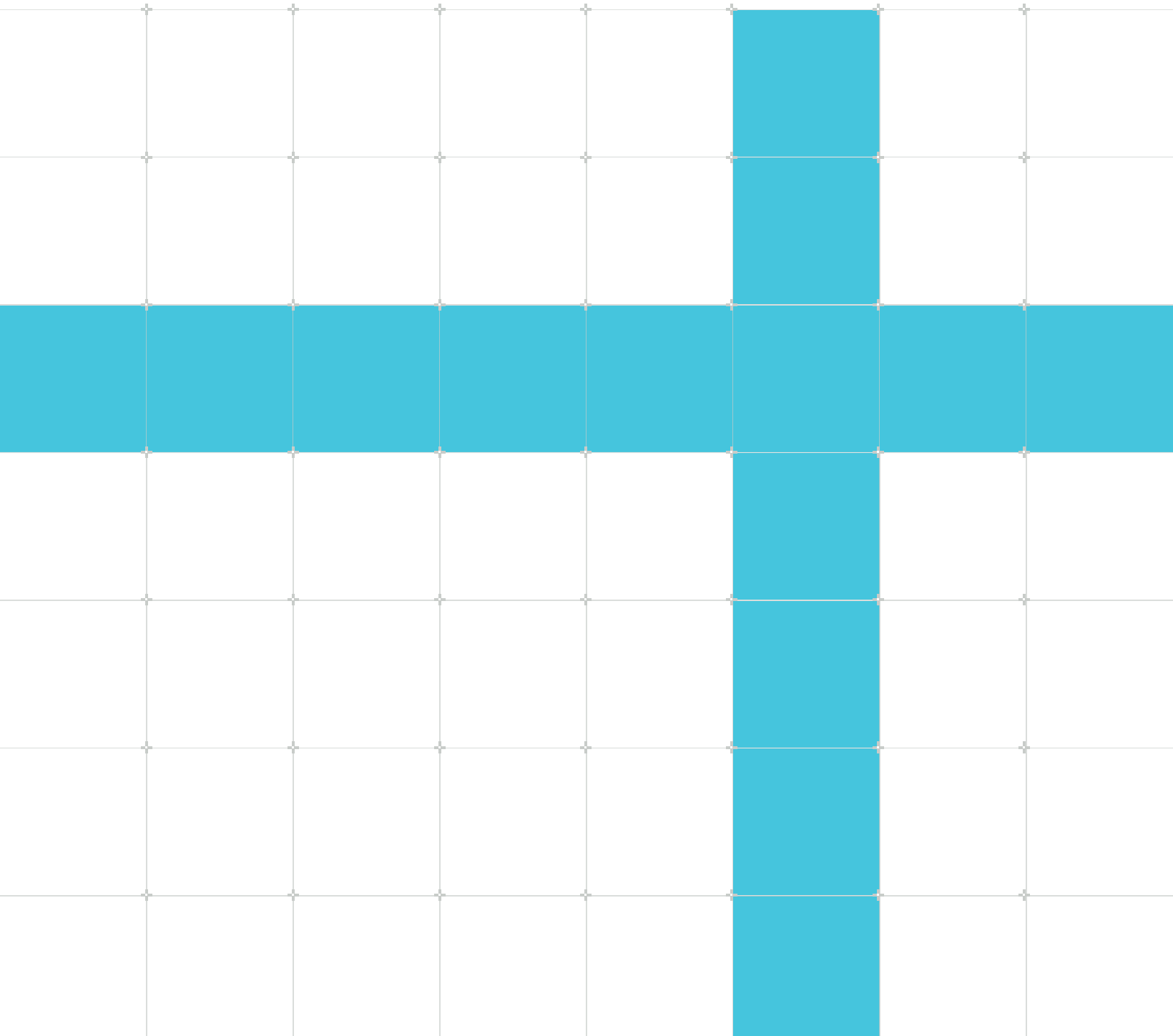
Arm® Corstone™ SSE-320 Example Subsystem

Revision r0p0

Software Programmers Guide

Issue 01
109759_0000_01_en



Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide

This document (109759_0000_01_en) was issued on 2024-10-04. There might be a later issue at https://developer.arm.com/documentation/109759

The product revision is r0p0.

See also: Proprietary notice | Product and document information | Useful resources

Start reading

If you prefer, you can skip to the start of the content.

Intended audience

This book is written for software developers that are developing software for the Arm® SSE-320 Example Subsystem. The documentation does not assume experience of Arm devices.

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.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Feedback

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

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

Contents

1. Overview of SSE-320 Subsystem

A subsystem is self-contained pre-integrated System IP that is designed to perform a specific function, it is intended for use in a SoC.

An Example subsystem showcases the integration of various IP products. The Example subsystem design requires a processor that is licensed and downloaded separately. The Example subsystem can be used as is, or as a starting point for creating a custom subsystem. The reference package includes the components that you would require to create a custom subsystem similar in function the Example subsystem.

Arm® Corstone™ SSE-320 Internet of Things (IoT) Example Subsystem (SSE-320) is designed to be the base of a low-power SoC or the low-power, ambient, hard real-time compute island in a larger SoC.

The Example subsystem design also integrates the following Arm IP, which are optional and can be licensed and downloaded separately:

- Arm® Ethos™-U85
- Arm® CoreLink™ DMA-350

In addition, an example Arm® Mali™-C55 integration on the expansion bus is provided as a reference integration.

SSE-320 supports the *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M* used for deploying secure IoT endpoints. The subsystem was designed with the Arm® Mali™-C55 ISP to target the wide range of IoT low-power applications. Object-recognition, keyword spotting and speech recognition software applications are available for this example subsystem, along with drivers for all the IP used, middleware, cloud integration, security software and the tools to build and debug the firmware.

An example subsystem design follows best practices for EAC, but validation ends at BETA quality (where EAC and BETA refer to Arm standards for the level of testing performed on RTL).

The Example Subsystem is licensed with full modification rights.

The expected partner use model for the Example Subsystem is as follows:

- Render their own specific configuration
- Modify the rendered RTL according to their specific requirements
- Integrate in the rest of the partner SoC
- Fully verify the SoC

SSE-320 integrates the following key Arm components:

- One Arm® Cortex® Cortex-M85 processor core with optional M-Profile Vector Extension (MVE), Floating Point Unit (FPU), Digital Signal Processing (DSP), extensions, caches, Tightly Coupled Memory (TCM)s and Embedded Trace Macrocell (ETM).

- (Optional) One Ethos-U85 neural processing unit (NPU)
- Four Volatile Memory (VM) banks, typically SRAMs.
- Support for two-way striping across two SRAM banks, and support for two-way striping across two DRAMs.
- Memory Protection Controllers (MPC)
- Exclusive Access Monitor (EAM)
- Arm® CoreLink™ NIC-400 System Interconnect
- Implementation Defined Attribution Unit (IDAU)
- Cortex®-M System Design Kit (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) One Arm® CoreLink™ DMA-350 Direct Memory Access Controller (DMAC)
- Lifecycle Manager (LCM)
- Key Management Unit (KMU)
- Security Alarm Manager (SAM)

The previously listed components are integrated to implement SSE-320 with the following features:

- TrustZone® aware system with the system segregated into Secure and Non-secure worlds
- Configurability to allow several features within the system to be included or removed
- Power Control infrastructure with several pre-defined voltage and power domains
- Each switchable power domain has a local power policy control, and coordinates with other power domains through a centralized dependency control or power interfaces. Switchable power domains provide the system with autonomous dynamic power control infrastructure that, while being software configurable, aiming to minimize software interaction.
- Clock control infrastructure that supports high level clock control including dynamic clock gating and provides clock request handshakes to clock generators
- Comprehensive reset generation and control
- A CoreSight SoC-600M 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
- Secured Debug Channel (SDC-600)

The integrator can tailor the final implementation to the target use-cases by using:

- The supported configuration options.

- The sockets for the Host processor, Host GIC, and External Systems.
- The expansion interfaces of SSE-320.

For more information, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*

1.1 Compliance

The Arm SSE-320 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 Protocol Specification*
- *AMBA® AHB Protocol Specification*
- *AMBA® ATB Protocol Specification*
- *AMBA® APB Protocol Specification*
- *AMBA® Low Power Interface Specification*
- *Arm® Corstone™ Reference Systems Architecture Specification Ma2*

This document complements the TRMs or RMs for included components, architecture reference manuals, architecture specifications, protocol specifications, and relevant external standards.

1.2 Product documentation

Each SSE-320 document has an intended audience and is associated with specific tasks in the design flow.

These documents do not reproduce SSE-320 architecture and protocol information. For relevant protocol and architectural information that relates to this product, see Useful resources on page 209.

SSE-320 documentation set includes:

Reference Manual

The RM describes how to integrate and implement the product, and describes the functionality, the effects of functional options on the behavior of the SSE-320. 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 document describes are not relevant.

If you are programming SSE-320 contact:

- The implementer to determine:
  - The build configuration of the implementation
  - What integration, if any, was performed before implementing SSE-320
- The integrator to determine:
  - The signal configuration of the device that you use
  - The available build configuration options
  - How to configure the Register Transfer Level (RTL) with the build configuration options
  - How to integrate SSE-320 into an SoC
  - How to implement SSE-320 into your design
  - The processes to validate the configured design

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

Software Programmers Guide

The software programmers guide contains:

- Descriptions of system architecture and functionality.
- Details of the SSE-320 registers that are available to Software developers. It does not duplicate the register information provided in the component documentation.

1.3 Product revisions

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

r0p0

First release

1.4 Arm architecture

SSE-320 is designed to use the Arm® Cortex-M85 Processor that implements the Armv8.1-M architecture.

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

1.5 Trusted Base System Architecture for Armv8-M

The Trusted Base System Architecture (TBSA-M) is part of the Arm Platform Security Architecture (PSA).

TBSA-M implements best practice security principles when designing systems around Armv8-M processing elements (PEs).

For more details, see *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M*.

SSE-320 partly fulfils the requirements specified within the TBSA-M. However, it implements features that help form the core of a system that complies to the TBSA-M, such as:

- A system architecture that partitions the memory areas into Secure and Non-secure areas.
- 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. For more details, see *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*.
- 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. For more details, see *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*.
- System security control registers and provides expansion control and status interfaces to allow map external memory regions and external peripherals that are shared between both worlds.
- An always on Secure watchdog for a system that integrates SSE-320 to comply with TBSA-M. When expanding the system the integrator must comply with TBSA-M requirements. For example:
    - Root of Trust (RoT)
    - A protected keystore
    - A Secure firmware update mechanism
    - A lifecycle management mechanism, for Secure control of debug, test, and access to provisioned secrets
    - A high-entropy random number generator, for reliable cryptography
    - When adding more managers and subordinates to the system, the memory space continues to obey Trusted and Non-trusted world partitioning

Adherence to TBSA-M requirements helps to create a Secure system, but it does not create a system that mitigates Denial of Service (DoS) attacks. As such, DoS is out of scope of SSE-320. 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

1.6 Interrupt controller architecture

SSE-320 implements the Arm Nested Vectored Interrupt Controller (NVIC) and the Arm External Wakeup Interrupt Controller (EWIC).

For more information, see:

- *Arm® Cortex®-M85 Processor Technical Reference Manual*

1.7 Power Policy Unit architecture

The power domains in SSE-320 are controlled by Power Policy Units (PPUs), which comply with the Arm PPU architecture.

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

Related information

- Policy Power Units

1.8 Advanced Microcontroller Bus architecture

SSE-320 complies with the Advanced Extensible Interface (AXI5) protocol, Advanced High Performance Bus (AHB5) protocol, and Advanced Peripheral Bus (APB4) protocol.

For more details, see:

- *AMBA® AXI Protocol Specification*
- *AMBA® AHB Protocol Specification*
- *AMBA® APB Protocol Specification*

1.9 Document-specific conventions

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

Text

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

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

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

Numbers

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

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

Signals

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

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

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

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

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

- <SIGNAL_NAME>[BIT_RANGE]

Where:

- SIGNAL_NAME is the name of the signal being referred to.
- BIT_RANGE is the bit range being referred to. If BIT_RANGE is omitted, then the text is referring to the whole signal.

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

- DEBUGPADDR[31:2]

Registers

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

- <REGISTER_NAME>.<FIELD_NAME>[BIT_RANGE]

Where:

- REGISTER_NAME is the short name of the register being referred to.

- FIELD_NAME is the name of the field within the register being referred to. If the FIELD_NAME is omitted, then the text is referring to the whole register.
- BIT_RANGE is the bit range, within the field, being referred to. If the BIT_RANGE is omitted, then the text is referring to the whole field or to the whole register if FIELD_NAME is omitted as well. When referring to the bit ranges of a field, the field starts at 0.

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

- SYSVERSION.DESIGNER_ID[1:0]

2. System architecture

The subsystem is divided into functional blocks, which are a combination of Arm IP and the supporting logic around them. The block-based design approach provides flexibility, scalability, and modularity. Additional functionality such as Power Management, Security, and Reliability, Availability and Serviceability (RAS) are relevant across the subsystem.

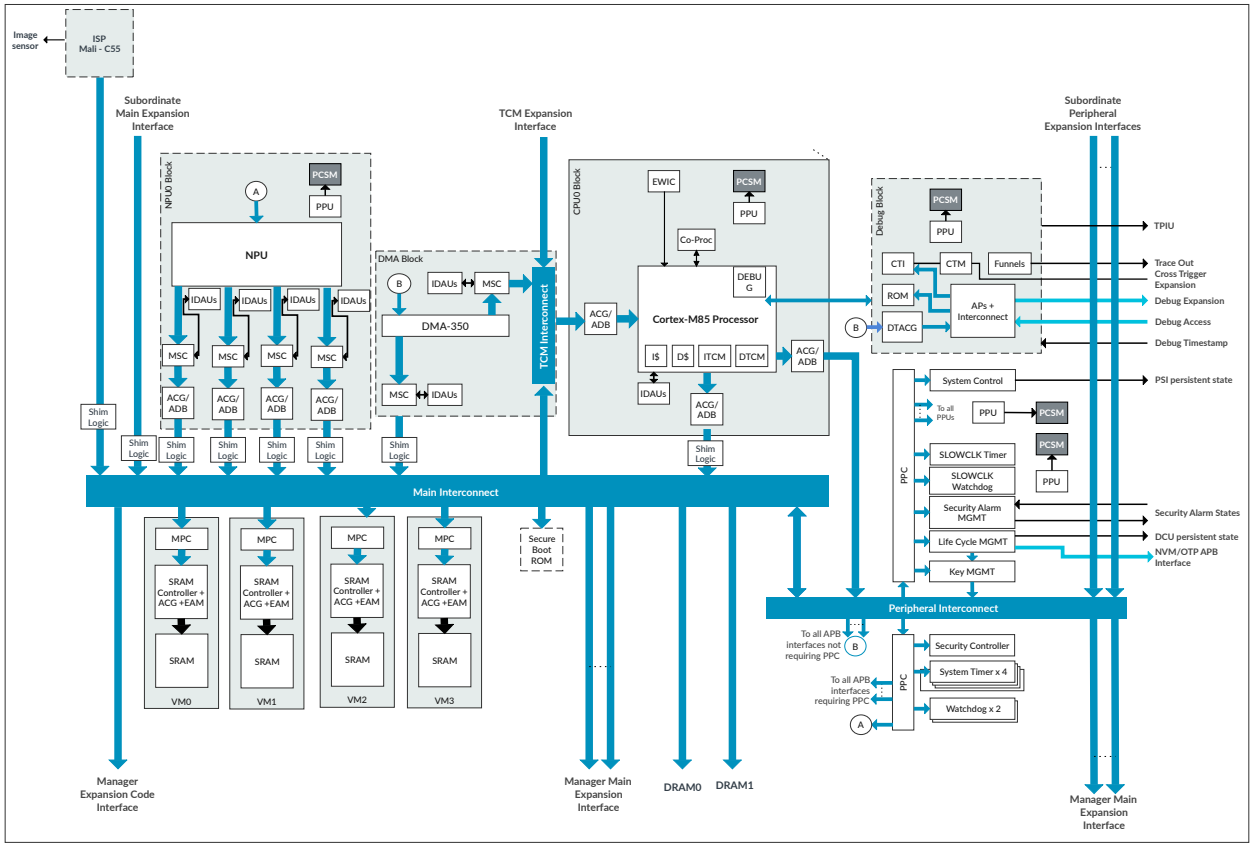
For information about cross-block features, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

Note

The term 'Blocks' is used in this document to group functionalities that are closely related to each other or have common configurability.

The following figure shows a representative system block diagram of the top-level topology of SSE-320.

Figure 2-1: SSE-320 Top-level topology



NPU block

Contains an Ethos-U85 NPU and its associated private infrastructure to integrate the NPU into the system. The NPU is optional. SSE-320 supports one NPU within the system. For more details, see NPU block.

Processor block

Contains a Cortex-M85 processor and its associated private infrastructure to integrate the core into the system. SSE-320 supports one CPU within the system. For more details, see Processor block.

Volatile memory bank block

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

Peripherals Block

Contains all common sets of peripherals expected in the system. This includes the infrastructure required to implement all configure, control, and monitor system states. These include security, clock, reset, and power control. For more details, see Peripherals and security block.

Debug system Block

Contains the debug infrastructures that allows the processor and the system to be debugged securely. For more details, see Debug system block.

DMA block

Contains the DMA controller that allows I/O devices to directly access memory without involvement of the processor. SSE-320 supports one DMA-350. For more details, see DMA block.

System Interconnect Infrastructure block

Contains the Peripheral interconnect, the TCM interconnect and the Main interconnect. For more details, see System interconnect infrastructure block.

ISP

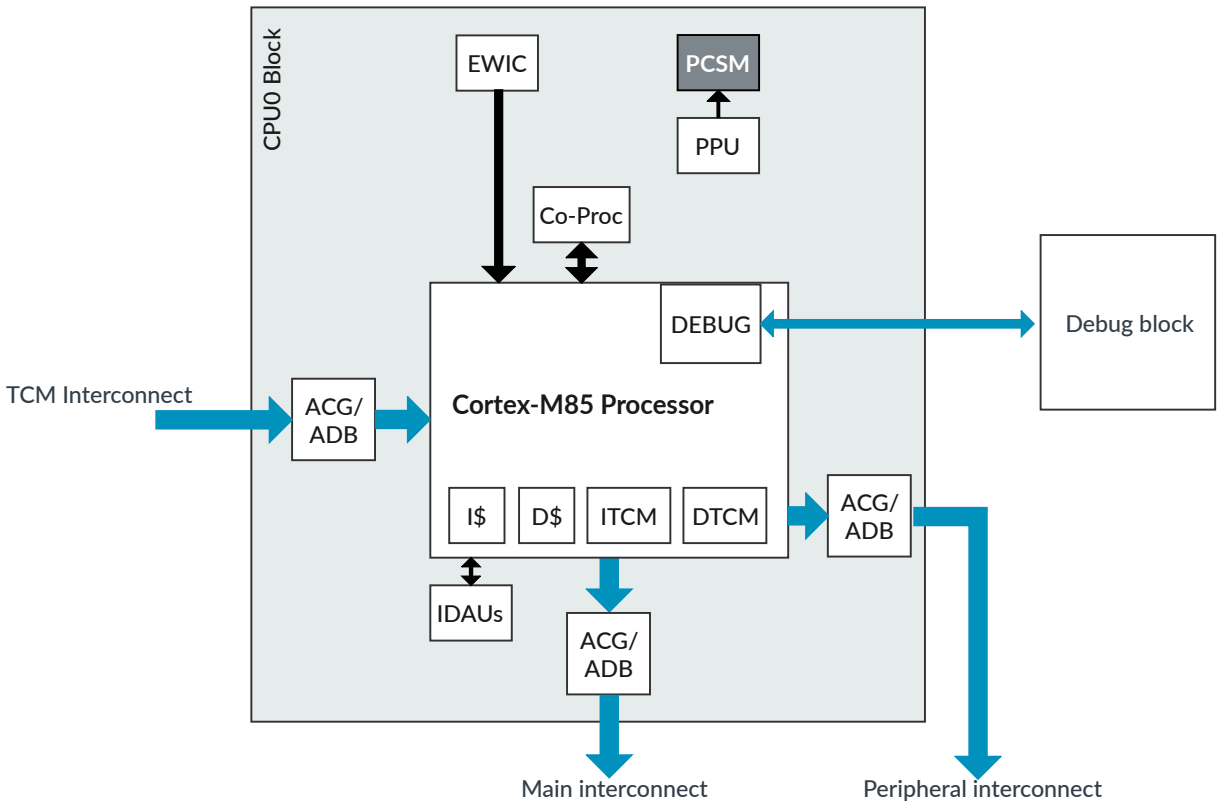
The Arm Mali™-C55 image signal processor (ISP) is designed to support IoT and embedded applications where energy-efficient, high-quality image output is required.

The ISP is optional, and is not integrated into the SSE-320 RTL. However, Arm has verified integration of this ISP and provides optimized integration details. The ISP interfaces with SSE-320 through the MAIN AXI Expansion interconnect. For more details, see Mali™-C55.

2.1 Processor block

SSE-320 supports one Cortex-M85 processor core (referred to as CPU0).

Figure 2-2: Processor block



PCSM

The *Power Control State Machine* (PCSM) controls low-level technology specific power switch and retention such as Multi-threshold CMOS (MTCMOS) sleep control and Retention control. For more details, see Power Policy Units.

PPU

The Policy Power Units control the power domains. For more details, see Power Policy Units.

EWIC

The *External Wakeup Interrupt Controller* (EWIC) wakes up the processor from the OFF state. From the EWIC, the IRQs are routed to the Nested Vector Interrupt Controller (NVIC) in the corresponding processor.

Main interconnect

The Main interconnect provides the highest amount of bus throughput. It is primarily used for code and data accesses that targets memories or high throughput interfaces. For more details, see System interconnect infrastructure block.

Peripheral interconnect

The Peripheral interconnect provides access to lower performance peripherals. For more details, see System interconnect infrastructure block.

TCM interconnect

This interconnect provides access from TCM subordinate interface and from the Main interconnect to the Tightly Coupled Memories (TCM) that are internal to CPU0. For more details, see System interconnect infrastructure block.

For more details see, *Arm® Cortex®-M85 Processor Technical Reference Manual*.

Related information

- BR_CPU0 power modes

2.1.1 EVENT Interfaces

CPU0 has Receive event (RXEV) and Transmit event (TXEV) event interfaces.

In SSE-320:

- TXEV drives the RXEV.
- Events do not wake the processor from its EWIC based low-power state
- Events do not wake the system from HIBERNATION0 state.

SSE-320 does not support the use of WFE for the CPU to enter the DeepSleep state.

2.1.2 Interrupts

SSE-320 provides several components that generate interrupts.

The following components can generate interrupts within the system:

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

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

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

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

When using the EWIC:

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

Table 2-1: CPU0 interrupt map

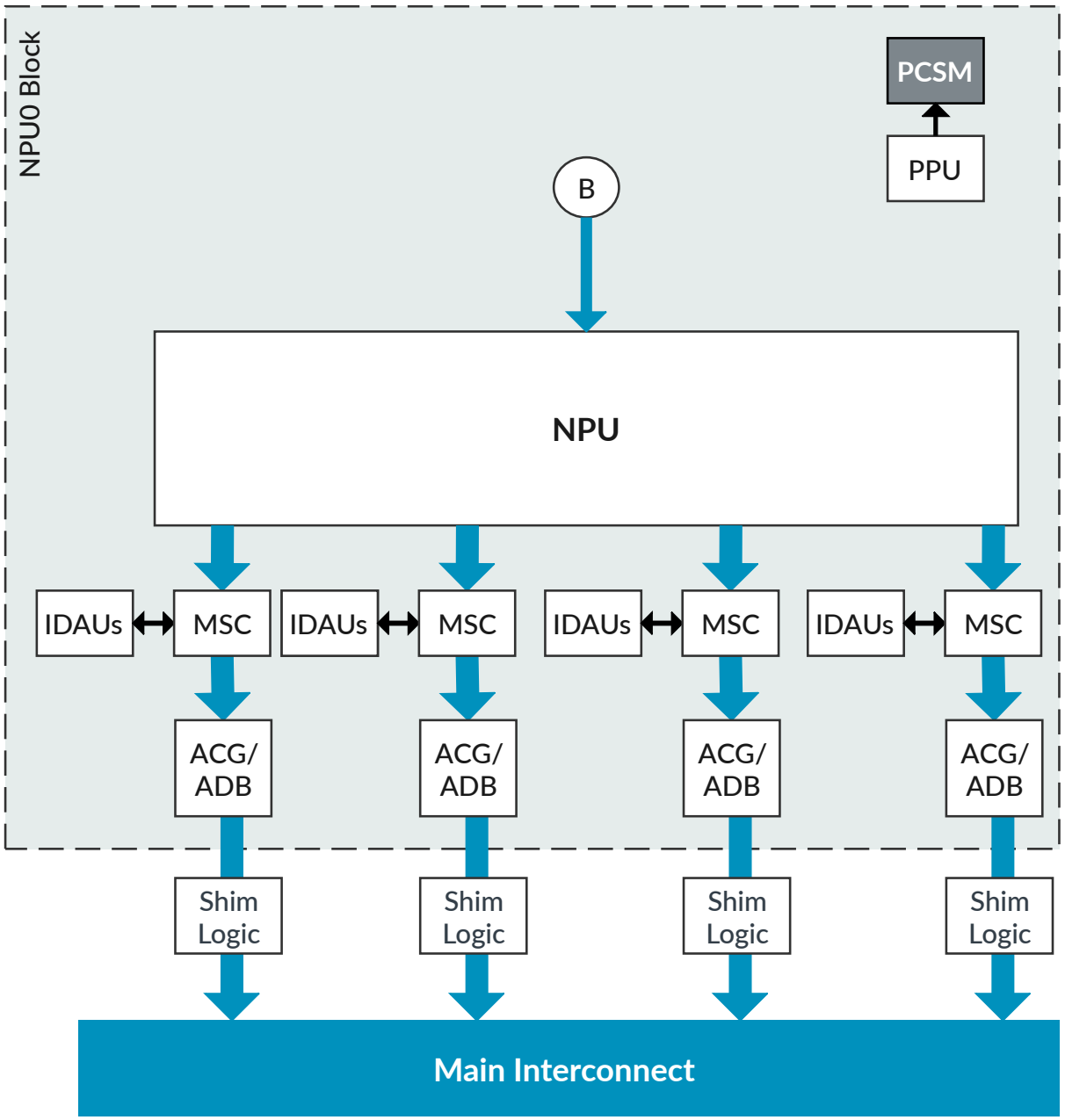
Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
NMI	Combined Secure System Watchdog, SLOWCLK Watchdog, CPU0EXPNMI and NMI of Security Alarm Manager.	Yes
IRQ[0]	Non-secure watchdog reset request	Yes
IRQ[1]	Non-secure watchdog interrupt	Yes
IRQ[2]	SLOWCLK Timer	Yes
IRQ[3]	Timer 0	Yes
IRQ[4]	Timer 1	Yes
IRQ[5]	Timer 2	Yes
IRQ[6]	Reserved	-
IRQ[7]	Reserved	-
IRQ[8]	Reserved	-
IRQ[9]	MPC Combined (Secure)	Yes
IRQ[10]	PPC Combined (Secure)	Yes
IRQ[11]	MSC Combined (Secure)	Yes
IRQ[12]	Bridge Error Combined interrupt (Secure)	Yes
IRQ[13]	Reserved	-
IRQ[14]	PPU combined	Yes
IRQ[15]	SDC-600. Reserved if SDC-600 does not exist	-
IRQ[16]	NPU0	Yes
IRQ[17]	Reserved	-
IRQ[18]	Reserved	-
IRQ[19]	Reserved	-
IRQ[20]	Key Management Unit	Yes
IRQ[23:21]	Reserved	-
IRQ[24]	DMA (Combined Secure)	-
IRQ[25]	DMA (Combined Non-secure)	-
IRQ[26]	DMA (Security violation)	-
IRQ[27]	Timer 3 AON	Yes
IRQ[28]	CPU0CTIIRQ0 (local CPU CTI only)	No

Interrupt input for CPU0	Interrupt source for CPU0	EWIC support
IRQ[29]	CPU0CTIIRQ1 (local CPU CTI only)	No
IRQ[30]	Critical Severity Fault Interrupt of Security Alarm Manager.	Yes
IRQ[31]	Severity Fault Interrupt of Security Alarm Manager.	Yes
IRQ[<CPU0EXPNUMIRQ +31>:32]	CPU0EXPIRQ[CPU0EXPNUMIRQ-1:0].	Yes

2.2 NPU block

SSE-320 supports one Ethos-U85 NPU core.

Figure 2-3: NPU block



MSC

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

Main interconnect

The Main interconnect is an AXI interface that connects all parts of the system.

NPU0 supports different static configurations, but the following External DMA interface configuration value must adhere to:

Table 2-2: Static configurations of Ethos-U85

Configuration	Value for NPU0	Description
External DMA interface	0	Specifies whether the NPU Includes an interface for an external DMA: • Interface not included • interface included

The Ethos-U85 NPU supports CTI based debug halt control when HASCSS = 1.

The NPU can access the system memory map as follows:

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

This memory mapping is enforced using Manager Security Controllers (MSC), IDAUs, and Memory Protection Controllers (MPCs) to separate the Trusted and non-Trusted world for the memories and Peripheral Protection Controllers (PPC) for Privilege and Unprivilege separation and trusted and Non-trusted world separation of peripherals.


Note

There is no protection provided for Privilege or Unprivilege memory access from the NPU. Therefore, Arm recommends that the NPU should be made available to Unprivileged software, only if the risk of Unprivilege software accessing Privilege memory is acceptable.

In SSE-320, the NPU is not allowed to fetch the Command stream from the Non-secure memory alias when the NPU security level is configured as Secure. For more details, see NPUSPPORSL, NPU Secure access security level reset control register.

The ARID signal of NPU AXI interfaces identifies the Command stream of NPU. Secure Command stream transactions to Non-secure memory alias are blocked and responded according to the SECRESPCFG register settings. For more details, see SECRESPCFG, Security Violation Response Configuration register.

When deploying Ethos-U85, if the configuration of Ethos-U85 results in two or four SRAM interfaces, we recommend the following to ensure that the NPU is not limited by system bandwidth:

- VM bank striping is enabled.
- The `VMCFGHASH0` and `VMCFGHASH1` configuration options are used to define Ethos-U85 `CFGSRAMHASH0` and `CFGSRAMHASH1` configurations for SRAM striping.
- If Ethos-U85 is configured with two DRAM interfaces, we recommend that two additional shared DRAM channels are added by the system integrator in the expansion system to support them.

Related information

Volatile memory block on page 29
Peripheral Protection Controllers on page 48
Manager Security Controller on page 42
Memory Protection Controllers on page 50

2.2.1 NPU security mapping

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

When the NPU reset is released by the PPU controlling the NPU power domain PD_NPU0, the NPU reset samples the signals driven by:

- NPUSPPORSL.SP_NPU0PORSL to determine its default security level.
- NPUSPPORPL.SP_NPU0PORPL or NPUNSPORPL.NS_NPU0PORPL to determine its default privilege level.

The NPUSPPORSL and NPUSPPORPL registers are also used to control the PPC, protecting the NPU.

Note

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

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

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

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

1. Read the current Security level from the corresponding system register: initial_security= NPUSPPORSL.SP_NPU0PORSL.

2. If initial_security=S then software reads the privilege to be retained in the target Security level from the corresponding system register: initial_privilege=NPUNSPORPL.NS_NPU0PORPL else skip remaining sequence.
3. Write on the current security alias of the NPU (S) the CMD.power_q_enable register bit of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register.
4. Write on the current security alias of the NPU (S) the RESET.pending_CSL register bit of the NPU to the new Security level (1= Non-secure) while preserving RESET.pending_CPL=initial_privilege.
5. Read on the current security alias of the NPU (S) the STATUS register of the NPU until the bit reset_status no longer yields the value 1.
6. Write the new security level into NPUSPPORSL.SP_NPU0PORSL register bit (1= Non-secure).
7. Write on the current security alias of the NPU (NS) the CMD.power_q_enable register bit of the NPU to the allow the NPU to speculatively power OFF (1 = Power OFF allowed) while preserving the rest of the CMD register.

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

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

1. Read the current Security level from the corresponding system register: initial_security= NPUSPPORSL.SP_NPU0PORSL.
2. If initial_security=NS then software reads the privilege to be retained in the target Security level from the corresponding system register: initial_privilege=NPUSPPORPL.SP_NPU0PORPL else skip remaining sequence.
3. Write on the current security alias of the NPU (NS) the CMD.power_q_enable register bit of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register.
4. Write the new security level into NPUSPPORSL.SP_NPU0PORSL register bit (0 = Secure).
5. Write on the current security alias of the NPU (S) the RESET.pending_CSL register bit of the NPU to the new Security level (0=Secure) while preserving RESET.pending_CPL=initial_privilege.
6. Read on the current security alias of the NPU (S) the STATUS register of the NPU until the bit reset_status no longer yields the value 1.
7. Write on the current security alias of the NPU (S) the CMD.power_q_enable register bit of the NPU to the allow the NPU to speculatively power OFF (1 = Power OFF allowed) while preserving the rest of the CMD register.

Changing privilege level

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

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

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

   Current Security level = Non-secure

   Initial_privilege=NPUNSPORPL.NS_NPU0PORPL

   Current Security level = Secure

   Initial_privilege=NPUSPPORPL.SP_NPU0PORPL

If initial_privilege is the desired level, then skip remaining sequence.

1. Write on the current security alias of the NPU (initial_security) the CMD.power_q_enable register bit of the NPU to prevent the NPU powering OFF (0 = Power OFF denied) while preserving the rest of the CMD register.
2. Write to the current security alias of the NPU (initial_security) the RESET.pending_CPL register bit of the NPU to the new privilege level (0=User; 1=Privileged) while preserving RESET.pending_CSL=initial_security.
3. Read from the current security alias of the NPU (initial_security) the STATUS register of the NPU until the bit reset_status no longer yields the value.
4. Write to the alias corresponding to the current security level of the NPU (initial_security) the new privilege level into:

   Current security level = Non-secure

   NPUNSPORPL.NS_NPU0PORPL

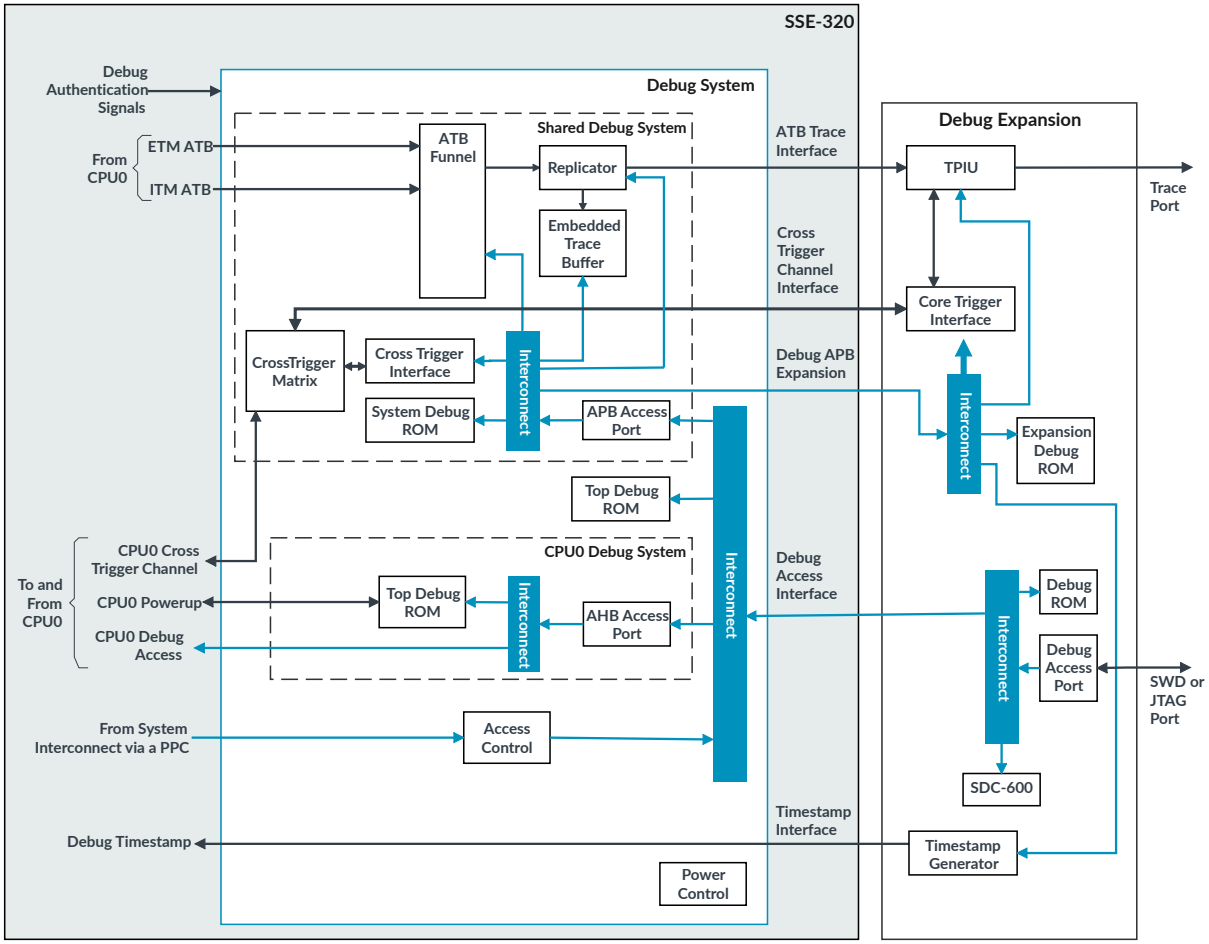
   Current security level = Secure

   NPUSPPORPL.SP_NPU0PORPL
5. Write on the current security alias of the NPU (initial_security) the CMD.power_q_enable register bit of the NPU to the allow the NPU to speculatively power OFF (1 = Power OFF allowed) while preserving the rest of the CMD register.

2.3 Debug system block

SSE-320 supports the optional CoreSight SoC-600M based common debug infrastructure. It is enabled when HASCSS = 1.

Figure 2-4: Debug block



Trace Port Interface Unit

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

Cross Trigger Matrix

The Cross Trigger Matrix (CTM) controls the distribution of channel events.

Cross Trigger Interface

The Cross Trigger Interface (CTI) enables expansion of the cross trigger infrastructure.

ATB Funnel

The ATB trace funnel combine multiple trace streams onto a single ATB bus. For more details, see *Arm® Coresight™ Components Technical Reference Manual*

SDC-600

The Secure Debug Channel (SDC) provides a dedicated channel for authentication between an external debugger and a debug target platform by using an unlocking mechanism. For more details, see *Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual*.

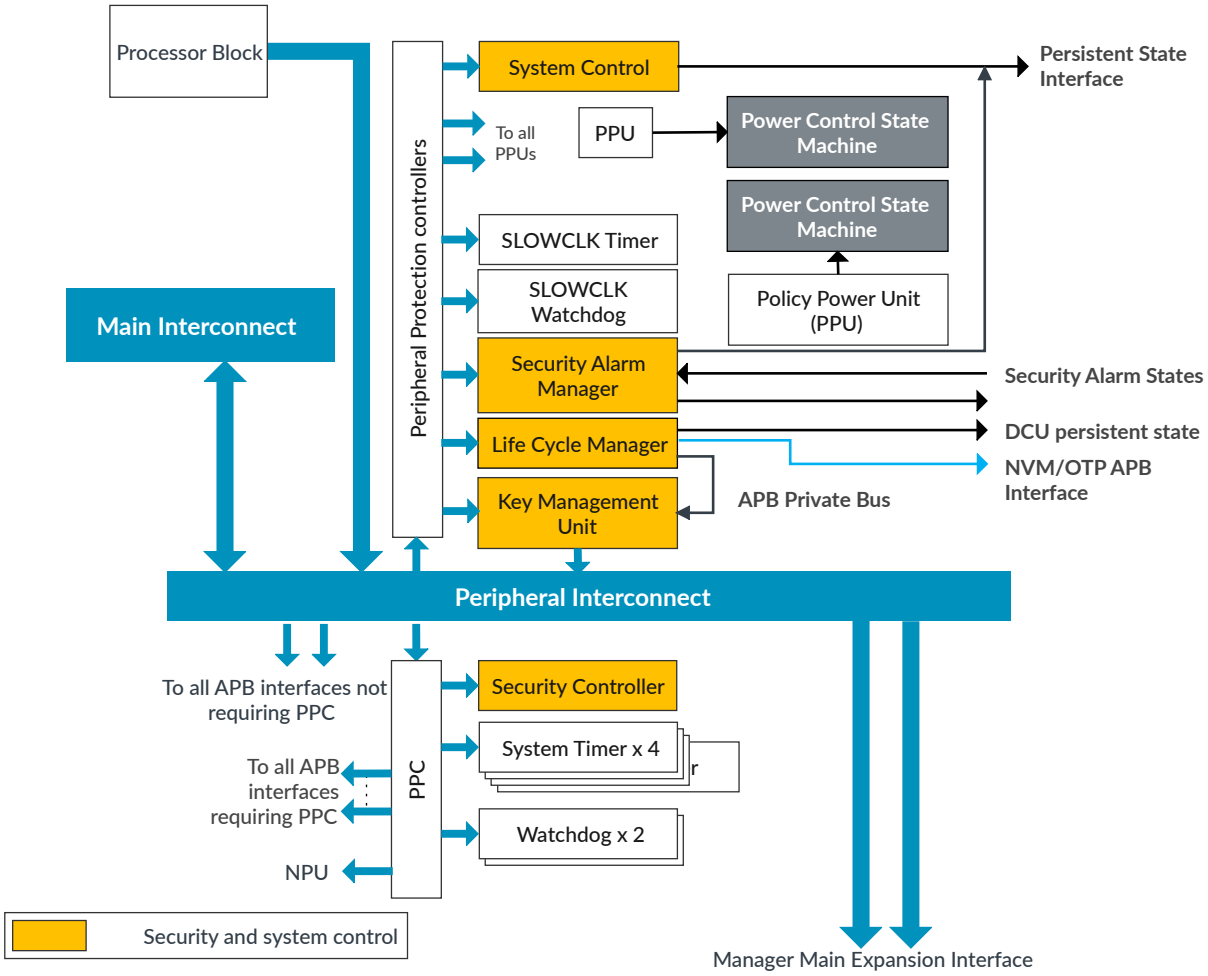
Related information

- Debug Infrastructure
- LCM Debug Control Unit
- BR_DEBUG power modes
- System Control Register Block

2.4 Peripherals and security block

The peripheral block contains the security and lifecycle components, the system timers, and additional components.

Figure 2-5: Peripherals and security block



Peripheral interconnect

The Peripheral interconnect provides access to lower performance peripherals.

Main interconnect

The Main interconnect provides the highest amount of bus throughput. It is primarily used for code and data accesses that targets memories or high throughput interfaces.

For more details of interconnects, see System Interconnect Infrastructure block.

System control

SSE-320 provides several registers in the system to allow various features of the system to be discovered, configured, and controlled. For more details, see System and Security Control.

SLOWCLK timer and SLOWCLK watchdog

The Slow clock AON timer and watchdog are expected to be used when the system is in the HIBERNATION0 System Power state, when potentially only SLOWCLK is available and running, and all other clocks are off. For more details, see SLOWCLK AON Timers.

PPU

The Policy Power Units control the power domains. For more details, see Power Policy Units.

PCSM

The Power Control State Machine (PCSM) controls low-level technology specific power switch and retention such as Multi-threshold CMOS (MTCMOS) sleep control and Retention control. For more details, see Power Policy Units.

PPC

Peripheral Protection controllers (PPC) separate the Privileged and Unprivileged world and separate the Trusted and Non-trusted world of peripherals. For more details, see Peripheral Protection controllers.

System timer

Memory-mapped System Timer with register access through the Peripheral Interconnect. For more details, see Timestamp based Timers.

Security and system control block

PSI persistent state interface

The Persistent State Interface (PSI), also known as the Always-On (AO) module keeps critical state in the embedding system. The PSI provide the means to convey information through a dedicated, sideband signaling to the SoC level. For more details, see Status Interface section in *Arm® Security Alarm Manager Specification*.

Security Alarm states

The PSI interface also includes the SAMSTATUS signals which are the reflection of the SAM event status registers. For more details, see Status Interface section in *Arm® Security Alarm Manager Specification*.

DCU persistent state

The interface to the General-Purpose Persistent Configuration (GPPC). For more details, see *Arm® Lifecycle Manager Specification*.

NVM/OTP APB interface

Lifecycle Manager interface to the One-Time Programmable memory (OTP). For more details, see *Arm® Lifecycle Manager Specification*.

Security Alarm manager

The Security Alarm manager (SAM) provides means to apply the programed response to security events detection. For more details, see *Arm® Security Alarm Manager Specification*.

Lifecycle manager

The Lifecycle Manager (LCM) controls the life cycles state of the system, debug control signals and RoT keys. For more details, see *Arm® Lifecycle Manager Specification*.

Key Management Unit

The Key Management Unit (KMU) is a centralized function that stores symmetric key material for the distributed hardware countermeasures and for the use of software with the different crypto devices. For more details, see *Arm® Key Management Unit Specification*.

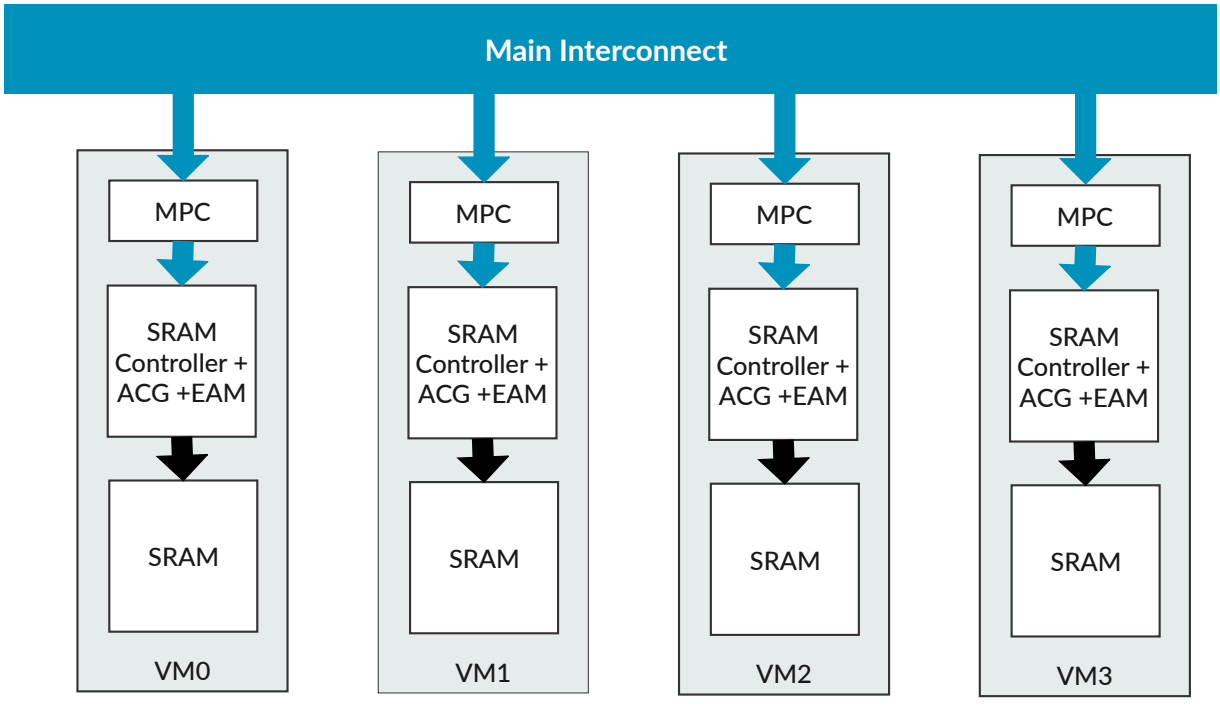
Related information

- Lifecycle Manager
- Security Alarm Manager
- Key Management Unit
- System Interconnect Infrastructure block
- System and Security Control

2.5 Volatile memory block

SSE-320 supports four memory banks, which are VM0, VM1, VM2 and VM3.

Figure 2-6: Volatile memory block



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

- The size of each must be in powers of two.
- The size of both banks is defined using the VMADDRWIDTH configuration option and is equal to $2^{VMADDRWIDTH}$ 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 `0x2100_0000`, which is also aliased to a starting address of `0x3100_0000`.

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

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

For more information about the MPC, see both:

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

In SSE-320, all VMs use the same MPC block size as defined by the configuration parameter VMMPCBLKSIZE. Each MPC is implemented so that out of reset, all memory locations are mapped to the Secure world by default.

SSE-320 supports a power domain for each volatile memory region: PD_VMR0, PD_VMR1, PD_VMR2, and PD_VMR3.

Each power domain contains only the actual volatile memory region of the volatile memory bank with all other logics of the memory banks, including the MPC, residing in the system power domain (PD_SYS). Therefore, any power gating or retention refers only to the memory itself.

For more information about related power control, see Power infrastructure.

All volatile memories run on the SYSSYSCLK signal.

Striping

SSE-320 supports memory striping across banks VM2 and VM3.


Note

SSE-320 also support striping accross DRAMs.

Related information

Striping on page 31
Power domains on page 76

2.5.1 Striping

Striping helps to avoid contentions at the VM bank, which leads to an increase total system bandwidth.

SSE-320 supports memory striping across banks VM2 and VM3. This presents the striped VM banks as one larger SRAM. When striping, each VM is divided into fixed size stripes, and the stripes are interleaved across these banks depending on the hash algorithm used. Ethos-U85 must also be configured to support striping.

The stripe size is defined by the `VMSTRIPEBIT` parameter, the striping mode is defined by the `VMSTRIPEMODE` parameter.

If two DRAM channels are integrated using the expansion system, then we recommend the implementation of DRAM striping across the two DRAM interfaces. using Ethos-U85 CFGEXTHASH0 configuration, and that a matching Hash algorithm is implemented in the expansion system for the DRAM channels.

VM striping affects how Memory Protection Controllers are programmed and the power control of the VM banks.

DRAM striping

SSE-320 supports striping across DRAMs. The configuration of Ethos-U85 results in two DRAM interfaces, DRAM stripping across two DRAM channels and a matching Hash algorithm is implemented in the expansion system.

SSE-320 implements two DRAM channels with the following characteristics:

- DRAM stripe size is 256Bytes
- The 256Bytes address space is aliased in the both the Secure world and Non-secure world and protection is provided via MPCs.

  The Non-secure world alias starts from `0x6000_0000` and the Secure world alias starts at `0x7000_0000`.

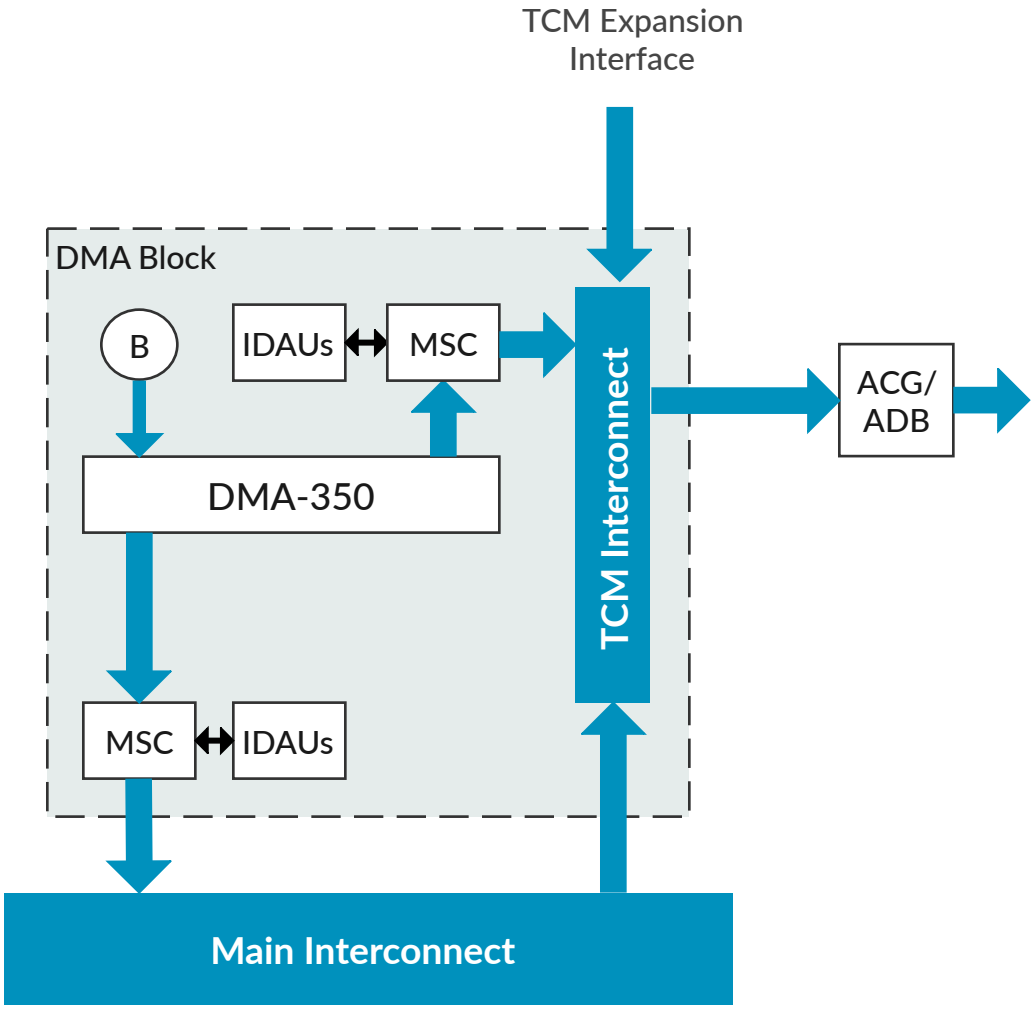
Related information

Volatile Memory Region on page 110

2.6 DMA block

SSE-320 supports an optional DMA-350.

Figure 2-7: DMA block



IDAU

The Implementation Defined Attribution Unit (IDAU) enables the security attribution of memory addresses.

MSC

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

TCM Interconnect

The TCM interface provides system access only to Tightly Coupled Memories (TCM) that are internal to CPU0. The protocol of this interface is AMBA AXI-5. For more details, see TCM Subordinate interface section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

The DMA is enabled by the NUMDMA configuration value:

- NUMDMA = 0, the DMA IP and its associated integration logic does not exist within the system.
- NUMDMA = 1 the DMA IP and its associated integration logic exists within the system.

When NUMDMA = 1, the DMA can support any static configuration, however you must adhere to following configurations in the table:

Table 2-3: DMA required static configurations

Configuration	Value for DMA	Description
AXI5_M1_PRESENT	1 when using DMA-350	Specifies whether an additional manager port is present. When set, the m1 manager port is present • 0: Interface not included • 1: Interface included
SECEXT_PRESENT	1	Specifies where TrustZone security is supported • 0: Security Extensions are not implemented • 1: Security Extensions are implemented
NUM_CHANNELS	>= (NUMCPU+1)*2	Specifies the number of DMA channels

We recommend that DMA channels are not shared between security and privilege levels and that each security and privilege level that requires DMA support has a dedicated channel configured in the DMA.

The DMA can access the system memory map as follows:

- M0 Interface: All of the system memory. However, when using DMA-350, the CPU0 S-AHB Instruction TCM and CPU0 S-AHB Data TCM areas are not accessible from this interface.
- M1 Interface: Only CPU0 S-AHB Instruction TCM and CPU0 S-AHB Data TCM areas.

Security mapping is enforced using:

- Manager Security Controllers (MSC), IDAUs and MPCs for Secure and Non-secure world separation of memories.
- MSC, IDAU and Peripheral Protection Controllers (PPC) for Privileged and Unprivileged separation and Secure and Non-secure world separation of peripherals.


Note

There is no protection provided for Privileged or Unprivileged memory access. Therefore, we recommend that only Privileged software is allowed to program the DMA.

The DMA can be configured with 0-32 trigger in and trigger out interfaces. Other configuration parameters may also create DMA Interfaces, such as GPO, Stream interfaces, and others. These other interfaces are routed directly to the top level of the subsystem and are in the same power and clock domain as the DMA.

DMA-350 supports automatic booting capability that allows the DMA to optionally run a sequence of DMA operation after reset without software setup. We recommend that this functionality is not used and the DMA's boot_En signal is tied LOW. However, if you choose to utilize this capability, you must ensure that the DMA operations performed are in line with the security requirements for your system.

In addition, DMA's boot_En configuration signal should also be forced LOW:

- When LCMRSTREQ is asserted to block any DMA from executing any commands after DMA reset.
- During the provisioning of confidential assets using the LCM Secure Provisioning feature.

For more information on the DMA configuration, see *Arm® CoreLink™ DMA-350 Controller Technical Reference Manual* and *Arm® CoreLink™ DMA-350 Controller Configuration and Integration Manual*.

Related information

- NA

2.7 System interconnect infrastructure block

The System interconnect infrastructure provides a bus infrastructure that transfers memory mapped access from bus managers to subordinates in the system. The system interconnect infrastructure uses the Arm® CoreLink™ NIC-400 interconnect.

SSE-320 defines the key interconnects that form the System interconnect:

Main interconnect

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

- In-subsystem volatile memories, that is VM0, VM1, VM2, and VM3
- Flash, DRAM controllers, or ROM that reside outside the system
- Other high throughput devices

Peripheral interconnect

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

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

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

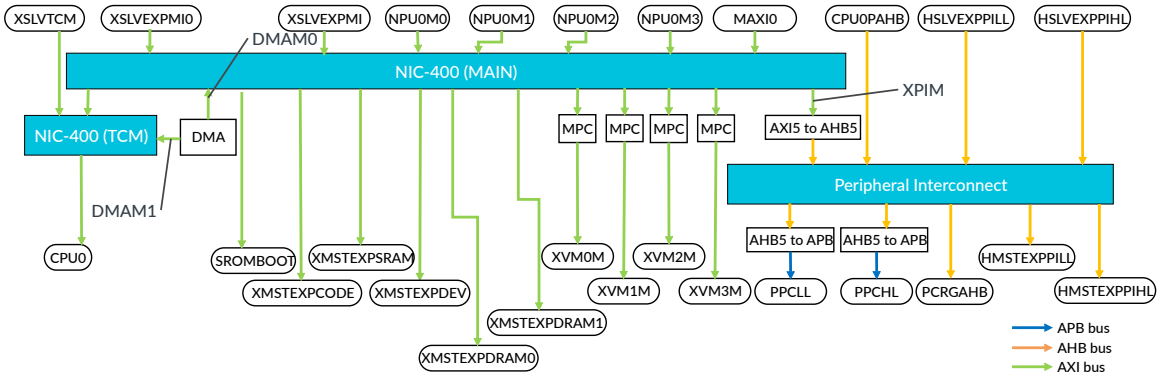
TCM interconnect

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

The System interconnect resides in PD_SYS. The interconnect runs primarily on SYSSYSCLK. The power domain also has its local derived Cold reset input and warm resets. For more information, see Clocking Infrastructure and Reset Distribution.

The following figure shows the connections between the interconnect components.

Figure 2-8: Interconnect connections



The NIC interconnect, which forms the System interconnect infrastructure is not a full crossbar, and certain managers can only access specific subordinates as shown in following table:

Table 2-4: NIC interface connections

Manager Interface	XSLVEXPMI0	XSLVEXPMI1	MAXI0	NPU0M0	NPU0M1	NPU0M2	NPU0M3	XSLVTCM	DMAM0	DMAM1
XMSTEXPCODE	Y	Y	Y	Y	Y	N	N	N	Y	N
XMSTEXPSRAM	Y	Y	Y	Y	Y	N	N	N	Y	N
XMSTEXPDEV	Y	Y	Y	Y	Y	N	N	N	Y	N
XVM0M	Y	Y	Y	Y	Y	N	N	N	Y	N
XVM1M	Y	Y	Y	Y	Y	N	N	N	Y	N
XVM2M	Y	Y	Y	Y	N	N	N	N	Y	N
XVM3M	Y	Y	Y	N	Y	N	N	N	Y	N
XPIM	Y	Y	Y	Y	Y	N	N	N	Y	N
TCMM	Y	Y	N	Y	Y	N	N	N	N	N
SROMBOOT	N	N	Y	N	N	N	N	N	Y	N
XMSTEXPDRAM0	Y	Y	Y	Y	Y	Y	Y	N	Y	N
XMSTEXPDRAM1	Y	Y	Y	Y	Y	Y	Y	N	Y	N
CPU0	N	N	N	N	N	N	N	Y	N	Y

Table 2-5: Peripheral interconnect connections

Manager Interface	CPU0PAHB	HSLVEXPPILL	HSLVEXPPIHL	KMU*	XHB-500
PPCLL	Y	Y	Y	Y	Y
PPCHL	Y	Y	N	Y	Y
PCRGAHB	N	N	Y	Y	Y
HMEXPLL	Y	Y	N	Y	N
HMEXPHL	Y	N	Y	Y	Y
ISP**	Y	Y	N	N	Y

* The KMU interface provides communication from the peripheral interface to the KMU component.

** The ISP interface is an external peripheral connected on XSLVEXPMI0 AXI interface to the ISP, which has a feedback connection from the Peripheral interconnect.

Table 2-6: NIC interface address ranges

Manager interface	Address ranges
XMSTEXPCODE	0x0100_0000 - 0x09FF_FFFF 0x1200_0000 - 0x19FF_FFFF
TCMM	0x0A00_0000 - 0x0AFF_FFFF 0x1A00_0000 - 0x1AFF_FFFF 0x2400_0000 - 0x24FF_FFFF 0x3400_0000 - 0x34FF_FFFF
XMSTEXPSRAM	0x2800_0000 - 0x2FFF_FFFF 0x3800_0000 - 0x3FFF_FFFF
XMSTEXPDRAM0	0x6000_0000 - 0x67FF_FFFF (NS) 0x7000_0000 - 0x77FF_FFFF(S)
XMSTEXPDRAM1	0x6800_0000 - 0x6FFF_FFFF (NS) 0x7800_0000 - 0x7FFF_FFFF(S)
XMSTEXPDEV	0xA000_0000 - 0xDFFF_FFFF 0x8000_0000 - 0x9FFF_FFFF
XPIM	0x4800_0000 - 0x4FFF_FFFF 0x5800_0000 - 0x5FFF_FFFF 0x5004_0000 - 0x500F_FFFF
XVM0M	0x2100_0000 - 0x211F_FFFF 0x3100_0000 - 0x311F_FFFF

Manager interface	Address ranges
XVM1M	0x2120_0000 - 0x213F_FFFF 0x3120_0000 - 0x313F_FFFF
XVM2M	0x2140_0000 - 0x215F_FFFF 0x3140_0000 - 0x315F_FFFF
XVM3M	0x2160_0000 - 0x217F_FFFF 0x3160_0000 - 0x317F_FFFF
CPU0SAHB	0x0A00_0000 - 0x0AFF_FFFF 0x1A00_0000 - 0x1AFF_FFFF 0x2400_0000 - 0x24FF_FFFF 0x3400_0000 - 0x34FF_FFFF
HMSTEXPPILL	0x4010_0000 - 0x47FF_FFFF 0x5010_0000 - 0x57FF_FFFF 0xE020_0000 - 0xEFFF_FFFF 0xF020_0000 - 0xFFFF_FFFF
HMSTEXPPIHL	0x4810_0000 - 0x481F_FFFF 0x5810_0000 - 0x581F_FFFF 0x4830_0000 - 0x4FFF_FFFF 0x5830_0000 - 0x5FFF_FFFF 0x4824_0000 - 0x482F_FFFF 0x5824_0000 - 0x582F_FFFF

Manager interface	Address ranges
PPCLL	0x1100_0000 - 0x11FF_FFFF 0x4000_2000 - 0x4000_3FFF 0x4008_0000 - 0x4008_0FFF 0x4800_0000 - 0x4800_0FFF 0x4800_1000 - 0x4800_1FFF 0x4800_2000 - 0x4800_2FFF 0x4800_3000 - 0x4800_3FFF 0x4804_0000 - 0x4804_0FFF 0x4804_1000 - 0x4804_1FFF 0x5000_2000 - 0x5000_3FFF 0x5008_0000 - 0x5008_0FFF 0x5800_0000 - 0x5800_0FFF 0x5800_1000 - 0x5800_1FFF 0x5800_2000 - 0x5800_2FFF 0x5800_3000 - 0x5800_3FFF 0x5804_0000 - 0x5804_0FFF 0x5804_1000 - 0x5804_1FFF 0x5008_3000 - 0x5008_3FFF 0x5008_4000 - 0x5008_4FFF 0x5008_5000 - 0x5008_5FFF 0x5008_6000 - 0x5008_6FFF

Manager interface	Address ranges
PPCLL (cont)	0xE010_0000 - 0xE010_0FFF 0xF010_0000 - 0xF010_0FFF 0xE010_2000 - 0xE010_5FFF 0xF010_2000 - 0xF010_5FFF 0xE010_4000 - 0xE010_5FFF 0xF010_4000 - 0xF010_5FFF 0xE010_6000 - 0xE010_7FFF 0xF010_6000 - 0xF010_7FFF
PPCHL	0x4008_1000 - 0x4008_FFFF 0x5008_1000 - 0x5008_2FFF 0x5009_E000 - 0x5009_EFFF 0x500A_0000 - 0x500A_FFFF 0x5804_2000 - 0x5804_2FFF
PCRGAHB	0x4802_0000 - 0x4802_0FFF 0x4802_1000 – 0x4802_EFFF 0x4802_F000 - 0x4802_FFFF 0x5802_0000 - 0x5802_0FFF 0x5802_1000 - 0x5802_1FFF 0x5802_2000 - 0x5802_2FFF 0x5802_3000 - 0x5802_3FFF 0x5802_8000 - 0x5802_8FFF 0x5802_9000 - 0x5802_9FFF 0x5802_A000 - 0x5802_AFFF 0x5802_E000 - 0x5802_EFFF 0x5802_F000 - 0x5802_FFFF
ISP	0x4820_0000 - 0x4823_FFFF 0x5820_0000 - 0x5823_FFFF

Related information

- Peripherals and security block
- Striping

2.7.1 ACC_WAIT control

SSE-320 provides a set of controls to let you add access control gates or block access through the system interconnect.

These controls:

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

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

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

2.8 ISP

Arm Mali™-C55 is a single and multi camera, multi-exposure High Dynamic Range (HDR) ISP for the consumer and surveillance market, supporting both single sensor and multi-sensor applications.

The ISP Software provides the following functionality:

- Single-camera support: the code is designed to be used with only one instance of sensor and lens driver.
- Multi-Calibration set: the calibration set can be changed dynamically as per the requirement.
- Multi-exposure HDR support: the software supports up to 2:1 multi exposure.
- Output can be sent directly to the ML accelerator (Ethos-U85 integrated into SSE-320).

The following algorithms are included in the standard driver delivery:

- AWB
- AE
- AF

- Gamma Contrast
- Noise Reduction Control (NRC):
  - Sharpening control.
  - Independent driver core which enables the driver to be platform agnostic and be compiled under different target platforms by providing a proper BSP layer.
- Dedicated channel which helps to update the internal parameters of the driver in real time without re-compilation, therefore speeding up the tuning and debugging procedure.
- V4L2 compatible layer for the Linux version of the driver ,and the reference example for the integration of the driver into the external framework.

Mali™-C55 image signal processor supports applications that require high-quality image output by providing enhancements of the following functions:

- Arm Iridix™ local tone mapping engine
- Arm Temper temporal noise reduction
- Arm Sinter spatial noise reduction
- High Dynamic Range (HDR) sensors support

Iridix local tone mapping

The process of applying intensity transformations to images to achieve better visualization by using information gathered from local regions within images. Iridix defines these local regions in an image as grids with equal sizes. It extracts statistics from each grid to apply the collected statistics to the corresponding local regions in the image. Mali™-C55 smooths each local tone mapping algorithm by smoothing smooths each local tone curve. Therefore, enabling a more natural fall-off around bright light sources.

Temper

A temporal noise reduction algorithm that improves the quality of images in low light conditions by combining consecutive frames. Mali™-C55 improves the image quality with updated noise reduction algorithms.

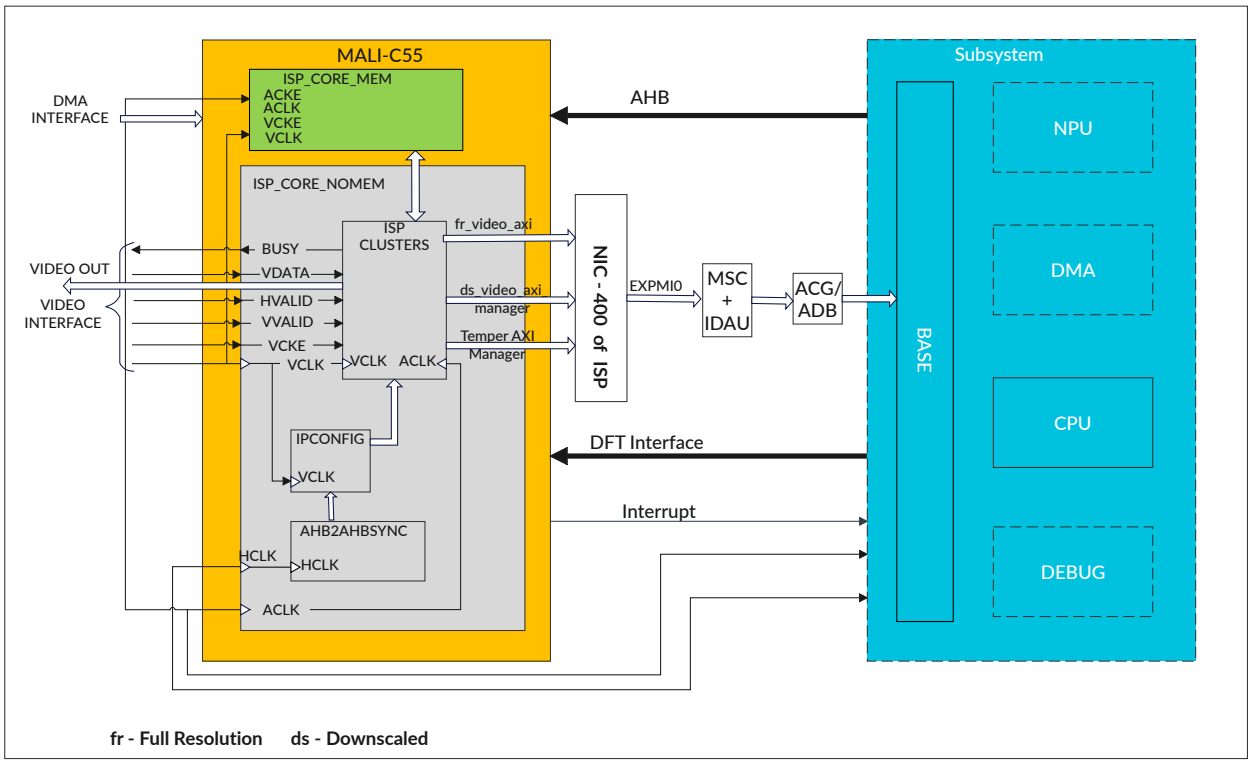
Sinter 2.6

An improved spatial noise reduction technique that improves the detail and noise balance in color channels.

The Temper and Sinter functional blocks were designed to work together for a significantly better image quality by sharing information between the modules to apply stronger noise reduction in various regions.

The Temper and Sinter block order is switched in the pipeline. This way the input motion mask from Temper improves the overall motion-adaptive noise-reduction performance, while providing per-plane noise profiling.

Figure 2-9: ISP integrated into SSE-320 subsystem



For integration specifics, see ISP integration.

For more information about Mali™-C55, see Mali-C55 arm Developer page.

Manager Security Controller

M-class processors (such as Cortex-M85), require a full security match, the security of the transactions must be overwritten by MSC based on the IDAU. MSC enables Managers to send Secure and Non-secure transactions as well based on the system memory map. Areas can be protected by further security components (MPC, PPC), if hardware protection is needed.

MSC interrupts

The MSC has a level-sensitive interrupt output, irq, that can indicate the occurrence of a security violation or a faulty security attribute conversion.

Implementation Defined Attribution Unit (IDAU)

The IDAU is used to indicate to the processor if a particular memory address is Secure, Non-secure Callable (NSC), or Non-secure, and provides the region number within which the memory address resides.

It can also mark a memory region to be exempted from security checking, for example, a ROM table. The IDAU interface in general is processor specific. However, there is a high similarity between the IDAU interfaces on different Cortex-M processors.

If the Armv8-M Security Extension is included in the processor, then the security state of a memory region is controlled by a combination of the internal Secure Attribution Unit (SAU) or an external Implementation Defined Attribution Unit (IDAU).

Designers can use an IDAU to define a fixed memory map and use a SAU to override the security attributes for some parts of the memory. A simple use could be to use the IDAU to split memory into 500Mb chunks of alternating Secure and Non-secure memory.

The designer of a microcontroller or SoC device divides the memory spaces into Secure and Non-secure areas. Software defines some of the regions using the Secure Attribution Unit (SAU), or by device-specific controller logic that is connected to a special Implementation Defined Attribution Unit (IDAU) interface on the processor. Memory partitioning is also used to define peripherals as Secure or Non-secure.

2.8.1 ISP integration

Arm has verified the integration of Mali™-C55 with SSE-320 and provides these recommendations based upon this effort. This information is in addition to the product documention.


Note

The Mali™-C55 documentation is only available to licensees of the product.

Mali™-C55 register map

The ISP registers are in the Peripheral Expansion Interface range.

Table 2-7: The base address for the ISP

Range	Security
0x4820_0000 - 0x4823_FFFF	NS
0x5820_0000 - 0x5823_FFFF	S

Mali™-C55 memory map

The address range for ISP interconnect is `0x0000_0000 – 0xAFFF_FFFF` for all three AXI interfaces. Only one AXI manager at a time can transfer data to the subsystem.

Mali™-C55 can access the following address range for Main interconnect.

Table 2-8: Mali™-C55 address map

Subordinate name	Address range
XMSTEXPCODE	0x0100_0000 - 0x09F_FFFFF 0x1200_0000 - 0x19FF_FFFF

Subordinate name	Address range
XMSTEXPSRAM	0x2800_0000 - 0x2FFF_FFFF 0x3800_0000 - 0x3FFF_FFFF
XMSTEXPDEV	0x8000_0000 - 0x9FFF_FFFF 0xA000_0000 - 0xDFFF_FFFF
XVM0M	0x2100_0000 - 0x211F_FFFF 0x3100_0000 - 0x311F_FFFF
XVM1M	0x2120_0000 - 0x213F_FFFF 0x3120_0000 - 0x313F_FFFF
XVM2M	0x2140_0000 - 0x215F_FFFF 0x3140_0000 - 0x315F_FFFF
XVM3M	0x2160_0000 - 0x217F_FFFF 0x3160_0000 - 0x317F_FFFF
XPIM	0x4800_0000 - 0x4FFF_FFFF 0x5004_0000 - 0x500F_FFFF 0x5800_0000 - 0x5FFF_FFFF
XMSTEXPDRAM0	0x6000_0000 - 0x67FF_FFFF 0x7000_0000 - 0x77FF_FFFF
XMSTEXPDRAM1	0x6800_0000 - 0x6FFF_FFFF 0x7800_0000 - 0x7FFF_FFFF
TCMM	0x0A00_0000 - 0x0AFF_FFFF 0x1A00_0000 - 0x1AFF_FFFF 0x2400_0000 - 0x24FF_FFFF 0x3400_0000 - 0x34FF_FFFF

Verified Mali™-C55 configuration parameters

Arm has verified the integration using the following proven values of the rendering parameters.

Table 2-9: Proven values of configuration parameters

Parameter name	Description	Default value	Proven values
AXI_RFIFO_T_AW	Address width of the Temper Direct Memory Access (DMA) reader First In First Out (FIFO). The FIFO depth is $2^{AXI\_RFIFO\_T\_AW}$.	7	7
AXI_WFIFO_T_AW	Address width of the Temper DMA writer FIFO. The FIFO depth is $2^{AXI\_WFIFO\_T\_AW}$.	7	7
AXI_WFIFO_Y_AW	Address width of the Y DMA writer FIFOs. The FIFO depth is $2^{AXI\_WFIFO\_Y\_AW}$.	9	9
AXI_WFIFO_UV_AW	Address width of the UV channel DMA writer FIFOs. The FIFO depth is $2^{AXI\_WFIFO\_UV\_AW}$.	8	8

Parameter name	Description	Default value	Proven values
PONG_CONFIG_FITTED	0 - Removes pong configuration space from the design 1 - Instantiates pong configuration space in the design	0	0
WDR_FITTED	0 - Removes Wide Dynamic Range (WDR) frame stitch, offset, and gain from the design. 1 - Instantiates WDR frame stitch, offset, and gain in the design	0	0
COMPRESSION_FITTED	0 - Removes Temper compression logic from the design. 1 - Instantiates Temper compression logic in the design.	0	0
TEMPER_FITTED	0 - Removes Temper, DMA, or merge from the design. 1 - Instantiates Temper, DMA, or merge in the design.	0	0
SINTER_LITE	0 - Instantiates standard Sinter version in the design. 1 - Instantiates lite Sinter version in the design.	0	0
SINTER_FITTED	0 - Removes Sinter from the design. 1 - Instantiates Sinter in the design.	1	1
IRIDIx_LTM_FITTED	0 - Removes Iridix™ local tone-mapping logic from the design. 1 - Instantiates Iridix™ local tone-mapping logic in the design.	1	1
IRIDIx_GTM_FITTED	0 -Removes Iridix™ global tone-mapping logic from the design. 1 - Instantiates Iridix™ global tone-mapping logic in the design.	1	0,1
IRIDIx_CTx_NUMBER	Number of contexts for Iridix™.	0	0
CNR_FITTED	0 - Removes Color Noise Reduction (CNR) and the square and square root for CNR from the design. 1 - Instantiated CNR and the square and square root for CNR in the design	1	1
FRSCALER_FITTED	0 - Removes full resolution pipe RGB scaler from the design. 1 - Instantiates full resolution pipe RGB scaler in the design	0	0
DSPIPE_FITTED	0 - Removes downscaled pipe from the design. 1 - Instantiated downscaled pipe in the design.	1	1
SCALER_COEF_SETS	Number of scaler coefficient sets.	8	8
OUTPUT_DMA	Enable the output with DMA format.	1	1
MAX_LINE_LENGTH_FR	Maximum full resolution frame width that the ISP must process.	1920	1920
MAX_LINE_LENGTH_DS	Maximum downscaled frame width that the ISP must process.	n/a	n/a

Mali™-C55 configuration examples

Arm has verified the following examples of SSE-320 feature configurations. You can configure the RTL of SSE-320 with any supported rendering values.

Config1

Removes output scalers, multiple contexts for multi-sensor support, frame stitch, temporal noise reduction, spatial noise reduction, local tone mapping, and CNR. Includes output DMA.

Config2

Adds spatial noise reduction and one output scaler to config1.

Config3

Adds local tone mapping. This configuration is a reasonable configuration and PPA trade off to config2.

Config4

Adds support for two sensors to config3.

Table 2-10: Proven SSE-320 feature configurations

Feature	Parameter name	Config1	Config2	Config3	Config4
Output scaler	FRSCALER_FITTED = column value to the right DSPIPE_FITTED = 0	0	1	1	1
Multiple contexts for multi-sensor support (Not verified for this release)	IRIDIx_CTx_NUMBER = column value to the right PONG_CONFIG_FITTED = 0 unless column value to the right is > 1	0	0	0	2 sensor support
Frame stitch	WDR_FITTED	0	0	0	0
Temporal noise reduction	TEMPER_FITTED	0	0	0	0
Spatial noise reduction	SINTER_LITE	0	1	1	1
Local tone mapping	IRIDIx_LTM_FITTED	0	0	1	1
CNR	CNR_FITTED	0	0	0	0
Output DMA	OUTPUT_DMA	1	1	1	1

Mali™-C55 clock domain integration

The ISP is fully synchronous. You must complete any Clock Domain Crossing (CDC) for the sensor interface outside the ISP. The application wrapper has an example of how CDC can be implemented. The application wrapper example is included in the deliverable in the `malic55/logical/malic55_application_integration` directory.

The ISP has the following clock and reset domains:

- Video clock and reset domain: vclk and rstn
- AXI clock and reset domain: aclk and aresetn
- AHB5 clock and reset domain: hclk and hresetn

AXI and AHB5 clock and reset are controlled by the expansion external power integration kit and clock and reset control module.

Mali™-C55 power domain integration

Mali™-C55 is placed in the PD_SYS power domain and ISP power on or power off is controlled by the software through PDCM_PD_SYS_SENSE register.

If PD_SYS_ON bit is set to HIGH, it cannot shut down. If it is not set to HIGH, it is permitted to shut down the PD_SYS domain. PDCM_PD_SYS_SENSE prevents the PD_SYS shutdown.

Mali™-C55 PPA

For details, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

Additional reading

- Mali-C55 arm Developer page

3. Functional description for SSE-320

The following sections describe the functional description of the components of SSE-320.

3.1 System and Security Control

SSE-320 provides several registers in the system to allow various features of the system to be discovered, configured, and controlled.

These registers are grouped into four register blocks:

System Information Register Block

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

System Control Register Block

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

Secure Access Configuration Register Block

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

Non-secure Access Configuration Register Block

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

3.1.1 Peripheral Protection Controllers

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

For more information, see *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* and *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*. In SSE-320, peripherals that are aliased to two memory areas, one Secure and one Non-secure, are protected by PPCs. The PPC defines in which region the peripheral resides.

In SSE-320, Timers, the NPU, and other peripherals are protected using PPCs. These are controlled by the Secure Access Configuration Register Block and Non-secure Access Configuration Register Block.

These registers also control Security Control Expansion Signals to drive external PPCs.

Two groups of peripherals are defined in SSE-320, that are protected behind Peripheral Protection Controllers.

These peripherals are:

- Peripheral Interconnect Peripheral Protection Controller Group 0 (PIPPC0). This group includes the following peripherals:
  - All Timestamp based Timers
  - NPU0
  - Watchdog Refresh Frames
  - Memory Protection Controller Configuration Register Block
  - Secure Access Configuration Register Block
  - Non-secure Access Configuration Register Block
  - Watchdog Control Frames
  - Debug system access
  - DMA subsystem
  - SDC-600 internal


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 (PIPPC1). This group includes the following peripherals:
  - SLOWCLOCK CMSDK Timers
  - System Control Register Block
  - SLOWCLOCK Watchdog Timer
  - PPUs
  - Lifecycle Manager (when LCM_KMU_SAM_PRESENT = 1)
  - Key Management Utility (when LCM_KMU_SAM_PRESENT = 1)
  - Security Alarm Manager (when LCM_KMU_SAM_PRESENT = 1)


Warning

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

Related information

- Power domains

3.1.2 Manager Security Controller

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

For additional 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*.

Related information

- NA

3.1.3 Memory Protection Controllers

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

In SSE-320, each memory page that is protected by the MPC, is aliased to two memory areas:

- Secure
- Non-secure

Depending on the security attributes 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 the Peripheral Region.

For more details, 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*.

3.2 Lifecycle Manager

SSE-320 defines a Lifecycle Manager (LCM) module that manages a set of RoT keys, debug controls and Security states stored in an OTP array. The LCM controls the life cycle state of the system, debug control signals and RoT keys.

The LCM also interacts with the Key Management Unit (KMU) by passing the RoT keys from the Non-Volatile One Time Programmable Memory into the KMU once per Cold reset. LCM supports Secure provisioning of confidential assets into OTP in an untrusted environment.

For more information on the LCM and KMU, see *Arm® Key Management Unit Specification* and *Arm® Lifecycle Manager Specification*. The configuration options of the LCM are defined in *Arm® Lifecycle Manager Specification*.

The LCM interfaces are integrated in SSE-320 as follows:

- APB3 subordinate interface to program the LCM registers.
    - This interface is PPC protected, always Secure access only and the Privileged or Uprivileged access is configurable. For details, see the PERIPHSPPPC1 description in *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.
    - Sparse writes are not allowed and must result in error response and memory intact.
    - The programming base address is defined in *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.
    - This interface also provides access to parts of the NVM (OTP).
- The Lifecycle Manager Non-Volatile Memory interface is exported to the top level. For more details, see Lifecycle Manager Non-Volatile Memory interface section in *Arm® Lifecycle Manager Specification*.
- The direct key APB3 manager interface is directly connected to Key Management Unit for exporting keys into the KMU.
- Lifecycle and debug control signals are exported to the top level. For details, see *Arm® Lifecycle Manager Specification*.
- Lifecycle Manager LFSR seed interface for loading a low-quality entropy random number into LCM is exported to the top level.
- Error indication connects to Security Alarm Manager and exported to the top level.
- The LCM clock is synchronous to MGMTSYSCLK.
- The LCM resides in the PD_MGMT power domain.
- The LCM reset is nCOLDRESETMGMT.

The context of the LCM must be saved and restored (retained) over HIBERNATION0. All signals that are driving PD_AON logic have to be latched and hold during HIBERNATION0.

3.2.1 LCM Debug Control Unit

The LCM Debug Control Unit provides Lifecycle dependent control for the Debug Authentication interface and the Lifecycle Manager Debug Control Unit interface outputs.

The Debug Authentication interface also has a separate override mechanism to force disable debug features during boot. This is controlled through LCM_DCU_FORCE_DISABLE System Control register through DCUEN[31:0] signals. For details of LCM_DCU_FORCE_DISABLE register, see LCM_DCU_FORCE_DISABLE register.

Note

The LCM_DCU_FORCE_DISABLE[21:0] are used in pairs where each pair of bit values represent a single debug control bit. Therefore, similarly, the DCUEN[21:0] signals are also used in pairs, where each pair of bit values represent a single debug control bit.

The LCM_DCU_FORCE_DISABLE[31:22] bits are used individually where each bit represents a single debug control bit. Therefore, similarly, the DCUEN[31:22] signals are used individually, where each bit represents a single debug control bit.

Assignment of DCUEN controls

The LCM DCUEN[31:0] signals are reserved for internal use and are not exposed to the expansion through the Lifecycle Manager Debug Control Unit interface.

The number of reserved signals is aligned to the number of signals that might be disabled by the LCM_DCU_FORCE_DISABLE register.

DCUEN[21:0]

The LCM DCUEN[21:0] signals and the LCM_DCU_FORCE_DISABLE[21:0] control bits are reserved for Debug Authentication interface control.

They are grouped in pairs and checked for odd parity.

The even bit of a DCUEN[i+1:i] signal pair has positive polarity (enabled), and the odd bit has negative polarity (disabled).

A pair value of:

- `0b01` signifies that the respective debug control is enabled.
- `0b10` signifies that the respective debug control is disabled.
- Pair values of `0b00` and `0b11` are illegal and behave as disabled and set an alarm signal.

The even bit of a LCM_DCU_FORCE_DISABLE.FD[i+1:i] pair has positive polarity (force disable), and the odd bit has negative polarity (force enable).

A pair value of:

- `0b01` signifies hat the respective debug control is force disabled.
- `0b10` signifies that the respective debug control value is determined by DCUEN[i+1:i] signal pair.
- Pair values of `0b00` and `0b11` are illegal and behave as force disabled and set an alarm signal.

Warning

Software setup of the DCU_EN<j> registers (see *Arm® Lifecycle Manager Specification*), the LCM_DCU_FORCE_DISABLE register, and hardware defaults LCM_DCU_FORCE_DISABLE_INIT must comply with the rules specified in the following table. Violating any of the rules results in an alarm.

The following table shows the DCUEN[21:0] rules.

In the table:

- FD[i] refers to LCM_DCU_FORCE_DISABLE.FD[i].
- Debug Alarm and Control Column describe how an alarm and a control are generated for each pair of DCUEN[21:0] bits.

Table 3-1: DCUEN[21:0] rules

LCM signal name	LCM signal use	FD	FD bit use	Debug Alarm and Control
DCUEN[0]	Positive polarity value	FD[0]	Positive polarity value	Alarm_1_0 = not(DCUEN[0] ^ DCUEN[1]) | not(FD[0] ^ FD[1])
DCUEN[1]	Negative polarity value	FD[1]	Negative polarity value	DBGEN = {DCUEN[0] & not(DCUEN[1])} & {not(FD[0] & not(FD[1]))} & {not(Alarm_1_0)}
DCUEN[2]	Positive polarity value	FD[2]	Positive polarity value	Alarm_3_2 = not(DCUEN[2] ^ DCUEN[3]) | not(FD[2] ^ FD[3])
DCUEN[3]	Negative polarity value	FD[3]	Negative polarity value	NIDEN = {DCUEN[2] & not(DCUEN[3])} & {not(FD[2] & not(FD[3]))} & {not(Alarm_3_2)}
DCUEN[4]	Positive polarity value	FD[4]	Positive polarity value	Alarm_5_4 = not(DCUEN[4] ^ DCUEN[5]) | not(FD[4] ^ FD[5])
DCUEN[5]	Negative polarity value	FD[5]	Negative polarity value	SPIDEN = { DCUEN[4] & not(DCUEN[5])} & {not(FD[4] & not(FD[5]))} & {not(Alarm_5_4)}
DCUEN[6]	Positive polarity value	FD[6]	Positive polarity value	Alarm_7_6 = not(DCUEN[6] ^ DCUEN[7]) | not(FD[6] ^ FD[7])
DCUEN[7]	Negative polarity value	FD[7]	Negative polarity value	SPNIDEN = {DCUEN[6] & not(DCUEN[7])} & {not(FD[6] & not(FD[7]))} & {not(Alarm_7_6)}
DCUEN[8]	Positive polarity value	FD[8]	Positive polarity value	Alarm_9_8 = not(DCUEN[8] ^ DCUEN[9]) | not(FD[8] ^ FD[9])
DCUEN[9]	Negative polarity value	FD[9]	Negative polarity value	DAPACCEN = {DCUEN[8] & not(DCUEN[9])} & {not(FD[8] & not(FD[9]))} & {not(Alarm_9_8)}
DCUEN[10]	Positive polarity value	FD[10]	Positive polarity value	Alarm_11_10 = not(DCUEN[10] ^ DCUEN[11]) | not(FD[10] ^ FD[11])
DCUEN[11]	Negative polarity value	FD[11]	Negative polarity value	DAPDSSACCEN {DCUEN[10] & not(DCUEN[11])} & {not(FD[10] & not(FD[11]))} & {not(Alarm_11_10)}
DCUEN[12]	Positive polarity value	FD[12]	Positive polarity value	Alarm_13_12 = not(DCUEN[12] ^ DCUEN[13]) | not(FD[12] ^ FD[13])
DCUEN[13]	Negative polarity value	FD[13]	Negative polarity value	SYSDSSACCEN0 = {DCUEN[12] & not(DCUEN[13])} & {not(FD[12] & not(FD[13]))} & {not(Alarm_13_12)}
DCUEN[14]	Positive polarity value	FD[14]	Positive polarity value	Alarm_15_14 = not(DCUEN[14] ^ DCUEN[15]) | not(FD[14] ^ FD[15])
DCUEN[15]	Negative polarity value	FD[15]	Negative polarity value	SYSDSSACCENX = {DCUEN[14] & not(DCUEN[15])} & {not(FD[14] & not(FD[15]))} & {not(Alarm_15_14)}
DCUEN[16]	Positive polarity value	FD[16]	Positive polarity value	Alarm_17_16 = not(DCUEN[16] ^ DCUEN[17]) | not(FD[16] ^ FD[17])
DCUEN[17]	Negative polarity value	FD[17]	Negative polarity value	SYSDSSACCEN1 = {DCUEN[16] & not(DCUEN[17])} & {not(FD[16] & not(FD[17]))} & {not(Alarm_17_16)}
DCUEN[18]	Positive polarity value	FD[18]	Positive polarity value	Alarm_19_18 = not(DCUEN[18] ^ DCUEN[19]) | not(FD[18] ^ FD[19])

LCM signal name	LCM signal use	FD	FD bit use	Debug Alarm and Control
DCUEN[19]	Negative polarity value	FD[19]	Negative polarity value	SYSDSSACCEN2 = {DCUEN[18] & not(DCUEN[19])} & {not(FD[18] & not(FD[19]))} & {not(Alarm_19_18)}
DCUEN[20]	Positive polarity value	FD[20]	Positive polarity value	Alarm_21_20 = not(DCUEN[20] ^ DCUEN[21]) | not(FD[20] ^ FD[21])
DCUEN[21]	Negative polarity value	FD[21]	Negative polarity value	SYSDSSACCEN3 = {DCUEN[20] & not(DCUEN[21])} & {not(FD[20] & not(FD[21]))} & {not(Alarm_21_20)}

All the alarm signals described in the previous table are OR-ed together to generate debug_signals_security_checker_err. This signal is then OR-ed with the FATALERR from the Lifecycle Manager Lifecycle Indication interface and used to generate an alarm input at the Security Alarm Manager (SAM).

For details of the associated SAM event, see *Arm® Security Alarm Manager Specification*.

DCUEN[31:22]

The DCUEN[31:22] signals and the LCM_DCU_FORCE_DISABLE[31:22] system register bits are individually used and do not generate alarms.

The DCUEN[31:22] signals are force disabled by the respective LCM_DCU_FORCE_DISABLE.FD[31:22] register bit signals as shown in the following table. In the table, FD[i] refers to LCM_DCU_FORCE_DISABLE.FD[i].

Table 3-2: DCUEN[31:22] rules

LCM signal name	LCM signal use	FD	FD bit use	Debug Control
DCUEN[31:22]	Reserved	FD[31:22]	Reserved	Reserved for internal usage (that is, BIST enable, ECOs). The use of each of DCUEN[31:22] is force disabled as follows: DCUEN[i] & not(FD[i]) for i=22 to 31.

DCUEN[127:32]

These DCU controls are connected to Lifecycle Manager Debug Control Unit interface and do not have corresponding force disable mechanism present.

3.3 Key Management Unit

SSE-320 uses a Key Management Unit (KMU). The KMU is a centralized function that stores symmetric key material for the distributed hardware countermeasures and for the use of software with the different crypto devices.

The KMU is a Secure only accessible device. Non-secure software must call Secure software to set or use a key. After a key slot is loaded, Secure software can lock it to prevent accidental reload or malicious overriding it.

The KMU implements hardware key slots and software key slots. The keys of the hardware key slots can only be set through a Private APB HW keys Port which connects point to point using **IMPLEMENTATION DEFINED** address to the Lifecycle Manager.

A software or other hardware entity cannot write to this private port. Once the LCM populates the hardware key slots, software can start using them in the same way as it uses a locked software key slot.

The context of Key Management Unit, must be saved and restored (retained) over HIBERNATION0.

All signals that are driving PD_AON logic have to be latched and hold during HIBERNATION0.

The KMU interfaces integrated in SSE-320 are as follows:

- APB3 subordinate interface to program the KMU registers:
  - This interface is PPC protected, always Secure access only and the Privileged/Unprivileged access is configurable through PERIPHSPPPC1.
  - Sparse writes are not allowed and must result in error response and memory intact.
  - The programming base address is defined in the *Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide*.
- The direct key APB3 subordinate interface.
  - This interface is used by the LCM hardware to load hardware keys to the hardware key slots of the KMU. This interface is connected in a point-to-point fashion to the Lifecycle Manager in a way that it is not accessible to software or any other hardware.
  - The memory map exposed in the direct key APB3 subordinate port is **IMPLEMENTATION DEFINED** for LCM. The only registers in the KMU which are accessible through this interface are the hardware key slots.
- AHB5 Manager interface:
  - This interface is used by the KMU to export keys to the target crypto devices.
  - The transaction attributes that the KMU provides through this interface are Secure, and the Privileged is configurable through PERIPHSPPPC1.
  - Additional transaction attributes that the KMU provides through this interface are defined in *Arm® Key Management Unit Specification*.
  - This interface is connected to top level as a dedicated Key Management Unit interface. Key Management Unit interface.
- Parity Error interface. This interface is used by the KMU to generate alarm on detection of internal parity error. This signal is connected to the subsystem's Security Alarm Manager (SAM).
- Interrupt interface. This interface is used by the KMU to interrupt the software. This is connected to system Interrupts.

KMU details

Clock

KMU clock is synchronous to MGMTSYSCLK.

Power domain

KMU resides in the PD_MGMT power domain.

Reset

KMU reset is primarily nCOLDRESETMGMT. However, during Secure Asset Provisioning Flow, when LCMRSTREQ is asserted, the nWARMRESETMGMT also resets the KMU. For more details of these signals, see Reset distribution.

3.4 Security Alarm Manager

SSE-320 defines Security Alarm Manager (SAM). The SAM provides means to apply the programed response to security events detection.

For more information regarding the SAM, see *Arm® Security Alarm Manager Specification*.

The SAM interfaces that are integrated in SSE-320, are as follows:

APB3 subordinate interface to program the SAM registers

This interface is PPC protected, always Secure access only and the Privileged/Unprivileged access is configurable, see PERIPHSPPPC1 register.

- Sparse writes are not allowed and must result in error response and memory intact.
- The programming base address is defined in Peripheral Region.

Input Events interface

This interface is available on Sensor Alarm Interface and supports both internal and external events. The integrator also connects its digital and analog attack detection sensors to these inputs. For more details, see *Arm® Security Alarm Manager Specification*.

The External Sensors Ready input signal

This input is available on Sensor Alarm Interface. The integrator must set this signal when all the external sensors are stable after power on to avoid false alarms.

Platform-specific event logging inputs

The SAM hosts platform-specific event logging registers for sources which provide many event signals, for example, Processors DCLS and RAS status signals or memory ECC. These platform-specific event logging inputs and the relevant registers are not used by SSE-320 and not required for PSA Level 2.

Status interface

This interface is available on Sensor Alarm Interface. These output signals indicate the detected events. Each signal indicates, when set, that its respective sensor reported alarm condition. For more details, see *Arm® Security Alarm Manager Specification*.

Response Action interface

This interface is available on Sensor Alarm Interface. Each signal indicates, when set, a response action to be taken by the system. The integrator connects each of the response signals to a mitigation action. The recommended actions are detailed in Security Alarm

Manager incoming events allocation. For details, see *Arm® Security Alarm Manager Specification*.

SAM configuration done interface

This interface is available on *Arm® Security Alarm Manager Specification*:

- SAM clock is synchronous to AONCLK
- SAM resides in the PD_AON power domain
- SAM resets are nCOLDRESETAON and nPORESETAON, see Reset distribution.

Related information

- NA

3.5 Timers and watchdogs

SSE-320 supports two main classes of timers and watchdogs: Timestamp based timers and SLOWCLK AON based timers.

3.5.1 Timestamp based timers

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

The four Timestamp-based timers support:

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

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

- Memory-mapped System Timer with register access through the Peripheral Interconnect
- 64-bit timestamp input, generating events through comparison with a timer value
- An 'auto-increment' feature to support 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

SSE-320 provides four Timestamp Timers and two Timestamp Watchdog timers. These are mapped to the following addresses:

- Timer 0 at address `0x4800_0000` and aliased to `0x5800_0000`

- Timer 1 at address `0x4800_1000` and aliased to `0x5800_1000`
- Timer 2 at address `0x4800_2000` and aliased to `0x5800_2000`
- Timer 3 at address `0x4800_3000` and aliased to `0x5800_3000`
- Secure Watchdog timer control frame at address `0x5804_0000` and refresh frame at `0x5804_1000`
- Non-secure Watchdog timer control frame at address `0x4804_0000` and refresh frame at `0x4804_1000`

Timer 0, Timer 1, Timer 2, and Timer 3 can be configured by software to be a Secure or a Non-secure timer through the Peripheral Protection Controller that is controlled through the Secure Access Configuration register block.

The security configuration of both watchdogs is non-configurable, one is always Non-secure and the other always Secure.

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

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

3.5.2 SLOWCLK AON timers

SLOWCLK AON timers and watchdogs are simple CMSDK based 32-bit timers that run on SLOWCLK.They reside in the PD_AON Power domain and are reset by nWARMRESETAON. A single timer and a single Secure Watchdog are provided.

The Slow clock AON timer and watchdog are expected to be used when the system is in the HIBERNATION0 System Power state, when potentially only SLOWCLK is available and running, and all other clocks are off.

These timers are mapped to the following addresses:

- SLOWCLK Timer at address `0x4802_F000` and aliased to `0x5802_F000`
- SLOWCLK Secure Watchdog Timer at address `0x5802_E000`

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

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*.

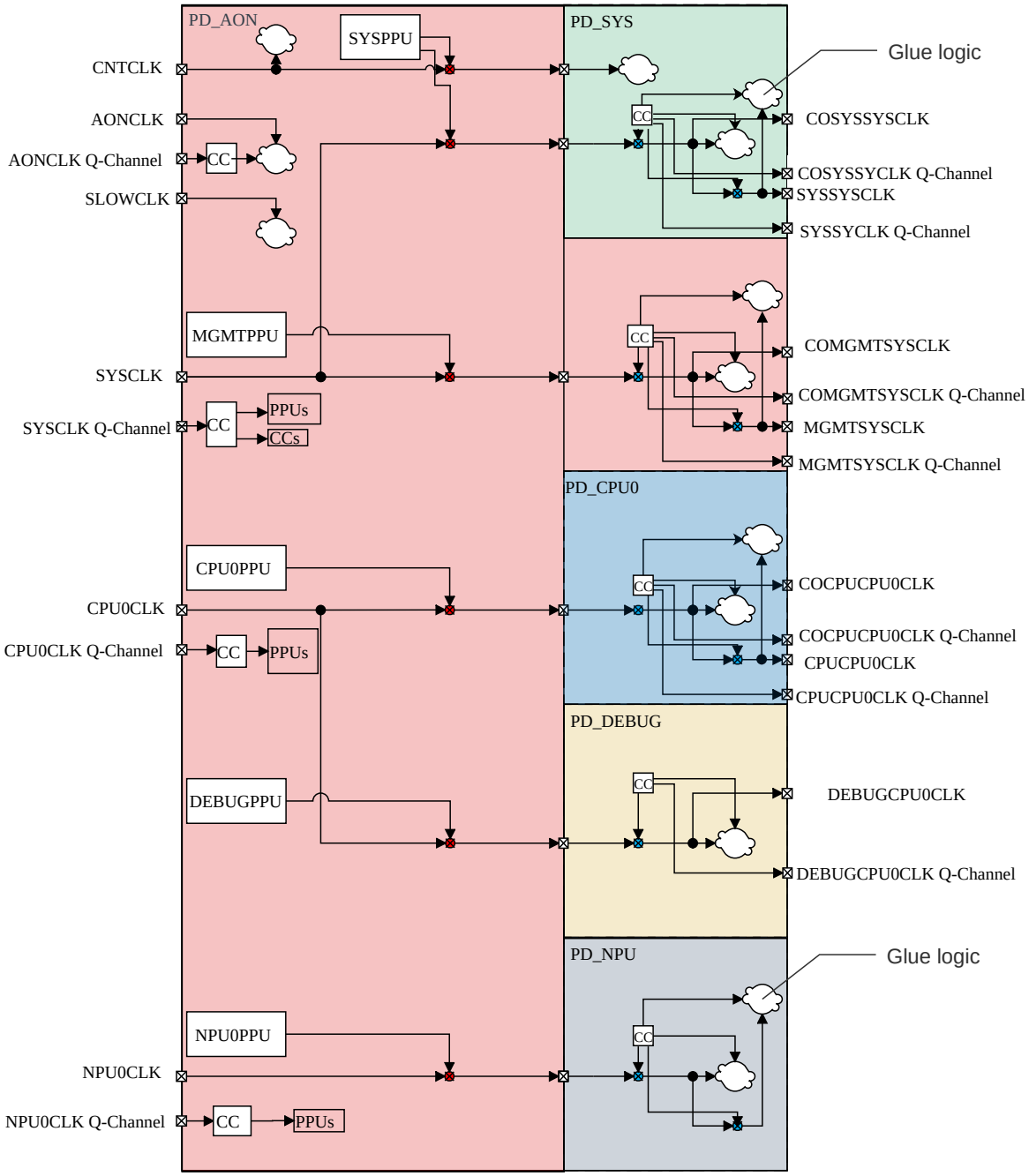
3.6 Clocking infrastructure

SSE-320 provides several input clocks and output clocks in the subsystem.

In typical use, Arm recommends to drive SLOWCLK using a 32kHz clock source. If reducing standby power is an important consideration for a product, AONCLK can be driven at a lower clock rate compared to SYSCLK (around 1MHz to 10MHz). This lower clock rate improves the transition time required to enter and leave the lowest power state of the subsystem, but still supports 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.

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

Figure 3-1: SSE-320 clock distribution structure



Note

Power Domain refers to the power domain where the clock is expected to be used in. All clocks always first enter via the PD_AON domain before being used in their respective power domain.

Input clocks

SLOWCLK

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

This clock resides in the PD_AON power domain.

SLOWCLK is also required to run while others can be turned off when the system is in its lowest power state (HIBERNATE0) other than OFF.

External gating of SLOWCLK is not supported by clock Q-Channels.

This clock drives the following logic that resides in the PD_AON power domain:

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

AONCLK

Always on 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 be independent from the rest of the system. AONCLK is asynchronous to the other clocks in the system.

This clock resides in the PD_AON power domain.

If the corresponding clock Q-Channel is in Q_STOPPED state, then AONCLK can be gated externally.

AONCLK drives all other logic that resides in the PD_AON domain, except logic that originates in the PD_MGMT power domain, which is merged into PD_AON.

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

This clock drives the following:

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

SYSCLK

This is the main system clock that drives most of the system that resides in the PD_SYS power domain.

This clock resides in the PD_AON power domain.

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

SYSCLK is completely asynchronous to the other clocks in the subsystem except for CPU0CLK, and NPU0CLK.

The clock source is gated internally by PPUs before it is used by any logic.

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

- Is not at its lowest HIBERNATION 0 or 1 state.
- In SYS_RET state, is waking from those states.
- Is forced to stay only via the CLOCK_FORCE register.

This clock drives the following:

- The Main Interconnect and Peripheral Interconnect and other related functionality like expansion interfaces.
- System and Security Control related registers and logic, except those that resides in PD_AON.
- Power Control logic that resides in PD_AON power domain that controls the PD_SYS, PD_CPU0, and PD_DEBUG power domains.
- All Volatile Memory interfaces and peripherals in the PD_SYS domain, along with the interfaces to Timers and Watchdog timers.

CPU0CLK

This clock input drives 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, and NPU0CLK.

This clock resides in the PD_AON power domain.

If the corresponding clock Q-Channel is in Q_STOPPED state, then CPU0CLK might be externally gated. The clock source is gated internally by PPUs before it is used by any logic.

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

This clock is normally expected to be requested to turn on when CPU0 core domain, PD_CPU0, or the CPU0 debug domain in the Debug System, PD_DEBUG, is not off or is requested to turn on.

DEBUGCLK

This clock must exist when HASCSS = 1.

This clock drives the Debug System. The Q-Channel Control interface, DEBUGCLK Q-Channel Control interface, allows the subsystem to request and handshake the availability of the DEBUGCLK.

This clock is expected to be requested to run in any of the following cases:

- Its associated CPU core's debug domain in the Debug System, PD_DEBUG, is not in OFF state.
- It is forced to run through the CLOCK_FORCE register.
- Clock request from the logic associated with this clock.

CNTCLK

This clock is associated with the CNTVALUEB System Counter Timestamp input. CNTCLK is asynchronous to the other clocks in the subsystem.

This clock resides in the PD_AON power domain.

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

External gating of CNTCLK is not supported by clock Q-Channels.

NPU0CLK

This is the clock used to drive the logic reside in the PD_NPU0 domain.

This clock resides in the PD_AON power domain.

NPU0CLK is asynchronous to the other clocks in the subsystem except for SYSCLK and CPU0CLK.

A Q-Channel Control interface, NPU0CLK Q-Channel Control Interface, allows the subsystem to request and handshake the availability of the NPU0CLK. This clock is normally expected to be requested to turn on when NPU0 domain, PD_NPU0, is not off or is requested to turn on.

Output clocks

SSE-320 provides a Q-Channel interface for each of the output clocks to allow expansion logic to control the availability of each clock output.

COMGMTSYSCLK

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.

This clock resides in the PD_AON power domain.

MGMTSYSCLK

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.

This clock resides in the PD_AON power domain.

COSYSSYSCLK

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.

This clock resides in the PD_AON power domain.

SYSSYSCLK

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.

This clock resides in the PD_SYS power domain.

COCPUCPU0CLK

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.

This clock resides in the PD_CPU0 power domain.

CPUCPU0CLK

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.

This clock resides in the PD_CPU0 power domain.

DEBUGCPU0CLK

The gated version of CPU0CLK, expected to be used to drive expansion logic that resides in the PD_DEBUG power domain.

This clock resides in the PD_DEBUG power domain.

Related information

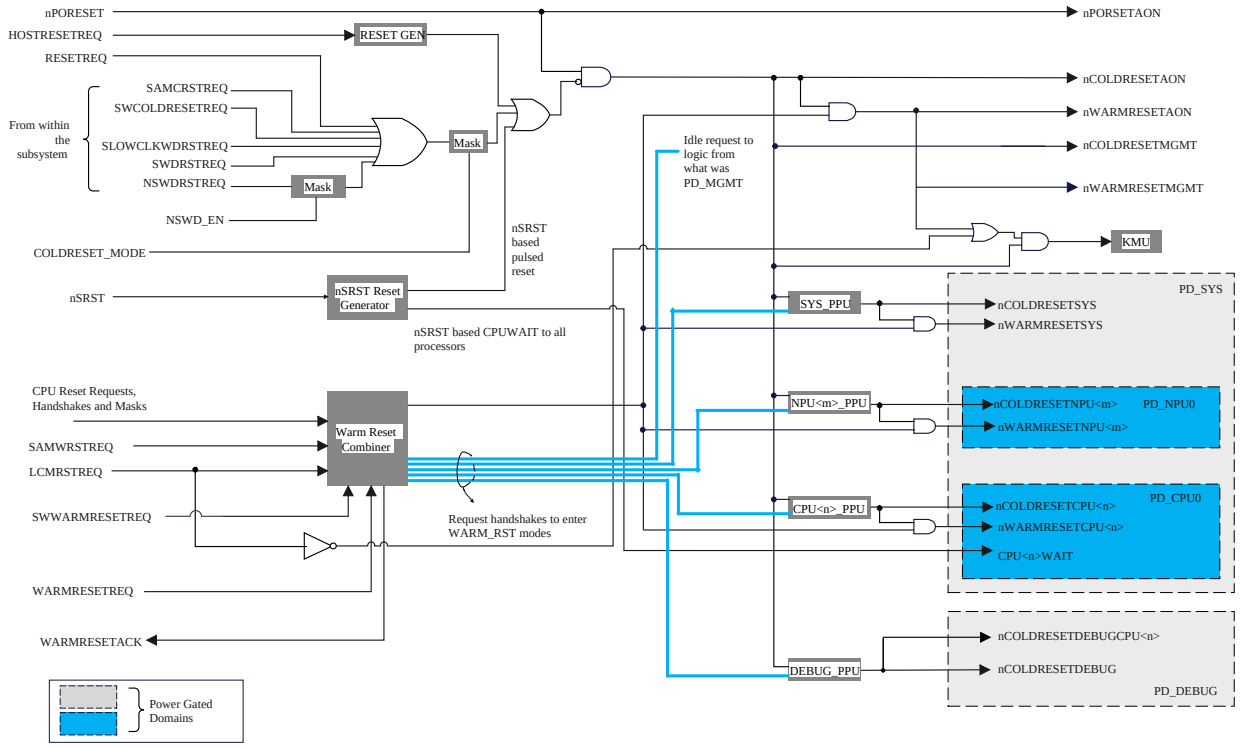
- NA

3.7 Reset distribution

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

Figure 3-2: Reset distribution in SSE-320 on page 65 shows the reset distribution in SSE-320. It also shows nPORESETAON, which is not an output of the subsystem and only resets the RESET_SYNDROME register. This figure does not show the infrastructure for reset resynchronization and how the resets are used within each power domain.

Figure 3-2: Reset distribution in SSE-320



3.7.1 Power-on and cold reset handling

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

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

- All reset requests of the watchdog timers. If there exists a relevent mask for the reset request and COLDRESET_MODE = 0, then this is through the mask.
- Reset request input RESETREQ
- Reset request through the SWRESET.SWRESETREQ register value
- Reset request input HOSTRESETREQ
- Reset generated from the nSRST request by using negative-edge detection first and then stretching the result

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

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.

Related information

- Reset distribution

3.7.2 CPU reset handling

The nPORESET reset input of CPU0 is driven by the CPU0PPU controlling the CPU core power domain, which is PD_CPU0. CPU0PPU is reset using nCOLDRESETAON.

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

The Warm reset mode is driven by a separate logic that resides in the PD_AON domain. This logic, along with all PPUs in the system is used to force the system to an idle (quiescent) state (WARM_RST System Power State is entered) before Warm reset mode assertion, which includes the nSYSRESET input of CPU0.

Related information

- Warm reset generation and control

3.7.3 Warm reset generation and control

The system provides a Warm reset signal nWARMRESETCPU0 for each power domain.

This reset is generated from the system reset request (SYSRESETREQ) of the processor and can be masked using the RESET_MASK register.

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

Note

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. If 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.

Related information

- Power domains

3.7.4 Boot after reset

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

The default address is configurable, but Arm recommends that the default is set to `0x0100_0000`, which is mapped to code memory through the Manager Code Main Expansion Interface (XMSTEXPCODE). This address can be modified by software before subsequent warm reboots of the processor.

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

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

Related information

- Reset distribution

3.7.5 NPU reset handling

The nRESET reset input of NPU0 is driven by the PPU that controls the NPU power domain PD_NPU0.

This PPU is reset using nCOLDRESETMGMT reset. A Warm reset is driven from Warm reset generation logic that resides in the PD_AON domain.

Related information

- NPU security mapping
- NPUSPPORPL
- NPUNSPORPL
- NPUSPPORSL

3.8 Debug infrastructure

SSE-320 supports the CoreSight SoC-600M based common debug infrastructure.

SSE-320 supports two possible configurations that define the extent of its debug system infrastructure.

The configurations defining the extent of the debug system infrastructure are as follows:

HASCSS = 0

In this basic debug configuration a CoreSight SoC-600M based common debug infrastructure is not defined and does not exist. Instead, all debug interfaces are brought out from the processor as expansion interfaces. When HASCSS = 0, NUMCPU must be 0.

HASCSS = 1

In this full debug configuration, a CoreSight SoC-600M based common debug infrastructure exists and is called the Debug System.

A debug access to any of the systems obeys normal memory map and decoding rules. A Debug access must not cause a security violation interrupt to be generated, even if the debugger performs an operation that would normally cause one.

For more details, see *Arm® Corstone™ Reference Systems Architecture Specification Ma2*.

Related information

- Debug system block

3.8.1 Debug access

The CoreSight SoC-600M based debug infrastructure provides a single APB4 Debug Access interface for an expansion Debug Access Port (DAP) to connect to the two Memory Access Ports (MEM-AP) of SSE-320, along with a Debug ROM table.

The purpose of the MEM-AP and the Debug ROM table are as follows:

- One MEM-AP is used for accessing the Shared Debug System infrastructure components, including the debug expansion logic on the Debug APB Expansion interface.
- One MEM-AP is used for CPU0 to provide access to CPU0 Debug System's access interface.

- The Debug System ROM table provides information about the MEM-APs.

For debug via the APB4 Debug Access interface, accesses out of the preceding MEM_APs are controlled by Debug Authentication signals. DAPDSSACCEN provides overall access control, combined with DBGEN or NIDEN to control Non-secure access, and SPIDEN or SPNIDEN to control Secure access.

System interconnect access to the MEM-APs is also provided. Access is gated to control which of the processors in the system, or **IMPLEMENTATION DEFINED** manager, or group of managers are allowed to access the MEM-APs components. Access gating is done by using the SYSDSSACCEN0 and SYSDSSACCENX Debug Authentication signals. Once access is granted to the debug system, DBGEN or NIDEN then controls Non-secure access and SPIDEN or SPNIDEN controls Secure access.

Access via the system interconnect are mapped to the addresses:

- from `0xE010_0000` to `0xE01F_FFFF` in the Non-secure world
- from `0xF010_0000` to `0xF01F_FFFF` in the Secure world

The PERIPHNSPPC0.NS_SYSDSS register determines which of the two regions is accessible and the PERIPHSPPPC0.SP_SYSDSS determines the privilege level.

CPU debug access

To provide debug access to each processor core, SSE-320 provides a MEM-AP accessible through the Debug Access interface. From this MEM-AP, a Class `0x9` Debug ROM table is provided to:

- Point to the CPU0 ROM table that is private to each processor at PPB address region at `0xE00F_F000`. This ROM table also provides Granular Power Requestor (GPR) function to allow the debugger to wake CPU0.
- Optionally, also point to another ROM table: CPU0 MCU ROM table. This ROM table is added by the system integrator that resides in the PPB address region and in the EPPB expansion bus at the address: CPU0MCUROMADDR. The existence of the pointer to the CPU0 MCU ROM is configurable.

All access from the MEM-AP that does not access the ROM table is to be forwarded to the Debug Access interface of the processor. It is **IMPLEMENTATION DEFINED** if the access goes through power and/or clock crossing bridge.


Note

Accessibility through the Debug Access interface of the processor depends on the DBGEN and SPIDEN Debug Authentication signals, and the DAUTHCTRL.UIDEN register in the processor core. For more information, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

Shared Debug System Access

SSE-320 provides a MEM-AP that is accessible through the Debug Access interface to provide access to the Shared Debug System. This Shared Debug System provides the following:

- Shared Debug System CoreSight ROM that describes all debug components in the Shared Debug System accessible through this MEM-AP. This ROM table includes an entry pointing to an external CoreSight ROM table at the `0x0008_0000` address through the Debug APB Expansion interface, which defines what the system integrator has added as expansion.
- Trace Funnel to funnel all trace data sources together.
- Replicator and an Embedded Trace Buffer (ETB) to allow traced data to be either:
  - To be forwarded to a Trace Port Interface Unit (TPIU) in the expansion logic
  - To be temporarily stored in the ETB so that the software or the debugger can read them through the Debug Access interface
- CTM and CTI that support triggers to and from:
  - DMA, Timers, and Watchdogs in the system
  - Triggers from the processor cores
  - Triggers from the expansion system
- Debug APB Expansion interface to add debug components to the system. This allows a system integrator to extend the Shared Debug System in accordance with the needs of the SoC.

For example, in Full Debug Configuration (HASCSS = 1) the Debug Expansion adds the following to complete the debug solution as a standalone microcontroller:

- DAP to allow an external debugger to access the platform.
- Access Control Gate, controlled using DAPDSSACCEN Debug Authentication Access Control signal.
- CTI to provide triggers to and from the Trace Port Interface Unit (TPIU).
- TPIU to output trace data.
- SDC-600 Secure Debug Channel External APBCOM to provide an interface for Secure communications between the debugger and the Secure firmware in the system. Through the interface Secure debug certificates can be injected into the platform. The External APBCOM connects to the optional SDC-600 Secure Debug Channel Internal APBCOM.
- Debug Timestamp generator.

3.8.2 Timestamps

SSE-320 provides a debug timestamp input, TSVALUE<B/G>[63:0], which is used by all debug logic that requires timestamp within the subsystem. These are primarily the processor cores in the system.

3.8.3 Cross Trigger

The Shared Debug System implements a Cross Trigger Matrix (CTM) and a single Cross Trigger Interface (CTI). These together allow the DMA, the timers, and the watchdog timers in the SSE-320 subsystem to be halted and restarted.

The halting and restarting occurs by using trigger sources from any of the processor cores and also from trigger sources that are external to the subsystem through the Cross Trigger Channel interface.

The first four Cross Trigger inputs and the first eight outputs are reserved for internal use to support the following:

- SLOWCLK watchdog and SLOWCLK timer halting as follows:
  - CTIEVENTOUT[0] of the CTI is used internally to halt the SLOWCLK timer and watchdog.
  - CTIEVENTOUT[1] of the CTI is used internally to restart the SLOWCLK timer and watchdog.
- Triggers to and from the ETB:
  - CTIEVENTOUT[2] of the CTI is used to drive TRIGIN input of the ETB to request the ETB to insert a trigger in a trace stream.
  - CTIEVENTOUT[3] of the CTI is used to drive FLUSHIN input of the ETB to initiate a flush.
  - CTIEVENTIN[0] of the CTI takes the event output FLUSHCOMP of the ETB to indicate that a flush operation is completed.
  - CTIEVENTIN[1] of the CTI takes the event output ACQCOMP of the ETB to indicate that a trace acquisition is completed.
  - CTIEVENTIN[2] of the CTI takes the event output FULL of the ETB to indicate that either the trace memory is full or the write pointer wrapped around.
  - CTIEVENTIN[3] of the CTI is reserved and tied 0.
- If NUMDMA > 0, the following triggers halt and restart the DMA, they are reserved if NUMDMA=0
  - CTIEVENTOUT[4] of the CTI is used internally to halt the DMA.
  - CTIEVENTOUT[5] of the CTI is used internally to restart the DMA.

If NUMNPU > 0, the following triggers halt and restart the NPU, they are reserved if NUMDMA=0.

- CTIEVENTOUT[8] of the CTI is used internally to halt the Ethos-U85.
  - CTIEVENTOUT[9] of the CTI is used internally to restart the Ethos-U85.

All other CTI inputs and outputs of the CTI blocks, are made available as expansion signals through CTIEVENTIN[<NUM_EVENT_SLAVES-1:4], CTIEVENTOUT[7:6] and CTIEVENTOUT[<NUM_EVENT_SLAVES-1:10>].

NUM_EVENT_SLAVES and NUM_EVENT_MASTERS are defined by the NUM_EVENT_SLAVES and NUM_EVENT_MASTERS configuration of the css600_cti device respectively.

Both NUM_TRIG NUM_EVENT_SLAVES and NUM_EVENT_MASTERS must be >= 10 if NUMNPU > 0.

For more information on the css600_cti, see *Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0*.

When integrating a SSE-320 based subsystem with HASCSS=1, we recommend that the trigger signals are used to also halt the System Timestamp counter in the following way:

- Use CTIEVENTOUT[6] to halt the System Timestamp counter.
- Use CTIEVENTOUT[7] to restart the System Timestamp counter.


Note

It is the responsibility of the debug software to halt and restart NPUs in the system, using their programming interface.

3.8.4 Trace infrastructure

When DEBUGLEVEL = 2, SSE-320 provides a trace infrastructure that funnels all trace source from all processor cores into a single trace data stream.

The single trace date stream is replicated. One of the streams drives the ATB Trace interface that is expected to be picked up by a TPIU. The other stream is taken up by the ETB, allowing the trace data to be read by the software or by an external debugger through the Debug Access interface.

When DEBUGLEVEL < 2, SSE-320 does not provide any trace infrastructure.

3.9 Power infrastructure

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

The power control infrastructure specifies:

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

SSE-320 supports the following Power Control Infrastructure:

- Power domain hierarchy and bounded regions introduces terms that are used in the power-related descriptions.
- Power domains depicts the distribution of SSE-320 components across the power domains.

- Power Policy Units describes the configuration and control of the PPUs that control the power domains of the subsystem.
- Bounded region power modes describes the valid power state combinations and state transitions of power domains in each Bounded Region.
- Power Control Wakeup Q-Channel Device interfaces defines the sources that can wake up the subsystem.
- Power Dependency Control defines the software configurable Power Dependency Control Matrix, that is responsible for keeping up power domains based on power domain states and external inputs.

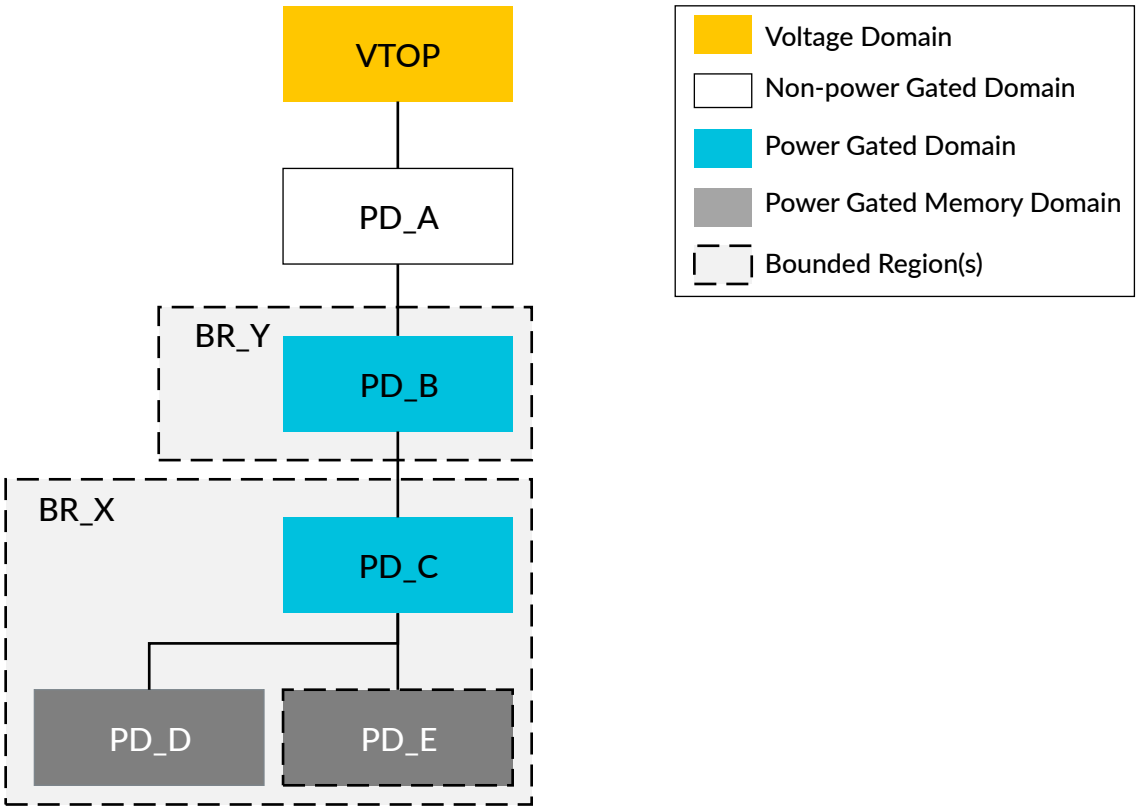
For the valid power state combinations of power domains in all Bounded Regions, see System power states.

3.9.1 Power domain hierarchy and bounded regions

SSE-320 has defined relationships between the power and voltage domains that are governed by relationship rules.

The following figure shows a simple power domain hierarchy diagram that describes the relationships between power and voltage domains.

Figure 3-3: Example power domain hierarchy



In Figure 3-3: Example power domain hierarchy on page 74:

- 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, where the power domains have an hierarchical relationship:

When a higher-level power domain in the hierarchy has lower-level power gated domains below it.

Before the higher-level domain can enter a lower power state, the lower-level power gated domains must:

1. Already be in a lower power state
2. 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 must already be in their highest power state.

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

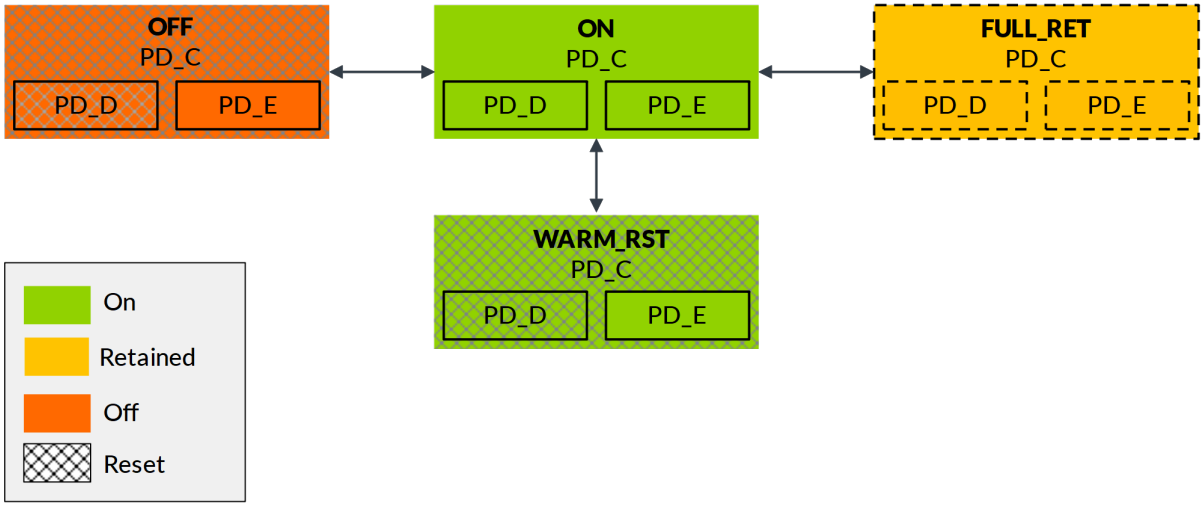
The power modes of a Bounded Region are represented using a state transition diagram that shows the valid power state combinations and transitions of power domains in the Bounded Region.

For example, the following figure shows a simple, four power mode transition diagram, for a Bounded Region with three power domains: PD_B, PD_C, and PD_D.

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

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

Figure 3-4: Example power mode transition diagram of bounded power regions PD_C, PD_D, and PD_E



Note

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

3.9.2 Power domains

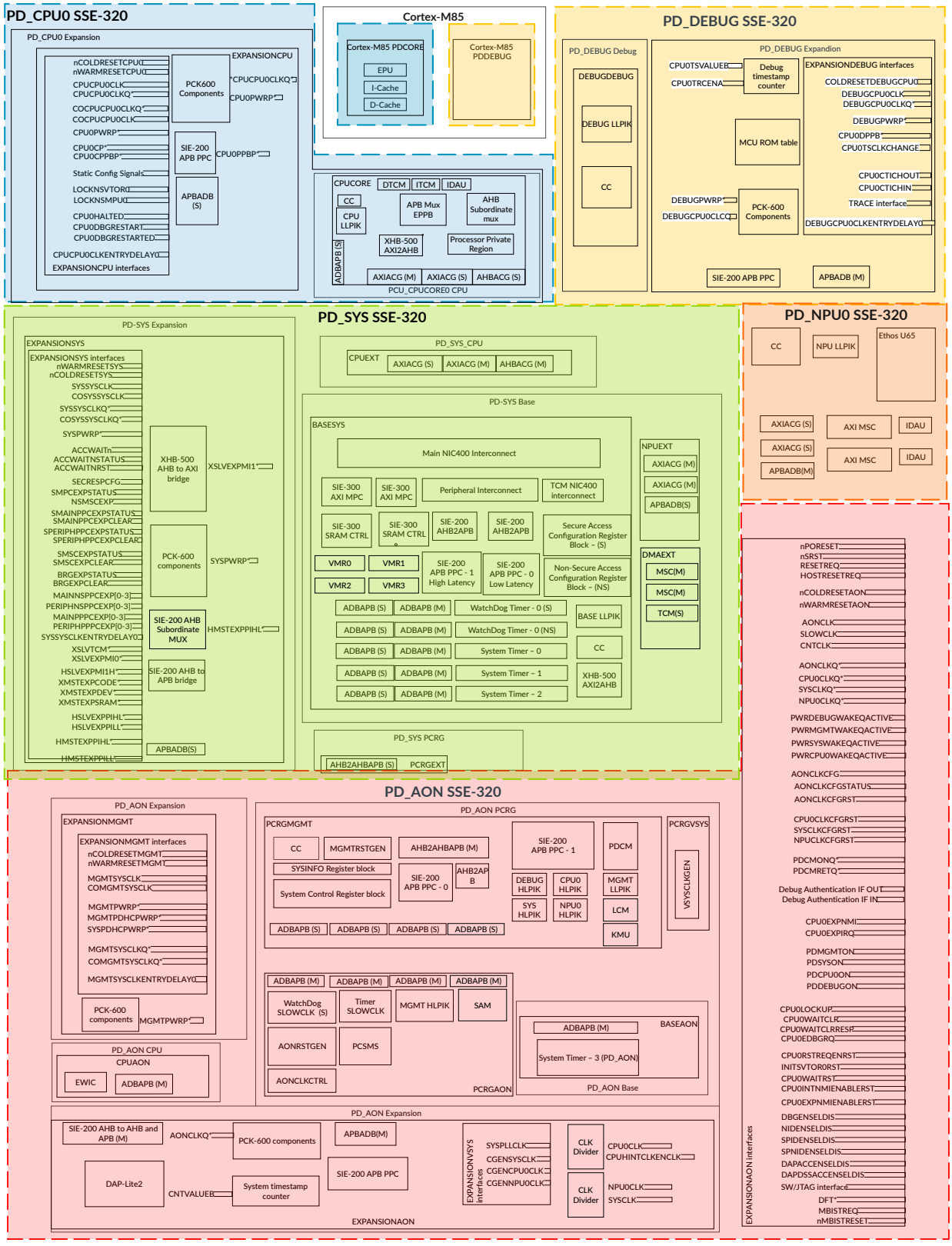
SSE-320 is partitioned into multiple power domains.

These power domains are depicted in Figure 3-5: SSE-320 power domains in the power aware integration layer on page 77.

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

The following figure shows the example expansion logic, which is only included if EXPLOGIC_PRESENT = 1. For details, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

Figure 3-5: SSE-320 power domains in the power aware integration layer



The location and name of top RTL file of the power domain structured system topology is:

```
/configuration/sse320/logical/trunk/top_iot_sse320_0_<CONFIG_NAME>/
top_iot_sse320_top_0_<CONFIG_NAME>/verilog/
top_iot_sse320_top_logical_0_<CONFIG_NAME>.sv
```

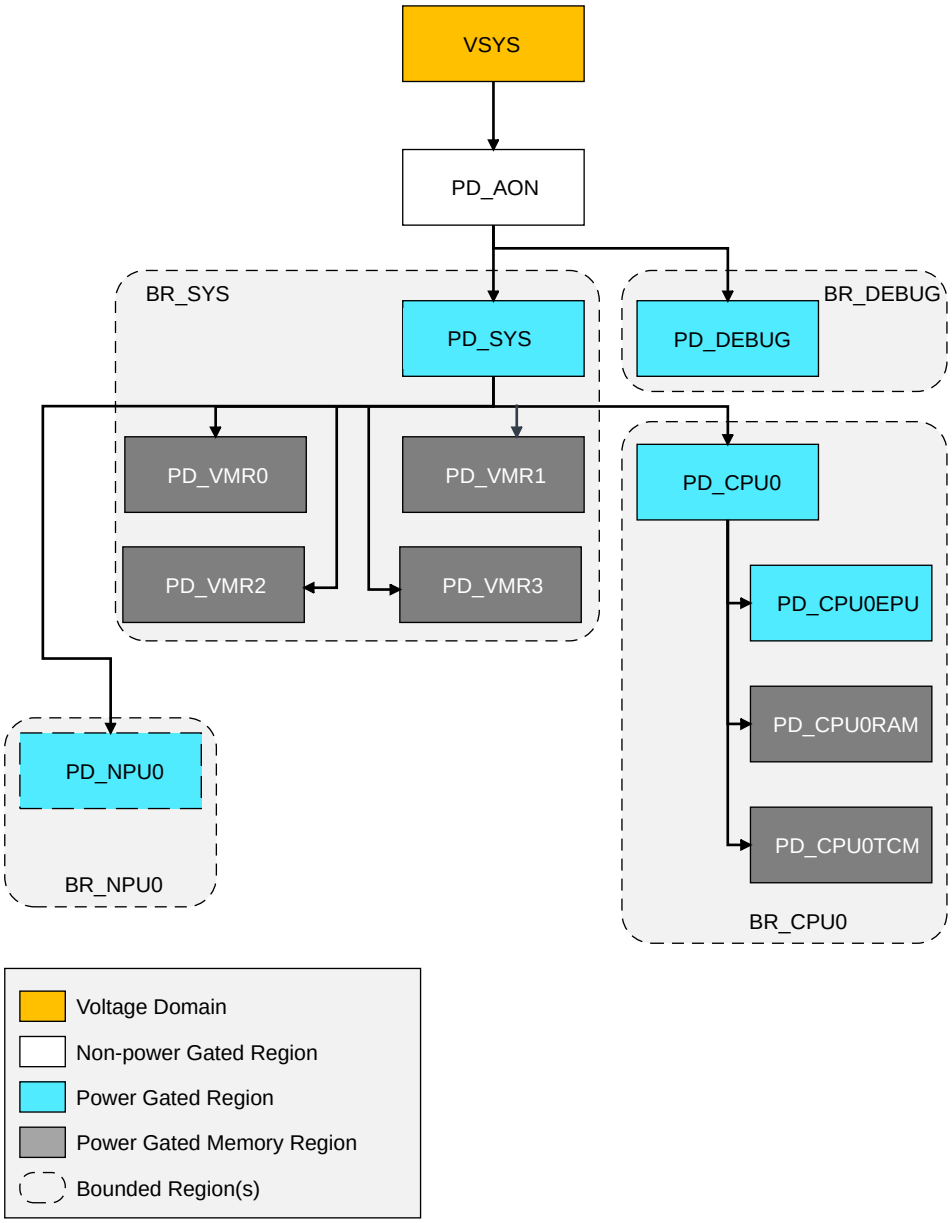
The location and name of top RTL file of the logical structured system topology is:

```
/configuration/sse320/logical/trunk/top_iot_sse320_0_<CONFIG_NAME>/
top_iot_sse320_top_0_<CONFIG_NAME>/verilog/top_iot_sse320_top_0_<CONFIG_NAME>.sv
```

These power domains are hierarchical and Figure 3-6: Voltage and power domain hierarchy of SSE-320 on page 79 shows the voltage and power domain hierarchy.

For more information regarding how the power domains are controlled, see Power Policy Units.

Figure 3-6: Voltage and power domain hierarchy of SSE-320



3.9.3 Power Policy Units

SSE-320 uses Power Policy Units (PPUs) with P-Channel device interfaces for power control for each *Bounded Region* (BR) in the system, except for the MGMTPPU.

The MGMTPPU does not control the power state of any power domains, since PD_AON is always on. It facilitates Power-on reset, Cold reset, and Warm reset related state transitions and reset generation in the subsystem.

SSE-320 leverages Power Policy Units (PPU) for power control of the Bounded Regions (BR) in Access to the PPU is normally not required for normal operation because SSE-320 is architect mode, where the request to enter or leave a power state is managed and handshake using the PP. Therefore, the only time access to PPUs might be required is for Debug purposes. The PPUs are mapped to Secure address space as defined in System Control Peripheral Region.

Device interface handshake is performed by all PPUs when transitioning to WARM_RST. All PPUs are reset on Cold reset and reside in the PD_AON power domain.

The following table lists additional configuration of the PPUs along with the power domains and bounded regions that they control.

Table 3-3: PPU associations and configurations

-	PPU configuration				
Power domains controlled by the PPU	**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 PD_VMR2 PD_VMR3	SYSPPU (BR_SYS)	16 OPMODEs with independent use model	ON	OFF, 0	WARM_RST, ON, FULL_RET, MEM_RET, OFF
PD_CPU0, PD_CPU0EPU, PD_CPU0RAM, PD_CPU0TCM	CPU0PPU (BR_CPU0)	4 OPMODEs with independent use model	ON	OFF, 0	WARM_RST, ON, FUNC_RET, MEM_OFF, FULL_RET, LOGIC_RET, MEM_RET, OFF
PD_NPU0	NPU0PPU (BR_NPU0)	Not supported	ON	OFF, 0	WARM_RST, ON, OFF
PD_DEBUG	DEBUGPPU (BR_DEBUG)	Not supported	ON	OFF, 0	WARM_RST, ON, OFF

-	PPU configuration				
Power domains controlled by the PPU	**PPU ID (BR ID)**	**OPMODE support**	**Default dynamic transition enable**	**Default power policy, default operation policy**	**Dynamic and static support of power modes**
PD_AON	MGMTPPU (NA)	Not supported	ON	ON, 0	WARM_RST, ON, OFF

All PPUs are configured to perform device interface handshake when transiting from ON to WARM_RESET (WARM_RST_DEVREQEN_CFG = 1).

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

See Power up after Cold reset for more information.

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

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

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

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

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*.

3.9.4 Bounded region power modes

SSE-320 supports multiple valid power state combinations and state transitions of power domains in each Bounded Region.

3.9.4.1 PD_AON power modes

PD_AON supports only ON and WARM_RST power modes.

SSE-320 provides expansion power P-Channel (MGMTPWRP*) interface for PD_AON to allow integration of power aware logic into the expansion block.


Caution

Failure to abide by the following rules results in system deadlock.

- If the interface is not used, perform the following actions:
  - MGMTPWRPACTIVE and MGMTPWRPDENY must be tied low.
  - MGMTPWRPPREQ must be looped back to MGMTPWRPACCEPT.
- If the interface is used, the following behaviors are not allowed:
  - Denying a lower to higher power mode change request.
  - Denying WARM_RST to ON power mode change request.

The interface requests OFF, ON, and WARM_RST states on PSTATE.

Expansion logic can indicate ON and OFF state request on PACTIVE. All other PACTIVE state requests are converted to ON state.

3.9.4.2 BR_DEBUG power modes

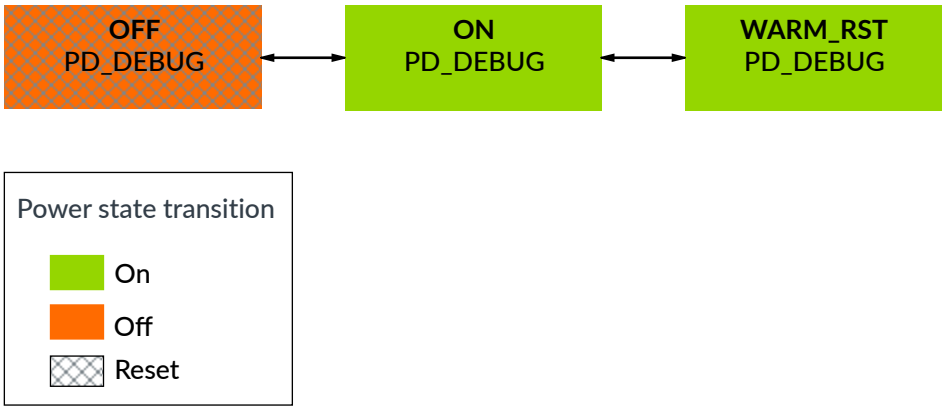
When a Warm reset is requested, the Bounded Region transitions to the WARM_RST Power Mode. The transition makes sure that the debug logic enters a safe state so that the system can be Warm reset cleanly.


Note

Logic in the PD_DEBUG power domain is not reset in the WARM_RST Power Mode.

The following figure shows the power modes supported by BR_DEBUG.

Figure 3-7: BR_DEBUG power mode transition diagram

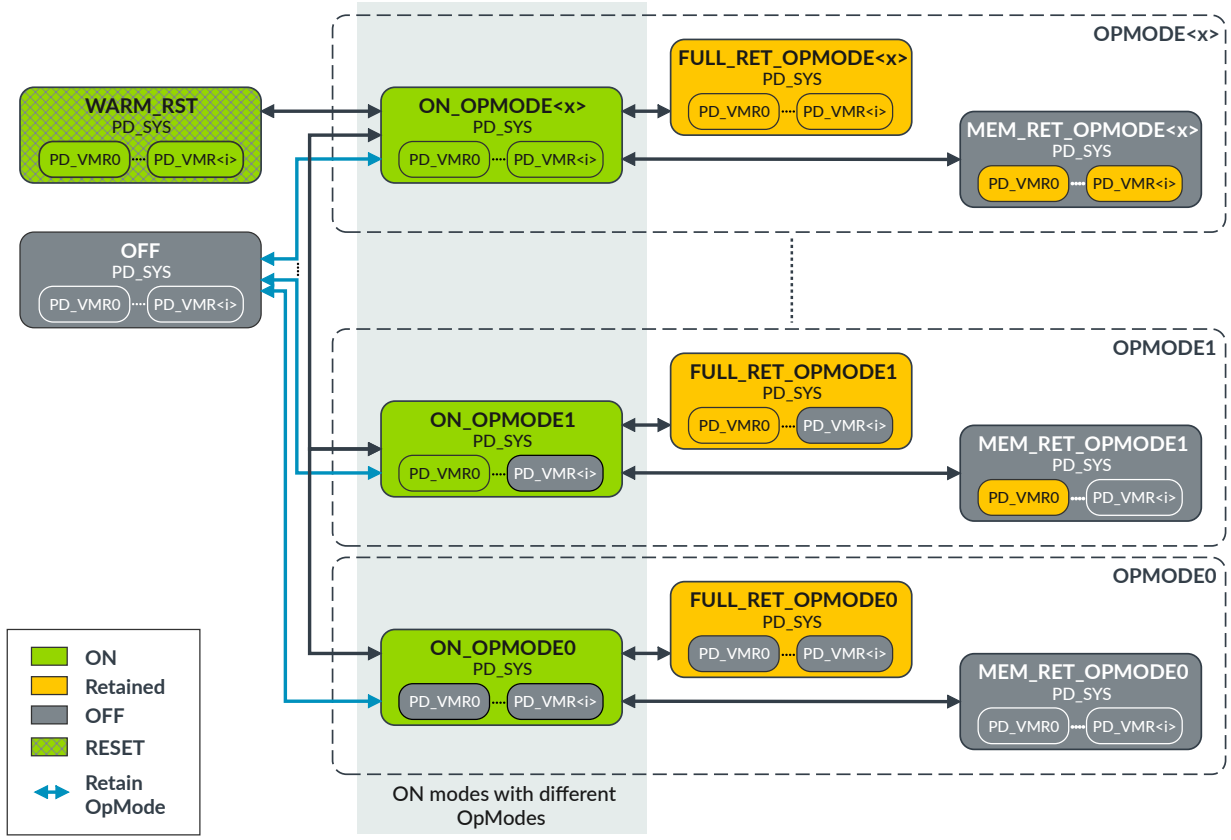


3.9.4.3 BR_SYS power modes

BR_SYS supports 16 power modes defined as OPMODE0 - OPMODE15. Changing OPMODEs is possible only when transitioning between the different ON power modes, except when entering and exiting the OFF or WARM_RST power modes.

The following figure shows the power modes that BR_SYS supports.

Figure 3-8: BR_SYS power mode transition diagram



In SSE-320, 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.

OPMODEs are encoded as binary 4-bit values with each bit representing the power state of a memory power domain, PD_VMR0, PD_VMR1, PD_VMR2 and PD_VMR3.

SSE-320 supports the following power modes:

OPMODE0

PD_VMR0, PD_VMR1, PD_VMR2 and PD_VMR3 are OFF

OPMODE1

PD_VMR0, PD_VMR1, PD_VMR2 are OFF and PD_VMR3 is ON[1]

OPMODE2

PD_VMR0 and PD_VMR1 are OFF, PD_VMR2 is ON[1] and PD_VMR3 is OFF

OPMODE3

PD_VMR0 and PD_VMR1 are OFF, PD_VMR2 and PD_VMR3 are ON[1]

OPMODE4

PD_VMR0 is OFF, PD_VMR1 is ON[1] , PD_VMR2 and PD_VMR3 are OFF

OPMODE5
PD_VMR0 is OFF, PD_VMR1 is ON[1], PD_VMR2 is OFF and PD_VMR3 is ON[1]

OPMODE6
PD_VMR0 is OFF, PD_VMR1 and PD_VMR2 are ON[1] and PD_VMR3 is OFF

OPMODE7
PD_VMR0 is OFF and PD_VMR1, PD_VMR2 and PD_VMR3 are ON[1]

OPMODE8
PD_VMR0 is ON[1] and PD_VMR1, PD_VMR2 and PD_VMR3 are OFF

OPMODE9
PD_VMR0 is ON[1], PD_VMR1 AND PD_VMR2 ARE off, PD_VMR3 is ON[1]

OPMODE10
PD_VMR0 is ON[1], PD_VMR1 is OFF, PD_VMR2 is ON[1] and PD_VMR3 is OFF

OPMODE11
PD_VMR0 is ON[1], PD_VMR1 is OFF, PD_VMR2 and PD_VMR3 is ON[1]

OPMODE12
PD_VMR0 and PD_VMR1 is ON[1], PD_VMR2 and PD_VMR3 are OFF

OPMODE13
PD_VMR0 and PD_VMR1 are ON[1], PD_VMR2 is OFF and PD_VMR3 is ON[1]

OPMODE14
PD_VMR0, PD_VMR1, PD_VMR2 are ON[1] and PD_VMR3 is OFF

OPMODE15
PD_VMR0, PD_VMR1, PD_VMR2 and PD_VMR3 are ON[1]

[1] Context saving power state depends on the PD_SYS power state when PD_SYS is in:

- ON or WARM_RST, then the context saving power state is ON.
- FULL_RET or MEM_RET, then the context saving power state is RET.

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

Related information

- NA

3.9.4.3.1 Controlling the PD_VMR<i> minimum power states

Software must configure the minimum power state of each PD_VMR<i>, as SSE-320 supports NUMVMBANK=4, there are PD_VMR0, PD_VMR1. PD_VMR2 and PD_VMR3 power domains.

To configure the minimum power state of each PD_VMR<i>, software must configure the register fields PDCM_PD_VMR<i>_SENSE.MIN_PWR_STATE as follows:

- Set to `0b0000` to set the minimum power state of the PD_VMR<i> to OFF.

  PD_VMR<i> only transitions between ON and OFF state, and is never in retention, meaning that in low-power state, all states in PD_VMR<i> are lost. With this setting, when PD_SYS is ON and the BR_SYS is in one of the ON_OPMODE<i> power modes where PD_VMR<i> is ON, BR_SYS transitions to another ON_OPMODE<i> where PD_VMR<i> is OFF automatically once the PD_VMR<i> domain is idle.

  Therefore, PD_VMR<i> is expected to turn OFF quickly once PDCM_PD_VMR<i>_SENSE.MIN_PWR_STATE is set to `0b0000`. Once PD_VMR<i> is OFF, it can only be returned to ON by an access on the bus targeting PD_VMR<i>.

  However, following that, when PD_VMR<i> is idle again, PD_VMR<i> turns OFF again. To avoid this, configure PDCM_PD_VMR<i>_SENSE.MIN_PWR_STATE to a non-zero value before accessing the PD_VMR<i>.
- Set to `0b0001` to set the minimum power state of the PD_VMR<i> to RET.

  PD_VMR<i> only ever transitions between ON and retention state, and never be in the OFF state. Therefore, in the low-power System Power States, all register states in PD_VMR<i> are retained. BR_SYS never transitions to a power mode that has PD_VMR<i> turned off.


Note

To place the PD_VMR<i> into Retention, the BR_SYS has to enter one of the FULL_RET_OPMODE<i> power modes or MEM_RET_OPMODE<i> power modes where PD_VMR<i> is in Retention. This means that it is not possible to place a PD_VMR<i> into Retention while keeping PD_SYS ON.

3.9.4.4 BR_CPU0 power modes

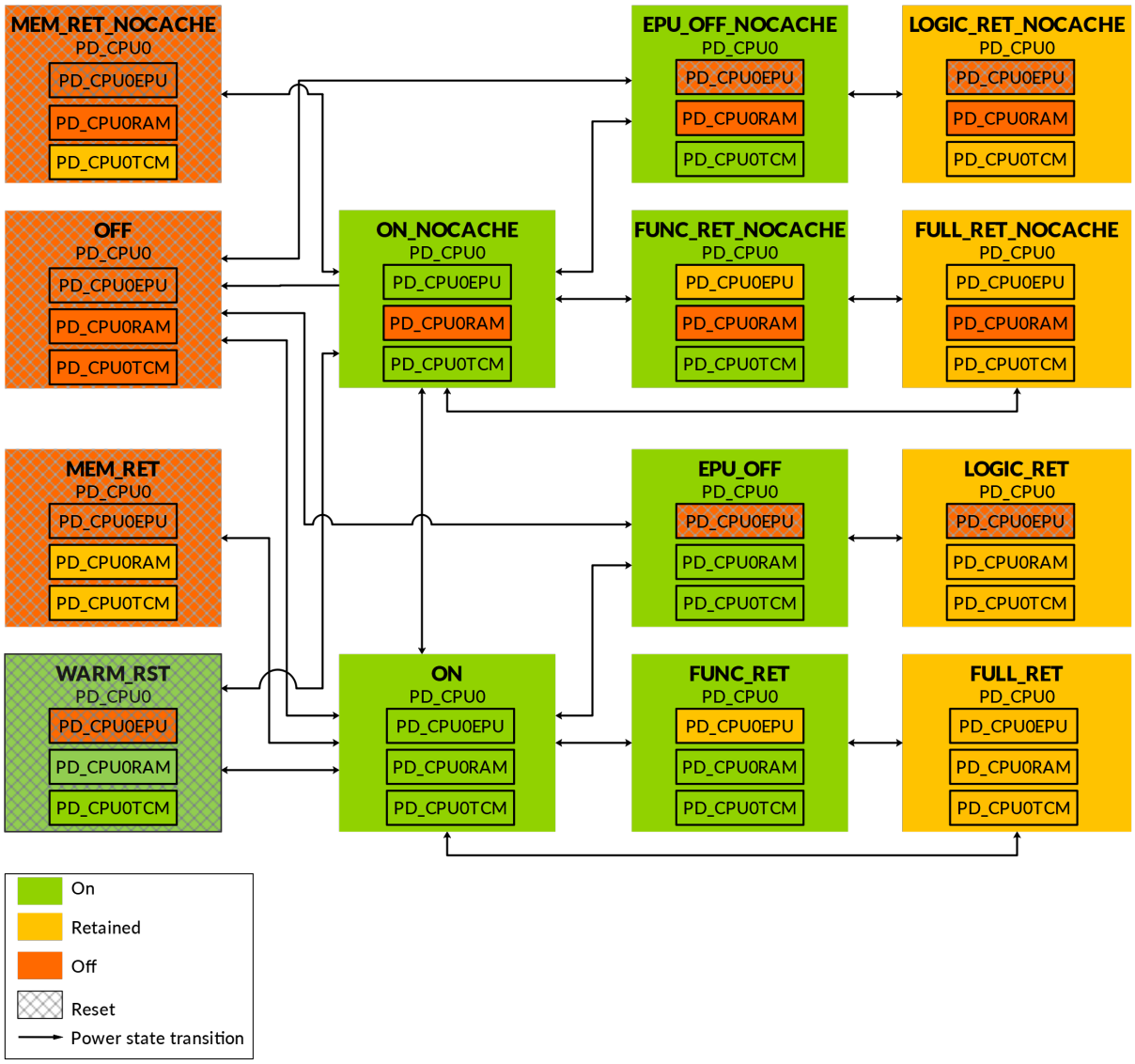
BR_CPU0 supports multiple power modes.

SSE-320 integrates Cortex-M85 logical top and does not split the CPU between power domains. For more information, see Figure 3-9: BR_CPU0 power mode transition diagram on page 87.

If you want to implement both the PD_CPU and the PD_DEBUG domains of the CPU as separate power domains in your SoC, then Arm recommends that you restructure the RTL to facilitate physical implementation:

- Move PDCORE of SSE-320 under the SSE-320 PD_CPU0 RTL hierarchy.
- Move PDDEBUG of Cortex-M85 under the SSE-320 PD_DEBUG RTL hierarchy.

Figure 3-9: BR_CPU0 power mode transition diagram



During pseudo power mode transitions, the physical power state of the power domains does not change. The power domains themselves, for example PD_CPU0, can observe pseudo transitions when the CPU0PPU exits the various MEM_RET power modes.

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

In SSE-320, logic retention power states are remapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells.

The EPU_OFF and EPU_OFF_NOCACHE power modes are mapped to MEM_OFF power mode of the PPU that controls BR_CPU0. Therefore, the MEM_OFF power mode of the BR_CPU0 PPU controls the power mode of logic in PD_CPU0EPU power domain, and not the power mode of a memory as the name might indicate.

The PD_CPU0TCM power state usually matches that of PD_CPU0. The exceptions are the power modes MEM_RET and MEM_RET_NOCACHE, where PD_CPU0TCM is retained. The choice of transitioning from ON_NOCACHE to either MEM_RET_NOCACHE or OFF is determined by the minimum power state register configuration CPUPWRCFG.TCM_MIN_PWR_STATE of the CPU0 local TCM.

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

3.9.4.4.1 Controlling PD_CPU0RAM power state

The BR_CPU0 bounded region uses operating modes to support the ability to turn on or off the cache RAMs in modes other than the OFF mode.

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

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

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

Transitions between the two operating modes can only occur between the ON and ON_NOCACHE power modes. In the NOCACHE operating modes, the PD_CPU0RAM is turned off. When operating in the CACHE operating modes, PD_CPU0RAM is retained in the MEM_RET, LOGIC_RET, or FULL_RET power modes.

3.9.4.4.2 Controlling PD_CPU0EPU power state

The PD_CPU0EPU and PD_CPU0 can only enter WARM_RST together. However, the BR_CPU0 bounded region lets PD_CPU0EPU enter a lower power state independently while the PD_CPU0 is ON.

To control whether the PD_CPU0EPU can be allowed to enter the retention (RET) or OFF state, the software can set the register CPDLPSTATE.ELPSTATE in the CPU as follows:

RET

Value: `0b10`

The EPU enters retention state only when in low power state. When CPDLPSTATE.ELPSTATE is set to RET, BR_CPU0 never enters EPU_OFF, EPU_OFF_NOCACHE, LOGIC_RET_NOCACHE, LOGIC_RET, OFF, and MEM_RET states.

OFF

Value: `0b11`

The EPU enters OFF state only when in lower power state. When CPDLPSTATE.ELPSTATE is set to OFF, BR_CPU0 never enters FUNC_RET, FUNC_RET _NOCACHE, FULL_RET_NOCACHE and FULL_RET states.

ON

Value: `0b00` or `0b01`

The EPU stays on. When CPDLPSTATE.ELPSTATE is set to ON, BR_CPU0 never enters any power modes to the right of the power modes ON_NOCACHE or ON in BR_CPU0 power mode. For details, see Figure 3-9: BR_CPU0 power mode transition diagram on page 87 except for FULL_RET and FULL_RET_NOCACHE.

3.9.4.4.3 Entering lower PD_CPU0 power states

For the PD_CPU0 to enter a lower power state, the software on the CPU0 must first configure its CPDLPSTATE.CLPSTATE register to define what power state it can enter when in a lower power state.

The register settings are as follows:

RET

Value: `0b10`

The PD_CPU0 enters retention state only when in low power state. When CPDLPSTATE.CLPSTATE is set to RET, BR_CPU0 never enters OFF, MEM_RET_NOCACHE, or MEM_RET.

OFF

Value: `0b11`

The PD_CPU0 enters off state only when in low power state. When CPDLPSTATE.CLPSTATE is set to OFF, BR_CPU0 never enters LOGIC_RET_NOCACHE, and LOGIC_RET modes. Other modes like FULL_RET and FULL_RET_NOCACHE can be entered depending on CPDLPSTATE.ELPSTATE and the current operating mode.

ON

Value: `0b00` or `0b01`

The PD_CPU0 stays on. When CPDLPSTATE.CLPSTATE is set to ON, BR_CPU0 never enters LOGIC_RET_NOCACHE, LOGIC_RET, FULL_RET_NOCACHE, FULL_RET, OFF, MEM_RET_NOCACHE, and MEM_RET modes.

For the PD_CPU0 to then enter a lower power state, the CPU must enter DEEPSLEEP enabled WFI state. In SSE-320, DEEPSLEEP always utilizes an EWIC. An *External Wakeup Interrupt Controller* (EWIC) for the CPU always exists, and the EWIC resides in the PD_AON domain. HASCPU0IWIC is 0 in SSE-320, so for CPU0, the *Internal Wakeup Interrupt Controller* (IWIC) does not exist, and the EWIC is always used.

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

The "OFF – DEEPSLEEP" state

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

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

The "RET – DEEPSLEEP" state

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

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

The "ON – DEEPSLEEP" state

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

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

The "ON – Sleep" state

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

The "ON" state

Is the CPU normal running state. In this state, the EPU and the RAMs in the CPU have a degree of separate control as detailed in the sections Controlling PD_CPU0RAM power state and Controlling PD_CPU0EPU power state respectively.

3.9.4.4.4 Wake-up sources

The PD_CPU0 power domain can be woken-up from low-power mode using certain wake-up sources.

The PD_CPU0 power domain can be woken-up by one of the following wake-up sources, these also wake-up the PD_SYS power domain via power hierarchy:

- Interrupts with EWIC support, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.
- CPU0EDBGRQ input.
- PWRCPU0WAKE Q-Channel Device interface.

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

- PWRMGMTWAKE Q-Channel Device interface for PD_AON. Given that PD_AON is always on, this interface should be tied LOW.
- PWRSYSWAKE Q-Channel Device interface for PD_SYS.
- PWRDEBUGWAKE Q-Channel Device interface for PD_DEBUG.

Related information

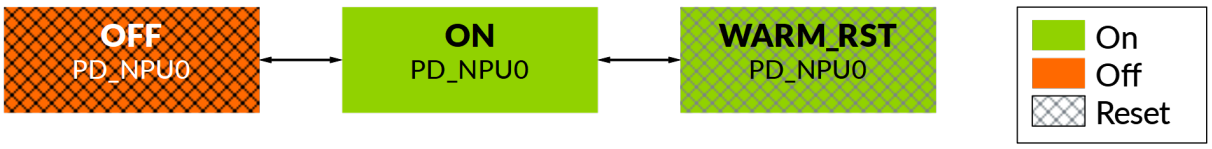
- NA

3.9.4.5 BR_NPU0 power modes

BR_NPU0 supports multiple power modes.

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

Figure 3-10: BR_NPU0 power mode transition diagram



The NPU has a power control register that can be programmed to prevent the system powering off.

After reset, once the NPU enters an idle state, it allows powering down of the BR_NPU0 domain.

3.9.5 Power dependency control

SSE-320 defines a control matrix that allows the power mode (ON state) of one domain to affect another power domain.

You can program the matrix and change how SSE-320 performs dynamic power transitions through PDCM registers PDCM_PD_SYS_SENSE and PDCM_PD_VMR<M>_SENSE.

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

Table 3-4: Power dependency control matrix on page 92 shows how the PD_SYS, PD_VMR0, and PD_VMR1 are affected by the power dependency inputs.

The right columns of the table list the power domains (PD_SYS, PD_VMR0, and PD_VMR1) that are being controlled.

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

- The sources of Power Domain ON Status Signals, PDSYSON, PDCPU0ON, and PDNPU0ON.
- The Q-Channel signals of Expansion Power Control Dependency interface, PDCMONQREQn, and PDCMRETQREQn

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

Table 3-4: Power dependency control matrix

Power Dependency Inputs/ Power Domain	PD_SYS	P_VMR0	PD_VMR1	P_VMR2	PD_VMR3
PD_SYS_ON	Conf	-	-	-	-
PD_CPU0_ON	Y	Conf	Conf	Conf	Conf
PD_NPU0_ON [1]	Y	Conf	Conf	Conf	Conf
PDCMONQREQn[{0-3}]	Conf	Conf	Conf	Conf	Conf
PDCMRETQREQn[{0-3}]	Conf	Conf	Conf	Conf	Conf

[1] PD_NPU0_ON row does not exist if NUMNPU = 0.

“Conf” indicates that it is software configurable.

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

PD_SYS

PD_SYS can be software configured to be sensitive to the ON state of PD_SYS and all Expansion Power Control Dependency inputs. PD_SYS is always sensitive to the ON state of PD_CPU0, PD_VMR0, PD_VMR1, PD_VMR2, PD_VMR3 and PD_NPU0.

When software configures PD_SYS to be sensitive to:

- PD_SYS_ON (self), then PD_SYS remains ON once it is ON.
- PDCMONQREQn, then once it is ON PD_SYS remains ON as long as PDCMONQREQn==PDCMONACCEPTn==1
- PDCMRETQREQn, then once it is ON PD_SYS changes state to FULL_RET instead of OFF/MEM_RET as long as PDCMRETQREQn==PDCMRETQACCEPTn==1

PD_VMR<i>

PD_VMR<i> can be software configured to be sensitive to the ON state of PD_CPU0, PD_NPU0, and all Expansion Power Control Dependency interface inputs.

When software configures PD_VMR<i> to be sensitive to:

- PD_CPU0_ON, then once it is ON, PD_VMR<i> remains ON as long as PD_CPU0_ON==1 (PD_CPU0 is ON)
- PD_NPU0_ON, then once it is ON, PD_VMR<i> remains ON as long as PD_NPU0_ON==1 (PD_NPU0 is ON)
- PDCMONQREQn, then once it is ON, PD_VMR<i> remains ON as long as PDCMONQREQn==PDCMONACCEPTn==1
- PDCMRETQREQn, then once it is ON, PD_VMR<i> will change state to RET instead of OFF as long as PDCMRETQREQn==PDCMRETQACCEPTn==1


Note

Because PD_VMR0, PD_VMR1, PD_VMR2 and PD_VMR3 are controlled by PD_SYS PPU, the PD_SYS PPU changes state to MEM_RET instead of OFF as long as PDCMRETQREQn for PD_VMR<i> is in effect, but this does not affect the power state of the logic in PD_SYS.

Power domain minimum power state

SSE-320 provides programmable registers for the PD_SYS and each of the PD_VMR<i> power domains PDCM_PD_SYS_SENSE and PDCM_PD_VMR<M>_SENSE, which define the lowest power state that each domain can enter.

The minimum power states of the power domains PD_DEBUG, PD_NPU0, and PD_CPU0 are not affected by the PDCM registers. The following table shows the minimum power states of the power domains.

Table 3-5: 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, the SYSPPU tries to enter the bounded region collectively to a low-power state if both the following conditions are met:

- PD_SYS is idle.
- All domains that PD_SYS depends on are not ON.

If MIN_PWR_STATE of PD_SYS is set to Retention, BR_SYS is not allowed to enter the OFF mode nor any of the MEM_RET_OPMODE<m> modes because PD_SYS is not allowed to turn off. SYSPPU tries to enter one of the associated FULL_RET_OPMODE<m> states.

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

- MIN_PWR_STATE of PD_VMR0 is ON.
- MIN_PWR_STATE of all other PD_VMR<i> is OFF.
- All other PD_VMR<i> are ON.

Then the SYSPPU transitions to ON_OPMODE1 state to turn off all other PD_VMR<i>. It never enters FULL_RET_OPMODE1 or MEM_RET_OPMODE1.

Related information

- NA

3.9.6 System power states

The system power states in SSE-320 are bounded by the relationships and minimum power states defined in the various Power Dependency Control Matrix Sensitivity registers and the low-power control registers in the CPU.

SSE-320 defines the following system power states:

SYS_OFF
: All voltage and power domains are OFF.

HIBERNATION0
: The voltage domain is ON, and the system is in the lowest power state that can still be woken from sleep. At wake, the system has to reboot.

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

SYS_ON
: The system is ON.

In SSE-320, logic retention power states are remapped to ON so that register values are persistent in power modes with logic retained even in an implementation without retention capable cells.

Table 3-6: System power states on page 95 defines the following for each System Power State:

- Supported power states of the power domains.

- Voltage supply state.
- Input clock state (SYSCLK, CPU0CLK and NPU0CLK).

The subsystem does not support state combinations that are not tabulated. For the definition of the states in the PD_CPU0 column, see Entering lower PD_CPU0 power states.

Table 3-6: System power states

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

[1] The subsystem requests clocks based on the Clock Force register and the power states of PD_DEBUG, PD_SYS, PD_NPU0, and PD_CPU0. The relationship of the clocks and the power states is defined in the Power State Based High-Level Clock Gating related descriptions in *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

[2] When PD_CPU0 is in the state OFF-DeepSleep, PD_CPU0TCM can be either OFF or retained as defined by CPUPWRCFG.TCM_MIN_PWR_STATE. In this state, PD_CPU0EPU must be OFF. The PD_CPU0RAM can be OFF permanently, which is defined by CPDLPSTATE.RLPSTATE.

The following points can be made, based on Table 3-6: System power states on page 95:

- When waking up any of the PD_CPU0 power domains, if the PD_SYS is OFF or RET, the PD_SYS domain is automatically woken to ON. This is because the PD_CPU0 ON state is only supported in the SYS_ON System Power State.
- PD_DEBUG can be woken independently, except in the SYS_OFF state.
- In HIBERNATION0 and SYS_RET, PD_VMR<i> cannot be woken independently from PD_SYS, since they belong to the bounded region BR_SYS.
- SSE-320 does not support debug scenarios when the processor is not in ON state. When EXPLOGIC_PRESENT=1 the provided example expansion logic wakes up PD_CPU0 to ON when PD_DEBUG is ON.

An additional transient WARM_RST state also exists for the system when a Warm reset is performed. This state can only be entered from the SYS_ON state with PD_CPU0 being ON. In this state, all power domains, including PD_AON, temporarily enter the WARM_RST power mode and exit it when Warm reset is completed.

3.9.6.1 Power up after Cold reset

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

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

1. PD_DEBUG and PD_SYS
2. PD_CPU0 and PD_NPU0

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

If there is no request towards PD_NPU0, when the power up sequence completes, the NPU0PPU turns off.

Power modes entered during the powerup:

SYSPPU
: ON_OPMODE0

CPU0PPU
: EPU_OFF_NOCACHE

NPU0PPU
: ON

DEBUGPPU
: ON

3.9.6.2 Entering HIBERNATION0

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

- PD_NPU0 must be in OFF state.
- The Minimum Power State in the registers PDCM_PD_VMR0_SENSE, PDCM_PD_VMR1_SENSE, PDCM_PD_VMR2_SENSE and PDCM_PD_VMR3_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, use one of the following settings:
  - Set sensitivity to the PD_CPU0 ON state in the related PDCM_PD_VMR<i>_SENSE register before entering HIBERNATION0.

    This ensures that on the first access to the VM after leaving HIBERNATION0 (which has caused the VM to power up) the VM stays powered until PD_CPU0 turns OFF.
  - Set the minimum power state in the PDCM_PD_VMR<i>_SENSE register to RET, so that the VM content is always retained once it is turned ON.
- If there is an Expansion logic in the PD_SYS that is connected to the PD_SYS Power P-Channel Interface, the expansion logic must be idle and able to enter a quiescent state.
- If EWIC is to wake the subsystem, PD_CPU0 must enter the “OFF-DeepSleep” with EWIC enabled. For more information, see Entering lower PD_CPU0 power states.

3.9.6.3 Wake from HIBERNATION0 using the PWRSYSWAKE Q-Channel Device Interface

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

This interface does not wake PD_CPU0 directly, accesses to the subsystem can result in a wake interrupt on the EWIC.

Note

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

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

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

If the Extension manager intends to wake and access peripherals or memories within SSE-320 without waking the CPU as well, the Extension manager accessing the subsystem must be a Secure manager. Alternatively, a Non-secure manager can be restricted externally to access a strictly controlled region of memory residing outside of the SSE-320, which is always Non-secure and therefore does not pose a security risk.

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

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

Arm does not recommend that you use only PWRSYSWAKEQACTIVE to wake the subsystem from HIBERNATION0, as it is limited. This is because when PD_SYS is awakened from HIBERNATION0, all registers in the power domain are in their reset state, and all peripherals that reside behind PPCs or MPCs defaults to Secure access only. Often, this requires waking and booting the CPU so that it can configure the subsystem before allowing accesses for Extension managers into the subsystem.

4. Programmer Model

The following sections describe the programmer model of SSE-320.

The following information applies to SSE-320 registers:

- The base address is not fixed and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can results in unexpected behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits
  - Ignore undefined register bits on reads
  - All register bits are reset to the reset value specified in the register summary table of each block

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

Table 4-1: Register bit field abbreviations

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

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

Register values are defined using actual values that are written in or read from the registers.

They are expressed as numbers, either as binary numbers or hexadecimal numbers. Alternatively, for single-bit values, they can be expressed as either 1 or 0, representing `0b1` and `0b0` respectively.

We recommend that all SSE-320 registers reside in the always-on power domain and are reset by the system Cold reset.

4.1 System Memory Map Overview

The High level system address map shows the high-level view of the memory map defined by SSE-320. This memory map is divided into Secure and Non-secure regions.

The memory alternates between Secure and Non-secure regions on 256Mbyte regions, with only a few address areas exempted from security mapping because they are related to debug functionality.

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

Except when specifically stated, the following occur:

- All access to unmapped regions of the memory result in bus-error response.
- When accessing unmapped address space within a mapped region taken by a peripheral, the access results in *Read-As-Zero and Write-Ignored* (**RAZ/WI**) except when specifically stated otherwise.
- Any accesses that result in security violations are 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 `0x2100_0000` to `0x21FF_FFFF` and `0x3100_0000` to `0x31FF_FFFF` support exclusive access since they implement exclusive access monitoring, provided the accesses are from:

- CPU0
- External managers through the Subordinate Main Expansion Interfaces
- Subordinate peripheral expansion interfaces

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

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

4.1.1 High level system address map

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

The System Address Map defines the following values for accessibility:

S
: Secure Access

NS
: Non-secure Access

NSC
: Non-secure Callable

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

Table 4-2: High-Level System Address Map

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
1	`0x0000_0000` - `0x00FF_FFFF`	16MB	ITCM	5	Security: NS IDAUID: 0 NSC: 0	CPU Instruction TCM See CPU TCM memories
2	`0x0100_0000` - `0x09FF_FFFF`	144MB	Code Expansion	6	Security: NS IDAUID: 0 NSC: 0	Manager Code Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
2.1	`0x0A00_0000` - `0x0AFF_FFFF`	16MB	CPU0 ITCM	6.1	Security: NS IDAUID: 0 NSC: 0	CPU0 S-AHB Instruction TCM Access See TCM Subordinate interface section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
2.2	`0x0B00_0000` - `0x0BFF_FFFF`	16MB	Reserved	6.2	Security: NS IDAUID: 0 NSC: 0	Reserved

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
2.3	0x0C00_0000 - 0x0CFF_FFFF	16MB	Reserved	6.3	Security: NS IDAUID: 0 NSC: 0	Reserved
2.4	0x0D00_0000 - 0x0DFF_FFFF	16MB	Reserved	6.4	Security: NS IDAUID: 0 NSC: 0	Reserved
3	0x0E00_0000 - 0x0E00_1FFF	8KB	Reserved	7	Security: NS IDAUID: 0 NSC: 0	Reserved
4	0x0E00_2000 - 0x0FFF_FFFF	-	Reserved	-	Security: NS IDAUID: 0 NSC: 0	Reserved
5	0x1000_0000 - 0x10FF_FFFF	16MB	ITCM	1	Security: S IDAUID: 1 NSC: CODENSC	CPU Instruction TCM See CPU TCM memories
5.1	0x1100_0000 - 0x11FF_FFFFF	16MB	ROM	-	Security: S IDAUID: 1 NSC: CODENSC	Boot ROM See ROM.
6	0x1200_0000 - 0x19FF_FFFF	128MB	Code Expansion	2	Security: S IDAUID: 1 NSC: CODENSC	Manager Code Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
6.1	0x1A00_0000 - 0x1AFF_FFFFF	16MB	CPU0 ITCM	2.1	Security: S IDAUID: 1 NSC: CODENSC	CPU0 S-AHB Instruction TCM Access See TCM Subordinate interface section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
6.2	0x1B00_0000 - 0x1BFF_FFFF	16MB	Reserved	2.2	Security: S IDAUID: 1 NSC: CODENSC	Reserved
6.3	0x1C00_0000 - 0x1CFF_FFFF	16MB	Reserved	2.3	Security: S IDAUID: 1 NSC: CODENSC	Reserved
6.4	0x1D00_0000 - 0x1DFF_FFFF	16MB	Reserved	2.4	Security: S IDAUID: 1 NSC: CODENSC	Reserved
7	0x1E00_0000 - 0x1E00_1FFF	8KB	Reserved	3	Security: S IDAUID: 1 NSC: CODENSC	Reserved
8	0x1E00_2000 - 0x1FFFFFFF	-	Reserved	-	Security: S IDAUID: 1 NSC: CODENSC	Reserved
9	0x2000_0000 - 0x20FF_FFFF	16MB	DTCM	13	Security: NS IDAUID: 2 NSC: 0	CPU Data TCM See CPU TCM memories
10	0x2100_0000 - 0x21FF_FFFF	16MB	Volatile Memory	14	Security: NS IDAUID: 2 NSC: 0	Volatile Memory See Volatile Memory Region
11	0x2200_0000 - 0x23FF_FFFF	32MB	Reserved	-	Security: NS IDAUID: 2 NSC: 0	Reserved

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
11.1	0x2400_0000 - 0x24FF_FFFF	16MB	CPU0 DTCM	15.1	Security: NS IDAUID: 2 NSC: 0	CPU0 S-AHB Data TCM Access
11.2	0x2500_0000 - 0x25FF_FFFF	16MB	Reserved	15.2	Security: NS IDAUID: 2 NSC: 0	Reserved
11.3	0x2600_0000 - 0x26FF_FFFF	16MB	Reserved	15.3	Security: NS IDAUID: 2 NSC: 0	Reserved
11.4	0x2700_0000 - 0x27FF_FFFF	16MB	Reserved	15.4	Security: NS IDAUID: 2 NSC: 0	Reserved
12	0x2800_0000 - 0x2FFF_FFFF	128MB	Main Expansion	-	Security: NS IDAUID: 2 NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
13	0x3000_0000 - 0x30FF_FFFF	16MB	DTCM	9	Security: S IDAUID: 3 NSC: RAMNSC	CPU Data TCM See CPU TCM memories
14	0x3100_0000 - 0x31FF_FFFF	16MB	Volatile memory	10	Security: S IDAUID: 3 NSC: RAMNSC	Internal Multi-bank Volatile Memory See Volatile Memory Region
15	0x3200_0000 - 0x33FF_FFFF	32MB	Reserved	-	Security: S IDAUID: 3 NSC: RAMNSC	Reserved

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
15.1	0x3400_0000 - 0x34FF_FFFF	16MB	CPU0 DTCM	11.1	Security: S IDAUID: 3 NSC: RAMNSC	CPU0 S-AHB Data TCM Access
15.2	0x3500_0000 - 0x35FF_FFFF	16MB	Reserved	11.2	Security: S IDAUID: 3 NSC: RAMNSC	Reserved
15.3	0x3600_0000 - 0x36FF_FFFF	16MB	Reserved	11.3	Security: S IDAUID: 3 NSC: RAMNSC	Reserved
15.4	0x3700_0000 - 0x37FF_FFFF	16MB	Reserved	11.4	Security: S IDAUID: 3 NSC: RAMNSC	Reserved
16	0x3800_0000 - 0x3FFF_FFFF	128MB	Main Expansion	-	Security: S IDAUID: 3 NSC: RAMNSC	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
17	0x4000_0000 - 0x4000_FFFF	64KB	Peripherals	27	Security: NS IDAUID: 4 NSC: 0	Peripheral Region See Peripheral Region
18	0x4001_0000 - 0x4001_FFFF	64KB	Private CPU	-	Security: NS IDAUID: 4 NSC: 0	CPU Private Peripheral Region See Processor Private Region
19	0x4002_0000 - 0x4003_FFFF	128KB	System Control Peripheral Region.	-	Security: NS IDAUID: 4 NSC: 0	System Control Peripheral Region See System Control Peripheral Region

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
20	0x4004_0000 - 0x400F_FFFF	768KB	Peripherals	-	Security: NS IDAUID: 4 NSC: 0	Peripheral Region See System Control Peripheral Region
21	0x4010_0000 - 0x47FF_FFFF	127MB	Peripheral Expansion	-	Security: NS IDAUID: 4 NSC: 0	Manager Peripheral Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
22	0x4800_0000 - 0x4800_FFFF	64KB	Peripherals	32	Security: NS IDAUID: 4 NSC: 0	Peripheral Region See Peripheral Region
23	0x4801_0000 - 0x4801_FFFF	64KB	Private CPU	-	Security: NS IDAUID: 4 NSC: 0	CPU Private Peripheral Region See Processor Private Region
24	0x4802_0000 - 0x4803_FFFF	128KB	System Control	-	Security: NS IDAUID: 4 NSC: 0	System Control Peripheral Region See System Control Peripheral Region
25	0x4804_0000 - 0x480F_FFFF	768KB	Peripherals	-	Security: NS IDAUID: 4 NSC: 0	Peripheral Region See Peripheral Region
26	0x4810_0000 - 0x4FFF_FFFF	127MB	Peripheral Expansion	-	Security: NS IDAUID: 4 NSC: 0	Manager Peripheral Expansion Interface See Peripheral Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual* and Peripheral Expansion Region
27	0x5000_0000 - 0x5000_FFFF	64KB	Peripherals	17	Security: S IDAUID: 5 NSC: 0	Peripheral Region See Peripheral Region

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
28	0x5001_0000 - 0x5001_FFFF	64KB	Private CPU	-	Security: S IDAUID: 5 NSC: 0	CPU Private Peripheral Region See Processor Private Region
29	0x5002_0000 - 0x5003_FFFF	128KB	System Control	-	Security: S IDAUID: 5 NSC: 0	System Control Peripheral Region See System Control Peripheral Region
30	0x5004_0000 - 0x500F_FFFF	786KB	Peripherals	-	Security: S IDAUID: 5 NSC: 0	Peripheral Region See Peripheral Region
31	0x5010_0000 - 0x57FF_FFFF	127MB	Peripheral Expansion	-	Security: S IDAUID: 5 NSC: 0	Manager Peripheral Expansion Interface See Peripheral Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
32	0x5800_0000 - 0x5800_FFFF	64KB	Peripherals	22	Security: S IDAUID: 5 NSC: 0	Peripheral Region See Peripheral Region
33	0x5801_0000 - 0x5801_FFFF	64KB	Private CPU	-	Security: S IDAUID: 5 NSC: 0	Processor Private Peripheral Region See Processor Private Region
34	0x5802_0000 - 0x5803_FFFF	128KB	System Control	-	Security: S IDAUID: 5 NSC: 0	System Control Peripheral Region See System Control Peripheral Region
35	0x5804_0000 - 0x580F_FFFF	768KB	Peripherals	-	Security: S IDAUID: 5 NSC: 0	Peripheral Region See System Control Peripheral Region
36	0x5810_0000 - 0x5FFF_FFFF	127MB	Peripheral Expansion	-	Security: S IDAUID: 5 NSC: 0	Manager Peripheral Expansion Interface See Peripheral Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual* and Peripheral Expansion Region

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
37	0x6000_0000 - 0x6FF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: 6 NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
38	0x7000_0000 - 0x7FFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: 7 NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
39	0x8000_0000 - 0x8FF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: 8 NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
40	0x9000_0000 - 0x9FFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: 9 NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
41	0xA000_0000 - 0xAFFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: A NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
42	0xB000_0000 - 0xBFFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: B NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
43	0xC000_0000 - 0xCFFF_FFFF	256MB	Main Expansion	-	Security: NS IDAUID: C NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
44	0xD000_0000 - 0xDFFF_FFFF	256MB	Main Expansion	-	Security: S IDAUID: D NSC: 0	Manager Main Expansion Interface See Main Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
45	0xE000_0000 - 0xE00F_FFFF	1MB	PPB	-	Security: NS IDAUID: E NSC: 0	CPU Private Peripheral Bus Region. Local to the CPU See CPU Private Peripheral Bus Region

Row ID	Address	Size	Region	Alias	IDAU Region Values	Description
46	0xE010_0000 - 0xE01F_FFFF	1MB	Debug System	49	Security: NS IDAUID: E NSC: 0	Debug System Access Region See Debug System Access Region
47	0xE020_0000 - 0xEFFF_FFFF	254MB	Peripheral Expansion	-	Security: NS IDAUID: E NSC: 0	Manager Peripheral Expansion Interface See Peripheral Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*
48	0xF000_0000 - 0xF00F_FFFF	1MB	Reserved	-	Exempt	Reserved
49	0xF010_0000 - 0xF01F_FFFFF	1MB	Debug System	46	Security: S IDAUID: F NSC: 0	Debug System Access Region See Debug System Access Region
50	0xF020_0000 - 0xFFFF_FFFF	254MB	Peripheral Expansion	-	Security: S IDAUID: F NSC: 0	Manager Peripheral Expansion Interface See Peripheral Interconnect Expansion Interfaces section of *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*

4.2 CPU TCM memories

The CPU0 in SSE-320 is configured to implement *Tightly Coupled Memories* (TCM) for Instruction and Data.

These memories reside in the following location from the perspective of the CPU0 core:

- `0x0000_0000` to `0x00FF_FFFF` and `0x1000_0000` to `0x10FF_FFFF` for Instruction TCM. Both regions are aliased, and each provides up to 16MB of TCM space.
- `0x2000_0000` to `0x20FF_FFFF` and `0x3000_0000` to `0x30FF_FFFF` for Data TCM. Both regions are aliased, and each provides up to 16MB of TCM space.

Each CPU has access to its own local TCMs through its private address and other CPUs TCMs through the Main Interconnect. A CPU does not have access to its own TCMs through the Main Interconnect. Other managers on the Main interconnect have access to the TCMs. A TCM DMA subordinate interface to allow expansion managers to access the TCMs is provided.

For more details, see the TCM Subordinate interface section in *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.


Note

While a CPU can directly access the TCM memory of another CPU through the Main Interconnect using the remapped TCM regions. These TCMs are in reality provided only for use by the CPU that owns the TCM. As a result, these TCMs should not be used as shared memory between CPUs. If there is a need to move data from one TCMs to another, we recommend either using DMA, or software emulating DMA, to move blocks of data from one TCM to another for processing after determining that it is safe to move the data.

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.

4.3 ROM

SSE-320 supports an **IMPLEMENTATION DEFINED** ROM bank that contains the First Stage Boot Loader (FSBL).

If the ROM is implemented, the ROM size is defined according to ROMADDRWIDTH configuration parameter. For detais, see *sse320-RM?*. The ROM is not aliased, that is, used for Secure access only, and located within the ROM region as defined in System Memory Map overview.

The ROM resides in the following location within a 16MB ROM region:

- `0x1100_0000` to `0x11FF_FFFF` for Secure access

Any memory areas in that region that are not utilized are reserved and return a bus error response if accessed.

4.4 Volatile memory region

SSE-320 supports four internal *Volatile Memory* (VM) Banks. These are implemented as SRAMs.

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

The following table shows an example, where four 512KB memory banks are configured with 2 MB total size.

Table 4-3: Volatile memory region example address map with striped VM2 and VM3

Row ID	From address	To address	Size	Region name	Alias with row ID	Security[1]	Description
1	0x2100_0000	0x2107_FFFF	512KB	VM0	5	NS_MPC	Maps to Internal Volatile Memory Bank 0
2	0x2108_0000	0x210F_FFFF	512KB	VM1	6	NS_MPC	Maps to Internal Volatile Memory Bank 1
3	0x2110_0000	0x211F_FFFF	1MB	Striped VM2 + VM3	7	NS_MPC	Maps to Internal Volatile Memory Bank 2 + 3
4	0x2120_0000	0x21FF_FFFF	14MB	Reserved	-	-	Reserved
5	0x3100_0000	0x3107_FFFF	512KB	VM0	1	S_MPC	Maps to Internal Volatile Memory Bank 0
6	0x3108_0000	0x310F_FFFF	512KB	VM1	2	S_MPC	Maps to Internal Volatile Memory Bank 1
7	0x3110_0000	0x311F_FFFF	1MB	Striped VM2 + VM3	3	S_MPC	Maps to Internal Volatile Memory Bank 2 + 3
8	0x3120_0000	0x31FF_FFFF	14MB	Reserved	-	-	Reserved

[1] Legend

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

The VMMPCBLKSIZE configuration parameter sets the block-size of the MPC that configures the granularity for the software to define Secure and Non-Secure regions in the Volatile Memory. For details of configuration parameters, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

4.5 Peripheral region

The Peripheral Regions are memory regions where peripherals of the system reside.

There are eight regions as follows:

0x4000_0000 to 0x4000_FFFF

Non-secure region for low-latency peripherals that are expected to be aliased in its associated Secure region, 0x5000_0000 to 0x5000_FFFF.

0x4004_0000 to 0x400F_FFFF

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

0x4800_0000 to 0x4800_FFFF

Non-secure region for high-latency peripherals that are expected to be aliased in its associated Secure region, 0x5800_0000 to 0x5800_FFFF.

0x4804_0000 to 0x480F_FFFF

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

0x5000_0000 to 0x5000_FFFF
: Secure region for low-latency peripherals that are expected to be aliased in its associated Non-secure region, `0x4000_0000` to `0x4000_FFFF`.

0x5004_0000 to 0x500F_FFFF
: Secure region for low-latency peripherals that are expected to be not aliased.

0x5800_0000 to 0x5800_FFFF
: Secure region for high-latency peripherals that are expected to be aliased in its associated Non-secure region, `0x4800_0000` to `0x4800_FFFF`.

0x5804_0000 to 0x580F_FFFF
: 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.

For more details, see Secure access configuration register block.

Caution

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

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

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
: Privileged access only

UP
: Unprivileged and privileged access allowed

P_PPC
: Privileged access controlled by a PPC

Table 4-4: Peripheral Region Address Map

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
1	0x4000_0000 - 0x4000_0FFF	4KB	Reserved	Reserved (**RAZ/WI**)	23	NS_PPC, P_PPC
2	0x4000_1000 - 0x4000_1FFF	4KB	Reserved	Reserved (**RAZ/WI**)	24	
3	0x4000_2000 - 0x4000_3FFF	8KB	DMA	DMA Configuration interface	25	NS_PPC, P_PPC
4	0x4000_4000 - 0x4000_5FFF	8KB	NPU0	NPU0 Configuration interface **Note:** if NPUNUM >0 and Reserved if NPUNUM =0	26	NS_PPC, P_PPC
5	0x4000_6000 - 0x4000_7FFF	8KB	Reserved	Reserved	27	-
6	0x4000_8000 - 0x4000_9FFF	-	Reserved	Reserved	28	-
7	0x4000_A000 - 0x4000_BFFF	-	Reserved	Reserved	29;	-
8	0x4000_C000 - 0x4000_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
9	0x4004_0000 - 0x4007_FFFF	-	Reserved	Reserved	-	-
10	0x4008_0000 - 0x4008_0FFF	4KB	NSACFG	Non-secure Access Configuration register block. See Non-secure Access Configuration register block.	-	NS_PPC, P, P_PPC
11	0x4008_1000 - 0x4008_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
12	0x4009_0000 - 0x4009_3FFF	16KB	Reserved	Reserved (**RAZ/WI**)	-	NS
13	0x4009_4000 - 0x400F_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
14	0x4800_0000 - 0x4800_0FFF	4KB	TIMER0	Timer 0. See Timestamp Timers.	43	NS_PPC, P_PPC
15	0x4800_1000 - 0x4800_1FFF	4KB	TIMER1	Timer 1. See Timestamp timers.	44	
16	0x4800_2000 - 0x4800_2FFF	4KB	TIMER2	Timer 2. See Timestamp timers.	45	
17	0x4800_3000 - 0x4800_3FFF	4KB	TIMER3	Timer 3. See Timestamp timers.	46	
18	0x4800_F000 - 0x4800_FFFF	4KB	Non-secure SDC 600	SDC-600 Internal APBCOM. See Secure Debug Channel registers. If SDC-600 does not exist, this area is Reserved (**RAZ/WI**)	47	
19	0x4804_0000 - 0x4804_0FFF	4KB	NSWDCTRL	Non-secure Watchdog Control Frame. See Timestamp watchdogs.	-	NS_PPC, P_PPC, P
20	0x4804_1000 - 0x4804_1FFF	4KB	NSWDREF	Non-secure Watchdog Refresh Frame. See Timestamp watchdogs.	-	
21	0x4804_2000 - 0x4804_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
22	0x4805_0000 - 0x480F_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
23	0x5000_0000 - 0x5000_0FFF	4KB	Reserved	Reserved (**RAZ/WI**)	-	-
24	0x5000_1000 - 0x5000_1FFF	4KB	Reserved	Reserved (**RAZ/WI**)	-	-
25	0x5000_2000 - 0x5000_3FFF	8KB	DMA	DMA Configuration interface See DMA registers.	3	S, UP
26	0x5000_4000 - 0x5000_5FFF	8KB	NPU0	NPU0 Configuration interface **Note:** if NPUNUM >0 and Reserved if NPUNUM =0 See NPU0 registers.	4	S_PPC, P_PPC
27	0x5000_6000 - 0x5000_7FFF	4KB	Reserved	Reserved	5	-
28	0x5000_8000 - 0x5000_9FFF	4KB	Reserved	Reserved	6	-
29	0x5000_A000 - 0x5000_BFFF	4KB	Reserved	Reserved	7	-
30	0x5000_C000 - 0x5000_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
31	0x5004_0000 - 0x5007_FFFF	-	Reserved	Reserved	-	-
32	0x5008_0000 - 0x5008_0FFF	4KB	SACFG	Secure Access Configuration register block. See Secure Access Configuration register block.	-	NS_PPC, P_PPC, P
33	0x5008_1000 - 0x5008_2FFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
34	0x5008_3000 - 0x5008_3FFF	4KB	VM0MPC	VM0 Memory Protection Controller. See Volatile memory region.	-	NS_PPC, P_PPC, P
35	0x5008_4000 - 0x5008_4FFF	4KB	VM1MPC	VM1 Memory Protection Controller. See Volatile memory region.	-	
36	0x5008_5000 - 0x5008_5FFF	4KB	VM2PC	VM2 Memory Protection Controller. See Volatile memory region.	-	
37	0x5008_6000 - 0x5008_6FFF	4KB	VM2PC	VM3 Memory Protection Controller. See Volatile memory region.	-	
38	0x5008_7000 - 0x5009_DFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
39	0x5009_E000 - 0x5009_EFFF	4KB	KMU	Key Management Unit.	-	S_PPC, P_PPC
40	0x5009_F000 - 0x5009_FFFF	4KB;	Reserved	Reserved	-	-
41	0x500A_0000 - 0x500A_FFFF	64KB	LCM	See Lifecycle Manager	-	S_PPC, P_PPC
42	0x500B_0000 - 0x57FF_FFFF	-	Reserved	Reserved	-	-

Row ID	Address	Size	Region name	Description	Alias with row ID	Security
43	0x5800_0000 - 0x5800_0FFF	4KB	TIMER0	Timer 0. See Timestamp timers.	14	S_PPC, P_PPC
44	0x5800_1000 - 0x5800_1FFF	4KB	TIMER1	Timer 1. See Timestamp timers.	15	
45	0x5800_2000 - 0x5800_2FFF	4KB	TIMER2	Timer 2. See Timestamp timers.	16	
46	0x5800_3000 - 0x5800_3FFF	4KB	TIMER3	Timer 3. See Timestamp timers.	17	
47	0x5800_F000 - 0x5800_FFFF	4KB;	Secure SDC600	SDC-600 Internal APBCOM. See Secure Debug Channel registers. If SDC-600 does not exist, this area is Reserved (**RAZ/WI**)	18	
48	0x5804_0000 - 0x5804_0FFF	4KB	SWDCTRL	Secure Watchdog Control Frame. See Timestamp watchdogs.	-	S_PPC, P_PPC
49	0x5804_1000 - 0x5804_1FFF	4KB	SWDREF	Secure Watchdog Refresh Frame. See Timestamp watchdogs.	-	
50	0x5804_2000 - 0x5804_2FFF	4KB	SAM	See Security Alarm Manager	-	S_PPC, P_PPC
51	0x5804_3000 - 0x5804_FFFF	-	Reserved	Reserved (**RAZ/WI**)	-	-
52	0x5805_0000 - 0x580F_FFFF	-	Reserved	Reserved	-	-

4.5.1 Secure Access Configuration register block

The Secure Access Configuration Register Block implements program-visible states that allow software to control security gating units within the design. This register block base address is 0x5008_0000. This register block is Secure Privileged access only and supports 32 bit R/W accesses.

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

All registers reside in the PD_SYS power domain and is reset by nWARMRESETSYS.

Details of each register in the following table are described in separate sections, or in the relevant component documentation. The width of all registers is 32-bit.

Table 4-5: Secure Access Configuration Register Block Register Map

Offset	Name	Type	Reset value	Description
0x000	SPCSECCTRL	RW	0x0000_0000	Secure Privilege Controller Secure Configuration Control register
0x004	BUSWAIT	RW	Configurable	Bus Access Wait control after reset
0x008	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x010	SECRESPCFG	RW	0x0000_0000	Security Violation Response Configuration Register
0x014	NSCCFG	RW	0x0000_0000	Non-secure Callable Configuration for IDAU

Offset	Name	Type	Reset value	Description
0x018	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x01C	SECMPCINTSTAT	RO	0x0000_0000	Secure MPC interrupt Status
0x020	SECPPCINTSTAT	RO	0x0000_0000	Secure PPC nterrupt Status
0x024	SECPPCINTCLR	RW	0x0000_0000	Secure PPC interrupt Clear
0x028	SECPPCINTEN	RW	0x0000_0000	Secure PPC interrupt Enable
0x02C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x030	SECMSCINTSTAT	RO	0x0000_0000	Secure MSC interrupt Status
0x034	SECMSCINTCLR	RW	0x0000_0000	Secure MSC interrupt Clear
0x038	SECMSCINTEN	RW	0x0000_0000	Secure MSC interrupt Enable
0x03C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x040	BRGINTSTAT	RO	0x0000_0000	Bridge Buffer Error interrupt Status
0x044	BRGINTCLR	RW	0x0000_0000	Bridge Buffer Error interrupt Clear
0x048	BRGINTEN	RW	0x0000_0000	Bridge Buffer Error interrupt Enable
0x04C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x050	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x054 - 0x05C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x060	MAINNSPPCEXP0	RW	0x0000_0000	Expansion 0 Non-secure Access Peripheral Protection Control on the Main interconnect
0x064	MAINNSPPCEXP1	RW	0x0000_0000	Expansion 1 Non-secure Access Peripheral Protection Control on the Main interconnect
0x068	MAINNSPPCEXP2	RW	0x0000_0000	Expansion 2 Non-secure Access Peripheral Protection Control on the Main interconnect
0x06C	MAINNSPPCEXP3	RW	0x0000_0000	Expansion 3 Non-secure Access Peripheral Protection Control on the Main interconnect
0x070	PERIPHNSPPC0	RW	0x0000_0000	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
0x074	PERIPHNSPPC1	RW	0x0000_0000	Non-secure Access Peripheral Protection Control 1 on Peripheral interconnect
0x078	NPUSPPORSL	RW	Configurable	Secure Access NPU security level reset state control: Each Field Controls the PORSL inputs for an associated NPU.
0x07C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x080	PERIPHNSPPCEXP0	RW	0x0000_0000	Expansion 0 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x084	PERIPHNSPPCEXP1	RW	0x0000_0000	Expansion 1 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x088	PERIPHNSPPCEXP2	RW	0x0000_0000	Expansion 2 Non-secure Access Peripheral Protection Control on Peripheral Bus
0x08C	PERIPHNSPPCEXP3	RW	0x0000_0000	Expansion 3 Non-secure Access Peripheral Protection Control on Peripheral Bus

Offset	Name	Type	Reset value	Description
0x090	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x094 - 0x09C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x0A0	MAINSPPPCEXP0	RW	0x0000_0000	Expansion 0 Secure Unprivileged Access Peripheral Protection Control on Main Interconnect
0x0A4	MAINSPPPCEXP1	RW	0x0000_0000	Expansion 1 Secure Unprivileged Access Peripheral Protection Control on Main interconnect
0x0A8	MAINSPPPCEXP2	RW	0x0000_0000	Expansion 2 Secure Unprivileged Access Peripheral Protection Control on Main interconnect
0x0AC	MAINSPPPCEXP3	RW	0x0000_0000	Expansion 3 Secure Unprivileged Access Peripheral Protection Control on Main interconnect
0x0B0	PERIPHSPPPC0	RW	0x0000_0000	Secure Unprivileged Access Peripheral Protection Control 0 on Peripheral interconnect
0x0B4	PERIPHSPPPC1	RW	0x0000_0000	Secure Unprivileged Access Peripheral Protection Control 1 on Peripheral interconnect
0x0B8	NPUSPPORPL	RW	Configurable	NPU power on reset Secure privilege level control. Each Field Controls the PORPL inputs of the associated NPU.
0x0BC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x0C0	PERIPHSPPPCEXP0	RW	0x0000_0000	Expansion 0 Secure Unprivileged Access Peripheral Protection Control on Peripheral interconnect
0x0C4	PERIPHSPPPCEXP1	RW	0x0000_0000	Expansion 1 Secure Unprivileged Access Peripheral Protection Control on Peripheral interconnect
0x0C8	PERIPHSPPPCEXP2	RW	0x0000_0000	Expansion 2 Secure Unprivileged Access Peripheral Protection Control on Peripheral interconnect
0x0CC	PERIPHSPPPCEXP3	RW	0x0000_0000	Expansion 3 Secure Unprivileged Access Peripheral Protection Control on Peripheral interconnect
0x0D0	NSMSCEXP	RW	Configurable	Expansion MSC Non-secure Configuration
0x0D4 - 0xFCC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0xFD0	PIDR4	RO	0x0000_0004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_0052	Peripheral ID 0
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_003B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.5.1.1 SPCSECCTRL, Security Privilege Controller Security Configuration Control register

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x000

Type

RW

Reset value

0x0000_0000

Usage constraints

After 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
- MAINNSPPCEXP<N>
- PERIPHNSPPC0
- PERIPHNSPPC1
- PERIPHNSPPCEXP<N>
- MAINSPPPCEXP<N>
- PERIPHSPPPC0
- PERIPHSPPPC1
- PERIPHSPPPCEXP<N>
- NSMSCEXP
- NPUSPPORSL
- NPUSPPORPL

Bit descriptions

The following table shows the register bit assignments.

Table 4-6: SPCSECCTRL bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RAZ/WI	0x0000_0000
[0]	SPCSECCFGLOCK	Active High Control to Disable writes to Security related control registers in the Secure Access Configuration Register Block.	RW1S	0x00

4.5.1.2 BUSWAIT, Bus Access Wait register

The Bus Access Wait register allows software to gate access entering the interconnect from specific managers in the system, causing them to stall so that the processor can complete the configuration of the MPCs or other Security registers in the system prior to the stalled accesses commencing.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x004

Type

RO

Reset value

See individual bit resets

Bit descriptions

The following table shows the register bit assignments.

Table 4-7: BUSWAIT bit descriptions

Bits	Name	Description	Type	Reset
[31:17]	Reserved	Reserved	RAZ/WI	0x0000

Bits	Name	Description	Type	Reset
[16]	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.	RO	0x00
[15:1]	-	Reserved	RAZ/WI	0x0000
[0]	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 processor cores. This register also drives the output signal ACCWAITn. ACC_WAITN_STATUS and ACC_WAITN together, ensure that software can determine that all gates have reached the state that is requested.	RW	ACCWAITNRST

4.5.1.3 SECRESPCFG, Security Violation Response Configuration register

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Power domain

PD_SYS

Reset

nWARMRESETSYS

Address offset

0x010

Type

RW

Reset value

0x0000_0000

Usage constraints This register is Secure Privileged access only and supports 32-bit RW accesses. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

The following table shows the register bit assignments.

Table 4-8: SECRESPCFG bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RAZ/WI	0x0000_0000
[0]	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.	RW	0x00

4.5.1.4 NSCCFG, Non-secure Callable Configuration register

The Non-secure Callable Configuration register allows software to define if the region 0x1000_0000 to 0x1FFF_FFFF that normally host Secure code, and the region 0x3000_0000 to 0x3FFF_FFFF that normally implements Secure VMs, are Non-secure Callable regions of memory.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x014

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-9: NSCCFG bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Reserved.	RAZ/WI	0x0000_0000
[1]	RAMNSC	Configures if the region 0x3000_0000 to 0x3FFF_FFFF is Non-secure Callable: 0 - Not Non-secure Callable 1 - Non-secure Callable	RW	0x00
[0]	CODENSC	Configures if the CODE region 0x1000_0000 to 0x1FFF_FFFF is Non-secure Callable: 0 - Not Non-secure Callable 1 - Non-secure Callable	RW	0x00

4.5.1.5 SECMPCINTSTAT, Interrupt Status for Expansion Memory Protection Controller register

The interrupt signals from all Memory Protection Controllers (MPC), both within SSE-320 and in the expansion logic are merged and sent to CPU0 on a single interrupt signal. The Secure MPC Interrupt status register therefore provides Secure software with the ability to check which one of the MPC is causing the interrupt.

After the source of the interrupt is identified, you must use the MPC register interface to clear the interrupt.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x01C

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-10: SECMPCINTSTAT bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	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 option defines if each bit within this register is actually implemented such that if MPCEXPDIS[i] = `0b1` then SMPCEXP_STATUS[i] is disabled and always reads as zeros.	RO	`0x0000_0000`
[15:4]	Reserved	Reserved	**RAZ/WI**	`0x0000_0000`
[3]	SMPCVM3_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 3.	RO	`0x00`
[2]	SMPCVM2_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 2.	RO	`0x00`
[1]	SMPCVM1_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 1.	RO	`0x00`
[0]	SMPCVM0_STATUS	Interrupt Status for Memory Protection Controller of Volatile Memory Bank 0.	RO	`0x00`

4.5.1.6 SECPPCINTSTAT, SECPPCINTCLR and SECPPCINTEN

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

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- SECPPCINTSTAT - `0x20`
- SECPPCINTCLR - `0x24`
- SECPPCINTEN - `0x28`

Type

- SECPPCINTSTAT - RO
- SECPPCINTCLR - RW
- SECPPCINTEN - RW

Reset value

`0x0000_0000` (all)

Bit descriptions

The following tables show the register bit assignments.

Table 4-11: SECPPCINTSTAT bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved	**RAZ/WI**	0x000
[23:20]	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 state of the input signal SMAINPPCEXPSTATUS[n].	RO	0x00
[19:17]	Reserved	Reserved	**RAZ/WI**	0x00
[16]	SMAINPPC0_STATUS	Interrupt Status of Peripheral Protection Controller 0 on the Main Interconnect (MIPPC0). The boot ROM is protected by this PPC.	**RAZ/WI**	0x00
[15:8]	Reserved	Reserved	**RAZ/WI**	0x000
[7:4]	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 state of the input signal SPERIPHPPCEXPSTATUS[n].	RO	0x00
[3:2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	SPERIPHPPC1_STATUS	Interrupt status of Peripheral Protection Controller Group 1 on the Peripheral Interconnect within the System	RO	0x00
[0]	SPERIPHPPC0_STATUS	Interrupt status of Peripheral Protection Controller Group 0 on the Peripheral Interconnect within the System	RO	0x00

Table 4-12: SECPPCINTCLR bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved	**RAZ/WI**	0x000
[23:20]	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].	RAZW1C	0x00
[19:17]	Reserved	Reserved	**RAZ/WI**	0x00
[16]	SMAINPPC0_CLR	Interrupt clear of Peripheral Protection Controller 0 on the Main Interconnect (MIPPC0). Write '1' to clear SMAINPPC0_STATUS. The boot ROM is protected by this PPC.	**RAZ**/WIC	0x00
[15:8]	Reserved	Reserved	**RAZ/WI**	0x000
[7:4]	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].	RAZW1C	0x00
[3:2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	SPERIPHPPC1_CLR	Interrupt clear of Peripheral Protection Controller 1 on the Peripheral Interconnect within the System	RAZW1C	0x00
[0]	SPERIPHPPC0_CLR	Interrupt clear of Peripheral Protection Controller 0 on the Peripheral Interconnect within the System	RAZW1C	0x00

Table 4-13: SECPPCINTEN bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved	**RAZ/WI**	0x000

Bits	Name	Description	Type	Reset
[23:20]	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].	RW	0x00
[19:17]	Reserved	Reserved	**RAZ/WI**	0x00
[16]	SMAINPPC0_EN	Interrupt Enable of Peripheral Protection Controller 0 on the Main Interconnect (MIPPC0). Write '1' to enable interrupt from SMAINPPC0_STATUS. The boot ROM is protected by this PPC.	RW	0x00
[15:8]	Reserved	Reserved	**RAZ/WI**	0x000
[7:4]	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].	RW	0x00
[3:2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	SPERIPHPPC1_EN	Interrupt enable of Peripheral Protection Controller Group 1 on the Peripheral Interconnect within the System. Write 1 to enable interrupt from them.	RW	0x00
[0]	SPERIPHPPC0_EN	Interrupt enable of Peripheral Protection Controller Group 0 on the Peripheral Interconnect within the System. Write 1 to enable interrupt from them.	RW	0x00

4.5.1.7 SECMSCINTSTAT, SECMSCINTCLR and SECMSCINTEN

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

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset :

- SECMSCINTSTAT - 0x030
- SECMSCINTCLR - 0x034
- SECMSCINTEN - 0x038

Type :

- SECMSCINTSTAT - RO
- SECMSCINTCLR - RW

- SECMSCINTEN - RW

Reset value

0x0000_0000 (all)

Bit descriptions

The following tables show the register bit assignments.

Table 4-14: SECMSCINTSTAT bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	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 option MSCEXPDIS defines if each bit within this register is actually implemented such that if MSCEXPDIS[i] = 0b1 then SMSCEXP_STATUS[i] is disabled and always reads as zeros.	RO	0x0000
[15:12]	Reserved	Reserved	**RAZ/WI**	0x0
[11]	Reserved	Reserved	**RAZ/WI**	0x0
[10]	Reserved	Reserved	**RAZ/WI**	0x0
[9]	Reserved	Reserved	**RAZ/WI**	0x0
[8]	Reserved	Reserved	**RAZ/WI**	0x0
[7]	SMSCNPU0M3_STATUS	Interrupt status for NPU0 M3 MSC Note: **RAZ/WI** If NUMNPU < 1	RO	0x0
[6]	SMSCNPU0M2_STATUS	Interrupt status for NPU0 M2 MSC Note: **RAZ/WI** If NUMNPU < 1	RO	0x0
[5]	SMSCNPU0M1_STATUS	Interrupt status for NPU0 M1 MSC Note: **RAZ/WI** If NUMNPU < 1	RO	0x0
[4]	SMSCNPU0M0_STATUS	Interrupt status for NPU0 M0 MSC Note: **RAZ/WI** If NUMNPU < 1	RO	0x0
[3]	SMSCDMAM1_STATUS	Interrupt status for DMA M1 MSC **RAZ/WI** If NUMDMA < 1	R0	0x0
[2]	SMSCDMAM0_STATUS	Interrupt status for DMA M0 MSC **RAZ/WI** If NUMDMA < 1	R0	0x0
[1:0]	Reserved	Reserved	**RAZ/WI**	0x0

Table 4-15: SECMSCINTCLR bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	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 option MSCEXPDIS defines if each bit within this register is actually implemented, such that if MSCEXPDIS[*i*] = 0b1 then SMSCEXP_CLR[*i*] is disabled and any writes to it is ignored.	RAZW1C	0x0000

Bits	Name	Description	Type	Reset
[15:12]	Reserved	Reserved.	**RAZ/WI**	0x0
[11]	Reserved	Reserved.	**RAZ/WI**	0x0
[10]	Reserved	Reserved.	**RAZ/WI**	0x0
[9]	Reserved	Reserved.	**RAZ/WI**	0x0
[8]	Reserved	Reserved.	**RAZ/WI**	0x0
[7]	SMSCNPU0M3_CLR	Interrupt clear for NPU0 M3 MSC **RAZ/WI** If NUMNPU < 1	**RAZ**/WIC	0x0
[6]	SMSCNPU0M2_CLR	Interrupt clear for NPU0 M2 MSC **RAZ/WI** If NUMNPU < 1	**RAZ**/WIC	0x0
[5]	SMSCNPU0M1_CLR	Interrupt clear for NPU0 M1 MSC. SMSCDMAM1_CLR to delete this **RAZ/WI** If NUMNPU < 1	**RAZ**/WIC	0x0
[4]	SMSCNPU0M0_CLR	Interrupt clear for NPU0 M0 MSC **RAZ/WI** If NUMNPU < 1	**RAZ**/WIC	0x0
[3]	SMSCDMAM1_CLR	Interrupt clear for DMA M1 MSC **RAZ/WI** If NUMDMA < 1	**RAZ/WI**	0x0
[2]	SMSCDMAM0_CLR	Interrupt clear for DMA M0 MSC **RAZ/WI** If NUMDMA < 1	**RAZ/WI**	0x0
[1:0]	Reserved	Reserved.	**RAZ/WI**	0x0

Table 4-16: SECMSCINTEN bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	SMSCEXP_EN	Interrupt Enable for Expansion MSC. Each bit *n* enables or disables the input interrupt signal SMSCEXPSTATUS[n]. The parameter MSCEXPDIS defines if each bit within this register is actually implement such that if MSCEXPDIS[*i*] = `0b1` then SMSCEXP_EN[*i*] is disabled and any writes to it is ignored.	RW	0x0000
[15:12]	Reserved	Reserved.	**RAZ/WI**	0x0
[11]	Reserved	Reserved.	**RAZ/WI**	0x0
[10]	Reserved	Reserved.	**RAZ/WI**	0x0
[9]	Reserved	Reserved.	**RAZ/WI**	0x0
[8]	Reserved	Reserved.	**RAZ/WI**	0x0
[7]	SMSCNPU0M3_EN	Interrupt enable for NPU0 M3 MSC **RAZ/WI** If NUMNPU < 1	RW	0x0
[6]	SMSCNPU0M2_EN	Interrupt enable for NPU0 M2 MSC. **RAZ/WI** If NUMNPU < 1	RW	0x0
[5]	SMSCNPU0M1_EN	Interrupt enable for NPU0 M1 MSC. **RAZ/WI** If NUMNPU < 1	RW	0x0

Bits	Name	Description	Type	Reset
[4]	SMSCNPU0M0_EN	Interrupt enable for NPU0 M0 MSC. **RAZ/WI** If NUMNPU < 1	RW	0x0
[3]	SMSCDMAM1_EN	Interrupt enable for DMA M1 MSC **RAZ/WI** If NUMDMA < 1	**RAZ/WI**	0x0
[2]	SMSCDMAM0_EN	Interrupt enable for DMA M0 MSC **RAZ/WI** If NUMNPU < 1	**RAZ/WI**	0x0
[1:0]	Reserved	Reserved.	**RAZ/WI**	0x0

4.5.1.8 BRGINTSTAT, BRGINTCLR and BRGINTEN

SSE-320 and its expansion logic can contain bus bridges, which are necessary to handle clock domain crossing. To improve system performance, some of these bridges can buffer write data and complete a write access on their subordinate interfaces before any potential error response is received for the write access on their manager interfaces. When this occurs, these bridges can raise a combined interrupt.

The Bridge Buffer Error interrupt status, clear and enable register allow software to determine the source of the interrupt, clear the interrupt, and enable or disable (Mask) the interrupt.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- BRGINTSTAT - 0X040
- BRGINTCLR - 0X044
- BRGINTEN - 0X048

Type

- BRGINTSTAT - RO
- BRGINTCLR - RW
- BRGINTEN - RW

Reset value

0x0000_0000 (all)

Bit descriptions

The following tables show the register bit assignments.

Table 4-17: BRGINTSTAT bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	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 option BRGEXPDIS defines if each bit within this register is actually implemented such that if BRGEXPDIS[*i*] = `0b1` then BRGEXP_STATUS[*i*] is disabled and always reads as zeros.	RO	`0x0000`
[15:0]	Reserved	Reserved	**RAZ/WI**	`0x0000`

Table 4-18: BRGINTCLR Register

Bits	Name	Description	Type	Reset
[31:16]	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 option BRGEXPDIS defines if each bit within this register is actually implement such that if BRGEXPDIS[*i*] = `0b1` then BRGEXP_CLR[*i*] is disabled and any writes to it is ignored.	RAZW1C	`0x0000`
[15:0]	Reserved	Reserved	**RAZ/WI**	`0x0000`

Table 4-19: BRGINTEN Register

Bits	Name	Description	Type	Reset
[31:16]	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 option BRGEXPDIS defines if each bit within this register is actually implement such that if BRGEXPDIS[*i*] = `0b1` then BRGEXP_EN[*i*] is disabled and any writes to it is ignored.	RW	`0x0000`
[15:0]	Reserved	Reserved	**RAZ/WI**	`0x0000`

4.5.1.9 MAINNSPPCEXP<0 to 3>, Main Interconnect Non-secure Access Subordinate Peripheral Protection Controller Expansion register

The Main Interconnect Non-secure Access Subordinate Peripheral Protection Controller Expansion register 0, 1, 2, and 3 allow software to configure the security level of Main Interconnect peripherals. These peripherals reside in the subsystem expansion outside the subsystem.

These registers can be used to control the PPCs, outside the subsystem, which protect the Main Interconnect peripherals connected to the Main Interconnect through the Subordinate Main Expansion interfaces.

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

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

These controls directly control the expansion signals on the Security Control Expansion interface. All four registers are similar and each register x, where x is from 0 to 3, is defined as seen in the Bit descriptions table.

Configurations

The register implementation depends on the configuration of individual fields.

Attributes

Width

32-bit

Address offset

- MAINNSPPCEXP0 - `0x060`
- MAINNSPPCEXP1 - `0x064`
- MAINNSPPCEXP2 - `0x068`
- MAINNSPPCEXP3 - `0x06C`

Type

RW

Reset value

`0x0000_0000`

Bit descriptions

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.

The following table shows the register bit assignments.

Table 4-20: MAINNSPPCEXP<N> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved.	**RAZ/WI**	`0x0000`
[15:0]	MAINNSPPCEXP<x>	Expansion <x> Non-secure Access Main Interconnect Subordinate Peripheral Protection Control. Each bit i drives the output signal MAINNSPPCEXP<x>[i]. The configuration option MAINPPCEXP<x>DIS defines if each bit within this register is actually implemented such that if MAINPPCEXP<x>DIS[i] = `0b1` then MAINNSPPCEXP<x>[i] is disabled, **RAZ/WI**	RW	`0x0000`

4.5.1.10 PERIPHNSPPC0 and PERIPHNSPPC1, Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register

Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register allows software to configure Peripheral Interconnect peripherals. It controls whether each Peripheral Interconnect peripheral that it controls through a PPC has only Non-secure Privileged access or if is also allowed Non-secure Unprivileged access.

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offfset

- PERIPHNSPPC0 - `0x070`
- PERIPHNSPPC1 - `0x074`

Type

RW (all)

Reset value:

`0x0000_0000` (all)

Usage constraints

This register is Secure Privileged access only and supports 32-bit RW accesses. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

SSE-320 has two such groups of registers, for Peripheral Protection Controller Group 0 and for Peripheral Protection Controller Group 1, as follows:

The following tables show the register bit assignments.

Table 4-21: PERIPHNSPPC0 bit descriptions

Bits	Name	Description	Type	Reset
[31:9]	Reserved	Reserved.	**RAZ/WI**	`0x0000_00`
[8]	NS_SDC	Access Security for SDC-600. If SDC-600 does not exist, this field is reserved and **RAZ/WI**.	RW	`0x00`
[7]	NS_SYSDSS	Access Security for interconnect access to Debug System. When HASCSS = 0, this field is reserved and **RAZ/WI**.	**RAZ/WI**	`0x00`
[6]	Reserved	Reserved.	**RAZ/WI**	`0x00`
[5]	NS_TIMER3	Access Security for TIMER3	RW	`0x00`
[4]	Reserved	Reserved.	**RAZ/WI**	`0x00`
[3]	Reserved	Reserved.	**RAZ/WI**	`0x00`
[2]	NS_TIMER2	Access Security for TIMER2	RW	`0x00`

Bits	Name	Description	Type	Reset
[1]	NS_TIMER1	Access Security for TIMER1	RW	0x00
[0]	NS_TIMER0	Access Security for TIMER0	RW	0x00


Note

Access to several peripherals filtered by PIPPC0 through its security setting is fixed and is not represented in Table 4-21: PERIPHNSPPC0 bit descriptions on page 131. Because of this, if such peripheral is accessed using the wrong security it also results in interrupts being raised for PIPPC0. A full list of peripherals protected by PIPPC0 are listed in Peripheral Protection Controllers.

Table 4-22: PERIPHNSPPC1 bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved.	RAZ/WI	0x0000_0000
[0]	NS_SLOWCLK_TIMER	Access Non-Security for SLOWCLK_TIMER	RW	0x00


Note

Access to several peripherals filtered by PIPPC1 through its security setting is fixed and is not represented in Table 4-22: PERIPHNSPPC1 bit descriptions on page 132. Because of this, if such peripheral is accessed using the wrong security it also results in interrupts being raised for PIPPC1. A full list of peripherals protected by PIPPC1 are listed in Peripheral Protection Controllers.

4.5.1.11 NPUSPPORSL, NPU Secure access security level reset control register

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

The Value of NPUSPPORSL is only sampled by the NPU, when the NPU is released from reset.

The default reset value of this register is controlled by NPU0PORSLRST.

Configurations

The default reset value of this register is controlled by NPU0PORSLRST.

Attributes

Width

32-bit

Reset value

The Default reset value of this register is controlled by NPU0PORSLRST.

Where:

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

Bit descriptions

The following table shows the register bit assignments.

Table 4-23: NPUSPPORSL bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	**RAZ/WI**	0x0000_00
[0]	SP_NPU0PORSL	Configures the security level strap value for NPU0. This field does not exist, is reserved and **RAZ/WI** when NUMNPU < 1	RW	NPU0PORSLRST

4.5.1.12 PERIPHNSPPCEXP<0 to 3>, Peripheral Interconnect Non-secure Access Subordinate Peripheral Protection Controller Expansion register

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

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

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

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

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Bit descriptions

The following table shows the register bit assignments.

Table 4-24: PERIPHNSPPCEXP<x> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	**RAZ/WI**	0x0000

Bits	Name	Description	Type	Reset
[15:0]	PERIPHNSPPCEXP<x>	Expansion <x> Non-secure Access Peripheral Interconnect Subordinate Peripheral Protection Control. Each bit *n* drives the output signal PERIPHNSPPCEXP<x>[n]. The configuration option PERIPHPPCEXP<x>DIS defines if each bit within this register is actually implemented such that if PERIPHPPCEXP<x>DIS[*i*] = `0b1` then PERIPHPPCEXP<x>DIS[*i*] reads as zeros and any writes to it is ignored.	RW	`0x0000`

4.5.1.13 MAINSPPPCEXP<0 to 3>, Expansion Secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register

The Expansion Secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure Secure privilege level of each Main Interconnect peripheral that resides in the expansion logic outside the subsystem.

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

Each field defines this for an associated peripheral, by the following settings:

1

Allows Secure Unprivileged and Privileged access

0

Allows Secure Privileged access only

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

- MAINSPPPCEXP0 - `0x0A0`
- MAINSPPPCEXP0 - `0x0A4`
- MAINSPPPCEXP0 - `0x0A8`
- MAINSPPPCEXP0 - `0x0AC`

Type

RW

Reset value

`0x0000_0000`

Bit descriptions

These registers directly control the expansion signals on the Security Control Expansion interface. Registers MAINSPPPCEXP <N> (where N is from 0-3) are similar. The following table shows the register bit assignments.

Table 4-25: MAINSPPPCEXP<N> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	**RAZ/WI**	0x0000
[15:0]	MAINSPPPCEXP<N>	Expansion <*N*> Secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Control. Each bit n drives the output signal MAINPPPCEXP<N>[n] if MAINNSPPCEXP<*N*>[*n*] is also LOW, where *N* is 0 to 3. The configuration option MAINPPCEXP<*N*>DIS defines if each bit within this register is actually implemented, such that if MAINPPCEXP<*N*>DIS[*i*] = 0b1 then MAINSPPPCEXP<*N*>[*i*] is disabled, reads as zeros and any writes to it is ignored.	RW	0x0000

4.5.1.14 PERIPHSPPPC0 and PERIPHSPPPC1

The Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register allows software to configure Peripheral Interconnect peripherals. It configures whether 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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- PERIPHSPPPC0 - 0x0B0
- PERIPHSPPPC1 - 0x0B4

Type

RW

Reset value

0x0000_00

Bit descriptions

SSE-320 has two registers PERIPHSPPPC0 for Peripheral Protection Controller Group 0 and PERIPHSPPPC1 for Peripheral Protection Controller Group 1. The following table shows the register bit assignments

Table 4-26: PERIPHSPPPC0 bit descriptions

Bits	Name	Description	Type	Reset
[31:9]	Reserved	Reserved	**RAZ/WI**	0x0000_00
[8]	SP_SDC	Secure Privileged setting SDC-600. If SDC-600 does not exist, this field is reserved and **RAZ/WI**.	RW	0x0
[7]	SP_SYSDSS	Secure privileged setting for interconnect access to Debug System. When HASCSS =0, this field is reserved and **RAZ/WI**.	RW	0x0
[6]	SP_WATCHDOG_REF	Secure Unprivileged setting for Secure Watchdog Refresh Frame	RW	0x00
[5]	SP_TIMER3	Secure Unprivileged setting for TIMER3	RW	0x00
[4]	-	Reserved	**RAZ/WI**	0x00
[3]	Reserved	Reserved	**RAZ/WI**	0x00
[2]	SP_TIMER2	Secure Unprivileged setting for TIMER2	RW	0x00
[1]	SP_TIMER1	Secure Unprivileged setting for TIMER1	RW	0x00
[0]	SP_TIMER0	Secure Unprivileged setting for TIMER0	RW	0x00

Table 4-27: PERIPHSPPPC1 Register

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	**RAZ/WI**	0x0
[3]	SP_LCM	Secure privileged setting for Lifecycle Manager subordinate interface	RW	0x0
[2]	SP_SAM	Secure privileged setting for Security Alarm Manager subordinate interface	RW	0x0
[1]	SP_KMU	Secure privileged setting for Key Management Unit subordinate APB3 programming port and for AHB5 Manager interface	RW	0x0
[0]	SP_SLOWCLK_TIMER	Secure Unprivileged setting for SLOWCLK_TIMER	RW	0x00

4.5.1.15 NPUSPPORPL, NPU Secure access privilege level reset control register

The NPU Secure access privilege level reset control registers allows software to configure if NPU0 resets to Privileged or Unprivileged state. The value of this register is only sampled by the NPU, when the NPU is released from reset and NPUSPPORSL.SP_NPU0PORSL is in Secure State.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x0B8

Type

RW

Reset value

The default reset value of this register is controlled by NPU0PORPLRST.

Where:

- 1: Reset to Privileged state
- 0: Reset to Unprivileged state

Bit descriptions

The following table shows the register bit assignments.

Table 4-28: NPUSPPORPL bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	**RAZ/WI**	0x0000_00
[0]	SP_NPU0PORPL	Configures the Secure access privilege level for NPU0. When NUMNPU < 1 , this field does not exist, is reserved and **RAZ/WI**.	RW	NPU0PORPLRST

4.5.1.16 PERIPHSPPPCEXP<0 to 3>, Expansion Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register

The Expansion Secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2, and 3 allow software to configure the Secure privilege level of Peripheral Interconnect peripherals. These registers can be used to control the PPCs that are outside the subsystem, which protect the Peripheral Interconnect peripherals connected to the Peripheral Interconnect through the Subordinate Peripheral Expansion interfaces.

Each field defines this access for an associated peripheral, by the following settings:

- 1 - Allows Secure Unprivileged and Privileged access
- 0 - Allows Secure Privileged access only

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

- PERIPHSPPPCEXP0 - `0x0C0`
- PERIPHSPPPCEXP1 - `0x0C4`
- PERIPHSPPPCEXP2 - `0x0C8`
- PERIPHSPPPCEXP3 - `0x0CC`

Type

RW

Reset value

`0x0000`

Bit descriptions

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

The following table shows the register bit assignments.

Table 4-29: PERIPHSPPPCEXP<N> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved.	**RAZ/WI**	`0x0000`
[15:0]	PERIPHSPPPCEXP<N>	Expansion <*N*> Secure Unprivileged Access Peripheral Interconnect Subordinate 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 option PERIPHPPCEXP<*N*>DIS defines if each bit within this register is actually implemented such that if PERIPHPPCEXP<*N*>DIS[*i*] = `0b1` then PERIPHSPPPCEXP<*N*>[*i*] is disabled, reads as zeros and any writes to it is ignored.	RW	`0x0000`

4.5.1.17 NSMSCEXP, Non-secure Expansion Manager Security Controller register

The Non-secure Expansion Manager Security Controller register allows software to configure if each manager that is located behind each MSC in the subsystem expansion is a Secure or Non-secure device.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-30: NSMSCEXP bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	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 Manager as Non-secure, or LOW for Secure. The parameter MSCEXPDIS defines if each bit within this register is actually implemented such that if MSCEXPDIS[*i*] = `0b1` then NS_MSCEXP[*i*] is disabled, it reads as `0b1` and any writes to it is ignored. Resets to NSMSCEXPRST.	RW	NSMSCEXPRST
[15:0]	Reserved	Reserved	**RAZ/WI**	`0x0000`

4.5.1.18 PIDR4, Peripheral ID 4 register

The Peripheral ID4 register returns byte[4] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

`0xFD0`

Type

RO

Reset value

`0x0000_0004`

Bit descriptions

The following table shows the register bit assignments.

Table 4-31: Peripheral ID 4 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	`0x000000`

Bits	Name	Description	Type	Reset
[7:4]	SIZE	4KB Count, the number of 4K pages used. • 0x00: 4K • 0x01: 8K • 0x02: 16K • 0x03: 32K	RO	0x0
[3:0]	DES_2	JEP106 Continuation Code	RO	0x4

4.5.1.19 PIDR0, Peripheral ID 0 register

The Peripheral ID0 register returns byte[0] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFE0

Type

RO

Reset value

0x0000_00C0

Bit descriptions

The following table shows the register bit assignments.

Table 4-32: Peripheral ID 0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RAZ/WI	0x00
[7:0]	Peripheral ID 0	PART_0, Identification register part number, bits[7:0]	RO	0xC0

4.5.1.20 PIDR1, Peripheral ID 1 register

The Peripheral ID1 register returns byte[1] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFE4

Type

RO

Reset value

0x0000_00B3

Bit descriptions

The following table shows the register bit assignments.

Table 4-33: Peripheral ID 1 register bit description

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RAZ/WI	0x000000
[7:4]	DES_0	JEP106 identification code, bits[3:0]	RO	0xB
[3:0]	PART_1	Identification register part number, bits[11:8]	RO	0x3

4.5.1.21 PIDR2, Peripheral ID 2 register

The Peripheral ID2 register returns byte[2] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFE8

Type

RO

Reset value

0x0000_000B

Bit descriptions

The following table shows the register bit assignments.

Table 4-34: Peripheral ID 2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x000000
[7:4]	REVISION	Revision Code	RO	0x0
[3]	JEDEC	JEDEC	RO	0x1
[2:0]	DES_1	JEP106 identification code,bits[6:4]	RO	0x3

4.5.1.22 PIDR3, Peripheral ID 3 register

The Peripheral ID3 register returns byte[3] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFEC

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-35: Peripheral ID 3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x000000
[7:4]	REVAND	Manufacturer revision number	R0	0x0
[3:0]	CMOD	Customer Modified	R0	0x0

4.5.1.23 CIDR0, Component ID 0 register

The Component ID0 register returns byte[0] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFF0

Type

RO

Reset value

0x0000_000D

Bit descriptions

The following table shows the register bit assignments.

Table 4-36: Component ID 0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x000000
[7:0]	PRMBL_0	Preamble	RO	0x0D

4.5.1.24 CIDR1, Component ID 1 register

The Component ID1 register returns byte[1] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFF4

Type

RO

Reset value

0x0000_00F0

Bit descriptions

The following table shows the register bit assignments.

Table 4-37: Component ID 1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x000000

Bits	Name	Description	Type	Reset
[7:4]	Class	Component class	RO	0xF
[3:0]	PRMBL_1	Preamble	RO	0x0

4.5.1.25 CIDR2, Component ID 2 register

The Component ID2 register returns byte[2] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFF8

Type

RO

Reset value

0x0000_0005

Bit descriptions

The following table shows the register bit assignments.

Table 4-38: Component ID 2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	RAZ/WI	0x000000
[7:0]	PRMBL_2	Preamble	RO	0x05

4.5.1.26 CIDR3, Component ID 3 register

The Component ID3 register returns byte[3] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0xFFC

Type

RO

Reset value

0x0000_00B1

Bit descriptions

The following table shows the register bit assignments.

Table 4-39: Component ID 3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x000000
[7:0]	PRMBL_3	Preamble	RO	0xB1

4.5.1.27 SAM, KMU, and LCM component registers

SSE-320 integrates security components: Key Management Unit (KMU), Security Alarm Manager (SAM), and Lifcycle Manager (LCM).

Registers for these components reside in the Peripheral Region.

For details of their registers, see the relevant specification document:

- *Arm® Security Alarm Manager Specification*
- *Arm® Key Management Unit Specification*
- *Arm® Lifecycle Manager Specification*

4.5.1.28 NPU0 registers

SSE-320 implements one Ethos-U85 NPU.

The NPU registers reside in the Peripheral Region.

4.5.1.29 DMA registers

SSE-320 implements one DMA-350.

The DMA registers reside in the Peripheral Region.

The DMA handles security and privilege checking of accesses to DMA registers. The DMA resides in the PD_SYS power domain and is reset by nWARMRESETSYS.

For details of the DMA registers, see *Arm® CoreLink™ DMA-350 Controller Technical Reference Manual*.

4.5.2 Non-secure Access Configuration register block

The Non-secure Access Configuration Register Block implements program visible states that allows software to control various security gating units within the design.

This register block base address is `0x4008_0000`. This register block is Non-secure Privileged access only and supports 32-bit R/W accesses. For write access to these registers, only 32-bit writes are supported. Any Byte and Half word writes are ignored.

All registers reside in the PD_SYS power domain and are reset by nWARMRESETSYS.

The following table lists the registers within this unit. Details of each register are described in separate sections. The width of all registers is 32-bit.

Table 4-40: Non-secure Access Configuration Register Block Register Map

Offset	Name	Type	Reset value	Description
`0x000` - `0x08C`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0x090`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0x094` - `0x09C`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0x0A0`	MAINNSPPPCEXP0	RW	`0x0000_0000`	Expansion 0 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
`0x0A4`	MAINNSPPPCEXP1	RW	`0x0000_0000`	Expansion 1 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
`0x0A8`	MAINNSPPPCEXP2	RW	`0x0000_0000`	Expansion 2 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
`0x0AC`	MAINNSPPPCEXP3	RW	`0x0000_0000`	Expansion 3 Non-secure Unprivileged Access Peripheral Protection Control on Main Interconnect.
`0x0B0`	PERIPHNSPPPC0	RW	`0x0000_0000`	Non-secure Unprivileged Access Peripheral Protection Control 0 on Peripheral Interconnect.
`0x0B4`	PERIPHNSPPPC1	RW	`0x0000_0000`	Non-secure Unprivileged Access Peripheral Protection Control 1 on Peripheral Interconnect.
`0x0BC`	NPUNSPORPL	RW	Configurable	Non-secure Access NPU Privilege level reset state control
`0x0B8`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0x0C0`	PERIPHNSPPPCEXP0	RW	`0x0000_0000`	Expansion 0 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
`0x0C4`	PERIPHNSPPPCEXP1	RW	`0x0000_0000`	Expansion 1 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
`0x0C8`	PERIPHNSPPPCEXP2	RW	`0x0000_0000`	Expansion 2 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
`0x0CC`	PERIPHNSPPPCEXP3	RW	`0x0000_0000`	Expansion 3 Non-secure Unprivileged Access Peripheral Protection Control on Peripheral Interconnect.
`0x0D0` - `0xFCC`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved

Offset	Name	Type	Reset value	Description
0xFD0	PIDR4	RO	0x0000_0004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_0053	Peripheral ID 0
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_003B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.5.2.1 MAINNSPPPCEXP<0 to 3>, Expansion Non-secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register

The Expansion Non-secure Unprivileged Access Main Interconnect Subordinate Peripheral Protection Controller register 0, 1, 2 and 3 allows software to configure each Main Interconnect peripheral.

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

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

1

Allows Non-secure Unprivileged and Privileged access

0

Allows Non-secure Privileged access only

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

- MAINNSPPPCEXP0 - 0x0A0
- MAINNSPPPCEXP1 - 0x0A4
- MAINNSPPPCEXP2 - 0x0A8
- MAINNSPPPCEXP3 - 0x0AC

Type

RW

Reset value

0x0000_0000

Bit descriptions

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

Table 4-41: MAINNSPPPCEXP<N> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	RAZ/WI	0x0000
[15:0]	MAINNSPPPCEXP<N>	Expansion <N> Non-secure Privilege Access Main Interconnect Subordinate 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 option MAINPPCEXP<N>DIS defines if each bit within this register is actually implemented, such that if MAINPPCEXP<N>DIS[i] = 0b1 then MAINNSPPPCEXP<N>[i] is disabled, reads as zeros and any writes to it is ignored.	RW	0x0000

4.5.2.2 PERIPHNSPPPC0 and PERIPHNSPPPC1, Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register

Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller register allows software to configure if each Peripheral Interconnect peripheral that it controls through a PPC is only Non-secure Privileged access only or is allowed Non-secure Unprivileged access as well.

Each field defines this for an associated peripheral, by the following settings:

1

Allows Non-secure Unprivileged and Privileged access

0

Allows Non-secure Privileged access only

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- PERIPHNSPPPC0 - 0x0B0

- PERIPHNSPPPC1 - `0x0B4`

Type

RW

Reset value

`0x0000_0000`

Bit descriptions

SSE-320 has two registers PERIPHNSPPPC0 for Peripheral Protection Controller Group 0 and PERIPHNSPPP1 for Peripheral Protection Controller Group 1. The following table shows the register bit assignments

Table 4-42: PERIPHNSPPPC0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved	**RAZ/WI**	0x0000_000
[7]	NSP_SYSDSS	Non-secure privileged setting for Debug System. When HASCSS = 0, this field is reserved and **RAZ/WI**.	**RAZ/WI**	0x00
[6]	NSP_WATCHDOG_REF	Secure System Watchdog Refresh Frame	RW	0x00
[5]	NSP_TIMER3	Non-secure Unprivileged setting for TIMER3	RW	0x00
[4]	Reserved	Reserved	**RAZ/WI**	0x00
[3]	Reserved	Reserved	**RAZ/WI**	0x00
[2]	NSP_TIMER2	Non-secure Unprivileged setting for TIMER2	RW	0x00
[1]	NSP_TIMER1	Non-secure Unprivileged setting for TIMER1	RW	0x00
[0]	NSP_TIMER0	Non-secure Unprivileged setting for TIMER0	RW	0x00

Table 4-43: PERIPHNSPPPC1 bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	**RAZ/WI**	0x0000_0000
[0]	NSP_SLOWCLK_TIMER	Non-secure Unprivileged setting for SLOWCLK_TIMER	RW	0x00


Note

Access to several peripherals filtered by PIPPC1 though its security setting is fixed and is not represented in Table 4-43: PERIPHNSPPPC1 bit descriptions on page 149. Because of this, if such peripheral is accessed using the wrong security it also results in interrupts being raised for PIPPC1. A full list of peripherals protected by PIPPC1 are listed in Peripheral Protection Controllers.

4.5.2.3 NPUNSPORPL, NPU Non-secure access privilege level reset control registers

The NPU Non-secure access privilege level reset control registers allows software to configure if each NPU resets to Privileged or Unprivileged state.

The value of this register is only sampled by the NPU, when the NPU is released from reset and NPUSPPORSL.SP_NPU0PORSL is in Non-secure State.

The default reset value of this register is controlled by NPU0PORPLRST.

Where:

1: Reset to Privileged state

0: Reset to Unprivileged state

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

`0x0BC`

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-44: NPUNSPORPL bit descriptions

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	**RAZ/WI**	`0x0000_00`
[0]	NS_NPU0PORPL	Configures the Non-secure access privilege level for NPU0. When NUMNPU < 1, this field does not exist, is reserved and **RAZ/WI**.	RW	NPU0PORPLRST

4.5.2.4 PERIPHNSPPPCEXP<0 to 3>

The Expansion Non-secure Unprivileged Access Peripheral Interconnect Subordinate Peripheral Protection Controller registers 0, 1, 2, and 3 allow software to configure each Peripheral

Interconnect peripheral that it controls. These registers configure whether the peripheral is allowed only Non-secure privileged access or if is also allowed Non-secure unprivileged access.

Each field defines this for an associated peripheral, by the following settings:

1

Allows Non-secure Unprivileged and Privileged access

0

Allows Non-secure Privileged access only

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Reset value

`0x0000_0000`

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

Bit descriptions

The following table shows the register bit assignments.

Table 4-45: PERIPHNSPPPCEXP<N> bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved.	**RAZ/WI**	`0x0000`
[15:0]	PERIPHNSPPPCEXP<N>	Expansion <*N*> Non-secure Unprivileged Access Peripheral Interconnect Subordinate 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 option PERIPHPPCEXP<*N*>DIS defines if each bit within this register is actually implement such that if PERIPHPPCEXP<*N*>DIS[*i*] = `0b1`, then PERIPHNSPPPCEXP<*N*>[*i*] is disabled, reads as zeros and any writes to it is ignored.	RW	`0x0000`

4.5.3 Timestamp timers

SSE-320 implements four timestamp-based timers in the system, TIMER<n> where *N* is 0 to 3. All timers are mapped to the Secure or Non-secure world through PPC0, which also controls the accessibility of Unprivileged accesses.

For more details, see Secure Access Configuration register block.

All timestamp timers and watchdog, except for Timer3, reside in PD_SYS power domain and are reset by nWARMRESETSYS, while the Timer 3 resides in the PD_AON power domain and is reset by nWARMRESETAON.

For more information of the ARMv8-M System Counter Timer registers, see *Arm® Corstone™ Reference Systems Architecture Specification Ma2*.

4.5.4 Timestamp watchdogs

SSE-320 implements two timestamp-based watchdogs in the system. Both reside in PD_SYS power domain and are reset by nWARMRESETSYS. One watchdog timer is Secure access only, while another is Non-secure.

Accessibility of each watchdog to Unprivileged access is also controlled by through PPC0. For more details,see PERIPHSPPPC0 and PERIPHSPPPC1 and PERIPHNSPPPC0 and PERIPHNSPPPC1.

Each Watchdog timer implements two register frames, a Control Frame and a Refresh Frame.

For more information of the ARMv8-M Timestamp Watchdog registers, see *Arm® Corstone™ Reference Systems Architecture Specification Ma2*.

4.6 Processor Private region

The CPU0 has its own copy of the Processor Private region, which is only assessable to itself.

The Processor Private region consists of four regions as follows:

- `0x4001_0000` to `0x4001_FFFF` implements a Non-secure Low Access Latency region
- `0x4801_0000` to `0x4801_FFFF` implements a Non-secure High Access Latency region
- `0x5001_0000` to `0x5001_FFFF` implements a Secure Low Access Latency region
- `0x5801_0000` to `0x5801_FFFF` implements a Secure High Access Latency region

Of these four regions, only `0x40010000` to `0x4001_FFFF` and `0x5001_0000` to `0x5001_FFFF` implement any registers.

None of these regions are accessible from any other manager in the system, except when using the external debugger through CPU0.

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

Table 4-46: Processor Private Region address map

Row ID	Address	Size	Region name	Description	Alias with row ID	Security[1]
0	0x4001_0000 - 0x4001_1FFF	-	Reserved	Reserved	-	-
1	0x4001_2000 - 0x4001_2FFF	4KB	CPU0_PWRCTRL	CPU0 Power Control Block. See CPU0_PWRCTRL register block	7	NS, P
2	0x4001_3000 - 0x4001_EFFF	-	Reserved	Reserved	-	-
3	0x4001_F000 - 0x4001_FFFF	4KB	CPU0_IDENTITY	CPU0 Identity Block, See CPU0_IDENTITY register block	9	NS, UP
4	0x4801_0000 - 0x4801_FFFF	-	Reserved	Reserved	-	-
5	0x5001_0000 - 0x5001_0FFF	-	Reserved	Reserved	-	-
6	0x5001_1000 - 0x5001_1FFF	4KB	CPU0_SECCRTL	CPU0 Local Security Control Block, See CPU0_SECCTRL register block	-	S, P
7	0x5001_2000 - 0x5001_2FFF	4KB	CPU0_PWRCTRL	CPU0 Power Control Block. See CPU0_PWRCTRL register block	1	S, P
8	0x5001_3000 - 0x5001_EFFF	-	Reserved	Reserved	-	-
9	0x5001_F000 - 0x5001_FFFF	4KB	CPU0_IDENTITY	CPU0 Identity Block, See CPU0_IDENTITY register block	3	S, UP
10	0x5801_0000 - 0x5801_FFFF	-	Reserved	Reserved	-	-

[1] Legend

- S: Secure access only.
- NS: Non-secure access only.
- P: Privilege access only.
- UP: Unprivileged and Privilege access allowed.

4.6.1 CPU0_PWRCTRL register block

SSE-320 implements a CPU0_PWRCTRL register block for CPU0 in the subsystem. This block resides at address 0x40012000 in a Non-secure region and is also alias to 0x50012000 in the Secure region. The CPU0_PWRCTRL registers are read only registers when accessed from the Non-secure region starting at address 0x40012000 and any writes access to it in that region is ignored.

The following table lists the registers in the CPU0_PWRCTRL register block. The width of all registers is 32-bit.

Table 4-47: CPU0_PWRCTRL register map

Offset	Name	Type	Reset value	Description
0x000	CPUPWRCFG	RW	0x0000_0000	CPU0 Local Power Configuration register
0x004 - 0xFCC	Reserved	RO	0x0000_0000	Reserved
0xFD0	PIDR4	RO	0x0000_0004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	RO	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_005A	Peripheral ID 0
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_000B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.6.1.1 CPUPWRCFG, CPU software control registers

The CPUPWRCFG register provides the local CPU software control registers for power control.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Type

This register is Read Only if accessed from the Non-secure world.

Power domain

This register resides in the same reset domain as the CPU0 core, nWARMRESETCPU0 reset domain, and PD_CPU0 power domain. When CPU0 is powered down, the register is also powered down, and are cleared when powered back up.

Reset

This register resides in the same reset domain, nWARMRESETCPU0, as its associated CPU core so that when the CPU is powered down, the register is also powered down and is cleared when powered back up.

Bit descriptions

The following table shows the register bit assignments.

Table 4-48: CPUPWRCFG bit descriptions

Bits	Name	Description	Type	Reset
[31:5]	Reserved	Reserved.	**RAZ/WI**	0x0000_000
[4]	TCM_MIN_PWR_STATE	Defines the minimum powerstate of the TCM for CPU0. • '0' - OFF • '1' - Retention. This bit is read access only from the Non-secure world. When PD_CPU0 returns from MEM_RET or MEM_RET_NOCACHE state to one of the ON states, this bit is set to 0b1.	RW	0x00
[3:1]	Reserved	Reserved	**RAZ/WI**	0x00
[0]	USEIWIC	When HIGH selects the use of IWIC for CPU <n> when in DeepSleep. Else selects the use of EWIC. If HASCPU0IWIC for this CPU <n> is 0, this field is reserved and **RAZ/WI**. This bit is read access only from the Non-secure world.	**RAZ/WI**	0x00

4.6.2 CPU0_IDENTITY register block

SSE-320 implements a CPU0_IDENTITY register block for CPU0. This block resides at address 0x4001_F000 in a Non-secure region and is also alias to 0x5001_F000 in the Secure region. This is a read only register and any write access is ignored.

The following table lists the registers in the CPU0_IDENTITY block. The width of all registers is 32-bit.

Table 4-49: CPU0_IDENTITY register map

Offset	Name	Type	Reset value	Description
0x000	CPUID	RO	CPU0CPUIDRST	Unique CPU Identity Number
0x004 - 0xFCC	Reserved	RO	0x0000_0000	Reserved
0xFD0	PIDR4	RO	0x0000_0004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	RO	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_0055	Peripheral ID 0
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_000B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.6.2.1 CPUID, CPU ID register

The CPUID register is a read only register that when read by CPU0, provides an identity code to the CPU that is unique to that CPU.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

`0x000`

Type

RO

Reset value

CPU0CPUIDRST

Bit descriptions

The following table shows the register bit assignments.

Table 4-50: CPUID bit descriptions

Bits	Name	Description	Type	Rest
[31:4]	Reserved	Reserved.	**RAZ/WI**	`0x0000_000`
[3:0]	CPUID	CPU Identity. Defined by configuration CPU0CPUIDRST.	RO	CPU0CPUIDRST

4.6.3 CPU0_SECCTRL register block

CPU0 has a CPU0_SECCTRL register block that allows the security locks of the processor to be configured. The register block resides in nWARMRESETCPU0 reset domain and PD_CPU0 power domain so that when a core is powered down, they are also powered down and are cleared when powered back up. These registers are Secure access only and resides at address `0x5001_1000`.

The following table lists the registers in the CPU0_SECCTRL Register block. The width of all registers is 32-bit.

Table 4-51: CPU0_SECCTRL register map

Offset	Name	Type	Reset value	Description
`0x000`	CPUSECCFG	RW	`0x0000_0000`	CPU Local Security Configuration
`0x004 - 0xFCC`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0xFD0`	PIDR4	RO	`0x0000_0004`	Peripheral ID 4
`0xFD4 - 0xFDC`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0xFE0`	PIDR0	RO	`0x0000_0059`	Peripheral ID 0

Offset	Name	Type	Reset value	Description
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_001B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.6.3.1 CPUSECCFG, CPU Local Security Configuration Register

The CPU Local Security Configuration Register allows software to set security lock bits at the CPU interface.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x000

Type

RW

Reset value

0x0000_00000

Bit descriptions

The following table shows the register bit assignments.

Table 4-52: CPUSECCFG bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	Reserved	Reserved.	RAZ/WI	0x0000_0000
[5]	LOCKDTGU	When HIGH, disables writes to the CPU0 DTGU_CTRL and DTGU_LUTn registers from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00
[4]	LOCKITGU	When HIGH, disables writes to the CPU0 ITGU_CTRL and ITGU_LUTn from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00
[3]	LOCKTCM	When HIGH, disables writes to the CPU0 ITCMCR, DTCMCR from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00

Bits	Name	Description	Type	Reset
[2]	LOCKSMPU	When HIGH, disables write to the CPU0 MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An, MPU_RLAR_An registers associated with the Secure MPU from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00
[1]	LOCKSAU	When HIGH, disables writes to the CPU0 SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00
[0]	LOCKSVTAIRCR	When HIGH, disables writes to the CPU0 VTOR_S, AIRCR.PRIS, and AIRCR.BFHFNMINS registers. Once set to HIGH, it cannot be cleared until Reset.	RW1S	0x00

4.7 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:

- 0x4002_0000 to 0x4003_FFFF, 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, 0x5002_0000 to 0x5003_FFFF.
- 0x4802_0000 to 0x4803_FFFF, which is a Non-secure region for high latency system control peripherals. Some peripherals are expected to be aliased in its associated Secure region, 0x5802_0000 to 0x5803_FFFF.
- 0x5002_0000 to 0x5003_FFFF, 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, 0x4002_0000 to 0x4003_FFFF.
- 0x5802_0000 to 0x5803_FFFF, which is a Secure region for high latency system control peripherals. Some peripherals are expected to be aliased in its associated Non-secure region, 0x4802_0000 to 0x4803_FFFF.

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 4-53: System Control Peripheral Region address map

Row ID	Address	Size	Region Name	Description	Alias with row ID	Security[1]
1	0x4000_2000 - 0x4003_FFFF	128KB	Reserved	Reserved	-	-
2	0x4802_0000 - 0x4802_0FFF	4KB	SYSINFO	System Information register block. See SYSINFO register block.	-	NS, UP
3	0x4802_1000 - 0x4802_EFFF	56KB	Reserved	Reserved. When accessed, results in **RAZ/WI**.	-	NS, UP

Row ID	Address	Size	Region Name	Description	Alias with row ID	Security[1]
4	0x4802_F000 - 0x4802_FFFF	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See SLOWCLK AON Timers.	31	NS-PPC, P-PPC
5	0x4803_0000 - 0x4803_FFFF	64KB	Reserved	Reserved. When accessed, results in bus error.	-	-
6	0x5002_0000 - 0x5009_FFFF	128KB	Reserved	Reserved. When accessed, results in bus error.	-	-
7	0x5802_0000 - 0x5802_0FFF	4KB	SYSINFO	System Information register block. See SYSINFO register block.	-	S, UP
8	0x5802_1000 - 0x5802_1FFF	4KB	SYSCONTROL	System Control register block. See SOC_IDENTITY	-	S, P
18	0x5802_2000 - 0x5802_2FFF	4KB	SYSPPU	PPU for BR_SYS. See Power Policy Units.	-	S, P
19	0x5802_3000 - 0x5802_3FFF	4KB	CPU0PPU	PPU for BR_CPU0. See Power Policy Units.	-	S, P
20	0x5802_4000 - 0x5802_4FFF	4KB	Reserved	Reserved	-	-
21	0x5802_5000 - 0x5802_5FFF	4KB	Reserved	Reserved	-	-
22	0x5802_6000 - 0x5802_6_FFF	4KB	Reserved	Reserved	-	-
23	0x5802_7000 - 0x5802_7FFF	4KB	Reserved	Reserved	-	-
24	0x5802_8000 - 0x5802_8FFF	4KB	MGMTPPU	PPU for BR_MGMT. See Power Policy Units.	-	S, P
25	0x5802_9000 - 0x5802_9FFF	4KB	DEBUGPPU	PPU for BR_DEBUG. See Power Policy Units.	-	S, P
26	0x5802_A000 - 0x5802_AFFF	4KB	NPU0PPU	PPU for BR_NPU0. See Power Policy Units.	-	S, P
27-29	0x5802_B000 - 0x5802_DFFF	12KB	Reserved	Reserved. When accessed, results in **RAZ/WI**.	-	-
30	0x5802_E000 - 0x5802_EFFF	4KB	SLOWCLK Watchdog	Watchdog Timer running on SLOWCLK. See SLOWCLK AON timers.	-	S, P
31	0x5802_F000 - 0x5802_FFFF	4KB	SLOWCLK Timer	Timer running on SLOWCLK. See SLOWCLK AON timers.	4	S-PPC, P-PPC
32	0x5803_0000 - 0x5803_FFFF	64KB	Reserved	Reserved. When accessed, results in bus error.	-	-

[1] Legend

- NS-PPC: Non-secure access only, gated by a PPC.
  - S-PPC: Secure access only, gated by a PPC.
  - S: Secure access only.
  - NS: Non-secure access only.
  - P: Privilege access only.
  - UP: Unprivileged and privilege access allowed.

- P-PPC: Unprivileged access controlled by a PPC.

4.7.1 SYSINFO register block

The System Information register block provides information on the system configuration and identity. This register block is RO and is accessible by accesses of any security attributes.

This module resides at base address `0x5802_0000` in the Secure region, and `0x4802_0000` in the Non-secure region.

Details of each register are described in separate sections. The width of all registers is 32-bit.

Table 4-54: System Information register map

Offset	Name	Type	Reset value	Description
`0x000`	SOC_IDENTITY	RO	Configurable	SoC Identity Register
`0x004`	SYS_CONFIG0	RO	Configurable	System Hardware Configuration 0 Register
`0x008`	SYS_CONFIG1	RO	Configurable	System Hardware Configuration 1 Register
`0x00C`	SYS_CONFIG2	RO	Configurable	System Hardware Configuration 2 Register
`0x010 - 0xFC4`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0xFC8`	IIDR	RO	Configurable	Subsystem Implementation Identity Register
`0xFCC`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0xFD0`	PIDR4	RO	`0x0000_0004`	Peripheral ID 4
`0xFD4 - 0xFDC`	Reserved	**RAZ/WI**	`0x0000_0000`	Reserved
`0xFE0`	PIDR0	RO	`0x0000_0058`	Peripheral ID 0
`0xFE4`	PIDR1	RO	`0x0000_00B8`	Peripheral ID 1
`0xFE8`	PIDR2	RO	`0x0000_002B`	Peripheral ID 2
`0xFEC`	PIDR3	RO	`0x0000_0000`	Peripheral ID 3
`0xFF0`	CIDR0	RO	`0x0000_000D`	Component ID 0
`0xFF4`	CIDR1	RO	`0x0000_00F0`	Component ID 1
`0xFF8`	CIDR2	RO	`0x0000_0005`	Component ID 2
`0xFFC`	CIDR3	RO	`0x0000_00B1`	Component ID 3

4.7.1.1 SOC_IDENTITY, System-On-Chip (SoC) Identity register

The System-On-Chip (SoC) Identity register provides an area where software can find out about the SoC's part number, its implementor and revision number. These are defined by configuration parameters.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x000

Type

RO

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-55: SOC_IDENTITY bit descriptions

Bits	Name	Description	Type	Reset
[31:20]	SOC_PRODUCT_ID	Configurable value identifying the SoC.	RO	SOCPRTID
[19:16]	SOC_VARIANT	Configurable value indicating major revision of the SoC.	RO	SOCVAR
[15:12]	SOC_REVISION	Configurable value used to distinguish minor revisions of the SoC.	RO	SOCREV
[11:0]	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	RO	SOCIMPLID

When EXPLOGIC_PRESENT = 1, the SoC identity register fields define the TARGETID of the SoC debug port in the subsystem expansion as follows:

- TARGETID[31:28] uses SOCVAR
- TARGETID[27:16] uses SOCPRTID
- TARGETID[15:12] tied to 0x0
- TARGETID[11:1] uses {SOCIMPLID[11:8], SOCIMPLID[6:0]}
- TARGETID[0] tied to 0b1

For more information about TARGETID, see *Arm® Debug Interface Architecture Specification ADIv6.0*.

4.7.1.2 SYS_CONFIG0, SYS_CONFIG1 and SYS_CONFIG2, System Hardware Configuration registers

The System Hardware Configuration registers provides several registers to allow software to query the configuration of the SSE-320 based system.


Note

In these tables, the fields CPU0_TCM_BANK_NUM and CPU0_HAS_SYSTCM refer to TCMs that are implemented on the system interconnect close to each associated core, rather than the TCMs that are implemented within the processor core. SSE-320 does not support TCMs being implemented on the system interconnect and therefore these fields are reserved and **RAZ/WI**.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- SYS_CONFIG0 - `0x004`
- SYS_CONFIG1 - `0x008`
- SYS_CONFIG2 - `0x00C`

Type

RO

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-56: SYS_CONFIG0 bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	Reserved	Reserved	**RAZ/WI**	0x0
[27]	Reserved	Reserved	**RAZ/WI**	0x0
[26:24]	Reserved	Reserved	**RAZ/WI**	0x0
[23:20]	Reserved	Reserved	**RAZ/WI**	0x0
[19]	Reserved	Reserved	**RAZ/WI**	0x0

Bits	Name	Description	Type	Reset
[18:16]	CPU0_TYPE	CPU0 Core Type: • 000 - Does not exist • 100 - Cortex-M85 processor • Others - Reserved	RO	0x4
[15:13]	Reserved	Reserved	RO	0x0
[12:11]	PI_LEVEL	Power Infrastructure Level: • 00 - Basic Level • 01 - Intermediate Level • 10 - Advance Level • Others - Reserved	RO	0x1
[10]	HAS_CSS	It indicates whether the CoreSight SoC-600M-based common debug infrastructure is included. • 0 = No • 1 = Yes	RO	0x0
[9]	LCM_SAM_KMU_PRESENT	Includes LCM, KMU, and SAM: • 0 = No • 1 = Yes	RO	LCM_SAM_KMU_PRESENT (This value is always 1)
[8:4]	VM_ADDR_WIDTH	Volatile Memory Bank Address Width, where the size of each bank is equal to $2^{VM\_ADDR\_WIDTH}$ bytes.	RO	VMADDRWIDTH
[3:0]	NUM_VM_BANK	Number of Volatile Memory Banks.	RO	NUMVMBANK

Table 4-57: SYS_CONFIG1 bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	**RAZ/WI**	0x0000
[15:12]	Reserved	Reserved	**RAZ/WI**	0x0
[11]	Reserved	Reserved	**RAZ/WI**	0x0
[10:8]	Reserved	Reserved	**RAZ/WI**	0x0
[7:4]	Reserved	Reserved	**RAZ/WI**	0x0
[3]	Reserved	Reserved	**RAZ/WI**	0x0
[2:0]	Reserved	Reserved	**RAZ/WI**	0x0

Table 4-58: SYS_CONFIG2 bit descriptions

Bits	Name	Description	Type	Reset
[31:21]	Reserved	Reserved	**RAZ/WI**	0x0000
[20:18]	VM_STRIPE_BIT	Indicates the size of each stripe in each VM when VM_STRIPE_MODE is not '00'. Stripe size is defined as 2^(5+VM_STRIPE_BIT). Supported stripe size from 32bytes to 5Kbytes.	**RAZ/WI**	VMSTRIPEBIT

Bits	Name	Description	Type	Reset
[17:16]	VM_STRIPE_MODE	Indicates the Stripe Mode of the Voliate Memory banks: • 00: No striping • 01: Reserved • 10: VM2 and VM3 are 2-way Striped. No striping on VM0 or VM1. • 11: Reserved	**RAZ/WI**	VMSTRIPEMODE
[15]	Reserved	Reserved	**RAZ/WI**	`0x0`
[14:12]	DMA_TYPE	DMA Core Type: • 000: Does not exist • 001: DMA-350 • Others: Reserved	RO	DMATYPE
[11:3]	Reserved	Reserved	**RAZ/WI**	`0x0`
[2:0]	NPU0_TYPE	NPU 0 Core Type: • 000 - Does not exist • 001 - Reserved • 011 - Ethos-U85 • Others - Reserved	RO	NPU0TYPE

4.7.1.3 IIDR, Subsystem Implementation Identity register

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.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

`0xFC8`

Type

RO

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-59: IIDR bit descriptions

Bits	Name	Description	Type	Reset
[31:20]	IMP_PRODUCT_ID	Configurable value identifying the subsystem implementation.	RO	IMPLPRTID
[19:16]	IMP_VARIANT	Configurable value indicating variant or major revision of the subsystem implementation.	RO	IMPLVAR
[15:12]	IMP_REVISION	Configurable value used to distinguish minor revisions of the subsystem implementation.	RO	IMPLREV
[11:0]	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.	RO	IMPLID

When EXPLOGIC_PRESENT = 1, the subsystem implementation identity register fields define the PIDR values of the MCU debug ROM table - that is the first debug ROM table in the system - in the subsystem expansion, as follows:

- REVISION uses IMPLVAR
- {PART_1, PART_0} uses IMPLPRTID
- {DES_2, DES_1, DES_0} uses IMPLID
- REVAND uses IMPLREV

For more information about PIDR registers of debug ROM table, see *Arm® CoreSight™ Architecture Specification v3.0*.

4.7.2 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 `0x5802_1000` in the Secure region.

The System Control register block is Secure Privilege access only. For write access to these registers, only 32-bit writes are supported. Any Byte and Half-word writes result in its write data ignored.

The following table shows the details of this register block. The width of all registers is 32-bit.

This System Control Registers Block resides in the PD_AON power domain.

Table 4-60: System Control register map

Offset	Name	Type	Reset value	Description
0x000	SECDBGSTAT	RO	Configurable	Secure Debug Configuration Status Register
0x004	SECDBGSET	RW	0x0000_0000	Secure Debug Configuration Set Register
0x008	SECDBGCLR	WO	0x0000_0000	Secure Debug Configuration Clear Register
0x00C	SCSECCTRL	RW	0x0000_0000	System Control Security Controls Register

Offset	Name	Type	Reset value	Description
0x010	CLK_CFG0	RW	Configurable	Clock Configuration Register 0
0x014	CLK_CFG1	RW	Configurable	Clock Configuration Register 1
0x018	CLOCK_FORCE	RW	Configurable	Clock forces
0x01C	CLK_CFG2	RW	Configurable	Clock Configuration Register 2
0x020 - 0x0FF	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x100	RESET_SYNDROME	RW	0x0000_0001	Reset syndrome
0x104	RESET_MASK	RW	Configurable	Reset mask
0x108	SWRESET	WO	0x0000_0000	Software reset
0x10C	GRETREG	RW	0x0000_0000	General Purpose Retention Register
0x110	INITSVTOR0	RW	Configurable	CPU 0 Initial Secure Reset Vector Register
0x114	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x118	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x11C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x120	CPUWAIT	RW	Configurable	CPU Boot Wait Control
0x124	NMI_ENABLE	RW	Configurable	Enabling and Disabling Non Maskable Interrupts
0x128	PPUINTSTAT	RO	0x0000_0000	PPU Interrupt Status
0x12C - 0x1F8	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x1FC	PWRCTRL	RW	0x0000_0003	Power Configuration and Control
0x200	PDCM_PD_SYS_SENSE	RW	Configurable	PDCM PD_SYS sensitivity
0x204	PDCM_PD_CPU0_SENSE	RO	0x0000_0000	PDCM PD_CPU0 sensitivity
0x208	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x20C	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x210	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x214	PDCM_PD_VMR0_SENSE	RW	0x4000_0000	PDCM PD_VMR0 sensitivity
0x218	PDCM_PD_VMR1_SENSE	RW	0x4000_0000	PDCM PD_VMR1 sensitivity
0x21C	PDCM_PD_VMR2_SENSE	RW	0x4000_0000	PDCM PD_VMR2 sensitivity
0x220	PDCM_PD_VMR3_SENSE	RW	0x4000_0000	PDCM PD_VMR3 sensitivity
0x224 - 0x248	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0x24C	PDCM_PD_MGMT_SENSE	RW	Configurable	PDCM PD_MGMT sensitivity
0x250 - 0x258	Reserved	RO	0x0000_0000	Reserved
0x25C	LCM_DCU_FORCE_DISABLE	RW	Configurable	Disable LCM DCU
0x24C	Reserved	RO	0x0000_0000	Reserved
0xFD0	PIDR4	RO	0x0000_0004	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	RO	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_0054	Peripheral ID 0
0xFE4	PIDR1	RO	0x0000_00B8	Peripheral ID 1
0xFE8	PIDR2	RO	0x0000_004B	Peripheral ID 2
0xFEC	PIDR3	RO	0x0000_0000	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	Component ID 0
0xFF4	CIDR1	RO	0x0000_00F0	Component ID 1

Offset	Name	Type	Reset value	Description
0xFF8	CIDR2	RO	0x0000_0005	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	Component ID 3

4.7.2.1 SECDBGSTAT, SECDBGSET and SECDBGCLR, Secure Debug Configuration registers

The Secure Debug Configuration registers are used to select the source value for the Secure Debug Authentication, DBGEN, NIDEN, SPIDEN, SPNIDEN, DAPACCEN, and Debug Access Controls, DAPDSSACCEN, and SYSDSSACCEN0.

A selector is provided for each signal, to select between an internal register value and the value on the boundary of the Subsystem.

Secure software can set or clear the internal register and selector values by setting the associated bit in the SECDBGSET register or in the SECDBGCLR register, respectively:

- To set DBGEN_I value to HIGH, write 1 to the SECDBGSET.DBGEN_I_SET register.
- To set DBGEN_SEL value to HIGH, write 1 to the SECDBGSET.DBGEN_SEL_SET register.
- To set DBGEN_I value to LOW, write 1 to the SECDBGCLR.DBGEN_I_CLR register.
- To set DBGEN_SEL value to LOW, write 1 to the SECDBGSET.DBGEN_SEL_CLR register.

Secure software can read the output values used system wide by reading the associated SECDBGSTAT register bit. Secure software can read internal register values by reading SECDBGSET:

- To read the output value of DBGEN, read the SECDBGSTAT.DBGEN_STATUS register.
- To read the value of DBGEN_I, read the SECDBGSET.DBGEN_I_SET register.
- To read the selector value DBGEN_SEL, read the SECDBGSTAT for the DBGEN_SEL_STATUS field.

For example, the source of DBGEN value used in the system is selected by the DBGEN_SEL where:

- If DBGEN_SEL is LOW, the input DBGENIN signal defines the system wide DBGEN value.
- If DBGEN_SEL is HIGH the internal register value DBGEN_I defines the system wide DBGEN value.

The DBGEN value is also made available to external expansion logic through the DBGEN output signal of the subsystem.

Selector Disable Configuration options are provided to allow each of the selector to be forced to zero, forcing the associated SEL_STATUS field to LOW, forcing each respective debug control output to use its external value:

- DBGENSELDIS for disabling DBGEN_SEL
- NIDENSELDIS for disabling NIDEN_SEL

- SPIDENSELDIS for disabling SPIDEN_SEL
- SPNIDENSELDIS for disabling SPNIDEN_SEL
- DAPACCENSELDIS for disabling DAPACCEN_SEL
- DAPDSSACCENSELDIS for disabling DAPDSSACCEN_SEL

These can be used to disable the ability for Secure firmware to modify or override the Secure Debug Authentication and the Debug Access Controls values.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

- SECDBGSTAT - `0x000`
- SECDBGSET - `0x004`
- SECDBGCLR - `0x008`

Type

- SECDBGSTAT - RO
- SECDBGSET - RW
- SECDBGCLR - WO

Reset value

- SECDBGSTAT - Configurable
- SECDBGSET - `0x0000_0000`
- SECDBGCLR - `0x0000_0000`

These registers are reset by nCOLDRESETAON.

Bit descriptions

These registers reside in the PD_AON power domain.

Table 4-61: SECDBGSTAT bit descriptions

Bits	Name	Description	Type	Reset
[31]	Reserved	Reserved.	**RAZ/WI**	`0x00`
[30]	SYSDSSACCENSELDIS_STATUS	Returns the SYSDSSACCENSELDIS configuration value when read. Ignores Writes. Reserved and **RAZ/WI** if HASCSS = 0.	RO	SYSDSSACCENSELDIS

Bits	Name	Description	Type	Reset
[29]	DAPDSSACCENSELDIS_STATUS	Returns the DA PDSSACCENSELDIS configuration value when read.	RO	DAPDSSACCENSELDIS
[28]	DAPACCENSELDIS_STATUS	Returns the DAPACCENSELDIS configuration value when read.	RO	DAPACCENSELDIS
[27]	SPNIDENSELDIS_STATUS	Returns the SPNIDENSELDIS configuration value when read.	RO	SPNIDENSELDIS
[26]	SPIDENSELDIS_STATUS	Returns the SPIDENSELDIS configuration value when read.	RO	SPIDENSELDIS
[25]	NIDENSELDIS_STATUS	Returns the NIDENSELDIS configuration value when read.	RO	NIDENSELDIS
[24]	DBGENSELDIS_STATUS	Returns the DBGENSELDIS configuration value when read.	RO	DBGENSELDIS
[23:18]	Reserved	Reserved.	**RAZ/WI**	0x0000
[17]	SYSDSSACCEN_SEL_STATUS	Reserved. Active-HIGH System Mapped Debug Access Enable Selector Value. Used as a common selector for all SYSDSSACCEN<n>_ STATUS. This bit returns the SYSDSSACCEN_SEL value Forced to Zero if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0.	RO	0x00
[16]	SYSDSSACCENX_STATUS	Active-HIGH System Mapped Debug Access for Implementation Defined Manager(s). Reserved and **RAZ/WI** if HASCSS = 0. This bit reflects the value on the SYSDSSACCENX pin.	RO	SYSDSSACCENX
[15]	Reserved	Reserved.	**RAZ/WI**	0x0000
[14]	Reserved	Reserved.	**RAZ/WI**	0x0000
[13]	Reserved	Reserved.	**RAZ/WI**	0x0000
[12]	SYSDSSACCEN0_STATUS	Active-HIGH System Mapped Debug Access for CPU0 Enable Value. This bit reflects the value on the SYSDSSACCEN0 after selection. Reserved and **RAZ/WI** if HASCSS = 0.	RO	SYSDSSACCEN0
[11]	DAPDSSACCEN_SEL_STATUS	Active High DAP to Debug Subsystem Access Enable Selector Value. This bit returns the DAPDSSACCEN_SEL value. Forced to Zero if DAPDSSACCENSELDIS = 1.	RO	0x00
[10]	DAPDSSACCEN_STATUS	Active High DAP to Debug Subsystem Access Enable Value. This bit reflects the value on the DAPDSSACCEN pin.	RO	DAPDSSACCENIN
[9]	DAPACCEN_SEL_STATUS	Active High DAP Access Enable Selector Value. This bit returns the DAPACCEN_SEL value. Forced to Zero if DAPACCENSELDIS = 1.	RO	0x00
[8]	DAPACCEN_STATUS	Active High DAP Access Enable Value. This bit reflects the value on the DAPACCEN pin.	RO	DAPACCENIN

Bits	Name	Description	Type	Reset
[7]	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.	RO	0x00
[6]	SPNIDEN_STATUS	Active High Secure Privilege Non-Invasive Debug Enable Value. This bit reflects the value on the SPNIDEN pin.	RO	SPNIDENIN
[5]	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.	RO	0x00
[4]	SPIDEN_STATUS	Active High Secure Privilege Invasive Debug Enable Value. This bit reflects the value on the SPIDEN pin.	RO	SPIDENIN
[3]	NIDEN_SEL_STATUS	Active High Non-Invasive Debug Enable Selector Value. This bit returns the NIDEN_SEL value. Forced to Zero if NIDENSELDIS = 1.	RO	0x00
[2]	NIDEN_STATUS	Active High Non-Invasive Debug Enable Value. This bit reflects the value on the NIDEN pin.	RO	NIDENIN
[1]	DBGEN_SEL_STATUS	Active High Debug Enable Selector Value. This bit returns the DBGEN_SEL value. Forced to Zero if DBGENSELDIS = 1.	RO	0x00
[0]	DBGEN_STATUS	Active High Debug Enable Value. This bit reflects the value on the DBGEN pin.	RO	DBGENIN

Table 4-62: SECDBGSET bit descriptions

Bits	Name	Description	Type	Reset
[31:18]	Reserved	Reserved.	**RAZ/WI**	0x0000
[17]	SYSDSSACCEN_SEL_SET	Set Active-HIGH System Mapped Debug Access Enable Selector Value. Write HIGH to set SYSDSSACCEN_SEL. Reserved and **RAZ/WI** if HASCSS = 0 or SYSDSSACCENSELDIS = 1.	RAZW1S	0x0
[16]	SYSDSSACCENX_I_SET	Set internal version of Active-HIGH System Mapped Debug Access for Implementation Defined Manager(s) Enable. Write HIGH to set SYSDSSACCENX_I. When read returns SYSDSSACCENX_I. Reserved and **RAZ/WI** if HASCSS = 0 or SYSDSSACCENSELDIS = 1.	RW1S	0x0
[15]	SYSDSSACCEN3_I _SET	Set internal version of Active-HIGH System Mapped Debug Access for CPU 3 Enable Set Register. Write HIGH to set SYSDSSACCEN3_I. When read returns SYSDSSACCEN3_I. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 3 or SYSDSSACCENSELDIS = 1. When read, this returns the internal SYSDSSACCEN3 register value before selection.	RW1S	0x0
[14]	SYSDSSACCEN2_I _SET	Set internal version of Active-HIGH System Mapped Debug Access for CPU 2 Enable. Write HIGH to set SYSDSSACCEN2_I. When read returns SYSDSSACCEN2_I. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 2 or SYSDSSACCENSELDIS = 1.	RW1S	0x0
[13]	SYSDSSACCEN1_I _SET	Set internal version of Active-HIGH System Mapped Debug Access for CPU 1 Enable. Write HIGH to set SYSDSSACCEN1_I. When read returns SYSDSSACCEN1_I. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 1 or SYSDSSACCENSELDIS = 1.	RW1S	0x0

Bits	Name	Description	Type	Reset
[12]	SYSDSSACCEN0_I _SET	Set internal version of Active-HIGH System Mapped Debug Access for CPU 0 Enable. Write HIGH to set SYSDSSACCEN0_I. When read returns SYSDSSACCEN0_I. Reserved and **RAZ/WI** if HASCSS = 0 or SYSDSSACCENSELDIS = 1.	RW1S	0x0
[11]	DAPDSSACCEN_SEL_SET	Set Active-HIGH DAP to Debug Subsystem Access Enable Selector. Write HIGH to set DAPDSSACCEN_SEL. **RAZ/WI** if DAPDSSACCENSELDIS = 1.	RAZW1S	0x0
[10]	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.	RW1S	0x0
[9]	DAPACCEN_SEL_SET	Set Active-HIGH DAP Access Enable Selector. Write HIGH to set DAPACCEN_SEL. **RAZ/WI** if DAPACCENSELDIS = 1.	RAZW1S	0x0
[8]	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.	RW1S	0x0
[7]	SPNIDEN_SEL_SET	Set Active-HIGH Secure Privileged Non-Invasive Debug Enable Selector. Write HIGH to set SPNIDEN_SEL. **RAZ/WI** if SPNIDENSELDIS = 1.	RAZW1S	0x0
[6]	SPNIDEN_I _SET	Set internal version of Active-HIGH Secure Privileged Non-Invasive Debug Enable. Write HIGH to set SPNIDEN_I. When read returns SPNIDEN_I. **RAZ/WI** if SPNIDENSELDIS = 1.	RW1S	0x0
[5]	SPIDEN_SEL_SET	Set Active-HIGH Secure Privileged Invasive Debug Enable Selector. Write HIGH to set SPIDEN_SEL. **RAZ/WI** if SPIDENSELDIS = 1.	RAZW1S	0x0
[4]	SPIDEN_I _SET	Set internal version of Active-HIGH Secure Privileged Invasive Debug Enable. Write HIGH to set SPIDEN_I. When read returns SPIDEN_I. **RAZ/WI** ifSPIDENSELDIS = 1.	RW1S	0x0
[3]	NIDEN_SEL_SET	Set Active-HIGH Non-Invasive Debug Enable Selector. Write HIGH to set NIDEN_SEL. **RAZ/WI** if NIDENSELDIS = 1.	RAZW1S	0x0
[2]	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.	RW1S	0x0
[1]	DBGEN_SEL_SET	Set Active-HIGH Debug Enable Selector. Write HIGH to set DBGEN_SEL. **RAZ/WI** if DBGENSELDIS = 1.	RAZW1S	0x0
[0]	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.	RW1S	0x0

Table 4-63: SECDBGCLR bit descriptions

Bits	Name	Description	Type	Reset
[31:18]	Reserved	Reserved.	**RAZ/WI**	0x0000
[17]	SYSDSSACCEN_SEL_CLR	Clears Active-HIGH System Mapped Debug Access Enable Selector Value. Write HIGH to clear SYSDSSACCEN_SEL. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0.	RAZW1C	0x0
[16]	SYSDSSACCENX_I _CLR	Clears internal version of Active High System Mapped Debug Access for Implementation Defined Manager(s) Enable. Write HIGH to clear SYSDSSACCENX_I. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0.	RAZW1C	0x0
[15]	SYSDSSACCEN3_I _CLR	Clears internal version of Active High System Mapped Debug Access for CPU 3 Enable. Write HIGH to clear SYSDSSACCEN3_I. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 3.	RAZW1C	0x0

Bits	Name	Description	Type	Reset
[14]	SYSDSSACCEN2_I _CLR	Clears internal version of Active High System Mapped Debug Access for CPU 2 Enable. Write HIGH to clear SYSDSSACCEN2_I. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 2.	RAZW1C	0x0
[13]	SYSDSSACCEN1_I _CLR	Clears internal version of Active High System Mapped Debug Access for CPU 1 Enable. Write HIGH to clear SYSDSSACCEN1_I. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0 or NUMCPU < 1.	RAZW1C	0x0
[12]	SYSDSSACCEN0_I _CLR	Clears internal version of Active High System Mapped Debug Access for CPU 0 Enable. Write HIGH to clear SYSDSSACCEN0_I. **RAZ/WI** if SYSDSSACCENSELDIS = 1. Reserved and **RAZ/WI** if HASCSS = 0.	RAZW1C	0x0
[11]	DAPDSSACCEN_SEL_CLR	Clears Active-HIGH DAP to Debug Subsystem Access Enable Selector. Write HIGH to clear DAPDSSACCEN_SEL. **RAZ/WI** if DAPDSSACCENSELDIS = 1.	RAZW1C	0x0
[10]	DAPDSSACCEN_I _CLR	Clears internal version of Active High DAP to Debug Subsystem Access Enable. Write HIGH to clear DAPDSSACCEN_I. **RAZ/WI** if DAPDSSACCENSELDIS = 1.	RAZW1C	0x0
[9]	DAPACCEN_SEL_CLR	Clears Active-HIGH DAP Access Enable Selector. Write HIGH to clear DAPACCEN_SEL. **RAZ/WI** if DAPACCENSELDIS = 1.	RAZW1C	0x0
[8]	DAPACCEN_I _CLR	Clears internal version of Active High DAP Access Enable. Write HIGH to clear DAPACCEN_I. **RAZ/WI** if DAPACCENSELDIS = 1.	RA/W1C	0x0
[7]	SPNIDEN_SEL_CLR	Clears Active-HIGH Secure Privileged Non-Invasive Debug Enable Selector. Write HIGH to clear SPNIDEN_SEL. **RAZ/WI** if SPNIDENSELDIS = 1.	RAZW1C	0x0
[6]	SPNIDEN_I _CLR	Clears internal version of Active High Secure Privileged Non-Invasive Debug Enable. Write HIGH to clear SPNIDEN_I. **RAZ/WI** if SPNIDENSELDIS = 1.	**RAZ**/ W1C	0x0
[5]	SPIDEN_SEL_CLR	Clears Active-HIGH Secure Privileged Invasive Debug Enable Selector. Write HIGH to clear SPIDEN_SEL. **RAZ/WI** if SPIDENSELDIS = 1.	RAZW1C	0x0
[4]	SPIDEN_I _CLR	Clears internal version of Active High Secure Privileged Invasive Debug Enable. Write HIGH to clear SPIDEN_I. **RAZ/WI** if SPIDENSELDIS = 1.	RAZW1C	0x0
[3]	NIDEN_SEL_CLR	Clears Active-HIGH Non-Invasive Debug Enable Selector. Write HIGH to clear NIDEN_SEL. **RAZ/WI** if NIDENSELDIS = 1.	RAZW1C	0x0
[2]	NIDEN_I _CLR	Clears internal version of Active High Non-Invasive Debug Enable. Write HIGH to clear NIDEN_I. **RAZ/WI** if NIDENSELDIS = 1.	RAZW1C	0x0
[1]	DBGEN_SEL_CLR	Clears Active-HIGH Debug Enable Selector. Write HIGH to clear DBGEN_SEL. **RAZ/WI** if DBGENSELDIS = 1.	RAZW1C	0x0
[0]	DBGEN_I _CLR	Clears internal version of Active High Debug Enable. Write HIGH to clear DBGEN_I. **RAZ/WI** if DBGENSELDIS = 1.	**RAZ**/ W1C	0x0

4.7.2.2 SCSECCTRL, System Control Security Control register

The System Control Security Control provides a register bit to set the Secure Configuration lock of this register block.

These registers are reset by nCOLDRESETAON.

These registers reside in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x00C

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-64: SCSECCTRL bit descriptions

Bits	Name	Description	Type	Reset
[31:3]	Reserved	Reserved	RAZ/WI	0x0000_0000
[2]	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.	RW1S	0x00
[1:0]	Reserved	Reserved	RAZ/WI	0x00

4.7.2.3 CLK_CFG0, CLK_CFG1 and CLK_CFG2

The CLK_CFG0, CLK_CFG1, and CLK_CFG2 registers provide control register fields to drive expansion clock generation logic for this subsystem.

Each clock control handshake comprises of a configuration request field, that is *CLKCFG, and a status field, 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 value is the same as the *CLKCFG value before performing any other operations.

In addition, if it is the first write to the register after cold reset, software has to poll that the targeted *CLKCFG field value (default value set by the related configuration input) matches its associated *CLKCFGSTATUS field value before the write occurs. The actual number of bits being implemented in the related CLKCFG and CLKCFGSTATUS signals is defined in the following tables.

These registers are reset by nCOLDRESETAON.

These registers reside in the PD_AON power domain.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Address offset

- CLK_CFG0 - `0x010`
- CLK_CFG1 - `0x014`
- CLK_CFG2 - `0x01C`

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following tables show the register bit assignments.

Table 4-65: CLK_CFG0 bit descriptions

Bits	Name	Description	Type	Reset
[31:28]	Reserved	Reserved	RAZ/WI	0x00
[27:24]	Reserved	Reserved	RAZ/WI	0x00
[23:20]	Reserved	Reserved	RAZ/WI	0x00
[19:16]	CPU0CLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for CPU0CLK.	RO	CPU0CLKCFGSTATUS
[15:12]	Reserved	Reserved	RAZ/WI	0x00
[11:8]	Reserved	Reserved	RAZ/WI	0x00
[7:4]	Reserved	Reserved	RAZ/WI	0x00
[3:0]	CPU0CLKCFG	Clock Configuration value that drives CPU0CLKCFG signals.	RW	CPU0CLKCFGRST

Table 4-66: CLK_CFG1 bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved	RAZ/WI	0x000
[23:20]	AONCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for AONCLK.	RO	AONCLKCFGSTATUS
[19:16]	SYSCLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for SYSCLK.	RO	SYSCLKCFGSTATUS
[15:8]	Reserved	Reserved	RAZ/WI	0x000
[7:4]	AONCLKCFG	Clock Configuration value that drives AONCLKCFG signals.	RW	AONCLKCFGRST

Bits	Name	Description	Type	Reset
[3:0]	SYSCLKCFG	Clock Configuration value that drives SYSCLKCFG signals.	RW	SYSCLKCFGRST

Table 4-67: CLK_CFG2 bit descriptions

Bits	Name	Description	Type	Reset
[31:20]	Reserved	Reserved	**RAZ/WI**	0x000
[19:16]	NPU0CLKCFGSTATUS	Clock Configuration Status value that reports the status of clock control for NPU0CLK. **RAZ/WI** if NUMNPU < 1	RO	NPU0CLKCFGSTATUS
[15:4]	Reserved	Reserved	**RAZ/WI**	0x000
[3:0]	NPU0CLKCFG	Clock Configuration value that drives NPU0CLKCFG signals. **RAZ/WI** if NUMNPU < 1 **RAZ/WI** if NUMNPU < 1	RW	NPU0CLKCFGRST

4.7.2.4 CLOCK_FORCE, Clock force register

The Clock Force register allows software to override dynamic clock gating that might be implemented in the system and keep each clock running.

Bits 0 to 8 are clock forces that do not apply to clock gates within the design that are responsible for the gating of clocks when gating power to a power gated region. Instead, it is applied to hierarchical dynamic clock gating within the system, with one bit for each power domain. Forcing a clock ON can reduce the latency that is a result of dynamic clock control, but generally has a reverse side-effect of increasing the dynamic power consumption of the system.


Note

All clock force default values of these bits are set to HIGH at reset. This allows the system to boot in case any hierarchical dynamic clock control implementation is non-functional. The associated clock force register bit must be cleared to enable hierarchical dynamic clock control for each power domain.

Bits 16 to 23 are clock forces used to force the external clock generator to continue to generate its clock. This can be used to avoid clock generators like PLLs from turning off, which can result in a long wakeup time.

These registers are reset by nCOLDRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x018

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-68: CLOCK_FORCE bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved.	**RAZ/WI**	0x0000
[23]	NPU0CLK_FORCE	Set HIGH to request the input NPU0CLK source to stay ON. The field is reserved and **RAZ/WI** if NUMNPU < 1.	RW	0x0
[22]	Reserved	Reserved.	**RAZ/WI**	0x0
[21]	Reserved	Reserved.	**RAZ/WI**	0x0
[20]	Reserved	Reserved.	**RAZ/WI**	0x0
[19]	CPU0CLK_FORCE	Set HIGH to request the input CPU0CLK source to stay ON.	RW	0x0
[18]	Reserved	Reserved for DEBUGCLK_FORCE.	RW	0x0
[17]	SYSCLK_FORCE	Set HIGH to request the input SYSCLK source to stay ON.	RW	0x0
[16]	AONCLK_FORCE	Set HIGH to request the input AONCLK source to stay ON.	RW	0x0
[15:12]	Reserved	Reserved.	**RAZ/WI**	0x00000000
[11]	Reserved	Reserved.	**RAZ/WI**	0x0
[10]	Reserved	Reserved.	**RAZ/WI**	0x0
[9]	Reserved	Reserved.	**RAZ/WI**	0x0
[8]	NPU0_CLKFORCE	Set HIGH to force PD_NPU0 Local clocks to run. The field is reserved and **RAZ/WI** if NUMNPU < 1.	RW	0x01
[7]	Reserved	Reserved.	RO	0x00
[6]	Reserved	Reserved.	RO	0x00
[5]	Reserved	Reserved.	RO	0x00
[4]	CPU0_CLKFORCE	Set HIGH to force all clocks in PD_CPU0 to run.	RW	0x01
[3]	Reserved	Reserved.	RO	0x00

Bits	Name	Description	Type	Reset
[2]	DEBUG_CLKFORCE	Set HIGH to force all clocks in PD_DEBUG to run.	RW	0x01
[1]	SYS_CLKFORCE	Set HIGH to force all clocks in PD_SYS to run.	RW	0x01
[0]	MGMT_CLKFORCE	Set HIGH to force all clocks in PD_MGMT domain to run.	RW	0x01

4.7.2.5 RESET_SYNDROME

The RESET_SYNDROME register stores the reason for the last Reset event. Writing HIGH to a bit results in that bit write value to be ignored and the bit maintaining its previous value. RESET_SYNDROME is cleared by software writing zero to each bit to clear. If after starting from a reset event, RESET_SYNDROME is not cleared, on another reset event, the register may no longer accurately reflect the last reset event. CPU0LOCKUP does not actually generate reset, but when HIGH, it indicates that a CPU has locked-up and could be a precursor to another reset event, for example, watchdog timer reset request.

The RESET_SYNDROME register always stores Reset events regardless of the COLDRESET_MODE configuration and even if the Cold reset generation is performed externally into the system through the HOSTRESETREQ reset request signal. Also note that CPU0LOCKUP events are always stored, and only reset requests that are not masked by their respective mask bits are stored in the register.

Configurations

This register implementation depends on the configuration of individual fields.

Attributes

Width

32-bit

Power domain

The RESET_SYNDROME register resides in the PD_AON power domain but can also reside in PD_MGMT power domain if its states are saved and restored when entering and then leaving the lower power state, respectively and if its functionality is maintained while PD_MGMT is in low power state.

Type

RW

Reset value

nPORESETAON

Usage constraints

This register is Secure privileged access only. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

Table 4-69: RESET_SYNDROME bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	SWSYN	Software defined reset syndrome This field is set after reset caused by software setting the SWSYN field and the SWCOLDRESETREQ or SWWARMRESETREQ field in the SWRESET register. It may serve as a messaging box between software running before the reset and after the reset.	RW0C	0x0
[23:20]	Reserved	Reserved.	**RAZ/WI**	0x00
[19]	SWWARMRESETREQ	Software Warm Reset Request	RW0C	0x00
[18]	WARMRESETREQ	Subsystem Hardware Warm Reset Request Input If WARMRESETREQ is not implemented, this bit is reserved and **RAZ/WI**.	RW0C	0x0
[17]	SAMCRSTREQ	Security Alarm Manager Cold Reset Request.	RW0C	0x0
[16]	SAMWRSTREQ	Security Alarm Manager Warm Reset Request.	RW0C	0x0
[15]	Reserved	Reserved	**RAZ/WI**	0x0
[14]	Reserved	Reserved	RW0C	0x0
[13]	Reserved	Reserved	RW0C	0x0
[12]	CPU0LOCKUP	CPU 0 Lockup Status.	RW0C	0x0
[11]	Reserved	Reserved	RW0C	0x0
[10]	Reserved	Reserved	RW0C	0x0
[9]	Reserved	Reserved	RW0C	0x0
[8]	CPU0RSTREQ	CPU 0 Warm Reset Request.	RW0C	0x0
[7]	HOSTRESETREQ	Host Level Cold Reset Request Input.	RW0C	0x0
[6]	LCMRSTREQ	Lifecycle Manager Secure provisioning warm reset Request.	RW0C	0x0
[5]	SWCOLDRESETREQ	Software Cold Reset Request.	RW0C	0x0
[4]	RESETREQ	Subsystem Hardware Cold Reset Request Input.	RW0C	0x0
[3]	SLOWCLKWDRSTREQ	SLOWCLK Watchdog Cold Reset Request.	RW0C	0x0
[2]	SWDRSTREQ	Secure Watchdog Cold Reset Request.	RW0C	0x0
[1]	NSWDRSTREQ	Non-secure Watchdog Cold Reset Request.	RW0C	0x0
[0]	PoR	Power-On-Reset	RW0C	0x01

4.7.2.6 RESET_MASK, Rest mask register

The RESET_MASK register allows the software to control which reset sources are merged to generate the system wide warm reset,nWARMRESETAON or the nCOLDRESETAON signal.

Set each bit to HIGH to enable each source. This register is reset by the nWARMRESETAON.

Note

If cleared, each of these mask bits prevents the reset source from generating the reset, and also prevents the associated RESET_SYNDROME register bit from recording the event.

This register resides in the PD_AON power domain.

This register is reset by the nWARMRESETAON.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x104

Type

RW

Reset value

See individual bit resets

Bit descriptions

The following table shows the register bit assignments.

Table 4-70: RESET_MASK bit descriptions

Bits	Name	Description	Type	Reset
[31:12]	Reserved	Reserved	**RAZ/WI**	0x000000
[11]	Reserved	Reserved	**RAZ/WI**	0x00
[10]	Reserved	Reserved	**RAZ/WI**	0x00
[9]	Reserved	Reserved	**RAZ/WI**	0x00
[8]	CPU0RSTREQEN	CPU0 Warm Reset Request Enable	RW	CPU0RSTREQENRST
[7:2]	Reserved	Reserved	**RAZ/WI**	0x000
[1]	NSWDRSTREQEN	Non-secure Watchdog Reset Enable	RW	0x00
[0]	Reserved	Reserved	**RAZ/WI**	0x00

4.7.2.7 SWRESET

The SWRESET register allows software to request for a system Cold reset or Warm reset. To request for a reset, write '1' to the corresponding register bit. The register always returns zeros.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x108

Type

WO

Power domain

This register resides in the PD_AON or in the PD_MGMT power domain. Its content is not retained when PD_MGMT is turned off in HIBERNATION1 state when PILEVEL = 2.

Usage constraints

This register is Secure privileged access only. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

Table 4-71: SWRESET bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	SWSYN	Software Reset Software defined reset syndrome. It serves as a messaging box between software running before the reset and software running after the reset. The SWSYN is written along with SWCOLDRESETREQ or SWWARMRESETREQ, otherwise it is ignored. It propagates to the RESET_SYNDROME register when the reset occurs.	RAZW1S	0x0
[23:20]	Reserved	Reserved.	**RAZ/WI**	0x0
[19]	SWWARMRESETREQ	Software Warm Reset Request. Write '1' to set to HIGH, to request a Warm reset.	RAZW1S	0x0
[18:6]	Reserved	Reserved.	**RAZ/WI**	0x0000
[5]	SWCOLDRESETREQ	Software Cold Reset Request. Write '1' to set to HIGH, to request a Cold reset.	RAZW1S	0x0
[4:0]	Reserved	Reserved.	**RAZ/WI**	0x000

4.7.2.8 GRETREG, General Purpose Retention Register

The General Purpose Retention Register provides 16 bits of retention register for general storage through the HIBERNATION0 System Power States.

This register is reset by nCOLDRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x10C

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-72: GRETREG bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	Reserved	Reserved	**RAZ/WI**	0x00000
[15:0]	GRETREG	General Purpose Retention Register	RW	0x00000

4.7.2.9 INITSVTOR0

This register defines the CPU0 Initial Secure Vector table offset (VTOR_S.TBLOFF[31:7]) out of reset.

This register is reset by nWARMRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x110

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-73: INITSVTOR0 Register

Bits	Name	Description	Type	Reset
[31:7]	INITSVTOR0	Default Secure Vector table offset at reset for CPU0.	RW	INITSVTOR0RST[31:7]

Bits	Name	Description	Type	Reset
[6:1]	Reserved	Reserved	**RAZ/WI**	0x000
[0]	INITSVTOR0LOCK	Lock INITSVTOR0. When set to '1', stops any further writes to INITSVTOR0 and INITSVTOR0LOCK fields. Cleared only by warm reset.	RW1S	0x00

4.7.2.10 CPUWAIT

This Register provides controls to force CPU0 to wait after reset rather than boot immediately. This allows another entity in the expansion system or the debugger to access the system prior to the CPU booting.

This register is reset by nCOLDRESETAON only.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x120

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-74: CPUWAIT bit descriptions

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	**RAZ/WI**	0x0000_0000
[3]	Reserved	Reserved.	**RAZ/WI**	0x00
[2]	Reserved	Reserved.	**RAZ/WI**	0x00
[1]	Reserved	Reserved.	**RAZ/WI**	0x00
[0]	CPU0WAIT	CPU 0 waits at boot. • '0': boot normally. • '1': wait at boot. When CPU0WAITCLR input is 0b1, this bit is cleared.	RW	CPU0WAITRST

4.7.2.11 NMI_ENABLE, NMI_ENABLE register

This register provides controls to enable or disable the internally or externally generated Non-Maskable Interrupt sources from generating an NMI interrupt on CPU0.

This register is reset by nWARMRESETAON and its reset value is defined by the configuration options.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

`0x124`

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-75: NMI_ENABLE bit descriptions

Bits	Name	Description	Type	Reset
[31:20]	Reserved	Reserved	RAZ/WI	0x0000
[19]	Reserved	Reserved	RAZ/WI	0x00
[18]	Reserved	Reserved	RAZ/WI	0x00
[17]	Reserved	Reserved	RAZ/WI	0x00
[16]	CPU0_EXPNMI_ENABLE	CPU0 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	RW	CPU0EXPNMIENABLERST
[15:4]	Reserved	Reserved	RAZ/WI	0x0000
[3]	Reserved	Reserved	RAZ/WI	0x00

Bits	Name	Description	Type	Reset
[2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	Reserved	Reserved	**RAZ/WI**	0x00
[0]	CPU0_INTNMI_ENABLE	CPU0 Internally Sourced NMI Enable. This determines if the subsystem internally generated NMI interrupt sources can raise NMI interrupt on CPU0: • HIGH, allowed. • LOW, is masked and not allowed	RW	CPU0INTNMIENABLERST

4.7.2.12 PPUINTSTAT, PPU interrupt status register

This is the PPU interrupt status register for SSE-320

This register resides in the PD_AON power domain.

This register is reset by nWARMRESETAON.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x128

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-76: PPUINTSTAT register bit descriptions

Bits	Name	Description	Type	Reset
[31:12]	Reserved	Reserved	**RAZ/WI**	0x0000_000
[11]	DEBUG_PPU_INTSTAT	Debug PPU Interrupt	RO	0x0
[10]	Reserved	Reserved	**RAZ/WI**	0x0
[9]	Reserved	Reserved	**RAZ/WI**	0x0
[8]	Reserved	Reserved	**RAZ/WI**	0x0
[7]	NPU0_PPU_INTSTAT	If NUMNPU<1, NPU0 PPU interrupt is **RAZ/WI**	RO	0x0

Bits	Name	Description	Type	Reset
[6]	Reserved	Reserved	RAZ/WI	0x0
[5]	Reserved	Reserved	RAZ/WI	0x0
[4]	Reserved	Reserved	RAZ/WI	0x0
[3]	Reserved	Reserved	RAZ/WI	0x0
[2]	CPU0_PPU_INTSTAT	CPU0 PPU interrupt	RO	0x0
[1]	SYS_PPU_INTSTAT	System PPU interrupt	RO	0x0
[0]	MGMT_PPU_INTSTAT	Management PPU Interrupt	RO	0x0

4.7.2.13 PWRCTRL, Power Control register

The Power Control register configures the power control features in SSE-320 System.

This register is reset by nWARMRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bits

Address offset

0x1FC

Type

RW

Reset value

0x0000_0003

Bit descriptions

The following table shows the register bit assignments.

Table 4-77: PWRCTRL bit descriptions

Bits	Name	Description	Type	Reset
[31:2]	Reserved	Reserved.	RAZ/WI	0x0000_0000
[1]	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'.	RW0C	0x01

Bits	Name	Description	Type	Reset
[0]	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 related to PPU registers accessibility, see Power Policy Units.	RW	0x01

4.7.2.14 PDCM_PD_SYS_SENSE, Power Dependency Control Matrix System power domain Sensitivity register

The Power Dependency Control Matrix System power domain (PD_SYS) Sensitivity register defines what keeps awake the PD_SYS power domain and the minimum power state to use when the domain is in its low power state.

This register is reset by nWARMRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x200

Type

RW

Reset value

See individual bit resets.

Bit descriptions

The following table shows the register bit assignments.

Table 4-78: PDCM_PD_SYS_SENSE bit descriptions

Bits	Name	Description	Type	Reset
[31:30]	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	RW	0x00

Bits	Name	Description	Type	Reset
[29:24]	Reserved	Reserved	**RAZ/WI**	0x00
[23]	S_PDCMRETQREQ3	Enable sensitivity to PDCMQRETREQn[3] signal. If set to '1' PD_SYS stays in ON or RET if PDCMRETQREQn[3] signal is HIGH.	RW	0x00
[22]	S_PDCMRETQREQ2	Enable sensitivity to PDCMQRETREQn[2] signal. If set to '1' PD_SYS stays in ON or RET if PDCMRETQREQn[2] signal is HIGH	RW	0x00
[21]	S_PDCMRETQREQ1	Enable sensitivity to PDCMQRETREQn[1] signal. If set to '1' PD_SYS stays in ON or RET if PDCMRETQREQn[1] signal is HIGH.	RW	0x00
[20]	S_PDCMRETQREQ0	Enable sensitivity to PDCMQRETREQn[0] signal. If set to '1' PD_SYS stays in ON or RET if PDCMRETQREQn[0] signal is HIGH	RW	0x00
[19]	S_PDCMONQREQ3	Enables sensitivity to PDCMONQREQn[3] signal. If set to '1', PD_SYS stays ON if PDCMONQREQn[3] signal is HIGH.	RW	0x00
[18]	S_PDCMONQREQ2	Enables sensitivity to PDCMONQREQn[2] signal. If set to '1', PD_SYS stays ON if PDCMONQREQn[2] signal is HIGH.	RW	0x00
[17]	S_PDCMONQREQ1	Enables sensitivity to PDCMONQREQn[1] signal. If set to '1', PD_SYS stays ON if PDCMONQREQn[1] signal is HIGH.	RW	0x00
[16]	S_PDCMONQREQ0	Enables sensitivity to PDCMONQREQn[0] signal. If set to '1', PD_SYS stays ON if PDCMONQREQn[0] signal is HIGH.	RW	0x00
[15]	Reserved	Reserved	**RAZ/WI**	0x00
[14]	Reserved	Reserved	**RAZ/WI**	0x00
[13]	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.	RO	0x00
[12]	Reserved	Reserved.	**RAZ/WI**	0x00
[11:9]	Reserved	Reserved.	**RAZ/WI**	0x000
[8]	Reserved	Reserved.	**RAZ/WI**	0x01
[7]	Reserved	Reserved.	**RAZ/WI**	0x01
[6]	Reserved	Reserved.	**RAZ/WI**	0x01
[5]	S_PD_NPU0_ON	Tied to HIGH. PD_SYS will always try to stay ON if PD_NPU0 is ON. This bit is Reserved and **RAZ/WI** if NUMNPU is 0.	RO	0x01
[4]	Reserved	Reserved.	**RAZ/WI**	0x00
[3]	Reserved	Reserved.	**RAZ/WI**	0x00
[2]	Reserved	Reserved.	**RAZ/WI**	0x00
[1]	S_PD_CPU0_ON	Tied to HIGH. PD_SYS will always try to stay ON if PD_CPU0 is ON.	RO	0x01
[0]	S_PD_SYS_ON	Enable PD_SYS ON Sensitivity. Set this to HIGH to keep PD_SYS awake once powered ON.	RW	0x00

4.7.2.15 PDCM_PD_CPU0_SENSE, Power Dependency Control Matrix System Power domain Sensitivity register

The Power Dependency Control Matrix System Power domain (PD_CPU0) Sensitivity register defines what keeps awake the PD_CPU0 domain, and defines the minimum power state to use when the domain is in its low power state.

This register is reset by nWARMRESETAON.

In SSE-320, PD_CPU0 is not sensitive to any incoming dependencies.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

0x204

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-79: PDCM_PD_CPU0_SENSE bit descriptions

Bits	Name	Description	Type	Reset
[31:24]	Reserved	Reserved.	**RAZ/WI**	0x000
[23]	PDCMRETQREQ3	Tied to LOW. Ignores PDCMRETQREQn[3] signal.	RO	0x00
[22]	PDCMRETQREQ2	Tied to LOW. Ignores PDCMRETQREQn[2] signal.	RO	0x00
[21]	PDCMRETQREQ1	Tied to LOW. Ignores PDCMRETQREQn[1] signal.	RO	0x00
[20]	PDCMRETQREQ0	Tied to LOW. Ignores PDCMRETQREQn[0] signal.	RO	0x00
[19]	PDCMONQREQ3	Tied to LOW. Ignores PDCMONQREQn[3] signal.	RO	0x00
[18]	PDCMONQREQ2	Tied to LOW. Ignores PDCMONQREQn[2] signal.	RO	0x00
[17]	PDCMONQREQ1	Tied to LOW. Ignores PDCMONQREQn[1] signal.	RO	0x00
[16]	PDCMONQREQ0	Tied to LOW. Ignores PDCMONQREQn[0] signal.	RO	0x00
[15]	Reserved	Reserved	**RAZ/WI**	0x00
[14]	Reserved	Reserved	**RAZ/WI**	0x00
[13]	S_PD_DEBUG_ON	Tied to LOW. Ignores PD_DEBUG power state.	RO	0x00

Bits	Name	Description	Type	Reset
[12]	Reserved	Reserved	**RAZ/WI**	0x00
[11:5]	Reserved	Reserved	**RAZ/WI**	0x000
[4]	Reserved	Reserved	**RAZ/WI**	0x00
[3]	Reserved	Reserved	**RAZ/WI**	0x00
[2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	S_PD_CPU0_ON	Tied to LOW. Ignores PD_CPU0 power state.	RO	0x00
[0]	S_PD_SYS_ON	Tied to LOW. Ignores PD_SYS power state.	RO	0x00

4.7.2.16 PDCM_PD_VMR<M>_SENSE, Power Dependency Control Matrix Volatile Memory Region power domain sensitivity register

The Power Dependency Control Matrix Volatile Memory Region <*M*> power domain (PD_VMR<*M*>) sensitivity register defines what keeps awake the PD_VMR<*M*> domain and the minimum power state to use when the domain is in its low power state, where *M* is 0 to NUMVMBANK-1.

This register is reset by nWARMRESETAON.

This register resides in the PD_AON power domain.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Address offset

- PDCM_PD_VMR0_SENSE - 0x214
- PDCM_PD_VMR1_SENSE - 0x218
- PDCM_PD_VMR2_SENSE - 0x21C
- PDCM_PD_VMR3_SENSE - 0x220

Type

RW

Reset value

0x4000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 4-80: PDCM_PD_VMR<M>_SENSE bit descriptions

Bits	Name	Description	Type	Resets
[31:30]	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	RW	0x01
[29:24]	Reserved	Reserved	**RAZ/WI**	0x00
[23]	S_PDCMRETQREQ3	Enable sensitivity to PDCMRETQREQn[3] signal. If set to '1' PD_VMR<M> stays in ON or RET if PDCMRETQREQn[3] signal is HIGH.	RW	0x00
[22]	S_PDCMRETQREQ2	Enable sensitivity to PDCMRETQREQn[2] signal. If set to '1' PD_VMR<M> stays in ON or RET if PDCMRETQREQn[2] signal is HIGH.	RW	0x00
[21]	S_PDCMRETQREQ1	Enable sensitivity to PDCMRETQREQn[1] signal. If set to '1' PD_VMR<M> stays in ON or RET if PDCMRETQREQn[1] signal is HIGH.	RW	0x00
[20]	S_PDCMRETQREQ0	Enable sensitivity to PDCMQRETREQn[0] signal. If set to '1' PD_VMR<M> stays in ON or RET if PDCMRETQREQn[0] signal is HIGH.	RW	0x00
[19]	S_PDCMQREQ3	Enable sensitivity to PDCMONQREQn[3] signal. If set to '1' PD_SYS stays ON if PDCMONQREQn[3] signal is HIGH.	RW	0x00
[18]	S_PDCMQREQ2	Enable sensitivity to PDCMQONREQn[2] signal. If set to '1' PD_SYS stays ON if PDCMONQREQn[2] signal is HIGH.	RW	0x00
[17]	S_PDCMQREQ1	Enable sensitivity to PDCMQONREQn[1] signal. If set to '1' PD_SYS stays ON if PDCMONQREQn[1] signal is HIGH.	RW	0x00
[16]	S_PDCMQREQ0	Enable sensitivity to PDCMQONREQn[0] signal. If set to '1' PD_SYS stays ON if PDCMONQREQn[0] signal is HIGH.	RW	0x00
[15]	Reserved	Reserved.	**RAZ/WI**	0x00
[14]	Reserved	Reserved.	**RAZ/WI**	0x00
[13]	S_PD_DEBUG _ON	Tied to LOW. Ignores PD_DEBUG power state.	RO	0x00
[12]	Reserved	Reserved	**RAZ/WI**	0x00
[11:9]	Reserved	Reserved	**RAZ/WI**	0x000
[8]	Reserved	Reserved	**RAZ/WI**	0x00
[7]	Reserved	Reserved	**RAZ/WI**	0x00
[6]	Reserved	Reserved	**RAZ/WI**	0x00
[5]	S_PD_NPU0_ON	Enable PD_NPU0 sensitivity. If set to '1' PD_VMR<M> will stay ON if PD_NPU0 is ON. This bit is Reserved and **RAZ/WI** if NUMNPU < 1.	RW	0x00
[4]	Reserved	Reserved	**RAZ/WI**	0x00
[3]	Reserved	Reserved	**RAZ/WI**	0x00

Bits	Name	Description	Type	Resets
[2]	Reserved	Reserved	**RAZ/WI**	0x00
[1]	S_PD_CPU0_ON	Enable PD_CPU0 sensitivity. If set to 1 PD_VMR<M> will stay on if PD_CPU0 is on.	RW	0x00
[0]	S_PD_SYS_ON	Tied to LOW. Ignores PD_SYS power state.	RO	0x00

4.7.2.17 PDCM_PD_MGMT_SENSE

The Power Dependency Control Matrix PD_MGMT Power Domain Sensitivity register is used to define what keeps the PD_MGMT domains awake.

Configurations

This register implementation depends on the configuration of individual fields. When PILEVEL < 2, this register does not exist and the register area it occupies are Reserved, and **RAZ/WI**.

Attributes

Width

32-bit

Offset

0x24C

Type

RW

Power domain

This register resides in the PD_AON power domain but can also reside in PD_MGMT power domain if its states are saved and restored when entering and then leaving the lower power state, respectively.

Reset

nWARMRESETAON

Usage constraints

This register is Secure privileged access only. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

Table 4-81: PDCM_PD_MGMT_SENSE bit descriptions

Bits	Name	Description	Type	Default
[31:30]	MIN_PWR_STATE	Defines the Minimum Power State, when PD_MGMT is trying to enter a lower power state: • '00': Minimum power state is OFF, • Others: Reserved.	RO	0x0
[29:24]	Reserved	Reserved.	**RAZ/WI**	0x0

Bits	Name	Description	Type	Default
[23]	S_PDCMRETQREQ3	Enable sensitivity to PDCMRETQREQn[3] signal. If set to '1', PD_MGMT stays ON if PDCMRETQREQn[3] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 4.	RW	0x0
[22]	S_PDCMRETQREQ2	Enable sensitivity to PDCMRETQREQn[2] signal. If set to '1', PD_MGMT stays ON if PDCMRETQREQn[2] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 3.	RW	0x0
[21]	S_PDCMRETQREQ1	Enable sensitivity to PDCMRETQREQn[1] signal. If set to '1', PD_MGMT stays ON if PDCMRETQREQn[1] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 2.	RW	0x0
[20]	S_PDCMRETQREQ0	Enable sensitivity to [PDCMRETQREQn0 signal. If set to '1', PD_MGMT stays ON if PDCMRETQREQn[0] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 1.	RW	0x0
[19]	S_PDCMONQREQ3	Enable sensitivity to PDCMONQREQn[3] signal. If set to '1', PD_MGMT stays ON if PDCMONQREQn[3] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 4.	RW	0x0
[18]	S_PDCMONQREQ2	Enable sensitivity to PDCMQNQREQn[2] signal. If set to '1',PD_MGMT stays ON if PDCMONQREQn[2] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 3.	RW	0x0
[17]	S_PDCMONQREQ1	Enable sensitivity to PDCMONQREQn[1] signal. If set to '1', PD_MGMT stays ON if PDCMONQREQn[1] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 2.	RW	0x0
[16]	S_PDCMONQREQ0	Enable sensitivity to PDCMONQREQn[0] signal. If set to '1', PD_MGMT stays ON if PDCMONQREQn[0] signal is HIGH. This bit is reserved and **RAZ/WI** if PDCMQCHWIDTH < 1.	RW	0x0
[15]	S_PD_MGMT_ON	Enable sensitivity to PD_MGMT power state. If set to '1' PD_MGMT stays ON if PD_MGMT is already ON.	RW	0x01
[14]	Reserved	Reserved.	**RAZ/WI**	0x0
[13]	S_PD_DEBUG_ON	Tied to HIGH. PD_MGMT stays ON if PD_DEBUG power domain is ON.	RO	0x01
[12]	Reserved	Reserved.	**RAZ/WI**	0x0
[11:9]	Reserved	Reserved.	RO	0x000
[8]	S_PD_NPU3_ON	Reserved.	**RAZ/WI**	0x01
[7]	S_PD_NPU2_ON	Reserved.	**RAZ/WI**	0x01
[6]	S_PD_NPU1_ON	Reserved.	**RAZ/WI**	0x01
[5]	S_PD_NPU0_ON	Tied to HIGH. PD_MGMT always tries to stay ON if PD_NPU0 is ON. This bit is reserved and **RAZ/WI** if NUMNPU < 1.	RO	0x01
[4]	S_PD_CPU3_ON	Reserved.	**RAZ/WI**	0x01
[3]	S_PD_CPU2_ON	Reserved.	**RAZ/WI**	0x01
[2]	S_PD_CPU1_ON	Reserved.	**RAZ/WI**	0x01
[1]	S_PD_CPU0_ON	Tied to HIGH. PD_MGMT always tries to stay ON if PD_CPU0 is ON.	RO	0x01
[0]	S_PD_SYS_ON	Tied to HIGH. PD_MGMT always tries to stay ON if PD_SYS is ON.	RO	0x01

4.7.2.18 LCM_DCU_FORCE_DISABLE, LCM Force Disable Register

The LCM DCUEN Force Disable Register can be used to override and disable the LCM debug control enable signals during boot. Bit assignment are defined in LCM Debug Control Unit.

The LCM_DCU_FORCE_DISABLE register is set to the inverted value of LCM_DCU_FORCE_DISABLE_INIT by hardware when the TP-Mode Lifecycle State is either in Test Chip Indication (TCI) or Virgin state. This mechanism enables the debugger with the use of nSRST to debug the complete boot flow in TCI mode. For more information on nSRST, see CPU Reset Handling.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Power domain

This register resides in the PD_AON power domain but can also reside in PD_MGMT power domain if its states are saved and restored when entering and then leaving the lower power state, respectively.

Offset

`0x25C`

Reset

This register is reset by nCOLDRESETAON.

Usage constraints

The LCM_DCU_FORCE_DISABLE register is Secure Privileged access only and supports 32-bit RW accesses. For write access to this register, only 32-bit writes are supported. Any byte and halfword writes are ignored.

Bit descriptions

The following table shows the register bit assignments.

Table 4-82: LCM_DCU_FORCE_DISABLE bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	FD	Overrides LCM DCUEN signals to zero (debug disabled). These bits are associated with LCM DCUEN bits. For details, see LCM Debug Control Unit. This field is set to the inverted value of LCM_DCU_FORCE_DISABLE_INIT by hardware when the TP-Mode Lifecycle State is either in Test Chip Indication (TCI) or Virgin state.	RW	LCM_DCU_FORCE_DISABLE_INIT

4.8 CPU Private Peripheral Bus region

As defined by the ARMv8-M architecture specification, the processor hosts a local Private Peripheral Bus (PPB) region at address `0xE000_0000` to `0xE00F_FFFF`. This region is for integration with the CoreSight debug and trace components that are normally local to CPU0 and is not intended for general peripheral usage.

In SSE-320, this region has the memory map as shown in the following table.


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®-M85 Processor Technical Reference Manual*.

Depending on EXPLOGIC_PRESENT:

- EXPLOGIC_PRESENT = 1, the Cortex-M85 TPIU-M and the MCU debug ROM table are integrated on the CPU0 EPPB interface in the subsystem expansion and the address range reserved for ETB is **RAZ/WI**. The MCU debug ROM table is pointing to the TPIU-M and the CPU's internal ROM table.

These are integrated at the following addresses:

- TPIU at `0xE004_0000`
  - ETB at `0xE004_5000`
  - MCU debug ROM table at {CPU0MCUROMADDR, `0x000`} to where the DAP-Lite2 is pointing. Default address for MCU debug ROM table is `0xE00F_E000`.
- EXPLOGIC_PRESENT = 0, the TPIU, ETB, and MCU debug ROM table are not integrated and these regions are provided as a PPB expansion.

Table 4-83: CPU0 Private Peripheral Bus Region

Row ID	Address	Size	Region Name	Description
1	`0xE004_0000 - 0xE004_0FFF`	4KB	SSE-320 TPIU/ PPB Expansion	SSE-320 TPIU when EXPLOGIC_PRESENT = 1 CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
2	`0xE004_1000 - 0xE004_1FFF`	4KB	ETM[1]	Embedded Trace Module
3	`0xE004_2000 - 0xE004_2FFF`	4KB	CTI[1]	Cross Trigger Interface
4	`0xE004_3000 - 0xE004_4FFF`	8KB	Reserved	Reserved
5	`0xE004_5000 - 0xE004_5FFF`	4KB	ETB/PPB Expansion	Reserved for ETB (**RAZ/WI**) when EXPLOGIC_PRESENT = 1 CPU0 Expansion PPB Interface when EXPLOGIC_PRESENT = 0
6	`0xE004_6000 - 0xE004_6FFF`	4KB	PMC[1]	Programmable MBIST Controller

Row ID	Address	Size	Region Name	Description
7	`0xE004_7000` - `0xE004_7FFF`	4KB	EWIC	External Wakeup Interrupt Controller. For more information, see EWIC.
8	`0xE004_8000` - `0xE004_8FFF`	4KB	PPB Expansion	CPU0 Expansion PPB Interface. Reserved for Software Based Build In Self Test
9	`0xE004_9000` - `0xE00F_EFFF`	24KB	MCU debug ROM table/PPB Expansion	CPU0 Expansion PPB Interface. Includes MCU debug ROM table at {CPU0MCUROMADDR, `0x000`} when EXPLOGIC_PRESENT = 1
10	`0xE00F_F000` - `0xE00F_FFFF`	4KB	ROM Table	CPU0 ROM Table

1. The existence of these components depends on the configuration and the supported features of the integrated processor. If they do not exist, these regions are reserved, and return an error response when accessed.

4.8.1 EWIC

SSE-320 is designed to always support an External Wakeup Interrupt Controller (EWIC) for CPU0. This allows the system to support CPU0 being switched off, and for the EWIC to run at a lower clock rate to help reduce PD_AON dynamic and leakage power consumption.

The EWIC resides in the PD_AON power domain, run on AONCLK, and resides in the nWARMRESETAON reset domain.

The EWIC is only accessible through the Private Peripheral Bus Region of CPU0 at address `0xE0047000` to `0xE0047FFF`.


Note

- EVENTS[0] input of the EWIC is not connected in SSE-320 but tied LOW. WFE wakeup events are not supported to wake-up SSE-320.
- When modifying the EWIC_ASCR.ASPU and EWIC_ACSR.ASPD, the software must issue a DSB and an ISB instruction before performing any other action.

For more information about Cortex-M85 EWIC registers, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

4.8.1.1 Cortex-M85 TPIU-M registers

For information about Cortex-M85 TPIU-M registers, see *Arm® Cortex®-M85 Processor Technical Reference Manual*.

4.9 Debug System Access region

SSE-320 supports the CoreSight SoC-600M based common debug infrastructure.

SSE-320 provides several Memory Access Ports (MEM-APs) to provide access to CPU0 and to the shared debug components in the system.

These ports are mapped to the Debug system region with a relative address map as shown in the following table.

Table 4-84: Debug system address map

Row ID	Address	Size	Region mame	Description
1	`0x0000 - 0x0FFF`	4KB	DSROM	Debug System ROM
2	`0x1000 - 0x1FFF`	4KB	Reserved	Reserved
3	`0x2000 - 0x3FFF`	8KB	SYS APB-AP	Shared Debug System Access Port
4	`0x4000 - 0x5FFF`	8KB	CPU0 AHB-AP	CPU0 Debug System Access Port
5	`0x6000 - 0xF_FFFF`	-	Reserved	Reserved

This region is accessible through:

- The Debug Access Interface with an address offset of `0x0000`. Access is first controlled using the Debug Authentication Access Control signal DAPDSSACCEN. When DAPDSSACCEN = 0, accesses from the Debug Access Interface are blocked and return error responses. If DAPDSSACCEN = 1, accesses are allowed to pass through.

  Note that a separate 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 allows debug components outside the subsystem, with DAPACCEN = 1, to still be accessible while DAPDSSACCEN = 0 stops accesses arriving at the subsystem Debug Access Interface from accessing the system.
- The Main and Peripheral Interconnect are at the following two aliased address offsets:
  - `0xE010_0000` is the Non-secure alias
  - `0xF010_0000` is the Secure alias

  Access from the interconnect is gated by the Debug Authentication Access Control signal, SYSDSSACCEN0. When SYSDSSACCEN0 = 0, accesses from CPU0 through the Main Interconnect are blocked and return error responses. If SYSDSSACCEN0 = 1, accesses from CPU0 are allowed to pass through.

  When access can pass the first gate controlled by SYSDSSACCEN0 into the Debug System, a PPC0 is then used to perform the final mapping of all APs either to the Non-secure region from `0xE010_0000` to `0xE01F_FFFF` or to the Secure region from `0xF010_0000` to `0xF01F_FFFF`. For more information, see PERIPHNSPPC0, PERIPHNSPPC1, PERIPHSPPPC0, PERIPHSPPPC1, PERIPHNSPPPC0, and PERIPHNSPPPC1.

Each Memory Access Port (MEM-AP) is a twin MEM-AP that enables an external debugger to use one logical MEM-AP, and on-chip software to use a separate logical MEM-AP. A twin MEM-

AP consists of two sets of 4K registers. The bottom 4K is for the external debugger accesses exclusively while the top 4K is for on-chip software accesses exclusively. It is **IMPLEMENTATION DEFINED** if the external debugger only has access to the top 4K and vice versa if the on-chip software only has access to the bottom 4K.

4.9.1 Shared debug system MEM-AP memory map

The Shared debug system MEM-AP provides access to the Shared debug system that all CPUs in the system shares. It includes a trace funnel for funneling all trace data together, an Embedded Trace Buffer (ETB) that allows trace data to be stored and read by a debugger. In addition, it provides access to debug components that reside in the expansion system through the Debug APB Expansion Interface.

The MEM-AP's base address static configuration is configured to point to Shared debug system CoreSight ROM Table at address `0x0000_0000`.

The following table shows the memory map that is visible to the Shared Debug System MEM-AP.

Table 4-85: Shared debug system MEM-AP memory map

Row ID	Address	Size	Region name	Description
1	`0x0000_0000 - 0x0000_0FFF`	4KB	SDSROM	Shared Debug System ROM Table
2	`0x0000_1000 - 0x0000_1FFF`	4KB	SDSFUNNEL	Shared Debug System Trace Funnel
3	`0x0000_2000 - 0x0000_2FFF`	4KB	SDSCTI	Shared Debug system Cross Trigger Interface
4	`0x0000_3000 - 0x0000_3FFF`	4KB	SDSETB	Shared Debug System Embedded Trace Buffer
5	`0x0000_4000 - 0x0000_FFFF`	48KB	Reserved	Reserved
6	`0x0001_0000 - 0x0001_3FFF`	16KB	Debug APB Expansion Interface	Debug APB Expansion Interface Region

4.9.2 CPU0 debug system MEM-AP memory map

The CPU0 Debug System Memory Access Port provides access to the following;

- CPU0 Debug CoreSight ROM Table, which the CPU0 MEM-AP's base address static configuration is configured to point to. This ROM Table includes Granular Power Requester (GPR) functionality to allow software to wake CPU0.
- CPU's Debug access interface, which provides access to the processor control and debug logic, and to a view of memory which is consistent with that observed by CPU0. This includes the CPU's PPB region, where other CPU0 private debug components can be hosted. For more information, see CPU Private Peripheral Bus region and *Arm® Cortex®-M85 Processor Technical Reference Manual*.

The following table shows the memory map of the memory map as seen by the CPU0 MEM-AP.

Table 4-86: CPU0 MEM-AP memory map

Row ID	Address	Size	Region name	Description
1	0x0000_0000 - 0xF000_7FFF	-	SYSACC	System Memory Access through CPU0 Debug Access Interface
2	0xF000_8000 - 0xF000_8FFF	4KB	CPU0ROM	CPU0 Debug ROM Table
3	0xF000_9000 - 0x0000_9FFF	4KB	Reserved	Reserved
4	0xF000_A000 - 0xFFFF_FFFF	-	SYSACC	System Memory Access through CPU0 Debug Access Interface

4.9.3 DSROM, Debug System ROM

The Debug System ROM is a CoreSight Class 0x1 ROM table that provides a list of pointers to Access Port within the Debug System.

The following table lists the contents of the Debug System ROM Table. For details of registers, see the Programmers model section of *Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0*.

Table 4-87: Debug System CoreSight ROM Table contents

Offset	Name	Type	Reset	Width	Description
0x000	ROMENTRY0	RO	0x0000_2001	32-bit	ROM entry pointing to the Shared Debug System MEM-AP.
0x004	ROMENTRY1	RO	0x0000_4001	32-bit	ROM entry pointing to Debug System CPU0 MEM-AP.
0x008	ROMENTRY2	RO	0x0000_6001	32-bit	ROM entry pointing to CPU1 Debug System MEM-AP. This register is reserved and **RAZ/WI** if NUMCPU < 1.
0x00C	ROMENTRY3	RO	0x0000_8001	32-bit	ROM entry pointing to CPU2 Debug System MEM-AP. This register is reserved and **RAZ/WI** if NUMCPU < 2.
0x010	ROMENTRY4	RO	0x0000_A001	32-bit	ROM entry pointing to CPU3 Debug System MEM-AP. This register is reserved and **RAZ/WI** if NUMCPU < 3.
0x014 – 0xFB4	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFB8	AUTHSTATUS	RO	0x0000_0000	32-bit	Authentication Status Register.
0xFBC	DEVARCH	RO	0x4770_0AF7	32-bit	Device Architecture Register.
0xFC0 – 0xFC4	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFC8	DEVID	RO	0x0000_0000	32-bit	Device Configuration Register.
0xFCC	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFD0	PIDR4	RO	0x0000_0004	32-bit	Peripheral ID 4
0xFD4 – 0xFDC	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFE0	PIDR0	RO	0x0000_004B	32-bit	Peripheral ID 0: PIDR0[7:0] - Part Number bits [7:0].

Offset	Name	Type	Reset	Width	Description
0xFE4	PIDR1	RO	0x0000_00B7	32-bit	Peripheral ID 1: PIDR1[3:0] - Part Number [11:8], PIDR1[7:4] - JEP106 Identity Code [3:0].
0xFE8	PIDR2	RO	0x0000_000B	32-bit	Peripheral ID 2: PIDR2[2:0] - JEP106 Identity Code [6:4], PIDR2[3] - JEDEC identifier, PIDR2[7:4] - Revision Code.
0xFEC	PIDR3	RO	0x0000_0000	32-bit	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	32-bit	Component ID 0
0xFF4	CIDR1	RO	0x0000_0090	32-bit	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	32-bit	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	32-bit	Component ID 3

4.9.4 SDSROM, Shared Debug System ROM

The Shared Debug System Debug CoreSight ROM is a CoreSight Class 0x1 ROM table that provides a list of pointers to the following:

- CoreSight components within the Shared Debug System level.
- An external CoreSight ROM through the Debug APB Expansion Interface.

The following table lists the contents of the Debug System CoreSight ROM. For details of registers, see the Programmers model section of *Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0*.

Table 4-88: Debug System ROM Table contents

Offset	Name	Type	Reset	Width	Description
0x000	ROMENTRY0	RO	0x0000_1001	32-bit	ROM entry pointing to the trace funnel.
0x004	ROMENTRY1	RO	0x0000_2001	32-bit	ROM entry pointing to the Cross Trigger Interface.
0x008	ROMENTRY2	RO	0x0000_3001	32-bit	ROM entry pointing to the Embedded Trace Buffer.
0x00C	ROMENTRY3	RO	0x0000_4001	32-bit	ROM entry pointing to an external CoreSight ROM table at address 0x0008_0000 through the Debug APB Expansion Interface.
0x010 – 0xFB4	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFB8	AUTHSTATUS	RO	0x0000_0000	32-bit	Authentication Status Register.
0xFBC	DEVARCH	RO	0x4770_0AF7	32-bit	Device Architecture Register.
0xFC0 – 0xFC4	Reserved	**RAZ/WI**	0x0000_0000	32-bit	Reserved
0xFC8	DEVID	RO	0x0000_0000	32-bit	Device Configuration Register.

Offset	Name	Type	Reset	Width	Description
0xFCC	Reserved	RAZ/WI	0x0000_0000	32-bit	Reserved
0xFD0	PIDR4	RO	0x0000_0004	32-bit	Peripheral ID 4
0xFD4 – 0xFDC	Reserved	RAZ/WI	0x0000_0000	32-bit	Reserved
0xFE0	PIDR0	RO	0x0000_004C	32-bit	Peripheral ID 0: PIDR0[7:0] - Part Number bits [7:0].
0xFE4	PIDR1	RO	0x0000_00B7	32-bit	Peripheral ID 1: PIDR1[3:0] - Part Number [11:8], PIDR1[7:4] - JEP106 Identity Code [3:0].
0xFE8	PIDR2	RO	0x0000_000B	32-bit	Peripheral ID 2: PIDR2[2:0] - JEP106 Identity Code [6:4], PIDR2[3] - JEDEC identifier, PIDR2[7:4] - Revision Code.
0xFEC	PIDR3	RO	0x0000_0000	32-bit	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	32-bit	Component ID 0
0xFF4	CIDR1	RO	0x0000_0090	32-bit	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	32-bit	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	32-bit	Component ID 3

4.9.5 CPU0ROM, CPU0 Debug System ROM

The CPU0 Debug System ROM is a CoreSight Class 0x9 ROM table that provides a list of pointers to the following:

- CoreSight ROM in the processor core.
- An optional CoreSight ROM, call the MCUROM residing on the EPPB. It also provides GPR functional to allow a debugger to request the processor to turn on.

The following table lists the contents of the CPU0 Debug ROM. For details of registers not linked to here, see the Programmers model section of *Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0*.

Table 4-89: CPU0 Debug System ROM Table contents

Offset	Name	Type	Reset	Width	Description
0x000	ROMENTRY0	RO	0xE00F_F007	32-bit	ROM entry pointing to the CPU's internal CoreSight ROM table and provides a POWERID value of 0x0.

Offset	Name	Type	Reset	Width	Description
0x004	ROMENTRY1	RO	CFG_DEF	32-bit	ROM entry pointing to the MCU ROM CoreSight Debug ROM table: ROMENTRY1[31:12] =CPU0 MCUROMADDR[31:12], ROMENTRY1[11:9] = 0x0, ROMENTRY1[8:4] = POWERID = 0x0, ROMENTRY[3] = 0b0, ROMENTRY[2] = 0b1, ROMENTRY[1:0] = {CPU0MCUROMVALID, CPU0MCU ROMVALID}.
0x010 – 0xAFC	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xA00	DBGPCR0	RW	0x0000_0001	32-bit	Debug Power Control Register 0. For requesting the CPU to turn ON.
0xA04 – 0xA7C	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xA80	DBGPSR0	RO	0x0000_0000	32-bit	Debug Power Status Register 0
0xA84 – 0xBFC	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xC00	PRIDR0	RO	0x0000_0001	32-bit	Power Request Identification Register
0xC04 – 0xFB4	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xFB8	AUTHSTATUS	RO	0x0000_0000	32-bit	Authentication Status Register.
0xFBC	DEVARCH	RO	0x4770_0AF7	32-bit	Device Architecture Register.
0xFC0 - 0xFC4	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xFC8	DEVID	RO	0x0000_0030	32-bit	Device ID Registers
0xFCC	DEVTYPE	RO	0x0000_0000	32-bit	DEVTYPE
0xFD0	PIDR4	RO	0x0000_0004	32-bit	Peripheral ID 4
0xFD4 – 0xFDC	Reserved	**RAZ/ WI**	0x0000_0000	32-bit	Reserved
0xFE0	PIDR0	RO	0x0000_004D	32-bit	Peripheral ID 0: PIDR0[7:0] - Part Number bits [7:0].
0xFE4	PIDR1	RO	0x0000_00B7	32-bit	Peripheral ID: PIDR1[3:0] - Part Number [11:8], PIDR1[7:4] - JEP106 Identity Code [3:0].
0xFE8	PIDR2	RO	0x0000_000B	32-bit	Peripheral ID 2: PIDR2[2:0] - JEP106 Identity Code [6:4], PIDR2[3] - JEDEC identifier, PIDR2[7:4] - Revision Code.

Offset	Name	Type	Reset	Width	Description
0xFEC	PIDR3	RO	0x0000_0000	32-bit	Peripheral ID 3
0xFF0	CIDR0	RO	0x0000_000D	32-bit	Component ID 0
0xFF4	CIDR1	RO	0x0000_000	32-bit	Component ID 1
0xFF8	CIDR2	RO	0x0000_0005	32-bit	Component ID 2
0xFFC	CIDR3	RO	0x0000_00B1	32-bit	Component ID 3

4.10 Peripheral Expansion region

When EXPLOGIC_PRESENT = 1, a system timestamp generator (System Counter) is integrated in SSE-320 expansion and drives the system timestamp interface.

For more details, see the System Timestamp Interface section of the *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*.

The System Counter resides in the PD_AON Power domain, is clocked by CNTCLK and is reset by nWARMRESETAON.

The following table shows the Peripheral Expansion Region address map when EXPLOGIC_PRESENT = 1

Table 4-90: Peripheral Expansion Region

Row ID	Address	Size	Region name	Description	Alias with row ID	Security[1]
-	0x4810_0000 - 0x4810_0FFF	4KB	Reserved	Reserved (**RAZ/WI**)	-	-
1	0x4810_1000 - 0x4810_1FFF	4KB	Peripheral expansion	System Counter (System Timestamp Generator) Status Frame register.	3	NS
2	0x5810_0000 - 0x5810_0FFF	4KB	Peripheral expansion	System Counter (System Timestamp Generator) Control Frame register.	-	S
3	0x5810_1000 - 0x5810_1FFF	4KB	Peripheral expansion	System Counter (System Timestamp Generator) Status Frame register.	1	S

[1] Legend

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

For details of the ARMv8M System Counter registers, see *Arm® Corstone™ Reference Systems Architecture Specification Ma2*.

Proprietary Notice

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

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

The validity, construction and performance of this notice shall be governed by English Law.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

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

Product revision status

The r0p0 identifier indicates the revision status of the product described in this manual, where:

r*x* Identifies the major revision of the product.
p*y* Identifies the minor revision or modification status of the product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0000-01	4 October 2024	Non-Confidential	Initial release

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in Document release information on page 205.

Table 2: Issue 0000-01

Change	Location
Initial issue of the document	-

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	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

We recommend the following. If you do not follow these recommendations your system might not work.

Your system requires the following. If you do not follow these requirements your system will not work.

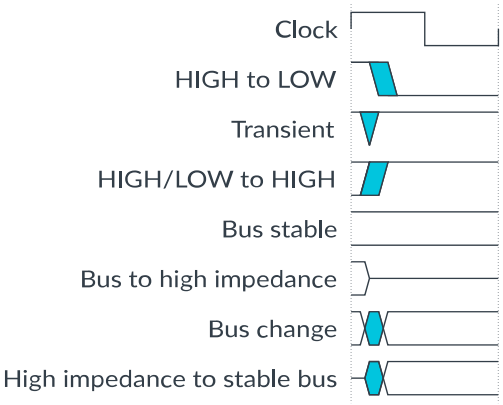
Danger	You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.
Note	This information is important and needs your attention.
Tip	This information might help you perform a task in an easier, better, or faster way.
Remember	This information reminds you of something important relating to the current content.

Timing diagrams

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

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

Figure 1: 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.

Useful resources

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

Access to Arm documents depends on their confidentiality:

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

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ DMA-350 Controller Configuration and Integration Manual	102483	Confidential
Arm® CoreLink™ DMA-350 Controller Technical Reference Manual	102482	Non-Confidential
Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual	101150	Non-Confidential
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	Non-Confidential
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual	101526	Non-Confidential
Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual	101130	Non-Confidential
Arm® Coresight™ Components Technical Reference Manual	DDI 0314	Non-Confidential
Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0	101883	Non-Confidential
Arm® Corstone™ SSE-320 Example Subsystem Reference Manual	109758	Confidential
Arm® Corstone™ SSE-320 Example Subsystem Software Programmers Guide	109759	Non-Confidential
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	Non-Confidential
Arm® Cortex®-M85 Processor Technical Reference Manual	101924	Non-Confidential
Arm® Cortex®-M85 Processor User Guide Reference Material	101927	Confidential
Arm® Ethos™-U85 NPU Technical Reference Manual	102685	Confidential
Arm® Mali™-C55 Image Signal Processor Configuration and Integration Manual	102565	Confidential
Arm® Mali™-C55 Image Signal Processor Register Map	102566	Confidential
Arm® Mali™-C55 Image Signal Processor Technical Reference Manual	102564	Confidential

Arm architecture and specifications	Document ID	Confidentiality
AMBA® AHB Protocol Specification	IHI 0033	Non-Confidential
AMBA® APB Protocol Specification	IHI 0024	Non-Confidential
AMBA® ATB Protocol Specification	IHI 0032	Non-Confidential
AMBA® AXI Protocol Specification	IHI 0022	Non-Confidential
AMBA® Low Power Interface Specification	IHI 0068	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® CoreSight™ Architecture Specification v3.0	IHI 0029	Non-Confidential
Arm® Corstone™ Reference Systems Architecture Specification Ma2	107610	Non-Confidential
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074	Non-Confidential
Arm® Key Management Unit Specification	107715	Non-Confidential
Arm® Lifecycle Manager Specification	107616	Non-Confidential
Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M	DEN 0083	Non-Confidential
Arm® Power Policy Unit Architecture Specification	DEN 0051	Non-Confidential
Arm® Security Alarm Manager Specification	107716	Non-Confidential
Arm®v8-M Architecture Reference Manual	DDI 0553	Non-Confidential