSE-310 implements the Armv8-M architecture, which executes the Armv8-M T32 instruction set.

For more information, see *Arm®v8-M Architecture Reference Manual*.

## 2.6 Trusted Base System Architecture for Armv8-M

The *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M* (TBSA-M) is part of the Arm Platform Security Architecture (PSA).

TBSA-M implements best practice security principles when designing systems around Armv8-M processing elements (PEs). For more details, see *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M*.

SSE-310 partly fulfils the requirements specified within the TBSA-M. However, it implements features that help to form the core of a system that complies to the TBSA-M:

- Implements a system architecture that partitions the memory areas into Secure and Non-secure areas
- Implements SIE-300 MPCs to allow mapping of system volatile memory regions to be shared between the Secure and Non-secure world. The Security configuration of MPCs can only be changed by Secure accesses.
- Implements SIE-200 PPCs to allow mapping of peripherals to be shared between Secure and Non-secure world. The Security configuration of PPCs can only be changed by Secure accesses.
- Implements system security control registers and providing expansion control and status interfaces to allow map external memory regions and external peripherals shared between both worlds
- Implements an always on Secure watchdog

For a system that integrates SSE-310 to comply with TBSA-M, when expanding the system the integrator must comply with TBSA-M requirements. For example:

- Root of Trust (RoT)
- A protected keystore
- A Secure firmware update mechanism
- A lifecycle management mechanism, for Secure control of debug, test, and access to provisioned secrets
- A high-entropy random number generator, for reliable cryptography
- Cryptographic acceleration
- When adding more managers and subordinates to the system, the memory space continues to obey Trusted and Non-trusted world partitioning

Adherence to TBSA-M requirements helps to create a Secure system, but it does not create a system that mitigates Denial of Service (DoS) attacks. As such, DoS is out of scope of SSE-310. Countermeasures are not implemented at the Arm Soft-IP subcomponent level for the following events unless specifically stated in the specification:

- Physical and board-level attacks
- Physical side channels
- Fault injection attacks

## 2.7 Interrupt controller architecture

SSE-310 implements the following features:

- Arm Nested Vectored Interrupt Controller (NVIC)
- Arm External Wakeup Interrupt Controller (EWIC)

### See also

- *Arm® Cortex®-M85 Processor Technical Reference Manual*
- *Arm® Cortex®-M85 Processor User Guide Reference Material*

## 2.8 Power Policy Unit architecture

The power domains in SSE-310 are controlled by Power Policy Units (PPUs), which comply with the Arm PPU architecture.

See *Arm® Power Policy Unit Architecture Specification* for more information.

## 2.9 Advanced Microcontroller Bus architecture

SSE-310 complies with the following protocols:

- Advanced Extensible Interface (AXI5) protocol
- Advanced High Performance Bus (AHB5) protocol
- Advanced Peripheral Bus (APB4) protocol

### See also

- *AMBA® AXI and ACE Protocol Specification AXI3, AXI4, AXI5, ACE and ACE5*
- *AMBA® 5 AHB Protocol Specification*
- *AMBA® APB Protocol Specification*

## 3 Topology

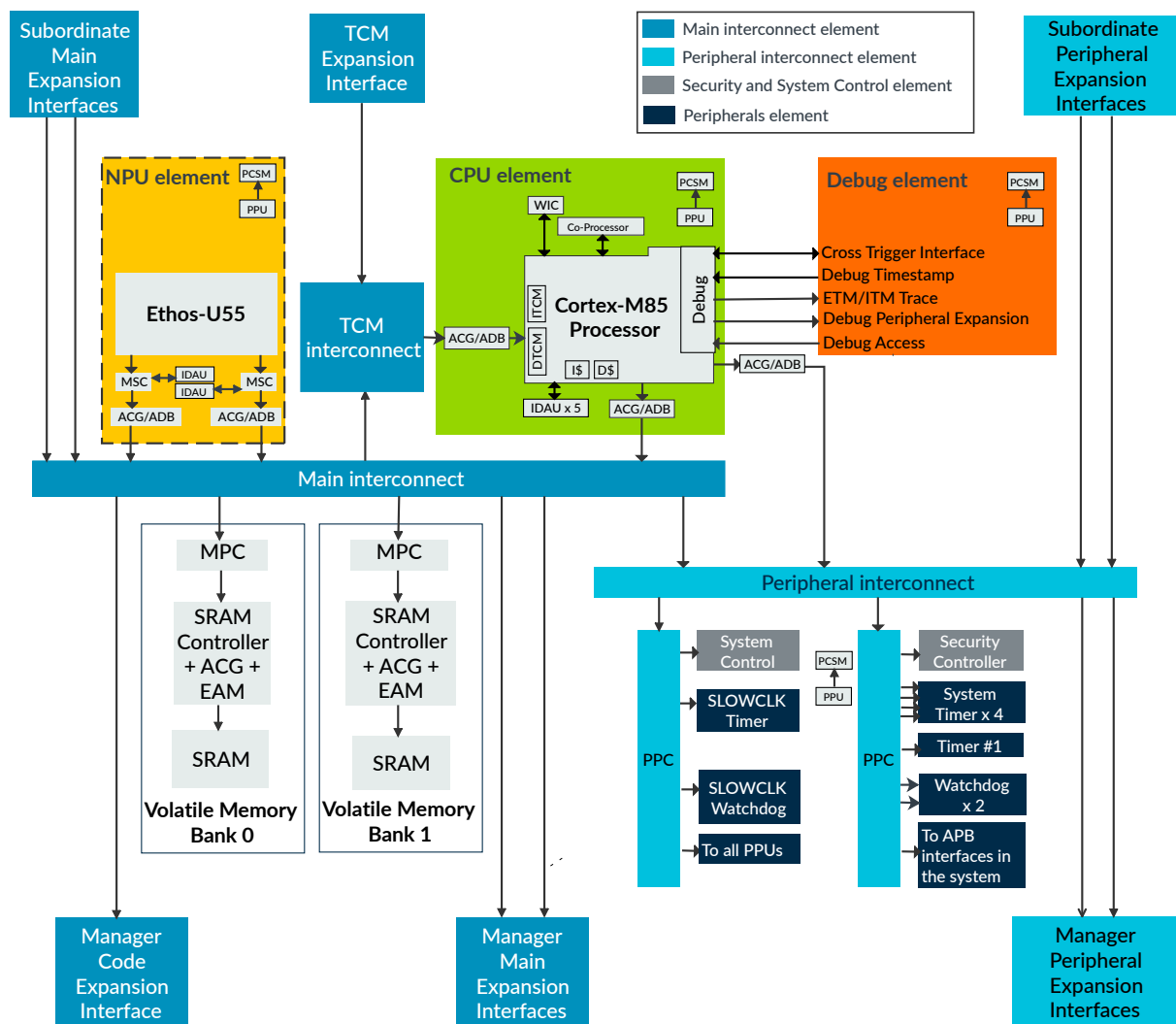
This section describes the topology of SSE-310.

### 3.1 System Block Diagram

The system block diagram show the topology of SSE-310.

The following figure shows a representative system block diagram of the top-level in SSE-310.

**Figure 3-1: SSE-310 logical structured system topology**



The subsystem is divided up into the following key groups of functionalities, which in this document is also referred to as *elements*.



Elements are used in this document simply to group functionalities that are closely related to each other or because of their common configurability.

---

### **NPU element**

The NPU element contains an Ethos-U55 NPU and its associated private infrastructure to integrate the NPU into the system. The NPU is optional. SSE-310 supports one NPU within the system.

### **CPU element**

The CPU element contains an Armv8.1-M processor CPU and its associated private infrastructure to integrate the core into the system. SSE-310 supports one CPU within the system.

### **Main interconnect element**

Connects all parts of the system.

### **Peripheral interconnect element**

Provides access to lower performance and often device type peripherals within the system.

### **TCM interconnect element**

Provides access from the TCM subordinate interface and from the Main interconnect to the Tightly Coupled Memories (TCM) internal to CPU0.

### **Volatile memory bank element**

Volatile memory banks collectively implement the main volatile storage within the subsystem and implement the functionality (Memory Protection Controller (MPC), Exclusive Access Monitor (EAM), SRAM control, Access Control Gate (ACG) and RAM wrappers) to manage the volatile storage.

### **Peripherals element**

Defines all common sets of peripherals expected in the system.

### **Security and system control element**

Defines the infrastructure required to implement all configure, control, and monitor system states. These include security, clock, reset, and power control.

### **Debug system element**

Defines the debug infrastructures that allows the processor and the system to be debugged securely.

SSE-310 provides several render configuration options. [Configurable render options](#) describes the legal (supported) configuration options.

## 4 Interfaces

The subsystem has several interfaces. This section provides the associated properties of each interface such as address and data width, along with the clock, power and reset domain that each belongs to.

In this section the following conventions are used:

### AMBA Manager Interface

An AMBA interface that is described as a manager interface is one where the subsystem is the manager and must be connected to a subordinate interface.

For an AXI manager interface the **ARVALID** signal is an output and **ARREADY** is an input.

For an AHB manager interface the **HADDR** signal is an output and **HRDATA** is an input.

For an APB manager interface, the **PADDR** signal is an output.

### AMBA Subordinate Interface

An AMBA interface that is described as a subordinate interface is one where the subsystem is the subordinate and must be connected to a manager interface.

For an AXI subordinate interface the **ARVALID** signal is an input and **ARREADY** is an output.

For an AHB subordinate interface the **HADDR** signal is an input and **HRDATA** is an output.

For an APB subordinate interface, the **PADDR** signal is an input.

### Q-Channel Control Interface

A Q-Channel interface described as a control interface, is one where the subsystem is the device and must be connected to a control interface.

For a Q-Channel control interface the **QREQn** signal is an input and **QACCEPTn**, **QDENY** and **QACTIVE** are outputs.

### Q-Channel Device Interface

A Q-Channel interface described as a device interface, is one where the subsystem is the controller and must be connected to a device interface.

For a Q-Channel device interface the **QREQn** signal is an output and **QACCEPTn**, **QDENY** and **QACTIVE** are inputs.

### P-Channel Control Interface

A P-Channel interface described as a control interface, is one where the subsystem is the device and must be connected to a control interface.

For a P-Channel control interface, the **PREQ** signal is an input and **PACTIVE**, **PSTATE**, **PACCEPT** and **PDENY** are outputs.

## P-Channel Device Interface

A P-Channel interface described as a device interface, is one where the subsystem is the controller and must be connected to a device interface.

For a P-Channel device interface, the **PREQ** signal is an output and **PACTIVE**, **PSTATE**, **PACCEPT** and **PDENY** are inputs.

## 4.1 Input and output clocks

SSE-310 has multiple input and output clocks that are described in this section.

**Table 4-1: SSE-310 input clocks**

ID	Power domain <sup>1</sup>	Description
<b>SLOWCLK</b>	PD_AON	<p>Slow clock. An always active slow clock that is asynchronous to the other clocks in the subsystem. <b>SLOWCLK</b> is one of the two clocks expected to be active in the lowest power state of the subsystem: HIBERNATE0. <b>SLOWCLK</b> is used primarily by timers residing in the PD_AON power domain.</p> <p>External gating of <b>SLOWCLK</b> is not supported by clock Q-Channels.</p>
<b>AONCLK</b>	PD_AON	<p>Always on clock is used for the low frequency logic in the PD_AON power domain that is not running on the <b>SLOWCLK</b> clock, such as the External Wakeup Interrupt Controller (EWIC) and the MGMTPPU. This allows a part of the PD_AON domain to run at a faster clock and be independent from the rest of the system. <b>AONCLK</b> is asynchronous to the other clocks in the system.</p> <p>If the corresponding clock Q-Channel is in Q_STOPPED state, then <b>AONCLK</b> can be gated externally.</p>
<b>CNTCLK</b>	PD_AON	<p>System Counter Timestamp clock associated with the <b>CNTVALUEB</b> System Counter Timestamp input. <b>CNTCLK</b> is asynchronous to the other clocks in the subsystem.</p> <p>External gating of <b>CNTCLK</b> is not supported by clock Q-Channels.</p>
<b>SYSCLK</b>	PD_AON	<p>Main System clock. This clock is the main clock to drive the main system that resides in PD_SYS. <b>SYSCLK</b> is also used for power management logic in PD_AON that is not running on <b>AONCLK</b>. <b>SYSCLK</b> is completely asynchronous to the other clocks in the subsystem except for <b>CPU0CLK</b>, and <b>NPU0CLK</b>.</p> <p>The clock source is gated internally by PPU before it is used by any logic.</p>
<b>CPU0CLK</b>	PD_AON	<p>CPU0 clock. This is the main clock used to drive the logic residing in the PD_CPU0 and PD_DEBUG domains. <b>CPU0CLK</b> clock is asynchronous to the other clocks in the subsystem except for <b>SYSCLK</b>, and <b>NPU0CLK</b>.</p> <p>If the corresponding clock Q-Channel is in Q_STOPPED state, then <b>CPU0CLK</b> might be externally gated.</p> <p>The clock source is gated internally by PPU before it is used by any logic.</p>
<b>NPU0CLK</b>	PD_AON	<p>NPU0 clock. This is the clock used to drive the logic reside in the PD_NPU0 domain.</p> <p><b>NPU0CLK</b> is asynchronous to the other clocks in the subsystem except for <b>SYSCLK</b> and <b>CPU0CLK</b>.</p> <p><b>NPU0CLK</b> might be externally gated if the corresponding clock Q-Channel is in Q_STOPPED state.</p> <p>The clock source is gated internally by PPU before it is used.</p>

<sup>1</sup> *Power Domain* refers to the power domain where this clock is expected to be used in. All clocks always first enter via the PD\_AON domain, before being used in their respective power domain.

In typical use, Arm recommends to drive **SLOWCLK** using a 32kHz clock source. If reducing standby power is an important consideration for a product, **AONCLK** can be driven at a lower clock rate compared to **SYCLK** (around 1MHz to 10MHz) to improve the transition time required to enter and leave the lowest power state of the subsystem, but still support the use of very low leakage implementation library cells. Alternatively, **AONCLK** can be driven using the same clock source as **SYCLK**. If there is no requirement to run the System Timestamp Counter at a different speed to the subsystem, then the **CNTCLK** can also be driven using the same clock source as that of **SYCLK**.

At minimum, two clock sources are needed. For example, one at 32kHz for **SLOWCLK** and another at 400MHz for the other clocks. Since **SYCLK**, **CPU0CLK**, and **NPU0CLK** must have the same frequency, if very large SRAMs with higher access time are needed in the subsystem, the frequency of both **SYCLK**, **CPU0CLK**, and **NPU0CLK** has to be reduced.

SSE-310 output clocks are defined in the following table. The clock outputs are synchronous to each other.

**Table 4-2: SSE-310 output clocks**

ID	Power domain <sup>1</sup>	Description
COMGMTSYCLK	PD_AON	The gated version of <b>SYCLK</b> . It is expected to be used to drive expansion logic that resides in the PD_AON power domain and is reset by <b>nCOLDRESETMGMT</b> .
MGMTSYCLK	PD_AON	The gated version of <b>SYCLK</b> . It is expected to be used to drive expansion logic that resides in the PD_AON power domain and is reset by <b>nWARMRESETMGMT</b> .
COSYSSYCLK	PD_SYS	The gated version of <b>SYCLK</b> , expected to be used to drive expansion logic that resides in the PD_SYS power domain and is reset by <b>nCOLDRESETSYS</b> .
SYSSYCLK	PD_SYS	The gated version of <b>SYCLK</b> , expected to be used to drive expansion logic that resides in the PD_SYS power domain and is reset by <b>nWARMRESETSYS</b> .
COCPU0CLK	PD_CPU0	The gated version of <b>CPU0CLK</b> , expected to be used to drive expansion logic that resides in the PD_CPU0 power domain and is reset by <b>nCOLDRESETCPU0</b> .
CPUCPU0CLK	PD_CPU0	The gated version of <b>CPU0CLK</b> , expected to be used to drive expansion logic that resides in the PD_CPU0 power domain and is reset by <b>nWARMRESETCPU0</b> .
DEBUGCPU0CLK	PD_DEBUG	The gated version of <b>CPU0CLK</b> , expected to be used to drive expansion logic that resides in the PD_DEBUG power domain.

<sup>1</sup> *Power Domain* refers to the power domain where this clock is expected to be used in. All clocks always first enter via the PD\_AON domain before being used in their respective power domain.

SSE-310 provides a Q-Channel interface for each of the output clocks to allow expansion logic to control the availability of each clock output.

For more information, see [Clock Control Q-Channel Device interfaces](#).



## 4.2 Reset inputs and outputs

Reset inputs are used to drive or to request for resets while reset outputs are used to reset expansion logic that resides in specific power domains.

The following table shows the reset inputs and outputs at SSE-310 top level. For more information, see [Reset infrastructure](#).

**Table 4-3: Top-level reset inputs and outputs**

Signal	Direction	Power domain <sup>1</sup>	Description
<b>nPORESET</b>	Input	PD_AON	<p>An active-LOW reset for power up the system. <b>nPORESET</b> resets all registers in the design. This includes resetting the Reset syndrome register.</p> <p>There is no functional reset duration requirement for this reset input. However, SSE-310 recommends that the reset is at least three <b>SLOWCLK</b> cycles long.</p> <p>This reset input performs reset asynchronously. The deassertion of the reset is synchronized to <b>SLOWCLK</b> in the subsystem.</p>
<b>nMBISTRESET</b>	Input	PD_AON	<p>A reset input provided for production MBIST. This reset input has the same effect as <b>nPORESET</b> and the same description applies to it.</p>
<b>nSRST</b>	Input	PD_AON	<p>An active-LOW system-wide Cold reset request, typically originating from an external debugger.</p> <p>When initially asserted, this signal results in a reset being applied to the subsystem and then subsequently removed. However, while <b>nSRST</b> is asserted, the CPUWAIT input of the processor is held HIGH, preventing the core from booting until <b>nSRST</b> is deasserted. This is independent of the register setting in the CPUWAIT register.</p> <p>It is recommended that the assertion of the <b>nSRST</b> input is at least three <b>SLOWCLK</b> cycles long. It can be held LOW for as long as it is required to hold off the processor from execution while debug related tasks are being performed.</p> <p>When deasserting <b>nSRST</b>, it must remain HIGH for at least five <b>SLOWCLK</b> cycles before asserting it again. This constraint is also applicable to the first assertion of <b>nSRST</b> that follows the deassertion of <b>nPORESET</b>.</p> <p>This asynchronous reset request input is synchronized to <b>SLOWCLK</b> before it is used for system-wide cold reset generation.</p>
<b>HOSTRESETREQ</b>	Input	PD_AON	<p>An active-HIGH system-wide Cold reset request, typically originating from an external higher authority. The higher authority can be, for example, the hosting system.</p> <p>When initially asserted, this signal must be held HIGH until the reset occurs on <b>nCOLDRESETAON</b>. Holding this input HIGH keeps <b>nCOLDRESETAON</b> asserted.</p> <p>This asynchronous reset request input is synchronized to <b>SLOWCLK</b> before it is used for system-wide cold reset generation.</p>

Signal	Direction	Power domain <sup>1</sup>	Description
<b>RESETREQ</b>	Input	PD_AON	<p>An active-HIGH system-wide Cold reset request.</p> <p>When initially asserted, this signal must be held HIGH until the reset occurs on <b>nCOLDRESETAON</b>, and must be cleared as a result of the reset being asserted.</p> <p>This asynchronous reset request input is synchronized to <b>SLOWCLK</b> before it is used for system-wide cold reset generation.</p>
<b>nCOLDRESETAON</b>	Output	PD_AON	<p>An active-LOW Cold reset for the expansion system. This cold reset merges other reset sources in the system with <b>nPORESET</b> to generate this reset.</p>
<b>nWARMRESETAON</b>	Output	PD_AON	<p>An active-LOW Warm reset for the expansion system.</p> <p>This Warm reset is a superset of <b>nCOLDRESETAON</b> and is also asserted when the system is in WARM_RST state.</p>
<b>nCOLDRESETMGMT</b>	Output	PD_AON	<p>An active-LOW Cold reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETAON</b> and is also asserted when the PD_MGMT power domain is in a low-power state.</p> <p>Since PD_MGMT is merged into PD_AON, the signal is the <b>COMGMTSYSCLK</b> synchronized version of <b>nCOLDRESETAON</b>.</p>
<b>nWARMRESETMGMT</b>	Output	PD_AON	<p>An active-LOW Warm reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETMGMT</b> and is also asserted when the system is in the WARM_RST state.</p> <p>Since PD_MGMT is merged into PD_AON, the signal is the <b>MGMTSYSCLK</b> synchronized version of <b>nWARMRESETAON</b>.</p>
<b>nCOLDRESETSYS</b>	Output	PD_SYS	<p>An active-LOW Cold reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETAON</b> and is also asserted when the PD_SYS power domain is in a low-power state with the logic being turned off. <b>nCOLDRESETSYS</b> supports Logic retention mode and is not asserted if the PD_SYS power domain is in a power mode with the logic being retained.</p>
<b>nWARMRESETSYS</b>	Output	PD_SYS	<p>An active-LOW Warm reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETSYS</b> and is also asserted when the system is in the WARM_RST state.</p>
<b>nCOLDRESETCPU0</b>	Output	PD_CPU0	<p>An active-LOW Cold reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETAON</b> and is also asserted when the PD_CPU0 power domain is in a low-power state with the logic being turned off. <b>nCOLDRESETCPU0</b> supports Logic retention mode and is not asserted if the PD_CPU0 power domain is in a power mode with the logic being retained.</p>
<b>nWARMRESETCPU0</b>	Output	PD_CPU0	<p>An active-LOW Warm reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETCPU0</b> and is also asserted when the system is in the WARM_RST state.</p>
<b>nCOLDRESETDEBUGCPU0</b>	Output	PD_DEBUG	<p>An active-LOW Cold reset for the expansion system.</p> <p>This reset is a superset of <b>nCOLDRESETAON</b> and is also asserted when the PD_DEBUG power domain is in a low-power state with the logic being turned off.</p>

<sup>1</sup> *Power Domain* refers to the power domain where this reset is used.

## 4.3 P-Channel and Q-Channel Device interfaces

Each P-Channel or Q-Channel Device interface in this section, independently, allows external expansion logic to handshake with the system to ensure that:

### For power control

All external managers and subordinate interfaces in a power domain are in quiescent state before entering a lower power state. It also allows external manager to request for a power domain to wake.

### For clock control

All external dependent logic can prepare itself before the hierarchical gating of clocks.

### For warm reset control

To prevent potential reset domain crossing issues and protocol violations on reset domain boundary and loss of data, and for physical protection (for example, flash memory).

During system integration, expansion logic that resides within a power or clock domain associated with each P-Channel or Q-Channel normally merges all P-Channel or Q-Channel interfaces within the expansion domain to drive each interface.

When merging, the following rules must be obeyed to prevent system deadlocks:

- All bus managers in the expansion system must, either individually or collectively have a full Q-channel interface, or a P-Channel interface.
  - Each Q-Channel interface must be able to deny a quiescence request if the logic that it controls has outstanding operations on the bus or is unable to enter quiescent state for any other reason. Similarly, each P-Channel interface must be able to deny a request to enter a different PSTATE if the logic is unable to enter the requested state. The exception is the WARM\_RST power state where managers are expected to suspend their activity cleanly, activate internal reset domain crossing protections, and always accept the request.
  - Each P-Channel interface for power control must be able to support all power modes that the Bounded Region implements.
- All bus subordinates in the expansion system must, either individually or collectively have a Q-Channel or P-Channel interface that must be able to deny the acceptance of a quiescence request or a Power state request if the subordinates idle conditions are not fulfilled. The exception is the WARM\_RST power state where subordinates are expected to suspend their activity cleanly, activate internal reset domain crossing protections if any and always accept the request.

Subordinates can delay the acceptance of a request if it has outstanding operations that can be independently completed of the power state of other system components. However, delaying the acceptance must not occur if the delay depends on the power state of other components in the system. Such situation may lead to deadlocks and must be avoided by denying the request.

- You must sequence the Q-Channels associated with these external bus interfaces in such a way to ensure that all bus managers are in quiescent state before any bus subordinates are requested to enter quiescent state.

Similarly, with P-Channels, you must ensure that all bus managers and subordinates of these interfaces can enter the new requested state in the right order depending on their dependencies before the P-Channel accepts entering that state.

- The SSE-310 implements separate Q-Channel and P-Channel interfaces for each clock domain and power domain respectively. To separately handshake managers, subordinates and even intermediate components in the expansion logic in sequence, additional external sequencing might be required in the expansion logic.
- You do not need to prevent data loss, protocol violation, or CPU lock-up of elements inside the reset domain if these are not visible to components outside the reset domain.
- The Warm reset boundary typically does not match a Power domain boundary, but includes logic from several power domains. Therefore, it is not required to close all LPI channels of a power domain before Warm reset assertion. The intention should be to close as few LPIs as possible to prevent deadlock/livelock/Warm reset denial scenarios.

If a Q-Channel Device interface is not used, then its associated **QACTIVE** and **QDENY** signals must be tied LOW and the **QREQn** output looped back into its **QACCEPTn** input. Similarly, if a P-Channel device interface is not used, then its associated **PACTIVE** and **PDENY** signals must be tied LOW, and the **PREQn** output looped back into its **PACCEPTn** input.

When P-Channel Device interface is used, the P-Channel encoding replicates the Device P-Channel bit assignment of the PCK-600 PPU. Each **DEVPACTIVE** bit is used to request entry to a Power mode and Operating mode, and **DEVSTATE** vector representing the Power mode and Operating mode that is being requested.

For more details, see Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual, and Arm® Power Policy Unit Architecture Specification.

For more information about the Q-Channel and P-Channel protocol, see AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces.

### 4.3.1 Clock Control Q-Channel Device interfaces

SSE-310 provides a Q-Channel Device interface for each of the output clocks to allow expansion logic to control the availability of each clock output. These are used to support high-level clock gating.

The following Q-Channels are provided:

- MGMTSYSCLK Q-Channel Device interface for **MGMTSYSCLK**. This interface resides in the PD\_AON domain.
- COMGMTSYSCLK Q-Channel Device interface for **COMGMTSYSCLK**. This interface resides in the PD\_AON power domain.

- SYSSYSCLK Q-Channel Device interface for **SYSSYSCLK**. This interface resides in the PD\_SYS power domain.
- COSYSSYSCLK Q-Channel Device interface for **COSYSSYSCLK**. This interface resides in the PD\_SYS power domain.
- CPU0CLK Q-Channel Device interface for **CPU0CLK**. This interface resides in the PD\_CPU0 power domain.
- COCPU0CLK Q-Channel Device interface for **COCPU0CLK**. This interface resides in the PD\_CPU0 power domain.
- DEBUGCPU0CLK Q-Channel Device interface for **DEBUGCPU0CLK**. This interface resides in the PD\_DEBUG power domain.

All clock Q-Channel Device interfaces are for clock control only, and do not support waking the system from HIBERNATION.

The clock Q-Channel Device interfaces COMGMTSYSCLK, COSYSSYSCLK, MGMTSYSCLK, SYSSYSCLK, CPU0CLK, COCPU0CLK, and DEBUGCPU0CLK must be handled as asynchronous.

### 4.3.2 Power Control P-Channel Expansion Device interfaces

SSE-310 provides power control P-Channel Device interfaces to allow expansion logic to handshake and coordinate the expansion logic power state.

Each power domain that supports the expansion logic is provided with P-Channel interfaces as follows:

- MGMT\_PWR P-Channel Device interface for PD\_AON. Given that PD\_AON is always on, this P-Channel is responsible for Warm reset related sequencing.
- SYS\_PWR P-Channel Device interface for BR\_SYS.
- DEBUG\_PWR P-Channel Device interface for BR\_DEBUG.
- CPU0\_PWR P-Channel Device interface for BR\_CPU0.

Each of these P-Channel Device interfaces is driven by expansion logic that resides within their respective power domain that each of the P-Channels control and is used to do the following:

- Used by the expansion logic to indicate through the **PACTIVE** signal that the expansion logic is IDLE or hint that it wants to enter a different power mode.
- For a power controller to request the expansion logic to enter a different power mode.
- Allow the expansion logic to accept or deny the request to enter a different power mode.

These interfaces do not support waking of any power domains because they reside within the power domain that is being controlled.

These Power Control P-Channel Device interfaces must be handled as synchronous to the clock used in each power domain that each of the P-Channel interfaces controls.

The following rules must be followed for the interfaces, not following these rules results in system deadlock:

- If the interface is not used, then:
  - **\*PWRPACTIVE** and **\*PWRPDENY** must be tied low
  - **\*PWRPPREQ** must be looped back to the corresponding **\*PWRPACCEPT**
- If the interface is used, then:
  - Denying a lower to higher power mode change request is not allowed
  - Denying WARM\_RST to ON power mode change request is not allowed

### 4.3.3 Power Control P-Channel Extension Device interfaces

SSE-310 provides power control P-Channel Device interfaces to allow extension of existing power domain hierarchy with new power domains by coordinating allowed power states of external power domain in the power domain hierarchy.

Each power domain that supports power domain hierarchy extension is provided with P-Channel interface, as follows:

- MGMTDPDHCPWR P-Channel Device Interface for PD\_MGMT. This interface resides in the PD\_AON power domain. The interface must only be used for handshaking and coordinating allowed power states of external power domains which are meant to be at the same level as PD\_SYS and PD\_DEBUG in the power domain hierarchy. The interface runs on **MGMTSYSCLK** and reset on **nCOLDRESETMGMT**.
- SYSPDHCPWR P-Channel Device interface for PD\_SYS. This interface resides in the PD\_AON power domain. The interface must only be used for handshaking and coordinating allowed power states of external power domains which are meant to be at the same level as PD\_CPU in the power domain hierarchy. The interface runs on **MGMTSYSCLK** and reset on **nCOLDRESETMGMT**.

### 4.3.4 Power Control Wakeup Q-Channel Device interfaces

SSE-310 provides Power Control Wakeup Q-Channel Device interfaces to allow expansion logic to request for specific logic power domains to wake up or power up.

The Q-Channel interfaces and the domains they control are:

- PWRMGMTWAKE Q-Channel Device interface for PD\_AON. Given that PD\_AON is always on, this interface should be tied LOW.
- PWRSYSWAKE Q-Channel Device interface for PD\_SYS.
- PWRDEBUGWAKE Q-Channel Device interface for PD\_DEBUG.
- PWRCPUOWAKE Q-Channel Device interface for PD\_CPU0.

These interfaces all reside in the PD\_AON power domain. Because they are wake up requests, only the **QACTIVE** signal of each interface is implemented.

## 4.4 Clock Control Q-Channel Control interfaces

Each Q-Channel Control interface in this section allows the subsystem independently to request the availability of a clock source. Each Q-Channel Control interface allows an external clock controller to handshake with the subsystem to safely turn the clock source OFF.

SSE-310 has the following Clock Control Q-Channel Control interfaces:

- AONCLK Q-Channel Control interface for **AONCLK**
- SYSCLK Q-Channel Control interface for **SYSCLK**
- CPU0CLK Q-Channel Control interface for **CPU0CLK**
- NPU0CLK Q-Channel Control interface for **NPU0CLK**

If an input clock source is always running, and there is no clock controller associated with this clock input, then you can tie its associated Clock Control Q-Channel Control interface by tying the **QREQn** input to HIGH.

These interfaces are in the PD\_AON power domain.

## 4.5 Expansion Power Control Dependency interface

SSE-310 provides two pairs of 4-bit width Q-Channel interfaces that allows external power domains to use the Power Dependency Control Matrix to keep power domains that are within the subsystem from entering a lower power state.

These signals are:

### **PDCMONQREQn[3:0]**

Power Dependency Control Matrix **ONQREQn** Inputs

When each bit is set to 1, it indicates that the external domain that drives it is active.

### **PDCMONQACCEPTn[3:0]**

Power Dependency Control Matrix **ONQACCEPTn** Outputs

Each bit acknowledges an associated bit on **PDCMONQREQn** by returning the request input value. This acts as a four-phase handshake so that the driver of each request bit can determine that the request has been observed by receiving unit.

### **PDCMRETQREQn[3:0]**

Power Dependency Control Matrix **RETQREQn** Inputs

When each bit is set to '1', it indicates that the external domain that drives it is active.

### **PDCMRETQACCPETn[3:0]**

Power Dependency Control Matrix **RETQACCEPTn** Outputs

Each bit acknowledges an associated bit on **PDCMRETQREQ<sub>n</sub>** by returning the request input value. This acts as a four-phase handshake so that the driver of each request bit can determine that the request has been seen by receiving unit.

These signals provides a way for an external power domain to request for a domain within the system to remain powered so that the external domain can access the internal power domain or at least request that the internal domain retain its register context. For example, the external domain may want to access the Main interconnect in the PD\_SYS domain:

1. External domain raises an interrupt with the host processor, which if not already ON will wake up PD\_SYS.
2. Host processor sets the relevant bit in **PDCM\_PD\_SYS\_SENSE.PDCMONQREQ<n>**.

The external system can now keep PD\_SYS powered using one of the **PDCMONQREQ<sub>n</sub>** signals.

Similarly if the external domain requires PD\_SYS to maintain state, but did not currently require access to it:

1. External domain raises an interrupt with the host processor.
2. Host processor sets the relevant bit in **PDCM\_PD\_SYS\_SENSE.PDCMRETQREQ<n>**.

The external system can now ensure PD\_SYS retains state using one of the **PDCMRETQREQ<sub>n</sub>** signals.

For more information about registers that use these signals and description of related functionality, see [System Control Register Block](#) and [Power Integration](#).

## 4.6 Power Domain ON Status Signals

SSE-310 provides a set of output signals that indicate whether the power domain they are associated with is in the ON Power mode.

These signals are:

### PDMGMTON

- HIGH indicates that the PPU of the PD\_MGMT power domain presumes that the domain is ON.
- LOW indicates that the PPU presumes that the power domain is in a lower power state.
- The signal is tied to '1', given that PD\_MGMT is merged into PD\_AON.

### PDSYSON

- HIGH indicates that the PPU of the PD\_SYS power domain presumes that the domain is ON.
- LOW indicates that the PPU of the power domain presumes that the domain is in a lower power state.



### PDCPU0

- HIGH indicates that the PPU of the PD\_CPU0 power domain presumes that the domain is ON.
- LOW indicates that the PPU of the power domain presumes that the domain is in a lower power state.

### PDNPU0ON

- HIGH indicates that the PPU of the PD\_NPU0 power domain presumes that the domain is ON.
- LOW indicates that the PPU of the power domain presumes that the domain is in a lower power state.

### PDDEBUGON

- HIGH indicates that the PPU of the PD\_DEBUG power domain presumes that the domain is ON.
- LOW indicates that the PPU of the power domain presumes that the domain is in a lower power state.



These are primarily status signals and are typically driven using **HWSTAT** signals of the PPU. Arm recommends that these are not directly used as power control signals.

---

## 4.7 System Timestamp Interface

When `EXPLOGIC_PRESENT = 0`, SSE-310 provides a system timestamp input from an expansion timestamp counter. This timestamp is expected to be driven by a timestamp generator in the subsystem expansion. This resides in the PD\_AON power domain.

The system timestamp interface has the following properties:

- Name: **CNTVALUEB[63:0]**
- Description: Timestamp input value. This value is binary coded.
- Width: 64-bit
- Direction: Input
- Clock domain: **CNTCLK**

When `EXPLOGIC_PRESENT = 1`, a system timestamp generator is integrated in the PD\_AON power domain of the subsystem expansion and drives the system timestamp interface.

There is no debug system to provide control signals for timestamp generation. Therefore, the **CPU0HALTED** signals are used to halt the system timestamp generator.

## 4.8 Main Interconnect Expansion interfaces

SSE-310 provides a pre-configured number of Manager and Subordinate Expansion interfaces of the Main Interconnect. These interfaces allow the system integrator to add additional bus managers and bus subordinates to the system.

The AXI5 AMBA protocol is used for this interface and supports the following properties:

- 32-bit address
- 64-bit data
- Synchronous to **SYSSYSCLK**
- On the **nWARMRESETSYS** reset
- TrustZone support enabled

The types of manager and subordinate Expansion interface on the Main Interconnect are as follows:

### Manager Code Main Expansion interface (XMSTEXP CODE)

This interface provides access to Code memory and is mapped to the following address range:

- 0x01000000 to 0x09FFFFFF
- 0x11000000 to 0x19FFFFFF

### Manager Main Expansion interfaces (XMSTEXP SRAM, XMSTEXP DEV)

These interfaces provide access to other subordinates in the system and are mapped to the following address range:

- 0x28000000 to 0x2FFFFFFF (on XMSTEXP SRAM)
- 0x38000000 to 0x3FFFFFFF (on XMSTEXP SRAM)
- 0x60000000 to 0x9FFFFFFF (on XMSTEXP SRAM)
- 0xA0000000 to 0xDFFFFFFF (on XMSTEXP DEV)

### Subordinate Main Expansion interfaces (XSLVEXPMIO and XSLVEXPMI1 or HSLVEXPMI1)

When EXPLOGIC\_PRESENT = 0, AXI subordinate interfaces are supported (XSLVEXPMIO, XSLVEXPMI1). When EXPLOGIC\_PRESENT = 1, an XHB-500 AHB to AXI bridge is instantiated in the SSE-310 expansion and AHB subordinate expansion interface (HSLVEXPMI1) is supported instead of XSLVEXPMI1.

Subordinate Main expansion interfaces allow the system integrator to connect additional bus managers to the system. These interfaces can be used to access the majority of memory-mapped regions in the system, except (but not limited) for regions that are private to the processor.

For more information, see [System Interconnect Infrastructure](#).

## 4.9 Peripheral Interconnect Expansion interfaces

SSE-310 provides a pre-configured number of Manager and Subordinate Expansion interfaces from the Peripheral Interconnect. These interfaces allow the system integrator to add additional bus managers and bus subordinates to the system that is expected to require lower latency access to peripherals.

The AMBA protocol used for this interface is AHB5 and supports the following properties:

- 32-bit address
- 32-bit data
- Synchronous to **SYSSYSCLK**
- On the **nWARMRESETSYS** reset
- TrustZone support enabled

The types of Manager and Subordinate Expansion interface on the Peripheral Interconnect are as follows:

### Manager Peripheral Expansion interfaces (HMSTEXPPILL, HMSTEXPPIHL)

These interfaces provide access to other subordinates in the system and are mapped to the following address range:

- 0x40100000 to 0x47FFFFFF (on HMSTEXPPILL)
- 0x48100000 to 0x4FFFFFFF (on HMSTEXPPIHL)
- 0x50100000 to 0x57FFFFFF (on HMSTEXPPILL)
- 0x58100000 to 0x5FFFFFFF (on HMSTEXPPIHL)
- 0xE0200000 to 0xEFFFFFFF (on HMSTEXPPILL)
- 0xF0200000 to 0xFFFFFFFF (on HMSTEXPPILL)

### Subordinate Peripheral Expansion interfaces (HSLVEXPPILL, HSLVEXPPIHL)

These interfaces allow system integrator to connect additional bus managers to the system. These interfaces can be used to access all peripheral memory-mapped regions in the system, except for regions that are private to CPU0:

- 0x40000000 to 0x47FFFFFF (except CPU private area, through HSLVEXPPILL)
- 0x48000000 to 0x4FFFFFFF (through HSLVEXPPIHL)
- 0x50000000 to 0x57FFFFFF (except CPU private area, through HSLVEXPPILL)
- 0x58000000 to 0x5FFFFFFF (through HSLVEXPPIHL)
- 0xE0100000 to 0xEFFFFFFF (through HSLVEXPPILL)
- 0xF0100000 to 0xFFFFFFFF (through HSLVEXPPILL)

## 4.10 Interrupt interfaces

SSE-310 includes interrupt signals for use by the subsystem expansion. These connect to the interrupt controller of CPU0 and to an External Wakeup Interrupt Controller (EWIC).

### Signal name

**CPU0EXPIRQ**

### Width

CPU0EXPNUMIRQ

CPU0EXPNUMIRQ defines the number of interrupts made available as expansion interrupts for CPU0. Note that each bit **CPU0EXPIRQ[n]** is ultimately connected to IRQ[32+n] of the CPU0 NVIC.

### Direction

Input

### Description

These are interrupt inputs from the subsystem expansion to the CPU0 interrupt controller and the EWIC within the subsystem.

CPU0 implements a configurable number of external interrupt lines and of these 32 are reserved for internal use and the rest are made available here.



When EXPLOGIC\_PRESENT=1, the system timestamp generator (system counter) is integrated in SSE-310 expansion and its security violation interrupt is mapped to **CPU0EXPIRQ[0]**. See Arm® Corstone™ Reference Systems Architecture Specification Ma1 for more information about the system counter.

### Signal name

**CPU0EXPNMI**

### Width

1

### Direction

Input

### Description

This provides a non-maskable interrupt input from the subsystem expansion to the interrupt controller of CPU0 and the EWIC within the subsystem.

This input is merged with other non-maskable interrupt sources within the subsystem before it is seen by the NVIC of the processor core.

## 4.11 CPU Coprocessor interface

The processor core of the subsystem is configured to have a co-processor interface.

This interface resides in the PD\_CPU0 power domain, **CPUCPU0CLK** clock domain and **nWARMRESETCPU0** reset domain.



Because of the actual processor implementation, the reset output **CPU0CPRESETOUTn** is part of the coprocessor interface that depends on **nWARMRESETCPU0**.

## 4.12 TCM Subordinate Interface

A 64-bit Subordinate TCM interface provides system access only to Tightly Coupled Memories (TCM) that are internal to CPU0. The protocol of this interface is AMBA AXI-5.

This interface is typically used together with DMA controllers to transfer data to and from the TCM interface of the processor.

This AXI interface supports the following properties:

- 32-bit address
- 64-bit data
- Normal Memory access only
- TrustZone Support enabled

The following table shows the TCM DMA subordinate interfaces memory map. Instruction TCM (ITCM) accesses are mapped into a single address space, while each 64-bit Data TCM (DTCM) access is mapped to four 32-bit wide DTCMs.

This interface resides in the PD\_SYS power domain, the interface is on **SYSSYSCLK** clock domain and **nWARMRESETSYS** reset domain.

**Table 4-4: TCM DMA Interface Memory Map**

Start Address	End Address	xADDRS[3:2]	TCM accessed
0x0A00_0000	0x0A00_0000 + {ITCM size}	-	Non-secure CPU0 ITCM
0x1A00_0000	0x1A00_0000 + {ITCM size}	-	Secure CPU0 ITCM
0x2400_0000	0x2400_0000 + {DTCM size}	2b00	Non-secure CPU0 D0TCM
0x2400_0000	0x2400_0000 + {DTCM size}	2b01	Non-secure CPU0 D1TCM
0x2400_0000	0x2400_0000 + {DTCM size}	2b10	Non-secure CPU0 D2TCM
0x2400_0000	0x2400_0000 + {DTCM size}	2b11	Non-secure CPU0 D3TCM
0x3400_0000	0x2400_0000 + {DTCM size}	2b00	Secure CPU0 D0TCM

Start Address	End Address	xADDRS[3:2]	TCM accessed
0x3400_0000	0x3400_0000 + {DTCM size}	2b01	Secure CPU0 D1TCM
0x3400_0000	0x3400_0000 + {DTCM size}	2b10	Secure CPU0 D2TCM
0x3400_0000	0x3400_0000 + {DTCM size}	2b11	Secure CPU0 D3TCM



Note

These addresses in the previous table are specific only for the TCM DMA subordinate interface. These TCMs reside at address offsets in the main memory map and are private. See [CPU TCM memories](#). To make these TCMs visible to other cores and managers within the system, these interfaces are mapped to expansion regions of the memory map. This is defined by the system integrator.

The Global Access monitor is not supported on the TCM Subordinate interface

## 4.13 Debug and Trace Related Interfaces

This section describes the debug and trace related interfaces of SSE-310.

### 4.13.1 Debug Access Interface

SSE-310 provides interfaces for debug access from an external debug access port or an external debug infrastructure.

The CPU Debug D-AHB access interface is provided as an expansion interface. This lets you drive the interface using a suitable CoreSight MEM-AP and provides debug access to the processor.

For more information on the D-AHB interface of the processor, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

The interface is in the PD\_CPU0 power domain and resides in the **COCPUCPU0CLK** clock domain and **nCOLDRESETCPU0** reset domain.

### 4.13.2 Serial Wire JTAG Interface

When EXPLOGIC\_PRESENT = 1, the DAP-Lite2 is integrated in the subsystem expansion and SSE-310 provides a Serial Wire JTAG (SWJ) interface instead of the debug access interface to enable an off-chip debugger to connect.

The SWJ interface supports both Serial Wire Debug (SWD) and JTAG data link protocols on a single set of shared pins, with dynamic switching between the two protocols.

The interface is in PD\_AON power domain and resides in the **SWCLKTCK** clock domain and **nPORESET** and **nTRST** reset domains.

### 4.13.3 Debug Timestamp Interface

The debug global timestamp input of the processor, **TSVALUEB[63:0]**, is provided as an expansion interface, **CPU0TSVALUEB[63:0]**. This should be driven by a global timestamp generator. For more information about these interfaces, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

This **CPU0TSVALUEB[63:0]** interface resides in the the following domains:

- PD\_DEBUG power domain
- **DEBUGCPU0CLK** clock domain
- **nCOLDRESETDEBUGCPU0** reset domain

The **TSCLKCHANGE** input of the processor is provided as an expansion interface as **CPU0TSCLKCHANGE** to allow CPU0 to be notified of a change in timestamp clock ratio. This interface resides in the respective **DEBUGCPU0CLK** clock domain and **nCOLDRESETDEBUGCPU0** reset domain.

When EXPLOGIC\_PRESENT = 1, a debug timestamp generator is integrated in the PD\_DEBUG power domain of the subsystem expansion and drives the **CPU0TSVALUEB[63:0]** input. The **CPU0TSCLKCHANGE** input is provided as an expansion interface.

### 4.13.4 Cross Trigger Channel Interface

SSE-310 includes a set of cross-trigger channel inputs and cross-trigger channel outputs to allow the integrator to expand the cross trigger infrastructure.

The cross-trigger channel interface of the processor is provided as expansion interface with two ports, **CPU0CTICHIN[3:0]** and **CPU0CTICHOUT[3:0]**. These two ports reside in PD\_DEBUG power domain, **nCOLDRESETCPU0** reset domain and are synchronous to **COCPU0CLK**.

For more details of the Cross Trigger Channel interface of the processor, see *Arm® Cortex®-M85 Processor Integration and Implementation Manual*.

### 4.13.5 CPU0 External Peripheral Interface EPPB

CPU0 provides interfaces that allow you to add peripherals to the external Private Peripheral Bus (PPB) region.

These are two 32-bit AMBA4 APB interfaces for integration with more CoreSight debug and trace components if required.

Data accesses are only allowed on each of these interfaces privately from the CPU0 at address 0xE0004000 to 0xE00FFFFF. Some of these regions are already reserved and others are available for integration of additional debug components. For more information, see [CPU Private Peripheral Bus region](#).

The APB interface that is associated to CPU0 Core PPB is called CPU0CPPB. It resides in the PD\_CPU0 power domain, the **CPUCPU0CLK** clock domain and the **nCOLDRESETCPU0** reset domain.

The interface that is associated to CPU0 Debug PPB is called CPU0DPPB. It resides in the PD\_DEBUG power domain, the **DEBUGCPU0CLK** clock domain and the **nCOLDRESETDEBUGCPU0** reset domain.

### 4.13.6 ATB Trace Interfaces

When EXPLOGIC\_PRESENT = 0, SSE-310 provides interfaces to output trace data to an expansion Trace Port Interface Unit (TPIU).

SSE-310 provides the following interfaces:

- The processor provides its ITM ATB trace interface and ETM ATB trace interface directly for expansion
- The trace synchronisation and trigger interface of the processor is also provided to expansion

All Interfaces reside in the PD\_DEBUG power domain, the **DEBUGCPU0CLK** clock domain, and the **nCOLDRESETDEBUGCPU0** reset domain.

### 4.13.7 Trace port interface

When EXPLOGIC\_PRESENT = 1, the TPIU-M is integrated in the subsystem expansion and SSE-310 provides trace port interface, instead of the ATB trace interfaces, to enable a Trace Port Analyzer to connect.

### 4.13.8 Debug Authentication Interface

The input signals define the debug authentication signal values when they are not overridden by the internal Secure debug configuration registers. Each of the output signals provide the merged authentication signals to the rest of the system.

The following table lists the debug authentication signals of the subsystem. These signals reside in the PD\_AON power domain.



Note

The debug authentication interface (**DBGGEN**, **NIDEN**, **SPIDEN** and **SPNIDEN**) of the processor can be changed at any time, however, the change is only guaranteed to take affect after a Context synchronization event (CSE).

For more information about the Context synchronization events, see *Arm®v8-M Architecture Reference Manual*.



**Table 4-5: Debug authentication interface**

Name	Width	Direction	Description
DBGENIN	1	Input	Debug Enable Input
NIDENIN	1	Input	Non-Invasive Debug Enable Input
SPIDENIN	1	Input	Secure Privilege Invasive Debug Enable Input
SPNIDENIN	1	Input	Secure Privilege Non-Invasive Debug Enable Input
DAPACCENIN	1	Input	External Debug Access Enable Input
DAPDSSACCENIN	1	Input	Debug Access Port to Debug System Access Enable Input
DBGEN	1	Output	Merged Debug Enable Output
NIDEN	1	Output	Merged Non-Invasive Debug Enable Output
SPIDEN	1	Output	Merged Secure Privilege Invasive Debug Enable Output
SPNIDEN	1	Output	Merged Secure Privilege Non-Invasive Debug Enable Output
DAPACCEN	1	Output	Merged External Debug Access Enable Output  <b>Note:</b> <b>DAPACCEN</b> is provided to allow the integrator to control a DAP interface directly to stop access even reaching the Debug Access Interface in the first place. This implementation allows disabling all debug access including the Armv8.1-M Unprivileged Debug Extension.
DAPDSSACCEN	1	Output	Merged Debug Access Port to Debug System Access Enable Output

### 4.13.9 Memory Protection Controller Expansion

SSE-310 supports up to 16 MPCs to be added to the expansion system.

The **SMPCEXPSTATUS** signal allows the interrupts of the MPCs to be internally merged to the single MPC Combined interrupt.

**Table 4-6: MPC Expansion Interrupt Status input**

Name	Width	Direction	Description
SMPCEXPSTATUS	16	Input	Interrupt Status inputs from all Expansion Memory Protection Controllers. These are associated to the SECMPCINTSTAT.SMPCEXP_STATUS register fields in the Secure Access Configuration Register Block and are used to raise an interrupt using the MPC Combined Interrupt.  Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being <b>RAZWI</b> .

## 4.13.10 Peripheral Interconnect Peripheral Protection Controller Expansion

SSE-310 supports up to four additional PPCs to be added to the Peripheral Interconnect in the expansion system. The following signals are provided to control the PPCs.

**Table 4-7: Peripheral Interconnect PPC Expansion Interface**

Name	Width	Direction	Description
<b>SPERIPHPPCEXPSTATUS</b>	4	Input	Peripheral Interconnect PPC Interrupt Status Input. Each bit ' <i>n</i> ' is to be connected to a single PPC < <i>n</i> > where <i>n</i> is 0-3.  These are associated to the SECPPCINTSTAT.SPERIPHPPCEXP_STATUS register fields.  Individual bit of this interface can be disabled if tied to 0, resulting in the associated register bit field being <b>RAZWI</b> .
<b>SPERIPHPPCEXPCLR</b>	4	Output	Peripheral Interconnect PPC Interrupt Clear Output. Each bit ' <i>n</i> ' is to be connected to a single PPC< <i>n</i> > where <i>n</i> is 0 to 3.  These are associated to the SECPPCINT CLR.SPERIPHPPCEXP_CLR register fields.  Individual bit of this interface can be disabled if the corresponding SPERIPHPPCEXPSTATUS bit is tied to 0, resulting in the the associated associated register bit field being <b>RAZWI</b> .
<b>PERIPHNSPPCEXP0</b>	16	Output	Peripheral Interconnect PPC Non-secure Gating Control. This is a set of four 16-bit interfaces.Each interface connects to a PPC. When each bit ' <i>m</i> ' of an interface is HIGH, it defines a specific < <i>m</i> > interface that the target PPC controls as Non-secure access only.  Each 16-bit signal <b>PERIPHNSPPCEXP&lt;<i>n</i>&gt;</b> is driven by the PERIPHNSPPCEXP< <i>n</i> > register, where <i>n</i> is 0 to 3.  Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being <b>RAZWI</b> .
<b>PERIPHNSPPCEXP1</b>	16	Output	See description of <b>PERIPHNSPPCEXP0</b> .
<b>PERIPHNSPPCEXP2</b>	16	Output	See description of <b>PERIPHNSPPCEXP0</b> .
<b>PERIPHNSPPCEXP3</b>	16	Output	See description of <b>PERIPHNSPPCEXP0</b> .
<b>PERIPHPPPCEXP0</b>	16	Output	Peripheral Interconnect PPC Privilege Gating Control. This is a set of four 16-bit interfaces. When each bit ' <i>m</i> ' of an interface is HIGH it defines the < <i>m</i> > interface that the target PPC controls as both Privileged and Unprivileged access. Having a control bit LOW allows Privilege access only.  Each bit PERIPHPPPCEXP< <i>n</i> >[ <i>m</i> ] is selected from either PERIPHSPPPCEXP< <i>n</i> >[ <i>m</i> ] if PERIPHNSPPCEXP< <i>n</i> >[ <i>m</i> ] is '0' or PERIPHNSPPCEXP< <i>n</i> >[ <i>m</i> ] otherwise, where <i>n</i> is 0 to 3  Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being <b>RAZWI</b> .
<b>PERIPHPPPCEXP1</b>	16	Output	See description of <b>PERIPHPPPCEXP0</b>
<b>PERIPHPPPCEXP2</b>	16	Output	See description of <b>PERIPHPPPCEXP0</b>
<b>PERIPHPPPCEXP3</b>	16	Output	See description of <b>PERIPHPPPCEXP0</b>

## 4.13.11 Main Interconnect Peripheral Protection Controller Expansion

SSE-310 can support up to four PPCs to be added to the Main Interconnect in the expansion system. The following signals are provided to control each PPC.

**Table 4-8: Main Interconnect PPC Expansion Interface**

Name	Width	Direction	Description
<b>SMAINPPCEXPSTATUS</b>	4	Input	<p>Main Interconnect PPC Interrupt Status Input. Each bit '<i>n</i>' is to be connected to a single PPC &lt;<i>n</i>&gt; where <i>n</i> is 0-3.</p> <p>These are associated to the SECPPCINTST AT.SMAINPPCEXP_STATUS register field.</p> <p>Individual bit of this interface can be disabled if tied to 0, resulting in the the associated register bit field being <b>RAZWI</b>.</p>
<b>SMAINPPCEXPCLR</b>	4	Output	<p>Main Interconnect PPC Interrupt Clear Output. Each bit '<i>n</i>' is to be connected to a single PPC&lt;<i>n</i>&gt; where <i>n</i> is 0-3.</p> <p>These are associated to the SECPPCINTCLR SMAINPPCEXP_CLR register field.</p> <p>Individual bit of this interface can be disabled if the corresponding SMAINPPCEXPSTATUS bit is tied to 0, resulting in the the associated register bit field being <b>RAZWI</b>.</p>
<b>MAINNSPPCEXP0</b>	16	Output	<p>Main Interconnect PPC Non-secure Gating Control. This is a set of four 16-bit interfaces. Each interface connects to a PPC. When each bit '<i>m</i>' of an interface is HIGH, it defines the &lt;<i>m</i>&gt; interface that the target PPC controls as Non-secure access only.</p> <p>Each 16-bit signal <b>MAINNSPPCEXP&lt;<i>n</i>&gt;</b> is driven by the MAINNSPPCEXP&lt;<i>n</i>&gt; register, where <i>n</i> is 0-3.</p> <p>Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being <b>RAZWI</b>.</p>
<b>MAINNSPPCEXP1</b>	16	Output	See description of <b>MAINNSPPCEXP0</b>
<b>MAINNSPPCEXP2</b>	16	Output	See description of <b>MAINNSPPCEXP0</b>
<b>MAINNSPPCEXP3</b>	16	Output	See description of <b>MAINNSPPCEXP0</b>
<b>MAINPPCEXP0</b>	16	Output	<p>Main Interconnect PPC Privilege Gating Control. This is a set of four 16bit interfaces. When each bit '<i>m</i>' of an interface is HIGH it defines the &lt;<i>m</i>&gt; interface that the target PPC controls as both Privileged and Unprivileged access. Having a control bit LOW allows Privilege access only.</p> <p>Each bit <b>MAINPPCEXP&lt;<i>n</i>&gt;[<i>m</i>]</b> is selected from either MAINPPCEXP&lt;<i>n</i>&gt;[<i>m</i>] if MAINNSPPCEXP&lt;<i>n</i>&gt;[<i>m</i>] is '0' or MAINNSPPCEXP&lt;<i>n</i>&gt;[<i>m</i>] otherwise, where <i>n</i> is 0-3.</p> <p>Individual bits of this interface can be unimplemented or disabled resulting in the associated register bit fields that contributes to this control signal being <b>RAZWI</b>.</p>
<b>MAINPPCEXP1</b>	16	Output	See description of <b>MAINPPCEXP0</b>
<b>MAINPPCEXP2</b>	16	Output	See description of <b>MAINPPCEXP0</b>
<b>MAINPPCEXP3</b>	16	Output	See description of <b>MAINPPCEXP0</b>

### 4.13.12 Manager Security Controller Expansion

SSE-310 can support up to 16 more *Manager Security Controllers* (MSC) to be added to the expansion system. The following signals are provided to control each MSC.

**Table 4-9: MSC Expansion Interface**

Name	Width	Direction	Description
<b>SMSCEXPSTATUS</b>	16	Input	<p>MSC Interrupt Status Input. Each bit 'n' is to be connected to a single MSC&lt;n&gt; where n is 0 to 15.</p> <p>These are associated with the SECMSCINTSTAT.SMSCEXP_STATUS register field.</p> <p>Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being <b>RAZ/WI</b>.</p>
<b>SMSCEXPCLR</b>	16	Output	<p>MSC Interrupt Clear Output. Each bit 'n' is to be connected to a single MSC&lt;n&gt; where n is 0 to 15.</p> <p>These are associated with the SECMSCINTCLR.SMSCEXP_CLR register field.</p> <p>Individual bits of this interface can be (tied) disabled, defined by the MSCEXPDIS parameter, resulting in the associated register bit field being <b>RAZWI</b>.</p>
<b>NSMSCEXP</b>	16	Output	<p>MSC Non-secure Configuration. Each bit 'n' is to be connected to a single MSC &lt;n&gt; where n is 0 to 15. Set HIGH to configure a manager as Non-secure.</p> <p>These are associated with the NSMSCEXP register.</p> <p>Individual bits of this interface can be (tied) disabled, defined by the MSCEXPDIS parameter. Any disabled bit of this interface are tied HIGH, resulting in the associated register bit field being read-as-one/write-ignored.</p>

### 4.13.13 Bridge Buffer Error Expansion

SSE-310 supports up to 16 more bridges with buffer error signalling to be added to the expansion system.

**Table 4-10: Bridge Error Interrupt Expansion Interface**

Name	Width	Direction	Description
<b>BRGEXPSTATUS</b>	16	Input	<p>Bridge Error Interrupt Status Input. Each bit 'n' is to be connected to a single bridge &lt;n&gt; where n is 0-15.</p> <p>These are associated with the BRGINTCLR.BRGEXP_STATUS register field.</p> <p>Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being <b>RAZWI</b>.</p>
<b>BRGEXPCLR</b>	16	Output	<p>Bridge Error Interrupt Clear Output. Each bit 'n' is to be connected to a single bridge &lt;n&gt; where n is 0-15.</p> <p>These are associated with the BRGINSTAT.BRGEXP_CLR register field.</p> <p>Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being <b>RAZWI</b>.</p>

When `EXPLOGIC_PRESENT = 1`, the XHB-500 AHB to AXI bridge is integrated in SSE-310 expansion and its interrupt output is mapped to **BRGEXPSTATUS[0]** and **BRGEXPCLR[0]**. The interrupt output goes HIGH for 1 clock cycle when the bridge receives an error response for an early terminated write. A glue logic converts the pulse interrupt to level interrupt and handshakes with the expansion interface.

For more information about the XHB-500 AHB to AXI bridge, see *Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual*.

#### 4.13.14 Other Security Expansion Signals

The following table list other signals that are related to Security that is needed by PPCs and MSCs in the expansion system.

**Table 4-11: Other Security Expansion Signals**

Name	Width	Direction	Description
<b>SECRESPCFG</b>	1	Output	<p>This signal configures how to respond to an access when a security violation occurs.</p> <ul style="list-style-type: none"> <li>0 - <b>RAZWI</b></li> <li>1 - Bus error</li> </ul> <p>The SECRESPCFG register controls this signal.</p>
<b>ACCWAITn</b>	1	Output	<p>This request signal is used to control any external gating unit that might be required to block accesses to the system through the Main and Peripheral Interconnect Expansion interfaces.</p> <ul style="list-style-type: none"> <li>1 - No gating</li> <li>0 - Access gated</li> </ul> <p>The BUSWAIT register controls this signal.</p>
<b>ACCWAITNSTATUS</b>	1	Input	<p>This status signal indicates the current state of any external gating unit that might be used to block access to the system through the Main and Peripheral Interconnect Expansion interfaces</p> <ul style="list-style-type: none"> <li>1 - No gating</li> <li>0 - Access gated</li> </ul> <p>Use the BUSWAIT register to read this signal.</p>

### 4.14 Clock configuration interfaces

SSE-310 provides control and status signals for the input clocks **AONCLK**, **CPU0CLK**, **NPU0CLK**, and **SYSCLK**. These signals support the configuration of generators or dividers that might exist in the expansion system. No divider is implemented in the subsystem.

The reset value of the control signals is configurable.

Each reset value must allow the related input clock to run at a default clock rate that allows the subsystem to boot without software configuring the related registers.

All signals in this interface reside in the PD\_AON power domain, are synchronous to **AONCLK** and are in the **nCOLDRESETAON** reset domain.

Because **SYSCLK**, **CPU0CLK**, and **NPU0CLK** must have the same frequency, the **CPU0CLKCFG** and **NPU0CLKCFG** outputs must not be used in the expansion logic, and the **CPU0CLKCFGSTATUS** and **NPU0CLKCFGSTATUS** input must be tied to LOW.

The register fields CLK\_CFG0.CPU0CLKCFG, CLK\_CFG2.NPU0CLKCFG, CLK\_CFG2.NPU0CLKCFGSTATUS, and CLK\_CFG0.CPU0CLKCFGSTATUS must not be used either.

**Table 4-12: Clock configuration interface signals**

Signal	Width	Direction	Description
<b>CPU0CLKCFG[3:0]</b>	4	Output	These control signals provide a set of 4-bit signals that allows the system to configure an external clock generation logic that drives <b>CPU0CLK</b> .  The CLK_CFG0.CPU0CLKCFG register field drives this signal.
<b>CPU0CLKCFGSTATUS[3:0]</b>	4	Input	4-bit status signals, used to read the status of any external clock generation logic that they drive.  Use the CLK_CFG0.CPU0CLKCFGSTATUS register field to read the values on this interface.
<b>SYSCLKCFG[3:0]</b>	4	Output	This control signal provides a 4-bit signal that allows the system to configure an external clock generation logic that drives <b>SYSCLK</b> clock.  The CLK_CFG1.SYSCLKCFG register field drives this signal.
<b>SYSCLKCFGSTATUS[3:0]</b>	4	Input	A 4-bit status signal, used to read the status of any external clock generation logic that drives <b>SYSCLK</b> clock.  Use the CLK_CFG1.SYSCLKCFGSTATUS register field to read the values on this interface.
<b>AONCLKCFG[3:0]</b>	4	Output	This control signal provides a 4-bit signal that allows the system to configure an external clock generation logic that drives <b>AONCLK</b> clock.  The CLK_CFG1.AONCLKCFG register field drives this signal.
<b>AONCLKCFGSTATUS[3:0]</b>		Input	A 4-bit status signal, used to read the status of any external clock generation logic that drives <b>AONCLK</b> clock.  Use the CLK_CFG1.AONCLKCFGSTATUS register field to read the values on this interface.
<b>NPU0CLKCFG[3:0]</b>	4	Output	These control signals provide a set of 4-bit signals that allows the system to configure an external clock generation logic that drives <b>NPU0CLK</b> . This signal is driven by the register fields CLK_CFG2.NPU0CLKCFG. Any unimplemented bits result in the associated register bit field being <b>RAZWI</b>
<b>NPU0CLKCFGSTATUS[4:0]</b>	5	Input	This sets of 4-bit status signals are used to read the status of any external clock generation logic that drives <b>NPU0CLK</b> . The values on this interface can be read by the register fields CLK_CFG2.NPU0CLKCFGSTATUS. Any unimplemented bits result in the associated register bit field being <b>RAZWI</b>

## 4.15 Miscellaneous Signals

The following table lists miscellaneous signals that are available for SSE-310.

**Table 4-13: Other Miscellaneous Top-Level Signals**

Signal name	Width	Direction	Clock that signal is synchronous to	Power domain	Description
LOCKNSVTOR0	1	Input	CPU0CLK	PD_CPU0	Disables writes to the VTOR_NS register. For more information on this register, see <i>Arm®v8-M Architecture Reference Manual</i> .  Asserting this signal prevents changes to the Non-secure vector table base address. This signal can be changed dynamically.
LOCKNSMPU0	1	Input	CPU0CLK	PD_CPU0	This signal disables writes to registers that are associated with the Non-secure MPU region from software or from a debug agent connected to CPU0. <ul style="list-style-type: none"> <li>MPU_CTRL_NS</li> <li>MPU_RNR_NS</li> <li>MPU_RBAR_NS</li> <li>MPU_RLAR_NS</li> <li>MPU_RBAR_A_NSn</li> <li>MPU_RLAR_A_NSn</li> </ul> For more information on these registers, see <i>Arm®v8-M Architecture Reference Manual</i> .
CPU0WAITCLR	1	Input	SYSCLK	PD_AON	When HIGH, this signal clears the register fields CPUWAIT.CPU0WAIT. This allows an external entity to release a processor that is already waited by <b>CPUWAIT</b> to start execution.  <b>Note:</b> While this is clocked using <b>SYSCLK</b> , this input can optionally be implemented as an asynchronous input.
CPU0WAITCLRRESP	1	Output	SYSCLK	PD_AON	This signal provides a response to the <b>CPU0WAITCLR</b> request. When <b>CPU0WAITCLR</b> is 1 and CPUWAIT.CPU0WAIT is 0, this signal is set to 1 until <b>CPU0WAITCLR</b> request goes 0.
CPU0LOCKUP	1	Output	AONCLK	PD_AON	Processor Lockup Status. This signal is an output directly from CPU0.
CPU0HALTED	1	Output	CPU0CLK	PD_CPU0	Processor Halted Status. There is one bit per Processor. Each bit indicates if the associated CPU0 has halted. This signal is an output directly from CPU0.
CPU0EDBGRQ	1	Input	CPU0CLK	PD_AON and PD_CPU0	External request for CPU0 to enter halt mode.  This signal is an input directly to CPU0 and to the EWIC.
CPU0DBGRESTART	1	Input	CPU0CLK	PD_CPU0	Request for CPU0 to perform synchronized exit from halt mode. This signal is an input directly to CPU0.
CPU0DBGRESTARTED	1	Output	CPU0CLK	PD_CPU0	Acknowledges <b>CPU0DBGRESTART</b>  This signal is an output directly from CPU0.

## 5 Functional Description

The following sections describe the functional description of the components of SSE-310.

### 5.1 Clocking infrastructure

SSE-310 provides several input clocks into the system.

#### **SLOWCLK**

An always running clock that is expected also to be the first available when the system first powers up.

**SLOWCLK** is also required to run while others can be turned off when the system is in its lowest power state other than OFF.

This clock drives the following logic that resides in the PD\_AON power domain:

- A 32bit Timer. This timer running at **SLOWCLK** allows the user to setup a wake event.
- A 32bit Watchdog Timer. This watchdog timer provides protection against an unresponsive system, particularly during the lowest power state.

#### **AONCLK**

This clock drives all other logic that resides in the PD\_AON domain, except logic that originates in the PD\_MGMT power domain that is merged into PD\_AON.

**AONCLK** drives all other logic that resides in the PD\_AON domain, except logic that originates in the PD\_MGMT power domain, which is merged into PD\_AON.

A Q-Channel Control interface, **AONCLK** Q-Channel Control Interface, allows the subsystem to request and handshake the availability of the AONCLK

This clock drives the following:

- External Wakeup Interrupt Controller (EWIC) that allows the processor to be woken via an interrupt.
- The PD\_MGMT PPU.
- All other logic in the PD\_AON domain that is not running on **SLOWCLK** and **SYSCLK**.

#### **SYSCLK**

This is the main system clock that drives most of the system that resides in the PD\_SYS power domain.

**SYSCLK** also drives all logic that resides in the PD\_AON domain and is not running on **SLOWCLK** and **AONCLK**. A Q-Channel Control interface, **SYSCLK** Q-Channel Control Interface, allows the subsystem to request and handshake the availability of the **SYSCLK**.

This clock is requested to run when the system is one of the following states:



- Is not at its lowest HIBERNATION 0 or 1 state.
- In SYS\_RET state, is waking from those states.
- Is forced to stay only via the CLOCK\_FORCE register.

This clock drives the following:

- The Main Interconnect and Peripheral Interconnect and other related functionality like expansion interfaces.
- System and Security Control related registers and logic, except those that resides in PD\_AON
- Power Control logic that resides in PD\_AON power domain that controls the PD\_SYS, PD\_CPU0, and PD\_DEBUG power domains.
- All Volatile Memory interfaces and peripherals in the PD\_SYS domain, along with the interfaces to Timers and Watchdog timers.

### CPU0CLK

This clock input drives CPU0 and its associated expansion interfaces and integration logic.

A Q-Channel Control interface, CPU0CLK Q-Channel Control Interface, allows the subsystem to request and handshake the availability of the **CPU0CLK**.

This clock is normally expected to be requested to turn on when CPU0 core domain, PD\_CPU0, or the CPU0 debug domain in the Debug System, PD\_DEBUG, is not off or is requested to turn on.

### CNTCLK

This clock is associated with the System Timestamp input, CNTVALUE<G/B>.

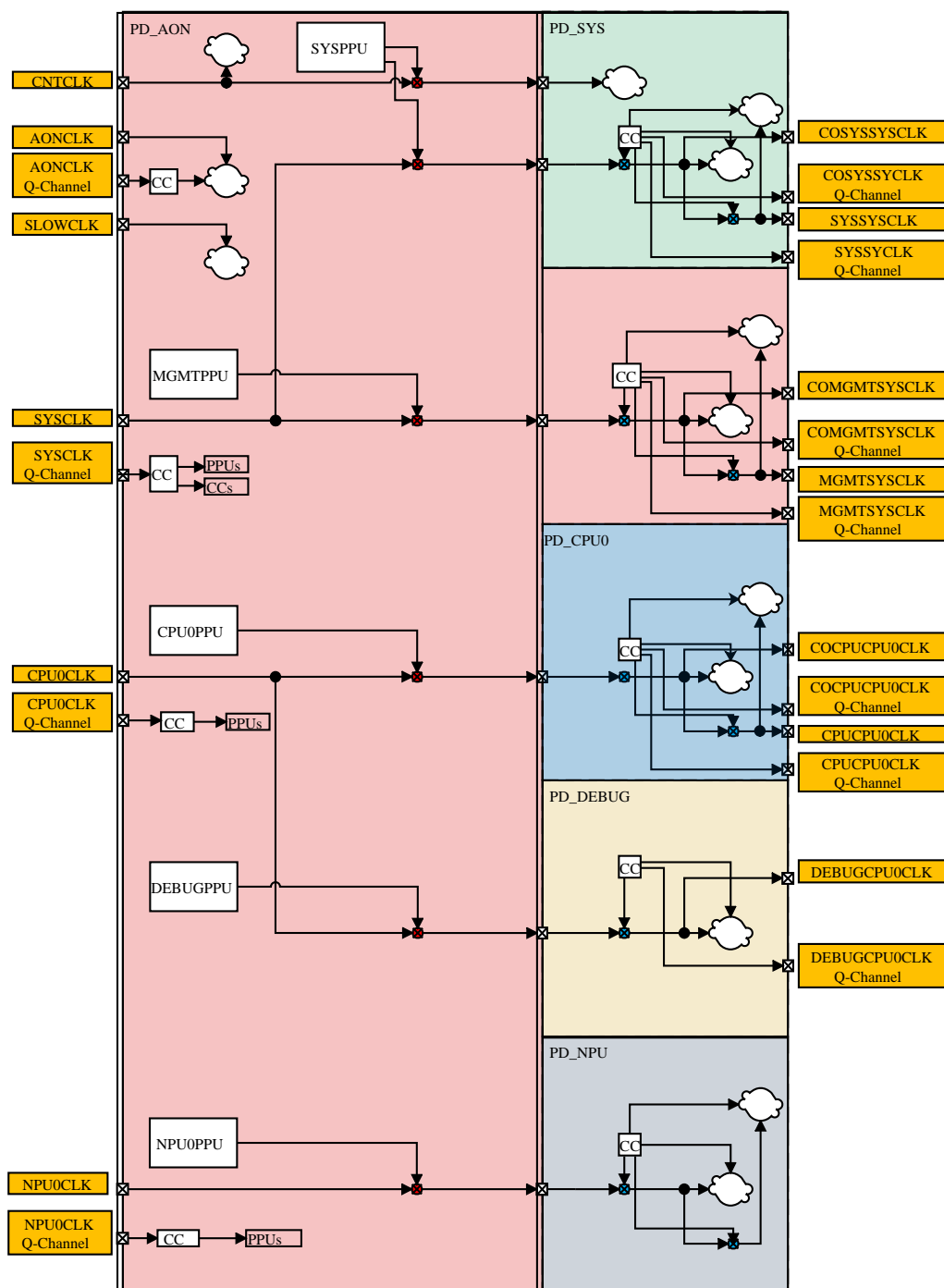
**CNTCLK** is used to drive timestamp related logic in all timestamp-based Timers and Watchdogs.

### NPU0CLK

This clock input drives NPU0 and its associated interfaces and integration logic.

A Q-Channel Control interface, NPU0CLK Q-Channel Control Interface, allows the subsystem to request and handshake the availability of the **NPU0CLK**. This clock is normally expected to be requested to turn on when NPU0 domain, PD\_NPU0, is not off or is requested to turn on.

**Figure 5-1: SSE-310 clock distribution structure**

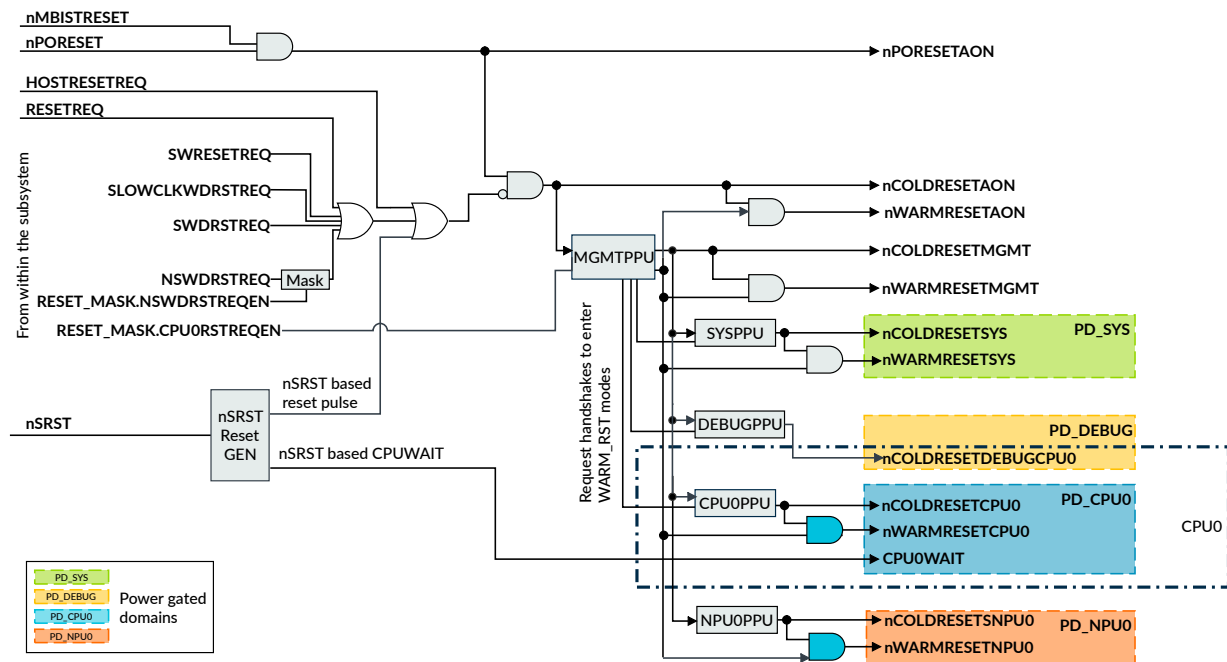


## 5.2 Reset infrastructure

Reset distribution defines how the output resets are related to the input resets and reset requests, and which power domains the reset outputs primarily drive.

Figure 5-2: Reset distribution in SSE-310 on page 51 shows the reset distribution in SSE-310. It also shows **nPORESETAON**, which is not an output of the subsystem and only resets the RESET\_SYNDROME register. This figure does not show the infrastructure for reset resynchronization and how the resets are used within each power domain.

**Figure 5-2: Reset distribution in SSE-310**



For the functional reset interfaces of SSE-310, see [Table 4-3: Top-level reset inputs and outputs](#) on page 25.

### 5.2.1 Power-on and Cold reset Handling

The input **nPORESET** is the power-on reset input, which after being combined with the production MBIST reset input, resets all registers in the design including the RESET\_SYNDROME register. The combined power-on reset is called **nPORESETAON**.

The **nPORESETAON** signal is further combined with other reset requests from the following list to generate the internal combined cold reset, which is available for use by expansion through the **nCOLDRESETAON** output:

- All reset requests of the watchdog timers, through a relevant mask if it exists for each.

- Reset request input **RESETREQ**.
- Reset request through the SWRESET.SWRESETREQ register value.
- Reset request input **HOSTRESETREQ**.
- Reset generated from the **nSRST** request by using negative-edge detection first and then stretching the result.

The **nCOLDRESETAON** signal resets almost all logic within the system except the RESET\_SYNDROME register.



In each power domain that is directly controlled by a PPU that resides in the PD\_AON power domain, the PPU is reset using **nCOLDRESETAON**. Each PPU then in turn generates the cold reset that is used in the power domain it controls. For example, the PPU for PD\_SYS is responsible for generating **nCOLDRESETSYS**.

## 5.2.2 CPU Reset Handling

The **nPORESET** reset input of CPU0 is driven by the CPU0PPU controlling the CPU core power domain, which is PD\_CPU0. CPU0PPU is reset using **nCOLDRESETAON**.

If the reset is the result of a cold reset request from the **nSRST** input, after a momentary cold reset, the **CPUWAIT** input of the CPU is forced HIGH as long as **nSRST** is held LOW. This action stops the processor from starting execution until **nSRST** is released. This allows a debugger to hold the CPU core and delay execution after reset, while it uses the Debug Access Port to perform debug operations.

The Warm reset mode is driven by a separate logic that resides in the PD\_AON domain. For more information, see [Warm reset generation and control](#). This logic, along with all PPUs in the system is used to force the system to an idle (quiescent) state (WARM\_RST System Power State is entered) before Warm reset mode assertion, which includes the **nSYSRESET** input of CPU0.

## 5.2.3 Warm reset generation and control

The system provides a Warm reset signal for each power domain. This reset is generated from the system reset request (**SYSRESETREQ**) of the processor and can be masked using the RESET\_MASK register.

This signal requests all PPUs in the system to enter the WARM\_RST power mode. After all PPUs have entered WARM\_RST (which is the WARM\_RST system power state), each Warm reset signal is asserted simultaneously to Warm reset the system.



The PPU's are not reset by Warm reset, they are only reset when **nCOLDRESETAON** is asserted.

## 5.2.4 Boot after reset

After resets, including power-on reset, the core boots using the values defined in the Initial Secure Reset Vector Register (INITSVTOR0) as the boot address. This value is stored in the System Control Register Block.

The default address is configurable, but Arm recommends that the default is set to 0x01000000, which is mapped to code memory through the Manager Code Main Expansion Interface. This address can be modified by software before subsequent warm reboots of the processor.

The TrustZone for Armv8-M states that boot must start from a Secure memory space. At boot, all volatile memory is Secure only. Software must change or restore the settings in the MPC to release memory for Non-secure world use.

The **CPU0WAIT** input to the core can force the processor to wait before executing the instruction. The processor in the system has an CPU0WAIT.CPU0WAIT register field that controls whether the processor starts running its boot code when it wakes. There is also a **CPU0WAITCLR** input that allows an external entity to clear the associated CPU0WAIT register bit.

## 5.2.5 NPU reset handling

The **nRESET** reset input of NPU0 is driven by the PPU that controls the NPU power domain PD\_NPU0.

This PPU is reset using **nCOLDRESETMGMT** reset. A Warm reset is driven from Warm reset generation logic that resides in the PD\_AON domain.

## 5.3 Cortex-M85 processor

SSE-310 supports one Cortex-M85 processor core (referred to as CPU0). The processor can support different static configurations.

The following table summarizes the configuration of CPU0 used in SSE-310 subsystem, and how configurations of the CPU0 are mapped to the configurable rendering options of the SSE-310 subsystem.

For more details, see [Configurable render options](#).

**Table 5-1: Supported CPU configurations**

Configuration	Supported value for CPU0	Related SSE-310 render option	Description
SECEXT	1	SECEXT	Specifies whether the Armv8-M Security Extension is included: <ul style="list-style-type: none"> <li>0 - Security Extension not included</li> <li>1 - Security Extension included</li> </ul>
MPU_NS	> 0	CPU0_MPU_NS	Specifies the number of Non-secure MPU regions included.
MPU_S	> 0	CPU0_MPU_S	Specifies the number of Secure MPU regions included.
SAU	> 0	CPU0_NUM_SAU_CONFIG	Specifies the number of SAU Regions.
NUMIRQ	CPU0EXPNUMIRQ + 32	CPU0EXPNUMIRQ	Specifies the number of external interrupts included for CPU0.
IWIC	0	HASCPU0IWIC	Specifies whether the Internal Wake up Interrupt Controller (IWIC) is included for CPU0: <ul style="list-style-type: none"> <li>0 - IWIC not included</li> <li>1 - IWIC included</li> </ul>
WICLINES	CPU0EXPNUMIRQ + 35	CPU0EXPNUMIRQ	Specifies the number of IRQ lines supported by the IWIC and EWIC. The value always includes the three internal events NMI, RXEV, and Debug request event, and at least one IRQ.
CTI	1	CPU0_CTI_PRESENT	Specifies whether the Cross Trigger Interface (CTI) unit is included: <ul style="list-style-type: none"> <li>0 - CTI not included</li> <li>1 - CTI included</li> </ul>
BUSPROT	0	BUSPROT_PRESENT	Specifies whether interface protection is supported on the M-AXI, P-AHB, EPPB, and S-AXI interfaces of the processor: <ul style="list-style-type: none"> <li>0 - Interface protection not included</li> <li>1 - Interface protection included</li> </ul>
ITGU	1	ITGU_PRESENT	Specifies whether the processor is configured with ITCM Security gate unit: <ul style="list-style-type: none"> <li>0 - TCM Gate Unit not included</li> <li>1 - TCM Gate Unit included</li> </ul>
DTGU	1	DTGU_PRESENT	Specifies whether the processor is configured with DTCM Security gate unit: <ul style="list-style-type: none"> <li>0 - TCM Gate Unit not included</li> <li>1 - TCM Gate Unit included</li> </ul>
ECC	0	ECC_PRESENT	Specifies whether the processor supports Error detection and correction in the L1 Data and Instruction cache (when configured) and the TCM.  0 - ECC not included
LOCKSTEP	0	CPU0_LOCKSTEP	Specifies whether the processor is configured for dual-redundant lock-step operation. The options are: <ul style="list-style-type: none"> <li>0 - Regular processor operation</li> <li>1 - Dual-redundant lockstep operation</li> </ul>
PMC	0	CPU0_PMC_PRESENT	Specifies whether the MBIST Controller (PMC-100) is included. <ul style="list-style-type: none"> <li>0 - PMC-100 not included</li> <li>1 - PMC-100 included</li> </ul>

Configuration	Supported value for CPU0	Related SSE-310 render option	Description
RAR	1	CPU0_RAR	Specifies if the synchronous states or the architectural required state is reset. The options are: <ul style="list-style-type: none"> <li>0 - Only reset the architectural required states</li> <li>1 - Reset all synchronous states</li> </ul>
DBGLVL	>0	CPU0_DBGLVL	Specifies the number of debug resources included. The options are: <ul style="list-style-type: none"> <li>0 - Minimal debug. No Halting debug or memory access.</li> <li>1 - Reduced set. Two Data Watchpoint and Trace (DWT) and four Breakpoint Unit (BPU) comparators.</li> <li>2: Full set. Four DWT and eight BPU comparators. Debug Monitoring mode and the Unprivileged Debug Extension (UDE) is always supported. The Performance Monitoring Unit (PMU) is included when CPU0_DBGLVL is nonzero.</li> </ul>
ICACHESZ[4:0]	ICACHESZ[0]=0b1	CPU0_INSTR_CACHE_SIZE	Specifies the inclusion of the instruction cache controller in the processor. If the instruction cache controller is included, CPU0_INSTR_CACHE_SIZE specifies the size of the cache. <ul style="list-style-type: none"> <li>CPU0_INSTR_CACHE_SIZE[0]=0 - Instruction cache is excluded</li> <li>CPU0_INSTR_CACHE_SIZE[0]=1 - Instruction cache is included</li> <li>If CPU0_INSTR_CACHE_SIZE[0]=1, then the cache sizes are: <ul style="list-style-type: none"> <li>CPU0_INSTR_CACHE_SIZE[4:1]=0b0000 4 KB instruction cache</li> <li>CPU0_INSTR_CACHE_SIZE[4:1]=0b0001 8 KB instruction cache</li> <li>CPU0_INSTR_CACHE_SIZE[4:1]=0b0011 16 KB instruction cache</li> <li>CPU0_INSTR_CACHE_SIZE[4:1]=0b0111 32 KB instruction cache</li> <li>CPU0_INSTR_CACHE_SIZE[4:1]=0b1111 64 KB instruction cache</li> </ul> </li> </ul>
DCACHESZ[4:0]	DCACHESZ[0]=0b1	CPU0_DATA_CACHE_SIZE	Specifies the Manager AXI (M-AXI) configuration and the inclusion and size of data cache controller. <ul style="list-style-type: none"> <li>CPU0_DATA_CACHE_SIZE[0]=0 - Area-optimized M-AXI data cache is excluded</li> <li>CPU0_DATA_CACHE_SIZE[0]=1 - Performance-optimized M-AXI data cache is included. The cache sizes are: <ul style="list-style-type: none"> <li>CPU0_DATA_CACHE_SIZE[4:1]=0b0000 4 KB instruction cache</li> <li>CPU0_DATA_CACHE_SIZE[4:1]=0b0001 8 KB instruction cache</li> <li>CPU0_DATA_CACHE_SIZE[4:1]=0b0011 16 KB instruction cache</li> <li>CPU0_DATA_CACHE_SIZE[4:1]=0b0111 32 KB instruction cache</li> <li>CPU0_DATA_CACHE_SIZE[4:1]=0b1111 64 KB instruction cache</li> </ul> </li> </ul>

Configuration	Supported value for CPU0	Related SSE-310 render option	Description
MVE	If FPU = 0 then MVE = 1	CPU0_MVE_CONFIG	M-profile Vector Extension (CPU0_MVE_CONFIG) parameter can have the following configuration options: <ul style="list-style-type: none"> <li>0 - MVE not included</li> <li>1 - Integer subset of MVE included</li> <li>2 - Integer and half single-precision floating-point MVE included. This option is only valid if CPU0_FPU_PRESENT=1.</li> </ul>
FPU	0,1	CPU0_FPU_PRESENT	The CPU0_FPU_PRESENT parameter determines the floating-point functionality of the processor. <ul style="list-style-type: none"> <li>0 - FPU not included</li> <li>1 - FPU included</li> </ul>
CPIF	0,1	HASCPUOCPIF	Specifies the inclusion of the external coprocessor interface. The options are: <ul style="list-style-type: none"> <li>0 - Coprocessor interface not included</li> <li>1 - Coprocessor interface included</li> </ul>
IDCACHEID	0	CPU0_IDCACHEID	Unique identifier for instruction and data cache RAM implementation.
IRQLVL	3-8	CPU0_IRQLVL	Specifies the number of exceptions priority bits.
ITM	0,1	CPU0_ITM_PRESENT	Specifies the level of instrumentation trace supported. The options are: <ul style="list-style-type: none"> <li>0 - Instrumentation Trace Macrocell and DWT trace excluded</li> <li>1 - Instrumentation Trace Macrocell and DWT trace included</li> </ul>
ETM	0,1	CPU0_ETM_PRESENT	Specifies support for Embedded Trace Macrocell (ETM) trace The options are: <ul style="list-style-type: none"> <li>0 - ETM trace is excluded</li> <li>1 - ETM trace is included</li> </ul>
ITGUBLKSZ	0-15	CPU0_ITGUBLKSZ	ITCM gate unit block size
ITGUMAXBLKS	(CPU0_CFGITCMSZ +4)- CPU0_ITGUBLKSZ	-	Maximum number of ITCM gate unit blocks
DTGUBLKSZ	0-15	CPU0_DTGUBLKSZ	DTCM gate unit block size
DTGUMAXBLKS	(CPU0_CFGDTCMSZ +4)- CPU0_DTGUBLKSZ	-	Maximum number of DTCM gate unit blocks
PACBTI	0,1	CPU0_PACBTI	Specifies inclusion of the Pointer Authentication and Branch Target Identification (PACBTI) extension

The following table shows:

- The render parameter of SSE-310 that can be used to modify the default value of the input signals.
- How miscellaneous interface input signals of the CPU are tied.



**Table 5-2: Render parameters of the input signals**

Interface signals	Value for CPU0	Related SSE-310 parameter	Description
CFGMEMALIAS[4:0]	0b10000	CFGMEMALIAS	Memory address alias bit for the ITCM, DTCM, and P-AHB regions. The address bit used for the memory alias are: <ul style="list-style-type: none"> <li>0b00001 - Alias bit = 24</li> <li>0b00010 - Alias bit = 25</li> <li>0b00100 - Alias bit = 26</li> <li>0b01000 - Alias bit = 27</li> <li>0b10000 - Alias bit = 28</li> <li>0b00000 - No alias. TCM Security gating disabled.</li> </ul>
CFGBIGEND	0	CFGBIGEND	Data endian format:  0 - Little-endian (LE)  1 - byte-invariant big-endian (BE8)
CFGITCMSZ[3:0]	Must be $\geq$ 0b0011	CPU0_CFGITCMSZ	Size of the instruction TCM region encoded as: <ul style="list-style-type: none"> <li>0b0000 - No ITCM implemented, which is not supported by this specification</li> <li>0b0011 - <math>2^{\text{CFGITCMSZ}-1}\text{KB}</math></li> <li>Others - Reserved</li> </ul>
CFGDTCMSZ[3:0]	Must be $\geq$ 0b0000	CPU0_CFGDTCMSZ	Size of the data TCM region encoded as: <ul style="list-style-type: none"> <li>0b0000 - No DTCM implemented, which is not supported by this specification</li> <li>0b0011 - <math>2^{\text{CFGDTCMSZ}-1}\text{KB}</math></li> <li>Others - Reserved</li> </ul>
CFGPAHBSZ[2:0]	0b010	CPU0_CFGPAHBSZ	Size of the P-AHB peripheral port memory region: <ul style="list-style-type: none"> <li>0b000 - P-AHB disabled.</li> <li>0b001 - 64MB</li> <li>0b010 - 128MB</li> <li>0b011 - 256MB</li> <li>0b100 - 512MB</li> </ul> <p><b>Note:</b> Setting CFGPAHBSZ to any other value results in <b>UNPREDICTABLE</b> behavior.</p>
SAUDISABLE	0	CPU0_SAUDISABLE	If the Security Attribution Unit (SAU) is configured, disables support.
INITTCMEN[1:0]	0b11	INITTCMEN	TCM enables initialization out of reset. Set to all ones to enable both TCMs.
WICCONTROL[0]	1	N/A	Setting to 1 causes the CPU to use WIC when in Wait for Interrupt (WFI) DEEPSLEEP.
INITPAHBEN	1	INITPAHBEN	Setting to 1 to always enable P-AHB interface. This signal controls the reset value of PAHBEN.
LOCKPAHB	1	CPU0_LOCKPAHB	Disable writes to the PAHBEN register from software or from a debug agent connected to the processor. Asserting this signal prevents changes to AHB peripheral port enable status in PAHBEN.
LOCKDAIC	1	LOCKDAIC	Disables access to the instruction cache direct cache access registers DCAICLR and DCAICRR in the processor.

### 5.3.1 EVENT Interfaces

CPU0 has **RXEV** and **TXEV** event interfaces.

In SSE-310:

- **TXEV** drives the **RXEV**.
- Events do not wake the processor from its EWIC based low-power state
- Events do not wake the system from Hibernation0 state.

SSE-310 does not support the use of WFE for the CPU to enter the DeepSleep state.

### 5.3.2 Interrupts

SSE-310 provides the following components that can generate interrupts within the system:

- PPU interrupts
- Security-based interrupts
- Timers and watchdogs
- Cross-Trigger interrupts
- NPU

Depending on the configuration option values of CPU0EXPNUMIRQ and CPU0EXPIRQDIS, interrupts of the CPU0 are made available to be driven through expansion logic.

The following table lists the interrupt map of the CPU. The software must ensure that all PPU interrupts, and interrupts marked as Secure, are handled as Secure interrupts. If an interrupt source does not exist because of the chosen configuration of the system, the unused interrupt pin is disabled and reserved.

The table also indicates those interrupts that also act as wakeup interrupts at the CPU associated EWIC. The EWIC acts primarily as an entity that takes over the masking and holding of an interrupt, on behalf of the CPU NVIC, when the CPU0 is in its OFF or low-power state and is therefore unable on its own to handle interrupts.

Using the EWIC:

1. The system can enter a lower power state that switches off the CPU0 along with most of the system, except for the EWIC itself.
2. Then the system can run the EWIC on a much lower clock frequency to lower power consumption to attain a very low standby operating power.

**Table 5-3: CPU0 interrupt map**

Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
NMI	Combined Secure watchdog, SLOWCLK watchdog and <b>CPU0EXPNMI</b>	Yes

Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
IRQ[0]	Non-secure watchdog reset request	Yes
IRQ[1]	Non-secure watchdog interrupt	Yes
IRQ[2]	SLOWCLK Timer	Yes
IRQ[3]	Timer 0	Yes
IRQ[4]	Timer 1	Yes
IRQ[5]	Timer 2	Yes
IRQ[6]	Reserved	-
IRQ[7]	Reserved	-
IRQ[8]	Reserved	-
IRQ[9]	MPC Combined (Secure)	Yes
IRQ[10]	PPC Combined (Secure)	Yes
IRQ[11]	MSC Combined (Secure)	Yes
IRQ[12]	Bridge Error Combined interrupt (Secure)	Yes
IRQ[13]	Reserved	-
IRQ[14]	PPU combined (Secure)	Yes
IRQ[15]	Reserved	-
IRQ[16]	NPU0	Yes
IRQ[17]	Reserved	-
IRQ[18]	Reserved	-
IRQ[19]	Reserved	-
IRQ[23:20]	Reserved	-
IRQ[24]	Reserved	-
IRQ[25]	Reserved	-
IRQ[26]	Reserved	-
IRQ[27]	Timer 3 AON	Yes
IRQ[28]	CPU0CTIIRQ0 (local CPU CTI only)	No
IRQ[29]	CPU0CTIIRQ1 (local CPU CTI only)	No
IRQ[31:30]	Reserved	-
IRQ[<CPU0EXPNUMIRQ+31>:32]	CPU0EXPIRQ[CPU0EXPNUMIRQ-1:0]. See <a href="#">Interrupt Interfaces</a> .	Yes

## 5.4 System Interconnect infrastructure

The system interconnect infrastructure provides a bus infrastructure that transfers memory mapped access from bus managers to subordinates in the system. SSE-310 defines the key interconnects that form the System Interconnect:

### Main Interconnect

This interconnect provides the highest amount of bus throughput. It is primarily, but not exclusively, for code and data accesses that targets memories or high throughput interfaces. For example:

- In-subsystem volatile memories, that is VM0 and VM1
- Flash, DRAM controllers, or ROM that reside outside the system
- Other high throughput devices

### Peripheral Interconnect

This interconnect provides access to lower performance peripherals. For example:

- System and Watchdog timers
- System and other security Configuration Registers
- Power control logic

Cortex-M85 has a direct interface to access this interconnect.

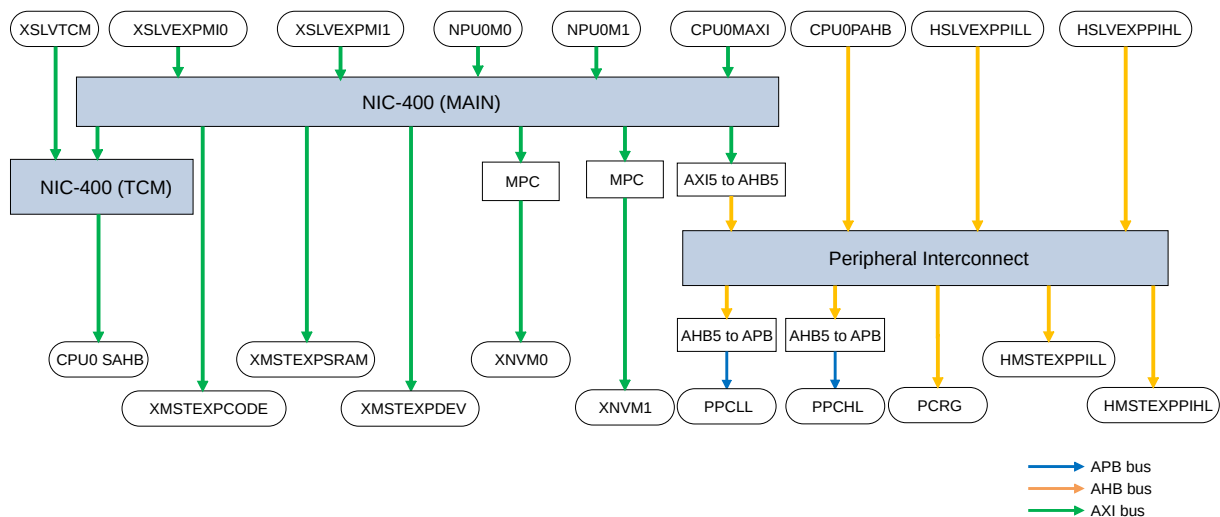
### TCM Interconnect

This interconnect provides access from TCM subordinate interface and from the Main interconnect to the Tightly Coupled Memories (TCM) that are internal to CPU0.

The System Interconnect resides in PD\_SYS. The interconnect runs primarily on **SYSSYSCLK**. The power domain also has its local derived Cold reset input and warm resets. For more information, see [Clocking Infrastructure](#) and [Reset Infrastructure](#).

The following figure shows the connections between the interconnect components.

**Figure 5-3: Interconnect connections**



The NIC Interconnect is not a full crossbar, and certain managers can only access specific subordinates as shown in following table:

**Table 5-4: NIC interface connections**

Manager interface	Subordinate Interface								
	MAXIO	XSLVEXPMIO	XSLVEXPMI1	NPU0M0	NPU0M1	XSLVTCM	CPU0PAHB	HSLVEXPPILL	HSLVEXPPIHL
XMSTEXPCODE	Y	Y	Y	Y	Y	N	N	N	N
XMSTEXPSRAM	Y	Y	Y	Y	Y	N	N	N	N
XMSTEXPDEV	Y	Y	Y	Y	Y	N	N	N	N
XVM0M	Y	Y	Y	Y	Y	N	N	N	N
XVM1M	Y	Y	Y	Y	Y	N	N	N	N
CPU0SAHB	N	Y	Y	N	N	Y	N	N	N
HMSTEXPPILL	N	N	N	Y	Y	N	Y	N	N
HMSTEXPPIHL	Y	Y	Y	N	N	N	N	Y	Y
PPCLL	N	N	N	N	N	N	Y	N	N
PPCHL	Y	Y	Y	Y	Y	N	N	Y	Y
PCRG_AHB	Y	Y	Y	Y	Y	N	N	N	Y

**Table 5-5: NIC interface address ranges**

Manager interface	Address range
XMSTEXPCODE	0x01000000 - 0x09FFFFFF
	0x11000000 - 0x19FFFFFF
XMSTEXPSRAM	0x28000000 - 0x2FFFFFFF
	0x38000000 - 0x3FFFFFFF
	0x60000000 - 0x9FFFFFFF
XMSTEXPDEV	0xA0000000 - 0xDFFFFFFF
XVM0M	0x21000000 - 0x21FFFFFF
	0x31000000 - 0x31FFFFFF
XVM1M	0x21000000 - 0x21FFFFFF
	0x31000000 - 0x31FFFFFF
CPU0SAHB	0x0A000000 - 0x0AFFFFFF
	0x1A000000 - 0x1AFFFFFF
	0x24000000 - 0x24FFFFFF
	0x34000000 - 0x34FFFFFF
HMSTEXPPILL	0x40100000 - 0x47FFFFFF
	0x50100000 - 0x57FFFFFF
	0xE0200000 - 0xEFFFFFFF
	0xF0200000 - 0xFFFFFFFF

Manager interface	Address range
HMSTEXPPIHL	0x48100000 - 0x4FFFFFFF 0x58100000 - 0x5FFFFFFF
PPCLL	0x40004000 - 0x40004FFF 0x40080000 - 0x40080FFF 0x50004000 - 0x50004FFF 0x50080000 - 0x50080FFF 0x50083000 - 0x50083FFF 0x50084000 - 0x50084FFF
PPCHL	0x48000000 - 0x48000FFF 0x48001000 - 0x48001FFF 0x48002000 - 0x48002FFF 0x48003000 - 0x48003FFF 0x48040000 - 0x48040FFF 0x48041000 - 0x48041FFF 0x58000000 - 0x58000FFF 0x58001000 - 0x58001FFF 0x58002000 - 0x58002FFF 0x58003000 - 0x58003FFF 0x58040000 - 0x58040FFF 0x58041000 - 0x58041FFF

Manager interface	Address range
PCRG_AHB	0x48020000 - 0x48020FFF
	0x4802F000 - 0x4802FFFF
	0x58020000 - 0x58020FFF
	0x58021000 - 0x58021FFF
	0x58022000 - 0x58022FFF
	0x58023000 - 0x58023FFF
	0x58028000 - 0x58028FFF
	0x58029000 - 0x58029FFF
	0x5802A000 - 0x5802AFFF
	0x5802E000 - 0x5802EFFF
	0x5802F000 - 0x5802FFFF

### 5.4.1 ACC\_WAIT Control

SSE-310 provides a set of controls to let you add access control gates or block access to the system.

- Add access control gates, driven using the **ACCWAITn** signal
- Block access to the system, when the system:
  - Leaves Hibernation state
  - Resets
  - Performs first power up
  - When software wants to reconfigure security settings in the system

This prevents access to the system until all security-related features of the system have been set up correctly.

After software is ready, it can release the gates by writing to the **BUSWAIT.ACC\_WAITN** register. Then software can check the status of all external gating units by reading the **ACCWAITNSTATUS** signal value through the **BUSWAIT.ACC\_WAITN\_STATUS** register.

## 5.5 Volatile Memory

SSE-310 supports two memory banks, which are VM0 and VM1.

Each volatile memory (VM) can support a configurable amount of SRAM memory, but must obey the following requirements:

- The size of each must be in powers of two.
- The size of both banks is defined using the VMADDRWIDTH configuration option and is equal to  $2^{\text{VMADDRWIDTH}}$  bytes.
- The combined total memory size of all volatile memory banks that exist within the system is less than 16 Mbytes.
- All volatile memories form a contiguous area of memory. They are mapped to a starting address of 0x21000000, which is also aliased to a starting address of 0x31000000.

All volatile memories support Exclusive Accesses from CPU0 and external managers.

Each VM has a *Memory Protection Controller* (MPC) associated with it that lets you map segments of each memory to Secure or Non-secure world.

For more information about the MPC, refer to *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* and *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*.

In SSE-310, all VMs use the same MPC block size as defined by the configuration VMMPCLBLKSIZE. Each MPC is implemented so that out of reset, all memory locations are mapped to the Secure world by default.

SSE-310 supports a power domain for each bank: PD\_VM0, PD\_VM1. Each power domain contains only the actual volatile memory of the memory bank with all other logics of the memory banks, including the MPC, residing in PD\_SYS. Therefore, any power gating or retention refers only to the memory itself. For more information about related power control, see [Power integration](#).

All volatile memories run on **SYSSYSCLK**.

## 5.6 Timers and Watchdogs

SSE-310 supports two main classes of timers and watchdogs: Timestamp based timers and SLOWCLK AON based timers.



## 5.6.1 Timestamp based Timers

Timestamp based timers and watchdogs use the timestamp value that is provided on the System Timestamp Interface.

For more information about the System Timestamp Interface, see [System Timestamp Interface](#).

The four Timestamp-based timers support:

- Memory-mapped System Timer with register access through the Peripheral Interconnect
- 64-bit timestamp input, generating events through comparison with a timer value
- An 'auto-increment' feature to support regular event generation
- Down counter emulation
- Maskable level interrupt generation

The two Timestamp-based Watchdog timers are simplified timers that supports:

- Memory-mapped System Timer with register access through the Peripheral Interconnect
- 64-bit timestamp input, generating events through comparison with a timer value
- An 'auto-increment' feature to support refreshing the watchdog
- Watchdog reset request generation on double watchdog timeout
- Separate refresh register access frame to support refreshing from a different security or privilege level

See [Timestamp Timers](#) and [Timestamp Watchdogs](#) for more information about the timer registers.

SSE-310 provides four Timestamp Timers and two Timestamp Watchdog timers. These are mapped to the following addresses:

- Timer 0 at address 0x48000000 and aliased to 0x58000000
- Timer 1 at address 0x48001000 and aliased to 0x58001000
- Timer 2 at address 0x48002000 and aliased to 0x58002000
- Timer 3 at address 0x48003000 and aliased to 0x58003000
- Secure Watchdog timer control frame at address 0x58040000 and refresh frame at 0x58041000
- Non-secure Watchdog timer control frame at address 0x48040000 and refresh frame at 0x48041000

Timer 0, Timer 1, Timer 2, and Timer 3 can be configured by software to be a Secure or a Non-secure timer through the Peripheral Protection Controller that is controlled through the Secure Access Configuration Register Block. For more information, see [Secure Access Configuration Register Block](#). The security configuration of both watchdogs is non-configurable, one is always Non-secure and the other always Secure.

All timers and watchdog timers generate interrupts, and the Non-secure Watchdog can generate an extra interrupt on a second timeout event for notifying the Secure world to act. The Secure Watchdog can request a reset of the system if dual timeout occurs. The Non-secure Watchdog can be configured by software to do the same if required, but by default this is not allowed.

Except for Timer 3, all other timestamp timers and watchdogs reside in the PD\_SYS power domain and are reset by **nWARMRESETSYS**. Timer 3 resides in the PD\_AON power domain and is reset by **nWARMRESETAON**. Therefore Timer 3 can generate interrupts to wake the system even if the system is in the Hibernation state where PD\_SYS is turned off, or when it is in retention.

## 5.6.2 SLOWCLK AON Timers

SLOWCLK AON timers and watchdogs are simple CMSDK based 32-bit timers that run on **SLOWCLK**. They reside in the PD\_AON Power domain and are reset by **nWARMRESETAON**. A single timer and a single Secure Watchdog are provided. The Slow clock AON timer and watchdog are expected to be used when the system is in the HIBERNATION0 System Power state, when potentially only **SLOWCLK** is available and running, and all other clocks are off. These timers are mapped to the following addresses:

- SLOWCLK Timer at address 0x4802F000 and aliased to 0x5802F000
- SLOWCLK Secure Watchdog Timer at address 0x5802E000

The SLOWCLK Timer can be configured by software to be Secure or Non-secure access only, and configured for Privileged or Unprivileged access through the Peripheral Protection Controllers. The Peripheral Protection Controllers are controlled through registers in the Secure Access Configuration Register Block and the Non-secure Access Configuration Register Block.

For more information, see [Secure Access Configuration Register Block](#) and [Non-secure Access Configuration Register Block](#). The watchdog is Secure access only. All CMSDK timers and watchdog timers generate interrupts, and the **SLOWCLK** Secure Watchdog can request a Cold reset of the system if dual timeout occurs. For more information about CMSDK Timers, see *Arm® Cortex®-M System Design Kit Technical Reference Manual*.

## 5.7 Power Policy Units

SSE-310 leverages Power Policy Units (PPU) for power control of the Bounded Regions (BR) in the system. Access to the PPU is normally not required for normal operation because SSE-310 is architected to use the PPU primarily in dynamic mode, where the request to enter or leave a power state is managed and handshake using the PPU's Device interface. Therefore, the only time access to PPUs might be required is for Debug purposes. The PPUs are mapped to Secure address space as defined in [System Control Peripheral Region](#).

See [Power Policy Units](#) for more information.

## 5.8 Peripheral Protection Controllers

Peripheral protection Controllers in the system enable the software to control whether a peripheral is accessible to the Secure or Non-secure world, and to control the Privileged access or Unprivileged access.

For more information, see *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* and *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*. In SSE-310, peripherals that are aliased to two memory areas, one Secure and one Non-secure, are protected by PPCs. The PPC defines in which region the peripheral resides.

In SSE-310, Timers, the NPU, and other peripherals are protected using PPCs. These are controlled by the Secure Access Configuration Register Block and Non-secure Access Configuration Register Block.

For more information on these register blocks, see [Secure access configuration register block](#) and [Non-secure access configuration register block](#). These registers also control Security Control Expansion Signals to drive external PPCs.

Two groups of peripherals are defined in SSE-310, that are protected behind Peripheral Protection Controllers. These peripherals are:

- Peripheral Interconnect Peripheral Protection Controller Group 0. This group includes the following peripherals:
  - All Timestamp based Timers
  - NPU0
  - Watchdog Refresh Frames
  - Memory Protection Controller configuration register block
  - Secure Access Configuration Register Block
  - Non-secure Access Configuration Register Block
  - Message Handling Units
  - Watchdog Control Frames



Secure or Non-secure mapping of Watchdog Refresh Frames is fixed, only their privilege levels are configurable.

- 
- Peripheral Interconnect Peripheral Protection Controller Group 1. This group includes the following peripherals:
    - SLOWCLOCK CMSDK Timers
    - System Control Register Block
    - SLOWCLOCK Watchdog Timer
    - PPU0s



Note

Care should be taken if Unprivileged access to bus managers in the system is permitted, for example a DMA. If Unprivileged access is permitted, then Unprivileged code has the potential to use these managers to access and modify Privileged memory. Arm recommends that only Privileged programming access is permitted to these managers.

For more information, see [Secure Access Configuration Register Block](#).

## 5.9 Manager Security Controller

Manager Security Controllers (MSC) in the system transform memory transactions issued by non-CPU managers that are designed for A-Class systems, into memory transactions suitable to M-Class systems.

For more details, see:

- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*
- *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*

## 5.10 Memory Protection Controllers

Memory Protection Controllers (MPC) in the system partitions memory modules into pages and allows the software to define if each region is Secure or Non-secure.

In SSE-310, each memory page that is protected by the MPC, is aliased to two memory areas:

- Secure
- Non-secure

Depending on the security attributed defined for that page in the MPC by the software, the page either only exists in the Secure region or the Non-secure region.

A single MPC is provided for each VM and each is in the main memory as defined in [Peripheral Region](#).

For more information, see *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* and *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*.

## 5.11 Debug Infrastructure

SSE-310 does not support the CoreSight SoC-600 based common debug infrastructure.

For more details, see *Arm® Corstone™ Reference Systems Architecture Specification Ma1*.

When `EXPLOGIC_PRESENT = 1`, the subsystem expansion includes an example debug infrastructure that instances DAP-Lite2, a debug timestamp generator, TPIU-M, and MCU debug ROM table.

This infrastructure is the source and target of some debug related interfaces of CPU0. The remaining interfaces are made available as expansion interfaces. The following subsections describe the example debug infrastructure.

When `EXPLOGIC_PRESENT = 0`, all debug related interfaces of CPU0 are made available as expansion interfaces.

For more information, see [Debug and trace related interfaces](#).

### 5.11.1 DAP-Lite2

Arm® CoreSight™ DAP-Lite2 enables an off-chip debugger to connect to a target system using a low pin-count JTAG or Serial Wire interface. The debugger can then control the target processor during a debug session. DAP-Lite2 provides *Debug Access Port* (DAP) that is compliant with Arm® *Debug Interface Architecture Specification ADIv6.0*. DAP-Lite2 supports AMBA AHB5 debug access interface.

For more information about DAP-Lite2, see *Arm® CoreSight™ DAP-Lite2 Technical Reference Manual*.

`SOC_IDENTITY` describes how the SoC identity register values are used to generate the `TARGETID` of the SoC debug port in the expansion system.

### 5.11.2 Debug timestamp generator

The debug timestamp generator is implemented as a 64-bit binary up counter enabled when `CPUOTRCENA = 1`.

The debug timestamp generator resides in the `PD_DEBUG` power domain, runs on `DEBUGCPU0CLK`, and resides in the `nCOLDRESETDEBUGCPU0` reset domain.

### 5.11.3 TPIU-M

The TPIU-M is a *Trace Port Interface Unit* (TPIU) that is designed for use in single-processor systems based on Arm Cortex-M processors.

The TPIU-M bridges between the on-chip trace data from the *Embedded Trace Macrocell* (ETM) and the *Instrumentation Trace Macrocell* (ITM), with separate IDs, to a data stream.

See [Trace port interface](#) for pin-level details of the data stream interface.

The TPIU-M is accessible through the Private Peripheral Bus Region at address `0xE0040000` to `0xE0040FFF`.

The TPIU-M resides in the PD\_DEBUG power domain, runs on **DEBUGCPU0CLK**, and resides in the **nCOLDRESETDEBUGCPU0** reset domain.

For more information about TPIU-M, see *Arm® Coresight™ TPIU-M Technical Reference Manual*.

### 5.11.4 MCU debug ROM table

The MCU debug ROM table is accessible through the Private Peripheral Bus Region at address {CPU0MCUROMADDR, 0x000} to {CPU0MCUROMADDR, 0xFFF}. The default base address is 0xE00FE000.

[IIDR](#) describes how the subsystem implementation identity register values are used to generate the PIDR values of the MCU debug ROM table in the expansion system.

The MCU debug ROM table resides in the PD\_DEBUG power domain, runs on **DEBUGCPU0CLK**, and resides in the **nCOLDRESETDEBUGCPU0** reset domain.

## 5.12 System and Security Control

SSE-310 provides several registers in the system to allow various features of the system to be discovered, configured, and controlled. These registers are grouped into four register blocks:

### System Information Register Block

The System Information Register Block provides information on the system configuration and identity.

### System Control Register Block

The System Control Register Block implements registers for power, clocks, resets and other general system control.

### Secure Access Configuration Register Block

The Secure Access Configuration Register Block provides registers to configure Peripheral Protection Controllers (PPC) and Manager Security Controllers (MSC) that reside in the system and in the expansion system through the Security Control Expansion interface. These are Secure access-only registers.

### Non-secure Access Configuration Register Block

The Non-secure Access Configuration Register Block provides registers to configure Peripheral Protection Controllers (PPC) that resides in the system and in the expansion system through the Security Control Expansion interface. These are Non-secure access-only registers.

### Related information

[SYSINFO Register Block](#) on page 133

[System Control Register Block](#) on page 137

[Secure Access Configuration Register Block](#) on page 108

[Non-secure Access Configuration Register Block](#) on page 123

## 5.13 Power integration

Low-power operation is essential for IoT endpoint devices that might rely on a battery or on harvested energy. The implementation of multiple power-gated regions in the design reduces leakage power.

The power control infrastructure specifies:

- The power domains that the system logic and memories are partitioned into
- The control mechanisms that support the coordination of the operation of these different power domains.

This section describes the Power Control Infrastructure that SSE-310 supports.

[Power domain hierarchy and bounded regions](#) introduces terms that are used in the power-related descriptions.

[Power domains](#) depicts the distribution of SSE-310 components across the power domains.

[Power Policy Units](#) describes the configuration and control of the PPUs that control the power domains of the subsystem.

[Bounded region power modes](#) describes the valid power state combinations and state transitions of power domains in each Bounded Region.

[Power Control Wakeup Q-Channel Device interfaces](#) defines the sources that can wake up the subsystem.

[Power Dependency Control](#) defines the software configurable Power Dependency Control Matrix, which is responsible for keeping up power domains based on power domain states and external inputs.

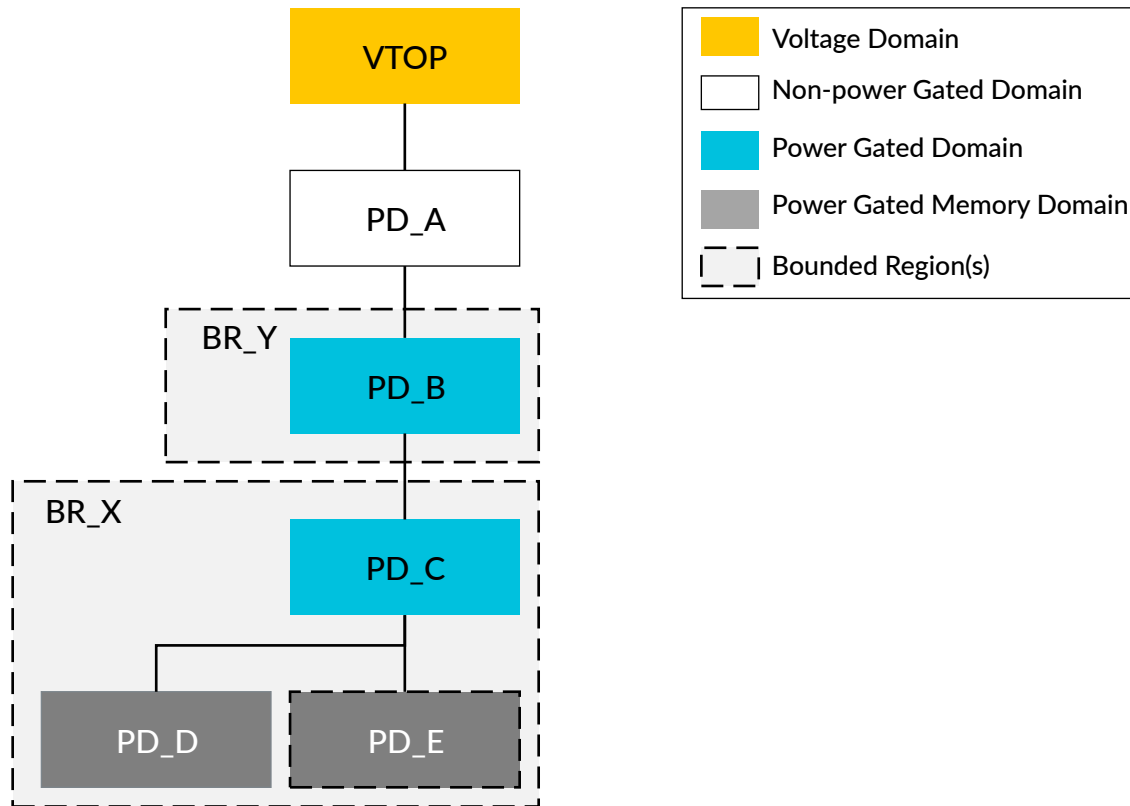
For the valid power state combinations of power domains in all Bounded Regions, see [System power states](#).

### 5.13.1 Power domain hierarchy and bounded regions

This section describes the relationships between the power and voltage domains.

The following figure shows a simple power domain hierarchy diagram that is used to describe relationships between power and voltage domains.

**Figure 5-4: Example power domain hierarchy**



In [Figure 5-4: Example power domain hierarchy](#) on page 72:

- Each rectangular block represents either a voltage domain or a power domain.
- Power Gated Domains are simply referred to as power domains in the descriptions.
- If a line connects two domains that are not at the same level in the same Bounded Region, there is a hierarchical relationship between the two domains.
- A block with dotted line indicates that the existence of the domain is configuration-dependent.

A rounded dotted bounding box over one or several domains indicate that their power states are controlled collectively. These boxes are called *Bounded Regions* (BR). For example, PD\_C, PD\_D, and PD\_E power domains are in the Bounded Region BR\_X. Power state transitions of a Bounded Region are controlled by a single *Power Policy Unit* (PPU). PPUs are complemented by LPI infrastructure components to bring together the quiescence status and control of IP blocks primarily within the power domains that are controlled by each PPU. The complemented PPUs are referred to as *Power Integration Kits* (PIKs).

The following hierarchical relationship rules are applicable to power state transitions of power domains with hierarchical relationship:

- When a higher-level power domain in the hierarchy has lower-level power gated domains below it, before the higher-level domain can enter a lower power state, the lower-level power



gated domains require to already be in a lower power state and remain in the same lower power state until the higher power gated domain completes its transition to a lower power state.

- When a lower-level power domain in the hierarchy has higher-level power domains above it, before the lower level domain can enter a different power state, the higher-level domains require to already be in their highest level power state.

The hierarchical relationship rules eliminate the possibility that simultaneous power state transitions of power domains with hierarchical relationship results in unintended power state combinations of the power domains. Unintended power state combinations do not occur even during power state transitions.

The power modes of a Bounded Region are represented using a state transition diagram that shows the valid power state combinations and transitions of power domains in the Bounded Region.

For example, the following figure shows a simple, four power mode transition diagram, for a Bounded Region with three power domains: PD\_B, PD\_C, and PD\_D.

- Each power mode is represented by a box that represents the power state of the power domain that is the highest in the hierarchy within the Bounded Region.
- The boxes include further boxes internally that represent the power states of the other power domains in the Bounded Region. A name is given to each power mode in bold.

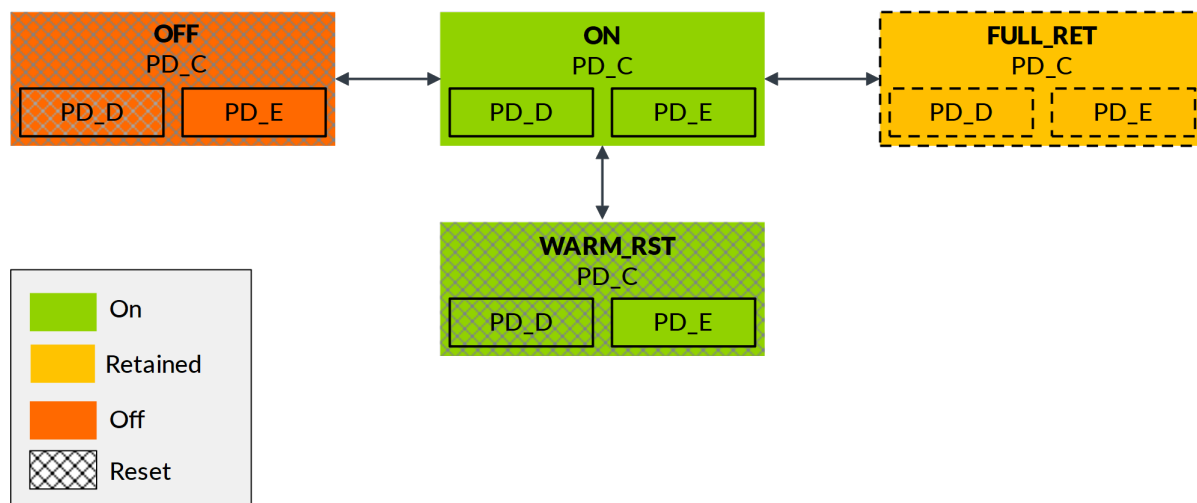
The following diagram only has four BR power modes, OFF, ON, WARM\_RST, and FULL\_RET. The different colors indicate the power state of the power domains. Patterned filled boxes indicate that reset can be asserted in the related domain. A dashed lined box indicates that the mode is optional (FULL\_RET in the example).



PD\_E is never reset because it is a memory power domain.

---

**Figure 5-5: Example power mode transition diagram of bounded power regions PD\_C, PD\_D, and PD\_E**



### 5.13.2 Power domains

SSE-310 is partitioned into multiple power domains.

These power domains are depicted in [Figure 5-6: SSE-310 power domains in the power aware integration layer](#) on page 75.

A block with a dotted line indicates that the existence of the domain is configuration-dependent.

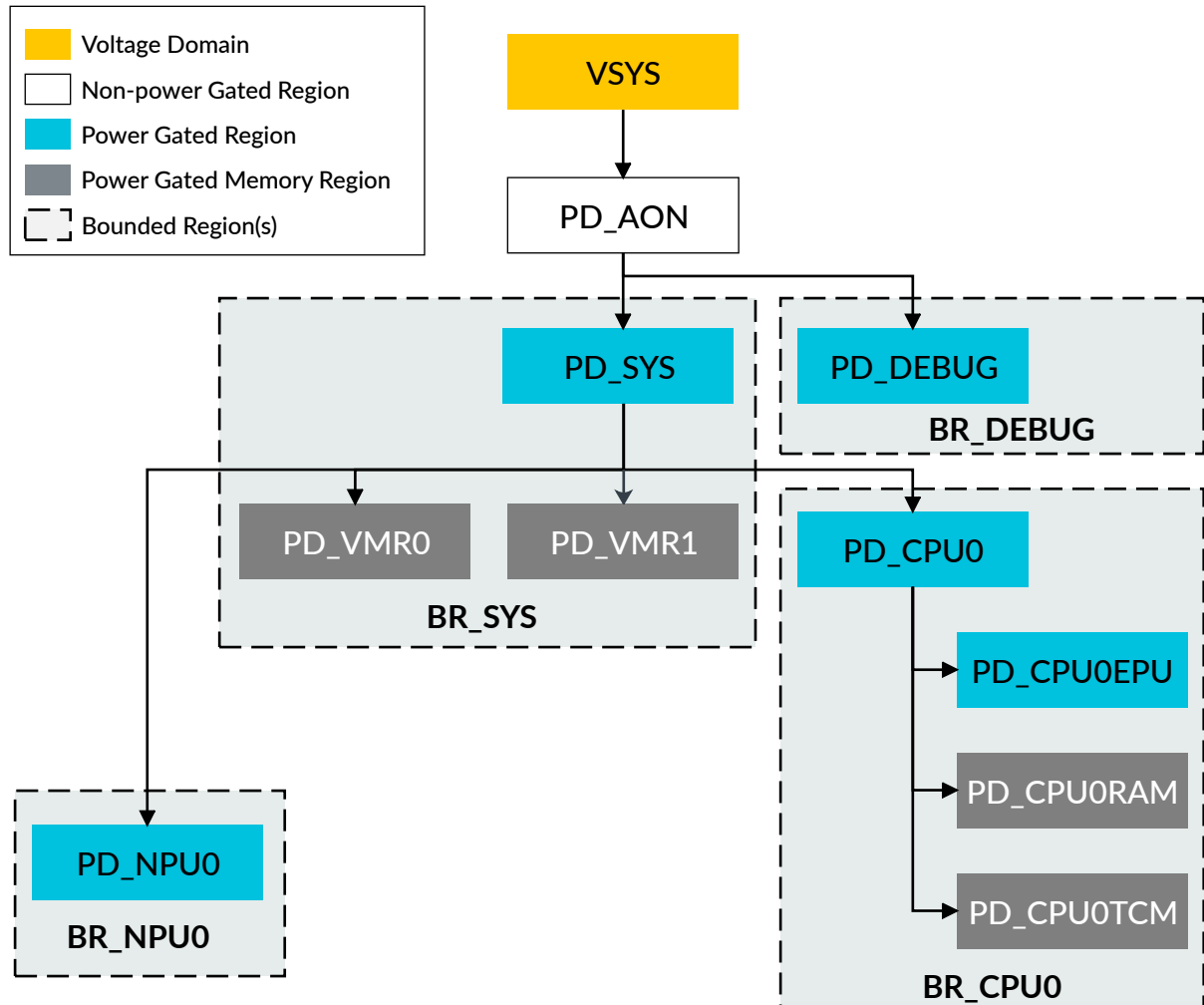
The following figure shows the example expansion logic, which is only included in the EXPLOGIC\_PRESENT=1 configuration of SSE-310.

[illegible]

These power domains are hierarchical and [Figure 5-7: Voltage and power domain hierarchy of SSE-310](#) on page 76 shows the voltage and power domain hierarchy.

For more information regarding how the power domains are controlled, see [Power Policy Units](#).

**Figure 5-7: Voltage and power domain hierarchy of SSE-310**



### 5.13.3 Power Policy Units

The SSE-310 uses *Power Policy Units* (PPUs) with P-Channel device interfaces for power control for each *Bounded Region* (BR) in the system, except for the MGMTPPU. The MGMTPPU does not control the power state of any power domains, since PD\_AON is always on. It facilitates Power-on reset, Cold reset, and Warm reset related state transitions and reset generation in the subsystem.

Device interface handshake is performed by all PPU's when transitioning to WARM\_RST. All PPU's are reset on Cold reset and reside in the PD\_AON power domain. The following table lists

additional configuration of the PPU along with the power domains and bounded regions that they control.

For more information about Power Policy Units, see *Arm® Power Policy Unit Architecture Specification* and *Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual*.

**Table 5-6: PPU associations and configurations**

Power domains controlled by the PPU	PPU configuration				
	PPU ID (BR ID)	OPMODE support	Default dynamic transition enable	Default power policy, default operation policy	Dynamic and static support of power modes
PD_SYS, PD_VMR0, PD_VMR1	SYSPPU (BR_SYS)	4 OPMODEs with independent use model	ON	OFF, 0	WARM_RST, ON, FULL_RET, MEM_RET, OFF
PD_CPU0, PD_CPU0EPU, PD_CPU0RAM, PD_CPU0TCM	CPU0PPU (BR_CPU0)	4 OPMODEs with independent use model	ON	OFF, 0	WARM_RST, ON, FUNC_RET, MEM_OFF, FULL_RET, LOGIC_RET, MEM_RET, OFF
PD_NPU0	NPU0PPU (BR_NPU0)	Not supported	ON	OFF, 0	WARM_RST, ON, OFF
PD_DEBUG	DEBUGPPU (BR_DEBUG)	Not supported	ON	OFF, 0	WARM_RST, ON, OFF
PD_AON	MGMTPPU (NA)	Not supported	ON	ON, 0	WARM_RST, ON, OFF

All PPUs are configured to perform device interface handshake when transiting from ON to WARM\_RESET (WARM\_RST\_DEVREQEN\_CFG = 1).

The default power mode of most PPUs is OFF. The subsystem implements a hardware autonomous power-up sequence without any software interaction.

See [Power up after Cold reset](#) for more information.

The write accessibility of the PPU registers is controlled through the PWRCTRL.PPU\_ACCESS\_FILTER. When it is set to 0b1, the system blocks all write accesses to the PPUs by ignoring the writes, except for the following registers for each PPU:

- Interrupt Mask Register, at address offset 0x030
- Additional Interrupt Mask Register, at address offset 0x034
- Interrupt Status Register, at address 0x038
- Additional Interrupt Status Register, at address 0x03C

When PWRCTRL.PPU\_ACCESS\_FILTER is set to '0', all PPU registers are freely accessible to the Secure world.



Arm recommends that the software does not disable any PPU access filtering and configure registers, except the IRQ-related registers listed previously. Disabling non-IRQ related registers can cause system deadlock.

---

## 5.13.4 Bounded Region power modes

The following subsections describe the valid power state combinations and state transitions of power domains in each Bounded Region.

### 5.13.4.1 BR\_DEBUG power modes

When a Warm reset is requested, the Bounded Region transitions to the WARM\_RST Power Mode. The transition makes sure that the debug logic enters a safe state so that the system can be Warm reset cleanly.

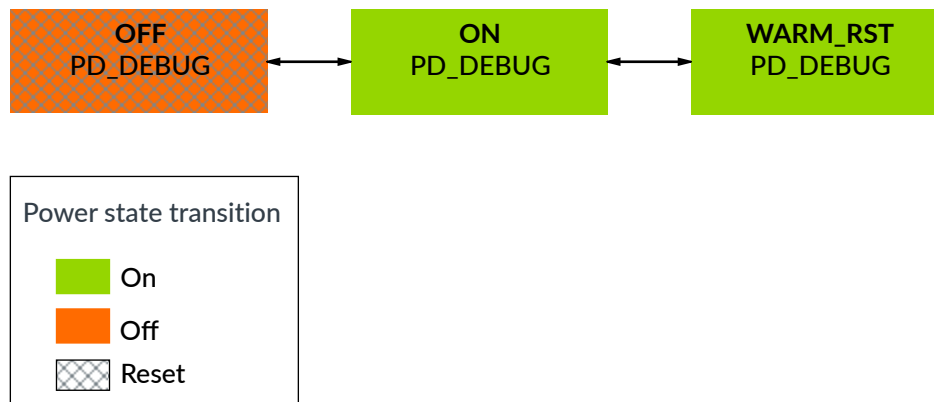


Logic in the PD\_DEBUG power domain is not reset in the WARM\_RST Power Mode.

---

The following figure shows the power modes supported by BR\_DEBUG.

**Figure 5-8: BR\_DEBUG power mode transition diagram**

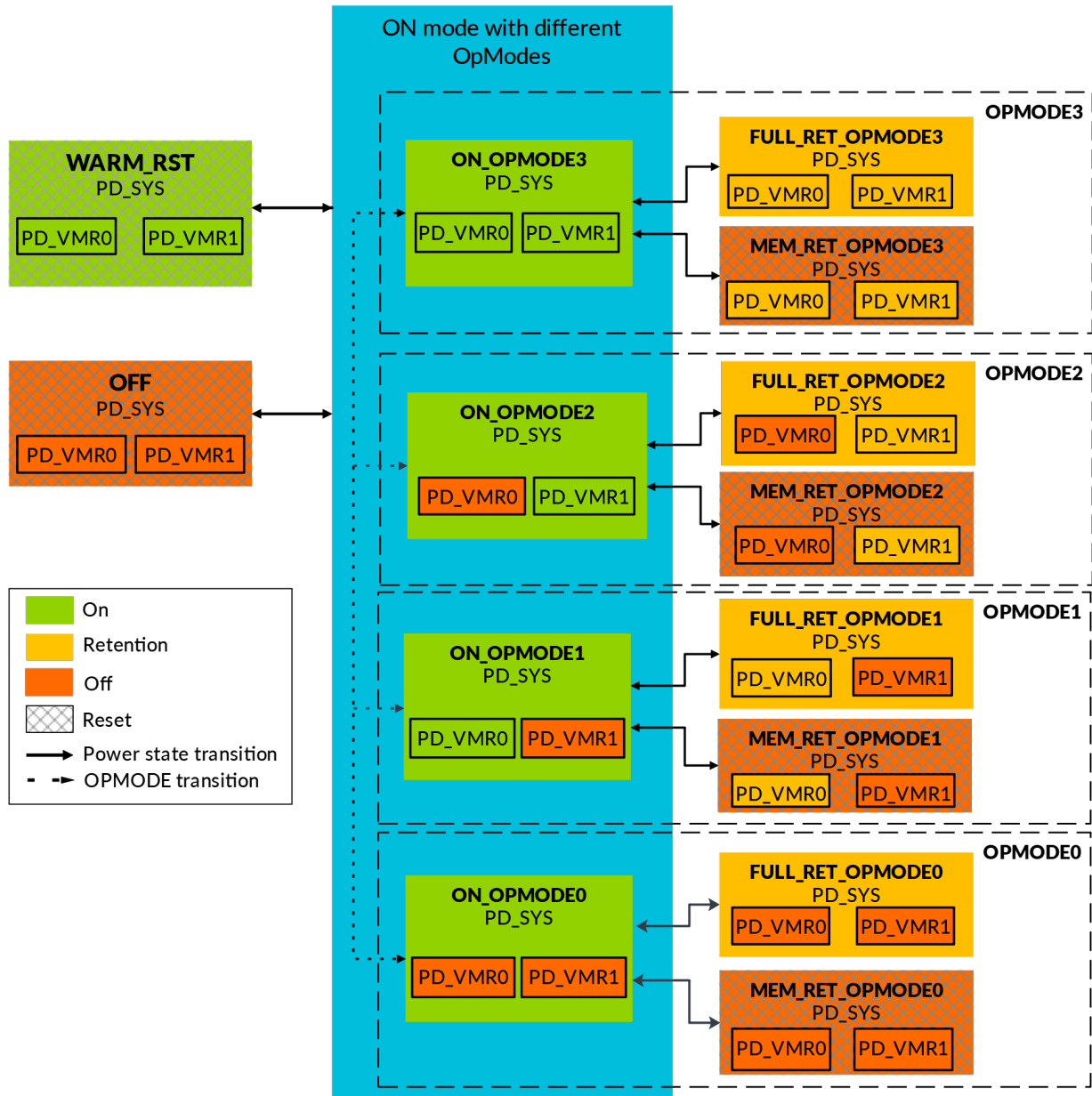


#### 5.13.4.2 BR\_SYS power modes

This section describes the transitions between the power modes that BR\_SYS supports.

The following figure shows the power modes that BR\_SYS supports.

**Figure 5-9: BR\_SYS power mode transition diagram**



In SSE-310, logic retention power states are remapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells.

BR\_SYS supports four OPMODES (OPMODE[0-3] in [Figure 5-9: BR\\_SYS power mode transition diagram](#) on page 80).

OPMODEs are encoded as binary 2-bit values with each bit representing the power state of a memory power domain, PD\_VMR0 and PD\_VMR1.



Changing OPMODEs is possible only when transitioning between the different ON power modes, except when entering and exiting the OFF or WARM\_RST power modes.

SSE-310 supports the following power modes:

- OPMODE0: PD\_VMR0 and PD\_VMR1 are both OFF.
- OPMODE1: PD\_VMR0 is ON<sup>1</sup> and PD\_VMR1 is OFF.
- OPMODE2: PD\_VMR0 is OFF and PD\_VMR1 is ON<sup>1</sup>.
- OPMODE3: PD\_VMR0 and PD\_VMR1 are both ON<sup>1</sup>.

<sup>1</sup> Context saving power state depends on the PD\_SYS power state when PD\_SYS is in:

- ON or WARM\_RST, then the context saving power state is ON
- FULL\_RET or MEM\_RET, then the context saving power state is RET

When a Warm reset is requested, the bounded region transitions to WARM\_RST through other power modes, only after the PD\_SYS power domain is idle and ready for being Warm reset.

#### 5.13.4.2.1 Controlling the PD\_VMR<i> minimum power states

Software must configure the minimum power state of each PD\_VMR<i>, as SSE-310 supports NUMVMBANK=2, there are PD\_VMR0 and PD\_VMR1 power domains.

To configure the minimum power state of each PD\_VMR<i>, software must configure the register fields PDCM\_PD\_VMR<i>\_SENSE.MIN\_PWR\_STATE as follows:

- Set to 0b00 to set the minimum power state of the PD\_VMR<i> to OFF.

PD\_VMR<i> only ever transitions between ON and OFF state, and it is never in retention, meaning that in low-power state, all states in PD\_VMR<i> are lost.

With this setting, when PD\_SYS is ON and the BR\_SYS is in one of the ON\_OPMODE<i> power modes where PD\_VMR<i> is ON, BR\_SYS transitions to another ON\_OPMODE<i> where PD\_VMR<i> is OFF automatically once the PD\_VMR<i> domain is idle.

Therefore, PD\_VMR<i> is expected to turn OFF quickly once PDCM\_PD\_VMR<i>\_SENSE.MIN\_PWR\_STATE is set to 0b00. Once PD\_VMR<i> is OFF, it can only be returned to ON by an access on the bus targeting PD\_VMR<i>.

However, following that, when PD\_VMR<i> is idle again, PD\_VMR<i> turns OFF again. To avoid this, configure PDCM\_PD\_VMR<i>\_SENSE.MIN\_PWR\_STATE to a non-zero value before accessing the PD\_VMR<i>.

- Set to 0b01 to set the minimum power state of the PD\_VMR<i> to RET. PD\_VMR<i> only ever transitions between ON and retention state, and never be in the OFF state. Therefore, in the low-power System Power States, all register states in PD\_VMR<i> are retained. BR\_SYS never transitions to a power mode that has PD\_VMR<i> turned off.

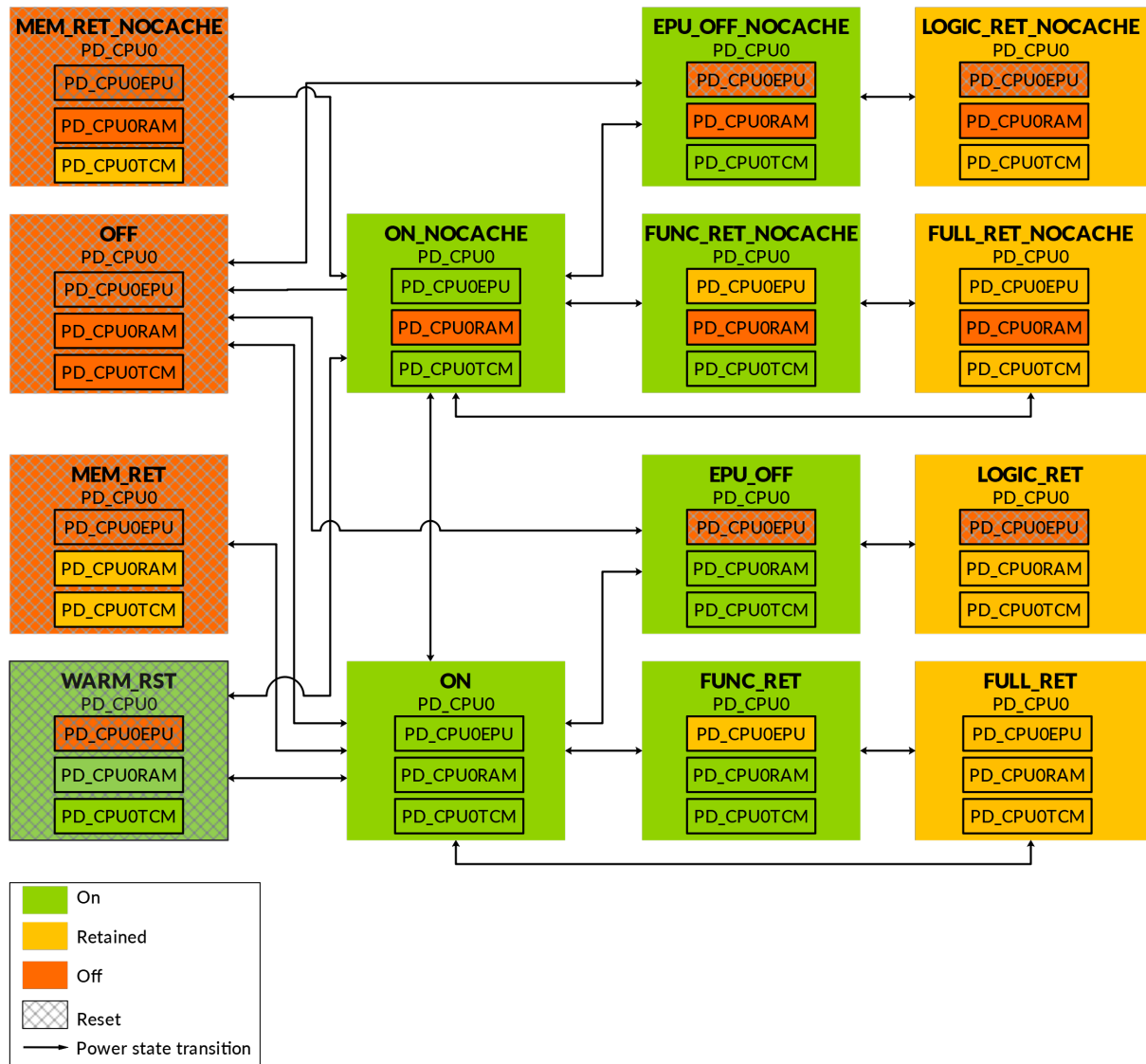


To place the PD\_VMR<i> into Retention, the BR\_SYS has to enter one of the FULL\_RET\_OPMODE<i> power modes or MEM\_RET\_OPMODE<i> power modes where PD\_VMR<i> is in Retention. This means that it is not possible to place a PD\_VMR<i> into Retention while keeping PD\_SYS ON.

### 5.13.4.3 BR\_CPU0 power modes

BR\_CPU0 supports multiple power modes.

**Figure 5-10: BR\_CPU0 power mode transition diagram**



During pseudo power mode transitions, the physical power state of the power domains does not change. The power domains themselves, for example PD\_CPU0, can observe pseudo transitions when the CPUOPPU exits the various MEM\_RET power modes.

Pseudo transitions are responsible for initializing logic that is turned on as part of leaving the various MEM\_RET power modes. Pseudo transitions are transient as they are followed by requests to the various ON power modes immediately. In the PPU and in the PCSM, only state changes that are part of the transitions between the various MEM\_RET and ON power modes are performed. The PPU and the PCSM do not enter the OFF power mode as part of the pseudo transition.

In SSE-310, logic retention power states are remapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells.

The EPU\_OFF and EPU\_OFF\_NOCACHE power modes are mapped to MEM\_OFF power mode of the PPU that controls BR\_CPU0. Therefore, the MEM\_OFF power mode of the BR\_CPU0 PPU controls the power mode of logic in PD\_CPU0EPU power domain, and not the power mode of a memory as the name might indicate.

The PD\_CPU0TCM power state usually matches that of PD\_CPU0. The exceptions are the power modes MEM\_RET and MEM\_RET\_NOCACHE, where PD\_CPU0TCM is retained. The choice of transitioning from ON\_NOCACHE to either MEM\_RET\_NOCACHE or OFF is determined by the minimum power state register configuration CPUPWRCFG.TCM\_MIN\_PWR\_STATE of the CPU0 local TCM.

When a Warm reset is requested, the bounded region transitions to the WARM\_RST through other power modes once the domains are idle and ready for reset.

#### 5.13.4.3.1 Controlling PD\_CPU0RAM power state

The BR\_CPU0 bounded region uses operating modes to support the ability to turn on or off the cache RAMs in modes other than the OFF mode.

Power modes with cache RAMs disabled, called the NOCACHE operating modes, are suffixed with “NOCACHE”. Power modes with cache RAMs enabled, called the CACHE operating modes, are modes without the “NOCACHE” suffix.

To select the use of the NOCACHE operating modes, configure the following registers:

- Set the CPDLPSTATE.RLPSTATE register in the processor to OFF, which is 0b11.
- To disable data cache, set the MSCR.DCACTIVE register in the processor to 0.
- To disable instruction cache, set the MSCR.ICACTIVE register in the processor set to 0.

Transitions between the two operating modes can only occur between the ON and ON\_NOCACHE power modes. In the NOCACHE operating modes, the PD\_CPU0RAM is turned off. When operating in the CACHE operating modes, PD\_CPU0RAM is retained in the MEM\_RET, LOGIC\_RET, or FULL\_RET power modes.

#### 5.13.4.3.2 Controlling PD\_CPU0EPU power state

The PD\_CPU0EPU and PD\_CPU0 can only enter WARM\_RST together. However, the BR\_CPU0 bounded region lets PD\_CPU0EPU enter a lower power state independently while the PD\_CPU0 is ON.

To control whether the PD\_CPU0EPU can be allowed to enter the retention (RET) or OFF state, the software can set the register CPDLPSTATE.ELPSTATE in the CPU as follows:

##### **RET**

Value: 0b10

The EPU enters retention state only when in low power state. When CPDLPSTATE.ELPSTATE is set to RET, BR\_CPU0 never enters EPU\_OFF, EPU\_OFF\_NOCACHE, LOGIC\_RET\_NOCACHE, LOGIC\_RET, OFF, and MEM\_RET states.

##### **OFF**

Value: 0b11

The EPU enters OFF state only when in lower power state. When CPDLPSTATE.ELPSTATE is set to OFF, BR\_CPU0 never enters FUNC\_RET, FUNC\_RET\_NOCACHE, FULL\_RET\_NOCACHE and FULL\_RET states.

##### **ON**

Value: 0b00 or 0b01

The EPU stays on. When CPDLPSTATE.ELPSTATE is set to ON, BR\_CPU0 never enters any power modes to the right of the power modes ON\_NOCACHE or ON in BR\_CPU0 power mode (see [Figure 5-10: BR\\_CPU0 power mode transition diagram](#) on page 82 except for FULL\_RET and FULL\_RET\_NOCACHE.

#### 5.13.4.3.3 Entering lower PD\_CPU0 power states

For the PD\_CPU0 to enter a lower power state, the software on the CPU0 must first configure its CPDLPSTATE.CLPSTATE register to define what power state it can enter when in a lower power state.

The register settings are as follows:

##### **RET**

Value: 0b10

The PD\_CPU0 enters retention state only when in low power state. When CPDLPSTATE.CLPSTATE is set to RET, BR\_CPU0 never enters OFF, MEM\_RET\_NOCACHE, or MEM\_RET.

##### **OFF**

Value: 0b11

The PD\_CPU0 enters off state only when in low power state. When CPDLPSTATE.CLPSTATE is set to OFF, BR\_CPU0 never enters LOGIC\_RET\_NOCACHE, and LOGIC\_RET modes. Other modes like FULL\_RET and FULL\_RET\_NOCACHE can be entered depending on CPDLPSTATE.ELPSTATE and the current operating mode.

## ON

Value: 0b00 or 0b01

The PD\_CPU0 stays on. When CPDLPSTATE.CLPSTATE is set to ON, BR\_CPU0 never enters LOGIC\_RET\_NOCACHE, LOGIC\_RET, FULL\_RET\_NOCACHE, FULL\_RET, OFF, MEM\_RET\_NOCACHE, and MEM\_RET modes

For the PD\_CPU0 to then enter a lower power state, the CPU must enter DEEPSLEEP enabled WFI state. In SSE-310, DEEPSLEEP always utilizes an EWIC. An *External Wakeup Interrupt Controller* (EWIC) for the CPU always exists, and the EWIC resides in the PD\_AON domain. HASCPU0IWIC is 0 in SSE-310, so for CPU0, the *Internal Wakeup Interrupt Controller* does not exist, and the EWIC is always used.

The following list shows the types of power states that the CPU supports from a programmer's point of view, and how to enter each:

### The “OFF – DEEPSLEEP” state

Allows the CPU to turn off but utilizes interrupts through the EWIC to wake the CPU. To enter this state, the CPU must perform the following actions before entering WFI:

1. Select to use the EWIC by setting the CPUPWRCFG.USEIWIC register.
2. Set the CPU0's CPDLPSTATE.CLPSTATE to OFF.
3. Enable DEEPSLEEP.

### The “RET – DEEPSLEEP” state

Allows the CPU to enter retention state and utilizes interrupts through the EWIC to wake the CPU. To enter this state, the CPU must perform the following actions before entering WFI:

1. Select to use the EWIC by setting the CPUPWRCFG.USEIWIC register.
2. Set CPDLPSTATE.CLPSTATE to RET.
3. Enable DEEPSLEEP, before entering WFI.

### The “ON – DEEPSLEEP” state

Allows the CPU to enter a low-power state that only supports stopping the CPU clock internally, including to the NVIC. The EWIC is used to wake the CPU. To enter this state, the CPU must perform the following actions before entering WFI:

1. Select to use the EWIC by setting the CPUPWRCFG.USEIWIC register.
2. Set CPDLPSTATE.CLPSTATE to ON.
3. Enable DEEPSLEEP, before entering WFI.

### The “ON – Sleep” state

Allows the CPU to enter a low-power state that still is ON keeping its NVIC clocking and running with the rest of the core clock turned off. To enter this state, the CPU must not enable DEEPSLEEP before entering WFI or WFE. WFE can be used only in this CPU low-power state if the intention is to wake using the event interface of the CPU.

### The “ON” state

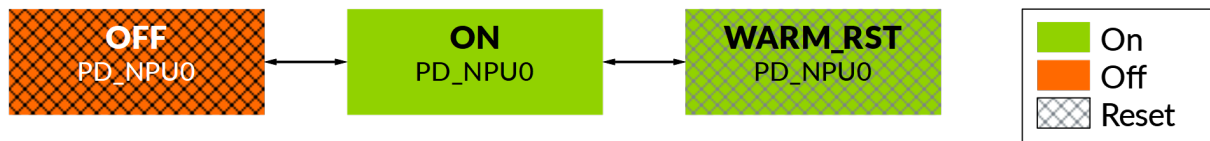
Is the CPU normal running state. In this state, the EPU and the RAMs in the CPU have a degree of separate control as detailed in the sections [Controlling PD\\_CPU0RAM power state](#) and [Controlling PD\\_CPU0EPU power state](#) respectively.

#### 5.13.4.4 BR\_NPU0 power modes

BR\_NPU0 supports multiple power modes.

- On Cold reset, the bounded domain enters the ON mode.
- When a Warm reset is requested, the bounded region transitions to the Warm reset mode only after all dependent domains are idle and ready for reset.

**Figure 5-11: BR\_NPU0 power mode transition diagram**



The NPU has a power control register that can be programmed to prevent the system powering off. For more details of this register, see *AMBA® Ethos™-U55 NPU Technical Reference Manual*.

After reset, once the NPU enters an idle state, it allows powering down of the BR\_NPU0 domain.

#### 5.13.5 Wake-up sources

The PD\_CPU0 power domain can be woken-up from low-power mode using one of the following wake-up sources – which also wakes-up the PD\_SYS power domain via power hierarchy:

- Interrupts with EWIC support, see [Interrupt interfaces](#).
- CPU0EDBGRQ input.
- PWRCPU0WAKE Q-Channel Device interface.

For more details, see [Power Control Wakeup Q-Channel Device interfaces](#).

Other power domains can be woken up by the following signals:

- PWRMGMTWAKE Q-Channel Device interface for PD\_AON. Given that PD\_AON is always on, this interface should be tied LOW.

- PWRSYSWAKE Q-Channel Device interface for PD\_SYS.
- PWRDEBUGWAKE Q-Channel Device interface for PD\_DEBUG.

For more details, see [Power Control Wakeup Q-Channel Device interfaces](#).

### 5.13.6 Power Dependency Control

SSE-310 defines a control matrix that allows the power mode (ON state) of one domain to affect another power domain.

You can program the matrix and change how the SSE-310 performs dynamic power transitions through PDCM registers [PDCM\\_PD\\_SYS\\_SENSE](#) and [PDCM\\_PD\\_VMR<M>\\_SENSE](#).

The *Power Dependency Control Matrix* (PDCM) can reduce the software interactions that are required for system management. This increases the responsiveness of the system and reduces its power consumption.

[Table 5-7: Power dependency control matrix](#) on page 87 shows how the PD\_SYS, PD\_VMR0, and PD\_VMR1 are affected by the power dependency inputs.

The right columns of the table list the power domains (PD\_SYS, PD\_VMR0, and PD\_VMR1) that are being controlled.

The left column lists the Power dependency inputs, which are:

- The sources of [Power Domain ON Status Signals](#), **PDSYSON**, **PDCPU0ON**, and **PDNPU0ON**
- The Q-Channel signals of [Expansion Power Control Dependency Interface](#), **PDCMONQREQn**, and **PDCMRETQREQn**

If a power domain is sensitive to a dependency input, you can use the power dependency input signals to keep the power domain on and not allow transition to lower power modes. Therefore, the PDCM is used primarily to define when a power domain should not enter a lower power state. PDCM supports keeping power domains on, it is not designed to support powering up of any power domain.

**Table 5-7: Power dependency control matrix**

Power Dependency Inputs/ Power Domain	PD_SYS	P_VMR0	PD_VMR1
PD_SYS_ON	Conf	-	-
PD_CPU0_ON	Y	Conf	Conf
PD_NPU0_ON <sup>1</sup>	Y	Conf	Conf
PDCMONQREQn[{0-3}]	Conf	Conf	Conf
PDCMRETQREQn[{0-3}]	Conf	Conf	Conf

<sup>1</sup> PD\_NPU0 row does not exist if NUMNPU = 0

“Conf” indicates that it is software configurable

“Y” indicates that it is always sensitive to the respective dependency input

## PD\_SYS

PD\_SYS can be software configured to be sensitive to the ON state of PD\_SYS and all Expansion Power Control Dependency inputs. PD\_SYS is always sensitive to the ON state of PD\_CPU0, PD\_VMR0, PD\_VMR1 and PD\_NPU0.

When software configures PD\_SYS to be sensitive to:

- PD\_SYS\_ON (self), then PD\_SYS remains ON once it is ON.
- **PDCMONQREQn**, then once it is ON PD\_SYS remains ON as long as PDCMONQREQn==PDCMONACCEPTn==1
- **PDCMRETQREQn**, then once it is ON PD\_SYS changes state to FULL\_RET instead of OFF/MEM\_RET as long as PDCMRETQREQn==PDCMRETQACCEPTn==1

## PD\_VMR<i>

PD\_VMR<i> can be software configured to be sensitive to the ON state of PD\_CPU0, PD\_NPU0, and all [Expansion Power Control Dependency Interface](#) inputs.

When software configures PD\_VMR<i> to be sensitive to:

- PD\_CPU0\_ON, then once it is ON, PD\_VMR<i> remains ON as long as PD\_CPU0\_ON==1 (PD\_CPU0 is ON)
- PD\_NPU0\_ON, then once it is ON, PD\_VMR<i> remains ON as long as PD\_NPU0\_ON==1 (PD\_NPU0 is ON)
- PDCMONQREQn, then once it is ON, PD\_VMR<i> remains ON as long as PDCMONQREQn==PDCMONACCEPTn==1
- PDCMRETQREQn, then once it is ON, PD\_VMR<i> will change state to RET instead of OFF as long as PDCMRETQREQn==PDCMRETQACCEPTn==1



Note

Because of PD\_VMR0 and PD\_VMR1 are controlled by PD\_SYS PPU, the PD\_SYS PPU changes state to MEM\_RET instead of OFF as long as **PDCMRETQREQn** for PD\_VMR<i> is in effect, but this does not affect the power state of the logic in PD\_SYS.

## Power domain minimum power state

The SSE-310 provides programmable registers for the PD\_SYS and each of the PD\_VMR<i> power domains ([PDCM\\_PD\\_SYS\\_SENSE](#) and [PDCM\\_PD\\_VMR<M>\\_SENSE](#)), which define the lowest power state that each domain can enter.

The minimum power states of the power domains PD\_DEBUG, PD\_NPU0, and PD\_CPU0 are not affected by the PDCM registers. The following table shows the minimum power states of the power domains.

**Table 5-8: Power domain minimum power state**

Power domain	Supported MIN_PWR_STATE for each domain
PD_SYS	ON, OFF, Retention



Power domain	Supported MIN_PWR_STATE for each domain
PD_VMR<i>	ON, OFF, Retention

The MIN\_PWR\_STATES of both PD\_SYS and PD\_VMR<i> affect the SYSPPU.

For example, the SYSPPU tries to enter the bounded region collectively to a low-power state if both the following conditions are met:

- PD\_SYS is idle.
- All domains that PD\_SYS depends on are not ON.

If MIN\_PWR\_STATE of PD\_SYS is set to Retention, BR\_SYS is not allowed to enter the OFF mode nor any of the MEM\_RET\_OPMODE<m> modes because PD\_SYS is not allowed to turn off. SYSPPU tries to enter one of the associated FULL\_RET\_OPMODE<m> states.

In another example, if all the following conditions are met, when SYSPPU is entering a low-power state:

- MIN\_PWR\_STATE of PD\_VMR0 is ON.
- MIN\_PWR\_STATE of all other PD\_VMR<i> is OFF.
- All other PD\_VMR<i> are ON.

Then the SYSPPU transitions to ON\_OPMODE1 state to turn off all other PD\_VMR<i>. It never enters FULL\_RET\_OPMODE1 or MEM\_RET\_OPMODE1.

### 5.13.7 System power states

The system power states in SSE-310 are bounded by the relationships and minimum power states defined in the various Power Dependency Control Matrix Sensitivity registers and the low-power control registers in the CPU.

SSE-310 defines the following system power states:

#### **SYS\_OFF**

All voltage and power domains are OFF.

#### **HIBERNATION0**

The voltage domain is ON, and the system is in the lowest power state that can still be woken from sleep. At wake, the system has to reboot.

#### **SYS\_RET**

The system is in retention, and at wake, the system can continue to execute since no system state is lost.

#### **SYS\_ON**

The system is ON.

In SSE-310, logic retention power states are remapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells.

Table 5-9: System power states on page 90 defines the following for each System Power State:

- Supported power states of the power domains.
- Voltage supply state.
- Input clock state (SYSCLK, CPU0CLK and NPU0CLK).

The subsystem does not support state combinations that are not tabulated. For the definition of the states in the PD\_CPU0 column, see [Entering lower PD\\_CPU0 power states](#).

**Table 5-9: System power states**

System Power State	VSYS(PD_AON)	PD_SYS	PD_CPU0	PD_DEBUG	PD_VMR<i>	PD_NPU0	Clocks
SYS_OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
HIBERNATION0	ON	OFF	OFF-DeepSleep <sup>2</sup>	OFF/ON	OFF/RET	OFF	ON/OFF <sup>1</sup>
SYS_RET	ON	RET	OFF-DeepSleep <sup>2</sup> RET-DeepSleep	OFF/ON	OFF/RET	OFF	ON/OFF <sup>1</sup>
SYS_ON	ON	ON	OFF-DeepSleep <sup>2</sup> ON-DeepSleep ON-Sleep ON	OFF/ON	OFF/ON	OFF/ON	ON/OFF <sup>1</sup>

<sup>1</sup> The subsystem requests clocks based on the Clock Force register and the power states of PD\_DEBUG, PD\_SYS, PD\_NPU0 and PD\_CPU0. The relationship of the clocks and the power states is defined in the Power State Based High-Level Clock Gating related descriptions in *Arm® Corstone™ SSE-310 Example Subsystem Configuration and Integration Manual*.

<sup>2</sup> When PD\_CPU0 is in the state OFF-DeepSleep, PD\_CPU0TCM can be either OFF or retained as defined by CPUPWRCFG.TCM\_MIN\_PWR\_STATE. In this state, PD\_CPU0EPU must be OFF. The PD\_CPU0RAM can be OFF permanently, which is defined by CPDLPSTATE.RLPSTATE.

The following points can be made, based on [Table 5-9: System power states](#) on page 90:

- When waking up any of the PD\_CPU0 power domains, if the PD\_SYS is OFF or RET, the PD\_SYS domain is automatically woken to ON. This is because the PD\_CPU0 ON state is only supported in the SYS\_ON System Power State.
- PD\_DEBUG can be woken independently, except in the SYS\_OFF state.
- In HIBERNATION0 and SYS\_RET, PD\_VMR<i> cannot be woken independently from PD\_SYS, since they belong to the bounded region BR\_SYS.
- SSE-310 does not support debug scenarios when the processor is not in ON state. When EXPLOGIC\_PRESENT=1 the provided example expansion logic wakes up PD\_CPU0 to ON when PD\_DEBUG is ON.

An additional transient WARM\_RST state also exists for the system when a Warm reset is performed. This state can only be entered from the SYS\_ON state with PD\_CPU0 being ON. In this

state, all power domains, including PD\_AON, temporarily enter the WARM\_RST power mode and exit it when Warm reset is completed.

#### 5.13.7.1 Power up after Cold reset

There is a power up sequence implemented that wakes the subsystem on each Cold reset and Power-on reset.

Due to the hierarchical power management, the power-on sequence is:

1. PD\_DEBUG and PD\_SYS
2. PD\_CPU0 and PD\_NPU0

If there is no request towards PD\_DEBUG, when the power up sequence completes, the DEBUGPPU turns off.

If there is no request towards PD\_NPU0, when the power up sequence completes, the NPU0PPU turns off.

Power modes entered during the power up by SYSPPU, CPU0PPU, NPU0PPU, and DEBUGPPU are in turn:

Power modes entered during the powerup:

- SYSPPU: ON\_OPMODE0
- CPU0PPU: EPU\_OFF\_NOCACHE
- NPU0PPU: ON
- DEBUGPPU: ON

#### 5.13.7.2 Entering HIBERNATION0

To enter the HIBERNATION0 state, the following conditions must be met:

- PD\_NPU0 must be in OFF state.
- The Minimum Power State in the registers PDCM\_PD\_VMR0\_SENSE and PDCM\_PD\_VMR1\_SENSE has to be set to OFF or RET
- The Minimum Power State in the register PDCM\_PD\_SYS\_SENSE has to be set to OFF
- Timestamp based System Timers 0, 1, and 2, along with all Timestamp-based Watchdogs must be disabled (these reside in PD\_SYS). If timers, watchdogs, or both are needed, the PD\_AON modules can be used: System Timer-3, SLOWCLK Timer, SLOWCLK Watchdog, or both.
- Interrupt status of enabled interrupts has to be cleared in the following registers: SECPPCINTSTAT, SECMSCINTSTAT, and BRGINTSTAT
- Interrupt status of internal and Expansion Memory Protection Controllers has to be cleared. The interrupt status is visible through the SECMPICINTSTAT register and can be cleared by of accessing the respective Memory Protection Controller.

- If a VM is expected to be used immediately after exiting HIBERNATION0. For example, to hold the stack, use one of the following settings:
  - Set sensitivity to the PD\_CPU0 ON state in the related PDCM\_PD\_VMR<i></i>\_SENSE register before entering HIBERNATION0.  
  
This ensures that on the first access to the VM after leaving HIBERNATION0 (which has caused the VM to power up) the VM stays powered until PD\_CPU0 turns OFF.
  - Set the minimum power state in the PDCM\_PD\_VMR<i></i>\_SENSE register to RET, so that the VM content is always retained once it is turned ON
- If there is an Expansion logic in the PD\_SYS that is connected to the PD\_SYS Power P-Channel Interface, the expansion logic must be idle and able to enter a quiescent state.
- If EWIC is to wake the subsystem, PD\_CPU0 must enter the “OFF-DeepSleep” with EWIC enabled. For more information, see [Entering lower PD\\_CPU0 power states](#).

### 5.13.7.3 Wake from HIBERNATION0 using the PWRSYSWAKE Q-Channel Device Interface

Components in the Expansion can use the PWRSYSWAKE Q-Channel Device Interface, among others, to wake the subsystem from HIBERNATION0.

This interface does not wake PD\_CPU0 directly, accesses to the subsystem can result in a wake interrupt on the EWIC.



A wake request on the **PWRSYSWAKEQACTIVE** input automatically results in a request for **SYSCLK** to be active.

Arm recommends that you use interrupt signals on the EWIC, as this provides a more robust approach to wake the subsystem from HIBERNATION0.

This allows a request to wake the CPU along with the subsystem (PD\_SYS), so the CPU can configure the subsystem before allowing access to it from Expansion managers. To delay access from Expansion managers, the system integrator must deploy access control gates at the subordinate expansion interfaces, which are facilitated by the register BUSWAIT, the configuration ACCWAITNRST, and the signal **ACCWAITn**.

If the Expansion manager intends to wake and access peripherals or memories within SSE-310 without waking the CPU as well, the Expansion manager accessing the subsystem must be a Secure manager. Alternatively, a Non-secure manager can be restricted externally to access a strictly controlled region of memory residing outside of the SSE-310, which is always Non-secure and therefore does not pose a security risk.

When the subsystem wakes without the CPU restoring the configuration of all MPCs and PPCs in the subsystem, Secure managers in the Expansion see all Non-secure memory spaces as Secure. If

any of these memories were retained before wake up, ensure that these memory locations are not used for code execution.

Arm does not recommend that you use only **PWRSYSWAKEQACTIVE** to wake the subsystem from HIBERNATION0, as it is limited.

This is because when PD\_SYS is awakened from HIBERNATION0, all registers in the power domain are in their reset state, and all peripherals that reside behind PPCs or MPCs defaults to Secure access only. Often, this requires waking and booting the CPU so that it can configure the subsystem before allowing accesses for Expansion managers into the subsystem.

## 5.14 NPU

SSE-310 supports one Ethos-U55 NPU core.

NPU0 supports different static configurations, but the following configuration value must be adhered to:

**Table 5-10: Static configurations of Ethos-U55**

Configuration	Value for NPU0	Description
External DMA interface	0	Specifies whether the NPU Includes an interface for an external DMA: <ul style="list-style-type: none"> <li>Interface not included</li> <li>interface included</li> </ul>

The Ethos-U55 NPU does not support HW debug control. Arm recommends that the debug software triggers a routine that controls the NPU when using HW flow control of the CPU and rest of system for debugging.

The NPU can access the system memory map as follows:

- M0 Interface : All of the system memory, except for the CPU0 S-AHB Instruction TCM and CPU0 S-AHB Data TCM areas
- M1 Interface : All of the system memory, except for the CPU0 S-AHB Instruction TCM and CPU0 S-AHB Data TCM areas

This memory mapping is enforced using Manager Security Controllers (MSC), IDAUs, and Memory Protection Controllers (MPCs) to separate the Trusted and non-Trusted world for the memories and peripheral protection controllers (PPC) for Privilege and Unprivilege separation and trusted and Non-trusted world separation of peripherals. There is no protection provided for Privilege or Unprivilege memory access.

Therefore, Arm recommends that the NPU should be made available to Unprivilege software, only if the risk of Unprivilege software accessing Privilege memory is acceptable.

In SSE-310, the NPU is not allowed to fetch the Command stream from the Non-secure memory alias when the NPU security level is configured as Secure (For more details, see [NPUSPPORSL](#)). The ARID signal of NPU AXI interfaces identifies the Command stream of NPU. Secure Command

stream transactions to Non-secure memory alias are blocked and responded according to the SECRESPCFG register settings (For more details, see [SECRESPCFG](#) ).

For full details on Ethos-U55, see *AMBA® Ethos™-U55 NPU Technical Reference Manual*.

### 5.14.1 NPU security mapping

The NPU can operate in different security levels. To change from operating in one security level to another, a reset is required.

When the NPU reset is released by the PPU controlling the NPU power domain PD\_NPU0, the NPU reset samples the signals driven by:

- NPUSPPORSL.SP\_NPU0PORSL to determine its default security level.
- NPUSPPORPL.SP\_NPU0PORPL or NPUNSPORPL.NS\_NPU0PORPL to determine its default privilege level.

The NPUSPPORSL and NPUSPPORPL registers are also used to control the PPC, protecting the NPU. For more detail, see sections [NPUNSPORPL](#), and [NPUSPPORPL](#).



The PPC protection setting takes effect as soon as the security and privilege registers are updated.

---

The following sequences are recommended when transitioning the NPU to a new security or privilege level.

#### Changing security level from Secure (S) to Non-secure (NS)

The following sequence details the steps for the Secure privilege software to change security level:

1. Read the current Security level from the corresponding system register:  
initial\_security= NPUSPPORSL.SP\_NPU0PORSL.
2. If initial\_security=S then software reads the privilege to be retained in the target Security level from the corresponding system register: initial\_privilege=NPUNSPORPL.NS\_NPU0PORPL else skip remaining sequence.
3. Write on the current security alias of the NPU (S) the CMD.power\_q\_enable register field of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register.
4. Write on the current security alias of the NPU (S) the RESET.pending\_CSL register field of the NPU to the new Security level (1= Non-secure) while preserving RESET.pending\_CPL=initial\_privilege.
5. Read on the current security alias of the NPU (S) the STATUS register of the NPU until the field reset\_status no longer yields the value 1.
6. Write the new security level into NPUSPPORSL.SP\_NPU0PORSL register field (1= Non-secure).

7. Write on the current security alias of the NPU (NS) the CMD.power\_q\_enable register field of the NPU to allow the NPU to speculatively power OFF (1 = Power OFF allowed ) while preserving the rest of the CMD register.

### Changing Security level from Non-secure (NS) to Secure (S)

The following sequence details the steps for the Secure privilege software to change security level to Secure:

1. Read the current Security level from the corresponding system register:  
initial\_security= NPUSPPORSL.SP\_NPUOPORSL
2. If initial\_security=NS then software reads the privilege to be retained in the target Security level from the corresponding system register: initial\_privilege=NPUSPPORPL.SP\_NPUOPORPL else skip remaining sequence.
3. Write on the current security alias of the NPU (NS) the CMD.power\_q\_enable register field of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register
4. Write the new security level into NPUSPPORSL.SP\_NPUOPORSL register field (0 = Secure)
5. Write on the current security alias of the NPU (S) the RESET.pending\_CSL register field of the NPU to the new Security level (0=Secure) while preserving RESET.pending\_CPL=initial\_privilege
6. Read on the current security alias of the NPU (S) the STATUS register of the NPU until the field reset\_status no longer yields the value 1
7. Write on the current security alias of the NPU (S) the CMD.power\_q\_enable register field of the NPU to allow the NPU to speculatively power OFF (1 = Power OFF allowed ) while preserving the rest of the CMD register.

### Changing privilege level

Secure or Non-secure privilege software can change the privilege state of the NPU, as long as the privilege state being moved to is equal or lower than the software enacting the change and the security level of the software is at least matching the current security level (initial\_security) of the NPU.

The following sequence details the steps for the privilege software to change the privilege level:

1. Read the current privilege level from the corresponding system register:

Current Security level = Non-secure

Initial\_privilege=NPUNSPORPL.NS\_NPUOPORPL

Current Security level = Secure

Initial\_privilege=NPUSPPORPL.SP\_NPUOPORPL

If initial\_privilege is the desired level, then skip remaining sequence.

1. Write on the current security alias of the NPU (initial\_security) the CMD.power\_q\_enable register field of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register.

2. Write to the current security alias of the NPU (initial\_security) the RESET.pending\_CPL register field of the NPU to the new privilege level (0=User; 1=Privileged) while preserving RESET.pending\_CSL=initial\_security.
3. Read from the current security alias of the NPU (initial\_security) the STATUS register of the NPU until the field reset\_status no longer yields the value 1.
4. Write to the alias corresponding to the current security level of the NPU (initial\_security) the new privilege level into:

Current security level = Non-secure

NPUNSPORPL.NS\_NPU0PORPL

Current security level = Secure

NPUSPPORPL.SP\_NPU0PORPL

5. Write on the current security alias of the NPU (initial\_security) the CMD.power\_q\_enable register field of the NPU to the allow the NPU to speculatively power OFF (1 = Power OFF allowed) while preserving the rest of the CMD register.



## 6 Programmer Model

The following sections describe the programmer model of SSE-310.

### 6.1 System Memory Map Overview

[High level system address map](#) shows the high-level view of the memory map defined by SSE-310. This memory map is divided into Secure and Non-secure regions. The memory alternates between Secure and Non-secure regions on 256Mbyte regions, with only a few address areas exempted from security mapping because they are related to debug functionality.

To provide memory blocks and peripherals that can be mapped either as Secure or Non-secure using software, several address regions are aliased as shown in [High level system address map](#). Software can then choose to allocate each memory block or peripheral as Secure or Non-secure using protection controllers. The *Implementation Defined Attribution Unit* (IDAU) Region values specify the Security of each area together with its ID and each region's *Non-secure Callable* (NSC) settings.

Except when specifically stated, the following occur:

- All access to unmapped regions of the memory result in bus-error response.
- When accessing unmapped address space within a mapped region taken by a peripheral, the access results in *Read-As-Zero and Write-Ignored* (**RAZWI**) except when specifically stated otherwise.
- Any accesses that result in security violations are either **RAZWI** or return a bus error response as defined by the SECRESPCFG register setting.

Some regions of memory map are reserved to maintain compatibility with past and future subsystems. Other areas are mapped to Expansion interfaces.

All accesses targeting populated Volatile Memory regions within 0x21000000 to 0x21FFFFFF and 0x31000000 to 0x31FFFFFF support exclusive access since they implement exclusive access monitoring, provided the accesses are from:

- CPU0
- Expansion managers through the Subordinate Main Expansion Interfaces

Exclusive access is not supported for other regions implemented within the subsystem. For regions that reside in user expansion areas, exclusive access support is defined by the user expansion logic.

If an exclusive access tries to access a region that does not support exclusive accesses, these accesses are not monitored for exclusive access, and might still update their target memory locations regardless of their associated exclusive responses.

## 6.1.1 High level system address map

Security values do not define Privileged or Unprivileged accessibility. These are defined by the PPC, or by the register blocks that is mapped to each area. See lower-level details of each area for details. The System Address Map defines the following values for accessibility:

**S**

Secure Access

**NS**

Non-secure Access

The NSC values are defined through registers in the Secure Access Configuration registers.

**Table 6-1: High-Level System Address Map**

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
1	0x00000000 - 0x00FFFFFF	16MB	ITCM	5	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	CPU Instruction TCM See <a href="#">CPU TCM memories</a>
2	0x01000000 - 0x09FFFFFF	192MB	Code Expansion	6	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Manager Code Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
2.1	0x0A000000 - 0x0AFFFFFF	16MB	CPU0 ITCM	6.1	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	CPU0 S-AHB Instruction TCM Access
2.2	0x0B000000 - 0x0BFFFFFF	16MB	Reserved	6.2	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Reserved
2.3	0x0C000000 - 0x0CFFFFFF	16MB	Reserved	6.3	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Reserved
2.4	0x0D000000 - 0x0DFFFFFF	16MB	Reserved	6.4	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Reserved
3	0x0E000000 - 0x0E001FFF	8KB	Reserved	7	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Reserved

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
4	0x0E002000 - 0x0FFFFFFF	-	Reserved	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 0</li> <li>NSC: 0</li> </ul>	Reserved
5	0x10000000 - 0x10FFFFFF	16MB	ITCM	1	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	CPU Instruction TCM See <a href="#">CPU TCM memories</a>
6	0x11000000 - 0x19FFFFFF	192MB	Code Expansion	2	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Manager Code Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
6.1	0x1A000000 - 0x1AFFFFFF	16MB	CPU0 ITCM	2.1	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	CPU0 S-AHB Instruction TCM Access
6.2	0x1B000000 - 0x1BFFFFFF	16MB	Reserved	2.2	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Reserved
6.3	0x1C000000 - 0x1CFFFFFF	16MB	Reserved	2.3	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Reserved
6.4	0x1D000000 - 0x1DFFFFFF	16MB	Reserved	2.4	<ul style="list-style-type: none"> <li>security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Reserved
7	0x1E000000 - 0x1E001FFF	8KB	Reserved	3	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Reserved
8	0x1E002000 - 0x1FFFFFFF	-	Reserved	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 1</li> <li>NSC: CODENSC</li> </ul>	Reserved
9	0x20000000 - 0x20FFFFFF	16MB	DTCM	13	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	CPU Data TCM See <a href="#">CPU TCM memories</a>

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
10	0x21000000 - 0x21FFFFFF	16MB	Volatile Memory	14	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Volatile Memory See <a href="#">Volatile Memory Region</a>
11	0x22000000 - 0x23FFFFFF	32MB	Reserved	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Reserved
11.1	0x24000000 - 0x24FFFFFF	16MB	CPU0 DTCM	15.1	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	CPU0 S-AHB Data TCM Access
11.2	0x25000000 - 0x25FFFFFF	16MB	Reserved	15.2	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Reserved
11.3	0x26000000 - 0x26FFFFFF	16MB	Reserved	15.3	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Reserved
11.4	0x27000000 - 0x27FFFFFF	16MB	Reserved	15.4	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Reserved
12	0x28000000 - 0x2FFFFFFF	128MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 2</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
13	0x30000000 - 0x30FFFFFF	16MB	DTCM	9	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	CPU Data TCM See <a href="#">CPU TCM memories</a>
14	0x31000000 - 0x31FFFFFF	16MB	Volatile memory	10	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Internal Multi-bank Volatile Memory See <a href="#">Volatile Memory Region</a>
15	0x32000000 - 0x33FFFFFF	32MB	Reserved	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Reserved

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
15.1	0x34000000 - 0x34FFFFFF	16MB	CPU0 DTCM	11.1	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	CPU0 S-AHB Data TCM Access
15.2	0x35000000 - 0x35FFFFFF	16MB	Reserved	11.2	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Reserved
15.3	0x36000000 - 0x36FFFFFF	16MB	Reserved	11.3	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Reserved
15.4	0x37000000 - 0x37FFFFFF	16MB	Reserved	11.4	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Reserved
16	0x38000000 - 0x3FFFFFFF	128MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 3</li> <li>NSC: RAMNSC</li> </ul>	Manager Main Expansion Interface  See <a href="#">Main Interconnect Expansion Interfaces</a>
17	0x40000000 - 0x4000FFFF	64KB	Peripherals	27	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Peripheral Region  See <a href="#">Peripheral Region</a>
18	0x40010000 - 0x4001FFFF	64KB	Private CPU	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	CPU Private Peripheral Region  See <a href="#">Processor Private Region</a>
19	0x40020000 - 0x4003FFFF	128KB	System Control Peripheral Region.	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	System Control Peripheral Region  See <a href="#">System Control Peripheral Region</a>
20	0x40040000 - 0x400FFFFF	768KB	Peripherals	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Peripheral Region  See <a href="#">Peripheral Region</a>
21	0x40100000 - 0x47FFFFFF	127MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface  See <a href="#">Main Interconnect Expansion Interfaces</a>

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
22	0x48000000 - 0x4800FFFF	64KB	Peripherals	32	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>
23	0x48010000 - 0x4801FFFF	64KB	Private CPU	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	CPU Private Peripheral Region See <a href="#">Processor Private Region</a>
24	0x48020000 - 0x4803FFFF	128KB	System Control	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	System Control Peripheral Region See <a href="#">System Control Peripheral Region</a>
25	0x48040000 - 0x480FFFFF	768KB	Peripherals	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>
26	0x48100000 - 0x4FFFFFFF	127MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 4</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface See <a href="#">Peripheral Interconnect Expansion Interfaces</a> and <a href="#">Peripheral Expansion Region</a>
27	0x50000000 - 0x5000FFFF	64KB	Peripherals	17	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>
28	0x50010000 - 0x5001FFFF	64KB	Private CPU	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	CPU Private Peripheral Region See <a href="#">Processor Private Region</a>
29	0x50020000 - 0x5003FFFF	128KB	System Control	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	System Control Peripheral Region See <a href="#">System Control Peripheral Region</a>
30	0x50040000 - 0x500FFFFF	786KB	Peripherals	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>
31	0x50100000 - 0x57FFFFFF	127MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface See <a href="#">Peripheral Interconnect Expansion Interfaces</a>
32	0x58000000 - 0x5800FFFF	64KB	Peripherals	22	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
33	0x58010000 - 0x5801FFFF	64KB	Private CPU	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Processor Private Peripheral Region See <a href="#">Processor Private Region</a>
34	0x58020000 - 0x5803FFFF	128KB	System Control	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	System Control Peripheral Region See <a href="#">System Control Peripheral Region</a>
35	0x58040000 - 0x580FFFFF	768KB	Peripherals	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Peripheral Region See <a href="#">Peripheral Region</a>
36	0x58100000 - 0x5FFFFFFF	127MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 5</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface See <a href="#">Peripheral Interconnect Expansion Interfaces</a> and <a href="#">Peripheral Expansion Region</a>
37	0x60000000 - 0x6FFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 6</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
38	0x70000000 - 0x7FFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 7</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
39	0x80000000 - 0x8FFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: 8</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
40	0x90000000 - 0x9FFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: 9</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
41	0xA0000000 - 0xAFFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: A</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a> .
42	0xB0000000 - 0xBFFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: B</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>
43	0xC0000000 - 0xCFFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: C</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface See <a href="#">Main Interconnect Expansion Interfaces</a>

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
44	0xD0000000 - 0xDFFFFFFF	256MB	Main Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: D</li> <li>NSC: 0</li> </ul>	Manager Main Expansion Interface  See <a href="#">Main Interconnect Expansion Interfaces</a>
45	0xE0000000 - 0xE00FFFFF	1MB	PPB	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: E</li> <li>NSC: 0</li> </ul>	CPU Private Peripheral Bus Region. Local to the CPU  See <a href="#">CPU Private Peripheral Bus Region</a>
46	0xE0100000 - 0xE01FFFFF	1MB	Debug System	49	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: E</li> <li>NSC: 0</li> </ul>	Debug System Access Region  See <a href="#">Debug System Access Region</a>
47	0xE0200000 - 0xEFFFFFFF	254MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: NS</li> <li>IDAUID: E</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface  See <a href="#">Peripheral Interconnect Expansion Interfaces</a>
48	0xF0000000 - 0xF00FFFFF	1MB	Reserved	-	Exempt	Reserved
49	0xF0100000 - 0xF01FFFFF	1MB	Debug System	46	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: F</li> <li>NSC: 0</li> </ul>	Debug System Access Region  See <a href="#">Debug System Access Region</a>
50	0xF0200000 - 0xFFFFFFFF	254MB	Peripheral Expansion	-	<ul style="list-style-type: none"> <li>Security: S</li> <li>IDAUID: F</li> <li>NSC: 0</li> </ul>	Manager Peripheral Expansion Interface  See <a href="#">Peripheral Interconnect Expansion Interfaces</a>

## 6.2 CPU TCM memories

The CPU0 in SSE-310 is configured to implement *Tightly Coupled Memories* (TCM) for Instruction and Data. These memories reside in the following location from the perspective of the CPU0 core:

- 0x00000000 to 0x00FFFFFF and 0x10000000 to 0x10FFFFFF for Instruction TCM. Both regions are aliased, and each provides up to 16MB of TCM space.
- 0x20000000 to 0x20FFFFFF and 0x30000000 to 0x30FFFFFF for Data TCM. Both regions are aliased, and each provides up to 16MB of TCM space.

CPU0 has access only to its own local TCM via its private address, and does not have access to its own TCMs via the Main Interconnect. Other Managers on the Main interconnect have access to the TCMs. The TCM Subordinate Interface allows expansion managers to access the TCMs.

For more detail, see [TCM Subordinate interface](#).

All TCMs start at the base address of their respective regions. Unused memory areas in those regions are reserved, and they return a bus error response when accessed.



## 6.3 Volatile Memory Region

SSE-310 supports two internal *Volatile Memory* (VM) Banks. These are implemented as SRAMs.

All VM banks in the system are of the same size. They form a contiguous memory area up to 16MB. This memory area is aliased onto both the Secure and Non-secure memory regions. A memory protection controller per VM divides the VM into pages and determines where each page resides in either the Secure or Non-secure regions. Any unused areas within that 16MB region are reserved.

The following table shows an example, where two memory banks are configured with 2 MB in size.

**Table 6-2: Example volatile memory region address map**

Row ID	Address - from	Address - to	Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
1	0x21000000	0x211FFFFF	2MB	VM0	Maps to Internal Volatile Memory Bank 0	5	NS-MPC
2	0x21200000	0x213FFFFF	2MB	VM1	Maps to Internal Volatile Memory Bank 1	6	NS-MPC
3	0x21400000	0x21FFFFFF	12MB	Reserved	Reserved	-	-
4	0x31000000	0x311FFFFF	2MB	VM0	Maps to Internal Volatile Memory Bank 0	1	S-MPC
5	0x31200000	0x313FFFFF	2MB	VM1	Maps to Internal Volatile Memory Bank 1	2	S-MPC
6	0x31400000	0x31FFFFFF	12MB	Reserved	Reserved	-	-

<sup>1</sup> Legend

- NS-MPC: Non-secure access only, gated by a MPC.
- S-MPC: Secure access only, gated by a MPC.

The VMMPCLKSIZE configuration parameter sets the block-size of the MPC that configures the granularity for the software to define Secure and Non-Secure regions in the Volatile Memory.

## 6.4 Peripheral Region

The Peripheral Regions are memory regions where peripherals of the system reside. There are eight regions in total as follows:

### **0x40000000 to 0x400FFFFF**

Non-secure region for low-latency peripherals that are expected to be aliased in its associated Secure region, 0x50000000 to 0x500FFFFF.

### **0x40040000 to 0x400FFFFF**

Non-secure region for low-latency peripherals that are expected to be not aliased.

### **0x48000000 to 0x480FFFFF**

Non-secure region for high-latency peripherals that are expected to be aliased in its associated Secure region, 0x58000000 to 0x580FFFFF.

#### 0x48040000 to 0x480FFFFF

Non-secure region for high-latency peripherals that are expected to be not aliased.

#### 0x50000000 to 0x5000FFFF

Secure region for low-latency peripherals that are expected to be aliased in its associated Non-secure region, 0x40000000 to 0x4000FFFF.

#### 0x50040000 to 0x500FFFFF

Secure region for low-latency peripherals that are expected to be not aliased.

#### 0x58000000 to 0x5800FFFF

Secure region for high-latency peripherals that are expected to be aliased in its associated Non-secure region, 0x48000000 to 0x4800FFFF.

#### 0x58040000 to 0x580FFFFF

Secure region for high-latency peripherals that are expected to be not aliased.

For regions that are aliased to both Secure and Non-secure region, the final mapping of a peripheral in these regions to either Secure or Non-secure region is determined by Peripheral Protection Controller (PPC) that are programmed using Secure Access Configuration registers. See [Secure access configuration register block](#).



Peripherals implemented in these regions support 32-bit R/W accesses. Any Byte and Half word access results in UNPREDICTABLE behavior, unless otherwise stated.

The following table shows the memory map of the Peripheral Regions.

**Table 6-3: Peripheral Region Address Map**

Row ID	Address		Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
	From	To					
1	0x40000000	0x40000FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	23	NS_PPC
2	0x40001000	0x40001FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	24	
3	0x40002000	0x40003FFF	-	Reserved	Reserved. ( <b>RAZWI</b> )	-	
4	0x40004000	0x40004FFF	4KB	NPU0	NPU0 Configuration interface  <b>Note:</b> if NPUNUM >0 and Reserved if NPUNUM =0	26	NS_PPC
5	0x40005000	0x40005FFF	-	Reserved	Reserved	-	-
6	0x40006000	0x40006FFF	-	Reserved	Reserved	-	-
7	0x40007000	0x40007FFF	-	Reserved	Reserved	-	-
8	0x40008000	0x40008FFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
9	0x40040000	0x4007FFFF	-	Reserved	Reserved	-	-
10	0x40080000	0x4008FFFF	4KB	NSACFG	Non-secure Access Configuration Register Block. See <a href="#">Non-secure Access Configuration Register Block</a> .	-	NS
11	0x40081000	0x4008FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-

Row ID	Address		Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
	From	To					
12	0x40090000	0x40093FFF	16KB	Reserved	Reserved ( <b>RAZWI</b> )	-	NS
13	0x40094000	0x400FFFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
14	0x48000000	0x48000FFF	4KB	TIMER0	Timer 0. See <a href="#">Timestamp Timers</a> .	41	NS_PPC
15	0x48001000	0x48001FFF	4KB	TIMER1	Timer 1. See <a href="#">Timestamp Timers</a> .	42	
16	0x48002000	0x48002FFF	4KB	TIMER2	Timer 2. See <a href="#">Timestamp Timers</a> .	43	
17	0x48003000	0x48003FFF	4KB	TIMER3	Timer 3. See <a href="#">Timestamp Timers</a> .	44	
18	0x48004000	0x4800FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
19	0x48040000	0x48040FFF	4KB	NSWDCTRL	Non-secure Watchdog Control Frame. See <a href="#">Timestamp Watchdogs</a> .	-	NS
20	0x48041000	0x48041FFF	4KB	NSWDREF	Non-secure Watchdog Refresh Frame. See <a href="#">Timestamp Watchdogs</a> .	-	
21	0x48042000	0x4804FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
22	0x48050000	0x480FFFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
23	0x50000000	0x50000FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	1	S_PPC
24	0x50001000	0x50001FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	2	
25	0x50002000	0x50003FFF	8KB	Reserved	Reserved ( <b>RAZWI</b> )	3	-
26	0x50004000	0x50004FFF	4KB	NPU0	NPU0 Configuration interface  <b>Note:</b> if NPUNUM >0 and Reserved if NPUNUM =0	4	S_PPC
27	0x50005000	0x50005FFF	4KB	Reserved	Reserved	-	
28	0x50006000	0x50006FFF	4KB	Reserved	Reserved	-	-
29	0x50007000	0x50007FFF	4KB	Reserved	Reserved	-	-
30	0x50008000	0x5000FFFF	-	Reserved	Reserved	-	-
31	0x50040000	0x5007FFFF	-	Reserved	Reserved	-	-
32	0x50080000	0x50080FFF	4KB	SACFG	Secure Access Configuration Register Block. See <a href="#">Secure Access Configuration Register Block</a> .	-	S
33	0x50081000	0x50082FFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
34	0x50083000	0x50083FFF	4KB	VM0MPC	VM0 Memory Protection Controller. See <a href="#">Volatile Memory</a> .	-	S
35	0x50084000	0x50084FFF	4KB	VM1MPC	VM1 Memory Protection Controller. See <a href="#">Volatile Memory</a> .	-	
36	0x50085000	0x50085FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	-	
37	0x50086000	0x50086FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	-	
38	0x50087000	0x5008FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
39	0x50090000	0x50093FFF	16KB	Reserved	Reserved ( <b>RAZWI</b> )	-	-
40	0x50094000	0x500FFFFF	-	Reserved	Reserved	-	-
41	0x58000000	0x58000FFF	4KB	TIMER0	Timer 0. See <a href="#">Timestamp Timers</a> .	14	S_PPC
42	0x58001000	0x58001FFF	4KB	TIMER1	Timer 1. See <a href="#">Timestamp Timers</a> .	15	
43	0x58002000	0x58002FFF	4KB	TIMER2	Timer 2. See <a href="#">Timestamp Timers</a> .	16	
44	0x58003000	0x58003FFF	4KB	TIMER3	Timer 3. See <a href="#">Timestamp Timers</a> .	17	
45	0x58004000	0x5800FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-

Row ID	Address		Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
	From	To					
46	0x58040000	0x58040FFF	4KB	SWDCTRL	Secure Watchdog Control Frame. See <a href="#">Timestamp Watchdogs</a> .	-	S
47	0x58041000	0x58041FFF	4KB	SWDREF	Secure Watchdog Refresh Frame. See <a href="#">Timestamp Watchdogs</a> .	-	
48	0x58042000	0x5804FFFF	-	Reserved	Reserved ( <b>RAZWI</b> )	-	-
49	0x58050000	0x580FFFFF	-	Reserved	Reserved	-	-

<sup>1</sup> Legend

- NS\_PPC: Non-secure access only, gated by a PPC.
- S\_PPC: Secure access only, gated by a PPC.
- S: Secure access only.
- NS: Non-secure access only.

## 6.4.1 Secure Access Configuration Register Block

The Secure Access Configuration Register Block implements program-visible states that allow software to control security gating units within the design. This register block base address is 0x50080000. This register block is Secure Privileged access only and supports 32 bit R/W accesses.

The following table list the registers within this block. For write access to these registers, only 32-bit writes are supported. Any Byte and Half word writes are ignored.

All registers reside in the PD\_SYS power domain and is reset by **nWARMRESETSYS**.

Details of each register in the table below are described in separate sections.

**Table 6-4: Secure Access Configuration Register Block Register Map**

Offset	Name	Access	Reset value	Description
0x000	SPCSECCTRL	Read-write	0x00000000	Secure Privilege Controller Secure Configuration Control register
0x004	BUSWAIT	Read-write	Configurable	Bus Access Wait control after reset
0x008	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x010	SECRESPCFG	Read-write	0x00000000	Security Violation Response Configuration Register
0x014	NSCCFG	Read-write	0x00000000	Non-secure Callable Configuration for IDAU
0x018	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x01C	SECMPICINTSTAT	Read-only	0x00000000	Secure MPC Interrupt Status

Offset	Name	Access	Reset value	Description
0x020	SECPPCINTSTAT	Read-only	0x00000000	Secure PPC Interrupt Status
0x024	SECPPCINTCLR	Read-write	0x00000000	Secure PPC Interrupt Clear
0x028	SECPPCINTEN	Read-write	0x00000000	Secure PPC Interrupt Enable
0x02C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x030	SECMSCINTSTAT	Read-only	0x00000000	Secure MSC Interrupt Status
0x034	SECMSCINTCLR	Read-write	0x00000000	Secure MSC Interrupt Clear
0x038	SECMSCINTEN	Read-write	0x00000000	Secure MSC Interrupt Enable
0x03C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x040	BRGINTSTAT	Read-only	0x00000000	Bridge Buffer Error Interrupt Status
0x044	BRGINTCLR	Read-write	0x00000000	Bridge Buffer Error Interrupt Clear
0x048	BRGINTEN	Read-write	0x00000000	Bridge Buffer Error Interrupt Enable
0x04C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x050	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x054 - 0x05C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x060	MAINNSPPCEXP0	Read-write	0x00000000	Expansion 0 Non-secure Access Peripheral Protection Control on the Main Interconnect
0x064	MAINNSPPCEXP1	Read-write	0x00000000	Expansion 1 Non-secure Access Peripheral Protection Control on the Main Interconnect
0x068	MAINNSPPCEXP2	Read-write	0x00000000	Expansion 2 Non-secure Access Peripheral Protection Control on the Main Interconnect
0x06C	MAINNSPPCEXP3	Read-write	0x00000000	Expansion 3 Non-secure Access Peripheral Protection Control on the Main Interconnect
0x070	PERIPHNSPPC0	Read-write	0x00000000	Non-secure Access Peripheral Protection Control 0 on Peripheral Interconnect. Each bit field defines the Non-secure access settings for an associated peripheral: <ul style="list-style-type: none"> <li>1 - Allow Non-secure access</li> <li>0 - Disallow Non-secure access</li> </ul>
0x074	PERIPHNSPPC1	Read-write	0x00000000	Non-secure Access Peripheral Protection Control 1 on Peripheral Interconnect
0x078	NPUSPPORSL	Read-write	Configurable	Secure Access NPU security level reset state control: Each Field Controls the PORSL inputs for an associated NPU.
0x07C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x080	PERIPHNSPPCEXP0	Read-write	0x00000000	Expansion 0 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x084	PERIPHNSPPCEXP1	Read-write	0x00000000	Expansion 1 Non-secure Access Peripheral Protection Control on Peripheral Bus

Offset	Name	Access	Reset value	Description
0x088	PERIPHNSPPCEXP2	Read-write	0x00000000	Expansion 2 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x08C	PERIPHNSPPCEXP3	Read-write	0x00000000	Expansion 3 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x090	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x094 - 0x09C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x0A0	MAINSPPPCEXP0	Read-write	0x00000000	Expansion 0 Secure Unprivileged Access Peripheral Protection Control on Main Interconnect
0x0A4	MAINSPPPCEXP1	Read-write	0x00000000	Expansion 1 Secure Unprivileged Access Peripheral Protection Control on Main Interconnect
0x0A8	MAINSPPPCEXP2	Read-write	0x00000000	Expansion 2 Secure Unprivileged Access Peripheral Protection Control on Main Interconnect
0x0AC	MAINSPPPCEXP3	Read-write	0x00000000	Expansion 3 Secure Unprivileged Access Peripheral Protection Control on Main Interconnect
0x0B0	PERIPHSPPPCO	Read-write	0x00000000	Secure Unprivileged Access Peripheral Protection Control 0 on Peripheral Interconnect
0x0B4	PERIPHSPPPC1	Read-write	0x00000000	Secure Unprivileged Access Peripheral Protection Control 1 on Peripheral Interconnect
0x0B8	NPUSPPORPL	Read-write	Configurable	Secure Access NPU privilege level reset state control
0x0BC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x0C0	PERIPHSPPPCEXP0	Read-write	0x00000000	Expansion 0 Secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect
0x0C4	PERIPHSPPPCEXP1	Read-write	0x00000000	Expansion 1 Secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect
0x0C8	PERIPHSPPPCEXP2	Read-write	0x00000000	Expansion 2 Secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect
0x0CC	PERIPHSPPPCEXP3	Read-write	0x00000000	Expansion 3 Secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect
0x0D0	NSMSCEXP	Read-write	Configurable	Expansion MSC Non-secure Configuration
0x0D4 - 0xFFC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000052	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000003B	Peripheral ID 2

Offset	Name	Access	Reset value	Description
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

### 6.4.1.1 SPCSECCTRL

The Security Privilege Controller Security Configuration Control Register implements the security lock register.

**Table 6-5: SPCSECCTRL Register**

Bits	Type	Default	Name	Description
31:1	RAZ/WI	0x00000000	-	Reserved.
0	RW1S	0x00	SPCSECCFGLOCK	<p>Active High Control to Disable writes to Security related control registers in the Secure Access Configuration Register Block. Once set to high, it can no longer be cleared to zero except through reset or the PD_SYS turning OFF. Registers that can no longer be modified when SPCSECCFGLOCK is set to HIGH are:</p> <ul style="list-style-type: none"> <li>• NSCCFG</li> <li>• MAINNSPPCEXP&lt;N&gt;</li> <li>• PERIPHNSPPCO</li> <li>• PERIPHNSPPC1</li> <li>• PERIPHNSPPCEXP&lt;N&gt;</li> <li>• MAINSPPPCEXP&lt;N&gt;</li> <li>• PERIPHSPPPCO</li> <li>• PERIPHSPPPC1</li> <li>• PERIPHSPPPCEXP&lt;N&gt;</li> <li>• NSMSCEXP</li> <li>• NPUSPPORSL</li> <li>• NPUSPPORPL</li> </ul>

### 6.4.1.2 BUSWAIT

The Bus Access Wait register allows software to gate access entering the interconnect from specific managers in the system, causing them to stall so that the processor can complete the

configuration of the MPCs or other Security registers in the system prior to the stalled accesses commencing.

**Table 6-6: BUSWAIT Register**

Bits	Type	Default	Name	Description
31:17	RAZWI	0x0000	-	Reserved
16	RO	0x00	ACC_WAITN_STATUS	<p>This status register indicates the status of any gating units that are used to block bus access to the system:</p> <p>1 - Allow access</p> <p>0 - Block access</p> <p>This register reflects the combined status of all gating units in the system, including status on the input signal <b>ACCWAITNSTATUS</b>, expected to be driven from external gating units.</p> <p>Both ACC_WAITN_STATUS and ACC_WAITN together, ensuring that software can determine that all gates have reached the state that is requested.</p>
15:1	RAZWI	0x0000	-	Reserved
0	RW	ACCWAITNRST	ACC_WAITN	<p>Request gating units to block bus access to system:</p> <p>1 - Allow access</p> <p>0 - Block access</p> <p>This control only affects the Access Control Gates (ACG) in the system that feeds into the interconnect, and it excludes access from CPU cores. This register also drives the output signal <b>ACCWAITn</b>.</p> <p>ACC_WAITN_STATUS and ACC_WAITN together, ensure that software can determine that all gates have reached the state that is requested.</p>

### 6.4.1.3 SECRESPCFG

The Security Violation Response Configuration register defines the response to an access that causes security violation on the bus fabric.

**Table 6-7: SECRESPCFG Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x00000000	-	Reserved
0	RW	0x00	SECRESPCFG	<p>This field configures the response in case of a security violation:</p> <ul style="list-style-type: none"> <li>0 - Read-Zero Write Ignore</li> <li>1 - Bus error</li> </ul> <p>Note that some components, for example, the AHB Memory Protection Controllers (MPC), provide their own control registers to configure their own response.</p>



### 6.4.1.4 NSCCFG

The Non-secure Callable Configuration register allows software to define if the region 0x10000000 to 0x1FFFFFFF that normally host Secure code, and the region 0x30000000 to 0x3FFFFFFF that normally implements Secure VMs, are Non-secure Callable regions of memory.

**Table 6-8: NSCCFG Register**

Bits	Type	Default	Name	Description
31:2	RAZWI	0x00000000	-	Reserved.
1	RW	0x00	RAMNSC	Configures if the region 0x30000000 to 0x3FFFFFFF is Non-secure Callable: <ul style="list-style-type: none"> <li>0 - Not Non-secure Callable</li> <li>1 - Non-secure Callable</li> </ul>
0	RW	0x00	CODNSC	Configures if the CODE region 0x10000000 to 0x1FFFFFFF is Non-secure Callable: <ul style="list-style-type: none"> <li>0 - Not Non-secure Callable</li> <li>1 - Non-secure Callable</li> </ul>

### 6.4.1.5 SECMPICINTSTAT

The interrupt signals from all Memory Protection Controllers (MPC), both within SSE-310 and in the expansion logic are merged and sent to CPU0 on a single interrupt signal. The Secure MPC Interrupt status register therefore provides Secure software with the ability to check which one of the MPC is causing the interrupt. Once the source of the interrupt is identified, you must use the MPC register interface to clear the interrupt.

**Table 6-9: SECMPICINTSTAT register**

Bits	Type	Default	Name	Description
31:16	RO	0x00000000	SMPCEXP_STATUS	Interrupt Status for Expansion Memory Protection Controller. Each bit <i>n</i> (0 to 15) local in this field shows the status of input signal <b>SMPCEXPSTATUS[n]</b> . The MPCEXPDIS configuration option defines if each bit within this register is actually implement such that if MPCEXPDIS[i] = 1'b1 then SMPCEXP_STATUS[i] is disabled and always reads as zeros.
15:4	RAZWI	0x00000000	-	Reserved
3	RAZWI	0x00	-	Reserved
2	RAZWI	0x00	-	Reserved
1	RO	0x00	SMPCVM1_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 1.
0	RO	0x00	SMPCVM0_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 0.

### 6.4.1.6 SECPPICINTSTAT, SECPPICINTCLR and SECPPICINTEN

The PPC Secure PPC Interrupt status, clear and enable registers allow software to determine source of the interrupt, clear the interrupt, and enable or disable (Mask) the interrupt.

When access violations occur on any Peripheral Protection Controller (PPC), a level interrupt is raised through a combined interrupt that is then sent to the CPU0.

**Table 6-10: SECPPCINTSTAT Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x000	-	Reserved
23:20	RO	0x00	SMAINPPCEXP_STATUS	Interrupt status of Expansion Peripheral Protection Controller on the Main Interconnect. Each bit <i>n</i> (0 to 3) local in this field captures the state of the input signal <b>SMAINPPCEXPSTATUS[n]</b> .
19:17	RAZWI	0x00	-	Reserved
16	RAZWI	0x00	-	Reserved
15:8	RAZWI	0x000	-	Reserved
7:4	RO	0x00	SPERIPHPPCEXP_STATUS	Interrupt status of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Each bit <i>n</i> (0 to 3) local in this field captures the state of the input signal <b>SPERIPHPPCEXPSTATUS[n]</b> .
3:2	RAZWI	0x00	-	Reserved
1	RO	0x00	SPERIPHPPC1_STATUS	Interrupt status of Peripheral Protection Controller Group 1 on the Peripheral Interconnect within the System
0	RO	0x00	SPERIPHPPC0_STATUS	Interrupt status of Peripheral Protection Controller Group 0 on the Peripheral Interconnect within the System

**Table 6-11: SECPPCINTCLR Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x000	-	Reserved
23:20	RAZW1C	0x00	SMAINPPCEXP_CLR	Interrupt clear of Expansion Peripheral Protection Controller on the Main Interconnect. Each bit <i>n</i> , when set to HIGH clears the interrupt status of the PPC connected to <b>SMAINPPCEXPSTATUS[n]</b> and <b>SMAINPPCEXPCLR[n]</b> .
19:17	RAZWI	0x00	-	Reserved
16	RAZWI	0x00	-	Reserved
15:8	RAZWI	0x000	-	Reserved
7:4	RAZW1C	0x00	SPERIPHPPCEXP_CLR	Interrupt clear of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Each bit <i>n</i> , when set to HIGH clears the interrupt status of the PPC connected to <b>SPERIPHPPCEXPSTATUS[n]</b> and <b>SPERIPHPPCEXPCLR[n]</b> .
3:2	RAZWI	0x00	-	Reserved
1	RAZW1C	0x00	SPERIPHPPC1_CLR	Interrupt clear of Peripheral Protection Controller 1 on the Peripheral Interconnect within the System
0	RAZW1C	0x00	SPERIPHPPC0_CLR	Interrupt clear of Peripheral Protection Controller 0 on the Peripheral Interconnect within the System

**Table 6-12: SECPPCINTEN Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x000	-	Reserved
23:20	RW	0x00	SMAINPPCEXP_EN	Interrupt enable of Expansion Peripheral Protection Controller on the Main Interconnect. Write 1 to bit <i>n</i> (0 to 3) local in this field to enable interrupt from <b>SMAINPPCEXPSTATUS[n]</b> .
19:17	RAZWI	0x00	-	Reserved
16	RAZWI	0x00	-	Reserved
15:8	RAZWI	0x000	-	Reserved

Bits	Type	Default	Name	Description
7:4	RW	0x00	SPERIPHPCEXP_EN	Interrupt enable of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Write 1 to bit <i>n</i> (0 to 3) local in this field to enable interrupt from <b>SPERIPHPCEXPSTATUS[n]</b> .
3:2	RAZWI	0x00	-	Reserved
1	RW	0x00	SPERIPHPPC1_EN	Interrupt enable of Peripheral Protection Controller Group 1 on the Peripheral Interconnect within the System. Write 1 to enable interrupt from them.
0	RW	0x00	SPERIPHPPC0_EN	Interrupt enable of Peripheral Protection Controller Group 0 on the Peripheral Interconnect within the System. Write 1 to enable interrupt from them.

#### 6.4.1.7 SECMSCINTSTAT, SECMSCINTCLR and SECMSCINTEN

When security violation occurs at any *Manager Security Controller* (MSC) in the Subsystem and in the expansion logic, an interrupt is raised through a combined interrupt to CPU0.

The Secure MSC Interrupt Status Clear register and Enable register allows software to determine source of the interrupt, clear the interrupt, and enable or disable (Mask) the interrupt.

**Table 6-13: SECMSCINTSTAT Register**

Bits	Type	Default	Name	Description
31:16	RO	0x0000	SMSCEXP_STATUS	Interrupt status for Expansion MSC. Each bit <i>n</i> (0 to 15) local in this field captures the active state of the input signal <b>SMSCEXPSTATUS[n]</b> . The configuration option MSCEXPDIS defines if each bit within this register is actually implemented such that if MSCEXPDIS[i] = 1'b1 then SMSCEXP_STATUS[i] is disabled and always reads as zeros.
15:12	RAZWI	0x0	-	Reserved
11	RAZWI	0x0	-	Reserved
10	RAZWI	0x0	-	Reserved
9	RAZWI	0x0	-	Reserved
8	RAZWI	0x0	-	Reserved
7	RAZWI	0x0	-	Reserved
6	RAZWI	0x0	-	Reserved
5	RO	0x0	SMSCNPU0M1_STATUS	Interrupt status for NPU0 M1 MSC Note: <b>RAZWI</b> If NUMNPU < 1
4	RO	0x0	SMSCNPU0M0_STATUS	Interrupt status for NPU0 M0 MSC Note: <b>RAZWI</b> If NUMNPU < 1
3	RAZWI	0x0	SMSCDMAM1_STATUS	Reserved
2	RAZWI	0x0	-	Reserved
1:0	RAZWI	0x0	-	Reserved

**Table 6-14: SECMSCINTCLR Register**

Bits	Type	Default	Name	Description
31:16	RAZW1C	0x0000	SMSCEXP_CLR	Interrupt Clear for Expansion MSC. Each bit 'n', when set to HIGH clears the interrupt status of the MSC connected to <b>SMSCEXPSTATUS[n]</b> and <b>SMSCEXPCLR[n]</b> . The configuration option MSCEXPDIS defines if each bit within this register is actually implemented, such that if MSCEXPDIS[i] = 1'b1 then SMSCEXP_CLR[i] is disabled and any writes to it is ignored.
15:12	RAZWI	0x0	-	Reserved.
11	RAZWI	0x0	-	Reserved.
10	RAZWI	0x0	-	Reserved.
9	RAZWI	0x0	-	Reserved.
8	RAZWI	0x0	-	Reserved.
7	RAZWI	0x0	-	Reserved.
6	RAZWI	0x0	-	Reserved.
5	RAZW1C	0x0	SMSCNPU0M1_CLR	Interrupt clear for NPU0 M1 MSC  <b>RAZWI</b> If NUMNPU < 1
4	RAZWIC	0x0	SMSCNPU0M0_CLR	Interrupt clear for NPU0 M0 MSC  <b>RAZWI</b> If NUMNPU < 1
3	RAZWI	0x0	-	Reserved.
2	RAZWI	0x0	-	Reserved.
1:0	RAZWI	0x0	-	Reserved.

**Table 6-15: SECMSCINTEN Register**

Bits	Type	Default	Name	Description
31:16	RW	0x0000	SMSCEXP_EN	Interrupt Enable for Expansion MSC. Each bit n enables or disables the input interrupt signal <b>SMSCEXPSTATUS[n]</b> . The parameter MSCEXPDIS defines if each bit within this register is actually implement such that if MSCEXPDIS[i] = 1'b1 then SMSCEXP_EN[i] is disabled and any writes to it is ignored.
15:12	RAZWI	0x0	-	Reserved.
11	RAZWI	0x0	-	Reserved.
10	RAZWI	0x0	-	Reserved.
9	RAZWI	0x0	-	Reserved.
8	RAZWI	0x0	-	Reserved.
7	RAZWI	0x0	-	Reserved.
6	RAZWI	0x0	-	Reserved.
5	RW	0x0	SMSCNPU0M1_EN	Interrupt enable for NPU0 M1 MSC.  <b>RAZWI</b> If NUMNPU < 1
4	RW	0x0	SMSCNPU0M0_EN	Interrupt enable for NPU0 M0 MSC.  <b>RAZWI</b> If NUMNPU < 1
3	RAZWI	0x0	-	Reserved.
2	RAZWI	0x0	-	Reserved.

Bits	Type	Default	Name	Description
1:0	RAZWI	0x0	-	Reserved.

#### 6.4.1.8 BRGINTSTAT, BRGINTCLR and BRGINTEN

SSE-310 and its expansion logic can contain bus bridges, which are necessary to handle clock domain crossing. To improve system performance, some of these bridges can buffer write data and complete a write access on their subordinate interfaces before any potential error response is received for the write access on their manager interfaces. When this occurs, these bridges can raise a combined interrupt. The Bridge Buffer Error interrupt status, clear and enable register allow software to determine source of the interrupt, clear the interrupt, and enable or disable (Mask) the interrupt.

**Table 6-16: BRGINTSTAT Register**

Bits	Type	Default	Name	Description
31:16	RO	0x0000	BRGEXP_STATUS	Interrupt status for Expansion Bridge Buffer Error interrupts. Each bit n (0 to 15) local in this field captures the active state of the input signal <b>BRGEXPSTATUS[n]</b> . The configuration option BRGEXPDIS defines if each bit within this register is actually implemented such that if BRGEXPDIS[i] = 1'b1 then BRGEXP_STATUS[i] is disabled and always reads as zeros.
15:0	RAZWI	0x0000	-	Reserved

**Table 6-17: BRGINTCLR Register**

Bits	Type	Default	Name	Description
31:16	RAZW1C	0x0000	BRGEXP_CLR	Interrupt clear of Expansion Bridge Buffer Error interrupts. Each bit n when set to HIGH clears the interrupt status of the bridge connected to <b>BRGEXPSTATUS[n]</b> and <b>BRGEXPCLEAR[n]</b> . The configuration option BRGEXPDIS defines if each bit within this register is actually implement such that if BRGEXPDIS[i] = 1'b1 then BRGEXP_CLR[i] is disabled and any writes to it is ignored.
15:0	RAZWI	0x0000	-	Reserved

**Table 6-18: BRGINTEN Register**

Bits	Type	Default	Name	Description
31:16	RW	0x0000	BRGEXP_EN	Interrupt enable of Expansion Bridge Buffer Error interrupts. Each bit n (0 to 15) local in this field enables the input interrupt of <b>BRGEXPSTATUS[n]</b> . The configuration option BRGEXPDIS defines if each bit within this register is actually implement such that if BRGEXPDIS[i] = 1'b1 then BRGEXP_EN[i] is disabled and any writes to it is ignored.
15:0	RAZWI	0x0000	-	Reserved

#### 6.4.1.9 MAINNSPPCEXP<0 to 3>

The Main Interconnect Non-secure Access Subordinate Peripheral Protection Controller Expansion register 0, 1, 2, and 3 allows software to configure the security level of each Main Interconnect peripheral that resides in the subsystem expansion outside the subsystem. These registers can be used to control the PPCs that are outside the subsystem, which protect the Main Interconnect

peripherals connected to the Main Interconnect through the Subordinate Main Expansion interfaces.

Each field defines the Secure or Non-secure access setting for an associated peripheral, as follows:

- 1 - Allows Non-secure access only
- 0 - Allows Secure access only

These controls directly control the expansion signals on the Security Control Expansion interface. All four registers are similar and each register,  $N$  where  $N$  is from 0 to 3 is as follows:

**Table 6-19: MAINNSPPCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved.
15:0	RW	0x0000	MAINNSPPCEXP<N>	Expansion <N> Non-secure Access Main Interconnect Subordinate Peripheral Protection Control. Each bit $n$ drives the output signal <b>MAINNSPPCEXP&lt;N&gt;[n]</b> . The configuration option MAINPPCEXP<N>DIS defines if each bit within this register is actually implement such that if MAINPPCEXP<N>DIS[j] = 1'b1 then MAINNSPPCEXP<N>[j] is disabled, reads as zeros and any writes to it is ignored.

#### 6.4.1.10 PERIPHNSPPC0 and PERIPHNSPPC1

The Peripheral Interconnect Non-secure Access Peripheral Protection Controller registers allow software to configure whether each peripheral on the Peripheral Interconnect that it controls through a PPC is Secure access only or is Non-secure access only. Each field defines the Secure or Non-secure access setting for an associated peripheral, as follows:

- 1 - Allows Non-secure access only
- 0 - Allows Secure access only

SSE-310 has two such group of registers, Peripheral Protection Controller Group 0 and Peripheral Protection Controller Group 1, as follows:

**Table 6-20: PERIPHNSPPC0 Register**

Bits	Type	Default	Name	Description
31:8	RAZWI	0x000000	-	Reserved.
7	RAZWI	0x00	-	Reserved.
6	RAZWI	0x00	-	Reserved.
5	RW	0x00	NS_TIMER3	Access Non-Security for TIMER3
4	RAZWI	0x00	-	Reserved.
3	RAZWI	0x00	-	Reserved.
2	RW	0x00	NS_TIMER2	Access Non-Security for TIMER2
1	RW	0x00	NS_TIMER1	Access Non-Security for TIMER1
0	RW	0x00	NS_TIMER0	Access Non-Security for TIMER0

**Table 6-21: PERIPHNSPPC1 Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x00000000	-	Reserved.
0	RW	0x00	NS_SLOWCLK_TIMER	Access Non-Security for SLOWCLK_TIMER



- Access to the control interfaces of the Memory Protection Controllers in SSE-310 is fixed and is not represented in the table above. Access is filtered by PPC0 though its security settings. Therefore if this control interface is accessed using the wrong security level, it results in interrupts being raised for PPC0.
- Access to the control interface of the SLOWCLK Watchdog timer is also fixed. This access is filtered by PPC1. Therefore if this control interface is accessed using the wrong security, it results in interrupts being raised for PPC1..

#### 6.4.1.11 NPUSPPORSL

The NPU Secure access security level reset control registers (NPUSPPORSL) allows software to configure if each NPU resets to Secure or Non-secure state.

The Value of NPUSPPORSL is only sampled by the NPU, when the NPU is released from reset. The Default reset value of this register is controlled by **NPU0PORSLRST**.

Where:

- 1 = Reset to Non-secure state
- 0 = Reset to Secure state

**Table 6-22: NPUSPPORSL Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x000_000	-	Reserved
0	RW	NPU0PORSLRST	SP_NPU0PORSL	Configures the security level of the NPU0. When NUMNPU < 1, this field does not exist, is reserved and <b>RAZWI</b> .

#### 6.4.1.12 PERIPHNSPPCEXP<0 to 3>

The Peripheral Interconnect Non-secure Access Subordinate Peripheral Protection Controller Expansion registers 0, 1, 2, and 3 allow software to configure the security level of each Peripheral Interconnect peripheral that resides in the expansion logic outside the subsystem.

These registers control the PPCs that are outside the subsystem, which protect the Peripheral Interconnect peripherals connected to the Peripheral Interconnect through the Subordinate Peripheral Expansion interfaces.

Each field defines the Secure or Non-secure access setting for an associated peripheral, as follows:

- 1 - Allows Non-secure access only

- 0 - Allows Secure access only

These registers directly control the expansion signals on the Security Control Expansion interface. All four registers are similar and each register,  $N$  where  $N$  is from 0 to 3 is as follows:

**Table 6-23: PERIPHNSPPCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved
15:0	RW	0x0000	PERIPHNSPPCEXP<N>	Expansion <N> Non-secure Access Peripheral Interconnect Subordinate Peripheral Protection Control. Each bit $n$ drives the output signal <b>PERIPHNSPPCEXP&lt;N&gt;[n]</b> .  The configuration option PERIPHPPCEXP<N>DIS defines if each bit within this register is actually implemented such that if PERIPHPPCEXP<N>DIS[i] = 1'b1 then PERIPHPPCEXP<N>DIS[i] reads as zeros and any writes to it is ignored.

#### 6.4.1.13 MAINSPPPCCEXP<0 to 3>

The Expansion Secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure Secure privilege level of each Main Interconnect peripheral that resides in the expansion logic outside the subsystem.

These registers control the PPCs that are outside the subsystem, which protect the Main Interconnect peripherals connected to the Main Interconnect through the Subordinate Main Expansion interfaces.

Each field defines this for an associated peripheral, by the following settings:

- 1 - Allows Secure Unprivileged and Privileged access
- 0 - Allows Secure Privileged access only

These registers directly control the expansion signals on the Security Control Expansion interface. All four register are similar and each register,  $N$  where  $N$  is from 0 to 3 is as follows:

**Table 6-24: MAINSPPPCCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved
15:0	RW	0x0000	MAINSPPPCCEXP<N>	Expansion <N> Secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Control. Each bit $n$ driveS the output signal <b>MAINPPPCCEXP&lt;N&gt;[n]</b> if MAINNSPPCEXP<N>[n] is also LOW, where $N$ is 0 to 3.  The configuration option MAINPPCEXP<N>DIS defines if each bit within this register is actually implemented, such that if MAINPPCEXP<N>DIS[i] = 1'b1 then MAINSPPPCCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.



#### 6.4.1.14 PERIPHSPPPC0 and PERIPHSPPPC1

The Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register allows software to configure if each Peripheral Interconnect peripheral that it controls through a PPC is only Secure Privileged access only or is allowed Secure Unprivileged access as well. Each field defines this for an associated peripheral, by the following settings:

- 1 - Allows Secure Unprivileged and Privileged access
- 0 - Allows Secure Privileged access only

SSE-310 has two such registers as follows:

**Table 6-25: PERIPHSPPPC0 Register**

Bits	Type	Default	Name	Description
31:8	RAZWI	0x000000	-	Reserved
7	RAZWI	0x00	-	Reserved
6	RW	0x00	SP_WATCHDOG_REF	Secure Unprivileged setting for Secure Watchdog Refresh Frame
5	RW	0x00	SP_TIMER3	Secure Unprivileged setting for TIMER3
4	RAZWI	0x00	-	Reserved
3	RAZWI	0x00	-	Reserved
2	RW	0x00	SP_TIMER2	Secure Unprivileged setting for TIMER2
1	RW	0x00	SP_TIMER1	Secure Unprivileged setting for TIMER1
0	RW	0x00	SP_TIMER0	Secure Unprivileged setting for TIMER0

**Table 6-26: PERIPHSPPPC1 Register**

Bits	Type	Default	Name	Description
31:2	RAZWI	0x00000000	-	Reserved
0	RW	0x00	SP_SLOWCLK_TIMER	Secure Unprivileged setting for SLOWCLK_TIMER

#### 6.4.1.15 NPUSPPORPL

The NPU Secure access privilege level reset control registers allows software to configure if NPU0 resets to Privileged or Unprivileged state. The value of this register is only sampled by the NPU, when the NPU is released from reset and NPUSPPORSL.SP\_NPU0PORSL is Secure State.

The default reset value of this register is controlled by **NPU0PORPLRST**.

Where:

- 1 = Reset to Privileged state
- 0 = Reset to Unprivileged state

**Table 6-27: NPUSPPORPL Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x000_000	-	Reserved
0	RW	NPU0PORPLRST	SP_NPU0PORPL	Configures the Secure access privilege level for NPU0. When NUMNPU < 1, this field does not exist, is reserved and <b>RAZWI</b> .

#### 6.4.1.16 PERIPHSPPPCEXP<0 to 3>

The Expansion Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2, and 3 allow software to configure the Secure privilege level of Peripheral Interconnect peripherals. These registers can be used to control the PPCs that are outside the subsystem, which protect the Peripheral Interconnect peripherals connected to the Peripheral Interconnect through the Subordinate Peripheral Expansion interfaces.

Each field defines this access for an associated peripheral, by the following settings:

- 1 - Allows Secure Unprivileged and Privileged access
- 0 - Allows Secure Privileged access only

These registers directly control the expansion signals on the Security Control Expansion interface. All four register are similar and each register, N where N is from 0 to 3 is as follows:

**Table 6-28: PERIPHSPPPCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved.
15:0	RW	0x0000	PERIPHSPPPCEXP<N>	Expansion <N> Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Control. Each bit n drives the output signal <b>PERIPHPPPCEXP&lt;N&gt;[n]</b> if PERIPHNSPPCEXP<N>[n] is also LOW, where N is 0 to 3.  The configuration option PERIPHPPCEXP<N>DIS defines if each bit within this register is actually implemented such that if PERIPHPPCEXP<N>DIS[i] = 1'b1 then PERIPHSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

### 6.4.1.17 NSMSCEXP

The Non-secure Expansion Manager Security Controller register allows software to configure if each manager that is located behind each MSC in the subsystem expansion is a Secure or Non-secure device.

**Table 6-29: NSMSCEXP Register**

Bits	Type	Default	Name	Description
31:16	RW	NSMSCEXPST	NS_MSCEXP	Expansion MSC Non-secure configuration. Each bit n (0 to 15) local in this field controls the Non-secure configuration of each MSC and drives the signals <b>NSMSCEXP[n]</b> . Set HIGH to define a Manager as Non-secure, or LOW for Secure.  The parameter MSCEXPDIS defines if each bit within this register is actually implemented such that if MSCEXPDIS[i] = 1'b1 then NS_MSCEXP[i] is disabled, it reads as 0b1 and any writes to it is ignored. Resets to NSMSCEXPST.
15:0	RAZWI	0x0000	-	Reserved

### 6.4.2 Non-secure Access Configuration Register Block

The Non-secure Access Configuration Register Block implements program visible states that allows software to control various security gating units within the design.

This register block base address is 0x40080000. This register block is Non-secure Privileged access only and supports 32-bit R/W accesses. For write access to these registers, only 32-bit writes are supported. Any Byte and Half word writes are ignored.

The following table lists the registers within this unit. Details of each register are described in separate sections.

All registers reside in the PD\_SYS power domain and are reset by **nWARMRESETSYS**.

**Table 6-30: Non-secure Access Configuration Register Block Register Map**

Offset	Name	Access	Reset value	Description
0x000 - 0x08C	Reserved	RAZWI	0x00000000	Reserved
0x090	Reserved	RAZWI	0x00000000	Reserved
0x094 - 0x09C	Reserved	RAZWI	0x00000000	Reserved
0x0A0	MAINNSPPPCEXP0	Read-write	0x00000000	Expansion 0 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
0x0A4	MAINNSPPPCEXP1	Read-write	0x00000000	Expansion 1 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
0x0A8	MAINNSPPPCEXP2	Read-write	0x00000000	Expansion 2 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
0x0AC	MAINNSPPPCEXP3	Read-write	0x00000000	Expansion 3 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.

Offset	Name	Access	Reset value	Description
0x0B0	PERIPHNSPPPC0	Read-write	0x00000000	Non-secure Unprivileged Access Peripheral Protection Control 0 on Peripheral Interconnect.
0x0B4	PERIPHNSPPPC1	Read-write	0x00000000	Non-secure Unprivileged Access Peripheral Protection Control 1 on Peripheral Interconnect.
0x0BC	NPUNSPORPL	Read-write	Configurable	Non-secure Access NPU Privilege level reset state control
0x0B8	Reserved	<b>RAZWI</b>	0x00000000	
0x0C0	PERIPHNSPPPCEXP0	Read-write	0x00000000	Expansion 0 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
0x0C4	PERIPHNSPPPCEXP1	Read-write	0x00000000	Expansion 1 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
0x0C8	PERIPHNSPPPCEXP2	Read-write	0x00000000	Expansion 2 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
0x0CC	PERIPHNSPPPCEXP3	Read-write	0x00000000	Expansion 3 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
0x0D0 - 0xFCC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000053	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000003B	Peripheral ID 2
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

### 6.4.2.1 MAINNSPPPCEXP<0 to 3>

The Expansion Non-secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure each Main Interconnect peripheral.

These registers can be used to control the PPCs that are outside the subsystem, which protect the Main Interconnect peripherals connected to the Main Interconnect through the Subordinate Main Expansion interfaces.

Each field defines the Secure or Non-secure access setting for an associated peripheral, as follows:

- 1 - Allows Non-secure Unprivileged and Privileged access
- 0 - Allows Non-secure Privileged access only

These registers directly control the expansion signals on the Security Control Expansion interface. All four register are similar and each register, N where N is from 0 to 3 is as follows:

**Table 6-31: MAINNSPPPCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved
15:0	RW	0x0000	MAINNSPPPCEXP<N>	Expansion <N> Non-secure Privilege Access Main Interconnect Subordinate Peripheral Protection Control. Each bit n drives the output signal <b>MAINPPPCEXP&lt;N&gt;[n]</b> if MAINNSPPPCEXP<N>[n] is also HIGH, where N is 0 to 3. The configuration option MAINPPPCEXP<N>DIS defines if each bit within this register is actually implemented, such that if MAINPPPCEXP<N>DIS[i] = 1'b1 then MAINNSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

### 6.4.2.2 PERIPHNSPPPC0 and PERIPHNSPPPC1

Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register allows software to configure if each Peripheral Interconnect peripheral that it controls through a PPC is only Non-secure Privileged access only or is allowed Non-secure Unprivileged access as well. Each field defines this for an associated peripheral, by the following settings:

- 1 - Allows Non-secure Unprivileged and Privileged access
- 0 - Allows Non-secure Privileged access only

SSE-310 has two such registers as follows:

**Table 6-32: PERIPHNSPPPC0 Register**

Bits	Type	Default	Name	Description
31:8	RAZWI	0x0000000	-	Reserved
7	RAZWI	0x00	-	Reserved
6	RW	0x00	NSP_WATCHDOG_REF	Non-secure Unprivileged setting for Non-secure Watchdog Refresh Frame
5	RW	0x00	NSP_TIMER3	Non-secure Unprivileged setting for TIMER3

Bits	Type	Default	Name	Description
4	RAZWI	0x00	-	Reserved
3	RAZWI	0x00	-	Reserved
2	RW	0x00	NSP_TIMER2	Non-secure Unprivileged setting for TIMER2
1	RW	0x00	NSP_TIMER1	Non-secure Unprivileged setting for TIMER1
0	RW	0x00	NSP_TIMER0	Non-secure Unprivileged setting for TIMER0

**Table 6-33: PERIPHNSPPPC1 Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x00000000	-	Reserved
0	RW	0x00	NSP_SLOWCLK_TIMER	Non-secure Unprivileged setting for SLOWCLK_TIMER

### 6.4.2.3 NPUNSPORPL

The NPU Non-secure access privilege level reset control registers allows software to configure if each NPU resets to Privileged or Unprivileged state.

The value of this register is only sampled by the NPU, when the NPU is released from reset and NPUSPPORSL.SP\_NPUOPORSL is in Non-secure State. The default reset value of this register is controlled by **NPUOPORPLRST**.

Where:

- 1 = Reset to Privileged state
- 0 = Reset to Unprivileged state

**Table 6-34: NPUNSPORPL Register**

Bits	Type	Default	Name	Description
31:1	RAZWI	0x000000	-	Reserved
0	RW	NPUOPORPLRST	NS_NPUOPORPL	Configures the non-secure access privilege level for NPU0. When NUMNPU < 1, this field does not exist, is reserved and <b>RAZWI</b> .

### 6.4.2.4 PERIPHNSPPPCEXP<0 to 3>

The Expansion Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2, and 3 allow software to configure each Peripheral Interconnect peripheral.

These registers can be used to control the PPCs that are outside the subsystem, which are protecting the Peripheral Interconnect peripherals connected to the Peripheral Interconnect through the Subordinate Peripheral Expansion interfaces.

Each field defines this for an associated peripheral, by the following settings:

- 1 - Allows Non-secure Unprivileged and Privileged access

- 0 - Allows Non-secure Privileged access only

These registers directly control the expansion signals on the Security Control Expansion interface. All four register are similar and each register, N where N is from 0 to 3 is as follows:

**Table 6-35: PERIPHNSPPPCEXP<N> Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved.
15:0	RW	0x0000	PERIPHNSPPPCEXP<N>	Expansion <N> Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Control. Each bit n drives the output signal <b>PERIPHPPPCEXP&lt;N&gt;[n]</b> if PERIPHNSPPPCEXP<N>[n] is also HIGH, where N is 0 to 3. The configuration option PERIPHPPPCEXP<N>DIS defines if each bit within this register is actually implement such that if PERIPHPPPCEXP<N>DIS[i] = 1'b1, then PERIPHNSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

### 6.4.3 Timestamp Timers

SSE-310 implements four timestamp-based timers in the system, TIMER<n> where N is 0 to 3. All timers are mapped to the Secure or Non-secure world through PPC0, which also controls the accessibility of Unprivileged accesses.

For more details, see [Secure Access Configuration Register Block](#).

All timestamp timers and watchdog, except for Timer3, reside in PD\_SYS power domain and is reset by **nWARMRESETSYS**, while the Timer 3 resides in the PD\_AON power domain and is reset by **nWARMRESETAON**.

For more information of the ARMv8M System Counter Timer registers, see Arm® Corstone™ Reference Systems Architecture Specification Ma1.

### 6.4.4 Timestamp Watchdogs

SSE-310 implements two timestamp-based watchdogs in the system. Both reside in PD\_SYS power domain and are reset by **nWARMRESETSYS**. One watchdog timer is Secure access only, while another is Non-secure.

Accessibility of each watchdog to Unprivileged access is also controlled by through PPC0. For more details, see [PERIPHSPPPCO](#) and [PERIPHSPPPC1](#) and [PERIPHNSPPPCO](#) and [PERIPHNSPPPC1](#).

Each Watchdog timer implements two register frames, a Control Frame and a Refresh Frame.

For more information of the ARMv8M Timestamp Watchdog registers, see Arm® Corstone™ Reference Systems Architecture Specification Ma1.

## 6.4.5 NPU registers

SSE-310 implements one Ethos-U55 U55 NPU.

For details of the NPU software interface, see *AMBA® Ethos™-U55 NPU Technical Reference Manual*.

## 6.5 Processor Private Region

The CPU0 has its own copy of the Processor Private Region which is only assessable to itself. The Processor Private Region consists of four subregions as follows:

- 0x40010000 to 0x4001FFFF implements a Non-secure Low Access Latency Region
- 0x48010000 to 0x4801FFFF implements a Non-secure High Access. Latency Region
- 0x50010000 to 0x5001FFFF implements a Secure Low Access Latency Region
- 0x58010000 to 0x5801FFFF implements a Secure High Access Latency Region

Of these four regions above, only 0x40010000 to 0x4001FFFF and 0x50010000 to 0x5001FFFF implement any registers.

None of these regions are accessible from any other manager in the system, except when using the external debugger through CPU0.

The memory map of the Processor Private Region is as follows:

**Table 6-36: Processor Private Region address map**

Row ID	Address - from	Address - to	Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
0	0x40010000	0x40011FFF	-	Reserved	Reserved	-	-
1	0x40012000	0x40012FFF	4KB	CPU0_PWRCTRL	CPU0 Power Control Block. See <a href="#">CPU0_PWRCTRL Register Block</a>	7	NS, P
2	0x40013000	0x4001EFFF	-	Reserved	Reserved	-	-
3	0x4001F000	0x4001FFFF	4KB	CPU0_IDENTITY	CPU0 Identity Block, See <a href="#">CPU0_IDENTITY Register Block</a>	9	NS, UP
4	0x48010000	0x4801FFFF	-	Reserved	Reserved	-	-
5	0x50010000	0x50010FFF	-	Reserved	Reserved	-	-
6	0x50011000	0x50011FFF	4KB	CPU0_SECCRTL	CPU0 Local Security Control Block, See <a href="#">CPU0_SECCRTL Register Block</a>	-	S, P
7	0x50012000	0x50012FFF	4KB	CPU0_PWRCTRL	CPU0 Power Control Block. See <a href="#">CPU0_PWRCTRL Register Block</a>	1	S, P
8	0x50013000	0x5001EFFF	-	Reserved	Reserved	-	-
9	0x5001F000	0x5001FFFF	4KB	CPU0_IDENTITY	CPU0 Identity Block, See <a href="#">CPU0_IDENTITY Register Block</a>	3	S, UP
10	0x58010000	0x5801FFFF	-	Reserved	Reserved	-	-

<sup>1</sup> Legend



- S: Secure access only.
- NS: Non-secure access only.
- P: Privilege access only.
- UP: Unprivileged and Privilege access allowed.

## 6.5.1 CPU0\_PWRCTRL Register Block

SSE-310 implements a CPU0\_PWRCTRL register block for CPU0 in the subsystem. This block resides at address 0x40012000 in a Non-secure region and is also alias to 0x50012000 in the Secure region. The CPU0\_PWRCTRL registers are read only registers when accessed from the Non-secure region starting at address 0x40012000 and any writes access to it in that region is ignored.

The following table lists the registers in the CPU0\_PWRCTRL register block.

**Table 6-37: CPU0\_PWRCTRL register map**

Offset	Name	Access	Reset value	Description
0x000	CPUPWRCFG	Read-write	0x00000000	CPU0 Local Power Configuration register. See <a href="#">CPUPWRCFG</a> .
0x004 - 0xFCC	Reserved	Read-only	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x0000005A	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000000B	Peripheral ID 2
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

### 6.5.1.1 CPUPWRCFG

The CPUPWRCFG register provides the local CPU software control registers for power control.

This register is Read Only if accessed from the Non-secure world. This register resides in the same reset domain as the CPU0 core, **nWARMRESETCPU0** reset domain, and PD\_CPU0 power domain. When CPU0 is powered down, the register is also powered down, and are cleared when powered back up.

**Table 6-38: CPUPWRCFG Register**

Bits	Type	Default	Name	Description
31:5	RAZWI	0x00000000	-	Reserved.

Bits	Type	Default	Name	Description
4	RW	0x00	TCM_MIN_PWR_STATE	Defines the minimum power state of the TCM for CPU0. <ul style="list-style-type: none"> <li>'0' - OFF</li> <li>'1' - Retention.</li> </ul> <p>This bit is read access only from the Non-secure world.</p> <p>When PD_CPU0 returns from MEM_RET or MEM_RET_NOCACHE state to one of the ON states, this bit is set to 1'b1.</p>
3:1	RAZWI	0x00	-	Reserved
0	RAZWI	0x00	-	Reserved

## 6.5.2 CPU0\_IDENTITY Register Block

SSE-310 implements a CPU0\_IDENTITY register block for CPU0 . This block resides at address 0x4001F000 in a Non-secure region and is also alias to 0x5001F000 in the Secure region. This is a read only register and any write access is ignored.

The following table lists the registers in the CPU0\_IDENTITY block.

**Table 6-39: CPU0\_IDENTITY register map**

Offset	Name	Access	Reset value	Description
0x000	CPUID	Read-only	CPU0CPUIRST	Unique CPU Identity Number
0x004 - 0xFCC	Reserved	Read-only	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000055	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000000B	Peripheral ID 2
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

### 6.5.2.1 CPUID

The CPUID register is a read only register that when read by CPU0, provides an identity code to the CPU that is unique to that CPU.

**Table 6-40: CPUID Register**

Bits	Type	Default	Name	Description
31:4	RAZWI	0x00000000	-	Reserved.

Bits	Type	Default	Name	Description
3:0	RO	CPU0CPUIDRST	CPUID	CPU Identity. Defined by configuration CPU0CPUIDRST.

### 6.5.3 CPU0\_SECCTRL Register Block

CPU0 has a CPU0\_SECCTRL register block that allows the security locks of the processor to be configured. The register block resides in nWARMRESETCPU0 reset domain and PD\_CPU0 power domain so that when a core is powered down, they are also powered down and are cleared when powered back up. These registers are Secure access only and resides at address 0x50011000.

The following table lists the registers in the CPU0\_SECCTRL Register block.

**Table 6-41: CPU0\_SECCTRL register map**

Offset	Name	Access	Reset value	Description
0x000	CPUSECCFG	Read-write	0x00000000	CPU Local Security Configuration. See <a href="#">CPUSECCFG</a>
0x004 - 0xFCC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000059	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000001B	Peripheral ID 2
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

#### 6.5.3.1 CPUSECCFG

The CPU Local Security Configuration Register allows software to set security lock bits at the CPU interface.

**Table 6-42: CPUSECCFG Register.**

Bits	Type	Default	Name	Description
31:6	<b>RAZWI</b>	0x00000000	-	Reserved.
5	RW1S	0x00	LOCKDTGU	When HIGH, disables writes to the CPU0 DTGU_CTRL and DTGU_LUTn registers from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.
4	RW1S	0x00	LOCKITGU	When HIGH, disables writes to the CPU0 ITGU_CTRL and ITGU_LUTn from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.
3	RW1S	0x00	LOCKTCM	When HIGH, disables writes to the CPU0 ITCMCR, DTCMCR from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.

Bits	Type	Default	Name	Description
2	RW1S	0x00	LOCKSMPU	When HIGH, disables write to the CPU0 MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An, MPU_RLAR_An registers associated with the Secure MPU from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.
1	RW1S	0x00	LOCKSAU	When HIGH, disables writes to the CPU0 SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.
0	RW1S	0x00	LOCKSVTAIRCR	When HIGH, disables writes to the CPU0 VTOR_S, AIRCR.PRIS, and AIRCR.BFHFNMINS registers. Once set to HIGH, it cannot be cleared until Reset.

## 6.6 System Control Peripheral Region

The System Control Peripheral Region is a collection of memory regions where system control related peripherals are mapped. These peripherals reside in the PD\_AON power domain. There are four regions in total as follows:

- 0x40020000 to 0x4003FFFF, which is a Non-secure region for low latency system control peripherals. Some peripherals might be expected to be aliased in its associated Secure region, 0x50020000 to 0x5003FFFF.
- 0x48020000 to 0x4803FFFF, which is a Non-secure region for high latency system control peripherals. Some peripherals are expected to be aliased in its associated Secure region, 0x58020000 to 0x5803FFFF.
- 0x50020000 to 0x5003FFFF, which is a Secure region for low latency system control peripherals. Some peripherals might be expected to be aliased in its associated Non-secure region, 0x40020000 to 0x4003FFFF.
- 0x58020000 to 0x5803FFFF, which is a Secure region for low latency system control peripherals. Some peripherals are expected to be aliased in its associated Non-secure region, 0x48020000 to 0x4803FFFF.

For an aliased peripheral in these regions, mapping of each to either Secure or Non-secure region is determined by Peripheral Protection Controllers (PPC) that are controlled using the Secure Access Configuration Register Block. These PPCs also define Privileged or Unprivileged accessibility. For more information about the Secure Access Configuration Register Block, see [Secure access configuration register block](#).

**Table 6-43: System Control Peripheral Region address map**

Row ID	Address - from	Address - to	Size	Region Name	Description	Alias with row ID	Security <sup>1</sup>
1	0x40020000	0x4003FFFF	128KB	Reserved	Reserved. When accessed, results in bus error.	-	-
2	0x48020000	0x48020FFF	4KB	SYSINFO	System Information Register Block. See <a href="#">SYSINFO Register Block</a> .	-	NS, UP
3	0x48021000	0x4802EFFF	56KB	Reserved	Reserved. When accessed, results in <b>RAZWI</b> .	-	NS, UP
4	0x4802F000	0x4802FFFF	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See <a href="#">SLOWCLK AON Timers</a> .	31	NS-PPC, P-PPC
5	0x48030000	0x4803FFFF	64KB	Reserved	Reserved. When accessed, results in bus error.	-	-

Row ID	Address - from	Address - to	Size	Region Name	Description	Alias with row ID	Security <sup>1</sup>
6	0x50020000	0x5003FFFF	128KB	Reserved	Reserved. When accessed, results in bus error.	-	-
7	0x58020000	0x58020FFF	4KB	SYSINFO	System Information Register Block. See <a href="#">SYSINFO Register Block</a> .	-	S, UP
8	0x58021000	0x58021FFF	4KB	SYSCONTROL	System Control Register Block. See <a href="#">SOC_IDENTITY</a> .	-	S, P
18	0x58022000	0x58022FFF	4KB	SYSPPU	PPU for BR_SYS. See <a href="#">Power Policy Units</a> .	-	S, P
19	0x58023000	0x58023FFF	4KB	CPU0PPU	PPU for BR_CPU0. See <a href="#">Power Policy Units</a> .	-	S, P
20	0x58024000	0x58024FFF	4KB	Reserved	Reserved	-	-
21	0x58025000	0x58025FFF	4KB	Reserved	Reserved	-	-
22	0x58026000	0x58026FFF	4KB	Reserved	Reserved	-	-
23	0x58027000	0x58027FFF	4KB	Reserved	Reserved	-	-
24	0x58028000	0x58028FFF	4KB	MGMTPPU	PPU for BR_MGMT. See <a href="#">Power Policy Units</a> .	-	S, P
25	0x58029000	0x58029FFF	4KB	DEBUGPPU	PPU for BR_DEBUG. See <a href="#">Power Policy Units</a> .	-	S, P
26	0x5802A000	0x5802_AFFF	4KB	NPU0PPU	PPU for BR_NPU0. See <a href="#">Power Policy Units</a> .	-	S, P
27-29	0x5802B000	0x5802DFFF	12KB	Reserved	Reserved. When accessed, results in <b>RAZWI</b> .	-	-
30	0x5802E000	0x5802EFFF	4KB	SLOWCLK Watchdog	Watchdog Timer running on SLOWCLK. See <a href="#">SLOWCLK AON Timers</a> .	-	S, P
31	0x5802F000	0x5802FFFF	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See <a href="#">SLOWCLK AON Timers</a> .	4	S-PPC, P-PPC
32	0x58030000	0x5803FFFF	64KB	Reserved	Reserved. When accessed, results in bus error.	-	-

<sup>1</sup> Legend

- NS-PPC: Non-secure access only, gated by a PPC.
- S-PPC: Secure access only, gated by a PPC.
- S: Secure access only.
- NS: Non-secure access only.
- P: Privilege access only.
- UP: Unprivileged and privilege access allowed.
- P-PPC: Unprivileged access controlled by a PPC.

## 6.6.1 SYSINFO Register Block

The System Information Register Block provides information on the system configuration and identity. This register block is read-only and is accessible by accesses of any security attributes. This module resides at base address 0x58020000 in the Secure region, and 0x48020000 in the Non-secure region.

Details of each register are described in separate sections.

**Table 6-44: System Information register map**

Offset	Name	Access	Reset value	Description
0x000	SOC_IDENTITY	Read-only	Configurable	SoC Identity Register. See <a href="#">SOC_IDENTITY</a> .
0x004	SYS_CONFIG0	Read-only	Configurable	System Hardware Configuration 0 Register. See <a href="#">SYS_CONFIG0</a> , <a href="#">SYS_CONFIG1</a> and <a href="#">SYS_CONFIG2</a> .
0x008	SYS_CONFIG1	Read-only	Configurable	System Hardware Configuration 1 Register. See <a href="#">SYS_CONFIG0</a> , <a href="#">SYS_CONFIG1</a> and <a href="#">SYS_CONFIG2</a> .
0x00C	SYS_CONFIG2	Read-only	Configurable	System Hardware Configuration 2 Register. See <a href="#">SYS_CONFIG0</a> , <a href="#">SYS_CONFIG1</a> and <a href="#">SYS_CONFIG2</a> .
0x010 - 0xFC4	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0xFC8	IIDR	Read-only	Configurable	Subsystem Implementation Identity Register. See <a href="#">IIDR</a> .
0xFCC	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4.
0xFD4 - 0xFDC	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0xFE0	PIDR0	Read-only	0x00000058	Peripheral ID 0.
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1.
0xFE8	PIDR2	Read-only	0x0000002B	Peripheral ID 2.
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3.
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0.
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1.
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2.
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3.

### 6.6.1.1 SOC\_IDENTITY

The System-On-Chip (SoC) Identity register provides an area where software can find out about the SoC's part number, its implementor and revision number. These are defined by configuration parameters.

**Table 6-45: SOC\_IDENTITY Register**

Bits	Type	Default	Name	Description
31:20	RO	SOCPRID	SOC_PRODUCT_ID	Configurable value identifying the SoC.
19:16	RO	SOCVAR	SOC_VARIANT	Configurable value indicating major revision of the SoC.

Bits	Type	Default	Name	Description
15:12	RO	SOCREV	SOC_REVISION	Configurable value used to distinguish minor revisions of the SoC.
11:0	RO	SOCIMPLID	SOC_IMPLEMENTATOR	Contains the JEP106 code of the company that implemented the SoC: <ul style="list-style-type: none"> <li>[11:8] JEP106 continuation code of implementer</li> <li>[7] Always 0</li> <li>[6:0] JEP106 identity code of implementer</li> </ul>

When EXPLOGIC\_PRESENT = 1, the SoC identity register fields define the TARGETID of the SoC debug port in the subsystem expansion as follows:

- TARGETID[31:28] uses SOCVAR
- TARGETID[27:16] uses SOCPRTID
- TARGETID[15:12] tied to 0x0
- TARGETID[11:1] uses {SOCIMPLID[11:8], SOCIMPLID[6:0]}
- TARGETID[0] tied to 0b1

For more information about TARGETID, see Arm® *Debug Interface Architecture Specification* ADIv6.0.

### 6.6.1.2 SYS\_CONFIG0, SYS\_CONFIG1 and SYS\_CONFIG2

The System Hardware Configuration registers provides several registers to allow software to query the configuration of the SSE-310 based system.



In these tables, the fields CPU0\_TCM\_BANK\_NUM and CPU0\_HAS\_SYSTCM refer to TCMs that are implemented on the system interconnect close to each associated core, rather than the TCMs that are implemented within the processor core. SSE-310 does not support TCMs being implemented on the system interconnect.

**Table 6-46: SYS\_CONFIG0 register**

Bits	Type	Default	Name	Description
31:28	RAZWI	0x0	-	Reserved
27	RAZWI	0x0	-	Reserved
26:24	RAZWI	0x0	-	Reserved
23:20	RAZWI	0x0	Reserved	Reserved
19	RAZWI	0x0	Reserved	Reserved
18:16	RO	0x4	CPU0_TYPE	CPU0 Core Type: <ul style="list-style-type: none"> <li>000 - Does not exist</li> <li>100 - Cortex-M85 processor</li> <li>Others - Reserved</li> </ul>
15:13	RO	0x0	Reserved	Reserved

Bits	Type	Default	Name	Description
12:11	RO	0x1	PI_LEVEL	Power Infrastructure Level: <ul style="list-style-type: none"> <li>00 - Basic Level</li> <li>01 - Intermediate Level</li> <li>10 - Advance Level</li> <li>Others - Reserved</li> </ul>
10	RO	0x0	HAS_CSS	It reflects whether the CoreSight SoC-600-based common debug infrastructure is included. <ul style="list-style-type: none"> <li>0 = No</li> <li>1 = Yes</li> </ul>
9	RAZWI	0x0	Reserved	Reserved
8:4	RO	VMADDRWIDTH	VM_ADDR_WIDTH	Volatile Memory Bank Address Width, where the size of each bank is equal to $2^{\text{VM\_ADDR\_WIDTH}}$ bytes.
3:0	RO	0x2	NUM_VM_BANK	Number of Volatile Memory Banks.

**Table 6-47: SYS\_CONFIG1 register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x0000	-	Reserved
15:12	RAZWI	0x0	-	Reserved
11	RAZWI	0x0	-	Reserved
10:8	RAZWI	0x0	-	Reserved
7:4	RAZWI	0x0	-	Reserved
3	RAZWI	0x0	-	Reserved
2:0	RAZWI	0x0	-	Reserved

**Table 6-48: SYS\_CONFIG2 register**

Bits	Type	Default	Name	Description
31:15	RAZWI	0x0000	-	Reserved
14:12	RAZWI	0x0	-	Reserved
11:3	RAZWI	0x0000	-	Reserved
2:0	RO	NPU0TYPE	NPU0_TYPE	NPU 0 Core Type: <ul style="list-style-type: none"> <li>000 - Does not exist</li> <li>001 - Ethos-U55</li> <li>Others - Reserved</li> </ul>



### 6.6.1.3 IIDR

The Subsystem Implementation Identity register provides an area where software can find out about the Subsystem Implementation part number, its implementor, and revision number. These are defined by configuration parameters.

**Table 6-49: IIDR Register**

Bits	Type	Default	Name	Description
31:20	RO	IMPLPRTID	IMP_PRODUCT_ID	Configurable value identifying the subsystem implementation.
19:16	RO	IMPLVAR	IMP_VARIANT	Configurable value indicating variant or major revision of the subsystem implementation.
15:12	RO	IMPLREV	IMP_REVISION	Configurable value used to distinguish minor revisions of the subsystem implementation.
11:0	RO	IMPLID	IMP_IMPLEMENTATOR	Contains the JEP106 code of the company that implemented the subsystem: <ul style="list-style-type: none"> <li>[11:8] JEP106 continuation code of implementer.</li> <li>[7] Always 0.</li> <li>[6:0] JEP106 identity code of implementer.</li> </ul>

When EXPLOGIC\_PRESENT = 1, the subsystem implementation identity register fields define the PIDR values of the MCU debug ROM table - that is the first debug ROM table in the system - in the subsystem expansion, as follows:

- REVISION uses IMPLVAR
- {PART\_1, PART\_0} uses IMPLPRTID
- {DES\_2, DES\_1, DES\_0} uses IMPLID
- REVAND uses IMPLREV

For more information about PIDR registers of debug ROM table, see *Arm® CoreSight™ Architecture Specification v3.0*.

### 6.6.2 System Control Register Block

The System Control Register Block implements registers for power, clocks, resets and other general system control. This module resides at base address 0x58021000 in the Secure region.

The System Control Register Block is Secure Privilege access only. For write access to these registers, only 32-bit writes are supported. Any Byte and Half-word writes result in its write data ignored.

The following table shows the details of this register block.

This System Control Registers Block resides in the PD\_AON power domain.

**Table 6-50: System Control register map**

Offset	Name	Access	Reset value	Description
0x000	SECDBGSTAT	Read-only	Configurable	Secure Debug Configuration Status Register  See <a href="#">SECDBGSTAT</a> , <a href="#">SECDBGSET</a> and <a href="#">SECDBGCLR</a>
0x004	SECDBGSET	Read-write	0x00000000	Secure Debug Configuration Set Register  See <a href="#">SECDBGSTAT</a> , <a href="#">SECDBGSET</a> and <a href="#">SECDBGCLR</a>
0x008	SECDBGCLR	Write-only	0x00000000	Secure Debug Configuration Clear Register  See <a href="#">SECDBGSTAT</a> , <a href="#">SECDBGSET</a> and <a href="#">SECDBGCLR</a>
0x00C	SCSECCTRL	Read-write	0x00000000	System Control Security Controls Register  See <a href="#">SCSECCTRL</a>
0x010	CLK_CFG0	Read-write	Configurable	Clock Configuration Register 0  See <a href="#">CLK_CFG0</a> , <a href="#">CLK_CFG1</a> and <a href="#">CLK_CFG2</a>
0x014	CLK_CFG1	Read-write	Configurable	Clock Configuration Register 1  See <a href="#">CLK_CFG0</a> , <a href="#">CLK_CFG1</a> and <a href="#">CLK_CFG2</a>
0x018	CLOCK_FORCE	Read-write	Configurable	Clock forces  See <a href="#">CLOCK_FORCE</a>
0x01C	CLK_CFG2	Read-write	Configurable	Clock Configuration Register 2. See <a href="#">CLK_CFG0</a> , <a href="#">CLK_CFG1</a> and <a href="#">CLK_CFG2</a>
0x020 - 0x0FF	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x100	RESET_SYNDROME	Read-write	0x00000001	Reset syndrome  See <a href="#">RESET_SYNDROME</a>
0x104	RESET_MASK	Read-write	Configurable	Reset mask  See <a href="#">RESET_MASK</a>
0x108	SWRESET	Write-only	0x00000000	Software reset  See <a href="#">SWRESET</a>
0x10C	GRETREG	Read-write	0x00000000	General Purpose Retention Register  See <a href="#">GRETREG</a>
0x110	INITSVTOR0	Read-write	Configurable	CPU 0 Initial Secure Reset Vector Register  See <a href="#">INITSVTOR0</a>
0x114	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x118	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x11C	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x120	CPUWAIT	Read-write	Configurable	CPU Boot Wait Control  See <a href="#">CPUWAIT</a>

Offset	Name	Access	Reset value	Description
0x124	NMI_ENABLE	Read-write	Configurable	Enabling and Disabling Non Maskable Interrupts See <a href="#">NMI_ENABLE</a>
0x128	PPUINTSTAT	Read-only	0x00000000	PPU Interrupt Status. See <a href="#">PPUINTSTAT</a>
0x12C - 0x1F8	Reserved	<b>RAZWI</b>	0x00000000	Reserved
0x1FC	PWRCTRL	Read-write	0x00000003	Power Configuration and Control See <a href="#">PWRCTRL</a>
0x200	PDCM_PD_SYS_SENSE	Read-write	Configurable	PDCM PD_SYS sensitivity See <a href="#">PDCM_PD_SYS_SENSE</a>
0x204	PDCM_PD_CPU0_SENSE	Read-only	0x00000000	PDCM PD_CPU0 sensitivity See <a href="#">PDCM_PD_CPU0_SENSE</a>
0x208	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x20C	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x210	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x214	PDCM_PD_VMR0_SENSE	Read-write	0x40000000	PDCM PD_VMR0 sensitivity See <a href="#">PDCM_PD_VMR&lt;M&gt;_SENSE</a>
0x218	PDCM_PD_VMR1_SENSE	Read-write	0x40000000	PDCM PD_VMR1 sensitivity
0x21C	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x220	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x224 - 0x248	Reserved	<b>RAZWI</b>	0x00000000	Reserved.
0x24C	Reserved	Read-only	0x00000000	Reserved.
0x250 - 0xFCC	Reserved	Read-only	0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000054	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B8	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000004B	Peripheral ID 2
0xFEC	PIDR3	Read-only	0x00000000	Peripheral ID 3
0xFF0	CIDR0	Read-only	0x0000000D	Component ID 0

Offset	Name	Access	Reset value	Description
0xFF4	CIDR1	Read-only	0x000000F0	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

### 6.6.2.1 SECDBGSTAT, SECDBGSET and SECDBGCLR

The Secure Debug Configuration registers are used to select the source value for the Secure Debug Authentication, **DBGEN**, **NIDEN**, **SPIDEN**, **SPNIDEN**, **DAPACCEN**, and Debug Access Controls, **DAPDSSACCEN**, and **SYSDSSACCEN0**.

A selector is provided for each signal, to select between an internal register value and the value on the boundary of the Subsystem.

Secure software can set or clear the internal register and selector values by setting the associated bit in the SECDBGSET register or in the SECDBGCLR register, respectively:

- To set DBGEN\_I value to HIGH, write 1 to the SECDBGSET.DBGEN\_I\_SET register.
- To set DBGEN\_SEL value to HIGH, write 1 to the SECDBGSET.DBGEN\_SEL\_SET register.
- To set DBGEN\_I value to LOW, write 1 to the SECDBGCLR.DBGEN\_I\_CLR register.
- To set DBGEN\_SEL value to LOW, write 1 to the SECDBGSET.DBGEN\_SEL\_CLR register.

Secure software can read the output values used system wide by reading the associated SECDBGSTAT register bit. Secure software can read internal register values by reading SECDBGSET:

- To read the output value of **DBGEN**, read the SECDBGSTAT.DBGEN\_STATUS register.
- To read the value of DBGEN\_I, read the SECDBGSET.DBGEN\_I\_SET register.
- To read the selector value DBGEN\_SEL, read the SECDBGSTAT for the DBGEN\_SEL\_STATUS field.

For example, the source of **DBGEN** value used in the system is selected by the DBGEN\_SEL where:

- If DBGEN\_SEL is LOW, the input **DBGENIN** signal defines the system wide **DBGEN** value.
- If DBGEN\_SEL is HIGH the internal register value DBGEN\_I defines the system wide **DBGEN** value.

The **DBGEN** value is also made available to external expansion logic through the **DBGEN** output signal of the subsystem.

Selector Disable Configuration options are provided to allow each of the selector to be forced to zero, forcing the associated SEL\_STATUS field to LOW, forcing each respective debug control output to use its external value:

- DBGENSELDIS for disabling DBGEN\_SEL
- NIDENSELDIS for disabling NIDEN\_SEL
- SPIDENSELDIS for disabling SPIDEN\_SEL
- SPNIDENSELDIS for disabling SPNIDEN\_SEL
- DAPACCENSELDIS for disabling DAPACCEN\_SEL
- DAPDSSACCENSELDIS for disabling DAPDSSACCEN\_SEL

These can be used to disable the ability for Secure firmware to modify or override the Secure Debug Authentication and the Debug Access Controls values.

These registers are reset by **nCOLDRESETAON**.

These registers reside in the PD\_AON power domain.

**Table 6-51: SECDBGSTAT Register**

Bits	Type	Default	Name	Description
31	RAZWI	0x00	-	Reserved.
30	RAZWI	0x00	-	Reserved.
29	RO	DAPDSSACCENSELDIS	DAPDSSACCENSELDIS_STATUS	Returns the DA PDSSACCENSELDIS configuration value when read.
28	RO	DAPACCENSELDIS	DAPACCENSELDIS_STATUS	Returns the DAPACCENSELDIS configuration value when read.
27	RO	SPNIDENSELDIS	SPNIDENSELDIS_STATUS	Returns the SPNIDENSELDIS configuration value when read.
26	RO	SPIDENSELDIS	SPIDENSELDIS_STATUS	Returns the SPIDENSELDIS configuration value when read.
25	RO	NIDENSELDIS	NIDENSELDIS_STATUS	Returns the NIDENSELDIS configuration value when read.
24	RO	DBGENSELDIS	DBGENSELDIS_STATUS	Returns the DBGENSELDIS configuration value when read.
23:18	RAZWI	0x0000	-	Reserved.
17	RAZWI	0x00	-	Reserved.
16	RAZWI	0x00	-	Reserved.
15	RAZWI	0x00	-	Reserved.
14	RAZWI	0x00	-	Reserved.
13	RAZWI	0x00	-	Reserved.
12	RAZWI	0x00	-	Reserved.
11	RO	0x00	DAPDSSACCEN_SEL_STATUS	Active High DAP to Debug Subsystem Access Enable Selector Value. This bit returns the DAPDSSACCEN_SEL value.  Forced to Zero if DA PDSSACCENSELDIS = 1.
10	RO	DAPDSSACCENIN	DAPDSSACCEN_STATUS	Active High DAP to Debug Subsystem Access Enable Value. This bit reflects the value on the <b>DAPDSSACCEN</b> pin.
9	RO	0x00	DAPACCEN_SEL_STATUS	Active High DAP Access Enable Selector Value. This bit returns the DAPACCEN_SEL value.  Forced to Zero if DAPACCENSELDIS = 1.
8	RO	DAPACCENIN	DAPACCEN_STATUS	Active High DAP Access Enable Value. This bit reflects the value on the <b>DAPACCEN</b> pin.

Bits	Type	Default	Name	Description
7	RO	0x00	SPNIDEN_SEL_STATUS	Active High Secure Privilege Non-Invasive Debug Enable Selector Value. This bit returns the SPNIDEN_SEL value.  Forced to Zero if SPNIDENSELDIS = 1.
6	RO	SPNIDENIN	SPNIDEN_STATUS	Active High Secure Privilege Non-Invasive Debug Enable Value. This bit reflects the value on the <b>SPNIDEN</b> pin.
5	RO	0x00	SPIDEN_SEL_STATUS	Active High Secure Privilege Invasive Debug Enable Selector Value. This bit returns the SPIDEN_SEL value.  Forced to Zero if SPIDENSELDIS = 1.
4	RO	SPIDENIN	SPIDEN_STATUS	Active High Secure Privilege Invasive Debug Enable Value. This bit reflects the value on the <b>SPIDEN</b> pin.
3	RO	0x00	NIDEN_SEL_STATUS	Active High Non-Invasive Debug Enable Selector Value. This bit returns the NIDEN_SEL value.  Forced to Zero if NIDENSELDIS = 1.
2	RO	NIDENIN	NIDEN_STATUS	Active High Non-Invasive Debug Enable Value. This bit reflects the value on the <b>NIDEN</b> pin.
1	RO	0x00	DBGEN_SEL_STATUS	Active High Debug Enable Selector Value. This bit returns the DBGEN_SEL value.  Forced to Zero if DBGENSELDIS = 1.
0	RO	DBGENIN	DBGEN_STATUS	Active High Debug Enable Value. This bit reflects the value on the <b>DBGEN</b> pin.

**Table 6-52: SECDBGSET Register**

Bits	Type	Default	Name	Description
31:18	RAZWI	0x0000	-	Reserved
17	RAZWI	0x00	-	Reserved.
16	RAZWI	0x00	-	Reserved.
15	RAZWI	0x00	-	Reserved.
14	RAZWI	0x00	-	Reserved.
13	RAZWI	0x00	-	Reserved.
12	RAZWI	0x00	-	Reserved.
11	RAZW1S	0x00	DAPDSSACCEN_SEL_SET	Set Active High DAP to Debug Subsystem Access Enable Selector. Write HIGH to set DAPDSSACCEN_SEL. <b>RAZWI</b> if DAPDSSACCENSELDIS = 1.  <b>RAZWI</b> if DAPDSSACCENSELDIS = 1.
10	RW1S	0x00	DAPDSSACCEN_I_SET	Set internal version of Active High DAP to Debug Subsystem Access Enable. Write HIGH to set DAPDSSACCEN_I. When read returns DAPDSSACCEN_I.  <b>RAZWI</b> if DAPDSSACCENSELDIS = 1.
9	RAZW1S	0x00	DAPACCEN_SEL_SET	Set Active High DAP Access Enable Selector. Write HIGH to set DAPACCEN_SEL.  <b>RAZWI</b> if DAPACCENSELDIS = 1.

Bits	Type	Default	Name	Description
8	RW1S	0x00	DAPACCEN_I_SET	Set internal version of Active High DAP Access Enable. Write HIGH to set DAPACCEN_I. When read returns DAPACCEN_I.  <b>RAZWI</b> if DAPACCENSELDIS = 1.
7	RAZW1S	0x00	SPNIDEN_SEL_SET	Set Active High Secure Privilege Non-Invasive Debug Enable Selector. Write HIGH to set SPNIDEN_SEL.  <b>RAZWI</b> if SPNIDENSELDIS = 1.
6	RW1S	0x00	SPNIDEN_I_SET	Set internal version of Active High Secure Privilege Non-Invasive Debug Enable. Write HIGH to set SPNIDEN_I. When read returns SPNIDEN_I.  <b>RAZWI</b> if SPNIDENSELDIS = 1.
5	RAZW1S	0x00	SPIDEN_SEL_SET	Set Active High Secure Privilege Invasive Debug Enable Selector. Write HIGH to set SPIDEN_SEL.  <b>RAZWI</b> if SPIDENSELDIS = 1.
4	RW1S	0x00	SPIDEN_I_SET	Set internal version of Active High Secure Privilege Invasive Debug Enable. Write HIGH to set SPIDEN_I. When read returns SPIDEN_I.  <b>RAZWI</b> if SPIDENSELDIS = 1.
3	RAZW1S	0x00	NIDEN_SEL_SET	Set Active High Non-Invasive Debug Enable Selector. Write HIGH to set NIDEN_SEL.  <b>RAZWI</b> if NIDENSELDIS = 1.
2	RW1S	0x00	NIDEN_I_SET	Set internal version of Active High Non-Invasive Debug Enable. Write HIGH to set NIDEN_I. When read returns NIDEN_I.  <b>RAZWI</b> if NIDENSELDIS = 1.
1	RAZW1S	0x00	DBGEN_SEL_SET	Set Active High Debug Enable Selector. Write HIGH to set DBGEN_SEL.  <b>RAZWI</b> if DBGENSELDIS = 1.
0	RW1S	0x00	DBGEN_I_SET	Set internal version of Active High Debug Enable. Write HIGH to set DBGEN_I. When read returns DBGEN_I.  <b>RAZWI</b> if DBGENSELDIS = 1.

**Table 6-53: SECDBGCLR Register**

Bits	Type	Default	Name	Description
31:18	RAZWI	0x0000	-	Reserved.
17	RAZWI	0x00	-	Reserved.
16	RAZWI	0x00	-	Reserved.
15	RAZWI	0x00	-	Reserved.
14	RAZWI	0x00	-	Reserved.
13	RAZWI	0x00	-	Reserved.
12	RAZWI	0x00	-	Reserved.

Bits	Type	Default	Name	Description
11	RAZW1C	0x00	DAPDSSACCEN_SEL_CLR	<p>Clears Active High DAP to Debug Subsystem Access Enable Selector. Write HIGH to clear DAPDSSACCEN_SEL.</p> <p>This register is always RAZ. It is WI, if DAPDSSACCENSELDIS = 1.</p>
10	RAZW1C	0x00	DAPDSSACCEN_I_CLR	<p>Clears internal version of Active High DAP to Debug Subsystem Access Enable. Write HIGH to clear DAPDSSACCEN_I.</p> <p>This register is always RAZ. It is WI, if DAPDSSACCENSELDIS = 1.</p>
9	RAZW1C	0x00	DAPACCEN_SEL_CLR	<p>Clears Active High DAP Access Enable Selector. Write HIGH to clear DAPACCEN_SEL.</p> <p>This register is always RAZ. It is WI, if DAPACCENSELDIS = 1.</p>
8	RAZW1C	0x00	DAPACCEN_I_CLR	<p>Clears internal version of Active High DAP Access Enable. Write HIGH to clear DAPACCEN_I.</p> <p>This register is always RAZ. It is WI, if DAPACCENSELDIS = 1.</p>
7	RAZW1C	0x00	SPNIDEN_SEL_CLR	<p>Clears Active High Secure Privilege Non-Invasive Debug Enable Selector. Write HIGH to clear SPNIDEN_SEL.</p> <p>This register is always RAZ. It is WI, if SPNIDENSELDIS = 1.</p>
6	RAZW1C	0x00	SPNIDEN_I_CLR	<p>Clears internal version of Active High Secure Privilege Non-Invasive Debug Enable. Write HIGH to clear SPNIDEN_I.</p> <p>This register is always RAZ. It is WI, if SPNIDENSELDIS = 1.</p>
5	RAZW1C	0x00	SPIDEN_SEL_CLR	<p>Clears Active High Secure Privilege Invasive Debug Enable Selector. Write HIGH to clear SPIDEN_SEL.</p> <p>This register is always RAZ. It is WI, if SPIDENSELDIS = 1.</p>
4	RAZW1C	0x00	SPIDEN_I_CLR	<p>Clears internal version of Active High Secure Privilege Invasive Debug Enable. Write HIGH to clear SPIDEN_I.</p> <p>This register is always RAZ. It is WI, if SPIDENSELDIS = 1.</p>
3	RAZW1C	0x00	NIDEN_SEL_CLR	<p>Clears Active High Non-Invasive Debug Enable Selector. Write HIGH to clear NIDEN_SEL.</p> <p>This register is always RAZ. It is WI, if NIDENSELDIS = 1.</p>
2	RAZW1C	0x00	NIDEN_I_CLR	<p>Clears internal version of Active High Non-Invasive Debug Enable. Write HIGH to clear NIDEN_I.</p> <p>This register is always RAZ. It is WI, if NIDENSELDIS = 1.</p>
1	RAZW1C	0x00	DBGEN_SEL_CLR	<p>Clears Active High Debug Enable Selector. Write HIGH to clear DBGEN_SEL.</p> <p>This register is always RAZ. It is WI, if DBGENSELDIS = 1.</p>
0	RAZW1C	0x00	DBGEN_I_CLR	<p>Clears internal version of Active High Debug Enable. Write HIGH to clear DBGEN_I.</p> <p>This register is always RAZ. It is WI, if DBGENSELDIS = 1.</p>



### 6.6.2.2 SCSECCTRL

The System Control Security Control provides a register bit to set the Secure Configuration lock of this register block.

These registers are reset by **nCOLDRESETAON**.

These registers reside in the PD\_AON power domain.

**Table 6-54: SCSECCTRL Register**

Bits	Type	Default	Name	Description
31:3	RAZWI	0x00000000	-	Reserved
2	RW1S	0x00	SCSECCFGLOCK	Active High control to disable writes to Security related control registers SECDBGSET and SECDBGCLR. Once set to HIGH, it can no longer be cleared to zero except through Cold Reset.
1:0	RAZWI	0x00	-	Reserved

### 6.6.2.3 CLK\_CFG0, CLK\_CFG1 and CLK\_CFG2

The CLK\_CFG0, CLK\_CFG1, and CLK\_CFG2 registers provide control register fields to drive expansion clock generation logic that drives clock for this subsystem.

Each clock control handshake comprises of a configuration request register, that is CLKCFG, and a status register, that is CLKCFGSTATUS that acts as an acknowledgment. After writing to each CLKCFG field, the software has to poll the associated CLKCFGSTATUS field until the CLKCFGSTATUS is the same as the CLKCFG before doing any other operations.

In addition, if it is the first write to the register after Cold Reset, software has to poll that the targeted CLKCFG field value (default value set by the related configuration input) matches its associated CLKCFGSTATUS field value before the write occurs. The actual number of bits being implemented in the related CLKCFG and CLKCFGSTATUS signals is defined below.

These registers are reset by **nCOLDRESETAON**.

These registers reside in the PD\_AON power domain.

**Table 6-55: CLK\_CFG0 Register**

Bits	Type	Default	Name	Description
31:28	RAZWI	0x00	-	Reserved
27:24	RAZWI	0x00	-	Reserved
23:20	RAZWI	0x00	-	Reserved
19:16	RO	CPU0CLKCFGSTATUS	CPU0CLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for <b>CPU0CLK</b> .
15:12	RAZWI	0x00	-	Reserved
11:8	RAZWI	0x00	-	Reserved
7:4	RAZWI	0x00	-	Reserved

Bits	Type	Default	Name	Description
3:0	RW	CPU0CLKCFGRST	CPU0CLKCFG	Clock Configuration value that drives <b>CPU0CLKCFG</b> signals.

**Table 6-56: CLK\_CFG1 Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x000	-	Reserved
23:20	RO	AONCLKCFGSTATUS	AONCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for <b>AONCLK</b> .
19:16	RO	SYSCLKCFGSTATUS	SYSCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for <b>SYSCLK</b> .
15:8	RAZWI	0x000	-	Reserved
7:4	RW	AONCLKCFGRST	AONCLKCFG	Clock Configuration value that drives <b>AONCLKCFG</b> signals.
3:0	RW	SYSCLKCFGRST	SYSCLKCFG	Clock Configuration value that drives <b>SYSCLKCFG</b> signals.

**Table 6-57: CLK\_CFG2 Register**

Bits	Type	Default	Name	Description
31:20	RAZWI	0x000	-	Reserved
19:16	RO	NPU0CLKCFGSTATUS	NPU0CLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for <b>NPU0CLK</b> .  <b>RAZWI</b> if NUMNPU < 1
15:4	RAZWI	0x000	-	Reserved
3:0	RW	NPU0CLKCFGRST	NPU0CLKCFG	Clock Configuration value that drives <b>NPU0CLKCFG</b> signals. <b>RAZWI</b> if NUMNPU < 1  <b>RAZWI</b> if NUMNPU < 1

#### 6.6.2.4 CLOCK\_FORCE

The Clock Force register allows software to override dynamic clock gating that might be implemented in the system and keep each clock running.

Bits 0 to 8 are clock forces that do not apply to clock gates within the design that are responsible for the gating of clocks when gating power to a power gated region. Instead, it is applied to hierarchical dynamic clock gating within the system, with one bit for each power domain. Forcing a clock ON can reduce the latency that is a result of dynamic clock control, but generally has a reverse side-effect of increasing the dynamic power consumption of the system.



Note

All clock force default values of these bits are set to HIGH at reset. This allows the system to boot in case any hierarchical dynamic clock control implementation is non-functional. The associated clock force register bit must be cleared to enable hierarchical dynamic clock control for each power domain.

Bits 16 to 23 are clock forces used to force the external clock generator to continue to generate its clock. This can be used to avoid clock generators like PLLs from turning off, which can result in a long wakeup time.

These registers are reset by **nCOLDRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-58: CLOCK\_FORCE Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x0000	-	Reserved.
23	RAZWI	0x0	NPU0CLK_FORCE	Set HIGH to request the input NPU0CLK source to stay ON. The field is reserved and <b>RAZWI</b> if NUMNPU < 1.
22	RAZWI	0x0	-	Reserved.
21	RAZWI	0x0	-	Reserved.
20	RAZWI	0x0	-	Reserved.
19	RW	0x0	CPU0CLK_FORCE	Set HIGH to request the input CPU0CLK source to stay ON.
18	RW	0x0	-	Reserved for DEBUGCLK_FORCE.
17	RW	0x0	SYSCLK_FORCE	Set HIGH to request the input SYSCLK source to stay ON.
16	RW	0x0	AONCLK_FORCE	Set HIGH to request the input AONCLK source to stay ON.
15:12	RAZWI	0x00000000	-	Reserved.
11	RAZWI	0x0	-	Reserved.
10	RAZWI	0x0	-	Reserved.
9	RAZWI	0x0	-	Reserved.
8	RW	0x01	NPU0_CLKFORCE	Set HIGH to force PD_NPU0 Local clocks to run. The field is reserved and <b>RAZWI</b> if NUMNPU < 1.
7	RO	0x00	-	Reserved.
6	RO	0x00	-	Reserved.
5	RO	0x00	-	Reserved.
4	RW	0x01	CPU0_CLKFORCE	Set HIGH to force all clocks in PD_CPU0 to run.
3	RO	0x00	-	Reserved.
2	RW	0x01	DEBUG_CLKFORCE	Set HIGH to force all clocks in PD_DEBUG to run.
1	RW	0x01	SYS_CLKFORCE	Set HIGH to force all clocks in PD_SYS to run.
0	RW	0x01	MGMT_CLKFORCE	Set HIGH to force all clocks in PD_MGMT domain to run.

### 6.6.2.5 RESET\_SYNDROME

This register stores the reason for the last Reset event. All fields of this register except for PoR are cleared by the **nPORESETAON** input. Each field can be cleared by the software writing zero to the related bit. Writing HIGH to a bit results in that bit write value to be ignored and the bit

maintaining its previous value. If this register is not cleared after starting from a reset event, on another reset event the RESET\_SYNDROME might no longer accurately reflect the last reset event.



Note

CPU0LOCKUP does not actually generate reset, but when HIGH, it indicates that a CPU has locked-up and can be a precursor to another reset event. For example, watchdog timer reset request.

CPU0LOCKUP events are always stored, and only reset requests that are not masked by their respective mask bits are stored in the register.

This register resides in the PD\_AON power domain.

**Table 6-59: RESET\_SYNDROME Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x00000	-	Reserved
15	RAZWI	0x00	-	Reserved.
14	RAZWI	0x00	-	Reserved.
13	RAZWI	0x00	-	Reserved.
12	RWOC	0x00	CPU0LOCKUP	CPU0 Lockup Status
11	RAZWI	0x00	-	Reserved.
10	RAZWI	0x00	-	Reserved.
9	RAZWI	0x00	-	Reserved.
8	RWOC	0x00	CPU0RSTREQ	CPU0 Warm Reset Request
7	RWOC	0x00	HOSTRESETREQ	Host Level Cold Reset Request Input
6	RAZWI	0x00	-	Reserved.
5	RWOC	0x00	SWRESETREQ	Software Cold Reset Request
4	RWOC	0x00	RESETREQ	Subsystem Hardware Cold Reset Request Input
3	RWOC	0x00	SLOWCLKWDRSTREQ	SLOWCLK Watchdog Cold Reset Request
2	RWOC	0x00	SWDRSTREQ	Secure Watchdog Cold Reset Request
1	RWOC	0x00	NSWDRSTREQ	Non-secure Watchdog Cold Reset Request
0	RWOC	0x01	PoR	Power-On-Reset

### 6.6.2.6 RESET\_MASK

The RESET\_MASK register allows the software to control which reset sources are merged to generate the system wide warm reset, **nWARMRESETAON** or the **nCOLDRESETAON** signal. Set each bit to HIGH to enable each source. This register is reset by the **nWARMRESETAON**.



Note

If cleared, each of these mask bits prevents the reset source from generating the reset, and also prevents the associated RESET\_SYNDROME register bit from recording the event.

This register resides in the PD\_AON power domain.

This register is reset by the **nWARMRESETAON**.

**Table 6-60: RESET\_MASK Register**

Bits	Type	Default	Name	Description
31:12	RAZWI	0x000000	-	Reserved
11	RAZWI	0x00	-	Reserved
10	RAZWI	0x00	-	Reserved
9	RAZWI	0x00	-	Reserved
8	RW	CPU0RSTREQENRST	CPU0RSTREQEN	CPU0 Warm Reset Request Enable
7:2	RAZWI	0x000	-	Reserved
1	RW	0x00	NSWDRSTREQEN	Non-secure Watchdog Reset Enable
0	RAZWI	0x00	-	Reserved

### 6.6.2.7 SWRESET

The SWRESET register allows the software to request for a Cold Reset. To request for a Cold Reset, write '1' to the register. The register always returns zeros.

This register resides in the PD\_AON power domain.

**Table 6-61: SWRESET Register**

Bits	Type	Default	Name	Description
31:2	RAZWI	0x00000000	-	Reserved
9	RAZW1S	0x00	SWRESETREQ	Software Reset Request. Write '1' to request a Cold Reset.
8:0	RAZWI	0x000	-	Reserved

### 6.6.2.8 GRETREG

The General Purpose Retention Register provides 16 bits of retention register for general storage through the HIBERANTION0 System Power States. This register is reset by **nCOLDRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-62: GRETREG Register**

Bits	Type	Default	Name	Description
31:16	RAZWI	0x00000	-	Reserved
15:0	RW	0x00000	GRETREG	General Purpose Retention Register

### 6.6.2.9 INITSVTOR0

This register defines the CPU0 Initial Secure Vector table offset (VTOR\_S.TBLOFF[31:7]) out of reset.

This register is reset by **nWARMRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-63: INITSVTOR0 Register**

Bits	Type	Default	Name	Description
31:7	RW	INITSVTOR0RST[31:7]	INITSVTOR0	Default Secure Vector table offset at reset for CPU0.
6:1	RAZWI	0x000	-	Reserved
0	RW1S	0x00	INITSVTOR0LOCK	Lock INITSVTOR0. When set to '1', stops any further writes to INITSVTOR0 and INITSVTOR0LOCK fields. Cleared only by warm reset.

### 6.6.2.10 CPUWAIT

This Register provides controls to force CPU0 to wait after reset rather than boot immediately. This allows another entity in the expansion system or the debugger to access the system prior to the CPU booting.

This register is reset by **nCOLDRESETAON** only.

This register resides in the PD\_AON power domain.

**Table 6-64: CPUWAIT Register**

Bits	Type	Default	Name	Description
31:4	RAZWI	0x00000000	-	Reserved
3	RAZWI	0x00	-	Reserved.
2	RAZWI	0x00	-	Reserved.
1	RAZWI	0x00	-	Reserved.
0	RW	CPU0WAITRST	CPU0WAIT	<p>CPU 0 waits at boot.</p> <ul style="list-style-type: none"> <li>'0': boot normally.</li> <li>'1': wait at boot.</li> </ul> <p>When <b>CPU0WAITCLR</b> input is 1'b1, this bit is cleared.</p>

### 6.6.2.11 NMI\_ENABLE

This register provides controls to enable or disable the internally or externally generated Non-Maskable Interrupt sources from generating an NMI interrupt on CPU0.

This register is reset by **nWARMRESETAON** and its reset value is defined by the configuration options.

This register resides in the PD\_AON power domain.

**Table 6-65: NMI\_ENABLE Register**

Bits	Type	Default	Name	Description
31:20	RAZWI	0x0000	-	Reserved.
19	RAZWI	0x00	-	Reserved.
18	RAZWI	0x00	-	Reserved.
17	RAZWI	0x00	-	Reserved.
16	RW	CPU0EXPNMIENABLERST	CPU0_EXPNMI_ENABLE	CPU0 Externally Sourced NMI Enable. This determines if the input, <b>CPU0EXPNMI</b> , can raise NMI interrupt on CPU 0: <ul style="list-style-type: none"> <li>HIGH, allowed.</li> <li>LOW, is masked and not allowed.</li> </ul>
15:4	RAZWI	0x0000	-	Reserved.
3	RAZWI	0x00	-	Reserved.
2	RAZWI	0x00	-	Reserved.
1	RAZWI	0x00	-	Reserved.
0	RW	CPU0INTNMIENABLERST	CPU0_INTNMI_ENABLE	CPU0 Internally Sourced NMI Enable. This determines if the subsystem internally generated NMI interrupt sources can raise NMI interrupt on CPU0: <ul style="list-style-type: none"> <li>HIGH, allowed.</li> <li>LOW, is masked and not allowed.</li> </ul>

### 6.6.2.12 PPUINTSTAT

This is the PPU interrupt status register for SSE-310

This register resides in the PD\_AON power domain.

This register is reset by **nWARMRESETAON**.

**Table 6-66: PPUINTSTAT register bit descriptions**

Bits	Type	Default	Name	Description
31:12	RAZWI	0x0000_000	-	Reserved
11	RO	0x0	DEBUG_PPU_INTSTAT	Debug PPU Interrupt
10	RAZWI	0x0	-	Reserved
9	RAZWI	0x0	-	Reserved
8	RAZWI	0x0	-	Reserved
7	RO	0x0	NPU0_PPU_INTSTAT	If NUMNPU<1, NPU0 PPU interrupt is <b>RAZWI</b>
6	RAZWI	0x0	-	Reserved
5	RAZWI	0x0	-	Reserved
4	RAZWI	0x0	-	Reserved
3	RAZWI	0x0	-	Reserved
2	RO	0x0	CPU0_PPU_INTSTAT	CPU0 PPU interrupt

Bits	Type	Default	Name	Description
1	RO	0x0	SYS_PPU_INTSTAT	System PPU interrupt
0	RO	0x0	MGMT_PPU_INTSTAT	Management PPU Interrupt

### 6.6.2.13 PWRCTRL

The Power Control register configures the power control features in SSE-310 System. This register is reset by **nWARMRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-67: PWRCTRL Register**

Bits	Type	Default	Name	Description
31:2	RAZWI	0x00000000	-	Reserved.
1	RWOC	0x01	PPU_ACCESS_UNLOCK	PPU_ACCESS_FILTER write unlock. <ul style="list-style-type: none"> <li>When set to '1': Both PPU_ACCESS_FILTER and this register bits can be written.</li> <li>When set to '0': The PPU_ACCESS_FILTER and this register bit is no longer writable, and PPU_ACCESS_UNLOCK stays '0'.</li> </ul>
0	RW	0x01	PPU_ACCESS_FILTER	Filter Access to PPU Registers. <ul style="list-style-type: none"> <li>When set to '1': Only key PPU interrupt handling registers are open to write access, and all other PPU registers are read only.</li> <li>When set to '0', it releases all PPU register to full access.</li> </ul> For more information related to PPU registers accessibility, see <a href="#">Power Policy Units</a> .

### 6.6.2.14 PDCM\_PD\_SYS\_SENSE

The Power Dependency Control Matrix System power domain (PD\_SYS) Sensitivity register defines what keeps awake the PD\_SYS power domain and the minimum power state to use when the domain is in its low power state. This register is reset by **nWARMRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-68: PDCM\_PD\_SYS\_SENSE register**

Bits	Type	Default	Name	Description
31:30	RW	0x00	MIN_PWR_STATE	Defines the Minimum Power State when PD_SYS is trying to enter a lower power state: <ul style="list-style-type: none"> <li>00 - Minimum power state is OFF</li> <li>01 - Minimum power state is Retention</li> <li>10 - Minimum power state is ON</li> <li>Others - Reserved</li> </ul>
29:24	RAZWI	0x00	-	Reserved
23	RW	0x00	S_PDCMRETQREQ3	Enable sensitivity to <b>PDCMQRETREQn[3]</b> signal. If set to '1' PD_SYS stays in ON or RET if <b>PDCMRETQREQn[3]</b> signal is HIGH.



Bits	Type	Default	Name	Description
22	RW	0x00	S_PDCMRETQREQ2	Enable sensitivity to <b>PDCMQRETREQn[2]</b> signal. If set to '1' PD_SYS stays in ON or RET if <b>PDCMRETQREQn[2]</b> signal is HIGH
21	RW	0x00	S_PDCMRETQREQ1	Enable sensitivity to <b>PDCMQRETREQn[1]</b> signal. If set to '1' PD_SYS stays in ON or RET if <b>PDCMRETQREQn[1]</b> signal is HIGH.
20	RW	0x00	S_PDCMRETQREQ0	Enable sensitivity to <b>PDCMQRETREQn[0]</b> signal. If set to '1' PD_SYS stays in ON or RET if <b>PDCMRETQREQn[0]</b> signal is HIGH
19	RW	0x00	S_PDCMONQREQ3	Enables sensitivity to <b>PDCMONQREQn[3]</b> signal. If set to '1', PD_SYS stays ON if <b>PDCMONQREQn[3]</b> signal is HIGH.
18	RW	0x00	S_PDCMONQREQ2	Enables sensitivity to <b>PDCMONQREQn[2]</b> signal. If set to '1', PD_SYS stays ON if <b>PDCMONQREQn[2]</b> signal is HIGH.
17	RW	0x00	S_PDCMONQREQ1	Enables sensitivity to <b>PDCMONQREQn[1]</b> signal. If set to '1', PD_SYS stays ON if <b>PDCMONQREQn[1]</b> signal is HIGH.
16	RW	0x00	S_PDCMONQREQ0	Enables sensitivity to <b>PDCMONQREQn[0]</b> signal. If set to '1', PD_SYS stays ON if <b>PDCMONQREQn[0]</b> signal is HIGH.
15	<b>RAZWI</b>	0x00	-	Reserved
14	<b>RAZWI</b>	0x00	-	Reserved
13	RO	0x00	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	<b>RAZWI</b>	0x00	-	Reserved.
11:9	<b>RAZWI</b>	0x000	-	Reserved.
8	<b>RAZWI</b>	0x01	-	Reserved.
7	<b>RAZWI</b>	0x01	-	Reserved.
6	<b>RAZWI</b>	0x01	-	Reserved.
5	RO	0x01	S_PD_NPU0_ON	Tied to HIGH. PD_SYS will always try to stay ON if PD_NPU0 is ON. This bit is Reserved and <b>RAZWI</b> if NUMNPU is 0.
4	<b>RAZWI</b>	0x00	-	Reserved.
3	<b>RAZWI</b>	0x00	-	Reserved.
2	<b>RAZWI</b>	0x00	-	Reserved.
1	RO	0x01	S_PD_CPU0_ON	Tied to HIGH. PD_SYS will always try to stay ON if PD_CPU0 is ON.
0	RW	0x00	S_PD_SYS_ON	Enable PD_SYS ON Sensitivity. Set this to HIGH to keep PD_SYS awake once powered ON.

### 6.6.2.15 PDCM\_PD\_CPU0\_SENSE

The Power Dependency Control Matrix System Power domain (PD\_CPU0) Sensitivity register defines what keeps awake the PD\_CPU0 domain, and defines the minimum power state to use when the domain is in its low power state.

This register is reset by **nWARMRESETAON**.

In SSE-310, PD\_CPU0 is not sensitive to any incoming dependencies.

This register resides in the PD\_AON power domain.

**Table 6-69: PDCM\_PD\_CPU0\_SENSE Register**

Bits	Type	Default	Name	Description
31:24	RAZWI	0x000	-	Reserved.
23	RO	0x00	PDCMRETQREQ3	Tied to LOW. Ignores <b>PDCMRETQREQn[3]</b> signal.
22	RO	0x00	PDCMRETQREQ2	Tied to LOW. Ignores <b>PDCMRETQREQn[2]</b> signal.
21	RO	0x00	PDCMRETQREQ1	Tied to LOW. Ignores <b>PDCMRETQREQn[1]</b> signal.
20	RO	0x00	PDCMRETQREQ0	Tied to LOW. Ignores <b>PDCMRETQREQn[0]</b> signal.
19	RO	0x00	PDCMONQREQ3	Tied to LOW. Ignores <b>PDCMONQREQn[3]</b> signal.
18	RO	0x00	PDCMONQREQ2	Tied to LOW. Ignores <b>PDCMONQREQn[2]</b> signal.
17	RO	0x00	PDCMONQREQ1	Tied to LOW. Ignores <b>PDCMONQREQn[1]</b> signal.
16	RO	0x00	PDCMONQREQ0	Tied to LOW. Ignores <b>PDCMONQREQn[0]</b> signal.
15	RAZWI	0x00	-	Reserved
14	RAZWI	0x00	-	Reserved
13	RO	0x00	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	RAZWI	0x00	-	Reserved
11:5	RAZWI	0x000	-	Reserved
4	RAZWI	0x00	-	Reserved
3	RAZWI	0x00	-	Reserved
2	RAZWI	0x00	-	Reserved
1	RO	0x00	S_PD_CPU0_ON	Tied to LOW. Ignores PD_CPU0 power state.
0	RO	0x00	S_PD_SYS_ON	Tied to LOW. Ignores PD_SYS power state.

#### 6.6.2.16 PDCM\_PD\_VMR<M>\_SENSE

The Power Dependency Control Matrix Volatile Memory Region <M> power domain (PD\_VMR<M>) sensitivity register defines what keeps awake the PD\_VMR<M> domain and the minimum power state to use when the domain is in its low power state, where M is 0 to NUMVMBANK-1. This register is reset by **nWARMRESETAON**.

This register resides in the PD\_AON power domain.

**Table 6-70: PDCM\_PD\_VMR<M>\_SENSE register**

Bits	Type	Default	Name	Description
31:30	RW	0x01	MIN_PWR_STATE	Defines the Minimum Power State when PD_VMR<M> is trying to enter a lower power state: <ul style="list-style-type: none"> <li>00 - Minimum power state is OFF</li> <li>01 - Minimum power state is Retention</li> <li>10 - Minimum power state is ON.</li> <li>Others - Reserved</li> </ul>
29:24	RAZWI	0x00	-	Reserved
23	RW	0x00	S_PDCMRETQREQ3	Enable sensitivity to <b>PDCMRETQREQn[3]</b> signal. If set to '1' PD_VMR<m> stays in ON or RET if <b>PDCMRETQREQn[3]</b> signal is HIGH.

Bits	Type	Default	Name	Description
22	RW	0x00	S_PDCMRETQREQ2	Enable sensitivity to <b>PDCMRETQREQn[2]</b> signal. If set to '1' PD_VMR<m> stays in ON or RET if <b>PDCMRETQREQn[2]</b> signal is HIGH.
21	RW	0x00	S_PDCMRETQREQ1	Enable sensitivity to <b>PDCMRETQREQn[1]</b> signal. If set to '1' PD_VMR<m> stays in ON or RET if <b>PDCMRETQREQn[1]</b> signal is HIGH.
20	RW	0x00	S_PDCMRETQREQ0	Enable sensitivity to <b>PDCMRETQREQn[0]</b> signal. If set to '1' PD_VMR<m> stays in ON or RET if <b>PDCMRETQREQn[0]</b> signal is HIGH.
19	RW	0x00	S_PDCMQREQ3	Enable sensitivity to <b>PDCMQREQn[3]</b> signal. If set to '1' PD_SYS stays ON if <b>PDCMQREQn[3]</b> signal is HIGH.
18	RW	0x00	S_PDCMQREQ2	Enable sensitivity to <b>PDCMQREQn[2]</b> signal. If set to '1' PD_SYS stays ON if <b>PDCMQREQn[2]</b> signal is HIGH.
17	RW	0x00	S_PDCMQREQ1	Enable sensitivity to <b>PDCMQREQn[1]</b> signal. If set to '1' PD_SYS stays ON if <b>PDCMQREQn[1]</b> signal is HIGH.
16	RW	0x00	S_PDCMQREQ0	Enable sensitivity to <b>PDCMQREQn[0]</b> signal. If set to '1' PD_SYS stays ON if <b>PDCMQREQn[0]</b> signal is HIGH.
15	<b>RAZWI</b>	0x00	-	Reserved.
14	<b>RAZWI</b>	0x00	-	Reserved.
13	RO	0x00	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	<b>RAZWI</b>	0x00	-	Reserved
11:9	<b>RAZWI</b>	0x000	-	Reserved
8	<b>RAZWI</b>	0x00	-	Reserved
7	<b>RAZWI</b>	0x00	-	Reserved
6	<b>RAZWI</b>	0x00	-	Reserved
5	RW	0x00	S_PD_NPU0_ON	Enable PD_NPU0 sensitivity. If set to '1' PD_VMR<M> will stay ON if PD_NPU0 is ON. This bit is Reserved and <b>RAZWI</b> if NUMNPU < 1.
4	<b>RAZWI</b>	0x00	-	Reserved
3	<b>RAZWI</b>	0x00	-	Reserved
2	<b>RAZWI</b>	0x00	-	Reserved
1	RW	0x00	S_PD_CPU0_ON	Enable PD_CPU0 sensitivity. If set to 1 PD_VMR<M> will stay on if PD_CPU0 is on.
0	RO	0x00	S_PD_SYS_ON	Tied to LOW. Ignores PD_SYS power state.

## 6.7 CPU Private Peripheral Bus region

As defined by the ARMv8-M architecture specification, the processor hosts a local Private Peripheral Bus (PPB) region at address 0xE0000000 to 0xE00FFFFF. This region is for integration with the CoreSight debug and trace components that are normally local to CPU0 and is not intended for general peripheral usage.

In SSE-310, this region has the memory map as shown in the table below.



Other than the EWIC, CPU0 ROM Table, and the peripherals on the External PPB Expansion that are added in the subsystem expansion, the existence of all other

components depends on the CPU implementation and configuration. Refer to *Arm® Cortex®-M85 Processor Technical Reference Manual*.

Depending on EXPLOGIC\_PRESENT:

- EXPLOGIC\_PRESENT = 1, the Cortex-M85 TPIU-M and the MCU debug ROM table are integrated on the CPU0 EPPB interface in the subsystem expansion and the address range reserved for ETB is **RAZWI**. The MCU debug ROM table is pointing to the TPIU-M and the CPU's internal ROM table. These are integrated at the following addresses:
  - TPIU at 0xE0040000
  - ETB at 0xE0045000
  - MCU debug ROM table at {CPU0MCUROMADDR, 0x000} to where the DAP-Lite2 is pointing. Default address for MCU debug ROM table is 0xE00FE000.
- EXPLOGIC\_PRESENT = 0, the TPIU, ETB, and MCU debug ROM table are not integrated and the these regions are provided as a PPB expansion

**Table 6-71: CPU0 Private Peripheral Bus Region**

Row ID	Address from	Address to	Size	Region Name	Description
1	0xE0040000	0xE0040FFF	4KB	SSE-310 TPIU/ PPB Expansion	SSE-310 TPIU when EXPLOGIC_PRESENT = 1  CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
2	0xE0041000	0xE0041FFF	4KB	ETM <sup>1</sup>	Embedded Trace Module
3	0xE0042000	0xE0042FFF	4KB	CTI <sup>1</sup>	Cross Trigger Interface
4	0xE0043000	0xE0044FFF	8KB	Reserved	Reserved
5	0xE0045000	0xE0045FFF	4KB	ETB/PPB Expansion	Reserved for ETB ( <b>RAZWI</b> ) when EXPLOGIC_PRESENT = 1  CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
6	0xE0046000	0xE0046FFF	4KB	PMC <sup>1</sup>	Programmable MBIST Controller
7	0xE0047000	0xE0047FFF	4KB	EWIC	External Wakeup Interrupt Controller. For more information, see <a href="#">EWIC</a> .
8	0xE0048000	0xE0048FFF	4KB	PPB Expansion	CPU0 Expansion PPB Interface. Reserved for Software Based Build In Self Test
9	0xE0049000	0xE00FEFFF	24KB	MCU debug ROM table/PPB Expansion	CPU0 Expansion PPB Interface. Includes MCU debug ROM table at {CPU0MCUROMADDR, 0x000} when EXPLOGIC_PRESENT = 1
10	0xE00FF000	0xE00FFFFF	4KB	ROM Table	CPU0 ROM Table

1. The existence of these components depends on the configuration and the supported features of the integrated processor. If they do not exist, these regions are reserved, and return an error response when accessed.

## 6.7.1 EWIC

SSE-310 is designed to always support an External Wakeup Interrupt Controller (EWIC) for CPU0. This allows the system to support CPU0 being switched off, and for the EWIC to run at a lower clock rate to help reduce PD\_AON dynamic and leakage power consumption.

The EWIC resides in the PD\_AON power domain, run on AONCLK, and resides in the nWARMRESETAON reset domain.

The EWIC is only accessible through the Private Peripheral Bus Region of CPU0 at address 0xE0047000 to 0xE0047FFF.



- EVENTS[0] input of the EWIC is not connected in SSE-310 but tied LOW. WFE wakeup events are not supported to wake-up SSE-310.
- When modifying the EWIC\_ASCR.ASPU and EWIC\_ACSR.ASPD, the software must issue a DSB and an ISB instruction before performing any other action.

For more information about Cortex-M85 EWIC registers, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

## 6.7.2 Cortex-M85 TPIU-M registers

For information about Cortex-M85 TPIU-M registers, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

## 6.8 Debug System Access Region

SSE-310 does not support the CoreSight SoC-600 based common debug infrastructure.

As the debug system access region is not used and therefore the following regions are reserved:

- 0xE0100000 to 0xE01FFFFFF
- 0xF0100000 to 0xF01FFFFFF

When accessed, these regions return bus error response.

## 6.9 Peripheral Expansion Region

When EXPLOGIC\_PRESENT = 1, a system timestamp generator (System Counter) is integrated in SSE-310 expansion and drives the system timestamp interface.

For more details, see [System Timestamp Interface](#).

The System Counter resides in the PD\_AON Power domain, is clocked by CNTCLK and is reset by **nWARMRESETAON**.

The following table shows the Peripheral Expansion Region address map when  
EXPLOGIC\_PRESENT = 1

**Table 6-72: Peripheral Expansion Region**

Row ID	Address from	Address to	Size	Region name	Description	Alias with row ID	Security <sup>1</sup>
-	0x48100000	0x48100FFF	4KB	Reserved	Reserved ( <b>RAZWI</b> )	-	-
1	0x48101000	0x48101FFF	4KB	-	System Counter (System Timestamp Generator) Status Frame register. See <a href="#">System Timestamp Interface</a> .	3	NS
2	0x58100000	0x58100FFF	4KB	-	System Counter (System Timestamp Generator) Control Frame register. See <a href="#">System Timestamp Interface</a> .	-	S
3	0x58101000	0x58101FFF	4KB	-	System Counter (System Timestamp Generator) Status Frame register. See <a href="#">System Timestamp Interface</a> .	1	S

<sup>1</sup> Legend

- S: Secure access only.
- NS: Non-secure access only.

For details of the ARMv8M System Counter registers, see *Arm® Corstone™ Reference Systems Architecture Specification Ma1*.

# Appendix A Configurable render options

SSE-310 provides several render configuration options.

The configuration options are provided in the following format:

## CONFIGURATION OPTION FORMAT

- Legal range:

The legal range represents the Arm recommended ranges of the render configuration. Setting these configuration options out of the legal range might lead to the failure of the rendering, compilation, elaboration, or the Out of Box test.

To avoid illegal configurations, an RTL elaboration or Render configuration error occurs when a configuration or parameter value is set outside the Legal range. If you intentionally set a value outside the Legal range, then you can disable this error checking mechanism by setting the following option:

```
PERFORM_CONFIGCHECK: 0
```

- Description:

This section contains the detailed description and the value enumerations of the render configuration.

## A.1 Global configuration options

SSE-310 provides several Global configuration render options.

### CONFIG\_NAME

- Any string limited to 16 characters that obeys the following regular expression:

```
^[a-zA-Z0-9]{1,16}$
```

- Default value: Name of the subsystem configuration, it is expected to match the following regular expression and provided inside double quotes

```
^[a-zA-Z0-9]{1,16}$
```

The render process appends `_${CONFIG_NAME}` to the top-level Verilog file, folder, and each uniquified element. This option ensures that differently configured SSE-310 subsystems can be present in the same design without any naming conflicts when compiled to the same logical library. The CONFIG\_NAME is also feed in to the configuration of the configurable IPs instantiated in the subsystem.

## PILEVEL

- Legal range: 1
- Power Infrastructure Level. It defines the implemented power structure of the system:
  - 1 - Intermediate Power Structure

## NUMCPU

- Legal range: 0
- It describes the number of Arm Cortex-M CPU cores in the subsystem. The number of cores is equal to `NUMCPU + 1`.

## CPU0TYPE

- Legal range: 4
- It describes the type of CPU that is integrated:
  - 0 - Not implemented
  - 4 - Cortex-M85
  - Others - Reserved

## NUMNPU

- Legal range: 0,1
- It describes the number of NPU cores in the subsystem. The number of cores is equal to `NUMNPU`.

## NPU0TYPE

- Legal range: 0,1
- It describes the integrated NPU type for NPU0 if NPU0 exists:
  - 0 - Not present
  - 1 - Ethos-U55
  - Others - Reserved

## DEBUGLEVEL

- Legal range: 2
- It selects the debug level of the subsystem:
  - 0 - Debug System does not exist. There is no Debug Access and no Trace support.
  - 1 - Debug system exists without Trace support. Debug Access Interface(s) exists, but Trace is not supported.
  - 2 - Debug system exists with Trace support. Both Debug Access Interface(s) and Trace Interface exist.



## HASCSS

- Legal range: 0
- It defines whether the CoreSight SoC-600 based Debug infrastructure is included.
  - 0 - No
  - 1 - Yes

## NUMVMBANK

- Legal range: 2
- It selects the number of Volatile Memory Banks

## HASCRYPTO

- Legal range: 0
- It defines whether Crypto Accelerator is included:
  - 0 - No
  - 1 - Yes

## NUMDMA

- Legal range: 0
- Describes the number of DMA cores present in the system

## DMATYPE

- Legal range: 0
- It Describes the type of DMA that is integrated:
  - 0 - Not implemented.
  - 1 - ADA DMA

## A.2 Global CPU Configuration Options

SSE-310 provides several Global CPU configuration render options.

### SECEXT

- Legal range: 1
- Specifies the inclusion of the Security Extension. The options are:
  - 0 - Security Extension is excluded
  - 1 - Security Extension is included

Security Extensions are always included.

## BUSPROT\_PRESENT

- Legal range: 0
- Specifies whether interface protection is supported on the M-AXI, P-AHB, EPPB, and S-AXI interfaces on the processor.
  - 0 - Interface protection not included
  - 1 - Interface protection included

RAS is not supported by SSE-310.

## ECC\_PRESENT

- Legal range: 0
- Specifies whether the processor supports Error detection and correction in the L1 Data and Instruction cache (when configured) and the TCM.
  - 0 - ECC not included
  - 1 - ECC included

## INITTCMEN

- Legal range: 0b11
- TCM enable initialisation out of reset. Set to all ones to enable both TCMs Tightly Coupled Memory (TCM) enable initialization out of reset:
  - Bit[0] is HIGH: Instruction Tightly Coupled Memory (ITCM) is enabled
  - Bit[1] is HIGH: Data Tightly Coupled Memory (DTCM) is enabled

This signal controls the reset value of ITCMCR.EN and DTCMCR.EN bits. For more information on ITCMCR and DTCMCR, see the *Arm® Cortex®-M85 Processor Technical Reference Manual*.

In SSE-310, the NIC in the TCM address range responds with error on M-AXI of the CPU therefore after disabling the TCM any further accesses to this memory region results in error.

## INITPAHBEN

- Legal range: 1
- P-AHB enable initialization out of reset:
  - 0 - P-AHB is disabled
  - 1 - P-AHB is enabled

For more information on PAHBCCR, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

## LOCKDCAIC

- Legal range: 0,1
- Disable access to the instruction cache direct cache access registers DCAICLR and DCAICRR. Asserting this signal prevents direct access to the instruction cache Tag or Data RAM content.

This is required when using eXecutable Only Memory (XOM) on the M-AXI interface. When LOCKDCAIC is asserted:

- DCAICLR is **RAZ/WI**
- DCAICRR is RAZ

TCM XOM is not supported. If PMC-100 included in the CPU then M-AXI is not suitable for XOM Secure Privileged input.

## CFGBIGEND

- Legal range: 0
- Data endian format
  - 0 - Little-endian (LE)
  - 1 - Byte-invariant big-endian (BE8)

## CFGMEMALIAS

- Legal range: 0b10000
- Memory address alias bit for the ITCM, DTCM and P-AHB regions. The address bit used for the memory alias is determined by:
  - 0b00000 - No Alias
  - 0b00001 - Alias bit = 24
  - 0b00010 - Alias bit = 25
  - 0b00100 - Alias bit = 26
  - 0b01000 - Alias bit = 27
  - 0b10000 - Alias bit = 28

Setting CFGMEMALIAS to an invalid value results in UNPREDICTABLE behavior.

## A.3 CPU0 Element Configuration Options

SSE-310 provides several CPU0 Element Configuration render options.

### CPU0\_RAR

- Legal range: 1
- Specifies if the synchronous states or the architecturally required state are reset. The options are:
  - 0 - Only reset the architecturally required state
  - 1 - Reset all synchronous states



Setting CPU0\_RAR increases the size of the registers that are not reset by default. Setting CPU0\_RAR also increases the overall area of the implementation.

## CPU0\_LOCKSTEP

- Legal range: 0
- Specifies whether the processor is configured for dual-redundant lockstep operation. The options are:
  - 0 - Regular processor operation
  - 1 - Dual-redundant lockstep operation

## CPU0\_PACBTI

- Legal range: 0, 1
- Specifies inclusion of the Pointer Authentication and Branch Target Identification (PACBTI) extension. The options are:
  - 0 - PACBTI not included
  - 1 - PACBTI included

## CPU0\_INITNSVTOR\_ADDR\_INIT

Legal range:

- IDAU NS regions:
  - CPU0\_INITNSVTOR\_ADDR\_INIT[28] = 1'b0
  - CPU0\_INITNSVTOR\_ADDR\_INIT[6:0] = 0x0
- It indicates the Non-secure vector table offset address out of reset, VTOR\_NS.TBLOFF[31:7]. For more information on VTOR\_NS, see the *Arm®v8-M Architecture Reference Manual*.

## CPU0\_IRQLVL

- Legal range: 3-8
- Specifies the number of exception priority bits.

## CPU0EXPNUMIRQ

- Legal range: 2-448
- It specifies the number of expansion interrupt for CPU0.

## CPU0EXPIRQDIS [CPU0EXPNUMIRQ-1:0]

- Legal range: CPU0EXPNUMIRQ{1'b0} - CPU0EXPNUMIRQ{1'b1}
- It specifies for CPU0 whether each expansion interrupt bit is implemented or disabled. CPU0EXPIRQDIS[i] = 1'b1 indicates that IRQ[i+32] is not present on CPU0

## CPU0\_DBGLVL

- Legal range: 1, 2
- Specifies the number of debug resources included. The options are:
  - 0 - Minimal debug. No Halting debug or memory access.
  - 1 - Reduced set. Two Data Watchpoint and Trace (DWT) and four Breakpoint Unit (BPU) comparators.
  - 2 - Mid set. Four DWT and eight BPU comparators.

Debug Monitoring mode and the Unprivileged Debug Extension (UDE) is always supported. The Performance Monitoring Unit (PMU) is included when CPU0\_DBGLVL is nonzero.

## CPU0\_ITM\_PRESENT

- Legal range: 0,1
- Specifies the level of instrumentation trace supported. The options are:
  - 0 - Instrumentation Trace Macrocell (ITM) and DWT trace excluded
  - 1 - ITM and DWT trace included. If CPU0\_DBGLVL is set to 0, no trace is included in the processor.

This parameter is not passed directly to expansion element but through the Cortex-M85 configuration file. The TPIU logic is optimized in case no ITM trace present. No optimization when CoreSight SoC-600 TPIU-M is used.

## CPU0\_ETM\_PRESENT

- Legal range: 0,1
- Specifies support for Embedded Trace Macrocell (ETM) trace. The options are:
  - 0 - ETM trace is excluded
  - 1 - ETM is included



Note

The ETM unit must be licensed before you can set it to 1. If CPU0\_DBGLVL is set to 0, no trace is included in the processor. The Cortex-M85 TPIU-M logic is optimized in case no ETM trace present. No optimization when CoreSight SoC-600 TPIU-M is used.

---

## CPU0\_FPU\_PRESENT

- Legal range: 0,1
- The CPU0\_FPU\_PRESENT parameter determines the floating-point functionality of the processor.
  - 0 No floating-point functionality is included.
  - 1 Scalar half, single and double-precision floating-point included.



The floating-point functionality is separately licensable. See *Arm® Corstone™ SSE-310 Example Subsystem Release Note* for more information about the Floating Point Unit bundle name and availability.

## CPU0\_MVE\_CONFIG

- Legal range: CPU0\_FPU\_PRESENT ? (2|1|0) : 1
- M-profile Vector Extension (CPU0\_MVE\_CONFIG) parameter can have the following configuration options:
  - 0 - MVE not included
  - 1 - Integer subset of MVE included
  - 2 - Integer and half and single-precision floating-point MVE included. This option is only valid if [CPU0\\_FPU\\_PRESENT=1](#).

SSE-310 only supports the no M-profile Vector Extension when Floating Point Unit is configured.

- For more details about the effect of this parameter value on MVE, see *Arm® Cortex®-M85 Processor Integration and Implementation Manual*

## CPU0\_MPU\_S

- Legal range: 4,8,12,16
- Specifies the number of Secure MPU regions included. The options are:
  - 4 - 4 MPU regions
  - 8 - 8 MPU regions
  - 12 - 12 MPU regions
  - 16 - 16 MPU regions



- If SECEXT is set to 0, CPU0\_MPU\_S is ignored.
- If SECEXT is set to 1, then all Secure MPU regions can be disabled using the CPU0MPUSDISABLE signal.

## CPU0\_MPU\_NS

- Legal range: 4,8,12,16
- Specifies the number of Non-secure Memory Protection Unit (MPU) regions included. The options are:
  - 4 - 4 MPU regions
  - 8 - 8 MPU regions
  - 12 - 12 MPU regions
  - 16 - 16 MPU regions



If SECEXT is set to 0, then CPU0\_MPU\_NS indicates the total number of MPU regions included. CPU0MPUNSDISABLE disables all Non-secure MPU regions.

## CPU0\_SAUDISABLE

- Legal range: 0
- If the Security Attribution Unit (SAU) is configured, disables support

## CPU0\_NUM\_SAU\_CONFIG

- Legal range: 4,8
- Specifies the number of Security Attribution Unit (SAU) regions included. The options are:
  - 4 - 4 SAU regions
  - 8 - 8 SAU regions



- If SECEXT is set to 0, SAU is ignored.
- If SAU is set to 0, an external component uses the Implementation Defined Attribution Unit (IDAU) interface to specify memory regions. If SECEXT is set to 1, use CPU0\_SAUDISABLE to disable all SAU regions. The Security Extension is still implemented when CPU0\_SAUDISABLE is asserted.

## CPU0\_IDCACHEID

- Legal range: 0-255
- Instruction and Data cache ID. Unique identifier for instruction and data cache RAM implementation. This parameter allows multiple instances of the Cortex-M85 processor (though only one within the SSE-310) to use different library cells for the embedded cache RAM.

## CPU0\_INSTR\_CACHE\_SIZE

- Legal range:
  - CPU0\_INSTR\_CACHE\_SIZE[0] = 0b1
  - CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b0000, 0b0001, 0b0011, 0b0111, 0b1111
- Specifies the inclusion of the instruction cache controller in the processor.

If the instruction cache controller is included:

- CPU0\_INSTR\_CACHE\_SIZE - Specifies the size of the cache
- CPU0\_INSTR\_CACHE\_SIZE[0] = 0 - Instruction cache is excluded
- CPU0\_INSTR\_CACHE\_SIZE[0] = 1 - Instruction cache is included

If CPU0\_INSTR\_CACHE\_SIZE[0]=1, then the cache sizes are:

- CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b0000 - 4 KB instruction cache
- CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b0001 - 8 KB instruction cache
- CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b0011 - 16 KB instruction cache
- CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b0111 - 32 KB instruction cache
- CPU0\_INSTR\_CACHE\_SIZE[4:1] = 0b1111 - 64 KB instruction cache



Setting CPU0\_INSTR\_CACHE\_SIZE to any other value causes **UNPREDICTABLE** behavior.

## CPU0\_DATA\_CACHE\_SIZE

- Legal range:
  - CPU0\_DATA\_CACHE\_SIZE[0] = 0b1
  - CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b0000, 0b0001, 0b0011, 0b0111, 0b1111
- Specifies the Manager AXI (M-AXI) configuration and the inclusion and size of the data cache controller.
  - CPU0\_DATA\_CACHE\_SIZE[0] = 0 - Area-optimized M-AXI data cache is excluded
  - CPU0\_DATA\_CACHE\_SIZE[0] = 1 - Performance-optimized M-AXI data cache is included

The cache sizes are:

- CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b0000 - 4KB data cache
- CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b0001 - 8KB data cache
- CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b0011 - 16KB data cache
- CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b0111 - 32 KB data cache
- CPU0\_DATA\_CACHE\_SIZE[4:1] = 0b1111 - 64 KB data cache



Setting CPU0\_DATA\_CACHE\_SIZE to any other value causes **UNPREDICTABLE** behavior.

## CPU0\_CFGITCMSZ

- Legal range: CPU0\_CFGITCMSZ > 0b0010
- Size of the Instruction Tightly Coupled Memory (ITCM) region encoded as:
  - CPU0\_CFGITCMSZ = 0b0000 - ITCM is not implemented
  - CPU0\_CFGITCMSZ > 0b0010 -  $2^{(\text{CPU0\_CFGITCMSZ}-1)}$  KB





Note

- The minimum size of the *Tightly Coupled Memory* (TCM) is 4KB and the maximum size is 16MB. Setting CPU0\_CFGITCMSZ to 0b0001 or 0b0010 results in **UNPREDICTABLE** behavior.
- CPU0\_CPU0\_CFGITCMSZ = 0b0000 - No ITCM implemented status is not supported by SSE-310.

## CPU0\_CFGDTCMSZ

- Legal range: CPU0\_CFGDTCMSZ > 0b0010
- Size of the data TCM region encoded as:
  - CPU0\_CFGDTCMSZ = 0b0000 - No DTCM implemented
  - CPU0\_CFGDTCMSZ > 0b0010 -  $2^{(\text{CPU0\_CFGDTCMSZ}-1)}$  KB



Note

- The minimum size of the TCM is 4KB, the maximum is 16MB. Setting CPU0\_CFGDTCMSZ to 0b0001 or 0b0010 results in **UNPREDICTABLE** behavior
- CPU0\_CPU0\_CFGDTCMSZ= 0b0000: No DTCM implemented, is not supported by SSE-310

## CPU0\_ITGUBLKSZ

- Legal range: 3-15
- ITCM gate unit block size
  - Size in bytes =  $2^{(\text{CPU0\_ITGUBLKSZ}+5)}$
  - Minimum block size is 256 bytes
  - Maximum block size is 1MB

## CPU0\_DTGUBLKSZ

- Legal range: 3-15
- DTCM gate unit block size.
  - Size in bytes =  $2^{(\text{CPU0\_DTGUBLKSZ}+5)}$
  - Minimum block size is 256 bytes
  - Maximum block size is 1MB

## HASCPU0CPIF

- Legal range: 0,1
- Specifies the inclusion of the external coprocessor interface. The options are:
  - 0 - Coprocessor interface is excluded
  - 1 - Coprocessor interface is included

## CPU0MCUROMADDR

- Legal range: 0xE0049000 - 0xE00FE000
- The address pointer to MCU ROM table private to CPU0. Only the 20 most significant bits are configurable. All lower address bits are zeros. This configuration option must exist when `HASCSS` = 1.

The CPU0MCUROMADDR represents the [31:12] address range. The remainder range is zero [11:0] = 0x000

## CPU0MCUROMVALID

- Legal range: 1
- The address pointer to MCU ROM table private to each CPU core is valid. This configuration option must exist when `HASCSS` = 1.

## CPU0\_PMC\_PRESENT

- Legal range: 0
- Specifies whether the Programmable MBIST Controller (PMC-100) is included
  - 0 - PMC-100 is excluded
  - 1 - PMC-100 is included



The PMC-100 is delivered with the other processor deliverables.

---

## HASCPU0IWIC

- Legal range: 0
- Specifies whether the Internal Wakeup Interrupt Controller (IWIC) is included
  - 0 - IWIC not included
  - 1 - IWIC included

## CPU0CPUIDRST

- Legal range: 0-15
- A unique Identity value defined for CPU0 It defines the values read at CPU0 local CPU0CPUID register. Legal values are within 0 to 15, inclusive.

## CPU0\_CTI\_PRESENT

- Legal range: 1
- Specifies whether the Cross Trigger Interface (CTI) unit is included

## CPU0\_INITECCEN

- Legal range: 0
- TCM and L1 cache ECC enable out of reset:
  - 1 - ECC enabled
  - 0 - ECC disabled

This signal has no effect if ECC support is not configured in the processor.

## CPU0\_CFGPAHBSZ

- Legal range: 0b010
- Size of the P-AHB peripheral port memory region:
  - 0b000 - P-AHB disabled
  - 0b001 - 64MB
  - 0b010 - 128MB
  - 0b011 - 256MB
  - 0b100 - 512MB



Setting CFGPAHBSZ to any other value results in **UNPREDICTABLE** behavior.

---

## CPU0\_LOCKPAHB

- Legal range: 1
- Disable writes to the PAHBCR register from software or from a debug agent connected to the processor. Asserting this signal prevents changes to AHB peripheral port enable status in PAHBCR.EN Fixed to 1 in SSE-310

## A.4 NPU0 Element Configuration Options

SSE-310 provides several NPU0 Element Configuration render options.

### NPU0\_NUM\_MACS

- Legal range: 32,64,128,256
- The number of 8x8 MACs that are performed per cycle

### NPU0\_CUSTOM\_DMA\_PRESENT

- Legal range: 0

- To instantiate a custom DMA, set this parameter to true. Instantiating a custom DMA also requires manual changes to the RTL.



This parameter cannot be combined with `NPU0_NUM_MACS=32`

---

## NPU0PORSLRST

- Legal range: 0,1
- Describes the default security level that NPU0 resets to:
  - 0 - Secure State
  - 1 - Non-secure State



Only exists when `NUMNPU > 0`.

---

## NPU0PORPLRST

- Legal range: 0,1
- Describes the default Privilege level that NPU0 resets to:
  - 0 - Unprivileged State
  - 1 - Privileged State



Only exists when `NUMNPU > 0`.

---

## A.5 Main and Peripheral interconnect element configuration options

SSE-310 provides several Main and Peripheral interconnect element configuration render options.

### NUM\_AXI\_SUBORDINATES\_EXP\_MI

- Legal range: 2
- Number of Main interconnect expansion AXI subordinate interfaces `XSLVEXPMI{0-<NUM_AXI_SUBORDINATES_EXP_MI-1>}`.

## NUM\_AHB\_SUBORDINATES\_EXP\_PHL

- Legal range: 1
- Number of High Latency Subordinate Peripheral Expansion Interface interfaces on Peripheral interconnect

## NUM\_AHB\_SUBORDINATES\_EXP\_PILL

- Legal range: 1
- Number of Low Latency Subordinate Peripheral Expansion Interface interfaces on Peripheral interconnect

## TCM\_ID\_WIDTH

- Legal range: 2-15
- TCM interface channel ID width of XSLVTCM Expansion AXI subordinate interface

## XS\_ID\_WIDTH

- Legal range:  $(\text{NUMNPU} > 0 ? 6 : 4) - 9$
- Channel ID width for Expansion Subordinate Main Interconnect interfaces, CPU M-AXI interface, subordinate interfaces of the Main Interconnect, ACGs implemented in the Main Interconnect and subordinate interfaces of the NIC-400, same width for all channels



When NPU is present, the legal range is different.

---

## S\_MID\_WIDTH

- Legal range: 4-24
- Manager ID width for Expansion Subordinate AXI, Expansion Subordinate AHB, NPU AXI, CPU M-AXI and CPU P-AHB interfaces. MID is propagated on **AxUSER** and **HAUSER** signals. It is used for exclusivity monitoring and might be used for Manager ID based access control. NPU0ID is provided as a MID value when the NPU is present.

## S\_HMANAGER\_WIDTH

- Legal range: 4-12
- S\_HMANAGER\_WIDTH defines the HMANAGER width for HSLVEXPPI\* interfaces

## VMADDRWIDTH

- Legal range: 14-22
- It defines the address width for all Volatile Memory Banks. This then defines the size of each bank as  $2^{\text{VMADDRWIDTH}}$  bytes.

## VMMPCBLKSIZE

- Legal range: 3-15
- It defines the Block size of the MPC associated with all Volatile Memory Banks. Volatile Memory Block size =  $2^{(VMMPCBLKSIZE+5)}$  bytes.
- The following must be true:  $(VMMPCBLKSIZE+5) < VMADDRWIDTH$

## PERIPHERAL\_INTERCONNECT\_ARBITRATION\_SCHEME

- Legal range: round, round\_nolat
- When a downstream port selects a different upstream port to service, this parameter can add latency:
  - round: Inserts one extra clock cycle of latency
  - round\_nolat: Zero additional clock latency added. With this setting, after a locked transaction, the bus matrix does not insert an IDLE transfer.



The AMBA® 5 AHB Protocol Specification recommends that a bus manager inserts an IDLE transfer after a locked transfer.

---

## XOM\_USER\_SIGNAL\_PRESENT

- Legal range: 0
- Enable the transfer of XOM attribute via (H)RUSER signaling in Main Interconnect and Peripheral interconnect

When set to 1: [LOCKDCAIC](#) configuration must be 1, the RUSER[0] signal on the XMSTEXPCODE and XMSTEXPSRAM (if HASCRYPTO==1) must be transferred to `XSLVEXPMI{0-<NUM_AXI_SUBORDINATES_EXP_MI-1>}`.

When set to 0, RUSER[0] is not present.



- If the PMC-100 is included in the processor configuration, the internal cache RAMs can always be accessed by on-line MBIST
  - The RUSER[0] =1 indicates the current data being returned is Execute Only. If data type access targets a cached XOM location, and there is an external unified cache to SSE-310 then the foregoing cache might store the XOM attribute in TAG memory and might use it to check access type upon cache hit to return or mask the read data.
  - For correct XOM operation the validity requirements of RUSER might be stricter than AMBA definition. Please see this in the corresponding product manuals.
-

## NSMSCEXPST

- Legal range: 0x0000 - 0xFFFF
- The reset value for NSMSCEXP.NS\_MSCEXP[15:0]. This value defines the security level of each expansion MSC when the PD\_SYS power domain is powered up or is reset. It defines up to 16 MSCs, where each bit is:
  - 0 - Secure
  - 1 - Non-secure

## MPCEXPDIS[15:0]

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the SMPCEXPSTATUS bus. If MPCEXPDIS[n] = 1'b1, then either SMPCEXPSTATUS[n] does not exist, or if it does exist, is not used.

## MSCEXPDIS[15:0]

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the SMSCEXPSTATUS, SMSCEXPCLR and NSMSCEXP buses.

If MSCEXPDIS[n] = 1'b1, then either SMSCEXPSTATUS[n], SMSCEXPCLR[n] and NSMSCEXP[n] does not exist, or if they do exist, SMSCEXPSTATUS[n] is not used, SMSCEXPCLR[n] are tied LOW and NSMSCEXP[n] are tied HIGH.

## BRGEXPDIS[15:0]

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the BRGEXPSTATUS and BRGEXPCLR buses.

If BRGEXPDIS[n] = 1'b1, then either BRGEXPSTATUS[n] and BRGEXPCLR[n] does not exist, or if they do exist, BRGEXPSTATUS[n] is not used and BRGEXPCLR[n] are tied LOW.

## PERIPHPPCEXPDIS

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the PERIPHNSPPCEXP{0-3} and PERIPHPPCEXP{0-3} buses.

If PERIPHPPCEXP{0-3}DIS[n] = 1'b1, then either PERIPHNSPPCEXP{0-3}[n] and PERIPHPPCEXP{0-3}[n] does not exist, or if they do exist, PERIPHNSPPCEXP{0-3}[n] and PERIPHPPCEXP{0-3}[n] are not used and tied LOW.

## PERIPHPPCEXP1DIS[15:0]

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the PERIPHNSPPCEXP{0-3} and PERIPHPPCEXP{0-3} buses.

If  $\text{PERIPHPPCEXP}\{0-3\}\text{DIS}[n] = 1'b1$ , then either  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0\text{ to }3\}[n]$  does not exist, or if they do exist,  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0-3\}[n]$  are not used and tied LOW.

### **PERIPHPPCEXP2DIS[15:0]**

- Legal range:  $0xFFFF - 0x0000$
- It disables support for individual bits on the  $\text{PERIPHNSPPCEXP}\{0-3\}$  and  $\text{PERIPHPPPCEXP}\{0-3\}$  buses.

If  $\text{PERIPHPPCEXP}\{0-3\}\text{DIS}[n] = 1'b1$ , then either  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0\text{ to }3\}[n]$  does not exist, or if they do exist,  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0-3\}[n]$  are not used and tied LOW.

### **PERIPHPPCEXP3DIS[15:0]**

- Legal range:  $0xFFFF - 0x0000$
- It disables support for individual bits on the  $\text{PERIPHNSPPCEXP}\{0-3\}$  and  $\text{PERIPHPPPCEXP}\{0-3\}$  buses.

If  $\text{PERIPHPPCEXP}\{0-3\}\text{DIS}[n] = 1'b1$ , then either  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0\text{ to }3\}[n]$  does not exist, or if they do exist,  $\text{PERIPHNSPPCEXP}\{0-3\}[n]$  and  $\text{PERIPHPPPCEXP}\{0-3\}[n]$  are not used and tied LOW.

### **MAINPPCEXP0DIS[15:0]**

- Legal range:  $0xFFFF - 0x0000$
- It disables support for individual bits on the  $\text{MAINNSPPCEXP}\{0-3\}$  and  $\text{MAINPPPCEXP}\{0-3\}$  buses.

If  $\text{MAINPPCEXP}\{0-3\}\text{DIS}[n] = 1'b1$ , then either  $\text{MAINNSPPCEXP}\{0-3\}[n]$  and  $\text{MAINPPPCEXP}\{0\text{ to }3\}[n]$  does not exist, or if they do,  $\text{MAINNSPPCEXP}\{0-3\}[n]$  and  $\text{MAINPPPCEXP}\{0\text{ to }3\}[n]$  are not used and tied LOW.

### **MAINPPCEXP1DIS[15:0]**

- Legal range:  $0xFFFF - 0x0000$
- It disables support for individual bits on the  $\text{MAINNSPPCEXP}\{0-3\}$  and  $\text{MAINPPPCEXP}\{0-3\}$  buses.

If  $\text{MAINPPCEXP}\{0-3\}\text{DIS}[n] = 1'b1$ , then either  $\text{MAINNSPPCEXP}\{0-3\}[n]$  and  $\text{MAINPPPCEXP}\{0\text{ to }3\}[n]$  does not exist, or if they do,  $\text{MAINNSPPCEXP}\{0-3\}[n]$  and  $\text{MAINPPPCEXP}\{0\text{ to }3\}[n]$  are not used and tied LOW.

### **MAINPPCEXP2DIS[15:0]**

- Legal range:  $0xFFFF - 0x0000$
- It disables support for individual bits on the  $\text{MAINNSPPCEXP}\{0-3\}$  and  $\text{MAINPPPCEXP}\{0-3\}$  buses.



If MAINPPCEXP{0-3}DIS[n] = 1'b1, then either MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0-3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0-3}[n] are not used and tied LOW.

### MAINPPCEXP3DIS

- Legal range: 0xFFFF - 0x0000
- It disables support for individual bits on the MAINNSPPCEXP{0-3} and MAINPPPCEXP{0-3} buses.

If MAINPPCEXP{0-3}DIS[n] = 1'b1, then either MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0-3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0-3}[n] are not used and tied LOW.

### UARCH\_CONFIG

- Legal range: 2
- Defines the micro architecture of the design to be:
  - 0 - Gate count optimized, includes lower latency paths
  - 1 - Reserved
  - 2 - Frequency optimized, includes pipelined path

## A.6 Power, Clock and Reset Control Configuration Options

SSE-310 provides several Power, Clock and Reset Control Configuration render options.

### PDCMQCHWIDTH

- Legal range: 4
- It selects the width of Power Dependency Control Matrix Q-Channel interface that the system supports. When set to 0, the Power Dependency Control Matrix Q-Channel interface does not exist.

### LOGIC\_RETENTION\_PRESENT

- Legal range: 0, 1
- Defines if the power domains PD\_CPU0, PD\_CPU0EPU and PD\_SYS supports logic retention.
  - 0 - No
  - 1 - Yes

If 0 then the retention requests targeting the corresponding PCSMs are mapped to ON requests

### COLDRESET\_MODE

- Legal range: 0

- Cold reset mode. It defines if the watchdogs, **RESETREQ** signal, and the SWRESETREQ register value can cause a Cold reset and therefore drive **nCOLDRESETAON**:
  - 0 - Watchdogs, **RESETREQ** signal, and the SWRESETREQ register value contributes to Cold reset.
  - 1 - Watchdogs, **RESETREQ** signal, and the SWRESETREQ register value does not contribute to Cold reset.

## A.7 Revision Information Configuration Options

SSE-310 provides several Revision Information Configuration render options.

### Revision Information Configuration Options

#### SOCVAR[3:0]

- Legal range: 0x0 - 0xF
- The SoC variant or major revision code. Connected to Debug Port targetid port TREVISION field SoC integrator is expected to change it.
  - 0x0 - corresponds to SSE-310 r0p0

#### SOCREV[3:0]

- Legal range: 0x0 - 0xF
- SoC minor revision code. SoC integrator is expected to change it

#### SOCPTID[11:0]

- Legal range: 0x0 - 0xFFFF
- SoC product identity code. Connected to Debug Port targetid port TPARTNO fields. The SoC integrator is expected to change it to its own ID code.

#### SOCIMPLID[11:0]

- Legal range: 0x000 - 0xFFFF
- The SoC integrator JEP106 code. Connected to Debug Port targetid port TDESIGNER field
  - SOCIMPLID[11:8] indicates the JEP106 continuation code
  - SOCIMPLID[7] is always 0
  - SOCIMPLID[6:0] indicates the JEP106 identification code. SoC integrator is expected to change it to its own JEP106 code. To obtain a number, or to see the assignment of these codes, contact [JEDEC](#).

#### IMPLVAR[3:0]

- Legal range: 0x0 - 0xF
- Subsystem variant or major revision code. Connected to the REVISION field (REVISION Parameter) of the MCU ROM table.

SoC integrator must use this to track the revision of their customized subsystem.

### **IMPLREV[3:0]**

- Legal range: 0x0 - 0xF
- Subsystem minor revision code. Connected to the REVAND field (ECOREVNUM input) of the MCU ROM table.

SoC integrator must use this to indicate metal fixes on their customized subsystem.

### **IMPLPRTID[11:0]**

- Legal range: 0x000 - 0xFFF
- Subsystem product identity code. Connected to the PART\_0,PART\_1 fields (PARTNUMBER parameter) of the MCU ROM table.

SoC integrator must modify product identity code to identify the their customized subsystem.

### **IMPLID[11:0]**

- Legal range: 0x000 - 0xFFF
- Subsystem implementor JEP106 code. Connected to the DES\_2,DES\_1,DES\_0 fields (JEPID and JEPCONTINUATION parameter values) of the MCU ROM table.
  - IMPLID[11:8] indicates the JEP106 continuation code
  - IMPLID[7] is always 0
  - IMPLID[6:0] indicates the JEP106 identification code.

The SoC integrator must change it.

## **A.8 Expansion Element Configuration Options**

SSE-310 provides a Expansion Element Configuration render option.

### **EXPLOGIC\_PRESENT**

- Legal range: 0,1
- Defines whether the integration layer inside expansion element is present.
  - 0 - Expansion logic absent
  - 1 - Expansion logic present

If present it instantiates the following:

- Debug Access Port
- Trace Port Interface Unit
- System Counter

- XHB-500 AHB2AXI bridge
- Interface blocks
- Example clock-divider logic



The OOB testbench is instantiated inside the Interface blocks. To run any of the OoB tests, you must enable the expansion logic.

---

## A.9 Rendering Control Configuration Options

SSE-310 provides several Rendering Control Configuration render options.

### PERFORM\_CONFIGCHECK

- Legal range: 0,1
- Defines whether to check the actual values of the Configurable render options against legal range
  - 1 - If the values are outside of the Legal range or contradict other configuration values then RTL rendering prevented
  - 0 - No legal range checks are performed

### SOCRATES\_CONFIG

- Legal range: any
- Passes additional switches to Socrates™ CLI call when NIC is configured. You may use this to configure your license type for Socrates, for example, to use Arm Flexible Access license you must specify: "`--license socrates_flexibleaccess_ms`"

## A.10 IP Version Options

SSE-310 provides several IP Version render options.

### SSE\_CONFIG

- Legal range:

#### PART

CG068-BU-50000

#### VERSION

rOp0-00eac0

- This parameter defines the part and the version of SSE-310 that is used for the rendering of the subsystem. Do not change the pre-configured value it is provided for information.

## OLYMPUS\_CONFIG

- Legal range:

### PART

AT640-BU-50000

### VERSION

rOp2-00rel0

- This parameter defines the part and the version of the Cortex-M85 product that is used for the rendering of the subsystem. The specified SSE-310 product version has to be present in the `<download_folder>` prior to subsystem rendering.

### PART

AT640-BU-50000 = The subsystem uses the selected part

### VERSION

rOp2-00rel0 = The subsystem uses the selected version of the PART



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the Cortex-M85 part and version names.

---

## OLYMPUS\_FPU\_CONFIG

- Legal range:

### PART

AT641-MN-22110

### VERSION

rOp2-00rel0

- This parameter defines the part and the version of the Cortex-M85 FPU product that is used for the rendering of the subsystem. The specified Cortex-M85 FPU product version has to be present in the `<download_folder>` prior to subsystem rendering.

### PART

AT641-MN-22110 = The subsystem uses the selected part

### VERSION

rOp2-00rel0 = The subsystem uses the selected version of the PART



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the Cortex-M85 FPU part and version names.

---

## OLYMPUS\_ETM\_CONFIG

- Legal range:

## PART

TM982-BU-50000

## VERSION

r0p2-00rel0

- This parameter defines the part and the version of the Cortex-M85 ETM product that is used for the rendering of the subsystem. The specified Cortex-M85 ETM product version has to be present in the <download\_folder> prior to subsystem rendering.

## PART

TM982-BU-50000 - The subsystem uses the selected part.

## VERSION

r0p2-00rel0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the Cortex-M85 ETM part and version names.

---

## ETHOS\_U55\_CONFIG

- Legal range:

### PART

ML004-BU-50000

### VERSION

r2p0-01eac0

- This parameter defines the part and the version of the Ethos-U55 product that is used for the rendering of the subsystem. The specified Ethos-U55 product version has to be present in the <download\_folder> prior to subsystem rendering.

### PART

ML004-BU-50000 - The subsystem uses the selected part.

### VERSION

r2p0-01eac0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the Ethos-U55 part and version names.

---

## CMSDK\_CONFIG

- Legal range:

### PART

BP210-BU-00000

## VERSION

r1p1-00rel0

- This parameter defines the part and the version of the CMSDK product that is used for the rendering of the subsystem. The specified CMSDK product version has to be present in the `<download_folder>` prior to subsystem rendering.

## PART

BP210-BU-00000 = The subsystem uses the Cortex-M System Design Kit components

## VERSION

r1p1-00rel0 = The subsystem uses the selected version of the PART



See *Arm® Corstone™ SSE-310 Example Subsystem Release Note* for more information about the CMSDK part and version names.

---

## SIE200\_CONFIG

- Legal range:

## PART

BP300-BU-50000

## VERSION

r3p2-00rel0

- This parameter defines the part and the version of the SIE-200 product that is used for the rendering of the subsystem. The specified SIE-200 product version has to be present in the `<download_folder>` prior to subsystem rendering.

## PART

BP300-BU-50000 - The subsystem uses the selected part.

## VERSION

r3p2-00rel0 - The subsystem uses the selected version of the PART.



See *Arm® Corstone™ SSE-310 Example Subsystem Release Note* for more information about the SIE-200 part and version names.

---

## SIE300\_CONFIG

- Legal range:

## PART

BP301-BU-50000

## VERSION

r1p2-00rel0

- This parameter defines the part and the version of the SIE-300 product that is used for the rendering of the subsystem. The specified SIE-300 product version has to be present in the `<download_folder>` prior to subsystem rendering.

**PART**

BP301-BU-50000 - The subsystem uses the selected part.

**VERSION**

r1p2-00rel0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the SIE-300 part and version names.

---

## PCK600\_CONFIG

- Legal range:

**PART**

PL608-BU-50000

**VERSION**

r0p5-00rel0

- This parameter defines the part and the version of the PCK-600 product that is used for the rendering of the subsystem. The specified PCK-600 product version has to be present in the `<download_folder>` prior to subsystem rendering.

**PART**

PL608-BU-50000 - The subsystem uses the selected part.

**VERSION**

r0p5-00rel0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the PCK-600 part and version names.

---

## XHB500\_CONFIG

- Legal range:

**PART**

PL417-BU-50000

**VERSION**

r0p1-00rel0



- This parameter defines the part and the version of the XHB-500 product that is used for the rendering of the subsystem. The specified XHB-500 product version has to be present in the `<download_folder>` prior to subsystem rendering.

**PART**

PL417-BU-50000 - The subsystem uses the selected part.

**VERSION**

r0p1-00rel0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the XHB-500 part and version names.

---

## DAPLITE2\_CONFIG

- Legal range:

**PART**

TM840-BU-50000

**VERSION**

r2p1-00rel0

- This parameter defines the part and the version of the CoreSight product that is used for the rendering of the subsystem. The specified CoreSight product version has to be present in the `<download_folder>` prior to subsystem rendering.

**PART**

TM840-BU-50000 - The subsystem uses the selected part.

**VERSION**

r2p0-00rel0 - The subsystem uses the selected version of the PART.



See Arm® Corstone™ SSE-310 Example Subsystem Release Note for more information about the CoreSight part and version names.

---

## NIC400\_CONFIG

- Legal range:

**PART**

PL401-BU-50000

**VERSION**

r1p2-00rel1

- This parameter defines the part and the version of the NIC-400 product that is used for the rendering of the subsystem. The specified NIC-400 product version has to be present in the `<download_folder>` prior to subsystem rendering.

**PART**

PL401-BU-50000 - The subsystem uses the NIC-400 part.

**VERSION**

r1p2-00rel0 - The subsystem uses the NIC-400 version of the PART.



See *Arm® Corstone™ SSE-310 Example Subsystem Release Note* for more information about the NIC-400 part and version names.

---

## TPIUM\_CONFIG

- Legal range:

**PART**

TM850-BU-50000

**VERSION**

r0p1-00rel0

- This parameter defines the part and the version of the TPIU-M product that is used for the rendering of the subsystem. The specified TPIU-M product version has to be present in the `<download_folder>` before subsystem rendering.

**PART**

TM850-BU-50000 - The subsystem uses the selected part

**VERSION**

r0p1-00rel0 - The subsystem uses the selected version of the PART.



See *Arm® Corstone™ SSE-310 Example Subsystem Release Note* for more information about the TPIU-M part and version names.

## Appendix B Revisions

This appendix describes the technical changes between released issues of this book.

**Table B-1: Issue 0000-01**

Change	Location
Initial issue of the document	-