Arm® Corstone™ SSE-300 Example Subsystem

Revision: r0p1

Technical Reference Manual

Non-Confidential

Copyright © 2020–2021 Arm Limited (or its affiliates).
All rights reserved.

Issue 03

101773_0001_03_en

Arm® Corstone™ SSE-300 Example Subsystem

Technical Reference Manual

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0000-01	8 May 2020	Non-Confidential	Initial release
0000-02	22 October 2020	Non-Confidential	EAC release for r0p0
0001-03	22 June 2021	Non-Confidential	EAC release for r0p1

Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

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

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

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

Copyright © 2020–2021 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.

Web address

developer.arm.com

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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Contents

1 Introduction........ 9

1.1 Product revision status........ 9

1.2 Intended audience........ 9

1.3 Conventions........ 9

1.4 Additional reading........ 11

1.5 Feedback........ 13

2 Overview of SSE-300........ 14

2.1 Document-specific conventions........ 15

2.2 Compliance........ 17

2.3 Product documentation........ 18

2.4 Product revisions........ 19

2.5 Arm architecture........ 19

2.6 Trusted Base System Architecture for Armv8-M........ 19

2.7 Interrupt controller architecture........ 20

2.8 Power Policy Unit architecture........ 21

2.9 Advanced Microcontroller Bus architecture........ 21

3 Topology........ 22

3.1 System Block Diagram........ 22

4 Functional Description........ 26

4.1 Clocking infrastructure........ 26

4.1.1 Clock generation and control........ 28

4.1.2 Clock Q-Channel control interface dependencies........ 30

4.1.3 Clock configuration interface........ 32

4.2 Reset infrastructure........ 33

4.2.1 Warm reset generation and control........ 34

4.2.2 Power-on and Cold Reset Handling........ 34

4.2.3 CPU Reset Handling........ 35

4.2.4 Boot after reset........ 35

4.3 CPU........ 35

4.3.1 EVENT Interfaces........ 40

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.3.2 Interrupts....40
4.4 System Interconnect infrastructure....42
4.4.1 ACC_WAIT Control.... 42
4.5 Volatile Memory....43
4.6 Timers and Watchdogs.... 43
4.6.1 Timestamp based Timers.... 44
4.6.2 SLOWCLK AON Timers.... 45
4.7 Message Handling Unit....45
4.8 Power Policy Units....46
4.9 Peripheral Protection Controllers....46
4.10 Memory Protection Controllers.... 47
4.11 CryptoCell.... 47
4.11.1 No-Crypto Configuration....47
4.12 Debug Infrastructure....47
4.12.1 Basic Debug Configuration.... 48
4.13 System and Security Control....50
4.14 Power integration....50
4.14.1 Power integration overview.... 50
4.14.2 Power domain hierarchy and bounded regions....52
4.14.3 Power domains....54
4.14.4 Power Policy Units.... 57
4.14.5 Bounded Region power modes....59
4.14.6 Wake-up sources.... 67
4.14.7 Power Dependency Control.... 67
4.14.8 System power states.... 68

5 Interfaces.... 72
5.1 Input and output clocks....73
5.2 Functional integration resets.... 75
5.3 P-Channel and Q-Channel Device interfaces....78
5.3.1 Clock Control Q-Channel Device interfaces.... 79
5.3.2 Power Control P-Channel Device interfaces....80
5.3.3 Power Control Wakeup Q-Channel Device interfaces....81
5.4 Clock Control Q-Channel Control interfaces....81
5.5 Expansion Power Control Dependency interface....82
5.6 Power Domain ON Status Signals....82

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.7 System Timestamp interface....83
5.8 Main Interconnect Expansion interfaces....84
5.9 Peripheral Interconnect Expansion interfaces....85
5.10 Interrupt interfaces....86
5.11 CPU Co-Processor interface....86
5.12 CPU Custom Datapath Extension Interface....87
5.13 TCM DMA Slave Interfaces....87
5.14 Debug and Trace Related Interfaces....88
5.14.1 Debug Access Interface....88
5.14.2 Serial Wire JTAG Interface....88
5.14.3 Debug Timestamp Interface....89
5.14.4 Cross Trigger Channel Interface....89
5.14.5 Cross Trigger Interface....90
5.14.6 Debug APB Expansion Interface....90
5.14.7 CPU<n> External Peripheral Interface EPPB....90
5.14.8 ATB Trace Interfaces....90
5.14.9 Trace port interface....91
5.14.10 Debug Authentication Interface....92
5.15 CryptoCell-Related expansion interfaces....93
5.16 Security Control Expansion Signals....93
5.16.1 Memory Protection Controller Expansion....94
5.16.2 Peripheral Interconnect Peripheral Protection Controller Expansion....94
5.16.3 Main Interconnect Peripheral Protection Controller Expansion....95
5.16.4 Master Security Controller Expansion....96
5.16.5 Bridge Buffer Error Expansion....97
5.16.6 Other Security Expansion Signals....97
5.17 Clock configuration interface....98
5.18 Miscellaneous Signals....99

6 Programmer Model....102
6.1 System Memory Map Overview....102
6.1.1 High Level System Address Map....103
6.2 CPU TCM memories....117
6.3 Volatile Memory Region....117
6.4 Peripheral Region....118
6.4.1 Message Handling Unit registers....120

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.4.2 Secure Access Configuration Register Block....121
6.4.3 Non-secure Access Configuration Register Block....135
6.4.4 Timestamp Timers....139
6.4.5 Timestamp Watchdogs....140
6.5 Processor Private Region....141
6.5.1 CPU<N>_PWRCTRL Register Block....142
6.5.2 CPU<N>_IDENTITY Register Block....143
6.5.3 CPU<N>_SECCTRL Register Block....144
6.6 System Control Peripheral Region....145
6.6.1 SYSINFO Register Block....147
6.6.2 System Control Register Block....151
6.7 CPU Private Peripheral Bus region....167
6.7.1 EWIC....169
6.7.2 Cortex-M55 TPIU registers....175
6.8 Debug System Access Region....175
6.8.1 HASCSS = 0....175
6.8.2 HASCSS = 1....175
6.9 Peripheral Expansion Region....175

A Mapping of the user signals of the AXI and AHB expansion interfaces....177

B Mapping of the HMASTER signals of the AHB expansion master interfaces....179

C Configurable render options....180

D Revisions....217

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1 Introduction

1.1 Product revision status

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

rx Identifies the major revision of the product, for example, r1.
py Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

This book is written for electronics engineers that are designing, configuring, laying out, validating, and verifying the Arm® Corstone™ SSE-300 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.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Convention	Use
monospace bold	Denotes language keywords when used outside example code.
monospace underline	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.
Note	This represents an important piece of information that needs your attention.
Tip	This represents a useful tip that might make it easier, better or faster to perform a task.
Remember	This is a reminder of something important that relates to the information you are reading.

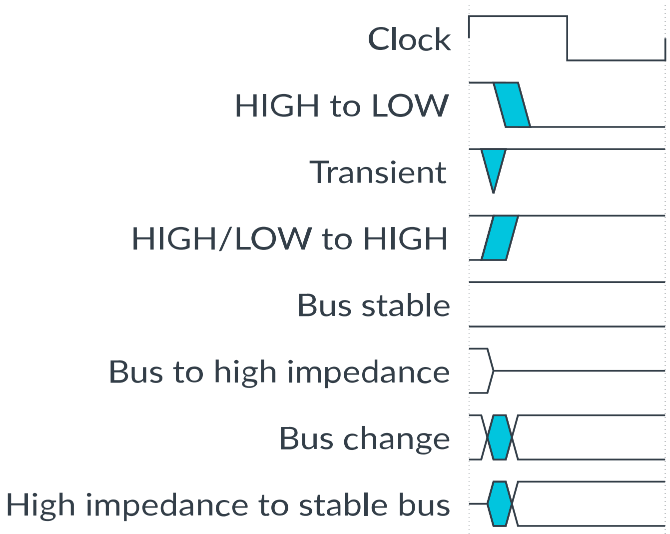
Timing diagrams

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

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 APB Protocol Specification	IHI 0024C	No
AMBA® 4 ATB Protocol Specification	IHI 0032	No
AMBA® AXI and ACE Protocol Specification	IHI 0022	No
AMBA® Low Power Interface Specification	IHI 0068C	No
Arm® AMBA® 5 AHB Protocol Specification	IHI 0033	No
Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual	DDI 0475	No

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 Technical Reference Manual	100572	No
Arm® CoreSight™ ETM-M55 Technical Reference Manual	101053	No
Arm® Corstone™ SSE-300 Example Subsystem Configuration and Integration Manual	101774	Yes
Arm® Corstone™ SSE-300 Example Subsystem Release Note	CG067-DC-06003	Yes
Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual	101773	No
Arm® Cortex®-M55 Processor Integration and Implementation Manual	101052	Yes
Arm® Cortex®-M55 Processor User Guide Reference Material	101272	No
Arm® Cortex®-M55 Technical Reference Manual	101051	No
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074	No
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® Power Policy Unit Architecture Specification, version 1.1	DEN 0051D	No
Arm® Server Base System Architecture 5.0 Platform Design Document	DEN 0029	No
Arm® SSE-123 Example Subsystem Technical Reference Manual	101370	No
Arm® v8-M Architecture Reference Manual	DDI 0553	No

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1.5 Feedback

Arm welcomes feedback on this product and its documentation.

Feedback on this product

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

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

Feedback on content

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

- The title Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual.
- The number 101773_0001_03_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2 Overview of SSE-300

The *Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual* specifies the architecture of the Corstone *prodnameInternetofThings*(*IoT*)*ExampleSubsystem*(prodname).

SSE-300 integrates the following key components available from Arm that can be integrated into a larger system:

- Cortex-M processor core with optional MVE, FPU, DSP, and CDE extensions, caches, TCMs and ETM
- SSE-300 supports only one Arm® Cortex-M55 processor
- SSE-300 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)
- Optional Arm® Ethos™-U55 NPU as a testbench component

Note

This specification does not include descriptions of other components that are not directly architecturally visible but are necessary to implement the system.

The above components are integrated to implement SSE-300 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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, the following document-specific conventions apply.

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]**

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 returns 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.
RAZ/WI	Read as Zero Write Ignores	These always returns Zeros for read and ignores writes.
RAZ/WO	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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Abbreviation	Accessibility	Description
W0C	Write 0 to Clear	Each bit when written with zero is cleared to zero. Writing ones to these are ignored.
W1T	Write 1 to Trigger	Each bit when written with one triggers an action. Writing zeros to these are ignored. Unless stated, any read from these bit fields returns zeros. This is the same as W1S with RAZ.
W0T	Write 0 to Trigger	Each bit when written with zero triggers an action. Writing ones to these are ignored. Unless stated, any read from these bit fields returns zeros. This is like W0S with RAZ.

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

In these specification areas of the design, register values or behaviors, are described as CONFIGURATION DEFINED (CFG_DEF) if the value or behavior depends on the configuration of the component.

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-300 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*
- *Arm® AMBA® 5 AHB Protocol Specification*
- *AMBA® 4 ATB Protocol Specification*
- *AMBA® 4 APB Protocol Specification*
- *AMBA® Low Power Interface Specification*

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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2.3 Product documentation

Each SSE-300 document has an intended audience and is associated with specific tasks in the design flow.

These documents do not reproduce SSE-300 architecture and protocol information. For relevant protocol and architectural information that relates to this product, see Section 1.4 Additional reading.

The SSE-300 documentation is as follows:

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the SSE-300. 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-300 contact:

- The implementer to determine:
  - The build configuration of the implementation
  - What integration, if any, was performed before implementing SSE-300
- 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-300 into an SoC
- How to implement SSE-300 into your design
- The processes to validate the configured design

Arm product deliverables include reference scripts and information about using them to implement your design.

Note

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2.4 Product revisions

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

r0p0

First release

r0p0-01eac0

Introduced the Arm® Ethos™-U55 NPU in the SSE-300 testbench

r0p1

- Cortex-M55 upgraded to version r1p0-00eac0, as a result, ACI support has been introduced in SSE-300
- SIE-200 upgraded to version r3p2-00rel0
- SIE-300 upgraded to version r1p2-00rel0
- CoreLink XHB-500 Generic Global Bundle upgraded to r0p1-00rel0

2.5 Arm architecture

The Arm® Cortex M-55 Processor in SSE-300 implements the Armv8-M architecture which executes the Armv8-M T32 instruction set.

See *Arm® v8-M Architecture Reference Manual* for more information.

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 contains best practice security principles when designing systems around Armv8-M processing elements (PEs). See Arm Platform Security Architecture, Trusted Base System Architecture for Armv8-M Beta version 1.1.

SSE-300 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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-300 to comply with TBSA-M, when expanding the system the integrator must comply with TBSA-M requirements. For example:

- 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 masters and slaves to the system, the memory space continues to obey Trusted and Non-trusted world partitioning

While the adherence to TBSA-M requirements will greatly help to create a secure system, it does not however create a system that mitigates Denial of Service (DoS) attacks. As such, DoS is out of scope of $product. 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-300 implements the following features:

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

See also

- *Arm® Cortex®-M55 Technical Reference Manual*
- *Arm® Cortex®-M55 Processor User Guide Reference Material*

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2.8 Power Policy Unit architecture

The power domains in SSE-300 are controlled by Power Policy Units (PPUs), which comply with the Arm PPU architecture.

See *Arm® Power Policy Unit Architecture Specification, version 1.1* for more information.

2.9 Advanced Microcontroller Bus architecture

SSE-300 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*
- *Arm® AMBA® 5 AHB Protocol Specification*
- *AMBA® 4 APB Protocol Specification*

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3 Topology

The following sections describe the topology of SSE-300.

The main system is primarily composed of the following components:

- The main interconnect and peripheral interconnect and other related functionality like expansion interfaces
- System and security control related registers and logic
- Power control logic that resides in the PD_AON power domain that controls the power state of all switchable power domains
- All VM interfaces and peripherals in the PD_SYS domain along with the interfaces to timers and watchdog timers

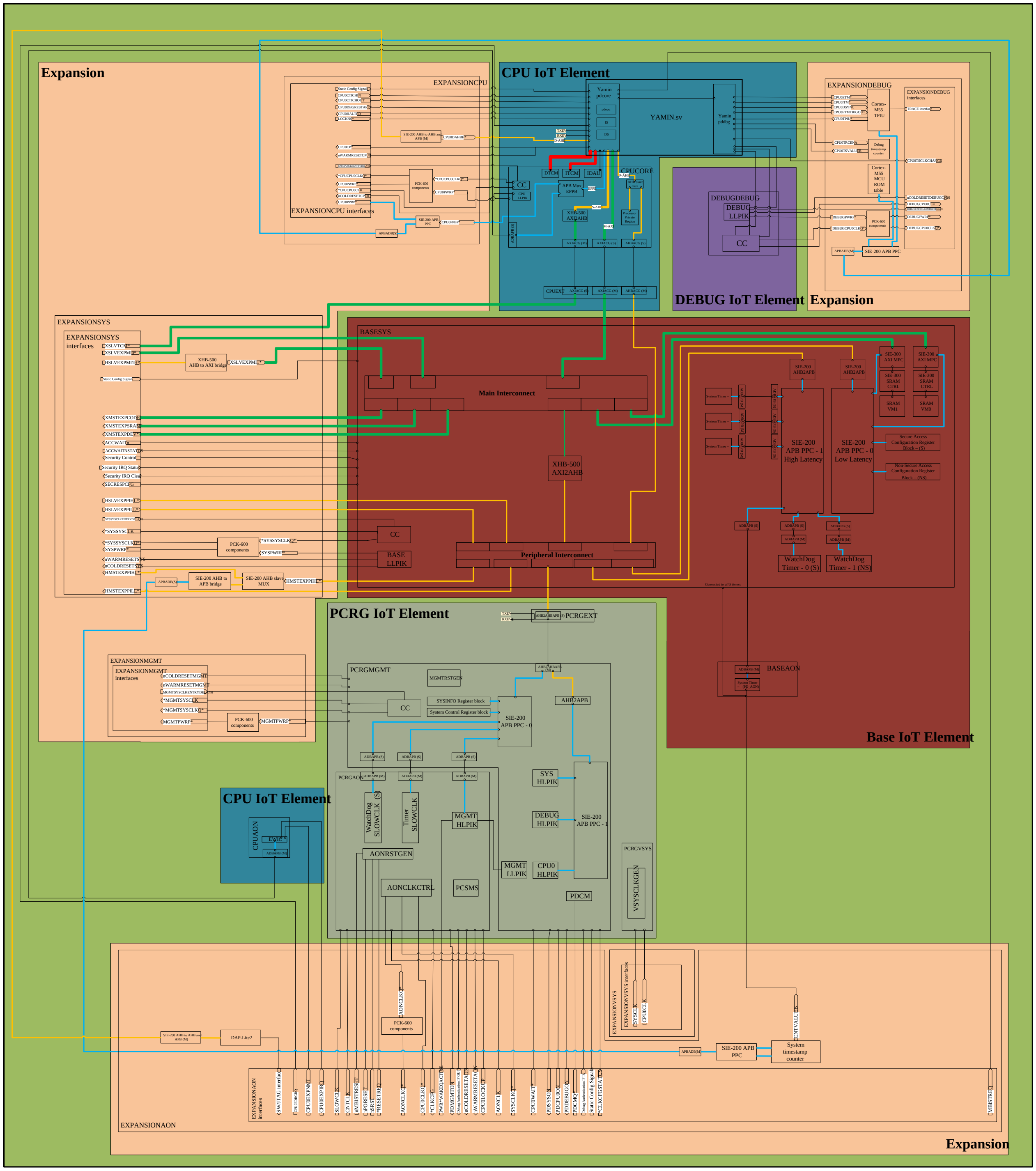
3.1 System Block Diagram

The system block diagrams show the topology of SSE-300 and the power domains of the subsystem.

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-1: SSE-300 logical (element) structured system topology



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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

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

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-300 supports one CPU element 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.

Volatile memory bank element

Volatile memory banks collectively implement the main volatile storage within the subsystem and the functionality (MPC, EAM, SRAM control, 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

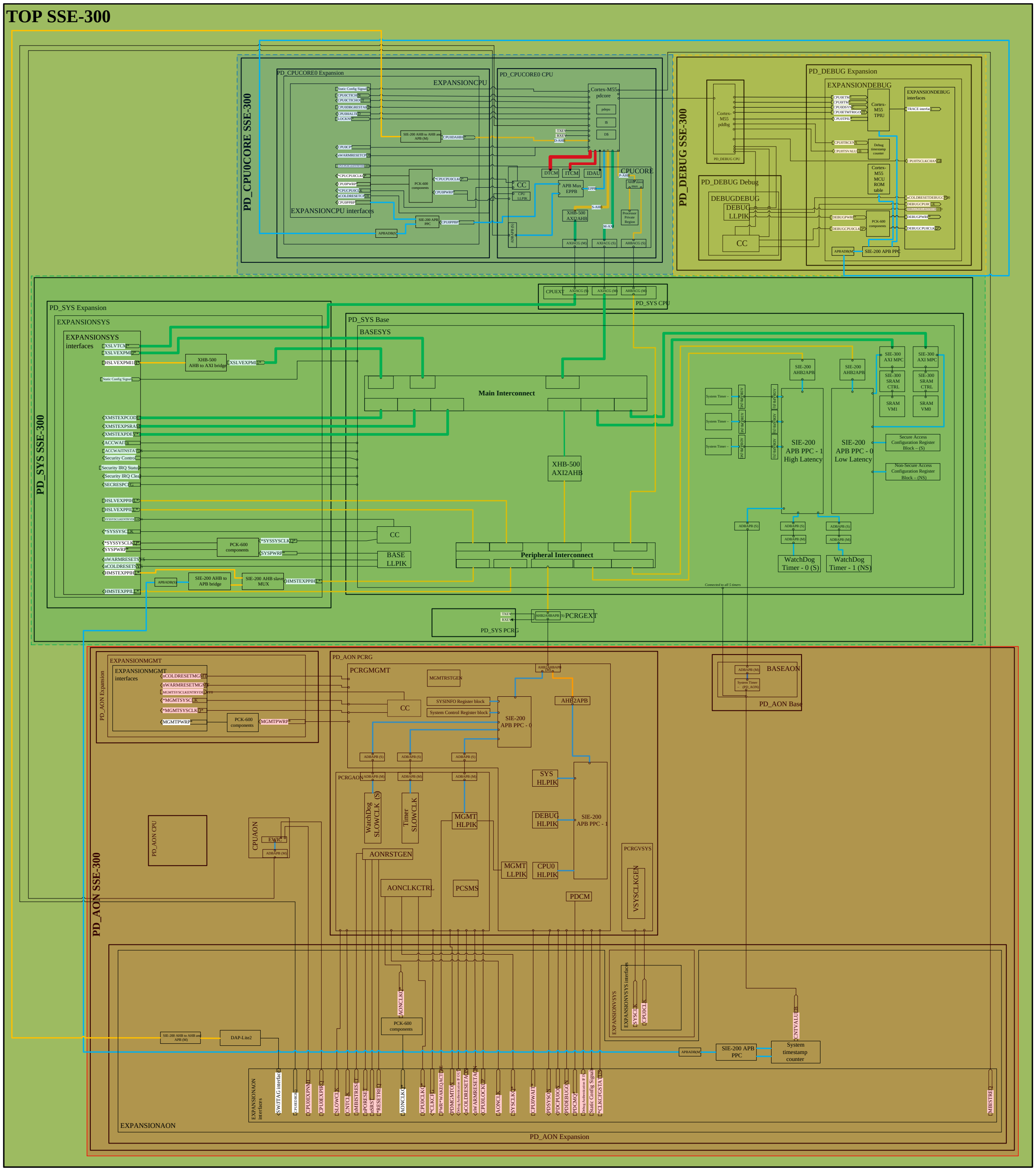
Defines the infrastructure required to implement all configure, control, and monitor system states. These include security, clock, reset, and power control.

Debug system element

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 3-2: SSE-300 power domain structured system topology



SSE-300 provides several render configuration options. Configurable render options describes the detailed configuration options.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4 Functional Description

The following sections describe the functional description of the components of SSE-300.

4.1 Clocking infrastructure

The SSE-300 has multiple clock inputs in the PD_AON power domain. AONCLK and SLOWCLK drive low speed logic in PD_AON. CNTCLK drives low speed logic both in PD_AON and PD_SYS. CPU0CLK and SYSCLK target components with higher frequency requirements, and drive logic primarily in the switchable power domains. Clocks entering the switchable domains are gated within the subsystem. The subsystem provides clock outputs for the expansion logic in the switchable power domains, that are driven by the gated versions of SYSCLK and CPU0CLK.

The clocking infrastructure supports high-level clock gating, which helps dynamic power reduction. The configuration of clock source components that are outside of the subsystem is supported by clock configuration interfaces.

The following figure shows a high-level representation of the clock distribution structure of SSE-300. Ports that are relevant from the perspective of SoC integration are also represented.

In the diagram:

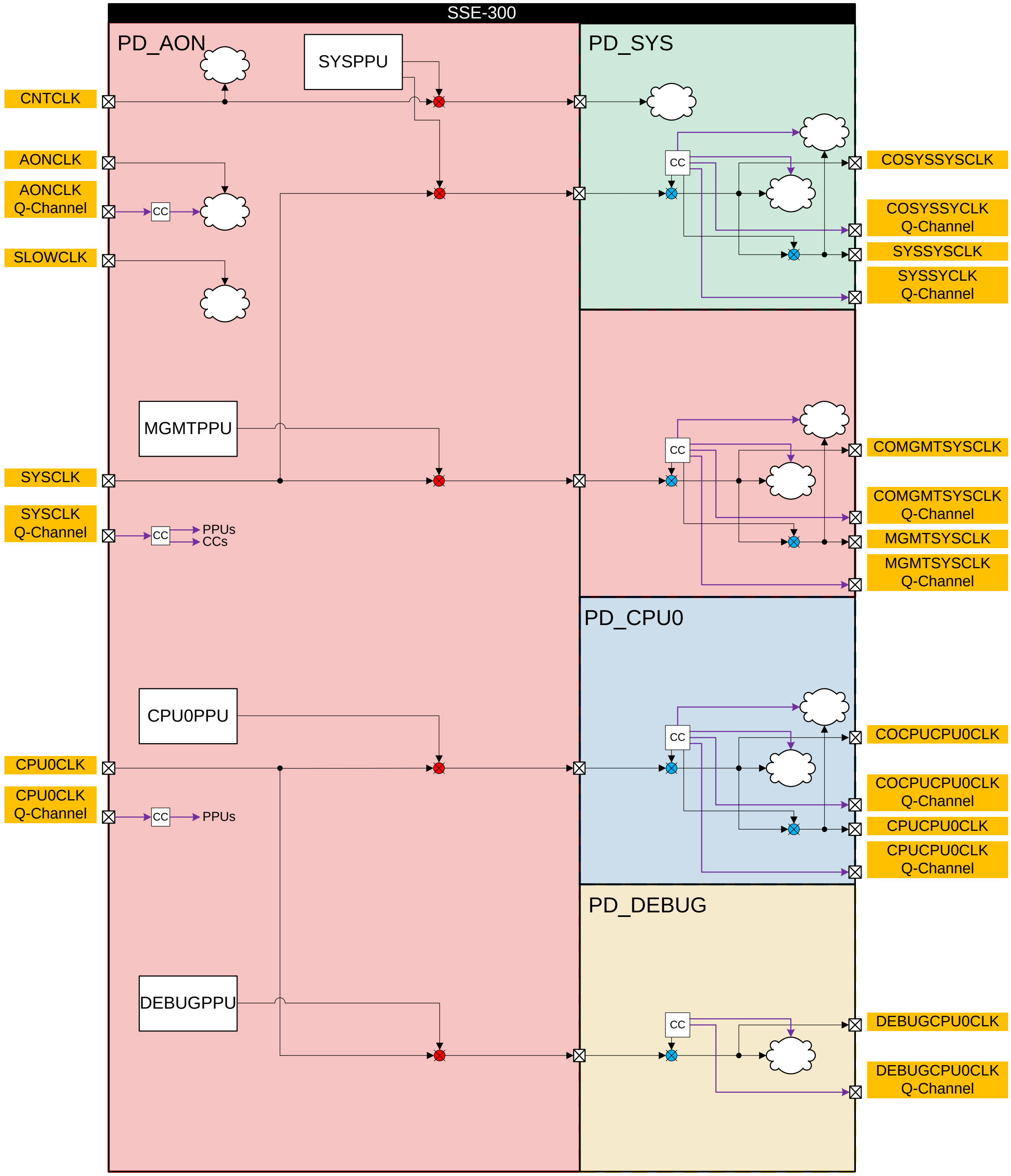
- The CC blocks are clock control components for high-level clock gating and clock gating in Warm reset
- The blue and red circles are clock gates
- The PPU blocks are Power Policy Units

Note

The diagram does not show the complete internal Q/P-Channel infrastructure that is required for each clock controller. Clocks are often gated twice. Once in relation to the state of the power domain that the clock is driving (red clock gates), and gated again internally within a domain, depending on the activity in the domain (blue clock gates).

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-1: SSE-300 clock distribution structure



The clock infrastructure in SSE-300 is described in the section Input and output clocks.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.1.1 Clock generation and control

SSE-300 provides clock control Q-Channel control and device interfaces for managing the availability of the clock inputs and outputs.

SSE-300 provides a Q-Channel control interface for each of the input clocks for the following purposes:

- To allow the subsystem to request for the availability of a clock source
- To allow an external clock controller to handshake with the subsystem to safely turn OFF the clock source

Clock gating control strategies supported by the Q-Channel control interfaces are detailed in Clock gating control of input clocks.

Dependencies between the various Q-Channel control interfaces are detailed in Clock Q-Channel control interface dependencies.

The SSE-300 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 Q-Channel interfaces are used to support high-level clock gating in the switchable domains. All clock Q-Channel Device Interfaces are for clock control only and do not support waking the subsystem from HIBERNATION0.

Between the acceptance of a request to WARM_RST on the SYSPWR P-Channel and the assertion of **nWARMRESETSYS**, **SYSSYSCLK** can be gated irrespective of the state of the SYSSYSCLK Q-Channel Device Interface.

Between the acceptance of a request to **WARM_RST** on the CPU0PWR P-Channel and the assertion of **nWARMRESETCPU0**, **CPUCPU0CLK** can be gated irrespective of the state of the CPUCPU0CLK Q-Channel Device Interface.

Between the acceptance of a request to **WARM_RST** on the MGMTPWR P-Channel and the assertion of **nWARMRESETMGMT**, **MGMTSYSCLK** can be gated irrespective of the state of the MGMTSYSCLK Q-Channel Device Interface.

For more information regarding the P-Channels, refer to Power control P-Channel Device Interfaces.

SSE-300 provides clock force registers to request the subsystem to continue clock generation on the output clocks regardless of the QACTIVEs of the various clock control Q-Channel device interfaces, provided that the related switchable Power Domain is turned ON.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.1.1.1 Clock gating control of input clocks

Clock inputs are associated with the clock control Q-Channel control interfaces. The exceptions are CNTCLK and SLOWCLK clocks. The granularity of the control of each clock input is specified in this section.

The clock control Q-Channel control interfaces support high-level clock gating in PD_AON (clock gating at the root of the clock tree in a power domain) by clock management components that are external to SSE-300. These components can be clock sources. There might be a higher cycle cost in starting and stopping the external clock sources compared to the latency of clock controllers in the subsystem.

The various clock gating control strategies supported by the clock Q-Channel control interfaces are described by the following terms:

SW Control Based High-Level Clock Gating Disable

SSE-300 provides clock source force registers to request the clock sources to continue clock generation regardless of the state of the subsystem. These clock source forces do not affect the power or high-level clock gating provided in the subsystem.

If a clock Q-Channel control IF is in the Q_RUN state, and the related CLOCK_FORCE bit is asserted, quiescence requests are denied on the Q-Channel until forcing is disabled.

Power State Based High-Level Clock Gating

The clock can be gated if it is allowed by the states of the power domains that the clock is driving.

Activity Based High-Level Clock Gating

If the logic driven by the gated clock is idle, the clock can be gated irrespective of the state of the power domain that the logic resides in. QACTIVE is kept asserted if the logic driven by the related clock is active. Quiescence requests that arrive when the logic is active are denied or delayed.

Clock gating strategies supported by the various clock control Q-Channel control interfaces are as follows.

AONCLK Q-Channel control interface

Supports both Activity Based and SW Control Based High-Level Clock Gating Disable.

Activity Based gating is affected by the following components:

- *External Wakeup Interrupt Controller* (EWIC) that allows the processor to be woken by an interrupt
- PD_MGMT PPU, which handles various wake-up requests
- All other logics in the PD_AON domain that are not running on SLOWCLK and SYSCLK

SYSCLK Q-Channel control interface

Supports Activity Based, Power State Based, and SW Control Based High-Level Clock Gating Disable.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Power State Based

  In the following Power Modes, or when the SYSPPU is changing Power Mode, QACTIVE is kept asserted and quiescence requests are denied.

  - SYSPPU Power Modes: ON, WARM_RST (functional Power Modes, when the subsystem is outside of the HIBERNATION0 and SYS_RET System Power States)
- Activity Based gating is affected by the following components:
  - Power Control logic that resides in PD_MGMT power domain that controls the PD_SYS, PD_CPU0, and PD_DEBUG power domains

CPU0CLK Q-Channel control interface

Supports Power State Based and SW Control Based High-Level Clock Gating Disable.

- Power State Based

  In the following Power Modes, or when the DEBUGPPU or the CPU0PPU is changing Power Mode, QACTIVE is kept asserted and quiescence requests are denied:

  - DEBUGPPU Power Modes: ON, WARM_RST (functional Power Modes)
  - CPU0PPU Power Modes: ON, WARM_RST, FUNC_RET, MEM_OFF (functional Power Modes)

4.1.2 Clock Q-Channel control interface dependencies

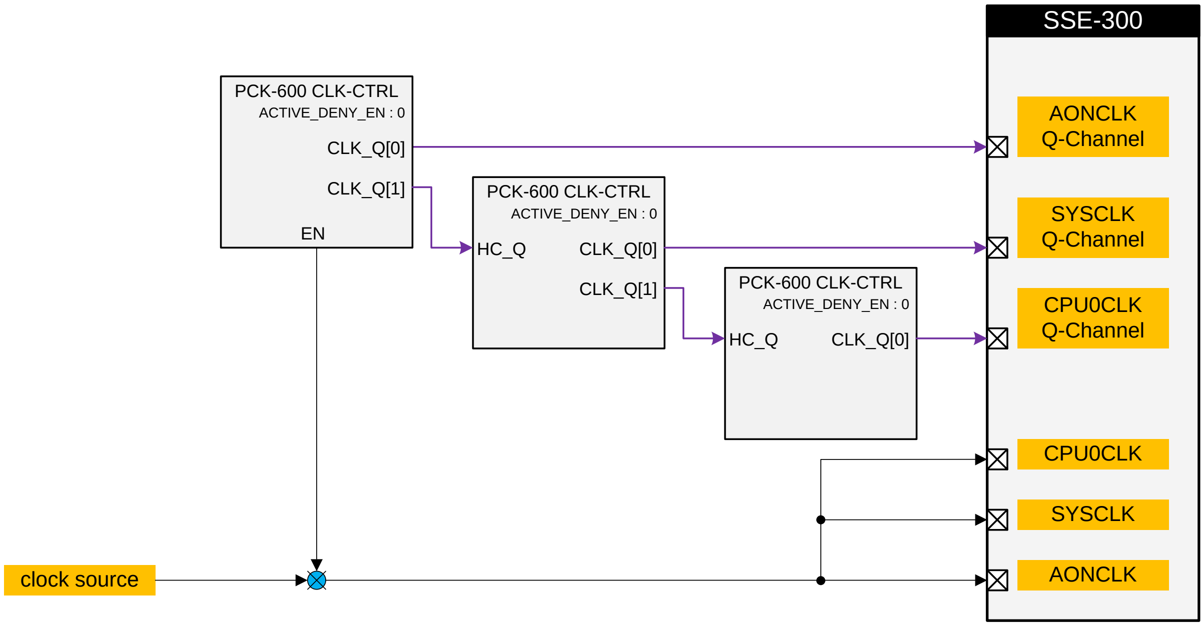
There is a dependency between the various clock control Q-Channel interfaces. Transitions on the CPU0CLK Q-Channel require SYSCLK and AONCLK to run, and transitions on the SYSCLK Q-Channel require AONCLK to run. If the clock inputs SYSCLK, CPU0CLK and AONCLK are driven by independent clock sources with independent clock control, the foregoing dependency does not impact the clock control network.

If some of the clock inputs are driven by the same clock source and the clock control logic is not independent, the foregoing dependency must be handled by the clock control network.

In the first example, the same clock source is connected to all three clocks.

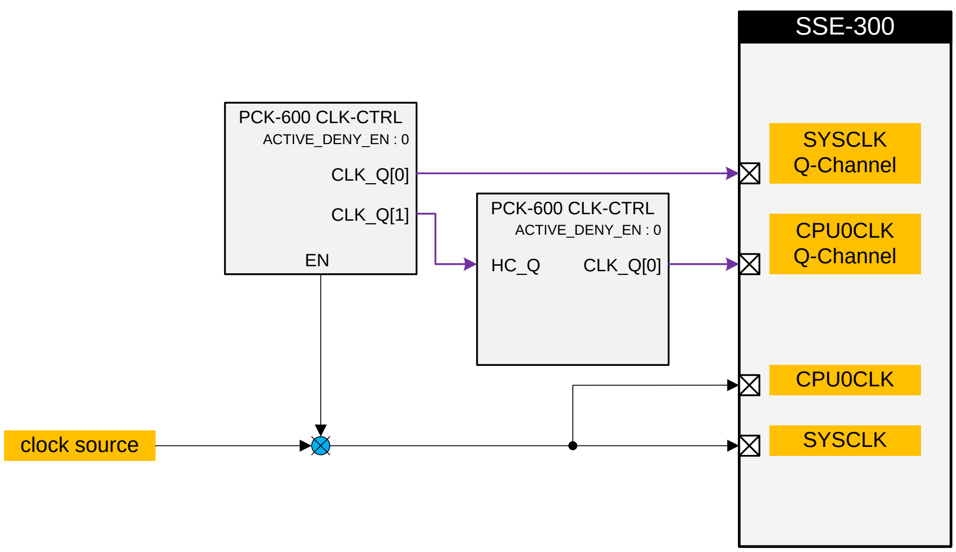
Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-2: AONCLK, CPU0CLK, and SYSCLK driven by the same clock source



In the second example, the same clock source is connected to SYSCLK and CPU0CLK.

Figure 4-3: CPU0CLK and SYSCLK driven by the same clock source



In both examples, cascaded PCK-600 CLK-CTRL components make sure that the dependent Q-Channels can reach Q_STOPPED. Active denial related functionality has to be turned off in the clock control logic of the dependent clocks. Failing to do so might result in livelock scenarios or excessive delays in accepting quiescence requests on the clock control Q-Channels. If the clock control networks in the examples are further expanded hierarchically, the usage of the clock enable output depicted is optional.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

Only ports of PCK-600 CLK-CTRL components are depicted that are relevant from the perspective of the Q-Channel dependency.

4.1.3 Clock configuration interface

Description of the clock configuration interface.

SSE-300 provides control and status signals for the input clocks **AONCLK**, **CPU0CLK**, and **SYSCLK**. These signals support the configuration of generators or dividers that might exist in the expansion logic. 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.

Since **SYSCLK** and **CPU0CLK** must have the same frequency, the CPU0CLKCFG output must not be used in the expansion logic, and the CPU0CLKCFGSTATUS input must be tied to LOW.

The register fields CLK_CFG0.CPU0CLKCFG and CLK_CFG0.CPU0CLKCFGSTATUS should not be used either.

Table 4-1: Clock configuration interface signals

Signal	Direction	Description
CPU0CLKCFG[3:0]	Output	These control signals provide a set of four-bit signals that allows the system to configure an external clock generation logic that drives CPU0CLK. The CLK_CFG0.CPU0CLKCFG register field drives this signal.
CPU0CLKCFGSTATUS[3:0]	Input	Four-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.
SYSCLKCFG[3:0]	Output	This control signal provides a four-bit signal that allows the system to configure an external clock generation logic that drives SYSCLK clock. The CLK_CFG1.SYSCLKCFG register field drives this signal.
SYSCLKCFGSTATUS[3:0]	Input	A four-bit status signal, used to read the status of any external clock generation logic that drives SYSCLK clock. Use the CLK_CFG1.SYSCLKCFGSTATUS register field to read the values on this interface.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Description
AONCLKCFG[3:0]	Output	This control signal provides a four-bit signal that allows the system to configure an external clock generation logic that drives AONCLK clock. The CLK_CFG1.AONCLKCFG register field drives this signal.
AONCLKCFGSTATUS[3:0]	Input	A four-bit status signal, used to read the status of any external clock generation logic that drives AONCLK clock. Use the CLK_CFG1.AONCLKCFGSTATUS register field to read the values on this interface.

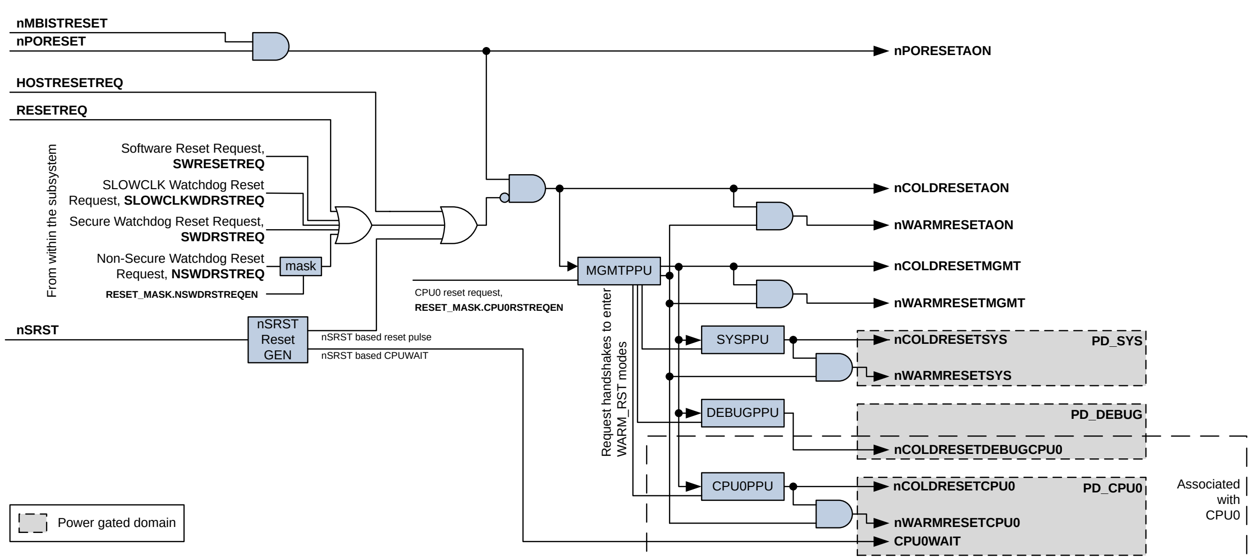
4.2 Reset infrastructure

The following figure shows the reset distribution in SSE-300. 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. **nPORESETAON** is also shown, which is not an output of the subsystem and only resets the RESET_SYNDROME register.

Note

The diagram does not show infrastructure for reset re-synchronisation and how the resets are used within each power domain.

Figure 4-4: Reset distribution in SSE-300



Functional reset interfaces

For the functional reset interfaces of SSE-300, see Functional integration resets.

4.2.1 Warm reset generation and control

The system provides a warm reset signal for each power domain. This reset is generated by first merging system reset requests from all processor cores in the system after masking each request with the related value in the RESET_MASK register. The merged and masked reset request is used to request all PPUs in the system to enter the WARM_RST power mode. Once all PPUs entered WARM_RST (which is the WARM_RST system power state), each warm reset signal is asserted simultaneously. The PPUs are not reset by warm reset, they are only reset when **nCOLDRESETAON** is asserted.

The expansion logic can delay the assertion of Warm reset, so that it can complete some critical operations before the reset occurs. For this purpose, the Power control P-Channel device interfaces can be used, through which the WARM_RST power mode is requested before the assertion of Warm reset.

Since each Warm reset is a superset of its counterpart Cold reset, if a Cold reset is asserted, the counterpart Warm reset is also asserted. In case of both resets being asserted, the related PPU that controls the power domain that the resets reside in is not requested to enter WARM_RST in advance. An example for counterpart resets and related PPU is **nWARMRESETCPU0**, **nCOLDRESETCPU0**, and CPU0PPU.

4.2.2 Power-on and Cold Reset Handling

The input **nPORESET** is the power-on reset input that 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**

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

- All watchdog timers' reset requests, 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.

nCOLDRESETAON resets almost all logic within the system except the RESET_SYNDROME register.

Note

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**.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.2.3 CPU Reset Handling

The CPU's **nPORESET** reset input is driven by the CPU<n>PPU controlling the CPU core power domain, that is PD_CPU<n>. This PPU itself 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 to stop 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 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 assertion, which includes the CPU's **nSYSRESET** input.

4.2.4 Boot after reset

After resets, including power-on reset, all cores boot using the values defined in the Initial Secure Reset Vector Register (INITSVTOR0) in the System Control Register Block as the boot address. The default address is configurable, but Arm recommends that the default is set to `0x01000000` which is mapped to code memory through the Master Code Main Expansion Interface. These addresses can be modified by software before subsequent warm re-boots 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 CPU<n>WAIT input to the core can force the processor to wait before executing the instruction. The processor in the system has an associated CPUWAIT.CPU<n>_WAIT register field that controls if it starts running its boot code when it wakes. There is also a CPU<n>WAITCLR input for each CPU<n> that allows an external entity to clear the associated CPU<n>_WAIT register bit.

4.3 CPU

SSE-300 supports one to four ARMv8-M MVE processor cores. Each processor can support different static configurations. The following that must be adhered to:

Note

NUMCPU value must be 0 for SSE-300.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 4-2: CPU Configurations that must be supported

Configuration	Value for CPU<n>	Related SSE-300 parameter	Description
SECEXT	1	SECEXT	Specifies whether the Armv8-M Security Extension is included: 0 - Security Extension not included 1 - Security Extension included
MPU_NS	> 0	CPU<n>_MPU_NS	Specifies the number of Non-secure MPU regions included.
MPU_S	> 0	CPU<n>_MPU_S	Specifies the number of Secure MPU regions included.
SAU	> 0	CPU<n>_NUM_SAU_CONFIG	Specifies the number of SAU Regions.
NUMIRQ	CPU<*n*>EXPNUMIRQ + 32	CPU<n>EXPNUMIRQ	Specifies the number of external interrupts included for each CPU, where n is 0 to NUMCPU.
IWIC	0	HASCPU<n>IWIC	Specifies whether the Internal Wake-up Interrupt Controller (IWIC) is included for each CPU, where n is 0 to NUMCPU: 0 - IWIC not included 1 - IWIC included
WICLINES	CPU<n>EXPNUMIRQ + 35	CPU<n>EXPNUMIRQ	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	If DEBUGLEVEL = 0 then 0, else 1	CPU<n>_CTI_PRESENT	Specifies whether the Cross-Trigger Interface (CTI) unit is included: 0 - CTI not included 1 - CTI included
BUSPROT	0	BUSPROT_PRESENT	Specifies whether interface protection is supported on the M-AXI, P-AHB, EPPB, and S-AXI interfaces of the processor: 0 - Interface protection not included 1 - Interface protection included
ITGU	1	ITGU_PRESENT	Specifies whether the processor is configured with ITCM Security gate unit: 0 - TCM Gate Unit not included 1 - TCM Gate Unit included
DTGU	1	DTGU_PRESENT	Specifies whether the processor is configured with DTCM Security gate unit: 0 - TCM Gate Unit not included 1 - TCM Gate Unit included

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Configuration	Value for CPU<n>	Related SSE-300 parameter	Description
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 1 - ECC included
LOCKSTEP	0	CPU<n>_LOCKSTEP	Specifies whether the processor is configured for dual-redundant lockstep operation. The options are: 0 - Regular processor operation 1 - Dual-redundant lockstep operation **Note:** When LOCKSTEP is set to 1, Arm recommends that RAR is also set to 1. Otherwise, software must initialize all registers to prevent accidental triggering of Dual-Core Lock-Step (DCLS) mismatch when the **UNKNOWN** state gets propagated to the top level. However, such software requirements might not be practical in certain software system environments. Therefore, the RAR option is strongly recommended for DCLS designs.
PMC	0	CPU<n>_PMC_PRESENT	Specifies whether the MBIST Controller (PMC-100) is included. 0 - PMC-100 not included 1 - PMC-100 included **Note:** PMC-100 is delivered as part of the optional licensable Safety package.
RAR	1	CPU<n>_RAR	Specifies if the synchronous states or the architectural required state is reset. The options are: 0 - Only reset the architectural required states 1 - Reset all synchronous states
DBGLVL	> 0	CPU<n>_DBGLVL	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: Full set. Four DWT and eight BPU comparators. Debug Monitoring mode and the Unpriviliged Debug Extension (UDE) is always supported. The Performance Monitoring Unit (PMU) is included when CPU<n>_DBGLVL is nonzero.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Configuration	Value for CPU<n>	Related SSE-300 parameter	Description
ICACHESZ[4:0]	ICACHESZ[0]=`0b1`	CPU<n>_INSTR_CACHE_SIZE	Specifies the inclusion of the instruction cache controller in the processor. If the instruction cache controller is included, CPU<n>_INSTR_CACHE_SIZE specifies the size of the cache. • CPU<n>_INSTR_CACHE_ SIZE[0]=0 - Instruction cache is excluded • CPU<n>_INSTR_CACHE_ SIZE[0]=1 - Instruction cache is included • If CPU<n>_INSTR_CACHE _SIZE[0]=1, then the cache sizes are: • CPU<n>_INSTR_CACHE_ SIZE[4:1]=`0b0000` 4 KB instruction cache • CPU<n>_INSTR_CACHE_ SIZE[4:1]=`0b0001` 8 KB instruction cache • CPU<n>_INSTR_CACHE_ SIZE[4:1]=`0b0011` 16 KB instruction cache • CPU<n>_INSTR_CACHE_ SIZE[4:1]=`0b0111` 32 KB instruction cache • CPU<n>_INSTR_CACHE_ SIZE[4:1]=`0b1111` 64 KB instruction cache
DCACHESZ[4:0]	DCACHESZ[0]=`0b1`	CPU<n>_DATA_CACHE_SIZE	Specifies the Master AXI (M-AXI) configuration and the inclusion and size of data cache controller. • CPU<n>_DATA_CACHE_SIZE[0]=0 - Area-optimized M-AXI data cache is excluded • CPU<n>_DATA_CACHE_SIZE[0]=1 - Performance-optimized M-AXI data cache is included. The cache sizes are: • CPU<n>_DATA_CACHE_SIZE[4:1]=`0b0000` 4 KB instruction cache • CPU<n>_DATA_CACHE_SIZE[4:1]=`0b0001` 8 KB instruction cache • CPU<n>_DATA_CACHE_SIZE[4:1]=`0b0011` 16 KB instruction cache • CPU<n>_DATA_CACHE_SIZE[4:1]=`0b0111` 32 KB instruction cache • CPU<n>_DATA_CACHE_SIZE[4:1]=`0b1111` 64 KB instruction cache
MVE	If FPU = 0 then MVE = 1	CPU<n>_MVE_CONFIG	M-profile Vector Extension (CPU<n>_MVE_CONFIG) parameter can have the following configuration options: 0 - MVE not included 1 - Integer subset of MVE included 2 - Integer and half single-precision floating-point MVE included. This option is only valid if CPU<n>_FPU_PRESENT=1. SSE-300 only supports the no M-profile Vector Extension when Floating Point Unit is configured.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

In the above table, the processor configurations ITGUBLKSZ, and DTGUBLKSZ are not defined. However, Arm recommends that ITGUBLKSZ and DTGUBLKSZ are equal to VMMPCBLKSIZE.

Each CPU also has several miscellaneous interface input signals that are tied as shown in the following table.

Table 4-3: CPU interface ties that are required

Interface signals	Value for CPU<n>	Related SSE-300 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: • `0b00001` - Alias bit = 24 • `0b00010` - Alias bit = 25 • `0b00100` - Alias bit = 26 • `0b01000` - Alias bit = 27 • `0b10000` - Alias bit = 28 • `0b00000` - No alias. TCM Security gating disabled.
CFGBIGEND	0	CFGBIGEND	Data endian format: 0 - Little-endian (LE) 1 - Byte invariant big-endian (BE8)
CFGITCMSZ[3:0]	Must not be `0b0000`	CPU<n>_CFGITCMSZ	Size of the instruction TCM region encoded as: • `0b0000` - No ITCM implemented, which is not supported by this specification • `0b0011` - $2^{CFGITCMSZ-1}$KB • Others - Reserved
CFGDTCMSZ[3:0]	Must not be `0b0000`	CPU<n>_CFGDTCMSZ	Size of the data TCM region encoded as: • `0b0000` - No DTCM implemented, which is not supported by this specification • `0b0011` - $2^{CFGDTCMSZ-1}$KB • Others - Reserved
CFGPAHBSZ[2:0]	`0b010`	CPU<n>_CFGPAHBSZ	Size of the P-AHB peripheral port memory region: • `0b000` - P-AHB disabled. • `0b001` - 64MB • `0b010` - 128MB • `0b011` - 256MB • `0b100` - 512MB **Note:** Setting CFGPAHBSZ to any other value results in UNPREDICTABLE behavior.
SAUDISABLE	0	CPU<n>_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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Interface signals	Value for CPU<n>	Related SSE-300 parameter	Description
WICCONTROL[0]	1	N/A	Setting to 1 causes the CPU to use WIC when in WFI DEEPSLEEP.
INITPAHBEN	1	INITPAHBEN	Setting to 1 to always enable P-AHB interface. This signal controls the reset value of PAHBCR.EN.
LOCKPAHB	1	CPU<n>_LOCKPAHB	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.

4.3.1 EVENT Interfaces

Each CPU also has event interfaces, which are **RXEV** and **TXEV**.

In SSE-300, all **TXEV**s are merged using logical OR and used to drive the **RXEV** of all CPUs.

Note

In this system, events do not wake a CPU from its EWIC based low power state, or makes it wake the system from Hiberation0 state.

SSE-300 does not support the use of WFE for the CPU entering DeepSleep state.

4.3.2 Interrupts

SSE-300 provides the following events that can generate interrupts within the system:

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

Depending on the configuration option values of CPU<*n*>EXPNUMIRQ and CPU<*n*>EXPIRQDIS, interrupts of each CPU<*n*> are made available to be driven through expansion logic. In this case, *n* is in 0 to NUMCPU.

The following table lists the interrupt map of each CPU<*n*>. Each CPU sees its own local CTIIRQ interrupts only. The software must ensure that all PPU interrupts 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 following table indicates which interrupts also act as wakeup interrupts at each CPU<*n*>'s associated External Wakeup Interrupt Controller (EWIC). The EWIC acts primarily as an entity that takes over the masking and holding of an interrupt, on behalf of the CPU<*n*>'s NVIC, when the CPU<*n*> is in its OFF or low power state and is unable on its own to handle interrupts. With the EWIC, the system can enter a lower power state that switches off each CPU along with most of

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

the system, except the EWIC itself. 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. An EWIC always exists with each CPU<*n*>.

Table 4-4: CPU <n> interrupt map, where n is 0 to NUMCPU

Interrupt input for CPU<*n*>	Interrupt source for CPU<*n*>	EWIC support
NMI	Combined Secure watchdog, SLOWCLK watchdog and **CPU<n>EXPNMI**	Yes
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]	MHU 0 CPU<n> interrupt. MHU interrupts are not shared, and each core only sees its own interrupt from the MHU. (Reserved if NUMCPU = 0). For more information, see Message Handling Unit.	Yes
IRQ[7]	MHU 1 CPU<n> interrupt. MHU interrupts are not shared, and each core only sees its own interrupt from the MHU. (Reserved if NUMCPU = 0). For more information, see Message Handling Unit.	Yes
IRQ[8]	CryptoCell™ interrupt. (Reserved if HASCRYPTO = 0)	Yes
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]	MGMT_PPU (Secure)	Yes
IRQ[15]	SYS_PPU (Secure)	Yes
IRQ[16]	CPU0_PPU (Secure)	Yes
IRQ[25:17]	Reserved	-
IRQ[26]	DEBUG_PPU (Secure)	Yes
IRQ[27]	Timer 3 AON	Yes
IRQ[28]	CPU<n>CTIIRQ0 (local CPU CTI only)	No
IRQ[29]	CPU<n>CTIIRQ1 (local CPU CTI only)	No
IRQ[31:30]	Reserved	-
IRQ[<CPU<n>EXPNUMIRQ +31>:32]	**CPU<n>EXPIRQ[CPU<n>EXPNUMIRQ-1:0]**. See Interrupt Interfaces.	Yes

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.4 System Interconnect infrastructure

The system interconnect infrastructure provides a bus infrastructure that transfers memory mapped access from bus masters to slaves in the system. SSE-300 defines two key interconnects that form the System Interconnect:

Main Interconnect

This interconnect is planned to provide 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-M MVE based processors have a direct interface to access this interconnect.

The System Interconnect is able to achieve the following:

- Transfers the following key properties of the access, from a master to a slave. The slave is required to ensure that any security related infrastructure can have all the following necessary information to make decisions on access rights:
    - Security attribute indicating an access is marked as Secure or Non-secure
    - The privilege level of the access, which is either Privileged or Unprivileged, or of a similar type
    - Bus access address, read and write attribute, and optionally, master identity
- Complete an access that has been issued. At completion of an access, minimally communicate as a response that the access is successful or has failed.
- Support the ability to atomically read and modify a memory location

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.

4.4.1 ACC_WAIT Control

SSE-300 provides a set of controls to allow the following:

- Add access control gates, driven using the **ACCWAITn** signal

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Block access to the system when the system leaves Hibernation state, is reset, at first powering up, or when software wants to reconfigure security settings in the system. This prevents access to the system until all security related features of the system has been set up correctly. Once software is ready, it can release the gates by writing to the BUSWAIT.ACC_WAITN register. Then it can check the current status of all external gating units by reading the **ACCWAITNSTATUS** signal value through the BUSWAIT.ACC_WAITN_STATUS register.

4.5 Volatile Memory

SSE-300 supports multiple volatile memory banks, that are VM0 and VM1.

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

- The size of each is powers of two
- The size of each bank is defined using the VMADDRWIDTH configuration point 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 VMs support Exclusive Accesses from the processors and external masters.

Each VM has a *Memory Protection Controller* (MPC) associated with it that provides the ability to 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-300, all VMs use the same MPC block size as defined by the configuration VMMPCBLKSIZE. Each MPC is implemented so that out of reset, all memory locations are mapped to the Secure world by default.

SSE-300 supports a power domain for each bank: PD_VMR0, PD_VMR1. 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. Hence, any power gating or retention refers only to the memory itself. For more information about related power control, see Power integration.

All VMs run on SYSSYSCLK.

4.6 Timers and Watchdogs

This section describes the two main classes of timers and watchdogs in SSE-300.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.6.1 Timestamp based Timers

The first class of timers and watchdogs are timestamp based. They utilize the timestamp value provided on the System Timestamp Interface.

For more information about the System Timestamp Interface, see System Timestamp Interface.

The Timestamp-based Timer supports the following:

- Memory-mapped System Timer with register access through the Peripheral Interconnect
- Works with 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 Timestamp-based Watchdog timer is a simplified timer that supports the following:

- Memory-mapped System Timer with register access through the Peripheral Interconnect
- Works with 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.

Four Timestamp Timers are provided along with 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`

Of these timers, 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 are controlled through the Secure Access Configuration Register Block. For more information, see Secure Access Configuration Register Block. Both watchdogs are permanently assigned for security with is one Non-secure and another Secure. All timers and watchdog timers generate interrupts, and the Non-secure Watchdog can generate an additional 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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**. Hence, 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.

4.6.2 SLOWCLK AON Timers

The second class of 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. These components 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 privilege or unprivileged access through the Peripheral Protection Controllers that 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*.

4.7 Message Handling Unit

When NUMCPU > 0, SSE-300 implements two Message Handling Unit (MHU) to allow CPUs to interrupt each other to pass message. Two MHUs are provided so the software can place one MHU in the Secure world and one in the non-secure world. Both MHUs reside in the PD_SYS power domain and are reset using the **nWARMRESETSYS**.

For information about the MHU registers, see Message Handling Unit registers.

When NUMCPU = 0, there are no MHUs in the system.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.8 Power Policy Units

The PPUs are mapped to Secure address space as defined in System Control Peripheral Region.

See Power Policy Units for more information.

4.9 Peripheral Protection Controllers

Peripheral Protection Controllers in the system enable the software to control if a peripheral is accessible to the Secure or Non-secure world, and to 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-300, peripherals that are aliased to two memory areas, one Secure and one Non-secure, are protected by PPCs. And the PPC defines which region the peripheral resides in.

In SSE-300, Timers 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 them, 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-300, 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
  - Watchdog Refresh Frames

  Note

  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

Note

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

For more information, see Secure Access Configuration Register Block.

4.10 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.

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-300, each memory page that is protected by the MPC is aliased to two memory areas: a Secure and a 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.

4.11 CryptoCell

SSE-300 has the HASCRYPTO configuration option for cryptographic functions.

For supported CryptoCell configurations, see HASCRYPTO.

- HASCRYPTO = 0. In this no-crypto configuration, the CryptoCell-312 IP and its associated integration logic does not exist within the system.
- HASCRYPTO = 1. In this has-crypto configuration, the CryptoCell-312 IP and its associated integration logic exists within the system.

4.11.1 No-Crypto Configuration

When HASCRYPTO = 0, CryptoCell-312 does not exist in the system. No associated interfaces or configuration that are associated with CryptoCell-312 require to exist.

In a such a system, the root of trust can still be within the system. If TBSA-M compliance is still required, the system integrator must ensure that the integrated system contains all the necessary resources that a root of trust requires.

4.12 Debug Infrastructure

SSE-300 has the HASCSS configuration option for debug infrastructure configurations.

For supported debug infrastructure configurations, see HASCSS.

- HASCSS = 0. In this basic debug configuration, a CoreSight SoC-600 based common debug infrastructure is not defined and does not exist. When HASCSS = 0, NUMCPU must be 0.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- HASCSS = 1. In this full debug configuration, a CoreSight Soc-600 based common debug infrastructure exists.

4.12.1 Basic Debug Configuration

When HASCSS = 0 and EXPLOGIC_PRESENT = 1, the subsystem expansion includes an example debug infrastructure that instances DAP-Lite2, debug timestamp generator, Cortex-M55 TPIU, and MCU debug ROM table. This infrastructure is the source and target of some debug related interfaces of each core. The remaining interfaces are made available as expansion interfaces. The following subsections describe the example debug infrastructure.

When HASCSS = 0 and EXPLOGIC_PRESENT = 0, all debug related interfaces of each processor core are made available as expansion interfaces.

For more information, see Debug and trace related interfaces.

4.12.1.1 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.

The debug power up handshake interface - including the cdbgpwrupreq and cdbgpwrupack signals - connect to PD_DEBUG and PC_CPU0 power control logic. cdbgpwrupreq is combined into the PWRDEBUGWAKEQACTIVE and PWRCPU0WAKEQACTIVE wake up power Q-channels as well, therefore, the debug power handshake interface is capable to keep up and wake up the PD_DEBUG and PD_CPU0 power domains.

The system power up and debug reset handshake interfaces of the DAP-Lite2 are unused.

The **DAPACCEN** and **DAPDSSACCEN** debug authentication signals control the access to the processor's D-AHB interface. **DAPACCEN** drives both ap_en and ap_secure_en inputs of DAP-Lite2. When **DAPACCEN** is LOW, any memory access result in authentication failure as described in *Arm® CoreSight™ DAP-Lite2 Technical Reference Manual*. When **DAPDSSACCEN** is LOW, the SIE-200 AHB to AHB and APB bridge, residing between DAP-Lite2 and the D-AHB interface of the processor, responds with error.

Warm reset can always be applied safely when there are no active transactions on the D-AHB interface of Cortex-M55. Therefore, the SIE-200 AHB to AHB and APB bridge stalls the transactions when PD_MGMT enters WARM_RST power mode.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The DAP-Lite2 resides in the PD_AON power domain.

The DAP-Lite2 has the following reset inputs:

- power up reset, driven by **nPORESET**
- JTAG TAP reset, driven by the logical AND of **nTRST** and **nPORESET**
- AHB master interface reset, driven by the synchronously deasserted version of **nCOLDRESETAON**

The DAP-Lite2 has the following clock inputs:

- Serial Wire and JTAG interface clock, driven by **SWCLKTCK**
- AHB master interface clock, driven by AONCLK

4.12.1.2 Debug timestamp generator

The debug timestamp generator is implemented as a 64 bit binary up counter enabled when CPU0TRCENA = 1.

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

4.12.1.3 Cortex-M55 TPIU

The Cortex-M55 *Trace Port Interface Unit* (TPIU) 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 Cortex-M55 TPIU is accessible through the Private Peripheral Bus Region at address `0xE0040000` to `0xE0040FFF`.

See *Arm® Cortex®-M55 Technical Reference Manual* for more information about Cortex-M55 TPIU.

The Cortex-M55 TPIU resides in the PD_DEBUG power domain, runs on DEBUGCPU0CLK, and resides in the nCOLDRESETDEBUGCPU0 reset domain.

4.12.1.4 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.13 System and Security Control

SSE-300 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 Master 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 147
System Control Register Block on page 151
Secure Access Configuration Register Block on page 121
Non-secure Access Configuration Register Block on page 135

4.14 Power integration

The following sections describe power integration in SSE-300.

4.14.1 Power integration overview

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 as well as the control mechanisms supporting the coordination of the operation of these different domains.

SSE-300 supports the Intermediate Level Power Control Infrastructure (PILEVEL = 1), which is detailed in this section.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Power domain hierarchy and bounded regions introduces terms that are used in the power related descriptions.

Power domains depicts the distribution of SSE-300 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.

Wake-up sources 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.

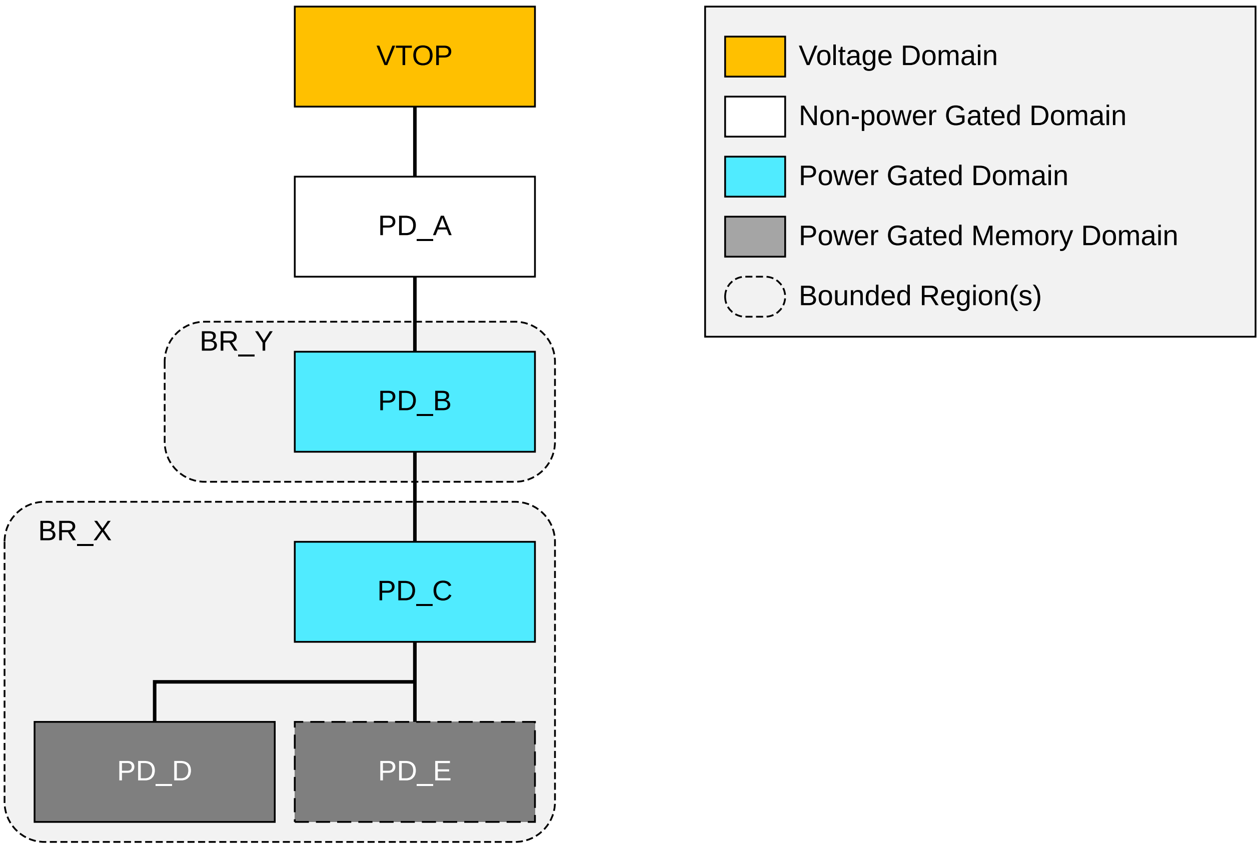
System Power States describes the valid power state combinations of power domains in all Bounded Regions.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.14.2 Power domain hierarchy and bounded regions

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

Figure 4-5: Example power domain hierarchy



In the figure above, 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:

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

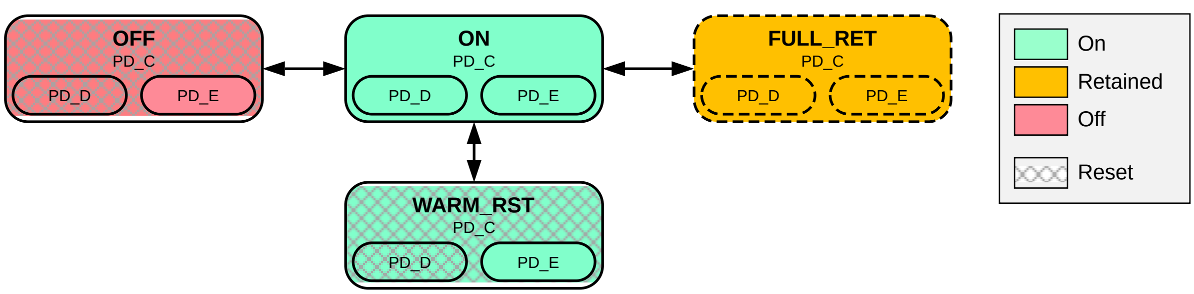
The power modes of a Bounded Region are represented using a state transition diagram that shows the valid power state combinations and transitions of power domains in the Bounded Region.

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 diagram below 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).

Note

PD_E is never reset as it is a memory power domain.

Figure 4-6: Example power mode transition diagram of bounded power regions PD_B, PD_C, and PD_D



These mode diagrams are very similar to PPU power mode transition diagrams. However, the diagrams here, and their modes do not indicate that a power mode in the PPU with the same name is used.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.14.3 Power domains

SSE-300 is partitioned into multiple power domains.

These power domains are depicted in Figure 4-8: SSE-300 power domains in the logical integration layer on page 55 and Figure 4-9: SSE-300 power domains in the power aware integration layer on page 56.

Layers representing the logic power domains are defined in Figure 4-7: SSE-300 power domain layers on page 54.

Figure 4-7: SSE-300 power domain layers

Note

The following diagrams represent the EXPLOGIC_PRESENT = 1 configuration of the subsystem, where the expansion logic is present.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-8: SSE-300 power domains in the logical integration layer

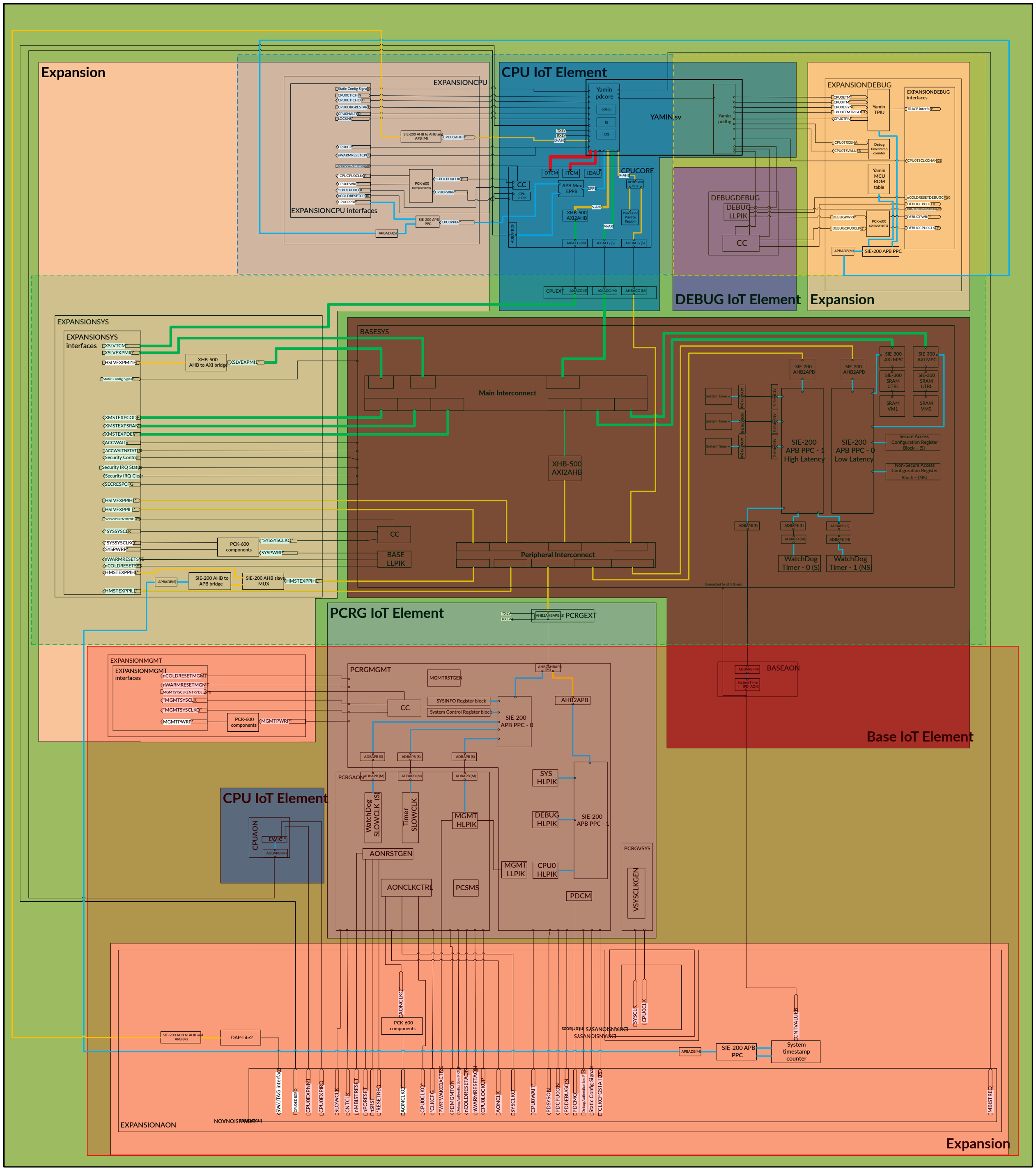
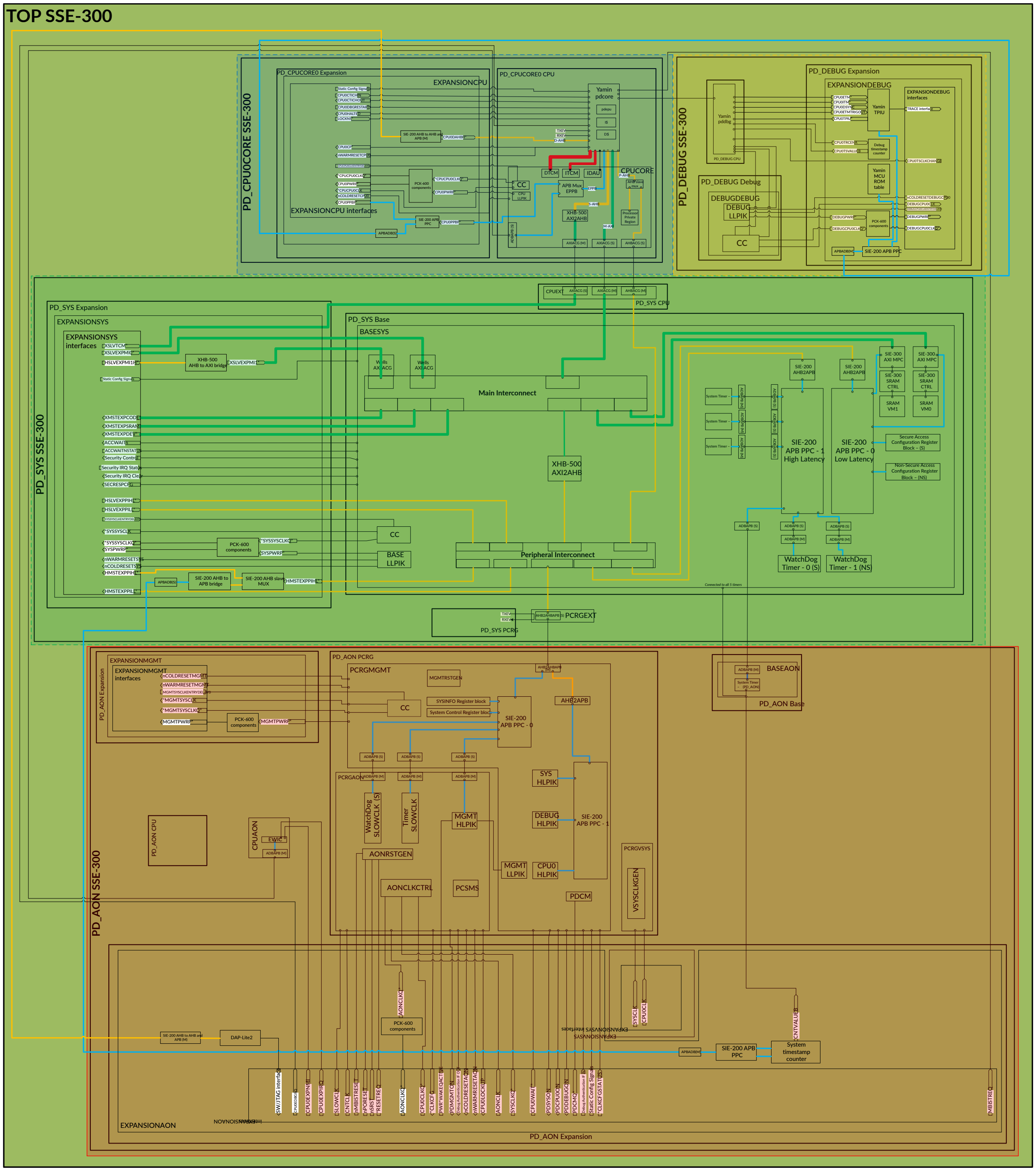


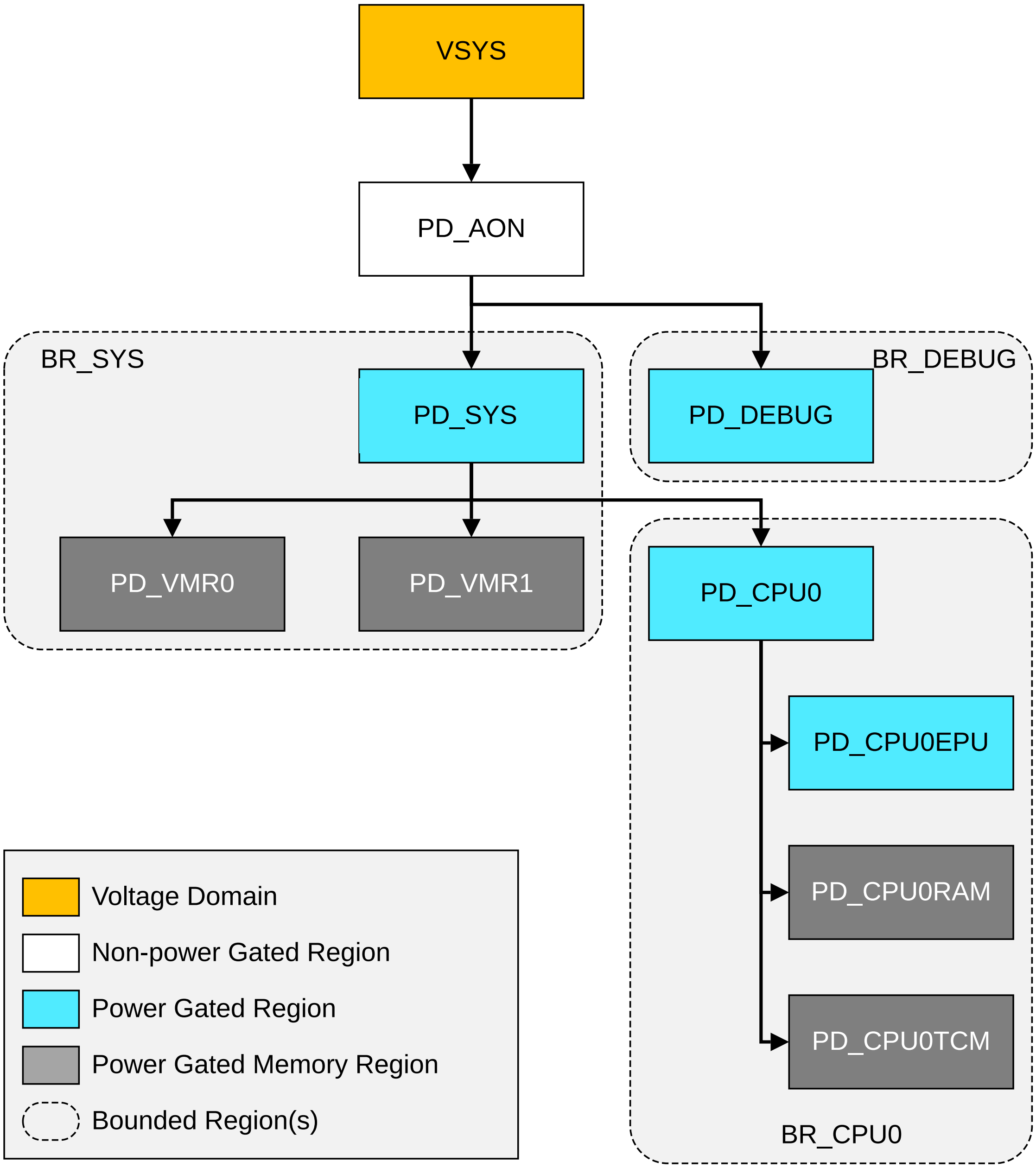
Figure 4-9: SSE-300 power domains in the power aware integration layer



These power domains are hierarchical and Figure 4-10: Voltage and power domain hierarchy of SSE-300 on page 57 shows the voltage and power domain hierarchy. For more information regarding how the power domains are controlled, see Power Policy Units.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-10: Voltage and power domain hierarchy of SSE-300



4.14.4 Power Policy Units

The SSE-300 leverages *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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Power on reset, Cold reset, and Warm reset related state transitions and reset generation in the subsystem.

Device interface handshake is performed by all PPUs when transitioning to WARM_RST by default. All PPUs are Cold reset and reside in the power domain PD_AON. The following table lists additional configuration of the PPUs 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 4-5: 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_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

The Default Perform Device Interface Handshake When Transition From ON to WARM_RST Mode Enable configuration is 1 for all PPUs.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

While the default power mode of most PPUs is OFF, a hardware autonomous power-up sequence is implemented by the subsystem. See Power-up after Cold reset for more information.

Write accessibility of PPU registers are 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`

Access to the PPUs is not required for normal operation because SSE-300 is architected to use the PPUs primarily in their default dynamic mode, where the request to enter or leave a power state or mode is managed using only the PPU's Device interface, and does not require the programming of the PPUs. Hence the only time access to PPUs might be required is for Debug purposes.

When PWRCTRL.PPU_ACCESS_FILTER is set to '0', all PPU registers are freely accessible to the secure world. Turning off PPU access filtering and controlling the PPU registers other than the foregoing IRQ related registers can lead to system deadlock. It is for advanced users only and should be used only at your own risk.

4.14.5 Bounded Region power modes

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

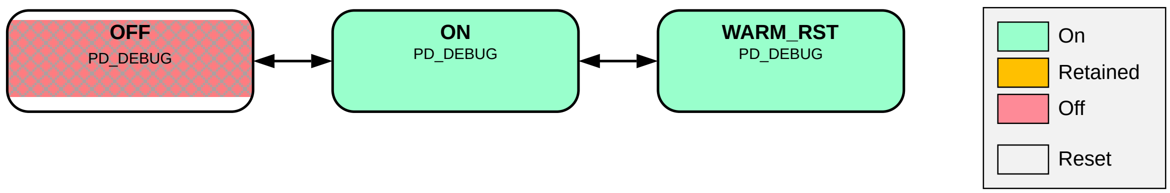
4.14.5.1 BR_DEBUG power modes

The following figure shows the the power modes supported by BR_DEBUG. 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.

Note

The PD_DEBUG itself is not warm reset in the WARM_RST Power Mode.

Figure 4-11: BR_DEBUG power mode transition diagram

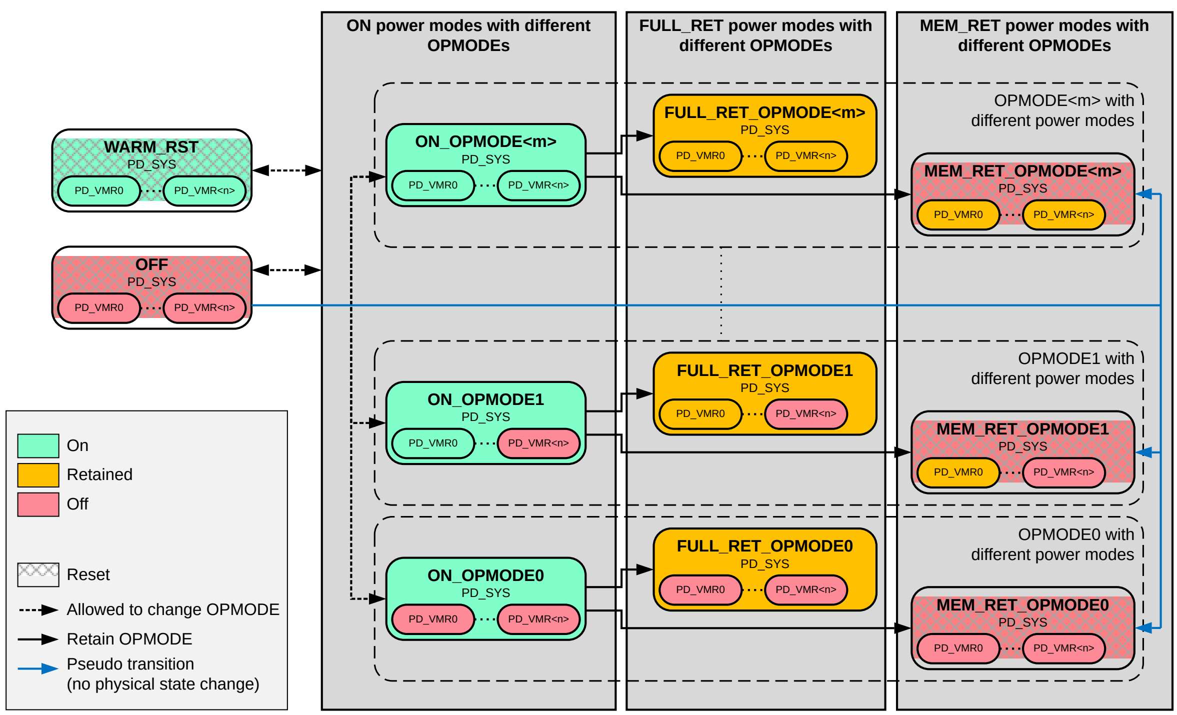


4.14.5.2 BR_SYS power modes

Power modes that BR_SYS supports.

The following figure shows the power modes that BR_SYS supports.

Figure 4-12: BR_SYS power mode transition diagram



Note

During pseudo power mode transitions, the (physical) power state of the power domains does not change. The power domains themselves (for example PD_SYS) can observe pseudo transitions upon the CPU0PPU exiting the various MEM_RET power modes.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 such state changes are performed that are part of the transitions between the various MEM_RET and ON power modes. The PPU and the PCSM do not enter the OFF power mode as part of the pseudo transition.

In SSE-300, 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.

For more information, see Power mode remapping.

BR_SYS has $2^{NUMVMBANK}$ different OPMODEs (m is 3 in Figure 4-12: BR_SYS power mode transition diagram on page 60). OPMODEs are encoded as binary 2-bit values with each bit representing the power state of a memory power domain, PD_VMR<i>, where i is 0 to NUMVMBANK-1. Other than entering and exiting the OFF or WARM_RST power modes, changing OPMODEs is possible only when transitioning between the different ON power modes.

For example, for with NUMVMBANK = 2, which is supported by SSE-300:

- OPMODE0: PD_VMR0 and PD_VMR1 are both OFF.
- OPMODE1: PD_VMR0 is ON and PD_VMR1 is OFF.
- OPMODE2: PD_VMR0 is OFF and PD_VMR1 is ON.
- OPMODE3: PD_VMR0 and PD_VMR1 are both ON.

When a warm reset is requested, the bounded region transitions to WARM_RST through other power modes once the PD_SYS power domain is idle and ready for being warm reset.

4.14.5.2.1 Controlling the PD_VMR<i> minimum power states

To configure the minimum power state of each PD_VMR<i>, SW 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 be 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.

  Hence, 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 anaccess 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>.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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 OFF. Therefore, in the low power System Power States, all states in PD_VMR<i> are retained. BR_SYS never transitions to a power mode that has PD_VMR<i> turned off.

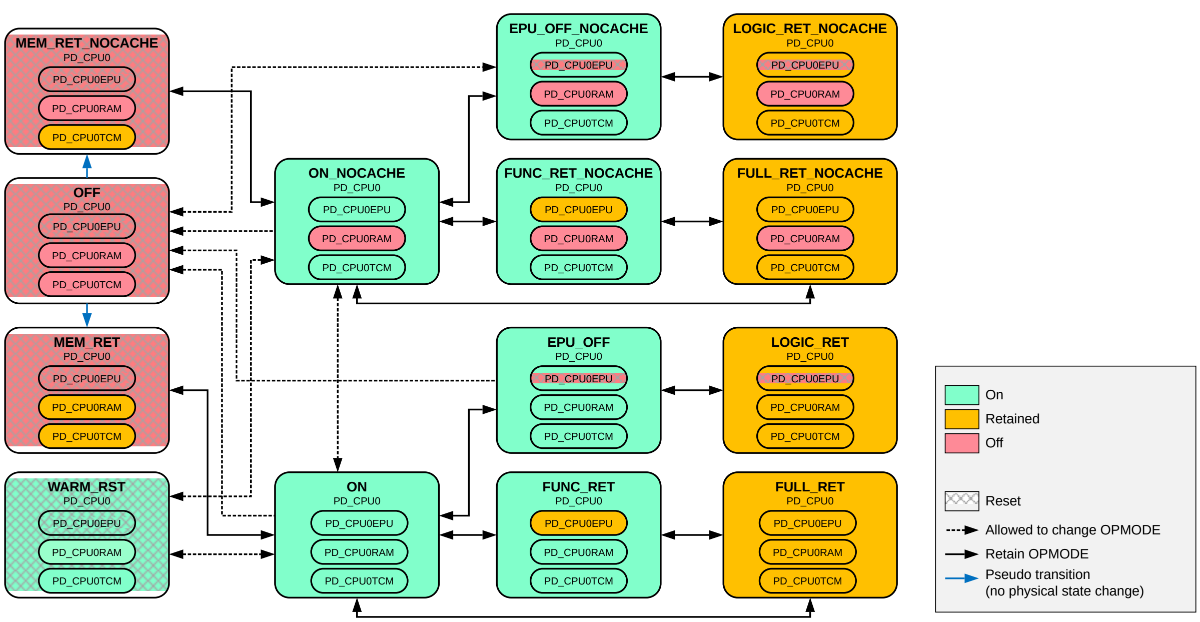
Note

To place the PD_VMR<i> into Retention, the BR_SYS has to enter one of the FULL_RET_OPMOE<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.

4.14.5.3 BR_CPU0 power modes

The following figure shows the power modes supported by BR_CPU0.

Figure 4-13: BR_CPU0 power mode transition diagram



Note

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 upon the CPU0PPU exiting 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 such state changes are performed that are part of the

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

transitions between the various MEM_RET and ON power modes. The PPU and the PCSM do not enter the OFF power mode as part of the pseudo transition.

In SSE-300, 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.

For more information, see Power mode remapping.

BR_CPU0 uses a different name (EPU_OFF) for the MEM_OFF encoding in the SSE-300 PCSA as the corresponding Power Mode affects PD_CPU0EPU, which is logic and not memory.

The PD_CPU0TCM power state always matches that of PD_CPU0, except for 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 CPU's local TCM minimum power state register configuration CPU0PWRCFG.TCM_MIN_PWR_STATE.

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.

4.14.5.3.1 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 NONCACHE operating modes, the following registers must be configured:

- CPDLPSTATE.RLPSTATE register in the CPU set to OFF, which is `0b11`
- MSCR.DCACTIVE register in the CPU set to `0b0` to disable Data Cache
- MSCR.ICACTIVE register in the CPU set to `0b0` to disable Instruction Cache

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.

4.14.5.3.2 Controlling PD_CPU0EPU power state

Other than WARM_RST state, the BR_CPU0 bounded region provides the ability for the PD_CPU0EPU to enter a lower power state independently while the PD_CPU0 is ON.

To control if the PD_CPU0EPU can be allowed to enter the retention or off state, the software can set the register CPDLPSTATE.ELPSTATE in the CPU as follows:

- Set CPDLPSTATE.ELPSTATE to RET, `0b10` to allow the EPU to enter retention state only when in low power state. When set to RET, BR_CPU0 never enters EPU_OFF, EPU_OFF_NOCACHE, LOGIC_RET_NOCACHE, LOGIC_RET, OFF, and MEM_RET states.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Set CPDLPSTATE.ELPSTATE to OFF, `0b11` to allow the EPU to enter OFF state only when in lower power state. When set to OFF, BR_CPU0 never enters FUNC_RET, FUNC_RET _NOCACHE, FULL_RET_NOCACHE and FULL_RET states.
- Set CPDLPSTATE.ELPSTATE to ON, which is `0b00` or `0b01`, the EPU is never allowed to be turned off or retained when the PD_CPU0 is ON. Hence BR_CPU0 never enters any power modes to the right of the power modes ON_NOCACHE or ON in the Figure "BR_CPU0 power mode transition diagram" in BR_CPU0 power modes except for FULL_RET and FULL_RET_NOCACHE.

4.14.5.3.3 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 as follows:

- Set CPDLPSTATE.CLPSTATE to RET, to allow the PD_CPU0 to enter retention state only when in low power state. When set to RET, BR_CPU0 never enters OFF, MEM_RET_NOCACHE or MEM_RET.
- Set CPDLPSTATE.CLPSTATE to OFF, to allow the PD_CPU0 to enter off state only when in low power state. When 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.
- Set CPDLPSTATE.CLPSTATE to ON, that is `0b00` or `0b01` to keep PD_CPU0 on, which means that 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 requires to enter WFI DeepSleep state. In SSE-300, DeepSleep always utilises a WIC. External WIC for the CPU always exists, and the External WIC resides in the PD_AON domain. HASCPU0IWIC is `0b0` in SSE-300, so for CPU0, the Internal WIC does not exist, and the External WIC 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 external WIC to wake the CPU. To enter this state, the CPU must select to use the External WIC through the CPU0PWRCFG.USEIWIC register, set the CPU0's CPDLPSTATE.CLPSTATE to OFF and enable DeepSleep, before entering WFI.
- The “RET – DeepSleep” state allows the CPU to enter retention state and utilizes interrupts through the external WIC to wake the CPU. To enter this state, the CPU must select to use the External WIC through the CPU0PWRCFG.USEIWIC register, set CPDLPSTATE.CLPSTATE to RET and 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 External WIC is used to wake the CPU. To enter this state, the CPU must set CPDLPSTATE.CLPSTATE to ON, enable DeepSleep before entering WFI. The CPU must select to use the External WIC through the CPU0PWRCFG.USEIWIC register prior to entering WFI.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

4.14.5.4 Power mode remapping

In SSE-300, the legal range is 0 for the render parameter LOGIC_RETENTION_PRESENT representing an implementation without retention-capable cells.

PPU power modes with logic being retained are re-mapped to power modes with logic being ON in the PCSMs so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells. The programmers model and the software view of the subsystem is not altered due the remapping being applied in the PCSMs. The remapping does not alter the default behavior of the isolation control. This results in power states with the logic and the related isolation being ON and active respectively (for example see the columns PD_CPU0EPU power state and PD_CPU0EPU isolation in the row BR_CPU0 power mode FUNC_RET in the following table). The foregoing behavior of the isolation does not corrupt the functional behavior of the subsystem.

The remapping applied to the PPUs SYSPPU and CPU0PPU are defined in the following tables.

Table 4-6: BR_CPU0 power mode remapping

CPU0PPU power mode	BR_CPU0 power mode	PD_CPU0EPU power state	PD_CPU0EPU isolation	PD_CPU0 power state	PD_CPU0 isolation	PD_CPU0RAM power state	PD_CPU0TCM power state
WARM_RST	WARM_RST	ON	Inactive	ON	Inactive	ON	ON
ON	ON	ON	Inactive	ON	Inactive	ON	ON
ON	ON_NOCACHE	ON	Inactive	ON	Inactive	OFF	ON
FUNC_RET	FUNC_RET	ON	Active	ON	Inactive	ON	ON
FUNC_RET	FUNC_RET_NOCACHE	ON	Active	ON	Inactive	OFF	ON
MEM_OFF	EPU_OFF	OFF	Active	ON	Inactive	ON	ON
MEM_OFF	EPU_OFF_NOCACHE	OFF	Active	ON	Inactive	OFF	ON
FULL_RET	FULL_RET	ON	Active	ON	Active	ON	ON
FULL_RET	FULL_RET_NOCACHE	ON	Active	ON	Active	OFF	ON
LOGIC_RET	LOGIC_RET	OFF	Active	ON	Active	ON	ON
LOGIC_RET	LOGIC_RET_NOCACHE	OFF	Active	ON	Active	OFF	ON
MEM_RET	MEM_RET	OFF	Active	OFF	Active	RET	RET
MEM_RET	MEM_RET_NOCACHE	OFF	Active	OFF	Active	OFF	RET
OFF	OFF	OFF	Active	OFF	Active	OFF	OFF

The CPU0PPU operating mode is OPMODE_02 in the following power modes:

- ON_NOCACHE

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- FUNC_RET_NOCACHE
- EPU_OFF_NOCACHE
- FULL_RET_NOCACHE
- LOGIC_RET_NOCACHE
- MEM_RET_NOCACHE

The CPU0PPU operating mode is OPMODE_03 in the following power modes:

- ON
- FUNC_RET
- EPU_OFF
- FULL_RET
- LOGIC_RET
- MEM_RET

Note

The power modes OFF and WARM_RST do not have operating mode context.

Table 4-7: BR_SYS power mode remapping

SYSPPU power mode	BR_SYS power mode	PD_SYS power state	PD_SYS isolation	PD_VMR1 power state	PD_VMR0 power state
WARM_RST	WARM_RST	ON	Inactive	ON	ON
ON	ON_OPMODE3	ON	Inactive	ON	ON
ON	ON_OPMODE2	ON	Inactive	ON	OFF
ON	ON_OPMODE1	ON	Inactive	OFF	ON
ON	ON_OPMODE0	ON	Inactive	OFF	OFF
FULL_RET	FULL_RET_OPMODE3	ON	Active	ON	ON
FULL_RET	FULL_RET_OPMODE2	ON	Active	ON	OFF
FULL_RET	FULL_RET_OPMODE1	ON	Active	OFF	ON
FULL_RET	FULL_RET_OPMODE0	ON	Active	OFF	OFF
MEM_RET	MEM_RET_OPMODE3	OFF	Active	RET	RET
MEM_RET	MEM_RET_OPMODE2	OFF	Active	RET	OFF
MEM_RET	MEM_RET_OPMODE1	OFF	Active	OFF	RET
MEM_RET	MEM_RET_OPMODE0	OFF	Active	OFF	OFF
OFF	OFF	OFF	Active	OFF	OFF

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

The power modes OFF and WARM_RST do not have operating mode context.

4.14.6 Wake-up sources

Contents of this section are to be provided in an upcoming release.

4.14.7 Power Dependency Control

The following table shows the Power Dependency Control Matrix (PDCM) table and how the PD_SYS and PD_VMR<i> are affected by the power dependency inputs.

Table 4-8: Power dependency control matrix

Power domain	Power dependency input		
	PD_SYS_ON	PD_CPU0_ON	PDCMQREQn[{0-<PDCMQCHWIDTH-1>}]
PD_SYS	Conf	Y	Conf
PD_VMR0	-	Conf	Conf
PD_VMR1	-	Conf	Conf

The left column of the table lists the power domains that are being controlled. The heading row lists the Power dependency inputs. Power dependency inputs are either:

- The "ON" state of power domains in the system. For example, PD_SYS_ON means PD_SYS is ON when asserted
- The Power Dependency Control Matrix QREQn inputs, PDCMQREQn, that are driven by expansion logic from outside the subsystem indicating a keep-up request from external power domains

"Conf" indicates that it is software configurable, "Y" indicates that it is always sensitive to the respective dependency input. For example, PD_SYS can be software configured to be sensitive to the ON state of PD_SYS and all Expansion Power Control Dependency inputs, and it is always sensitive to the ON state of PD_CPU0. If a power domain is sensitive to a dependency input, it means that once the power domain being controlled is already ON, if any of the dependency inputs is "ON" or true, then the power domain remains ON. Therefore, the PDCM is used primarily to define when a power domain should not enter a lower power state. PDCM supports keeping up power domains, it is not designed to support powering up of any power domain.

PD_VMR<i> can also be configured to be sensitive to a power domain. For example, if PD_VMR0 sensitivity is configured by software to be sensitive to PD_CPU0_ON, if PD_CPU0 is ON, the memory also remains ON. However, PD_VMR0 is controlled using the same PPU as PD_SYS and as a result of Figure 4-12: BR_SYS power mode transition diagram on page 60 shown in BR_SYS power modes, whenever PD_VMR0 is ON, PD_SYS also remains in one of the ON_OPMODE

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

states where PD_VMR0 is ON. PD_SYS can also be configured to be sensitive to itself. When the software configures a domain to be sensitive to self, the domain remains ON once it is ON.

The intention of the PDCM and all other sensitivity defined for each power domain is to allow, as much as possible for the power control of the System to be performed primarily using dynamic power transitions. This reduces the amount of software interactions needed for system management and therefore improves system responsiveness and contributes to further power reduction.

The SSE-300 provides programmable registers for the PD_SYS and each PD_VMR<i> power domains (PDCM_PD_SYS_SENSE, PDCM_PD_VMR<i>_SENSE), which define the lowest power state that each domain can enter when entering its lowest power state. The minimum power states of the power domains PD_DEBUG and PD_CPU0 are not affected by the PDCM registers.

Table 4-9: Power domain minimum power state

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

The MIN_PWR_STATEs of both PD_SYS and PD_VMR<i> affect the SYSPPU. For example, if PD_SYS is idle and all domains that PD_SYS depends on are not ON, the SYSPPU tries to enter the bounded region collectively to a low power state. 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 MIN_PWR_STATE of PD_VMR0 is "ON" and MIN_PWR_STATE of all other PD_VMR<i> is OFF, then if currently all other PD_VMR<i> are ON, when SYSPPU is entering a low power state, it transitions to ON_OPMODE1 state to turn off all other PD_VMR<i>. It never enters FULL_RET_OPMODE1 or MEM_RET_OPMODE1.

4.14.8 System power states

By the relationships and minimum power states defined in the various Power Dependency Control Matrix Sensitivity registers and the low power control registers within the CPU, the following System Power States can be defined.

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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

SYS_ON

The system is ON.

In SSE-300, logic retention power states are re-mapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells. For more information see Power mode remapping.

The following table defines the following for each System Power State:

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

State combinations not tabulated are not supported by the subsystem. See Entering lower PD_CPU0 power states for the definition of the states in the PD_CPU0 column.

Table 4-10: System power states

System power state	VSYS(PD_AON)	PD_SYS	PD_CPU0	PD_DEBUG	PD_VMR<i>	Clocks
SYS_OFF	OFF	OFF	OFF	OFF	OFF	OFF
HIBERNATION0	ON	OFF	OFF – DeepSleep[2]	OFF/ON	OFF/RET	ON/OFF[1]
SYS_RET	ON	RET	OFF - DeepSleep[2]	OFF/ON	OFF/RET	ON/OFF[1]
			RET - DeepSleep			
SYS_ON	ON	ON	OFF - DeepSleep[2]	OFF/ON	OFF/RET/ON	ON/OFF[1]
			RET - DeepSleep			
			ON - DeepSleep			
			ON - Sleep			
			ON			

1. Clocks are requested by the subsystem based on the Clock Force register and the power states of PD_DEBUG, PD_SYS 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 the section Clock gating control of input clocks.
2. When PD_CPU0 is in the state OFF-DeepSleep, PD_CPU0TCM can be either OFF or retained as defined by CPU0PWRCFG.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.

From the above table the following points can be made:

- 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, since 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

An additional transient WARM_RST state also exists for the system when a warm reset is being 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.

4.14.8.1 Power-up after Cold reset

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

Due to the hierarchical power management, SYSPPU and CPU0PPU wake up in turn, and DEBUGPPU simultaneously with SYSPPU. When the power-up sequence completes, the DEBUGPPU turns off if there is no request towards PD_DEBUG.

Power modes entered during the power-up by SYSPPU, CPU0PPU, and DEBUGPPU are ON_OPMODE3, EPU_OFF_NOCACHE, and ON in turn.

4.14.8.2 Entering HIBERNATION0

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

- 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 SECMPCINTSTAT 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, either of the following must be followed:
    - Sensitivity to the PD_CPU0 ON state must be set in the related PDCM_PD_VMR<i>_SENSE register prior to entering HIBERNATION0. This ensures that upon first access to the VM after leaving HIBERNATION0 that is causing the VM to power up, VM stays powered until PD_CPU0 turns OFF.
    - The minimum power state in the PDCM_PD_VMR<i>_SENSE register must be set 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
- PD_CPU0 must enter the “OFF-DeepSleep” (with EWIC enabled) if the intention is for the subsystem to be woken by EWIC. See Entering lower PD_CPU0 power states for more information.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.14.8.3 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. If this does not wake PD_CPU0 directly, accesses to the subsystem can result in a wake interrupt on the EWIC.

Note

SYSCLK can not be running during HIBERNATION0, and a wake request on the **PWRSYSWAKEQACTIVE** input automatically results in a request for SYSCLK to be active.

When the Base IoT Element in 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. In many cases, this necessitates to also wake and boot the CPU so that it can configure the subsystem before allowing accesses for Expansion masters into the subsystem. Consequently, using only **PWRSYSWAKEQACTIVE** as a means of waking the subsystem from HIBERNATION0 is limited and not recommended.

Contrarily, using interrupt signals on the EWIC provides a more robust approach. 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 masters. To delay access from Expansion masters, the system integrator must deploy access control gates at the slave expansion IFs, which is facilitated by the register BUSWAIT, the configuration ACCWAITNRST, and the signal **ACCWAITn**.

Provided that an Expansion master intends to wake and access peripherals or memories within SSE-300 without waking the CPU as well, it must be ensured that the Expansion master accessing the subsystem is a Secure master. Alternatively, a non-Secure master can be restricted externally to access a strictly controlled region of memory residing outside of the SSE-300, which is always non-Secure and hence does not pose a security risk. Mind that when the subsystem wakes without the CPU restoring the configuration of all MPCs and PPCs in the subsystem, Secure masters in the Expansion see all Non-secure memory spaces as Secure. Care must be taken to ensure that if any of these memories were retained before wake-up, these memory locations are not used for code execution.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5 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 Master Interface

An AMBA interface that is described as a master interface is one where the subsystem is the master and must be connected to a slave interface. For an AXI master interface the **ARVALID** signal is an output and **ARREADY** is an input. For an AHB master interface the **HADDR** signal is an output and **HRDATA** is an input. For an APB master interface, the **PADDR** signal is an output.

AMBA Slave Interface

An AMBA interface that is described as a slave interface is one where the subsystem is the slave and must be connected to a master interface. For an AXI slave interface the **ARVALID** signal is an input and **ARREADY** is an output. For an AHB slave interface the **HADDR** signal is an input and **HRDATA** is an output. For an APB slave 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.1 Input and output clocks

SSE-300 input clocks are defined in the following table.

Table 5-1: SSE-300 input clocks

ID	Power domain	Description	Connection information
SLOWCLK	PD_AON	Slow Clock. An always active slow clock that is asynchronous to the other clocks in the subsystem. **SLOWCLK** is one of the two clocks expected to be active in the lowest power state of the subsystem: HIBERNATE0. **SLOWCLK** is used primarily by timers residing in the PD_AON power domain. External gating of **SLOWCLK** is not supported by clock Q-Channels.	Connect **SLOWCLK** to a slow clock source that is always available.
AONCLK	PD_AON	Always ON Clock. This clock is used for the low frequency logic in the PD_AON power domain that is not running on the **SLOWCLK** 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 yet be independent from the rest of the system. **AONCLK** is asynchronous to the other clocks in the system. **AONCLK** can be externally gated if the corresponding clock Q-channel is in Q_STOPPED state.	Connect **AONCLK** to a slow clock source such as an output of a clock divider.
CNTCLK	PD_AON	System Counter Timestamp Clock associated with the CNTVALUEB System Counter Timestamp input. **CNTCLK** is asynchronous to the other clocks in the subsystem. External gating of **CNTCLK** is not supported by clock Q-Channels.	Connect **CNTCLK** to the clock source that drives the timestamp logic in the expansion logic.
SYSCLK	PD_AON	Main System Clock. This clock is the main clock to drive the main system that resides in PD_SYS. **SYSCLK** is also used for power management logic in PD_AON that is not running on **AONCLK**. **SYSCLK** is completely asynchronous to the other clocks in the subsystem except for **CPU0CLK**. The clock source is gated internally by PPUs before it is used by any logic.	Connect **SYSCLK** to a fast clock source, such as an output of a clock divider. In SSE-300, **SYSCLK** and **CPU0CLK** have the same frequency and are synchronous, so they can be driven by the same source.
CPU0CLK	PD_AON	CPU0 clock. This is the main clock used to drive the logic residing in the PD_CPU0 and PD_DEBUG domains. **CPU0CLK** clock is asynchronous to the other clocks in the subsystem except for **SYSCLK**. **CPU0CLK** might be externally gated if the corresponding clock Q-channel is in Q_STOPPED state. The clock source is gated internally by PPUs before it is used by any logic.	Connect **CPU0CLK** to a fast clock source, such as an output of a clock divider. In SSE-300, **SYSCLK** and **CPU0CLK** have the same frequency and are synchronous, so they can be driven by the same source.

In typical use, it is recommended 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 **SYSCLK** (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 **SYSCLK**. 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 **SYSCLK**.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

At minimum, two clock sources are needed. For example, one at 32kHz for **SLOWCLK** and another at 200MHz for the other clocks. Since **SYSCLK** and **CPU0CLK** must have the same frequency, if very large SRAMs with higher access time are needed in the subsystem, the frequency of both **SYSCLK** and **CPU0CLK** has to be reduced.

SSE-300 output clocks are defined in the following table. The clock outputs are synchronous to each other.

Table 5-2: SSE-300 output clocks

ID	Power domain	Description	Connection information
COMGMTSYSCLK	PD_AON	The gated version of **SYSCLK**. It is expected to be used to drive expansion logic that resides in the PD_AON power domain and is reset by **nCOLDRESETMGMT**.	Connect it to the PD_AON expansion logic that requires a clock that can be gated. The COMGMTSYSCLK Q-Channel device interface controls the gating of this clock.
MGMTSYSCLK	PD_AON	The gated version of **SYSCLK**. It is expected to be used to drive expansion logic that resides in the PD_AON power domain and is reset by **nWARMRESETMGMT**.	Connect it to the PD_AON expansion logic that requires a clock that can be gated. The MGMTSYSCLK Q-Channel device interface controls the gating of this clock.
COSYSSYSCLK	PD_SYS	The gated version of **SYSCLK**, expected to be used to drive expansion logic that resides in the PD_SYS power domain and is reset by **nCOLDRESETSYS**.	Connect it to the PD_SYS Expansion logic that requires a system clock that can be gated. The COSYSSYSCLK Q-Channel device interface controls the gating of this clock.
SYSSYSCLK	PD_SYS	The gated version of **SYSCLK**, expected to be used to drive expansion logic that resides in the PD_SYS power domain and is reset by **nWARMRESETSYS**.	Connect it to the PD_SYS Expansion logic that requires a system clock that can be gated. The SYSSYSCLK Q-Channel device interface controls the gating of this clock.
COCPUCPU0CLK	PD_CPU0	The gated version of **CPU0CLK**, expected to be used to drive expansion logic that resides in the PD_CPU0 power domain and is reset by **nCOLDRESETCPU0**.	Connect it to the PD_CPU0 expansion logic that requires a fast clock that can be gated. The COCPUCPU0CLK Q-Channel device interface controls the gating of this clock.
CPUCPU0CLK	PD_CPU0	The gated version of **CPU0CLK**, expected to be used to drive expansion logic that resides in the PD_CPU0 power domain and is reset by **nWARMRESETCPU0**.	Connect it to the PD_CPU0 expansion logic that requires a fast clock that can be gated. The CPUCPU0CLK Q-Channel device interface controls the gating of this clock.
DEBUGCPU0CLK	PD_DEBUG	The gated version of **CPU0CLK**, expected to be used to drive expansion logic that resides in the PD_DEBUG power domain.	Connect it to the PD_DEBUG expansion logic that requires a clock that can be gated. The DEBUGCPU0CLK Q-Channel device interface controls the gating of this clock.

SSE-300 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.2 Functional integration resets

Connection information for the top-level reset inputs and reset outputs.

The following table shows the reset inputs and outputs at SSE-300 top level. For more information, see Reset infrastructure.

Table 5-3: Top-level reset inputs and outputs

Signal	Direction	Power domain	Description	Connection information
nPORESET	Input	PD_AON	An active-LOW reset for power-on the system. **nPORESET** resets all registers in the design. This includes resetting the Reset syndrome register. There is no functional reset duration requirement for this reset input. However, SSE-300 recommends that the reset is at least one **SLOWCLK** cycle long. This reset input performs reset asynchronously. The deassertion of the reset is synchronized to **SLOWCLK** in the subsystem.	Connect it to the reset Power-on generator.
nMBISTRESET	Input	PD_AON	A reset input provided for production MBIST. This reset input has the same effect as **nPORESET** and the same description applies to it.	Connect it to the MBIST controller.
nSRST	Input	PD_AON	An active-LOW system-wide cold reset request, typically originating from an external debugger. When initially asserted, this signal results in a reset being applied to the subsystem and then subsequently removed. However, while **nSRST** is asserted, the CPUWAIT input of the CPU is held HIGH, preventing the core from booting until **nSRST** is deasserted. This is independent of the register setting in the CPUWAIT register. It is recommended that the assertion of the **nSRST** input is at least three **SLOWCLK** cycles long. It can be held LOW for as long as it is required to hold off the CPU from execution while debug related tasks are being performed. When deasserting **nSRST**, it must remain HIGH for at least five **SLOWCLK** cycles before asserting it again. This constraint is also applicable to the first assertion of **nSRST** that follows the deassertion of **nPORESET**. This asynchronous reset request input is synchronized to **SLOWCLK** before it is used for system-wide cold reset generation.	Connect to an external signal on the SoC, so that a debugger can reset SSE-300.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Power domain	Description	Connection information
HOSTRESETREQ	Input	PD_AON	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. When initially asserted, this signal must be held HIGH until the reset occurs on **nCOLDRESETAON**. Holding this input HIGH keeps **nCOLDRESETAON** asserted. This asynchronous reset request input is synchronized to **SLOWCLK** before it is used for system-wide cold reset generation.	Connect it to an external higher authority such as the hosting system.
RESETREQ	Input	PD_AON	An active-HIGH system-wide cold reset request. When initially asserted, this signal must be held HIGH until the reset occurs on **nCOLDRESETAON**, and must be cleared as a result of the reset being asserted. This asynchronous reset request input is synchronized to **SLOWCLK** before it is used for system-wide cold reset generation.	Connect to the expansion logic.
nCOLDRESETAON	Output	PD_AON	An active-LOW cold reset for the expansion system. This cold reset merges other reset sources in the system with **nPORESET** to generate this reset.	Connect it to the expansion logic that resides in the **PD_AON** power domain and requires a cold reset such as debug related logic. This reset output is asynchronous and must be resynchronized before use.
nWARMRESETAON	Output	PD_AON	An active-LOW Warm reset for the expansion system. This Warm reset is a superset of **nCOLDRESETAON** and is also asserted when the system is in n the system is in WARM_RST state.	Connect it to the expansion logic that resides in the PD_AON power domain and requires a Warm reset, such as non-debug related logic. This reset output is asynchronous and must be synchronized before use.
nCOLDRESETMGMT	Output	PD_AON	An active-LOW cold reset for the expansion system. This reset is a superset of **nCOLDRESETAON** and is also asserted when the PD_MGMT power domain is in a low-power state. Since PD_MGMT is merged into PD_AON, the signal can be regarded as the COMGMTSYSCLK synchronized version of **nCOLDRESETAON**.	Connect it to the expansion logic that was in the PD_MGMT power domain that is merged into the PD_AON power domain and requires a cold reset such as debug related logic. This reset output is asynchronously asserted and is synchronously deasserted to COMGMTSYSCLK.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Power domain	Description	Connection information
nWARMRESETMGMT	Output	PD_AON	An active-LOW Warm reset for the expansion system. This reset is a superset of **nCOLDRESETMGMT** and is also asserted when the system is in the WARM_RST state. Since PD_MGMT is merged into PD_AON, the signal can be regarded as the MGMTSYSCLK synchronized version of **nWARMRESETAON**.	Connect it to the expansion logic that was in the PD_MGMT power domain that is merged into the PD_AON power domain and requires a Warm reset such as non-debug related logic. This reset output is asynchronously asserted and is synchronously deasserted to MGMTSYSCLK.
nCOLDRESETSYS	Output	PD_SYS	An active-LOW cold reset for the expansion system. This reset is a superset of **nCOLDRESETAON** and is also asserted when the PD_SYS power domain is in a low-power state with the logic being turned off. **nCOLDRESETSYS** supports logic retention and is not asserted if the PD_SYS power domain is in a power mode with the logic being retained.	Connect it to the expansion logic that resides in the PD_SYS power domain and requires a cold reset, such as debug related logic. This reset output is asynchronously asserted and is synchronously deasserted to COSYSSYSCLK.
nWARMRESETSYS	Output	PD_SYS	An active-LOW Warm reset for the expansion system. This reset is a superset of **nCOLDRESETSYS** and is also asserted when the system is in the WARM_RST state.	Connect it to the expansion logic that resides in the PD_SYS power domain and requires a Warm reset. This reset output is asynchronously asserted and is synchronously deasserted to SYSSYSCLK.
nCOLDRESETCPU0	Output	PD_CPU0	An active-LOW cold reset for the expansion system. This reset is a superset of **nCOLDRESETAON** and is also asserted when the PD_CPU0 power domain is in a low-power state with the logic being turned off. **nCOLDRESETCPU0** supports logic retention and is not asserted if the PD_CPU0 power domain is in a power mode with the logic being retained.	Connect it to the expansion logic that resides in the PD_CPU0 power domain and requires a cold reset, such as debug related logic. This reset output is asynchronously asserted and is synchronously deasserted to COCPUCPU0CLK.
nWARMRESETCPU0	Output	PD_CPU0	An active-LOW Warm reset for the expansion system. This reset is a superset of **nCOLDRESETCPU0** and is also asserted when the system is in the WARM_RST state.	Connect it to the expansion logic that resides in the PD_CPU0 power domain and requires a Warm reset such as non-debug related logic. This reset output is asynchronously asserted and is synchronously deasserted to CPUCPU0CLK.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Power domain	Description	Connection information
nCOLDRESETDEBUGCPU0	Output	PD_DEBUG	An active-LOW cold reset for the expansion system. This reset is a superset of **nCOLDRESETAON** and is also asserted when the PD_DEBUG power domain is in a low-power state with the logic being turned off.	Connect it to the expansion logic that resides in the PD_DEBUG power domain and requires a reset, such as debug logic associated with CPU0. This reset output is asynchronously asserted and is synchronously deasserted to DEBUGCPU0CLK.

5.3 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 masters and slave interfaces in a power domain are in quiescent state before entering a lower power state. It also allows external master 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 will normally merge 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 masters 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 quiescent 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. Each P-Channel interface for power control must be able to support all power modes that the Bounded Region implements.
- All bus slaves in the expansion system must either individually or collectively have a Q-channel or P-Channel interface that must be able to delay the acceptance of a quiescence request or a Power mode request if the current bus operation is about to complete. The LPI interface must also be able to deny a quiescent request or a power mode request if the logic that it controls has other outstanding operations that prevents it from entering quiescent state or the requested mode.
- You must sequence the Q-Channels associated with these external bus interfaces in such a way to ensure that all bus masters are in quiescent state before any bus slaves are requested

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

to enter quiescent state. Similarly, with P-Channels, you must ensure that all bus masters and slaves 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-300 implements separate Q-Channel and P-Channel interfaces for each clock domain and power domain respectively. To separately handshake masters, slaves and even intermediate components in the expansion logic in sequence, additional external sequencing might be required in the expansion logic.

- It is not required to prevent data loss, protocol violation or CPU lock-up inside the reset domain if that has no affect outside the reset domain. 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 so that deadlock/livelock/Warm reset denial scenarios are prevented.

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, with each DEVPACTIVE bit used to request entry to a Power mode and Operating mode, and DEVPSTATE vector representing the Power mode and Operating mode that is being requested. For more information, refer to *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*.

5.3.1 Clock Control Q-Channel Device interfaces

SSE-300 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_MGMT power domain when PILEVEL = 2, or in the PD_AON domain when PILEVEL < 2.
- 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.
- CPUCPU{0-<NUMCPU>}CLK Q-Channel Device interface for CPUCPU{0-<NUMCPU>}CLK. This interface resides in the PD_CPU{0-<NUMCPU>} power domain when PILEVEL > 0, or in the PD_SYS Power domain.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- COCPUCPU{0-<NUMCPU>}CLK Q-Channel Device interface for COCPUCPU{0 to <NUMCPU>}CLK. This interface resides in the PD_CPU{0 to <NUMCPU>} Power domain.
- DEBUGCPU{0-<NUMCPU>}CLK Q-Channel Device interface for DEBUGCPU{0-<NUMCPU>}CLK. 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 MGMTSYSCLK, SYSSYSCLK and CPUCPU0CLK are synchronous to the clock that each of the Q-Channel interface controls.

The clock Q-Channel device interfaces COMGMTSYSCLK, COSYSSYSCLK and DEBUGCPU0CLK are asynchronous.

5.3.2 Power Control P-Channel Device interfaces

SSE-300 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:

- MGMTPWR P-Channel Device interface for PD_AON. Given that PD_AON is always on, this P-Channel is responsible for Warm reset related sequencing.
- SYSPWR P-Channel Device interface for BR_SYS.
- DEBUGPWR P-Channel Device interface for BR_DEBUG.
- CPU{0-<NUMCPU>}PWR P-Channel Device interface for BR_CPU{0-<NUMCPU>}.

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 are 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:

- If the interface is not used then:
    - *PWRPACTIVE and *PWRPDENY must be tied low
    - *PWRPPREQ must be looped back to the corresponding *PWRPACCEPT

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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

Not following the above rules will lead to system deadlock

5.3.3 Power Control Wakeup Q-Channel Device interfaces

SSE-300 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.
- PWRCPU{0-<NUMCPU>}WAKE Q-Channel Device interface for PD_CPU{0 to <NUMCPU>}.

These interfaces all reside in the PD_AON power domain. Because they are “wake up” requests and only the **QACTIVE** signal of each interface is implemented. When using these to wake a domain, you must drive and hold the appropriate **QACTIVE** signal to request turning on the power domain until the power domain is ON.

Note

Note that PD_CPU{0-<NUMCPU>} is ON in all the ON, MEM_OFF (EPU_OFF) and FUNC_RET power modes.

For example, to wake PD_SYS, you must set **PWRSYSWAKEQACTIVE** Q-Channel signal to request it to turn ON until the the SYSPWR P-Channel Device interface for BR_SYS indicates that the power domain is ON.

These Wakeup Device Q-Channel interfaces are asynchronous.

5.4 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-300 has the following Clock Control Q-Channel Control interfaces:

- AONCLK Q-Channel Control interface for **AONCLK**

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- SYSCLK Q-Channel Control interface for **SYSCLK**
- CPU{0-<NUMCPU>}CLK Q-Channel Control interface for CPU{0-<NUMCPU>}CLK

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.

5.5 Expansion Power Control Dependency interface

SSE-300 provides a 4-bit width Q-Channel interface that allows external power domains to use the Power Dependency Control Matrix to keep power domains within the subsystem from entering a lower power state.

These signals are:

- Power Dependency Control Matrix QREQn Inputs, **PDCMQREQn[3:0]**. When each bit is set to 1, it indicates that the external domain that drives it is active.
- Power Dependency Control Matrix QACCEPTn Outputs, **PDCMQACCEPTn[3:0]**. Each bit acknowledges an associated bit on **PDCMQREQn** 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.

These signals are essential for an external domain which might want to ensure that a different power domain remain powered while the external domain is trying to work with it. For example, the external domain might want to access the Main interconnect in the PD_SYS domain. It can raise an interrupt with the host CPU, request that a dependency is setup between the external system and PD_SYS via a signal in PDCMQREQn, by setting the relevant bit in PDCM_PD_SYS_SENSE register. Therefore, the external system can now keep PD_SYS powered using one of the **PDCMQREQn** signals. The four-phase handshake must be used to ensure that there is no race condition between ending a bus access that wakes up the domain and activating the keeping up of the domain.

These are asynchronous signals that reside in the PD_AON power domain. For more information about registers that uses these signals and description of related functionality, see System Control Register Block and Power Integration.

5.6 Power Domain ON Status Signals

SSE-300 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:

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- **PDMGMTON**
  - HIGH indicates that the PD_MGMT power domain's PPU considers that the domain is ON
  - LOW indicates that the PPU acts as though 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 PD_SYS power domain's PPU acts as though the domain is ON
  - LOW indicates that the power domain's PPU acts as though the domain is in a lower power state
- **PDCPU{0-<NUMCPU>}ON**
  - HIGH indicates that the PD_CPU{0-<NUMCPU>} power domain's PPU acts as though the domain is ON
  - LOW indicates that the power domain's PPU acts as though the domain is in a lower power state
- **PDDEBUGON**
  - HIGH indicates that the PD_DEBUG power domain's PPU considers that the domain is ON
  - LOW indicates that the power domain's PPU acts as though the domain is in a lower power state

---

Note

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

5.7 System Timestamp interface

When EXPLOGIC_PRESENT = 0, SSE-300 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 is the following:

CNTVALUEB[63:0]

- Width - 64 bit
- IN/OUT - IN
- Clock domain - **CNTCLK**
- Description - Timestamp input value. This value is binary coded.

When HASCSS = 1, Arm recommends that the expansion system uses the CTI triggers to implement timestamp halting. For more information about the CTI interface, see Cross Trigger Interface.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

When HASCSS = 0, a debug system does not exist to provide these control signals and hence, the **CPU{0-<NUMCPU>}HALTED** signals are used to halt the system timestamp generator.

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.

5.8 Main Interconnect Expansion interfaces

SSE-300 provides a pre-configured number of Master and Slave Expansion interfaces off the Main Interconnect. These interfaces allow the system integrator to add additional bus masters and bus slaves 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 master and slave Expansion interface on the Main Interconnect are as follows:

- Master Code Main Expansion interface (XMSTEXPCODE). This interface provides access to Code Memory and is mapped to the following address range:
    - `0x01000000` to `0x0DFFFFFF`
    - `0x11000000` to `0x1DFFFFFF`
- Master Main Expansion interfaces (XMSTEXPSRAM, XMSTEXPDEV). These interfaces provide access to other Slaves in the system and are mapped to the following address range:
    - `0x28000000` to `0x2FFFFFFF` (on XMSTEXPSRAM)
    - `0x38000000` to `0x3FFFFFFF` (on XMSTEXPSRAM)
    - `0x60000000` to `0x9FFFFFFF` (on XMSTEXPSRAM)
    - `0xA0000000` to `0xDFFFFFFF` (on XMSTEXPDEV)
- Slave Main Expansion interfaces (XSLVEXPMI0 and XSLVEXPMI1 or HSLVEXPMI1). When EXPLOGIC_PRESENT = 0, AXI slave interfaces are supported (XSLVEXPMI0, XSLVEXPMI1). When EXPLOGIC_PRESENT = 1, an XHB-500 AHB to AXI bridge is instantiated in the SSE-300 expansion and AHB slave expansion interface (HSLVEXPMI1) is supported instead of XSLVEXPMI1. These interfaces provide access to the system from the expansion master interfaces. 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(s). See System Interconnect Infrastructure for more information.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.9 Peripheral Interconnect Expansion interfaces

SSE-300 provides a pre-configured number of Master and Slave Expansion interfaces from the Peripheral Interconnect. These interfaces allow the system integrator to add additional bus masters and bus slaves 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 Master and Slave Expansion interface on the Peripheral Interconnect are as follows:

- Master Peripheral Expansion interfaces (HMSTEXPPILL, HMSTEXPPIHL). These interfaces provide access to other Slaves 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)
- Slave Peripheral Expansion interfaces (HSLVEXPPILL, HSLVEXPPIHL). These interfaces provide access to the Peripheral bus from expansion master interfaces. These interfaces can be used to access all peripheral memory-mapped regions in the system except for regions that are private to the CPUs:
  - `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)
  - `0xF010000` to `0xFFFFFFFF` (through HSLVEXPPILL)

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.10 Interrupt interfaces

This table lists the interrupt signals for use by the subsystem expansion. These connect to the interrupt controller of each processor within the system and to an External Wakeup Interrupt Controller (EWIC) associated with the processor.

Table 5-4: Interrupt interfaces

Signal name	Width	IN/ OUT	Description
CPU{0-<NUMCPU>}EXPIRQ[CPU{0 to <NUMCPU>}EXPNUMIRQ-1:0]	CPU{0 to <NUMCPU>}EXPNUMIRQ	IN	These are Interrupt inputs from the subsystem expansion to the CPU{x} interrupt controller and the EWIC within the subsystem, where x is between 0 and NUMCPU. Each core in the subsystem implements a configurable number of external interrupt lines and of these 32 are reserved for internal use and the rest are made available here. CPU{x}EXPNUMIRQ defines the number of interrupts made available as expansion interrupts for CPU{x} where x is between 0 to NUMCPU. Note that each bit CPU{x}EXPIRQ[n] is ultimately connected to IRQ[32+n] of the CPU{x}'s NVIC. **Note:** When EXPLOGIC_PRESENT=1, the system timestamp generator (system counter) is integrated in SSE-300 expansion and its security violation interrupt is mapped to CPU0EXPIRQ[0]. See *Arm® SSE-123 Example Subsystem Technical Reference Manual* for more information about the system counter.
CPU{0-<NUMCPU>}EXPNMI	1	IN	This provides a non-maskable interrupt input from the subsystem expansion to the interrupt controller of CPU{x} and the EWIC within the subsystem, where x is between 0 and NUMCPU. This input is merged with other non-maskable interrupt sources within the subsystem before it is seen by the NVIC of the processor core.

5.11 CPU Co-Processor interface

Each processor core of the subsystem can be configured to have a co-processor interface. If a CPU<*n*> co-processor interface exists, then HASCPU<*n*>CPIF = 1, for *n* in 0 to NUMCPU.

These interfaces reside in the PD_CPU<n> power domain of their respective CPU, **CPUCPU<n>CLK** clock domain and **nWARMRESETCPU<n>** reset domain.

Note

Because of the actual processor implementation, the reset output CPU0CPRESETOUTn is part of the co-processor interface that is dependent on **nWARMRESETCPU<n>**.

5.12 CPU Custom Datapath Extension Interface

If supported by the CPU Core, each CPU core of the subsystem can be configured to have a Custom Datapath Extension interface.

The method to determine if this interface is present is CPU dependent.

For the corresponding configurations, see Configurable render options.

If a CPU<*n*> Custom Datapath Extension interface exists, then the protocol of this interface is CPU dependent. These interfaces reside in the PD_CPU<*n*> power domain of their respective CPU, the **CPUCPU<n>CLK** clock domain and the **nWARMRESETCPU<n>** reset domain.

Note

Because of the actual CPU implementation, the reset can be a special output that is dependent at least on **nWARMRESETCPU<n>**.

For more details on the Custom Datapath Extensions and related interfaces, see *Arm® Cortex®-M55 Technical Reference Manual*.

5.13 TCM DMA Slave Interfaces

A 64 bit Slave DMA interface per core provides system access only to Tightly Coupled Memories (TCM) internal to each core. Therefore, there are up to NUMCPU of these TCM DMA Slave Interfaces. The protocol of this interface is AMBA AXI-5.

These expansion interfaces are typically used together with DMA controllers to transfer data to and from the processor's TCM Interface.

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 slave 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 two 32 bit wide DTCMs.

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 5-5: TCM DMA Interface Memory Map

Start address	End address	AxADDR[3:2]	TCM accessed
`0x00000000`	`0x00000000` + {ITCM size}	-	ITCM
`0x20000000`	`0x20000000` + {DTCM size}	2b00	D0TCM
`0x20000000`	`0x20000000` + {DTCM size}	2b01	D1TCM
`0x20000000`	`0x20000000` + {DTCM size}	2b10	D2TCM
`0x20000000`	`0x20000000` + {DTCM size}	2b11	D3TCM

Note

These addresses in the previous table are specific only for this interface. These TCMs will reside at address offsets in the main memory map and are private to each core. See CPU TCM memories. To make these TCMs visible to other cores and masters within the system, these interfaces must be mapped to expansion regions of the memory map. This is defined by the system integrator.

5.14 Debug and Trace Related Interfaces

The following sections describe the debug and trace related interfaces of SSE-300.

5.14.1 Debug Access Interface

When EXPLOGIC_PRESENT = 0 and DEBUGLEVEL > 0, SSE-300 provides interfaces for debug access from an external debug access port or an external debug infrastructure. Depending on the HASCSS configuration, the interfaces provide the following:

- When HASCSS = 0, where there can only be one CPU, the CPU Debug D-AHB access interface is provided as an expansion interface. This allows you to drive the interface using a suitable CoreSight MEM-AP and provides debug access to the processor. For more information on the processor's D-AHB interface, see *Arm® Cortex®-M55 Technical Reference Manual*.

  The interface is in the PD_CPU0 power domain and resides in the COCPUCPU0CLK clock domain and nCOLDRESETCPU0 reset domain.

See HASCSS and DEBUGLEVEL for supported values.

5.14.2 Serial Wire JTAG Interface

When <EXPLOGIC_PRESENT = 1> and <DEBUGLEVEL > 0>, the DAP-Lite2 is integrated in the subsystem expansion and SSE-300 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The interface is in PD_AON power domain and resides in the **SWCLKTCK** clock domain and nPORESET and nTRST reset domains. SSE-300 does not support DEBUGLEVEL = 0.

See Trace port interface for more information about sharing JTAG-TDO with TRACESWO.

5.14.3 Debug Timestamp Interface

When EXPLOGIC_PRESENT = 0 and DEBUGLEVEL = 2, SSE-300 provides 64-bit timestamp input. This timestamp is expected to be driven by a timestamp generator in the subsystem expansion.

Depending on the HASCSS configuration, the interfaces are as follows:

- When HASCSS = 0 where there can only be one processor core, the processor's debug global timestamp input, TSVALUEB[63:0], is provided as an expansion interface, CPU0TSVALUEB[63:0]. This is expected to be driven by a global timestamp generator. For more information about these interfaces, see *Arm® Cortex®-M55 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 the processors 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 and DEBUGLEVEL = 2, 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 when EXPLOGIC_PRESENT = 1.

See HASCSS and DEBUGLEVEL for supported values.

5.14.4 Cross Trigger Channel Interface

When DEBUGLEVEL > 0, SSE-300 includes a set of cross-trigger channel inputs and cross-trigger channel outputs to allow you to expand the cross trigger infrastructure.

When HASCSS = 0 where there can only be one CPU, the cross-trigger channel interface of the CPU is provided as expansion interface, CPU{0-<NUMCPU>}CTICHIN[3:0] and CPU{0 to <NUMCPU>}CTICHOUT[3:0]. This interfaces resides in PD_CPU{0 to <NUMCPU>} power domain and is synchronous to COCPUCPU{0 to <NUMCPU>}CLK, resides in the nCOLDRESETCPU{0 to <NUMCPU>} reset domain.

See HASCSS and DEBUGLEVEL for supported values.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.14.5 Cross Trigger Interface

Since SSE-300 only supports HASCSS = 0, a cross trigger interface providing some trigger signals from an internal shared cross-trigger interface unit (CTI) does not exist.

5.14.6 Debug APB Expansion Interface

Since SSE-300 only supports HASCSS = 0, a debug APB expansion interface that could be used to add more debug functionality to a CoreSight SoC-600 based debug system does not exist.

5.14.7 CPU<n> External Peripheral Interface EPPB

Each CPU<*n*> in the system, where *n* is 0 to NUMCPU, provides an interface that allows you to add peripherals to the external PPB region that are private to each processor. This is a 32-bit AMBA4 APB interface for integration with additional CoreSight debug and trace components if required.

Data accesses are only allowed on each of these interfaces privately from each processor at address `0xE00040000` 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.

Each interface that is associated to CPU<*n*> resides in the PD_CPU<*n*> power domain, the COCPUCPU<*n*>CLK clock domain and the nCOLDRESETCPU<*n*> reset domain.

5.14.8 ATB Trace Interfaces

When EXPLOGIC_PRESENT = 0 and DEBUGLEVEL = 2 SSE-300 provides interfaces to output trace data to an expansion trace port interface unit (TPIU).

The number and types of interfaces varies depending on HASCSS configuration as follows:

- When HASCSS = 0 where there can only be one CPU, 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.

- SSE-300 does not support HASCSS = 1
- SSE-300 does not support DEBUGLEVEL < 2

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.14.9 Trace port interface

When EXPLOGIC_PRESENT = 1 and DEBUGLEVEL = 2, the Cortex-M55 TPIU is integrated in the subsystem expansion and SSE-300 provides trace port interface instead of the ATB trace interfaces to enable a trace port analyzer to connect.

The Cortex-M55 TPIU can operate in clocked or asynchronous port mode depending on the SW configuration.

Table 5-6: Trace port interface

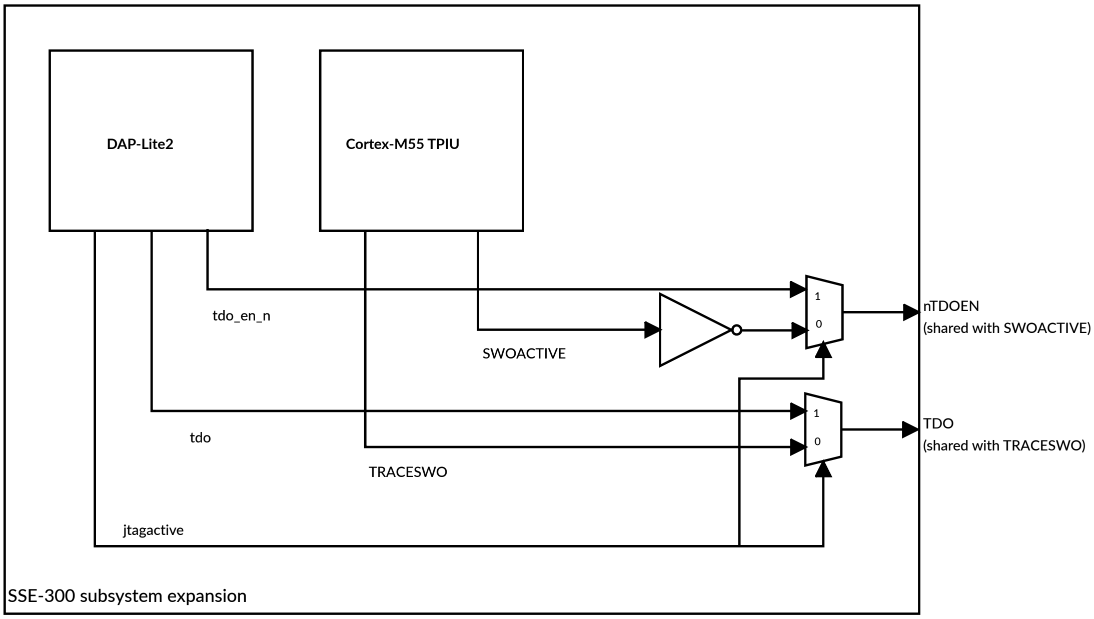
Signal name	Width	IN/ OUT	Description
TRACECLK	1	OUT	TRACECLK is used as a sampling clock for TRACEDATA[3:0] outside the system. Data changes in relation to both rising and falling edges. **Note:** This clock must not be used anywhere in the system.
TRACEPORTSIZE[1:0]	2	OUT	Indicates the enabled bits in TRACEDATA[3:0]: • 2'b00 - TRACEDATA[0] enabled. • 2'b01 - TRACEDATA[1:0] enabled. • 2'b11 - TRACEDATA[3:0] enabled.
TRACEDATA[3:0]	4	OUT	Output data for clocked port mode operation.
TRACESWO	1	OUT	Output data for asynchronous port mode operation.
SWOACTIVE	1	OUT	Indicates that TRACESWO is active.

Note

For minimal pin count, the JTAG debug TDO output and the TPIU serial wire output, TRACESWO are overlaid on the same package pin. Because of this approach, the instrumentation trace is not accessible while the debug port is being used in a JTAG configuration. Serial wire debug and SWO can be used together at the same time. In order to implement TRACESWO shared with JTAG-TDO, the jtagactive output from the DAP-Lite2 is used to control the multiplexor. The following image shows the TRACESWO shared with JTAG-TDO implementation.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 5-1: TRACESWO shared with JTAG-TDO



5.14.10 Debug Authentication Interface

The following table lists the debug authentication signals of the subsystem. The input signals define the Debug Authentication signal values when they are not overridden by the internal secure debug configuration registers. The final debug authentication signals are then made available as outputs to the rest of the system.

These signals reside in the PD_AON power domain.

Table 5-7: Debug authentication interface

Name	Width	IN/ OUT	Description
DBGENIN	1 bit	IN	Debug Enable Input
NIDENIN	1 bit	IN	Non-Invasive Debug Enable Input
SPIDENIN	1 bit	IN	Secure Privilege Invasive Debug Enable Input
SPNIDENIN	1 bit	IN	Secure Privilege Non-Invasive Debug Enable Input
DAPACCENIN	1 bit	IN	External Debug Access Enable Input
DAPDSSACCENIN	1 bit	IN	Debug Access Port to Debug System Access Enable Input
DBGEN	1 bit	OUT	Merged Debug Enable Output
NIDEN	1 bit	OUT	Merged Non-Invasive Debug Enable Output
SPIDEN	1 bit	OUT	Merged Secure Privilege Invasive Debug Enable Output
SPNIDEN	1 bit	OUT	Merged Secure Privilege Non-Invasive Debug Enable Output

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Name	Width	IN/ OUT	Description
DAPACCEN	1 bit	OUT	Merged External Debug Access Enable Output **Note:** **DAPACCEN** 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 bit	OUT	Merged Debug Access Port to Debug System Access Enable Output

5.15 CryptoCell-Related expansion interfaces

SSE-300 does not support HASCRYPTO = 1. Therefore, no CryptoCell related interfaces are supported.

5.16 Security Control Expansion Signals

SSE-300 provides additional status and control signals to handle additional *Master Security Controllers* (MSC), *Memory Protection Controllers* (MPC), *Peripheral Protection Controllers* (PPC) and Bridges with write buffers in the expansion system. These signals allow all the components to be controlled using the same set of security control registers already implemented within the subsystem.

All signals in this section are synchronous to SYSSYSCLK. The SYSSYSCLK Q-Channel Device Interface is needed to control the availability of SYSSYSCLK. These signals reside in the PD_SYS power domain and in the nWARMRESETSYS reset domain.

Note

While SSE-300 defines a full set of signals in this document, many of these interfaces and some of their individual bits can be unimplemented (tied) or disabled. These are defined as configuration options in Configurable render options.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.16.1 Memory Protection Controller Expansion

SSE-300 supports up to 16 MPCs to be added to the expansion system. The following signals allow the interrupts of the MPCs to be merged to the single MPC Combined interrupt internally.

Table 5-8: MPC Expansion Interrupt Status input.

Name	Width	IN/ OUT	Description
SMPCEXPSTATUS	16 bit	IN	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 **RAZ/WI**.

5.16.2 Peripheral Interconnect Peripheral Protection Controller Expansion

SSE-300 supports up to 4 additional PPCs to be added to the Peripheral Interconnect in the expansion system. The following signals are provided to control the PPCs.

Table 5-9: Peripheral Interconnect PPC Expansion Interface

Name	Width	IN/ OUT	Description
SPERIPHPPCEXPSTATUS	4-bit	IN	Peripheral Interconnect PPC Interrupt Status Input. Each bit 'n' is to be connected to a single PPC <n> where n is 0 to 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 **RAZ/WI**.
SPERIPHPPCEXPCLEAR	4-bit	OUT	Peripheral Interconnect PPC Interrupt Clear Output. Each bit 'n' is to be connected to a single PPC<n> where n 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 **RAZ/WI**.
PERIPHNSPPCEXP0	16-bit	OUT	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 'm' of an interface is HIGH, it defines a specific <m> interface that the target PPC controls as Non-secure access only. Each 16-bit signal **PERIPHNSPPCEXP<n>** is driven by the PERIPHNSPPCEXP<n> register, where n is 0 to 3. Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being **RAZ/WI**.
PERIPHNSPPCEXP1	16-bit	OUT	See description of **PERIPHNSPPCEXP0**.
PERIPHNSPPCEXP2	16-bit	OUT	See description of **PERIPHNSPPCEXP0**.
PERIPHNSPPCEXP3	16-bit	OUT	See description of **PERIPHNSPPCEXP0**.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Name	Width	IN/ OUT	Description
PERIPHPPPCEXP0	16-bit	OUT	Peripheral Interconnect PPC Privilege Gating Control. This is a set of four 16-bit interfaces. When each bit 'm' of an interface is HIGH it defines the <m> interface that the target PPC controls as both privilege and un-privilege access. Having a control bit LOW allows privilege access only. Each bit PERIPHPPPCEXP<n>[m] is selected from either PERIPHSPPPCEXP<n>[m] if PERIPHNSPPCEXP<n>[m] is '0' or PERIPHNSPPPCEXP<n>[m] otherwise, where n is 0 to 3 Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being **RAZ/WI**.
PERIPHPPPCEXP1	16-bit	OUT	See description of **PERIPHPPPCEXP0**
PERIPHPPPCEXP2	16-bit	OUT	See description of **PERIPHPPPCEXP0**
PERIPHPPPCEXP3	16-bit	OUT	See description of **PERIPHPPPCEXP0**

5.16.3 Main Interconnect Peripheral Protection Controller Expansion

SSE-300 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 5-10: Main Interconnect PPC Expansion Interface

Name	Width	IN/ OUT	Description
SMAINPPCEXPSTATUS	4-bit	IN	Main Interconnect PPC Interrupt Status Input. Each bit 'n' is to be connected to a single PPC <n> where n is 0 to 3. These are associated to the SECPPCINTST AT.SMAINPPCEXP_STATUS register field. Individual bit of this interface can be disabled if tied to 0, resulting in the the associated register bit field being **RAZ/WI**.
SMAINPPCEXPCLEAR	4-bit	OUT	Main Interconnect PPC Interrupt Clear Output. Each bit 'n' is to be connected to a single PPC<n> where n is 0 to 3. These are associated to the SECPPCINTCLR SMAINPPCEXP_CLR register field. 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 **RAZ/WI**.
MAINNSPPCEXP0	16-bit	OUT	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 'm' of an interface is HIGH, it defines the <m> interface that the target PPC controls as Non-secure access only. Each 16-bit signal **MAINNSPPCEXP<n>** is driven by the MAINNSPPCEXP<n> register, where n is 0 to 3. Individual bit of this interface can be (tied) disabled, resulting in the the associated register bit field being **RAZ/WI**.
MAINNSPPCEXP1	16-bit	OUT	See description of **MAINNSPPCEXP0**
MAINNSPPCEXP2	16-bit	OUT	See description of **MAINNSPPCEXP0**
MAINNSPPCEXP3	16-bit	OUT	See description of **MAINNSPPCEXP0**

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Name	Width	IN/ OUT	Description
MAINPPPCEXP0	16-bit	OUT	Main Interconnect PPC Privilege Gating Control. This is a set of four 16bit interfaces. When each bit 'm' of an interface is HIGH it defines the <m> interface that the target PPC controls as both Privileged and Unprivileged access. Having a control bit LOW allows privilige access only. Each bit **MAINPPPCEXP<n>[m]** is selected from either MAINSPPPCEXP<n>[m] if MAINNSPPCEXP<n>[m] is '0' or MAINNSPPPCEXP<n>[m] otherwise, where n is 0 to 3. Individual bits of this interface can be unimplemented or disabled resulting in the associated register bit fields that contributes to this control signal being **RAZ/WI**.
MAINNPPPCEXP1	16-bit	OUT	See description of **MAINNPPPCEXP0**
MAINNPPPCEXP2	16-bit	OUT	See description of **MAINNPPPCEXP0**
MAINNPPPCEXP3	16-bit	OUT	See description of **MAINNPPPCEXP0**

5.16.4 Master Security Controller Expansion

SSE-300 can support up to 16 additional *Master Security Controllers* (MSC) to be added to the expansion system. The following signals are provided to control each MSC.

Table 5-11: MSC Expansion Interface

Name	Width	IN/ OUT	Description
SMSCEXPSTATUS	16 bit	IN	MSC Interrupt Status Input. Each bit 'n' is to be connected to a single MSC<n> where n is 0 to 15. These are associated with the SECMSCINTSTAT.SMSCEXP_STATUS register field. Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being **RAZ/WI**.
SMSCEXPCLEAR	16 bit	OUT	MSC Interrupt Clear Output. Each bit 'n' is to be connected to a single MSC<n> where n is 0 to 15. These are associated with the SECMSCINTCLR.SMSCEXP_CLR register field. Individual bits of this interface can be (tied) disabled, defined by the MSCEXPDIS parameter, resulting in the associated register bit field being **RAZ/WI**.
NSMSCEXP	16 bit	OUT	MSC Non-secure Configuration. Each bit 'n' is to be connected to a single MSC <n> where n is 0 to 15. Set HIGH to configure a master as Non-secure. These are associated with the NSMSCEXP register. 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5.16.5 Bridge Buffer Error Expansion

SSE-300 supports up to 16 additional bridges with buffer error signalling to be added to the expansion system.

Table 5-12: Bridge Error Interrupt Expansion Interface

Name	Width	IN/ OUT	Description
BRGEXPSTATUS	16 bit	IN	Bridge Error Interrupt Status Input. Each bit 'n' is to be connected to a single bridge <n> where n is 0 to 15. These are associated with the BRGINTCLR.BRGEXP_STATUS register field. Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being **RAZ/WI**.
BRGEXPCLEAR	16 bit	OUT	Bridge Error Interrupt Clear Output. Each bit 'n' is to be connected to a single bridge <n> where n is 0 to 15. These are associated with the BRGINSTAT.BRGEXP_CLR register field. Individual bits of this interface can be (tied) disabled, resulting in the associated register bit field being **RAZ/WI**.

When EXPLOGIC_PRESENT = 1, the XHB-500 AHB to AXI bridge is integrated in SSE-300 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.

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

5.16.6 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 5-13: Other Security Expansion Signals

Name	Width	IN/ OUT	Description
SECRESPCFG	1 bit	OUT	This signal configures how to respond to an access when a security violation occurs. • 0 - **RAZ/WI** • 1 - Bus error The SECRESPCFG register controls this signal.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Name	Width	IN/ OUT	Description
ACCWAITn	1 bit	OUT	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. • 1 - No gating • 0 - Access gated The BUSWAIT register controls this signal.
ACCWAITNSTATUS	1 bit	IN	This status signal is used to indicate 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 • 1 - No gating • 0 - Access gated Use the BUSWAIT register to read this signal.

5.17 Clock configuration interface

Description of the clock configuration interface.

SSE-300 provides control and status signals for the input clocks **AONCLK**, **CPU0CLK**, and **SYSCLK**. These signals support the configuration of generators or dividers that might exist in the expansion logic. 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.

Since **SYSCLK** and **CPU0CLK** must have the same frequency, the CPU0CLKCFG output must not be used in the expansion logic, and the CPU0CLKCFGSTATUS input must be tied to LOW.

The register fields CLK_CFG0.CPU0CLKCFG and CLK_CFG0.CPU0CLKCFGSTATUS should not be used either.

Table 5-14: Clock configuration interface signals

Signal	Direction	Description
CPU0CLKCFG[3:0]	Output	These control signals provide a set of four-bit signals that allows the system to configure an external clock generation logic that drives CPU0CLK. The CLK_CFG0.CPU0CLKCFG register field drives this signal.
CPU0CLKCFGSTATUS[3:0]	Input	Four-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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Description
SYSCLKCFG[3:0]	Output	This control signal provides a four-bit signal that allows the system to configure an external clock generation logic that drives SYSCLK clock. The CLK_CFG1.SYSCLKCFG register field drives this signal.
SYSCLKCFGSTATUS[3:0]	Input	A four-bit status signal, used to read the status of any external clock generation logic that drives SYSCLK clock. Use the CLK_CFG1.SYSCLKCFGSTATUS register field to read the values on this interface.
AONCLKCFG[3:0]	Output	This control signal provides a four-bit signal that allows the system to configure an external clock generation logic that drives AONCLK clock. The CLK_CFG1.AONCLKCFG register field drives this signal.
AONCLKCFGSTATUS[3:0]	Input	A four-bit status signal, used to read the status of any external clock generation logic that drives AONCLK clock. Use the CLK_CFG1.AONCLKCFGSTATUS register field to read the values on this interface.

5.18 Miscellaneous Signals

The following table lists miscellaneous signals available for SSE-300.

Table 5-15: Other Miscellaneous Top-Level Signals

Signal name	Width	IN/ OUT	Sync to	Power domain	Description
LOCKNSVTOR<n>	1	IN	CPU<n>CLK	PD_CPU<n>	Disables writes to the VTOR_NS register. For more information on this register, see *Arm® v8-M Architecture Reference Manual*. Asserting this signal prevents changes to the Non-secure vector table base address. This signal can be changed dynamically. Note that when SECEXT=0, only VTOR_NS exists and this signal is used to disable writes to the register. When HIGH, disables writes to the CPU <n> Non-secure vector table base address register, VTOR_NS, where n is 0 to NUMCPU. If not used, tie to LOW. Tying this signal HIGH causes loss of vector table control.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal name	Width	IN/ OUT	Sync to	Power domain	Description
LOCKNSMPU<n>	1	IN	CPU<n>CLK	PD_CPU<n>	This signal disables writes to registers that are associated with the Non-secure MPU region from software or from a debug agent connected to the CPU <n> (where n is 0 to NUMCPU). • MPU_CTRL_NS • MPU_RNR_NS • MPU_RBAR_NS • MPU_RLAR_NS • MPU_RBAR_A_NSn • MPU_RLAR_A_NSn For more information on these registers, see *Arm® v8-M Architecture Reference Manual*. Asserting this signal prevents changes to the memory regions which have been programmed in the Non-secure MPU. All writes to the registers are ignored. This signal has no effect if the processor has not been configured with support for Non-secure MPU regions. This signal can be changed dynamically. **Note:** If you want the registers unlocked, tie all bits LOW, otherwise drive with external logic. Tying this signal HIGH causes loss of Non-secure memory protection control.
CPU<n>WAITCLR	1	IN	SYSCLK	PD_AON	When HIGH, clears the register fields CPUWAIT.CPU<n>WAIT, where n is 0 to NUMCPU. This allows an external entity to release a processor that is already waited by CPUWAIT to start execution. Once set to 1 this signal must be held at 1 until **CPU<n>WAITCLRRESP** is 1. **Note:** While this is clocked using SYSCLK, this input can optionally be implemented as an asynchronous input.
CPU<n>WAITCLRRESP	1	OUT	SYSCLK	PD_AON	This signal provides a response to the **CPU<n>WAITCLR** request, where n is 0 to NUMCPU. When **CPU<n>WAITCLR** is 1 and CPUWAIT.CPU<n>WAIT is 0, this signal is set to 1 until **CPU<n>WAITCLR** request goes 0.
NSWDRSTREQSTATUS	1	OUT	SYSCLK	PD_AON	Non-secure watchdog reset request status. This signal is 1 when the Non-secure Watchdog is raising a reset request and RESET_MASK.NSWDRSTREQ_EN is 1. Once set to high it will not return to low unless a reset clearing this status occurs. This port is optional but must exist when COLDRESET_MODE = 1.
SWDRSTREQSTATUS	1	OUT	SYSCLK	PD_AON	Secure watchdog reset request status. This signal is 1 when the Secure Watchdog is raising a reset request. Once set to high it will not return to low unless a reset clearing this status occurs. This port is optional but must exist when COLDRESET_MODE = 1.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal name	Width	IN/ OUT	Sync to	Power domain	Description
SSWDRSTREQSTATUS	1	OUT	SYSCLK	PD_AON	SLOWCLK Secure watchdog reset request status. This signal is 1 when the SLOWCLK Watchdog is raising a reset request. Once set to high it will not return to low unless a reset clearing this status occurs. This port is optional but must exist when COLDRESET_MODE = 1.
RESETREQSTATUS	1	OUT	SYSCLK	PD_AON	Hardware Reset Request status. This signal is set to 1 if RESETREQ input is 1. Once set to high, it must not be cleared unless the system is reset.
SWRSTREQSTATUS	1	OUT	SYSCLK	PD_AON	Software Reset Request Status. This signal is 1 when SWRESET.SWRESETREQ is set to 1. Once set to high, it must not be cleared unless the system is reset while restores the register field to 0. This port is optional but must exist when COLDRESET_MODE = 1.
CPU<n>LOCKUP	1	OUT	AONCLK	PD_AON	Processor Lockup Status. There is one bit per Processor. Each bit indicates if the associated CPU<n> has lockup where n is 0 to NUMCPU. This signal is an output directly from CPU<n>.
CPU<n>HALTED	1	OUT	CPU<n>CLK	PD_CPU<n>	Processor Halted Status. There is one bit per Processor. Each bit indicates if the associated CPU<n> has halted where n is 0 to NUMCPU. This signal is an output directly from CPU<n>.
CPU<n>EDBGRQ	1	IN	CPU<n>CLK	PD_CPU<n>	External request for CPU<n> to enter halt mode where n is 0 to to NUMCPU. This signal is an input directrly to CPU<n>.
CPU<n>DBGRESTART	1	IN	CPU<n>CLK	PD_CPU<n>	Request for for CPU<n> to perform synchronized exit from halt mode where n is 0 to NUMCPU. This signal is an input directly to CPU<n>.
CPU<n>DBGRESTARTED	1	OUT	CPU<n>CLK	PD_CPU<n>	Acknowledges **CPU<n>DBGRESTART** where n is 0 to NUMCPU. This signal is an output directly from CPU<n>.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6 Programmer Model

The following sections describe the programmer model of SSE-300.

6.1 System Memory Map Overview

High Level System Address Map shows the high-level view of the memory map defined by SSE-300. 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 each area's Security along with its ID and each region's *Non-secure Callable* (NSC) settings.

Except when specifically stated, the following will occur:

- All access to unmapped regions of the memory will result in bus-error response
- When accessing unmapped address space within a mapped region taken by a peripheral, the access will be result in *Read-As-Zero and Write-Ignored* (**RAZ/WI**) except when specifically stated otherwise
- Any accesses that result in security violations will either **RAZ/WI** 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:

- CPUs
- Expansion masters via the Slave 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 will not be monitored for exclusive access and might still update their target memory locations regardless of their associated exclusive responses.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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

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

If HASCRYPTO = 0, the CrytpoCell NVM Code region is reserved and will respond with bus error.

`0x00000000 - 0x00FFFFFF`

- Size: 16MB
- Region Name: ITCM
- Description: CPU Instruction TCM. See CPU TCM memories.
- Alias with Row ID: 5
- IDAU Region Values
  - Security: NS
  - IDAUID: 0
  - NSC: 0

`0x01000000 - 0x0DFFFFFF`

- Size: 208MB
- Region Name: Code Expansion
- Description: Master Code Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: 6
- IDAU Region Values
  - Security: NS
  - IDAUID: 0
  - NSC: 0

`0x0E000000 - 0x0E001FFF`

- Size: 8KB
- Region Name: CrytpoCell NVM Code
- Description: CryptoCell Code Access to NVM[3].

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Alias with Row ID: 7
- IDAU Region Values
    - Security: NS
    - IDAUID: 0
    - NSC: 0

0x0E002000 - 0x0FFFFFFF

- Size: -
- Region Name: Reserved
- Description: Reserved
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 0
    - NSC: 0

0x10000000 - 0x10FFFFFF

- Size: 16MB
- Region Name: ITCM
- Description: CPU Instruction TCM. See CPU TCM memories.
- Alias with Row ID: 1
- IDAU Region Values
    - Security: S
    - IDAUID: 1
    - NSC: CODENSC

0x11000000 - 0x1DFFFFFF

- Size: 208MB
- Region Name: Code Expansion
- Description: Master Code Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: 2
- IDAU Region Values
    - Security: S
    - IDAUID: 1
    - NSC: CODENSC

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

`0x1E000000 - 0x1E001FFF`

- Size: 8KB
- Region Name: CrytpoCell NVM Code
- Description: CryptoCell Code Access to NVM[3].
- Alias with Row ID: 3
- IDAU Region Values
  - Security: S
  - IDAUID: 1
  - NSC: CODENSC

`0x1E002000 - 0x1FFFFFFF`

- Size: -
- Region Name: Reserved
- Description: Reserved
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 1
  - NSC: CODENSC

`0x20000000 - 0x20FFFFFF`

- Size: 16MB
- Region Name: DTCM
- Description: CPU Data TCM. See CPU TCM memories.
- Alias with Row ID: 13
- IDAU Region Values
  - Security: NS
  - IDAUID: 2
  - NSC: 0

`0x21000000 - 0x21FFFFFF`

- Size: 16MB
- Region Name: Volatile Memory
- Description: Internal Multi-bank Volatile Memory. See Volatile Memory Region.
- Alias with Row ID: 14

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
    - Security: NS
    - IDAUID: 2
    - NSC: 0

0x22000000 - 0x27FFFFFF

- Size: 96MB
- Region Name: Reserved
- Description: Reserved
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 2
    - NSC: 0

0x28000000 - 0x2FFFFFFF

- Size: 128MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 2
    - NSC: 0

0x30000000 - 0x30FFFFFF

- Size: 16MB
- Region Name: DTCM
- Description: CPU Data TCM. See CPU TCM memories.
- Alias with Row ID: 9
- IDAU Region Values
    - Security: S
    - IDAUID: 3
    - NSC: RAMNSC

0x31000000 - 0x31FFFFFF

- Size: 16MB

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Region Name: Volatile Memory
- Description: Internal Multi-bank Volatile Memory. See Volatile Memory Region.
- Alias with Row ID: 10
- IDAU Region Values
  - Security: S
  - IDAUID: 3
  - NSC: RAMNSC

`0x32000000 - 0x37FFFFFF`

- Size: 96MB
- Region Name: Reserved
- Description: Reserved
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 3
  - NSC: RAMNSC

`0x38000000 - 0x3FFFFFFF`

- Size: 128MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 3
  - NSC: RAMNSC

`0x40000000 - 0x4000FFFF`

- Size: 64KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: 27

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
  - Security: NS
  - IDAUID: 4
  - NSC: 0

`0x40010000 - 0x4001FFFF`

- Size: 64KB
- Region Name: Private CPU
- Description: CPU Private Peripheral Region. See Processor Private Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: NS
  - IDAUID: 4
  - NSC: 0

`0x40020000 - 0x4003FFFF`

- Size: 128KB
- Region Name: System Control
- Description: System Control Peripheral Region. See System Control Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: NS
  - IDAUID: 4
  - NSC: 0

`0x40040000 - 0x400FFFFF`

- Size: 768KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: NS
  - IDAUID: 4
  - NSC: 0

`0x40100000 - 0x47FFFFFF`

- Size: 127MB

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x48000000 - 0x4800FFFF

- Size: 64KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: 32
- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x48010000 - 0x4801FFFF

- Size: 64KB
- Region Name: Private CPU
- Description: CPU Private Peripheral Region. See Processor Private Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x48020000 - 0x4803FFFF

- Size: 128KB
- Region Name: System Control
- Description: System Control Peripheral Region. See System Control Peripheral Region.
- Alias with Row ID: -

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x48040000 - 0x480FFFFF

- Size: 768KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x48100000 - 0x4FFFFFFF

- Size: 127MB
- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces and Peripheral Expansion Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 4
    - NSC: 0

0x50000000 - 0x5000FFFF

- Size: 64KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: 17
- IDAU Region Values
    - Security: S
    - IDAUID: 5
    - NSC: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

`0x50010000 - 0x5001FFFF`

- Size: 64KB
- Region Name: Private CPU
- Description: CPU Private Peripheral Region. See Processor Private Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: 5
    - NSC: 0

`0x50020000 - 0x5003FFFF`

- Size: 128KB
- Region Name: System Control
- Description: System Control Peripheral Region. See System Control Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: 5
    - NSC: 0

`0x50040000 - 0x500FFFFF`

- Size: 768KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: 5
    - NSC: 0

`0x50100000 - 0x57FFFFFF`

- Size: 127MB
- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces.
- Alias with Row ID: -

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x58000000 - 0x5800FFFF`

- Size: 64KB
- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: 22
- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x58010000 - 0x5801FFFF`

- Size: 64KB
- Region Name: Private CPU
- Description: Processor Private Peripheral Region. See Processor Private Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x58020000 - 0x5803FFFF`

- Size: 128KB
- Region Name: System Control
- Description: System Control Peripheral Region. See System Control Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x58040000 - 0x580FFFFF`

- Size: 768KB

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Region Name: Peripherals
- Description: Peripheral Region. See Peripheral Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x58100000 - 0x5FFFFFFF`

- Size: 127MB
- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces and Peripheral Expansion Region.
- Alias with Row ID: -
- IDAU Region Values
  - Security: S
  - IDAUID: 5
  - NSC: 0

`0x60000000 - 0x6FFFFFFF`

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
  - Security: NS
  - IDAUID: 6
  - NSC: 0

`0x70000000 - 0x7FFFFFFF`

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
    - Security: S
    - IDAUID: 7
    - NSC: 0

0x80000000 - 0x8FFFFFFF

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: 8
    - NSC: 0

0x90000000 - 0x9FFFFFFF

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: 9
    - NSC: 0

0xA0000000 - 0xAFFFFFFF

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: A
    - NSC: 0

0xB0000000 - 0xBFFFFFFF

- Size: 256MB

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: B
    - NSC: 0

`0xC0000000 - 0xCFFFFFFF`

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: C
    - NSC: 0

`0xD0000000 - 0xDFFFFFFF`

- Size: 256MB
- Region Name: Main Expansion
- Description: Master Main Expansion Interface. See Main Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: D
    - NSC: 0

`0xE0000000 - 0xE00FFFFF`

- Size: 1MB
- Region Name: PPB
- Description: CPU Private Peripheral Bus Region. Local to Each CPU. See CPU Private Peripheral Bus Region.
- Alias with Row ID: -
- IDAU Region Values: Exempt

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

`0xE0100000 - 0xE01FFFFF`

- Size: 1MB
- Region Name: Debug System
- Description: Debug System Access Region. See Debug System Access Region.
- Alias with Row ID: 49
- IDAU Region Values
    - Security: NS
    - IDAUID: E
    - NSC: 0

`0xE0200000 - 0xEFFFFFFF`

- Size: 254MB
- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: NS
    - IDAUID: E
    - NSC: 0

`0xF0000000 - 0xF00FFFFF`

- Size: 1MB
- Region Name: Reserved
- Description: Reserved
- Alias with Row ID: -
- IDAU Region Values: Exempt

`0xF0100000 - 0xF01FFFFF`

- Size: 1MB
- Region Name: Debug System
- Description: Debug System Access Region. See Debug System Access Region.
- Alias with Row ID: 46

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- IDAU Region Values
    - Security: S
    - IDAUID: F
    - NSC: 0

`0xF0200000 - 0xFFFFFFFF`

- Size: 254MB
- Region Name: Peripheral Expansion
- Description: Master Peripheral Expansion Interface. See Peripheral Interconnect Expansion Interfaces.
- Alias with Row ID: -
- IDAU Region Values
    - Security: S
    - IDAUID: F
    - NSC: 0

6.2 CPU TCM memories

The CPUs in SSE-300 are configured to implement *Tightly Coupled Memories* (TCM) for Instruction and Data. These memories reside in the following location from the perspective of each CPU core:

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

Each CPU has access only to its own local TCMs and masters on the Main Interconnect and Peripheral Interconnect do not have access to these TCMs. However, each CPU provides TCM DMA Slave Interfaces to allow expansion masters to access the TCMs. Refer to TCM DMA slave interfaces.

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-300 supports up to four internal *Volatile Memory* (VM) Banks but it is limited to two Volatile Memory 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 256Kbytes in size each.

Table 6-1: Example Volatile Memory region address map

Row ID	Address - from	Address - to	Size	Region name	Description	Alias with row ID	Security[1 2]
1	0x21000000	0x2103FFFF	256KB	VM0	Maps to Internal Volatile Memory Bank 0	5	NS-MPC
2	0x21040000	0x2107FFFF	256KB	VM1	Maps to Internal Volatile Memory Bank 1	6	NS-MPC
3	0x21080000	0x210FFFFF	512KB	Reserved	Reserved	-	-
4	0x21100000	0x21FFFFFF	15MB	Reserved	Reserved	-	-
5	0x31000000	0x3103FFFF	256KB	VM0	Maps to Internal Volatile Memory Bank 0	1	S-MPC
6	0x31040000	0x3107FFFF	256KB	VM1	Maps to Internal Volatile Memory Bank 1	2	S-MPC
7	0x31080000	0x310FFFFF	512KB	Reserved	Reserved	-	-
8	0x31100000	0x31FFFFFF	15MB	Reserved	Reserved	-	-

1. NS-MPC: Non-secure access only, gated by a MPC. S-MPC: Secure access only, gated by a MPC.
2. Based on MPCs, where block size is determined by VMMPCBLKSIZE.

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 0x4000FFFF

Non-secure region for low-latency peripherals that are expected to be aliased in its associated Secure region, 0x50000000 to 0x5000FFFF.

0x40040000 to 0x400FFFFF

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

0x48000000 to 0x4800FFFF

Non-secure region for high-latency peripherals that are expected to be aliased in its associated Secure region, 0x58000000 to 0x5800FFFF.

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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

`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.

Caution

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

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

Table 6-2: Peripheral Region Address Map

Row ID	Address		Size	Region name	Description	Alias with row ID	Security[1]
	From	To					
1	`0x40000000`	`0x40000FFF`	4KB	MHU0[3]	Message Handling Unit 0. See Message Handling Unit registers.	20	NS-PPC
2	`0x40001000`	`0x40001FFF`	4KB	MHU1[3]	Message Handling Unit 1. See Message Handling Unit registers.	21	
3	`0x40002000`	`0x4000FFFF`	-	Reserved	Reserved. (**RAZ/WI**)	-	-
4	`0x40040000`	`0x4007FFFF`	-	Reserved	Reserved	-	-
5	`0x40080000`	`0x40080FFF`	4KB	NSACFG	Non-secure Access Configuration Register Block. See Non-secure Access Configuration Register Block.	-	NS
8	`0x40081000`	`0x4008FFFF`	-	Reserved	Reserved (**RAZ/WI**)	-	-
9	`0x40090000`	`0x40093FFF`	16KB	CryptoCell312	CryptoCell 312	-	NS
10	`0x40094000`	`0x400FFFFF`	-	Reserved	Reserved	-	-
11	`0x48000000`	`0x48000FFF`	4KB	TIMER0	Timer 0. See Timestamp Timers.	33	NS_PPC
12	`0x48001000`	`0x48001FFF`	4KB	TIMER1	Timer 1. See Timestamp Timers.	34	
13	`0x48002000`	`0x48002FFF`	4KB	TIMER2	Timer 2. See Timestamp Timers.	35	
14	`0x48003000`	`0x48003FFF`	4KB	TIMER3	Timer 3. See Timestamp Timers.	36	
15	`0x48004000`	`0x4800FFFF`	-	Reserved	Reserved (**RAZ/WI**)	-	-
16	`0x48040000`	`0x48040FFF`	4KB	NSWDCTRL	Non-secure Watchdog Control Frame. See Timestamp Watchdogs.	-	NS
17	`0x48041000`	`0x48041FFF`	4KB	NSWDREF	Non-secure Watchdog Refresh Frame. See Timestamp Watchdogs.	-	
18	`0x48042000`	`0x4804FFFF`	-	Reserved	Reserved (**RAZ/WI**)	-	-
19	`0x48050000`	`0x480FFFFF`	-	Reserved	Reserved	-	-
20	`0x50000000`	`0x50000FFF`	4KB	MHU0	Message Handling Unit 0. See Message Handling Unit.	1	S_PPC

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Row ID	Address		Size	Region name	Description	Alias with row ID	Security[1]
	From	To					
21	0x50001000	0x50001FFF	4KB	MHU1	Message Handling Unit 1. See Message Handling Unit.	2	
22	0x50002000	0x5000FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
23	0x50040000	0x5007FFFF	-	Reserved	Reserved	-	-
24	0x50080000	0x50080FFF	4KB	SACFG	Secure Access Configuration Register Block. See Secure Access Configuration Register Block.	-	S
25	0x50081000	0x50082FFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
26	0x50083000	0x50083FFF	4KB	VM0MPC[2]	VM0 Memory Protection Controller. See Volatile Memory.	-	S
27	0x50084000	0x50084FFF	4KB	VM1MPC[2]	VM1 Memory Protection Controller. See Volatile Memory.	-	
28	0x50085000	0x50085FFF	4KB	VM2MPC[2]	VM2 Memory Protection Controller. See Volatile Memory.	-	
29	0x50086000	0x50086FFF	4KB	VM3MPC[2]	VM3 Memory Protection Controller. See Volatile Memory.	-	
30	0x50087000	0x5008FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
31	0x50090000	0x50093FFF	16KB	CryptoCell312	CryptoCell 312	-	-
32	0x50094000	0x500FFFFF	-	Reserved	Reserved	-	-
33	0x58000000	0x58000FFF	4KB	TIMER0	Timer 0. See Timestamp Timers.	11	S_PPC
34	0x58001000	0x58001FFF	4KB	TIMER1	Timer 1. See Timestamp Timers.	12	
35	0x58002000	0x58002FFF	4KB	TIMER2	Timer 2. See Timestamp Timers.	13	
36	0x58003000	0x58003FFF	4KB	TIMER3	Timer 3. See Timestamp Timers.	14	
37	0x58004000	0x5800FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
38	0x58040000	0x58040FFF	4KB	SWDCTRL	Secure Watchdog Control Frame. See Timestamp Watchdogs.	-	S
39	0x58041000	0x58041FFF	4KB	SWDREF	Secure Watchdog Refresh Frame. See Timestamp Watchdogs.	-	
40	0x58042000	0x5804FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
41	0x58050000	0x580FFFFF	-	Reserved	Reserved	-	-

1. 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.
2. The number of VM<*n*>MPC regions depends on NUMVMBANK. If VM<*n*> does not exist, then the VM<*n*>MPC region is Reserved.
3. MHU0 and MHU1 will only exist if NUMCPU > 0.

6.4.1 Message Handling Unit registers

SSE-300 implements up to two Message Handling Units (MHU). These allow software to raise interrupts to the CPU cores. Both MHU are mapped twice into both Secure and non-Secure regions as follows, and a PPC then controls in which area each MHU will reside:

- MHU0 in non-Secure region at `0x48000000` and Secure region at `0x58000000`

- MHU1 in non-Secure region at `0x48001000` and Secure region at `0x58001000`

If there is only one CPU core in the system, both MHUs will not exist and the two regions will be reserved and any accesses to them will result in **RAZ/WI**. For write access to these registers, only 32 bit writes are supported. Any Byte and Half word writes will result in its write data ignored.

Note

There are two MHU in the system so that software can map one for Secure use and another for non-Secure use. However, Software can choose to map both to Secure or non-Secure if needed.

Each MHU has the following register map shown in the table below, where *n* is 0 to NUMCPU. The security of each MHU register area is defined by PPCs drive by registers in Secure Access Configuration Register Block and Non-Secure Access Configuration Register Block. See PERIPHNSPPC0 and PERIPHNSPPC1, PERIPHSPPPC0 and PERIPHSPPPC1, and PERIPHNSPPPC0 and PERIPHNSPPPC1. All registers reside in the PD_SYS power domain and is reset by **nWARMRESETSYS**. The SSE-300 defines up to two MHUs.

Table 6-3: Message Handling Unit (MHU) Register Map

Offset	Name	Access	Reset Value	Description
`0x000` + n(0x010)	CPU<n>INTR_STAT[1]	Read-only	`0x00000000`	CPU <n> Interrupt Status Register
`0x004` + n(0x010)	CPU<n>INTR_SET[1]	Write-only	`0x00000000`	CPU <n> Interrupt Set Register
`0x008` + n(0x010)	CPU<n>INTR_CLR[1]	Write-only	`0x00000000`	CPU <n> Interrupt Clear Register
`0x00C` + n(0x010)	Reserved	Read-only	`0x00000000`	Reserved
(NUMCPU+1)(0x010) - `0xFC8`	Reserved	Read-only	`0x00000000`	Reserved
`0xFD0`	PIDR4	read-only	`0x00000004`	Peripheral ID 4
`0xFD4` - `0xFDC`	Reserved	read-only	`0x00000000`	Reserved
`0xFE0`	PIDR0	read-only	`0x00000056`	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

1. There is a set of these register per CPU that exist with n in 0 to NUMCPU.

6.4.2 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 will result in its write data 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		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		0x00000000	Reserved
0x01C	SECMPCINTSTAT	Read-only	0x00000000	Secure MPC Interrupt Status
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		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		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		0x00000000	Reserved
0x050	MAINNSPPC0		0x00000000	Reserved
0x054 - 0x05C	Reserved		0x00000000	Reserved

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
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: • 1 - Allow Non-secure access • 0 - Disallow Non-secure access Resets to 0.
0x074	PERIPHNSPPC1	Read-write	0x00000000	Non-secure Access Peripheral Protection Control 1 on Peripheral Interconnect
0x078 - 0x07C	Reserved		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
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	MAINSPPPC0		0x00000000	Reserved
0x094 - 0x09C	Reserved		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	PERIPHSPPPC0	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 - 0x0BC	Reserved		0x00000000	Reserved
0x0C0	PERIPHSPPPCEXP0	Read-write	0x00000000	Expansion 0 Secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
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 - 0xFCC	Reserved		0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved		0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000052	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

6.4.2.1 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	RO	0x00000000	-	Reserved.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
0	W1S	0x00	SPCSECCFGLOCK	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: • NSCCFG, • MAINNSPPC0, • MAINNSPPCEXP<N>, • PERIPHNSPPC0, • PERIPHNSPPC1, • PERIPHNSPPCEXP<N>, • MAINSPPPC0, • MAINSPPPCEXP<N>, • PERIPHSPPPC0, • PERIPHSPPPC1, • PERIPHSPPPCEXP<N>, • NSMSCEXP.

6.4.2.2 BUSWAIT

The Bus Access Wait register allows software to gate access entering the interconnect from specific masters 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	RO	0x0000	-	Reserved
16	RO	0x00	ACC_WAITN_STATUS	This status register indicates the status of any gating units that are used to block bus access to the system: 1 - Allow access 0 - Block access This register reflects the combined status of all gating units in the system, including status on the input signal **ACCWAITNSTATUS**, expected to be driven from external gating units. Both ACC_WAITN_STATUS and ACC_WAITN together, ensuring that software can determine that all gates have reached the state that is requested.
15:1	RO	0x0000	-	Reserved

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
0	RW	ACCWAITNRST	ACC_WAITN	Request gating units to block bus access to system: 1 - Allow access 0 - Block access 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 **ACCWAITn**. Both ACC_WAITN_STATUS and ACC_WAITN together, ensuring that software can determine that all gates have reached the state that is requested.

6.4.2.3 SECRESPCFG

The Security Violation Response Configuration register is used to define a response to an access that causes security violation on the bus fabric.

Table 6-7: SECRESPCFG Register

Bits	Type	Default	Name	Description
31:1	RO	`0x00000000`	-	Reserved
0	RW	`0x00`	SECRESPCFG	This field configures the response in case of a security violation: • 0 - Read-Zero Write Ignore • 1 - Bus error Note that some components, for example, the AHB Memory Protection Controllers (MPC), provide their own control registers to configure their own response.

6.4.2.4 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	RO	`0x00000000`	-	Reserved.
1	RW	`0x00`	RAMNSC	Configures if the region `0x30000000` to `0x3FFFFFFF` is Non-secure Callable: • 0 - Not Non-secure Callable • 1 - Non-secure Callable
0	RW	`0x00`	CODENSC	Configures if the CODE region `0x10000000` to `0x1FFFFFFF` is Non-secure Callable: • 0 - Not Non-secure Callable • 1 - Non-secure Callable

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.4.2.5 SECMPCINTSTAT

The interrupt signals from all Memory Protection Controllers (MPC), both within SSE-300 and in the expansion logic are merged and sent to the processors 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: SECMPCINTSTAT register

Bits	Type	Default	Name	Description
31:16	RO	`0x00000000`	SMPCEXP_STATUS	Interrupt Status for Expansion Memory Protection Controller. Each bit *n* (0 to 15) local in this field shows the status of input signal **SMPCEXPSTATUS[n]**. The MPCEXPDIS configuration point 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	RO	`0x00000000`	-	Reserved
3	RO	`0x00`	-	Reserved for SMPCVM3_STATUS. Interrupt Status for Memory Protection Controller of Volatile Memory Bank 3. This register is not used and reserved if NUMVMBANK < 4
2	RO	`0x00`	-	Reserved for SMPCVM2_STATUS. Interrupt Status for Memory Protection Controller of Volatile Memory Bank 2. This register is not used and reserved if NUMVMBANK < 3
1	RO	`0x00`	SMPCVM1_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 1. This register is not used and reserved if NUMVMBANK < 2
0	RO	`0x00`	SMPCVM0_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 0. This register is not used and reserved if NUMVMBANK < 1

6.4.2.6 SECPPCINTSTAT, SECPPCINTCLR and SECPPCINTEN

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 processors. 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.

Table 6-10: SECPPCINTSTAT Register

Bits	Type	Default	Name	Description
31:24	RO	`0x000`	-	Reserved
23:20	RO	`0x00`	SMAINPPCEXP_STATUS	Interrupt status of Expansion Peripheral Protection Controller on the Main Interconnect. Each bit n (0 to 3) local in this field captures the active state of the input signal **SMAINPPCEXPSTATUS[n]**.
19:17	RO	`0x00`	-	Reserved
16	RO	`0x00`	SMAINPPC0_STATUS	Reserved
15:8	RO	`0x000`	-	Reserved
7:4	RO	`0x00`	SPERIPHPPCEXP_STATUS	Interrupt status of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Each bit n (0 to 3) local in this field captures the active state of the input signal **SPERIPHPPCEXPSTATUS[n]**.
3:2	RO	`0x00`	-	Reserved
1	RO	`0x00`	SPERIPHPPC1_STATUS	Interrupt status of Peripheral Protection Controller Group 1 on the Peripheral Interconnect within the System

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
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	RO	0x000	-	Reserved
23:20	W1T	0x00	SMAINPPCEXP_CLR	Interrupt clear of Expansion Peripheral Protection Controller on the Main Interconnect. Each bit n, when set to HIGH clears the interrupt status of the PPC connected to **SMAINPPCEXPSTATUS[n]** and **SMAINPPCEXPCLEAR[n]**.
19:17	RO	0x00	-	Reserved
16	W1T	0x00	SMAINPPC0_CLR	Reserved
15:8	RO	0x000	-	Reserved
7:4	W1T	0x00	SPERIPHPPCEXP_CLR	Interrupt clear of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Each bit n, when set to HIGH clears the interrupt status of the PPC connected to **SPERIPHPPCEXPSTATUS[n]** and **SPERIHPPCEXPCLEAR[n]**.
3:2	RO	0x00	-	Reserved
1	W1T	0x00	SPERIPHPPC1_CLR	Interrupt clear of Peripheral Protection Controller 1 on the Peripheral Interconnect within the System
0	W1T	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	RO	0x000	-	Reserved
23:20	RW	0x00	SMAINPPCEXP_EN	Interrupt enable of Expansion Peripheral Protection Controller on the Main Interconnect. Write 1 to bit n (0 to 3) local in this field to enable interrupt from **SMAINPPCEXPSTATUS[n]**.
19:17	RO	0x00	-	Reserved
16	RW	0x00	SMAINPPC0_EN	Reserved
15:8	RO	0x000	-	Reserved
7:4	RW	0x00	SPERIPHPPCEXP_EN	Interrupt enable of Expansion Peripheral Protection Controller on the Peripheral Interconnect. Write 1 to bit n (0 to 3) local in this field to enable interrupt from **SPERIPHPPCEXPSTATUS[n]**.
3:2	RO	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.

6.4.2.7 SECMSCINTSTAT, SECMSCINTCLR and SECMSCINTEN

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 n (0 to 15) local in this field captures the active state of the input signal **SMSCEXPSTATUS[n]**. The configuration point 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.
15:0	RO	`0x0000`	-	Reserved

Table 6-14: SECMSCINTCLR Register

Bits	Type	Default	Name	Description
31:16	W1T	`0x0000`	SMSCEXP_CLR	Interrupt clear for expansion MSC. Each bit n, when set to HIGH, clears the interrupt status of the MSC connected to **SMSCEXPSTATUS[n]** and **SMSCEXPCLEAR[n]**. The configuration point 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.
15:0	RO	`0x0000`	-	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 **SMSCEXPSTATUS[n]**. The configuration point 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.
15:0	RO	`0x0000`	-	Reserved

6.4.2.8 BRGINTSTAT, BRGINTCLR and BRGINTEN

SSE-300 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 slave interfaces before any potential error response is received for the write access on their master 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 **BRGEXPSTATUS[n]**. The configuration point 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	RO	`0x0000`	-	Reserved

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-17: BRGINTCLR Register

Bits	Type	Default	Name	Description
31:16	W1T	`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 **BRGEXPSTATUS[n]** and **BRGEXPCLEAR[n]**. The configuration point BRGEXPDIS defines if each bit within this register is actually implement such that if BRGEXPDIS[i] = 1'b1 then BRGEXPCLR[i] is disabled and any writes to it is ignored.
15:0	RO	`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 **BRGEXPSTATUS[n]**. The configuration point 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	RO	`0x0000`	-	Reserved

6.4.2.9 MAINNSPPC0

The Main Interconnect Non-secure Access Peripheral Protection Controller register allows software to configure if each peripheral on the Main 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-300 currently does not have any internal interfaces on the Main Interconnect that require security configuration support of the PPC. Therefore, this register is reserved.

Table 6-19: MAINNSPPC0 Register

Bits	Type	Default	Name	Description
31:0	RO	`0x00000000`	-	Reserved

6.4.2.10 MAINNSPPCEXP<0 to 3>

The Main Interconnect Non-secure Access Slave Peripheral Protection Controller Expansion register 0, 1, 2, and 3 allows software to configure each Main Interconnect peripheral that it controls through each PPC that resides in the subsystem expansion outside the subsystem. 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:

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-20: MAINNSPPCEXP<N> Register

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

6.4.2.11 PERIPHNSPPC0 and PERIPHNSPPC1

The Peripheral Interconnect Non-secure Access Peripheral Protection Controller registers allow software to configure if 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-300 has two such group of registers, Peripheral Protection Controller Group 0 and Peripheral Protection Controller Group 1 as follows:

Table 6-21: PERIPHNSPPC0 Register

Bits	Type	Default	Name	Description
31:8	RO	0x000000	-	Reserved.
7	RO	0x00	-	Reserved for NS_SYSDSS. Access Non-Security for interconnect access to debug system. When HASCSS = 0, this field is reserved and **RAZ/WI**.
6	RO	0x00	-	Reserved
5	RW	0x00	NS_TIMER3	Access Non-Security for TIMER3
4	RO	0x00	-	Reserved for NS_MHU1. Access Non-Security for MHU 1. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
3	RO	0x00	-	Reserved for NS_MHU0. Access Non-Security for MHU 0. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
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 6-22: PERIPHNSPPC1 Register

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

Access to the control interfaces of the Memory Protection Controllers in the system are filtered by PPC0 though its security settings are fixed and are not represented in the table above. This is the same with SLOWCLK Watchdog timer's control interface which is filtered by PPC group 1. Hence if accessed using the wrong security at their configuration interfaces, it will also result in interrupts being raised for PPC0 or PPC1 registers respectively.

6.4.2.12 PERIPHNSPPCEXP<0 to 3>

The Peripheral Interconnect Non-secure Access Slave Peripheral Protection Controller Expansion register 0, 1, 2, and 3 allows software to configure each Peripheral Interconnect peripheral that it controls through each PPC that resides in the expansion logic outside the subsystem. 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 6-23: PERIPHNSPPCEXP<N> Register

Bits	Type	Default	Name	Description
31:16	RO	`0x0000`	-	Reserved
15:0	RW	`0x0000`	PERIPHNSPPCEXP<N>	Expansion <*N*> Non-secure Access Peripheral Interconnect Slave Peripheral Protection Control. Each bit *n* drives the output signal **PERIPHNSPPCEXP<N>[n]**. The configuration point PERIPHPPCEXP<N>DIS defines if each bit within this register is actually implement such that if PERIPHPPCEXP<N>PDIS[i] = 1'b1 then PERIPHPPCEXP<N>DIS[i] reads as zeros and any writes to it is ignored.

6.4.2.13 MAINSPPPC0

Secure Unprivileged Access Main Interconnect Slave Peripheral Protection Controller Register allows software to configure if each peripheral on the Main Interconnect that it controls through a PPC is only Secure Privileged Secure 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-300 currently does not have any internal interfaces on the Main Interconnect that require Secure Unprivileged Access configuration support of the PPC. Therefore, this register is reserved.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-24: MAINSPPPC0 Register

Bits	Type	Default	Name	Description
31:0	RO	0x00000000	-	Reserved.

6.4.2.14 MAINSPPPCEXP<0 to 3>

The Expansion Secure Unprivileged Access Main Interconnect Slave Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure each Main Interconnect peripheral that it controls through each PPC, that resides in the expansion logic outside the subsystem, 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

These directly controls 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 6-25: MAINSPPPCEXP<N> Register

Bits	Type	Default	Name	Description
31:16	RO	0x0000	-	Reserved
15:0	RW	0x0000	MAINSPPPCEXP<N>	Expansion <N> Secure Unprivileged Access Main Interconnect Slave Peripheral Protection Control. Each bit n will drive the output signal **MAINPPPCEXP<N>[n]** if MAINNSPPCEXP<N>[n] is also LOW, where N is 0 to 3. The configuration point MAINPPCEXP<N>DIS defines if each bit within this register is actually implement such that if MAINPPCEXP<N>DIS[i] = 1'b1 then MAINSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

6.4.2.15 PERIPHSPPPC0 and PERIPHSPPPC1

Secure Unprivileged Access Peripheral Interconnect Slave 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-300 has two such registers as follows:

Table 6-26: PERIPHSPPPC0 Register

Bits	Type	Default	Name	Description
31:8	RO	0x000000	-	Reserved
7	RO	0x00	-	Reserved for SP_SYSDSS. Secure Unprivileged setting for interconnect access to Debug System. When HASCSS = 0, this field is reserved and RAZ/WI.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
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	RO	0x00	-	Reserved for SP_MHU1. Secure Unprivileged setting for MHU 1. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
3	RO	0x00	-	Reserved for SP_MHU0. Secure Unprivileged setting for MHU 0. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
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 6-27: PERIPHSPPPC1 Register

Bits	Type	Default	Name	Description
31:2	RO	0x00000000	-	Reserved
0	RW	0x00	SP_SLOWCLK_TIMER	Secure Unprivileged setting for SLOWCLK_TIMER

6.4.2.16 PERIPHSPPPCEXP<0 to 3>

The Expansion Secure Unprivileged Access Peripheral Interconnect Slave Peripheral Protection Controller register 0, 1, 2, and 3 allows software to configure each Peripheral Interconnect peripheral that it controls through each PPC, that resides in the expansion logic outside the subsystem, 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

These directly controls 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	RO	0x0000	-	Reserved.
15:0	RW	0x0000	PERIPHSPPPCEXP<N>	Expansion <*N*> Secure Unprivileged Access Peripheral Interconnect Slave Peripheral Protection Control. Each bit n drives the output signal **PERIPHPPPCEXP<N>[n]** if PERIPHNSPPCEXP<N>[n] is also LOW, where N is 0 to 3. The configuration point PERIPHPPCEXP<N>DIS defines if each bit within this register is actually implement 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.4.2.17 NSMSCEXP

The Non-secure Expansion Master Security Controller register allows software to configure if each master 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	NSMSCEXPRST	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 **NSMSCEXP[n]**. Set HIGH to define a Master 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 NSMSCEXPRST.
15:0	RO	`0x0000`	-	Reserved

6.4.3 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 will result in its write data 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	-	`0x00000000`	Reserved
`0x090`	MAINNSPPPC0		`0x00000000`	Reserved
`0x094` - `0x09C`	Reserved	-	`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.
`0x0B0`	PERIPHNSPPPC0	Read-write	`0x00000000`	Non-secure Unprivileged Access Peripheral Protection Control 0 on Peripheral Interconnect.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
0x0B4	PERIPHNSPPPC1	Read-write	0x00000000	Non-secure Unprivileged Access Peripheral Protection Control 1 on Peripheral Interconnect.
0x0B8 - 0x0BC	Reserved		0x00000000	Reserved
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		0x00000000	Reserved
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved		0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000053	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

6.4.3.1 MAINNSPPPC0

Non-secure Unprivileged Access Main Interconnect Slave Peripheral Protection Controller register allows software to configure if each peripheral on the Main Interconnect 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

SSE-300 does not have any internal interfaces on the Main Interconnect that require Non-secure Unprivileged Access configuration support of the PPC. Therefore, this register is reserved.

Table 6-31: MAINNSPPPC0 register

Bits	Type	Default	Name	Description
31:0	RO	0x00000000	-	Reserved

6.4.3.2 MAINNSPPPCEXP<0 to 3>

The Expansion Non-secure Unprivileged Access Main Interconnect Slave Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure each Main Interconnect peripheral that it controls through each PPC, that resides in the expansion logic outside the subsystem, 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

These directly controls 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 6-32: MAINNSPPPCEXP<N> Register

Bits	Type	Default	Name	Description
31:16	RO	0x0000	-	Reserved
15:0	RW	0x0000	MAINNSPPPCEXP<N>	Expansion <*N*> Non-secure Privilege Access Main Interconnect Slave Peripheral Protection Control. Each bit n drives the output signal **MAINPPPCEXP<N>[n]** if MAINNSPPCEXP<N>[n] is also HIGH, where N is 0 to 3. The configuration point MAINPPCEXP<N>DIS defines if each bit within this register is actually implement such that if MAINPPCEXP<N>DIS[i] = 1'b1 then MAINNSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

6.4.3.3 PERIPHNSPPPC0 and PERIPHNSPPPC1

Non-secure Unprivileged Access Peripheral Interconnect Slave 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-300 has two such registers as follows:

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-33: PERIPHNSPPPC0 Register

Bits	Type	Default	Name	Description
31:6	RO	0x0000000	-	Reserved
7	RO	0x00	-	Reserved for NSP_SYSDSS. Non-secure Privileged setting for Debug System. When HASCSS = 0, this field is reserved and **RAZ/WI**.
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
4	RO	0x00	-	Reserved for NSP_MHU1. Non-secure Unprivileged setting for MHU 1. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
3	RO	0x00	-	Reserved for NSP_MHU0. Non-secure Unprivileged setting for MHU 0. When NUMCPU = 0, this field is reserved and **RAZ/WI**.
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 6-34: PERIPHNSPPPC1 Register

Bits	Type	Default	Name	Description
31:2	RO	0x00000000	-	Reserved
0	RW	0x00	NSP_SLOWCLK_TIMER	Non-secure Unprivileged setting for SLOWCLK_TIMER

6.4.3.4 PERIPHNSPPPCEXP<0 to 3>

The Expansion Non-secure Unprivileged Access Peripheral Interconnect Slave Peripheral Protection Controller register 0, 1, 2, and 3 allows software to configure each Peripheral Interconnect peripheral that it controls through each PPC, that resides in the expansion logic outside the subsystem, 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

These directly controls 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	RO	0x0000	-	Reserved.
15:0	RW	0x0000	PERIPHNSPPPCEXP<N>	Expansion <*N*> Non-secure Unprivileged Access Peripheral Interconnect Slave Peripheral Protection Control. Each bit n drives the output signal **PERIPHPPPCEXP<N>[n]** if PERIPHNSPPCEXP<N>[n] is also HIGH, where N is 0 to 3. The configuration point PERIPHPPCEXP<N>DIS defines if each bit within this register is actually implement such that if PERIPHPPCEXP<N>DIS[i] = 1'b1 then PERIPHNSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.4.4 Timestamp Timers

SSE-300 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. See Secure access configuration register block.

All timestamp timers and watchdog, except for Timer3, resides 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**.

The following table lists the registers implemented in a timestamp-based timer.

Table 6-36: Timestamp-based timer register map

Offset	Name	Access	Reset Value	Description
`0x000`	CNTPCT[31:0]	Read-only	`0xXXXXXXXX`	Physical Count Register Lower Word
`0x004`	CNTPCT[63:32]	Read-only	`0xXXXXXXXX`	Physical Count Register Higher Word
`0x008 - 0x00C`	Reserved	Read-only	`0x00000000`	Reserved
`0x010`	CNTFRQ	Read-write	`0x00000000`	Counter Frequency Register
`0x014 - 0x01C`	Reserved	Read-only	`0x00000000`	Reserved
`0x020`	CNTP_CVAL[31:0]	Read-write	`0x00000000`	Timer Compare Value Lower Word Register
`0x024`	CNTP_CVAL[63:32]	Read-write	`0x00000000`	Timer Compare Value Higher Word Register
`0x028`	CNTP_TVAL	Read-write	`0xXXXXXXXX`	Timer Value register
`0x02C`	CNTP_CTL	Read-write	`0xXXXXXXXX`	Timer Control register
`0x030 - 0x3C`	Reserved	Read-only	`0x00000000`	Reserved
`0x040`	CNTP_AIVAL[31:0]	Read-only	`0xXXXXXXXX`	AutoIncrValue Lower Word Register
`0x044`	CNTP_AIVAL[63:32]	Read-only	`0xXXXXXXXX`	AutoIncrValue Higher Word Register
`0x048`	CNTP_AIVAL_RELOAD	Read-write	`0xXXXXXXXX`	AutoIncrValue Reload register
`0x04C`	CNTP_AIVAL_CTL	Read-write	`0xXXXXXXXX`	AutoIncrValue Control register
`0x050`	CNTP_CFG	Read-only	`0x00000001`	Timer Configuration register
`0x054 - 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	`0x000000B7`	Peripheral ID 0
`0xFE4`	PIDR1	Read-only	`0x000000B0`	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

For more information of the ARMv8M System Counter Timer registers, see *Arm® SSE-123 Example Subsystem Technical Reference Manual*.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.4.5 Timestamp Watchdogs

SSE-300 implements two timestamp-based timer 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. Each watchdog accessibility to unprivileged access is also controlled by through PPC0. See PERIPHSPPPC0 and PERIPHSPPPC1 and PERIPHNSPPPC0 and PERIPHNSPPPC1.

Each Watchdog timer implements two register frames, a Control Frame and a Refresh Frame. The following tables list the registers implemented in a timestamp-based watchdog.

Table 6-37: Timestamp-based generic watchdog control frame register map

Offset	Name	Access	Reset value	Description
0x000	WCS	Read-write	0x00000000	Watchdog Control and Status
0x004	Reserved	Read-only	0x00000000	Reserved
0x008	WOR	Read-write	0x00000000	Watchdog Offset Register
0x00C	Reserved	Read-only	0x00000000	Reserved
0x010	WCV[31:0]	Read-write	0x00000000	Watchdog Compare Value Lower Word
0x014	WCV[63:32]	Read-write	0x00000000	Watchdog Compare Value Higher Word
0x018 - 0xFCB	Reserved	Read-only	0x00000000	Reserved
0xFCC	W_IIDR	Read-only	0x0000143B	Watchdog Interface Identification Register
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x000000B1	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B0	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

Table 6-38: Timestamp-based generic watchdog refresh frame register map

Offset	Name	Access	Reset Value	Description
0x000	WRR	write-only	0x00000000	Watchdog Refresh Register
0x004 - 0xFCB	Reserved	Read-only	0x00000000	Reserved
0xFCC	W_IIDR	Read-only	0x0000143B	Watchdog Interface Identification Register
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x000000B0	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000B0	Peripheral ID 1
0xFE8	PIDR2	Read-only	0x0000002B	Peripheral ID 2

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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

For more information of the ARMv8M Timestamp Watchdog registers, see *Arm® SSE-123 Example Subsystem Technical Reference Manual*.

6.5 Processor Private Region

Each processor in the system has its own copy of the Processor Private Region which is only assessable to itself. Each 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

Each of these regions are not accessible from any other master in the system, including from the expansion slave interfaces on the Main and Peripheral Interconnect, except through the external debugger through the local CPUs. Of the four regions above only `0x40010000` to `0x4001FFFF` and `0x50010000` to `0x5001FFFF` implements any registers.

The memory map of the Processor Private Region is as follows:

Note

In this section and the following subsections CPU<N> always refers to CPU0.

Table 6-39: Processor Private Region address map

Row ID	Address - from	Address - to	Size	Region name	Description	Alias with row ID	Security[1]
0	`0x40010000`	`0x40011FFF`	-	Reserved	Reserved	-	-
1	`0x40012000`	`0x40012FFF`	4KB	CPU<N>_PWRCTRL	CPU<N> Power Control Block. See CPU<N>_PWRCTRL Register Block	7	NS, P
2	`0x40013000`	`0x4001EFFF`	-	Reserved	Reserved	-	-
3	`0x4001F000`	`0x4001FFFF`	4KB	CPU<N>_IDENTITY	CPU<N> Identity Block, See CPU<N>_IDENTITY Register Block	9	NS, UP
4	`0x48010000`	`0x4801FFFF`	-	Reserved	Reserved	-	-
5	`0x50010000`	`0x50010FFF`	-	Reserved	Reserved	-	-

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Row ID	Address - from	Address - to	Size	Region name	Description	Alias with row ID	Security[1]
6	`0x50011000`	`0x50011FFF`	4KB	CPU<N>_SECCRTL	CPU<N> Local Security Control Block, See CPU<N>_SECCTRL Register Block	-	S, P
7	`0x50012000`	`0x50012FFF`	4KB	CPU<N>_PWRCTRL	CPU<N> Power Control Block. See CPU<N>_PWRCTRL Register Block	1	S, P
8	`0x50013000`	`0x5001EFFF`	-	Reserved	Reserved	-	-
9	`0x5001F000`	`0x5001FFFF`	4KB	CPU<N>_IDENTITY	CPU<N> Identity Block, See CPU<N>_IDENTITY Register Block	3	S, UP
10	`0x58010000`	`0x5801FFFF`	-	Reserved	Reserved	-	-

1. S: Secure access only. NS: Non-secure access only

   P: Privilege access only. UP: Unprivileged and privilege access allowed

6.5.1 CPU<N>_PWRCTRL Register Block

SSE-300 implements a CPU<N>_PWRCTRL register block for each CPU <N> in the subsystem, where N is 0 to NUMCPU. All blocks reside at address `0x40012000` in a Non-secure region and is also alias to `0x50012000` in the Secure region. Each CPU<N> can only see its own CPU<N>_PWRCTRL registers. These are read only registers when accessed from the Non-secure region starting at address `0x40012000` and any writes access to it in that region will be ignored.

The following table lists the registers in each CPU<N>_PWRCTRL register block.

Table 6-40: CPU<n>_PWRCTRL register map

Offset	Name	Access	Reset value	Description
`0x000`	CPUPWRCFG	Read-write	`0x00000000`	CPU <N> Local Power Configuration register. See CPUPWRCFG.
`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, provide 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

reset domain, nWARMRESETCPU<N> and power domain, PD_CPU<N> as its associated CPU core so that when the CPU is powered down, the register is also powered down and are cleared when powered back up.

Table 6-41: CPU<N>PWRCFG Register

Bits	Type	Default	Name	Description
31:5	RO	`0x0000000`	-	Reserved.
4	RW	`0x00`	TCM_MIN_PWR_STATE	Defines the minimum power state of the TCM for CPU <N>. • '0' - OFF • '1' - Retention. This bit is read access only from the Non-secure world. When PD_CPU<N> returns from MEM_RET or MEM_RET_NOCACHE state to one of the ON states, this bit is set to 1'b1.
3:1	RO	`0x00`	-	Reserved
0	RW	`0x00`	USEIWIC	When HIGH, Select the use of IWIC for CPU<N> when in DeepSleep. Else select the use of EWIC. If CPU<N>HASIWIC for this CPU<N> is 0, this field is reserved and is **RAZ/WI**. This bit is read access only from the Non-secure world.

6.5.2 CPU<N>_IDENTITY Register Block

SSE-300 implements a CPU<N>_IDENTITY register block for each CPU<N> in the subsystem, where N is 0 to NUMCPU. All blocks reside at address `0x4001F000` in a Non-secure region and is also alias to `0x5001F000` in the Secure region. Each CPU<N> can only see its own CPU<N>_IDENTITY registers. These are read only registers and any writes access to it will be ignored.

The following table lists the registers in each CPU<N>_IDENTITY block.

Table 6-42: CPU<N>_IDENTITY register map

Offset	Name	Access	Reset value	Description
`0x000`	CPUID	Read-only	CPU<N>CPUIDRST	Unique CPU Identity Number, where <N> is used for the view that CPU <N> sees. See CPUID.
`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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
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 CPU <N>, provides an identity code to the CPU that is unique to that CPU.

Table 6-43: CPUID Register

Bits	Type	Default	Name	Description
31:4	RO	0x0000000	-	Reserved.
3:0	RO	CPU<N>CPUIDRST	CPUID	CPU Identity. Defined by configuration CPU<N>CPUIDRST. The identity value for each CPU <N> must be unique.

6.5.3 CPU<N>_SECCTRL Register Block

Each CPU <N> in the system, where N is 0 to NUMCPU, has associated with it a CPU<N>_SECCTRL register block that allow the security locks of each CPU to be configured. Each register block resides in the same reset domain, **nWARMRESETCPU<N>** and power domain as its associated processor core 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 each CPU<N>_SECCTRL Register block.

Table 6-44: CPU<N>_SECCTRL register map

Offset	Name	Access	Reset value	Description
0x000	CPUSECCFG	Read-write	0x00000000	CPU Local Security Configuration. See CPUSECCFG
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	0x00000059	Peripheral ID 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
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 6-45: CPUSECCFG Register.

Bits	Type	Default	Name	Description
31:6	RO	0x00000000	-	Reserved.
5	W1S	0x00	LOCKDTGU	When HIGH, disables writes to the CPU <N> 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	W1S	0x00	LOCKITGU	When HIGH, disables writes to the CPU <N> 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	W1S	0x00	LOCKTCM	When HIGH, disables writes to the CPU <N> ITCMCR, DTCMCR from software or from a debug agent connected to the processor. Once set to high, it cannot be cleared until Reset.
2	W1S	0x00	LOCKSMPU	When HIGH, disables write to the CPU <N> 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	W1S	0x00	LOCKSAU	When HIGH, disables writes to the CPU <N> 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	W1S	0x00	LOCKSVTAIRCR	When HIGH, disables writes to the CPU <N> 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`.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- `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 6-46: System Control Peripheral Region address map

Row ID	Address - from	Address - to	Size	Region Name	Description	Alias with row ID	Security[1]
1	`0x40020000`	`0x4003FFFF`	128KB	Reserved	Reserved. When accessed, results in bus error.		
2	`0x48020000`	`0x48020FFF`	4KB	SYSINFO	System Information Register Block. See SYSINFO Register Block.		NS, UP
3	`0x48021000`	`0x4802EFFF`	56KB	Reserved	Reserved. When accessed, results in **RAZ/WI**.		NS, UP
4	`0x4802F000`	`0x4802FFFF`	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See SLOWCLK AON Timers.	28	NS-PPC, P-PPC
5	`0x48030000`	`0x4803FFFF`	64KB	Reserved	Reserved. When accessed, results in bus error.		
6	`0x50020000`	`0x5003FFFF`	128KB	Reserved	Reserved. When accessed, results in bus error.		
7	`0x58020000`	`0x58020FFF`	4KB	SYSINFO	System Information Register Block. See SYSINFO Register Block.		S, UP
8	`0x58021000`	`0x58021FFF`	4KB	SYSCONTROL	System Control Register Block. See SOC_IDENTITY.		S, P
18	`0x58022000`	`0x58022FFF`	4KB	SYS_PPU	PPU for BR_SYS. See Power Policy Units.		S, P
19	`0x58023000`	`0x58023FFF`	4KB	CPU0_PPU	PPU for BR_CPU0. See Power Policy Units.		S, P
20	`0x58024000`	`0x58024FFF`	4KB	Reserved	Reserved for CPU1_PPU. PPU for BR_CPU1. **RAZ/WI** when accessed.		
21	`0x58025000`	`0x58025FFF`	4KB	Reserved	Reserved for CPU2_PPU. PPU for BR_CPU2. **RAZ/WI** when accessed.		
22	`0x58026000`	`0x58026FFF`	4KB	Reserved	Reserved for CPU3_PPU. PPU for BR_CPU3. **RAZ/WI** when accessed.		
23	`0x58027000`	`0x58027FFF`	4KB	Reserved	Reserved for CRYPTO_PPU. PPU for BR_CRYPTO. **RAZ/WI** when accessed.		
24	`0x58028000`	`0x58028FFF`	4KB	MGMT_PPU	PPU for BR_MGMT. See Power Policy Units.		S, P
25	`0x58029000`	`0x58029FFF`	4KB	DEBUG_PPU	PPU for BR_DEBUG. See Power Policy Units.		S, P
26	`0x5802A000`	`0x5802DFFF`	16KB	Reserved	Reserved. When accessed, results in **RAZ/WI**.		
27	`0x5802E000`	`0x5802EFFF`	4KB	SLOWCLK Watchdog	Watchdog Timer running on SLOWCLK. See SLOWCLK AON Timers.		S, P

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Row ID	Address - from	Address - to	Size	Region Name	Description	Alias with row ID	Security[1]
28	`0x5802F000`	`0x5802FFFF`	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See SLOWCLK AON Timers.	4	S-PPC, P-PPC
29	`0x58030000`	`0x5803FFFF`	64KB	Reserved	Reserved. When accessed, results in bus error.		

1. Use the following list for the meaning of the abbreviations in this column:
   - 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 6-47: System Information register map

Offset	Name	Access	Reset value	Description
`0x000`	SOC_IDENTITY	Read-only	Configurable	SoC Identity Register. See SOC_IDENTITY.
`0x004`	SYS_CONFIG0	Read-only	Configurable	System Hardware Configuration 0 Register. See SYS_CONFIG0 and SYS_CONFIG1.
`0x008`	SYS_CONFIG1	Read-only	Configurable	System Hardware Configuration 1 Register. See SYS_CONFIG0 and SYS_CONFIG1.
`0x010` - `0xFC4`	Reserved		`0x00000000`	Reserved.
`0xFC8`	IIDR	Read-only	Configurable	Subsystem Implementation Identity Register. See IIDR.
`0xFCC`	Reserved		`0x00000000`	Reserved.
`0xFD0`	PIDR4	Read-only	`0x00000004`	Peripheral ID 4.
`0xFD4` - `0xFDC`	Reserved		`0x00000000`	Reserved.
`0xFE0`	PIDR0	Read-only	`0x00000058`	Peripheral ID 0.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
`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.

6.6.1.1 SOC_IDENTITY

The System-On-Chip (SoC) Identity register provide 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 6-48: SOC_IDENTITY Register

Bits	Type	Default	Name	Description
31:20	RO	SOCPRTID	SOC_PRODUCT_ID	Configurable value identifying the SoC.
19:16	RO	SOCVAR	SOC_VARIANT	Configurable value indicating major revision of the SoC.
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: • [11:8] JEP106 continuation code of implementer • [7] Always 0 • [6:0] JEP106 identity code of implementer

When EXPLOGIC_PRESENT = 1, the SoC identity register fields are used to 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*.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.6.1.2 SYS_CONFIG0 and SYS_CONFIG1

The System Hardware Configuration registers provides several registers to allow software to find out about the configuration of the SSE-300 based system.

Note

In these tables, the fields CPU0_TCM_BANK_NUM and CPU0_HAS_SYSTCM refers to TCM that are implemented on the system interconnect close to each associated core, rather than the TCMs that are implemented within the processor core. SSE-300 does not support TCM being implemented on the system interconnect.

Table 6-49: SYS_CONFIG0 register

Bits	Type	Default	Name	Description
31:28	RO	`0x0`	CPU1_TCM_BANK_NUM	The VM Bank TCM memory for CPU 1. This field is **RAZ/WI**.
27	RO	`0x0`	CPU1_HAS_SYSTCM	CPU 1 has System TCM. • 0 = No • 0 = Yes Note that this is not the CPU's local ITCM or DTCM, but instead are TCMs implemented at system level.
26:24	RO	CPU1TYPE	CPU1_TYPE	CPU 1 Core Type: • 000 - Does not exist • 011 - Cortex-M55 processor • Others - Reserved
23:20	RO	`0x0`	CPU0_TCM_BANK_NUM	The VM Bank that is the TCM memory for CPU 0.
19	RO	`0x0`	CPU0_HAS_SYSTCM	CPU 0 has System TCM. • 0 = No • 0 = Yes Note that this is not the CPU's local ITCM or DTCM, but instead are TCMs implemented at system level.
18:16	RO	CPU0TYPE	CPU0_TYPE	CPU 0 Core Type: • 000 - Does not exist • 011 - Cortex-M55 processor • Others - Reserved
15:13	RO	`0x0`	Reserved	Reserved
12:11	RO	PILEVEL	PI_LEVEL	Power Infrastructure Level: • 00 - Basic Level • 01 - Intermediate Level • 10 - Advance Level • Others - Reserved
10	RO	HASCSS	HAS_CSS	It reflects whether the CoreSight SoC-600-based common debug infrastructure is included. • 0 = No • 0 = Yes

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
9	RO	HASCRYPTO	HAS_CRYPTO	It reflects whether CryptoCell-312 is included. 0 = No 0 = Yes
8:4	RO	VMADDRWIDTH	VM_ADDR_WIDTH	Volatile Memory Bank Address Width, where the size of each bank is equal to $2^{VM\_ADDR\_WIDTH}$ bytes.
3:0	RO	NUMVMBANK	NUM_VM_BANK	Number of Volatile Memory Banks.

Table 6-50: SYS_CONFIG1 register

Bits	Type	Default	Name	Description
31:16	RO	`0x0000` -	Reserved	Reserved
15:12	RO	`0x0`	CPU3_TCM_BANK_NUM	The VM Bank that is the TCM memory for CPU 3.
11	RO	`0x0`	CPU3_HAS_SYSTCM	CPU 3 has System TCM. • 0 = No • 0 = Yes Note that this is not the CPU's local ITCM or DTCM, but instead are TCMs implemented at system level.
10:8	RO	CPU3TYPE	CPU3_TYPE	CPU 3 Core Type: • 000 - Does not exist • 011 - Cortex-M55 processor • Others - Reserved
7:4	RO	`0x0`	CPU2_TCM_BANK_NUM	The VM Bank that is the TCM memory for CPU 2. This field is **RAZ/WI**.
3	RO	`0x0`	CPU2_HAS_SYSTCM	CPU 2 has System TCM. • 0 = No • 0 = Yes Note that this is not the CPU's local ITCM or DTCM, but instead are TCMs implemented at system level.
2:0	RO	CPU2TYPE	CPU2_TYPE	CPU 2 Core Type: • 000 - Does not exist • 011 - Cortex-M55 processor • Others - Reserved

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 6-51: IIDR Register

Bits	Type	Default	Name	Description
31:20	RO	IMPLPRTID	IMP_PRODUCT_ID	Configurable value identifying the subsystem implementation.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
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: • [11:8] JEP106 continuation code of implementer. • [7] Always 0. • [6:0] JEP106 identity code of implementer.

When EXPLOGIC_PRESENT = 1, the subsystem implementation identity register fields are used to 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 will 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 6-52: System Control register map

Offset	Name	Access	Reset value	Description
`0x000`	SECDBGSTAT	Read-only	CFG_DEF	Secure Debug Configuration Status Register See SECDBGSTAT, SECDBGSET and SECDBGCLR
`0x004`	SECDBGSET	Read-write	`0x00000000`	Secure Debug Configuration Set Register See SECDBGSTAT, SECDBGSET and SECDBGCLR
`0x008`	SECDBGCLR	Write-only	`0x00000000`	Secure Debug Configuration Clear Register See SECDBGSTAT, SECDBGSET and SECDBGCLR
`0x00C`	SCSECCTRL	Read-write	`0x00000000`	System Control Security Controls Register See SCSECCTRL

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
`0x010`	CLK_CFG0	Read-write	CFG_DEF	Clock Configuration Register 0 See CLK_CFG0 and CLK_CFG1
`0x014`	CLK_CFG1	Read-write	CFG_DEF	Clock Configuration Register 1 See CLK_CFG0 and CLK_CFG1
`0x018`	CLOCK_FORCE	Read-write	CFG_DEF	Clock forces See CLOCK_FORCE
`0x01C` - `0x0FF`	Reserved	Read-only	`0x00000000`	Reserved
`0x100`	RESET_SYNDROME	Read-write	`0x00000001`	Reset syndrome See RESET_SYNDROME
`0x104`	RESET_MASK	Read-write	CFG_DEF	Reset mask See RESET_MASK
`0x108`	SWRESET	Write-only	`0x00000000`	Software reset See SWRESET
`0x10C`	GRETREG	Read-write	`0x00000000`	General Purpose Retention Register See GRETREG
`0x110`	INITSVTOR0	Read-write	CFG_DEF	CPU 0 Initial Secure Reset Vector Register See INITSVTOR0
`0x114`	Reserved	Read-only	`0x00000000`	CPU 1 Initial Secure Reset Vector Register. Reserved for INITSVTOR1. See INITSVTOR0
`0x118`	Reserved	Read-only	`0x00000000`	CPU 2 Initial Secure Reset Vector Register. Reserved for INITSVTOR2. See INITSVTOR0
`0x11C`	Reserved	Read-only	`0x00000000`	CPU 3 Initial Secure Reset Vector Register. Reserved for INITSVTOR3. See INITSVTOR0
`0x120`	CPUWAIT	Read-write	CFG_DEF	CPU Boot Wait Control See CPUWAIT
`0x124`	NMI_ENABLE	Read-write	CFG_DEF	Enabling and Disabling Non Maskable Interrupts See NMI_ENABLE
`0x128` - `0x1F8`	Reserved	Read-only	`0x00000000`	Reserved
`0x1FC`	PWRCTRL	Read-write	`0x00000003`	Power Configuration and Control See PWRCTRL

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
0x200	PDCM_PD_SYS_SENSE	Read-write	CFG_DEF	PDCM PD_SYS sensitivity See PDCM_PD_SYS_SENSE
0x204	PDCM_PD_CPU0_SENSE	Read-only	0x00000000	PDCM PD_CPU0 sensitivity See PDCM_PD_CPU0_SENSE
0x208	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_CPU1_SENSE. PDCM PD_CPU1 sensitivity
0x20C	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_CPU2_SENSE. PDCM PD_CPU2 sensitivity
0x210	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_CPU3_SENSE. PDCM PD_CPU3 sensitivity
0x214	PDCM_PD_VMR0_SENSE[1]	Read-write	0x40000000	PDCM PD_VMR0 sensitivity See PDCM_PD_VMR<M>_SENSE
0x218	PDCM_PD_VMR1_SENSE[2]	Read-write	0x40000000	PDCM PD_VMR1 sensitivity See PDCM_PD_VMR<M>_SENSE
0x21C	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_VMR2_SENSE. PDCM PD_VMR2 sensitivity See PDCM_PD_VMR<M>_SENSE
0x220	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_VMR3_SENSE. PDCM PD_VMR3 sensitivity See PDCM_PD_VMR<M>_SENSE
0x224 - 0x248	Reserved	Read-only	0x00000000	Reserved
0x24C	Reserved	Read-only	0x00000000	Reserved for PDCM_PD_MGMT_SENSE. PDCM PD_MGMT sensitivity See PDCM_PD_MGMT_SENSE
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	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

1. These registers do not exist and are reserved if NUMVMBANK < 1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2. These registers do not exist and are reserved if NUMVMBANK < 2

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**, **SYSDSSACCENX** and SYSDSSACCEN<N> where N is 0 to NUMCPU. For each signal and just one for all SYSDSSACCEN<N> and SYSDSSACCENX, a selector is provided 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. 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.

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 is used to define the system wide **DBGEN** value
- If DBGEN_SEL is HIGH the internal register value DBGEN_I is used to define the system wide **DBGEN** value

To set the DBGEN_I or DBGEN_SEL values to HIGH, write to the SECDBGSET register with DBGEN_I_SET or DBGEN_SEL_SET set to HIGH respectively.

To set DBGEN_I or DBGEN_SEL values to LOW, write to the SECDBGCLR register with DBGEN_I_CLR or DBGEN_SEL_CLR set to HIGH respectively.

To read the output value of **DBGEN**, read the SECDBGSTAT register for the DBGEN_STATUS field.

To read the internal register value DBGEN_I, read SECDBGSET for the DBGEN_I_SET field.

To read the selector value DBGEN_SEL, read the SECDBGSET for the DBGEN_SEL_SET field.

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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- DAPACCENSELDIS for disabling DAPACCEN_SEL
- DAPDSSACCENSELDIS for disabling DAPDSSACCEN_SEL
- SYSDSSACCENSELDIS for disabling SYSDSSACCEN_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, especially when CryptoCell exists (HASCRYPTO = '1') in the system and the intention is to use signals derived from CRYPTODCUEN to control debug instead.

These registers are reset by **nCOLDRESETAON**.

These registers reside in the PD_AON power domain.

Table 6-53: SECDBGSTAT Register

Bits	Type	Default	Name	Description
31	RO	`0x00`	-	Reserved.
30	RO	`0x00`	-	Reserved for SYSDSSACCE NSELDIS_STATUS that reports SY SDSSACCENSELDIS configuration value when read.
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	RO	`0x0000`	-	Reserved.
17	RO	`0x00`	-	Reserved for SYSDSSACCEN<n>_SEL_STATUS, Active High System Mapped Debug Access Enable Selector Value.
16	RO	`0x00`	-	Reserved for SYSDSSACCENX_STATUS, Active High System Mapped Debug Access for Implementation Defined Master(s).
15	RO	`0x00`	-	Reserved for SYSDSSACCEN3_STATUS, Active High System Mapped Debug Access for CPU 3 Enable Value.
14	RO	`0x00`	-	Reserved for SYSDSSACCEN2_STATUS, Active High System Mapped Debug Access for CPU 2 Enable Value.
13	RO	`0x00`	-	Reserved for SYSDSSACCEN1_STATUS, Active High System Mapped Debug Access for CPU 1 Enable Value.
12	RO	`0x00`	-	Reserved for SYSDSSACCEN0_STATUS, Active High System Mapped Debug Access for CPU 0 Enable Value.
11	RO	`0x00`	DAPDSSACCEN_SEL_STATUS	Active High DAP to Debug Subsystem Access Enable Selector Value. This bit returns the D APDSSACCEN_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 DAPDSSACCEN 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
8	RO	DAPACCENIN	DAPACCEN_STATUS	Active High DAP Access Enable Value. This bit reflects the value on the DAPACCEN pin.
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 SPNIDEN 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 SPIDEN 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 NIDEN 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 DBGEN pin.

Table 6-54: SECDBGSET Register

Bits	Type	Default	Name	Description
31:18	RO	0x0000	-	Reserved
17	RO	0x00	-	Reserved for SYSDSSACCEN_SEL_SET, Active High System Mapped Debug Access Enable Selector Value Set Register
16	RO	0x00	-	Reserved for SYSDSSACCENX_I_SET, Internal Version Active High System Mapped Debug Access for Implementation Defined Master(s) Enable Set Control
15	RO	0x00	-	Reserved for SYSDSSACCEN3_I_SET, Internal Version Active High System Mapped Debug Access for CPU 3 Enable Set Register
14	RO	0x00	-	Reserved for SYSDSSACCEN2_I_SET, Internal Version Active High System Mapped Debug Access for CPU 2 Enable Set Register
13	RO	0x00	-	Reserved for SYSDSSACCEN1_I _SET, Internal Version Active High System Mapped Debug Access for CPU 3 Enable Set Register
12	RO	0x00	-	Reserved for SYSDSSACCEN0_I_SET, Active High System Mapped Debug Access for CPU 3 Enable Set Register
11	W1T	0x00	DAPDSSACCEN_SEL_SET	Set Active High DAP to Debug Subsystem Access Enable Selector. Write HIGH to set DAPDSSACCEN_SEL. **RAZ/WI** if DAPDSSACCENSELDIS = 1.
10	W1S	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. **RAZ/WI** if DAPDSSACCENSELDIS = 1.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
9	W1T	0x00	DAPACCEN_SEL_SET	Set Active High DAP Access Enable Selector. Write HIGH to set DAPACCEN_SEL. **RAZ/WI** if DAPACCENSELDIS = 1.
8	W1S	0x00	DAPACCEN_I_SET	Set internal version of Active High DAP Access Enable. Write HIGH to set DAPACCEN_I. When read returns DAPACCEN_I. **RAZ/WI** if DAPACCENSELDIS = 1.
7	W1T	0x00	SPNIDEN_SEL_SET	Set Active High Secure Privilege Non-Invasive Debug Enable Selector. Write HIGH to set SPNIDEN_SEL. **RAZ/WI** if SPNIDENSELDIS = 1.
6	W1S	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. **RAZ/WI** if SPNIDENSELDIS = 1.
5	W1T	0x00	SPIDEN_SEL_SET	Set Active High Secure Privilege Invasive Debug Enable Selector. Write HIGH to set SPIDEN_SEL. **RAZ/WI** if SPIDENSELDIS = 1.
4	W1S	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. **RAZ/WI** if SPIDENSELDIS = 1.
3	W1T	0x00	NIDEN_SEL_SET	Set Active High Non-Invasive Debug Enable Selector. Write HIGH to set NIDEN_SEL. **RAZ/WI** if NIDENSELDIS = 1.
2	W1S	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. **RAZ/WI** if NIDENSELDIS = 1.
1	W1T	0x00	DBGEN_SEL_SET	Set Active High Debug Enable Selector. Write HIGH to set DBGEN_SEL. **RAZ/WI** if DBGENSELDIS = 1.
0	W1S	0x00	DBGEN_I_SET	Set internal version of Active High Debug Enable. Write HIGH to set DBGEN_I. When read returns DBGEN_I. **RAZ/WI** if DBGENSELDIS = 1.

Table 6-55: SECDBGCLR Register

Bits	Type	Default	Name	Description
31:18	RO	0x0000	-	Reserved.
17	RO	0x00	-	Reserved for SYSDSSACCEN_SEL_CLR, Active High System Mapped Debug Access Enable Selector Value Clear Register.
16	RO	0x00	-	Reserved for SYSDSSACCENX_I_CLR, Internal Version Active High System Mapped Debug Access for for Implementation Defined Master(s) Enable Clear Control.
15	RO	0x00	-	Reserved for SYSDSSACCEN3_I_CLR, Internal Version Active High System Mapped Debug Access for CPU 3 Enable Clear Register.
14	RO	0x00	-	Reserved for SYSDSSACCEN2_I_CLR, Internal Version Active High System Mapped Debug Access for CPU 2 Enable Clear Register.
13	RO	0x00	-	Reserved for SYSDSSACCEN1_I_CLR, Internal Version Active High System Mapped Debug Access for CPU 3 Enable Clear Register.
12	RO	0x00	-	Reserved for SYSDSSACCEN0_I_CLR, Internal Version Active High System Mapped Debug Access for CPU 3 Enable Clear Register.
11	W1T	0x00	DAPDSSACCEN_SEL_CLR	Clears Active High DAP to Debug Subsystem Access Enable Selector. Write HIGH to clear DAPDSSACCEN_SEL. Always RAZ. WI if DAPDSSACCENSELDIS = 1.
10	W1T	0x00	DAPDSSACCEN_I_CLR	Clears internal version of Active High DAP to Debug Subsystem Access Enable. Write HIGH to clear DAPDSSACCEN_I. Always RAZ. WI if DAPDSSACCENSELDIS = 1.
9	W1T	0x00	DAPACCEN_SEL_CLR	Clears Active High DAP Access Enable Selector. Write HIGH to clear DAPACCEN_SEL. Always RAZ. WI if DAPACCENSELDIS = 1.
8	W1T	0x00	DAPACCEN_I_CLR	Clears internal version of Active High DAP Access Enable. Write HIGH to clear DAPACCEN_I. Always RAZ. WI if DAPACCENSELDIS = 1.
7	W1T	0x00	SPNIDEN_SEL_CLR	Clears Active High Secure Privilege Non-Invasive Debug Enable Selector. Write HIGH to clear SPNIDEN_SEL. Always RAZ. WI if SPNIDENSELDIS = 1.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Type	Default	Name	Description
6	W1T	0x00	SPNIDEN_I_CLR	Clears internal version of Active High Secure Privilege Non-Invasive Debug Enable. Write HIGH to clear SPNIDEN_I. Always RAZ. WI if SPNIDENSELDIS = 1.
5	W1T	0x00	SPIDEN_SEL_CLR	Clears Active High Secure Privilege Invasive Debug Enable Selector. Write HIGH to clear SPIDEN_SEL. Always RAZ. WI if SPIDENSELDIS = 1.
4	W1T	0x00	SPIDEN_I_CLR	Clears internal version of Active High Secure Privilege Invasive Debug Enable. Write HIGH to clear SPIDEN_I. Always RAZ. WI if SPIDENSELDIS = 1.
3	W1T	0x00	NIDEN_SEL_CLR	Clears Active High Non-Invasive Debug Enable Selector. Write HIGH to clear NIDEN_SEL. Always RAZ. WI if NIDENSELDIS = 1.
2	W1T	0x00	NIDEN_I_CLR	Clears internal version of Active High Non-Invasive Debug Enable. Write HIGH to clear NIDEN_I. Always RAZ. WI if NIDENSELDIS = 1.
1	W1T	0x00	DBGEN_SEL_CLR	Clears Active High Debug Enable Selector. Write HIGH to clear DBGEN_SEL. Always RAZ. WI if DBGENSELDIS = 1.
0	W1T	0x00	DBGEN_I_CLR	Clears internal version of Active High Debug Enable. Write HIGH to clear DBGEN_I. Always RAZ. WI if DBGENSELDIS = 1.

6.6.2.2 SCSECCTRL

The System Control Security Controls provide register bits 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 6-56: SCSECCTRL Register

Bits	Type	Default	Name	Description
31:3	RO	0x00000000	-	Reserved
2	W1S	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	RO	0x00	-	Reserved

6.6.2.3 CLK_CFG0 and CLK_CFG1

The CLK_CFG0 and CLK_CFG1 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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.

Since SYSCLK and CPU0CLK must have the same frequency, the **CPU0CLKCFG** output must not be used in the expansion logic, and the **CPU0CLKCFGSTATUS** input must be tied to LOW. The register fields CLK_CFG0.CPU0CLKCFG and CLK_CFG0.CPU0CLKCFGSTATUS should not be used either.

These registers are reset by **nCOLDRESETAON**.

These registers reside in the PD_AON power domain.

Table 6-57: CLK_CFG0 Register

Bits	Type	Default	Name	Description
31:28	RO	0x00	-	Reserved for CPU3CLKCFGSTATUS. Clock Configuration Status value that reports the status of clock control for CPU3CLK.
27:24	RO	0x00	-	Reserved for CPU2CLKCFGSTATUS. Clock Configuration Status value that reports the status of clock control for CPU2CLK.
23:20	RO	0x00	-	Reserved for CPU1CLKCFGSTATUS. Clock Configuration Status value that reports the status of clock control for CPU1CLK.
19:16	RO	0x00	CPU0CLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for CPU0CLK.
15:12	RO	0x00	-	Reserved for CPU3CLKCFG. Clock Configuration value that drives CPU3CLKCFG signals.
11:8	RO	0x00	-	Reserved for CPU2CLKCFG. Clock Configuration value that drives CPU2CLKCFG signals.
7:4	RO	0x00	-	Reserved for CPU1CLKCFG. Clock Configuration value that drives CPU1CLKCFG signals.
3:0	RW	0x00	CPU0CLKCFG	Clock Configuration value that drives CPU0CLKCFG signals.

Table 6-58: CLK_CFG1 Register

Bits	Type	Default	Name	Description
31:24	RO	0x000	-	Reserved
23:20	RO	0x00	AONCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for AONCLK.
19:16	RO	0x00	SYSCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for SYSCLK.
15:8	RO	0x000	-	Reserved
7:4	RW	AONCLKCFGRST	AONCLKCFG	Clock Configuration value that drives AONCLKCFG signals.
3:0	RW	SYSCLKCFGRST	SYSCLKCFG	Clock Configuration value that drives SYSCLKCFG signals.

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 7 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

clock ON can reduce the latency that is incurred as 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 22 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 turn on time.

These registers are reset by **nCOLDRESETAON**.

This register resides in the PD_AON power domain.

Table 6-59: CLOCK_FORCE Register

Bits	Type	Default	Name	Description
31:23	RO	`0x0000`	-	Reserved
22	RW	`0x0`	-	Reserved for CPU3CLK_FORCE.
21	RW	`0x0`	-	Reserved for CPU2CLK_FORCE.
20	RW	`0x0`	-	Reserved for CPU1CLK_FORCE.
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.
15:8	RO	`0x00000000`	-	Reserved.
7	RO	`0x00`	-	Reserved for CPU3_CLKFORCE.
6	RO	`0x00`	-	Reserved for CPU2_CLKFORCE.
5	RO	`0x00`	-	Reserved for CPU1_CLKFORCE.
4	RW	`0x01`	CPU0_CLKFORCE	Set HIGH to force all clocks in PD_CPU0 to run.
3	RO	`0x00`	-	Reserved for CRYPTO_CLKFORCE.
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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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 will be stored in the register.

This register resides in the PD_AON power domain.

Table 6-60: RESET_SYNDROME Register

Bits	Type	Default	Name	Description
31:16	RO	0x00000	-	Reserved
15	RO	0x00	-	Reserved for CPU3LOCKUP
14	RO	0x00	-	Reserved for CPU2LOCKUP
13	RO	0x00	-	Reserved for CPU1LOCKUP
12	W0C	0x00	CPU0LOCKUP	CPU 0 Lockup Status
11	RO	0x00	-	Reserved for CPU3RSTREQ
10	RO	0x00	-	Reserved for CPU2RSTREQ
9	RO	0x00	-	Reserved for CPU1RSTREQ
8	W0C	0x00	CPU0RSTREQ	CPU 0 Warm Reset Request
7	W0C	0x00	HOSTRESETREQ	Host Level Cold Reset Request Input
6	RO	0x00	-	Reserved for CRYPTORSTREQ, CryptoCell Warm Reset Request
5	W0C	0x00	SWRESETREQ	Software Cold Reset Request
4	W0C	0x00	RESETREQ	Subsystem Hardware Cold Reset Request Input
3	W0C	0x00	SLOWCLKWDRSTREQ	SLOWCLK Watchdog Cold Reset Request
2	W0C	0x00	SWDRSTREQ	Secure Watchdog Cold Reset Request
1	W0C	0x00	NSWDRSTREQ	Non-secure Watchdog Cold Reset Request
0	W0C	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 being used to generate the reset, and also prevents the associated RESET_SYNDROME register bit from recording the event.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

This register resides in the PD_AON power domain.

This register is reset by the **nWARMRESETAON**.

Table 6-61: RESET_MASK Register

Bits	Type	Default	Name	Description
31:12	RO	0x000000	-	Reserved
11	RO	0x00	-	Reserved for CPU3RSTREQENRST, CPU 3 Warm Reset Request Enable.
10	RO	0x00	-	Reserved for CPU2RSTREQENRST, CPU 2 Warm Reset Request Enable.
9	RO	0x00	-	Reserved for CPU1RSTREQENRST, CPU 1 Warm Reset Request Enable.
8	RW	CPU0RSTREQENRST	CPU0RSTREQEN	CPU 0 Warm Reset Request Enable
7:2	RO	0x000	-	Reserved
1	RW	0x00	NSWDRSTREQEN	Non-secure Watchdog Reset Enable
0	RO	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 6-62: SWRESET Register

Bits	Type	Default	Name	Description
31:2	RO	0x00000000	-	Reserved.
9	W1T	0x00	SWRESETREQ	Software Reset Request. Write '1' to request a Cold Reset.
8:0	RO	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 6-63: GRETREG Register

Bits	Type	Default	Name	Description
31:16	RO	0x00000	-	Reserved
15:0	RW	0x00000	GRETREG	General Purpose Retention Register

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.6.2.9 INITSVTOR0

This register is used to define the CPU 0 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 6-64: INITSVTOR0 Register

Bits	Type	Default	Name	Description
31:7	RW	INITSVTOR0RST[31:7]	INITSVTOR0	Default Secure Vector table offset at reset for CPU 0.
6:1	RO	`0x000`	-	Reserved
0	W1S	`0x00`	INITSVTOR0LOCK	Lock INITSVTOR0. When set to '1', will stop any further writes to INITSVTOR0 and INITSVTOR0LOCK fields. Cleared only by warm reset.

6.6.2.10 CPUWAIT

This Register provides controls to force each CPU 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 6-65: CPUWAIT Register

Bits	Type	Default	Name	Description
31:4	RO	`0x00000000`	-	Reserved
3	RO	`0x00`	-	Reserved for CPU3WAIT.
2	RO	`0x00`	-	Reserved for CPU2WAIT.
1	RO	`0x00`	-	Reserved for CPU1WAIT.
0	RW	CPU0WAITRST	CPU0WAIT	CPU 0 waits at boot. • '0': boot normally. • '1': wait at boot.

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 each CPU core. This allows a CPU to take control of all internal NMI interrupt sources or allow all CPUs to see the same NMI interrupts. 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-66: NMI_ENABLE Register

Bits	Type	Default	Name	Description
31:20	RO	0x0000	-	Reserved.
19	RO	0x00	-	Reserved for CPU3_EXPNMI_ENABLE
18	RO	0x00	-	Reserved for CPU2_EXPNMI_ENABLE
17	RO	0x00	-	Reserved for CPU1_EXPNMI_ENABLE
16	RW	CPU0EXPNMIENABLERST	CPU0_EXPNMI_ENABLE	CPU 0 Externally Sourced NMI Enable. This determines if the input, **CPU0EXPNMI**, can raise NMI interrupt on CPU 0: • HIGH, allowed. • LOW, is masked and not allowed.
15:4	RO	0x0000	-	Reserved.
3	RO	0x00	-	Reserved for CPU3_INTNMI_ENABLE
2	RO	0x00	-	Reserved for CPU2_INTNMI_ENABLE
1	RO	0x00	-	Reserved for CPU1_INTNMI_ENABLE
0	RW	CPU0INTNMIENABLERST	CPU0_INTNMI_ENABLE	CPU 0 Internally Sourced NMI Enable. This determines if the subsystem internally generated NMI interrupt sources can raise NMI interrupt on CPU 0: • HIGH, allowed. • LOW, is masked and not allowed.

6.6.2.12 PWRCTRL

The Power Control register configures the power control features in SSE-300 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	RO	0x00000000	-	Reserved.
1	W0C	0x01	PPU_ACCESS_UNLOCK	PPU_ACCESS_FILTER write unlock. When set to '1', both PPU_ACCESS_FILTER and this register bits can be written. When set to '0', the PPU_ACCESS_FILTER and this register bit is no longer writable, and PPU_ACCESS_UNLOCK stays '0'.
0	RW	0x01	PPU_ACCESS_FILTER	Filter Access to PPU Registers. When set to '1', only key PPU interrupt handling registers are open to write access, and all other PPU registers are read only. When set to '0', it releases all PPU register to full access. For more information in PPU registers accessibility, see Power Policy Units.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.6.2.13 PDCM_PD_SYS_SENSE

The Power Dependency Control Matrix System power domain (PD_SYS) Sensitivity register is used to define 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: • 00 - Minimum power state is OFF • 01 - Minimum power state is Retention • 10 - Minimum power state is ON • Others - Reserved
29:24	RO	`0x00`	-	Reserved
23	RO	`0x00`	-	Reserved for **PDCMQREQn[7]**
22	RO	`0x00`	-	Reserved for **PDCMQREQn[6]**
21	RO	`0x00`	-	Reserved for **PDCMQREQn[5]**
20	RO	`0x00`	-	Reserved for **PDCMQREQn[4]**
19	RW	`0x00`	S_PDCMQREQ3	Enables sensitivity to **PDCMQREQn[3]** signal. If set to '1', PD_SYS stays ON if **PDCMQREQn[3]** signal is HIGH.
18	RW	`0x00`	S_PDCMQREQ2	Enables sensitivity to **PDCMQREQn[2]** signal. If set to '1', PD_SYS stays ON if **PDCMQREQn[2]** signal is HIGH.
17	RW	`0x00`	S_PDCMQREQ1	Enables sensitivity to **PDCMQREQn[1]** signal. If set to '1', PD_SYS stays ON if **PDCMQREQn[1]** signal is HIGH.
16	RW	`0x00`	S_PDCMQREQ0	Enables sensitivity to **PDCMQREQn[0]** signal. If set to '1', PD_SYS stays ON if **PDCMQREQn[0]** signal is HIGH.
15	RO	`0x00`	-	Reserved for S_PD_MGMT_ON.
14	RO	`0x00`	-	Reserved
13	RO	`0x00`	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	RO	`0x00`	-	Reserved for S_PD_CRYPTO_ON
11:5	RO	`0x000`	-	Reserved.
4	RO	`0x00`	-	Reserved for S_PD_CPU3_ON
3	RO	`0x00`	-	Reserved for S_PD_CPU2_ON
2	RO	`0x00`	-	Reserved for S_PD_CPU1_ON
1	RO	`0x01`	S_PD_CPU0_ON	Enable PD_CPU0 sensitivity. If set to 1 PD_SYS will stay on if PD_CPU0 is on.
0	RW	`0x00`	S_PD_SYS_ON	Tied to LOW. Ignores PD_SYS power state.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6.6.2.14 PDCM_PD_CPU0_SENSE

The Power Dependency Control Matrix System Power domain (PD_CPU0) Sensitivity register is used to define what keeps awake the PD_CPU0 domain and the minimum power state to use when the domain is in its low power state. This register is reset by **nWARMRESETAON**.

Currently the 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	RO	0x000	-	Reserved.
23	RO	0x00	-	Reserved for S_PDCMQREQ7.
22	RO	0x00	-	Reserved for S_PDCMQREQ6.
21	RO	0x00	-	Reserved for S_PDCMQREQ5.
20	RO	0x00	-	Reserved for S_PDCMQREQ4.
19	RO	0x00	S_PDCMQREQ3	Tied to LOW. Ignores **PDCMQREQn[3]**.
18	RO	0x00	S_PDCMQREQ2	Tied to LOW. Ignores **PDCMQREQn[2]**.
17	RO	0x00	S_PDCMQREQ1	Tied to LOW. Ignores **PDCMQREQn[1]**.
16	RO	0x00	S_PDCMQREQ0	Tied to LOW. Ignores **PDCMQREQn[0]**.
15	RO	0x00	-	Reserved for S_PD_MGMT_ON.
14	RO	0x00	-	Reserved
13	RO	0x00	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	RO	0x00	-	Reserved for S_PD_CRYPTO_ON.
11:5	RO	0x000	-	Reserved
4	RO	0x00	-	Reserved for S_PD_CPU3_ON.
3	RO	0x00	-	Reserved for S_PD_CPU2_ON.
2	RO	0x00	-	Reserved for S_PD_CPU1_ON.
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.

6.6.2.15 PDCM_PD_VMR<M>_SENSE

The Power Dependency Control Matrix Volatile Memory Region <*M*> power domain (PD_VMR<*M*>) sensitivity register is used to define 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 6-70: PDCM_PD_VMR<N>_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: • 00 - Minimum power state is OFF • 01 - Minimum power state is Retention • 10 - Minimum power state is ON. • Others - Reserved
29:24	RO	0x00	-	Reserved
23	RO	0x00	-	Reserved for **PDCMQREQn[7]**
22	RO	0x00	-	Reserved for **PDCMQREQn[6]**
21	RO	0x00	-	Reserved for **PDCMQREQn[5]**
20	RO	0x00	-	Reserved for **PDCMQREQn[4]**
19	RW	0x00	S_PDCMQREQ3	Enables sensitivity to **PDCMQREQn[3]** signal. If set to '1', PD_VMR<M> stays ON if **PDCMQREQn[3]** signal is HIGH.
18	RW	0x00	S_PDCMQREQ2	Enables sensitivity to **PDCMQREQn[2]** signal. If set to '1', PD_VMR<M> stays ON if **PDCMQREQn[2]** signal is HIGH.
17	RW	0x00	S_PDCMQREQ1	Enables sensitivity to **PDCMQREQn[1]** signal. If set to '1', PD_VMR<M> stays ON if **PDCMQREQn[1]** signal is HIGH.
16	RW	0x00	S_PDCMQREQ0	Enables sensitivity to **PDCMQREQn[0]** signal. If set to '1', PD_VMR<M> stays ON if **PDCMQREQn[0]** signal is HIGH.
15	RO	0x00	-	Reserved for S_PD_MGMT_ON
14	RO	0x00	-	Reserved
13	RO	0x00	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.
12	RO	0x00	-	Reserved for S_PD_CRYPTO_ON
11:5	RO	0x000	-	Reserved
4	RO	0x00	-	Reserved for S_PD_CPU3_ON.
3	RO	0x00	-	Reserved for S_PD_CPU2_ON
2	RO	0x00	-	Reserved for S_PD_CPU1_ON
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.

6.6.2.16 PDCM_PD_MGMT_SENSE

The Power Dependency Control Matrix PD_MGMT Power Domain Sensitivity register is used to define what keeps awake the PD_MGMT domains. This register does not exist and the register area it occupies is Reserved and **RAZ/WI**.

6.7 CPU Private Peripheral Bus region

As defined by the ARMv8-M architecture specification, each processor hosts a local Private Peripheral Bus (PPB) region at address `0xE0000000` to `0xE00FFFFF`. This region is for integration

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

with the CoreSight debug and trace components that are normally local to each CPU and is not intended for general peripheral usage.

In SSE-300, this region has the memory map as shown in the table below.

Note

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®-M55 Technical Reference Manual*.

Depending on HASCSS and EXPLOGIC_PRESENT:

- If HASCSS = 0 and EXPLOGIC_PRESENT = 1, the Cortex-M55 TPIU 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 **RAZ/WI**. The MCU debug ROM table is pointing to the TPIU 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`.
- When HASCSS = 0 and EXPLOGIC_PRESENT = 0, the TPIU, ETB, and MCU debug ROM table are not integrated and the these regions are provided as a PPB expansion

See HASCSS and DEBUGLEVEL for supported values.

Table 6-71: CPU0 Private Peripheral Bus Region

Row ID	Address from	Address to	Size	Region Name	Description
1	`0xE0040000`	`0xE0040FFF`	4KB	Cortex-M55 TPIU/ PPB Expansion	Cortex-M55 TPIU when EXPLOGIC_PRESENT = 1 CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
2	`0xE0041000`	`0xE0041FFF`	4KB	ETM[1]	Embedded Trace Module
3	`0xE0042000`	`0xE0042FFF`	4KB	CTI[1]	Cross Trigger Interface
4	`0xE0043000`	`0xE0044FFF`	8KB	Reserved	Reserved
5	`0xE0045000`	`0xE0045FFF`	4KB	ETB/PPB Expansion	Reserved for ETB (**RAZ/WI**) when EXPLOGIC_PRESENT = 1 CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
6	`0xE0046000`	`0xE0046FFF`	4KB	PMC[1]	Programmable MBIST Controller
7	`0xE0047000`	`0xE0047FFF`	4KB	EWIC	External Wakeup Interrupt Controller. For more information, see EWIC.
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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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-300 is designed to always support an External Wakeup Interrupt Controller (EWIC) for each core in the system. This allows the system to support each processor being switched off independently, and for the EWIC to run at a lower clock rate to help reduce PD_AON dynamic and leakage power consumption.

All EWICs reside in the PD_AON power domain, run on AONCLK, and reside in the nWARMRESETAON reset domain.

Each EWIC <N> is only accessible through the Private Peripheral Bus Region that is associated with CPU <N> at address `0xE0047000` to `0xE0047FFF`, where N is 0 to NUMCPU. The register map of the EWIC is as follows:

Table 6-72: EWIC register map

Offset	Name	Access	Reset value	Description
`0x000`	EWIC_CR	Read-write	`0x00000000`	EWIC Control Register. For more information, see EWIC_CR.
`0x004`	EWIC _ASCR	Read-write	`0x00000003`	Automatic Sequence Control Register. For more information, see EWIC_ASCR.
`0x008`	EWIC_CLRMASK	Write-only	`0x00000000`	Clear All Mask Register. For more information, see EWIC_CLRMASK.
`0x00C`	EWIC_NUMID	Read-only	CPU0EXPNUMIRQ +35	ID Register for the number of events supported. For more information, see EWIC_NUMID.
`0x010` - `0x1FC`	Reserved	Read-only	`0x00000000`	Reserved
`0x200`	EWIC_MASKA	Read-write	`0x00000000`	Set which internal events cause wakeup. For more information, see EWIC_MASKA and EWIC_MASK<N>.
`0x204` -	EWIC_MASK<N>	Read-write	`0x00000000`	Set which interrupts cause wakeup. For more information, see EWIC_MASKA and EWIC_MASK<N>.
`0x240` - `0x3FC`	Reserved	Read-only	`0x00000000`	Reserved
`0x400`	EWIC_PENDA	Read-only	`0x00000000`	Shows which internal events were pended while the EWIC was enabled. For more information, see EWIC_PENDA and EWIC_PEND<N>.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
0x404 - 0x43C	EWIC_PENDn<N>	Read-write	0x00000000	Shows which interrupts were pended while EWIC was enabled. For more information, see EWIC_PENDA and EWIC_PEND<N>.
0x440 - 0x5FC	Reserved	Read-only	0x00000000	Reserved
0x600	EWIC_PSR	Read-only	0x00000000	Pending Summary Register For more information, see EWIC_PSR.
0x604 - 0xEFC	Reserved	Read-only	0x00000000	Reserved
0xF00	ITCTRL	Read-only	0x00000000	Integration Mode Control Register For more information, see CoreSight Registers.
0xF04 - 0xF9C	Reserved	Read-only	0x00000000	Reserved
0xFA0	CLAIMSET	Read-write	0x0000000F	Claim Tag Set Register For more information, see CoreSight Registers.
0xFA4	CLAIMCLR	Read-write	0x00000000	Claim Tag Clear Register For more information, see CoreSight Registers.
0xFA8	DEVAFF0	Read-only	0x80000000	Device Affinity Register 0 For more information, see CoreSight Registers.
0xFAC	DEFAFF1	Read-only	0x00000000	Device Affinity Register 1 For more information, see CoreSight Registers.
0xFB0	LAR	Write-only	UNKNOWN	Lock Access Register For more information, see CoreSight Registers.
0xFB4	LSR	Read-only	0x00000000	Lock Status Register For more information, see CoreSight Registers.
0xFB8	AUTHSTATUS	Read-only	0x00000000	Authentication Status Register For more information, see CoreSight Registers.
0xFBC	DEVARCH	Read-only	0x47700A07	Device Architecture Register For more information, see CoreSight Registers.
0xFC0	DEVID2	Read-only	0x00000000	Device Configuration Register 2 For more information, see CoreSight Registers.
0xFC4	DEVID1	Read-only	0x00000000	Device Configuration Register 1 For more information, see CoreSight Registers.
0xFC8	DEVID	Read-only	0x00000000	Device Configuration Register For more information, see CoreSight Registers.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Access	Reset value	Description
0xFCC	DEVTYPE	Read-only	0x00000000	Device Type Register For more information, see CoreSight Registers.
0xFD0	PIDR4	Read-only	0x00000004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	Read-only	0x00000000	Reserved
0xFE0	PIDR0	Read-only	0x00000022	Peripheral ID 0
0xFE4	PIDR1	Read-only	0x000000BD	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	0x00000090	Component ID 1
0xFF8	CIDR2	Read-only	0x00000005	Component ID 2
0xFFC	CIDR3	Read-only	0x000000B1	Component ID 3

6.7.1.1 EWIC_CR

The EWIC Control Register allows software to enable or disable the EWIC.

Table 6-73: EWIC_CR register

Bits	Type	Default	Name	Description
31:1	RO	0x00000000	-	Reserved
0	RW	0x00	EN	EWIC Enable: • 0 = EWIC is disabled, events are not pended, WAKEUP is not signalled • 1 = EWIC is enabled, events are pended, WAKEUP can be signalled

6.7.1.2 EWIC_ASCR

External Wakeup Interrupt Controller Automatic Sequence Control Register (EWIC_ASCR) determines whether the processor generates APB transactions on entry and exit from Wakeup

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Interrupt Controller (WIC) sleep to set up the wakeup state in the External Wakeup Interrupt Controller (EWIC). The fields of the EWIC_ASCR are listed in the table below.

Note

Disabling ASPU and ASPD only stops the processor performing automatic updates of the EWIC. It does not affect the operation of the EWIC itself. Instead, a user can use the software to perform the same operation as that performed by the automatic sequence. For more information on the sequence performed, see the *Arm® Cortex®-M55 Technical Reference Manual*.

Table 6-74: EWIC_ASCR Register

Bits	Type	Default	Name	Description
31:2	RO	`0x00000000`	-	Reserved
1	RW	`0x01`	ASPU	Automatic Sequence on Power Up Control. This field is sent to the associated CPU and is used to decide if the CPU performs automatic EWIC access on transitioning from a low power state. • 0: No automatic sequence on power up • 1: Automatic sequence on power up
0	RW	`0x01`	ASPD	Automatic Sequence on Power Down Control. This field is sent to the associated CPU and is used to decide if the CPU performs automatic EWIC access on transitioning to a low power state. • 0: No automatic sequence on entry to a low-power state • 1: Automatic sequence on entry to a low-power state

6.7.1.3 EWIC_CLRMASK

When written, the EWIC clear mask register, causes all EWIC_MASKA and EWIC_MASK <N> registers to be cleared. The actual write data is irrelevant and ignored. Read access always returns zeros.

6.7.1.4 EWIC_NUMID

When read, this read-only register returns the number of supported events on the EWIC.

Table 6-75: EWIC_NUMID Register

Bits	Type	Default	Name	Description
31:16	RO	`0x00000`	-	Reserved
15:0	RO	CPU<N>EXPNUMIRQ + 35	NUMEVENT	Number of supported events

6.7.1.5 EWIC_MASKA and EWIC_MASK<N>

These EWIC mask registers define which events can cause the WAKEUP signal to be asserted. The EWIC_MASKA register defines the mask for special events and the EWIC_MASK<N> registers for interrupt (IRQ) events, both internal and external to SSE-300. There is one EWIC_MASK<N>

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

register implemented for every set of 32 interrupt events the EWIC supports. The EWIC_MASKA register is always implemented.

EWIC_MASKA is at address offset `0x200`. EWIC_MASK<N> is at address offset `0x204` + (N x 4).

The format of EWIC_MASKA and EWIC_MASKn is described in the following tables.

Bits	Type	Default	Name	Description
31:3	RO	`0x00000000`	-	Reserved
2	RW	`0x00`	EDBGREQ	Mask for external debug request
1	RW	`0x00`	NMI	Mask for NMI
0	RW	`0x00`	EVENT	Mask for WFE wakeup event

Note

EVENTS[0] input of the EWIC is not connected in SSE-300 but tied LOW. WFE wakeup events are not supported to wake-up SSE-300.

Table 6-77: EWIC_MASKA Register

Bits	Type	Default	Name	Description
31:0	RW	`0x00000000`	IRQ	Masks for interrupts (N x 32) to ((N+1) x 32) - 1. Any unused bit fields are reserved, and **RAZ/WI**.

Table: EWIC_MASK<N> Register

6.7.1.6 EWIC_PENDA and EWIC_PEND<N>

These EWIC Pending Registers indicate which events have been pended. The EWIC_PENDA register is used for special events and the EWIC_PEND<N> registers for interrupt (IRQ) events, both internal and external to SSE-300. There is one EWIC_PEND<N> register implemented for every set of 32 interrupt events the EWIC supports. EWIC_PENDA and at least one EWIC_PEND<N> register is always implemented.

EWIC_PENDA is at address offset `0x400`. EWIC_PEND<N> is at address offset `0x404` + (M x 4).

The format of EWIC_PENDA and EWIC_PEND<N> is described in the following tables.

Table 6-78: EWIC_PENDA Register

Bits	Type	Default	Name	Description
31:3	RO	`0x00000000`	-	Reserved
2	RO	`0x00`	EDBGREQ	External debug request is pended
1	RO	`0x00`	NMI	NMI is pended
0	RO	`0x00`	EVENT	WFE wakeup event is pended

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

EVENTS[0] input of the EWIC is not connected in SSE-300 but tied LOW. WFE wakeup events are not supported to wake-up SSE-300.

Table 6-79: EWIC_PEND<N> Register

Bits	Type	Default	Name	Description
31:0	RW	`0x00000000`	IRQ	Interrupts (M x 32) to ((M+1) x 32) - 1 are pended. Any unused bit fields are reserved, and **RAZ/WI**.

For every Event and IRQ bit that is implemented in the EWIC_PENDA and EWIC_PEND<N> register, the following apply:

- If an event occurs when EWIC_CR.EN is set, then the matching bit in EWIC_PENDA or EWIC_PEND<N> is set.
- EWIC_PENDA and all EWIC_PEND<N> registers are cleared if the EWIC is disabled when EWIC_CR.EN is cleared.

In addition, for EWIC_PEND<N>:

- EWIC_PEND<N> can be updated to `0b1` by writes to the register. Attempts to clear a bit by writing `0b0` are ignored. This allows the transfer for pending interrupts from the NVIC in the CPU to the EWIC before the CPU enters low power state, and the restoration of pending interrupts to the NVIC on power-up.

6.7.1.7 EWIC_PSR

The EWIC Pending Summary Register indicates which EWIC_PEND<N> registers are non-zero. This allows a processor to efficiently determine which EWIC_PEND<N> registers require to be read and can be used to speed up the power-on sequence. The format of EWIC_PSR is described in the following table.

Table 6-80: EWIC_PSR Register

Bits	Type	Default	Name	Description
31:16	RO	`0x00000`	-	Reserved
15:1	RO	`0x00`	NZ	Non-Zero Indication. If EWIC_PSR.NZ[N+1] is set, then EWIC_PEND<n> is non-zero.
0	RO	`0x00`	NZA	EWIC_PENDA Non-Zero Indication. This is set when EWIC_PENDA is non-zero.

6.7.1.8 CoreSight Registers

All other register from offset `0xF00` onwards are standard CoreSight registers.

For more information about CoreSight registers fields, see *Arm® CoreSight™ Architecture Specification v3.0*.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

For more information about Cortex-M55 EWIC registers, see *Arm® Cortex®-M55 Technical Reference Manual*.

6.7.2 Cortex-M55 TPIU registers

For information about Cortex-M55 TPIU registers, see *Arm® Cortex®-M55 Technical Reference Manual*.

6.8 Debug System Access Region

SSE-300 supports two key configuration for the debug system:

- HASCSS = 0. CoreSight SoC-600-based common debug infrastructure does not exist.
- HASCSS = 1. CoreSight SoC-600-based common debug infrastructure exists.

6.8.1 HASCSS = 0

When the CoreSight SoC-600 based common debug infrastructure does not exist, the debug system access region is not used and therefore the following regions are reserved:

- `0xE0100000` to `0xE01FFFFF`
- `0xF0100000` to `0xF01FFFFF`

When accessed, these regions return bus error response.

6.8.2 HASCSS = 1

SSE-300 does not support HASCSS = 1.

6.9 Peripheral Expansion Region

When EXPLOGIC_PRESENT = 1, a system timestamp generator (System Counter) is integrated in SSE-300 expansion and drives the system timestamp interface see System Timestamp Interface.

The System Counter resides in the PD_AON Power domain, is clocked by CNTCLK and is reset by **nWARMRESETAON**.

Table 6-81: Peripheral Expansion Region address map when EXPLOGIC_PRESENT = 1

Row ID	Address from	Address to	Size	Region name	Description	Alias with row ID	Security[1]
-	`0x48100000`	`0x48100FFF`	4KB	Reserved	Reserved (RAZ/WI)		

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Row ID	Address from	Address to	Size	Region name	Description	Alias with row ID	Security[1]
1	`0x48101000`	`0x48101FFF`	4KB		System Counter (System Timestamp Generator) Status Frame register[3]. See System Timestamp Interface.	3	NS
2	`0x58100000`	`0x58100FFF`	4KB		System Counter (System Timestamp Generator) Control Frame register[3]. See System Timestamp Interface.		S
3	`0x58101000`	`0x58101FFF`	4KB		System Counter (System Timestamp Generator) Status Frame register[3]. See System Timestamp Interface.	1	S

1. S: Secure access only.

   NS: Non-secure access only.

For details of the ARMv8M System Counter registers, see *Arm® SSE-123 Example Subsystem Technical Reference Manual*.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Appendix A Mapping of the user signals of the AXI and AHB expansion interfaces

The section defines the mapping (MSB to LSB) of the user signals of the AXI and AHB expansion interfaces.

User signals transfer side band signals of the processor and the master ID to identify the master that started the transaction. Users can use these signals for expanding the system, for example, add additional firewalls to filter transactions based on the master ID, so specific memory ranges are only visible to specific masters.

AxUSER mapping of the XSLVTCM TCM DMA slave interface

- MID: user-defined value to identify the master or master group targeting this interface
- DAI: Debug Access Identification

AxUSER mapping of the XSLVEXPMI0 Main interconnect expansion slave interface

- AxINNER: See *Arm® Cortex®-M55 Technical Reference Manual*
- AxDOMAIN: See *Arm® Cortex®-M55 Technical Reference Manual*

Note

AxDOMAIN value is used to calculate HPROT[6] value for transactions targeting Peripheral interconnect. See *Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual*

- MID: user-defined value to identify the master or master group targeting this interface
- DAI: Debug Access Identification

HAUSER mapping of the HSLVEXPMI1 Main interconnect expansion slave interface

- AxINNER: See *Arm® Cortex®-M55 Technical Reference Manual*
- MID: user-defined value to identify the master or master group targeting this interface
- DAI: Debug Access Identification

HAUSER mapping of the HSLVEXPPILL and HSLVEXPPIHL Peripheral interconnect expansion slave interfaces

- MID: user-defined value to identify the master or master group targeting this interface
- DAI: Debug Access Identification

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

AxUSER mapping of the XMSTEXPCODE, XMSTEXPSRAM and XMSTEXPDEV Main interconnect expansion master interfaces

- AxINNER: See *Arm® Cortex®-M55 Technical Reference Manual*
- AxDOMAIN: See *Arm® Cortex®-M55 Technical Reference Manual*
- MID:
    - ASIB_ID of the slave interface of the Main interconnect targeting the master interface
    - user-defined MID value from the targeting slave interface to identify the master or master group targeting this interface
- DAI: Debug Access Identification

HAUSER mapping of the HMSTEXPPILL and HMSTEXPPIHL Peripheral interconnect expansion master interfaces

- MID:
    - HSIF_ID of the slave interface of the Peripheral interconnect targeting the master interface

      HSIF_ID is the predefined ID of the slave interfaces of the Peripheral interconnect
        - `3'b000`: connected to CPU0 P-AHB interface
        - `3'b100`: connected to Main interconnect interface
        - `3'b101`: connected to HSLVEXPPILL Peripheral interconnect expansion slave interfaces
        - `3'b110`: connected to HSLVEXPPIHL Peripheral interconnect expansion slave interfaces
    - ASIB_ID of the slave interface of the Main interconnect targeting the master interface or padded zeros

      ASIB_ID is the predefined ID of the slave interfaces of the Main interconnect
        - `3'b000`: connected to CPU0 M-AXI interface
        - `3'b101`: connected to XSLVEXPMI0 Main interconnect expansion slave interface
        - `3'b110`: connected to HSLVEXPMI1 Main interconnect expansion slave interface
        - user-defined MID value from the targeting slave interface to identify the master or master group targeting this interface
- DAI: Debug Access Identification

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Appendix B Mapping of the HMASTER signals of the AHB expansion master interfaces

This section defines the mapping (MSB to LSB) of the HMASTER signals of the AHB expansion master interfaces.

The following list shows how the HMASTER signals are generated in SSE-300. For more information on HMASTER signaling, see *Arm® AMBA® 5 AHB Protocol Specification*.

HMASTER mapping of the HMSTEXPPILL and HMSTEXPPIHL master interfaces

- HSIF_ID:
    - See HAUSER mapping of the HMSTEXPPILL and HMSTEXPPIHL Peripheral interconnect expansion master interfaces for details
- HMASTER:
    - From NIC400 via XHB: AxID from NIC400
    - From CPU P-AHB: CPUID
    - From HSLVEXPPILL and HSLVEXPPIHL:

      HSLVEXPPILLHMASTER/HSLVEXPPIHLHMASTER user-defined value to identify the master targeting this interface

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Appendix C Configurable render options

SSE-300 provides several render configuration options.

About verification

SSE-300 is presented as an example of a Arm Cortex-M55-based IoT solution. As an example system, this has not been verified to the same level of quality when compared to other components delivered from Arm that are intended to be directly used in partner solutions.

If you want to reuse any part of SSE-300 for your own application, you must perform your own verification and validation scenarios that are specific to your use-cases to ensure the system operates as intended.

The delivered subsystem can be used without modification if this fits your given use-case. Alternatively, you can choose to update the design files provided as part of the example system to better fit your specific IoT application. In any of these cases, you must verify and validate your subsystem to ensure that the design is of suitable quality.

When using the Proven range, you are compliant with the Allowed Standard Modifications that [Modification rules] describes.

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 aid you in avoiding 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.

CPU0_INITNSVTOR_ADDR_INIT

Legal range:

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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*.

CPU0CPUIDRST

- Legal range: 0-15
- A unique Identity value defined for each CPU in the system. It defines the values read at CPU 0 local's CPU0CPUID register. Legal values are within 0 to 15, inclusive.

CPU0EXPNUMIRQ

- Legal range: 2-448
- It specifies the number of expansion interrupt for each CPU.

CPU0_INT_IRQTIER[31:0]

- Legal range: all values
- Specifies the interrupts that:
    - Support the lowest interrupt latency
    - Incur extra latency

  The options are:
    - CPU0_INT_IRQTIER[n] = 0 - This option indicates the lowest latency for IRQn.
    - CPU0_INT_IRQTIER[n] = 1 - This option indicates the highest latency for IRQn.

  If many interrupts are included, CPU0_INT_IRQTIER improves implemented frequency at the expense of interrupt latency CPU0_INT_IRQTIER[i] = 1'b0 enables lowest latency on CPU0's IRQ[i].

CPU0_EXP_IRQTIER [CPU0EXPNUMIRQ-1:0]

- Legal range: all values

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Specifies the interrupts that:
    - Support the lowest interrupt latency
    - Incur extra latency

    The options are:

    - CPU0_EXP_IRQTIER[n] = 0 - This option indicates the lowest latency for IRQn.
    - CPU0_EXP_IRQTIER[n] = 1 - This option indicates the highest latency for IRQn.

    If many interrupts are included, CPU0_EXP_IRQTIER improves implemented frequency at the expense of interrupt latency CPU0_EXP_IRQTIER[i] = 1'b0 enables lowest latency on CPU0's IRQ[i+32].

CPU0_EXP_IRQ_PULSE_SPT_PRESENT[CPU0EXPNUMIRQ-1:0]

- Legal range: {(CPU0EXPNUMIRQ){1'b0}}
- Enables support for Pulse type of interrupts on individual expansion interrupts that are used for CPU0. CPU0_EXP_IRQ_PULSE_SPT_PRESENT[i] = 1'b1 enables pulse interrupt capture on CPU0 IRQ[i+32].

CPU0EXPIRQDIS [CPU0EXPNUMIRQ-1:0]

- Legal range: CPU0EXPNUMIRQ{1'b0} - CPU0EXPNUMIRQ{1'b1}
- It specifies for each CPU whether each expansion interrupt bit is implemented or disabled. CPU0EXPIRQDIS[i] = 1'b1 indicates that IRQ[i+32] is not present on CPU0

CPU0_EXP_IRQ_SYNC_TO_CPU_PRESENT[CPU0EXPNUMIRQ-1:0]

- Legal range: {(CPU0EXPNUMIRQ){1'b1}}
- Enables IRQ synchronization for individual expansion interrupts that are used for CPU0. CPU0_EXP_IRQ_SYNC_TO_CPU_PRESENT[i] = 1'b1 enables synchronizer on CPU0 IRQ[i+32] to CPU interrupt input.

CPU0_EXP_IRQ_SYNC_TO_EWIC_PRESENT[CPU0EXPNUMIRQ-1:0]

- Legal range: {(CPU0EXPNUMIRQ){1'b1}}
- Enables IRQ synchronization for individual expansion interrupts that are used for CPU0 EWIC. CPU0_EXP_IRQ_SYNC_TO_EWIC_PRESENT[i] = 1'b1 enables synchronizer on CPU0 IRQ[i+32] to EWIC interrupt input.

CPU0_EXP_NMI_SYNC_TO_EWIC_PRESENT

- Legal range: 1
- Enables IRQ synchronization for NMI expansion interrupt that is used for CPU0. CPU0_EXP_NMI_SYNC_TO_EWIC_PRESENT = 1'b1 enables synchronizer on NMI to EWIC expansion interrupt input

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

CPU0_EXP_NMI_SYNC_TO_CPU_PRESENT

- Legal range: 1
- Enables IRQ synchronization for NMI expansion interrupt that is used for CPU0. CPU0_EXP_NMI_SYNC_TO_CPU_PRESENT = 1'b1 enables synchronizer on NMI to CPU expansion interrupt input

CPU0_EXP_NMI_PULSE_SPT_PRESENT

- Legal range: 0
- Enables support for Pulse type of interrupts on expansion NMI interrupt that are used for CPU. CPU0_EXP_NMI_PULSE_SPT_PRESENT = 1'b1 enables pulse interrupt capture on CPU expansion NMI input

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. Scalar floating point CDE instruction support included.

Note

The floating-point functionality is separately licensable. See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the Floating Point Unit bundle name and availability.

CPU0_MPU_NS

- Legal range: 4,8,12,16
- Specifies the number of Non-secure Memory Protection Unit (MPU) regions included. The options are:
    - 0 - No MPU regions
    - 4 - 4 MPU regions
    - 8 - 8 MPU regions
    - 12 - 12 MPU regions
    - 16 - 16 MPU regions

Note

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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

CPU0_MPU_S

- Legal range: 4,8,12,16
- Specifies the number of Secure MPU regions included. The options are:
    - 0 - No MPU regions
    - 4 - 4 MPU regions
    - 8 - 8 MPU regions
    - 12 - 12 MPU regions
    - 16 - 16 MPU regions

Note

- 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_NUM_SAU_CONFIG

- Legal range: 4,8
- Specifies the number of Security Attribution Unit (SAU) regions included. The options are:
    - 0 - No SAU regions
    - 4 - 4 SAU regions
    - 8 - 8 SAU regions

Note

- 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.

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

CPU0_DBGLVL

- Legal range: 1,2,3

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.
    - 3 - Full set. Eight 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_IRQLVL

- Legal range: 3-8
- Specifies the number of exception priority bits.

MPCEXPDIS

- 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

- Legal range: `0xFFFF` - `0x0000`
- It disables support for individual bits on the SMSCEXPSTATUS, SMSCEXPCLEAR and NSMSCEXP buses.

  If MSCEXPDIS[n] = 1'b1, then either SMSCEXPSTATUS[n], SMSCEXPCLEAR[n] and NSMSCEXP[n] does not exist, or if they do exist, SMSCEXPSTATUS[n] is not used, SMSCEXPCLEAR[n] are tied LOW and NSMSCEXP[n] are tied HIGH.

BRGEXPDIS

- Legal range: `0xFFFF` - `0x0000`
- It disables support for individual bits on the BRGEXPSTATUS and BRGEXPCLEAR buses.

  If BRGEXPDIS[n] = 1'b1, then either BRGEXPSTATUS[n] and BRGEXPCLEAR[n] does not exist, or if they do exist, BRGEXPSTATUS[n] is not used and BRGEXPCLEAR[n] are tied LOW.

PERIPHPPCEXP0DIS

- Legal range: `0xFFFF` - `0x0000`

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- It disables support for individual bits on the PERIPHNSPPCEXP{0-3} and PERIPHPPPCEXP{0-3} buses.

  If PERIPHPPCEXP{0-3}DIS[n] = 1'b1, then either PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0 to 3}[n] does not exist, or if they do exist, PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0-3}[n] are not used and tied LOW.

PERIPHPPCEXP1DIS

- Legal range: `0xFFFF` - `0x0000`
- It disables support for individual bits on the PERIPHNSPPCEXP{0-255} and PERIPHPPPCEXP{0-3} buses.

  If PERIPHPPCEXP{0-3}DIS[n] = 1'b1, then either PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0 to 3}[n] does not exist, or if they do exist, PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0-3}[n] are not used and tied LOW.

PERIPHPPCEXP2DIS

- Legal range: `0xFFFF` - `0x0000`
- It disables support for individual bits on the PERIPHNSPPCEXP{0-3} and PERIPHPPPCEXP{0-3} buses.

  If PERIPHPPCEXP{0-3}DIS[n] = 1'b1, then either PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0 to 3}[n] does not exist, or if they do exist, PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0-3}[n] are not used and tied LOW.

PERIPHPPCEXP3DIS

- Legal range: `0xFFFF` - `0x0000`
- It disables support for individual bits on the PERIPHNSPPCEXP{0-3} and PERIPHPPPCEXP{0-3} buses.

  If PERIPHPPCEXP{0-3}DIS[n] = 1'b1, then either PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0 to 3}[n] does not exist, or if they do exist, PERIPHNSPPCEXP{0-3}[n] and PERIPHPPPCEXP{0-3}[n] are not used and tied LOW.

MAINPPCEXP0DIS

- 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 to 3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0 to 3}[n] are not used and tied LOW.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

MAINPPCEXP1DIS

- 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 to 3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0 to 3}[n] are not used and tied LOW.

MAINPPCEXP2DIS

- 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 to 3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0 to 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 to 3}[n] does not exist, or if they do, MAINNSPPCEXP{0-3}[n] and MAINPPPCEXP{0 to 3}[n] are not used and tied LOW.

NSMSCEXPRST

- Legal range: `0x0000` - `0xFFFF`
- The reset value for NSMSCEXP.NS_MSCEXP[15:0]. This value defines the security world 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

SECEXT

- Legal range: 1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

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

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.

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-M55 configuration file. The TPIU logic is optimized in case no ITM trace present. No optimization when CoreSight SoC-600 TPIU is used.

CPU0_ETM_PRESENT

- Legal range: 0,1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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-M55 TPIU logic is optimized in case no ETM trace present. No optimization when CoreSight SoC-600 TPIU is used.

HASCPU0IWIC

- Legal range: 0
- Specifies whether the Internal Wake-up Interrupt Controller (IWIC) is included
  - 0 - IWIC not included
  - 1 - IWIC included

CPU0_CTI_PRESENT

- Legal range: 1
- Specifies whether the Cross Trigger Interface (CTI) unit is included

CPU0_CDERTLID

- Legal range: 0-255
- Specifies the RTL ID for the code that is present in the customizable CDE module. For systems that include several processors with different CDE customizations, this parameter must have a different value for each processor.
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP0

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
  - 0 - CPn instructions mapped to coprocessor n
  - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP1

- Legal range: 0,1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
  - 0 - CPn instructions mapped to coprocessor n
  - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP2

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
  - 0 - CPn instructions mapped to coprocessor n
  - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP3

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
  - 0 - CPn instructions mapped to coprocessor n
  - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP4

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
  - 0 - CPn instructions mapped to coprocessor n
  - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP5

- Legal range: 0,1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
    - 0 - CPn instructions mapped to coprocessor n
    - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP6

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
    - 0 - CPn instructions mapped to coprocessor n
    - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

CPU0_CDEMAPPEDONCP7

- Legal range: 0,1
- Specify whether the instruction for a given coprocessor is redirected to the CDE module:
    - 0 - CPn instructions mapped to coprocessor n
    - 1 - CPn instructions mapped to CDE

  The instruction classes supported will depend on MVE and FPU parameters
- For more details about the effect of this parameter value on CDE, see *Arm® Cortex®-M55 Processor Integration and Implementation Manual*

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-300.

CPU0_LOCKSTEP

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Specifies whether the processor is configured for dual-redundant lockstep operation. The options are:
  - 0 - Regular processor operation
  - 1 - Dual-redundant lockstep operation

Note

When CPU0_LOCKSTEP is set to 1, Arm recommends that CPU0_RAR must also be set to 1. Otherwise, software must initialize all registers to prevent accidental triggering of Dual-Core Lock-Step (DCLS) mismatch when the **UNKNOWN** state gets propagated to the top level. However, such software requirements might not be practical in certain software system environments. Therefore, the CPU0_RAR option is strongly recommended for DCLS designs.

CPU0_ITGUBLKSZ

- Legal range: 3-15
- ITCM gate unit block size
  - Size in bytes = $2^{(CPU0\_ITGUBLKSZ+5)}$
  - Minimum block size is 32 bytes
  - Maximum block size is 1MB
  - Arm recommends that CPU0_ITGUBLKSZ and CPU0_DTGUBLKSZ are equal to VMMPCBLKSIZE

CPU0_DTGUBLKSZ

- Legal range: 3-15
- DTCM gate unit block size.
  - Size in bytes = $2^{(CPU0\_DTGUBLKSZ+5)}$
  - Minimum block size is 32 bytes
  - Maximum block size is 1MB
  - Arm recommends that CPU0_ITGUBLKSZ and CPU0_DTGUBLKSZ are equal to VMMPCBLKSIZE

CPU0_PMC_PRESENT

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Specifies whether the Programmable MBIST Controller (PMC-100) is included
  - 0 - PMC-100 is excluded
  - 1 - PMC-100 is included

Note

The PMC-100 is delivered with the other processor deliverables.

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

Note

- 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.
- When CPU0_LOCKSTEP is set to 1, Arm recommends that CPU0_RAR must also be set to 1. Otherwise, software must initialize all registers to prevent accidental triggering of Dual-Core Lock-Step (DCLS) mismatch when the **UNKNOWN** state gets propagated to the top level. However, such software requirements might not be practical in certain software system environments. Therefore, the CPU0_RAR option is strongly recommended for DCLS designs.

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-M55 processor to use different library cells for the embedded cache RAM.

CPU0_INSTR_CACHE_SIZE

- Legal range: CPU0_INSTR_CACHE_SIZE[0] = `0b1`

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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

Note

Setting CPU0_INSTR_CACHE_SIZE to any other value causes UNPREDICTABLE behavior.

CPU0_DATA_CACHE_SIZE

- Legal range: CPU0_DATA_CACHE_SIZE[0] = `0b1`
- Specifies the Master 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

Note

Setting CPU0_DATA_CACHE_SIZE to any other value causes UNPREDICTABLE behavior.

XOM_USER_SIGNAL_PRESENT

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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, XMSTEXPSRAM and from CryptoCell™ (if HASCRYPTO==1) transferred to `XSLVEXPMI{0-<NUM_AXI_SLAVES_EXP_MI-1>}`.

  When set to 0, RUSER[0] is not present.

Note

- 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-300 then the foregoing cache might store the XOM attribute in TAG memory and might use it to check access type upon cache hit in order 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.

SOCIMPLID

- Legal range: NA
- 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 it's own JEP106 code. To obtain a number, or to see the assignment of these codes, contact JEDEC.

SOCREV

- Legal range: NA
- SoC minor revision code. SoC integrator is expected to change it

SOCVAR

- Legal range: NA
- 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-300 r0p0
  - `0x1` - corresponds to SSE-300 r0p1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

SOCPRTID

- Legal range: NA
- SoC product identity code. Connected to Debug Port targetid port TPARTNO fields SoC integrator is expected to change it to it's own ID code.

IMPLID

- Legal range: NA
- Subsystem implementor JEP106 code. Connected to MCU ROM table's DES_2,DES_1,DES_0 fields (JEPID and JEPCONTINUATION parameter values).
    - IMPLID[11:8] indicates the JEP106 continuation code
    - IMPLID[7] is always 0
    - IMPLID[6:0] indicates the JEP106 identification code. SoC integrator must change it.

IMPLREV

- Legal range: NA
- Subsystem minor revision code. Connected to MCU ROM table's REVAND field (ECOREVNUM input) SoC integrator must use this to indicate metal fixes on their customized subsystem.

IMPLVAR

- Legal range: NA
- Subsystem variant or major revision code. Connected to MCU ROM table's REVISION field (REVISION Parameter) SoC integrator must use this to track the revision of their customized subsystem.

IMPLPRTID

- Legal range: NA
- Subsystem product identity code. Connected to MCU ROM table's PART_0,PART_1 fields (PARTNUMBER Parameter) SoC integrator must modify product identity code to identify the their customized subsystem.

TCM_MID_WIDTH

- Legal range: 1-30
- MasterID (MID) width of the TCM interface, Expansion AXI slave (XSLVTCM). Master Identification (MID) support: MID information must be transferred on **AxUSER** and **HAUSER** signals. MID must be a unique ID might be used by the expansion logic to identify the master (not the thread) or group of masters initiating the access. Refer to TCM_AXUSER_WIDTH for more information about the signal encoding

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

TCM_ID_WIDTH

- Legal range: 2-32
- TCM interface channel ID width of XSLVTCM Expansion AXI slave interface

XS_ID_WIDTH

- Legal range: (NUMNPU > 0 ? 6 : 4) - 9
- Channel ID width for Expansion Slave Main Interconnect interfaces, CPU M-AXI interface, slave interfaces of the Main Interconnect, ACGs implemented in the Main Interconnect and slave interfaces of the NIC-400, same width for all channels

---

Note

When NPU is present, the legal range is different.

S_MID_WIDTH

- Legal range: 4-24
- Master ID width for Expansion Slave AXI, Expansion Slave AHB, 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 Master ID based access control. NPU<0 - NUMNPU-1>_ID is provided as MID value when NPU is present.

S_HMASTER_WIDTH

- Legal range: 4-12
- S_HMASTER_WIDTH defines the HMASTER width for HSLVEXPPI* interfaces

CPU0_SAUDISABLE

- Legal range: 0
- If the Security Attribution Unit (SAU) is configured, disables support

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-300

COLDRESET_MODE

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

  When set to 1, an entity outside the subsystem is expected to observe the following signals to decide when to drive HOSTRESETREQ:
  - **NSWDRSTREQSTATUS**
  - **SWDRSTREQSTATUS**
  - **SSWDRSTREQSTATUS**
  - **RESETREQSTATUS**
  - **SWRSTREQSTATUS**

Note

NSWDRSTREQSTATUS, SWDRSTREQSTATUS, SSWDRSTREQSTATUS, RESETREQSTATUS, and SWRSTREQSTATUS ports do not exist when COLDRESET_MODE = 0.

CFGBIGEND

- Legal range: 0
- Data endian format
  - 0 - Little-endian (LE)
  - 1 - Byte invariant big-endian (BE8) Global Configuration for all processors

CPU0_CFGDTCMSZ

- Legal range: CPU0_CFGDTCMSZ > `0b0010`
- Size of the data TCM region encoded as:
  - CPU0_CFGDTCMSZ = `0b0000` - No DTCM implemented
  - CPU0_CFGDTCMSZ > `0b0010` - $2^{(CPU0\_CFGDTCMSZ-1)}$ KB

Note

- The minimum size 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-300

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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` - $2^{(CPU0\_CFGITCMSZ-1)}$ KB

  The minimum size of Tightly Coupled Memory (TCM) is 4KB and the maximum size is 16MB. Setting CPU0_CFGITCMSZ to `0b0001` or `0b0010` results in UNPREDICTABLE behavior.

Note

CPU0_CPU0_CFGITCMSZ = `0b0000` - No ITCM implemented status is not supported by SSE-300.

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

Note

Setting CFGPAHBSZ to any other value results in UNPREDICTABLE behavior.

CFGMEMALIAS

- Legal range: `0b10000`

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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 Global configuration for all processors and the system.

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-M55 Processor Technical Reference Manual Global configuration for all processors.

  In SSE-300, the NIC in the TCM address range responds with error on M-AXI of the CPU thus 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:
  - HIGH P-AHB is enabled
  - LOW P-AHB is disabled

  For more information on PAHBCR, see ***???***.

  Global Configuration for all processors.

CPU0_INITECCEN

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

  ECC must not be enabled dynamically when the processor is in the MEM_RET Power Mode as the L1 cache will not be automatically invalidated when the Power Mode is switched to ON. This results in inconsistent ECC information relative to the data retained in the cache and will cause an ECC error to occur

CPU0MCUROMADDR

- Legal range: `0xE0049000` - `0xE00FE000`
- The address pointer to MCU ROM table private to each CPU core. Only the 20 most significant bits are configurable. All lower address bits are zeros. This configuration point 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 point must exist when HASCSS = 1.

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-300 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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

SSE_CONFIG

- Legal range:

  PART:

  CG067-BU-50000

  VERSION:

  r0p1-00eac0

- This parameter defines the part and the version of SSE-300 that is used for the rendering of the subsystem. Do not change the pre-configured value it is provided for information.

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

Note

The OOB testbench is instantiated inside the Interface blocks, in order to run any of the OOB tests you must enable the expansion logic.

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.
  - When PILEVEL = 0, NUMCPU must be set to 0.
  - When HASCSS = 0, there can only be one CPU

PILEVEL

- Legal range: 1

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Power Infrastructure Level. It defines the implemented power structure of the system:
    - 0 - Basic Power Structure
    - 1 - Intermediate Power Structure
    - 2 - Advance Power infrastructure
    - Others - Reserved

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.

  TCM's block sizes are defaulted to VM block sizes.

  Arm recommends that ITGUBLKSZ and DTGUBLKSZ are equal to VMMPCBLKSIZE

VMADDRWIDTH

- Legal range: if NUMVMBANK!= 0 then 14 - . (24-ceil($\log_2$(NUMVMBANK)))
- It defines the address width for all Volatile Memory Banks. This then defines the size of each bank as $2^{VMADDRWIDTH}$ bytes.

NUMVMBANK

- Legal range: 2
- It selects the number of Volatile Memory Banks

HASCRYPTO

- Legal range: 0
- It defines whether CryptoCell-312 is included:
    - 0 - No
    - 1 - Yes

HASCSS

- Legal range: 0
- It defines whether the CoreSight SoC-600 based Debug infrastructure is included.
    - 0 - No
    - 1 - Yes

  HASCSS must be 1 when NUMCPU > 0. If DEBUGLEVEL=0 then HASCSS must be set to 0.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

CPU0TYPE

- Legal range: 3
- It describes if each CPU exists, and the type of CPU that is integrated:
    - 0 - Not implemented
    - 3 - Cortex-M55
    - Others - Reserved

Note

CPU0TYPE must not be 0 and sparse CPU are not supported. Hence CPU0TYPE to CPU<NUMCPU>TYPE must not be 0s.

CPU1TYPE

- Legal range: 0
- It describes if each CPU exists, and the type of CPU that is integrated:
    - 0 - Not implemented
    - 3 - Cortex-M55
    - Others - Reserved

Note

CPU1TYPE must not be 0 and sparse CPU are not supported. Hence CPU1TYPE to CPU<NUMCPU>TYPE must not be 0s.

CPU2TYPE

- Legal range: 0
- It describes if each CPU exists, and the type of CPU that is integrated:
    - 0 - Not implemented
    - 3 - Cortex-M55
    - Others - Reserved

Note

CPU2TYPE must not be 0 and sparse CPU are not supported. Hence CPU2TYPE to CPU<NUMCPU>TYPE must not be 0s.

CPU3TYPE

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- It describes if each CPU exists, and the type of CPU that is integrated:
  - 0 - Not implemented
  - 3 - Cortex-M55
  - Others - Reserved

Note

CPU3TYPE must not be 0 and sparse CPU are not supported. Hence CPU3TYPE to CPU<NUMCPU>TYPE must not be 0s.

NUM_AHB_SLAVES_EXP_PIHL

- Legal range: 1
- Number of High Latency Slave Peripheral Expansion Interface interfaces on Peripheral interconnect

NUM_AHB_SLAVES_EXP_PILL

- Legal range: 1
- Number of Low Latency Slave Peripheral Expansion Interface interfaces on Peripheral interconnect

NUM_AXI_SLAVES_EXP_MI

- Legal range: 2
- Number of Main interconnect expansion AXI slave interfaces `XSLVEXPMI{0-<NUM_AXI_SLAVES_EXP_MI-1>}`.

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

  Global configuration for all processors.

  TCM XOM is not supported. If PMC-100 included in the CPU then M-AXI is not suitable for XOM Secure Privileged input.

LOGIC_RETENTION_PRESENT

- Legal range: 0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Defines if the power domains PD_CPU{0-<NUMCPU>}, PD_CPU{0-<NUMCPU>}EPU 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

NUMNPU

- Legal range: 0,1
- It describes the number of NPU cores in the subsystem. The number of cores is equal to NUMNPU.

NPU0_NUM_MACS

- Legal range: 32,64,128,256
- The number of 8x8 MACs that are performed per cycle

NPU1_NUM_MACS

- Legal range: 32,64,128,256
- The number of 8x8 MACs that are performed per cycle

NPU2_NUM_MACS

- Legal range: 32,64,128,256
- The number of 8x8 MACs that are performed per cycle

NPU3_NUM_MACS

- Legal range: 32,64,128,256
- The number of 8x8 MACs that are performed per cycle

NPU0TYPE

- Legal range: 0,1
- It describes the integrated NPU type for NPU0 if NPU0 exists:
    - 0 - Not present
    - 1 - Ethos-U55
    - Others - Reserved

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

Sparse NPUs are not supported. Hence, when NUMNPU>0, any NPU{0:<NUMNPU-1>}TYPE must not be 0

NPU1TYPE

- Legal range: 0
- It describes the integrated NPU type for NPU1 if NPU1 exists:
  - 0 - Not present
  - 1 - Ethos-U55
  - Others - Reserved

Note

Sparse NPUs are not supported. Hence, when NUMNPU>0, any NPU{0:<NUMNPU-1>}TYPE must not be 0

NPU2TYPE

- Legal range: 0
- It describes the integrated NPU type for NPU2 if NPU2 exists:
  - 0 - Not present
  - 1 - Ethos-U55
  - Others - Reserved

Note

Sparse NPUs are not supported. Hence, when NUMNPU>0, any NPU{0:<NUMNPU-1>}TYPE must not be 0

NPU3TYPE

- Legal range: 0
- It describes the integrated NPU type for NPU3 if NPU3 exists:
  - 0 - Not present
  - 1 - Ethos-U55
  - Others - Reserved

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

Sparse NPUs are not supported. Hence, when NUMNPU>0, any NPU{0:<NUMNPU-1>}TYPE must not be 0

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.

Note

This parameter cannot be combined with NPU0_NUM_MACS=32

NPU1_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.

Note

This parameter cannot be combined with NPU0_NUM_MACS=32

NPU2_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.

Note

This parameter cannot be combined with NPU0_NUM_MACS=32

NPU3_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.

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

This parameter cannot be combined with NPU0_NUM_MACS=32

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.

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.

Note

The *Arm® AMBA® 5 AHB Protocol Specification* recommends that a bus master inserts an IDLE transfer after a locked transfer.

DEBUGLEVEL

- Legal range: 2
- Selects the debug level of the subsystem:
    - 0 - Debug System does not exist. There are no Debug Access nor Trace support.
    - 1 - Debug system exist without Trace support. Debug Access Interface(s) exist, but Trace is not supported.
    - 2 - Debug system exist with Trace support. Both Debug Access Interface(s) and Trace Interface exist.

CPU0_MVE_CONFIG

- Legal range: `CPU0_FPU_PRESENT ? (2|1|0) : 1`

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- M-profile Vector Extension (CPU0_MVE_CONFIG) parameter can have the following configuration options:
    - 0 - MVE not included
    - 1 - Integer subset of MVE included, Scalar floating point CDE instruction support included.
    - 2 - Integer and half and single-precision floating-point MVE included. Scalar floating point, and vector CDE instruction support included . This option is only valid if CPU0_FPU_PRESENT=1.

  SSE-300 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®-M55 Processor Integration and Implementation Manual*

CMSDK_CONFIG

- Legal range:

  PART:

  BP200-BU-00000 | 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:

    - BP200-BU-00000 = The subsystem uses the Cortex-M0_M0+ System Design Kit components.
    - 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

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the CMSDK part and version names.

CORTEX_M55_CONFIG

- Legal range:

  PART:

  AT633-BU-50000

  VERSION:

  r1p0-00eac0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- This parameter defines the part and the version of the Cortex-M55 product that is used for the rendering of the subsystem. The specified Cortex-M55 product version has to be present in the <download_folder> prior to subsystem rendering.

  PART:

  AT633-BU-50000 = The subsystem uses the selected part

  VERSION:

  r1p0-00eac0 = The subsystem uses the selected version of the PART

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the Cortex-M55 part and version names.

CORTEX_M55_FPU_CONFIG

- Legal range:

  PART:

  AT634-MN-22110

  VERSION:

  r1p0-00eac0

- This parameter defines the part and the version of the Cortex-M55 FPU product that is used for the rendering of the subsystem. The specified Cortex-M55 FPU product version has to be present in the <download_folder> prior to subsystem rendering.

  PART:

  AT634-MN-22110 = The subsystem uses the selected part

  VERSION:

  r1p0-00eac0 = The subsystem uses the selected version of the PART

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the Cortex-M55 FPU part and version names.

CORTEX_M55_ETM_CONFIG

- Legal range:

  PART:

  TM981-BU-50000

  VERSION:

  r1p0-00eac0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- This parameter defines the part and the version of the Cortex-M55 ETM product that is used for the rendering of the subsystem. The specified Cortex-M55 ETM product version has to be present in the `<download_folder>` prior to subsystem rendering.

  PART:

  TM981-BU-50000 - The subsystem uses the selected part.

  VERSION:

  r1p0-00eac0 - The subsystem uses the selected version of the PART.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the Cortex-M55 ETM 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.

Note

See *Arm® Corstone™ SSE-300 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the SIE-300 part and version names.

PCK600_CONFIG

- Legal range:

  PART:

  PL608-BU-50000

  VERSION:

  r0p4-00eac0

- 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:

  r0p4-00eac0 - The subsystem uses the selected version of the PART.

Note

See *Arm® Corstone™ SSE-300 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

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the XHB-500 part and version names.

DAPLITE2_CONFIG

- Legal range:

  PART:

  TM840-BU-50000

  VERSION:

  r2p0-00rel0

- 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.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the CoreSight part and version names.

ETHOS_U55_CONFIG

- Legal range:

  PART:

  ML004-BU-50000

  VERSION:

  r1p0-00eac0

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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:

  r1p0-00eac0 - The subsystem uses the selected version of the PART.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the Ethos-U55 part and version names.

NIC400_CONFIG

- Legal range:

  PART:

  PL410-BU-50000| PL401-BU-50000

  VERSION:

  r1p1-00rel0| 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:

  - PL410-BU-50000 - The subsystem uses the NIC-400 Lite part.
  - PL401-BU-50000 - The subsystem uses the NIC-400 part.

  VERSION:

  - r1p1-00rel0 - The subsystem uses the NIC-400 Lite version of the PART.
  - r1p2-00rel0 - The subsystem uses the NIC-400 version of the PART.

Note

See *Arm® Corstone™ SSE-300 Example Subsystem Release Note* for more information about the NIC-400 part and version names.

UARCH_CONFIG

- Legal range: 0,2

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 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

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 need to specify: `"--license socrates_flexibleaccess_ms"`

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Appendix D Revisions

This appendix describes the technical changes between released issues of this book.

Revisions

Table D-1: Issue 01

Change	Location
Initial issue of the document	-

Table D-2: Differences between Issue 01 and Issue 02

Change	Location
Updated Configurable render options and moved to Appendix	Appendix D
Introduced the Ethos-U55 NPU in SSE-300	-

Table D-3: Differences between Issue 02 and Issue 03

Change	Location
Added CPU Custom Datapath Extension Interface section	-
Note added about the memory access of non-privileged masters	Peripheral Protection Controllers section
Description of CPU0_LOCKSTEP configuration updated, Note added	CPU section
NPU0_CUSTOM_DMA_PRESENT, NPU1_CUSTOM_DMA_PRESENT, NPU2_CUSTOM_DMA_PRESENT and NPU3_CUSTOM_DMA_PRESENT configuration options have been added for documentation purposes not configurable	Configurable render options section
Fixed PERIPHPPCEXP2DIS typo	Configurable render options section
UARCH_CONFIG configuration option has been added	Configurable render options section
SOCRATES_CONFIG configuration option has been added	Configurable render options section
SOCVAR configuration option has been updated	Configurable render options section
SSE_CONFIG configuration option has been added	Configurable render options section
HMASTER mapping for AHB expansion master interfaces has been added	New appendix Mapping of the HMASTER signals of the AHB expansion master interfaces.
Bit order of register fields of PWRCTRL register in System Control Register Block swapped	PWRCTRL section
SIE200_CONFIG configuration option has been updated to support r3p2-00rel0 version	Configurable render options section
CPU0_DBGLVL configuration option has been added with the new value 3 to legal range	Configurable render options section
Description of XOM_USER_SIGNAL_PRESENT updated with a note of online MBIST	Configurable render options section
CPU0_IDCACHEID configuration option has been added	Configurable render options section
CPU0_CDERTLID configuration option has been added	Configurable render options section
Added interface usage rules for expansion power P-channel device interfaces	Power Control P-Channel Device Interfaces chapter.
Updated product versions of CORTEX_M55_CONFIG, CORTEX_M55_FPU_CONFIG, CORTEX_M55_ETM_CONFIG and SIE300_CONFIG configurations.	Configurable render options section
Description of CPU0_PMC_PRESENT updated with more details	Configurable render options section
XHB500_CONFIG configuration option has been updated to support r0p1-00rel0 version	Configurable render options section

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Change	Location
CPU0_CDEMAPPEDONCP0 .. CPU0_CDEMAPPEDONCP7 configuration options have been added	Configurable render options section
Corrected typo in interface name, XMSTEXPMI* replaced with XMSTEXP*	Main Interconnect Expansion Interfaces section
CPU0_FPU_PRESENT and CPU0_MVE_CONFIG configuration options have been amended with CDE	Configurable render options section
Updated description of figure in System Block Diagram	System Block Diagram section

Copyright © 2020–2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential