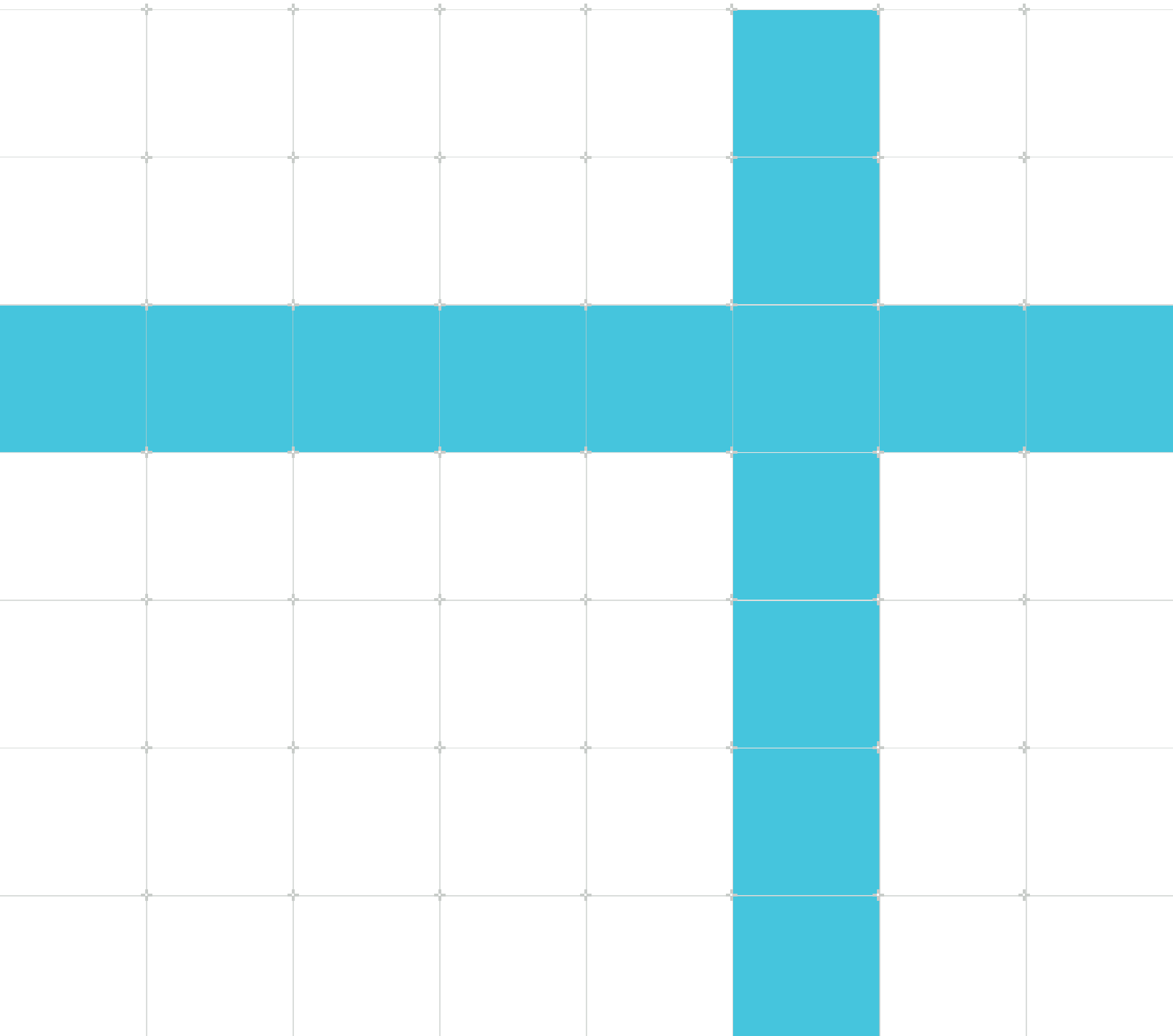
Arm® Corstone™ SSE-050 Subsystem

Revision: r0p1

Technical Reference Manual

Issue 01
100918_0001_01_en



Arm® Corstone™ SSE-050 Subsystem

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	15 June 2017	Non-Confidential	First release for r0p0
0001-00	8 December 2017	Non-Confidential	First release for r0p1
0001-01	29 July 2022	Non-Confidential	Second release for r0p1

Proprietary Notice

REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

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

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

Inclusive language commitment

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

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

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

Contents

1. Introduction

1.1 Product revision status

The r*x*p*y* identifier indicates the revision status of the product described in this manual, for example, r1p2, where:

r*x* Identifies the major revision of the product, for example, r1.

p*y* Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

This book is written for software engineers who want to work with an Arm reference platform. The manual describes the functionality of the IoT Subsystem.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Typographic conventions

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.

Convention	Use
<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**.

Recommendations. Not following these recommendations might lead to system failure or damage.

Requirements for the system. Not following these requirements might result in system failure or damage.

Requirements for the system. Not following these requirements will result in system failure or damage.

An important piece of information that needs your attention.

A useful tip that might make it easier, better or faster to perform a task.

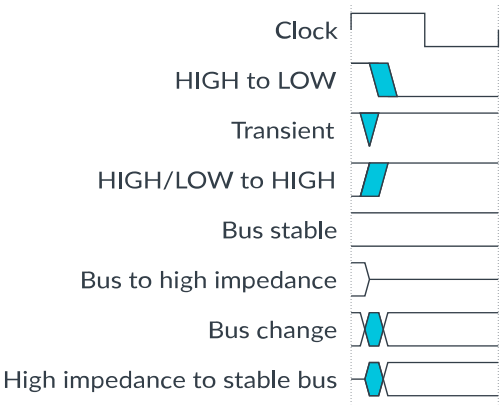
A reminder of something important that relates to the information you are reading.

Timing diagrams

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

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

1.4 Additional reading

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

Table 1-2: Arm publications

Document name	Document ID	Licensee only
Arm® AMBA® APB Protocol Specification, Version 2.0	IHI 0024	No
Arm® AMBA® 3 AHB-Lite Protocol Specification, Version 1.0	IHI 0033	No
Arm® CoreSight™ Components Technical Reference Manual	DDI 0314	No
Arm® CoreSight™ SoC-400 Technical Reference Manual	100536	No
Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual	100919	Yes
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® Cortex®-M3 Devices Generic User Guide	DUI 0552	No

Document name	Document ID	Licensee only
Arm® Cortex®-M3 Processor Technical Reference Manual	100165	No
Arm® Debug Interface Architecture Specification, ADIv5.0 to ADIv5.2	IHI 0031	No
Arm® Embedded Trace Macrocell Architecture Specification, ETMv1.0 to ETMv3.5	IHI 0014	No
Arm®v7-M Architecture Reference Manual	DDI 0403	No

Confidential documents are available to licensees only, through a product bundle.

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

2. Overview

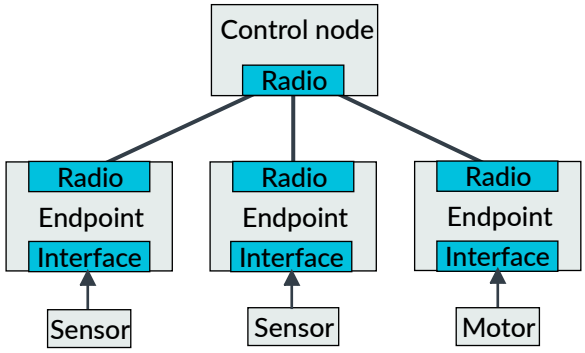
This chapter introduces the Arm® Corstone™ SSE-050.

2.1 About IoT endpoints

The SSE-050 delivers a process and technology agnostic reference that is preintegrated and validated, and a hardware and software subsystem that can be extended to provide an IoT endpoint system.

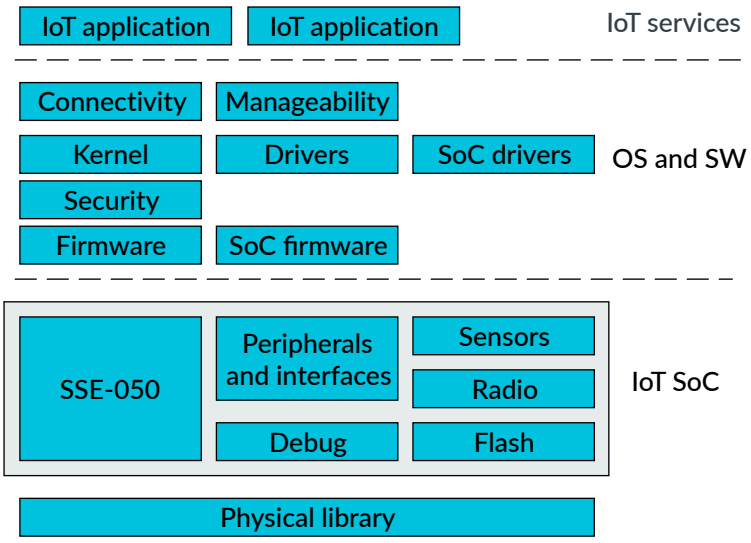
The following figure shows an IoT system consisting of several endpoints and a shared control node.

Figure 2-1: IoT endpoint HW and SW solution



The following figure shows a block diagram of the hardware and software in an endpoint solution.

Figure 2-2: IoT endpoint HW and SW solution



A complete endpoint system typically contains the following components:

Compute subsystem

The SSE-050 consists of the Corstone™ SSE-050 subsystem processor and associated bus, debug, expansion for persistent storage, and interface logic supplied by Arm.

AHB and APB interfaces are provided for connection to a flash memory controller, allowing an implementation that matches the target process to be integrated.

Reference system memory and peripherals

Additional memory, control, and peripheral components beyond the minimum SSE-050 components.

Licensees of the SSE-050 are provided with a testbench and example integration of some base peripherals. This example integration provides the starting point for customizing an SoC.

Communication interface

The endpoint can have some way of communicating with other nodes or masters in the system. This could be WiFi, Bluetooth, or a wired connection. Radio-specific interfaces such as clock, reset, and power control must be implemented at the SoC level.

Sensor or control component

To be useful as an endpoint, the reference design is typically extended by adding sensors or control logic, for example, temperature input or motor speed control output.

Software development environment

Arm provides a complete software development environment which includes Arm *Open Source Software* (OSS), Arm or GCC compilers and debuggers, and firmware.

Any custom peripherals typically require corresponding third-party firmware that can be integrated into the software stack.

2.2 Features

The SSE-050 contains the following components:

- A Corstone™ SSE-050 subsystem Cortex®-M3 processor:
  - Bit banding enables using standard instructions to read or modify of individual bits. The default implementation includes bit banding, but this can be configured during implementation.
  - Eight *Memory Protection Unit* (MPU) regions (optional)
  - *Nested Vectored Interrupt Controller* (NVIC) providing deterministic, high-performance interrupt handling with a configurable number of interrupts
  - *Wakeup Interrupt Controller* (WIC) with configurable number of WIC lines (optional). Optionally you can replace the standard Cortex-M3 WIC with a latch-based version. See the *Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual* for more information.
  - Little-endian memory addressing only for compatibility with typical eFlash controller and eFlash cache

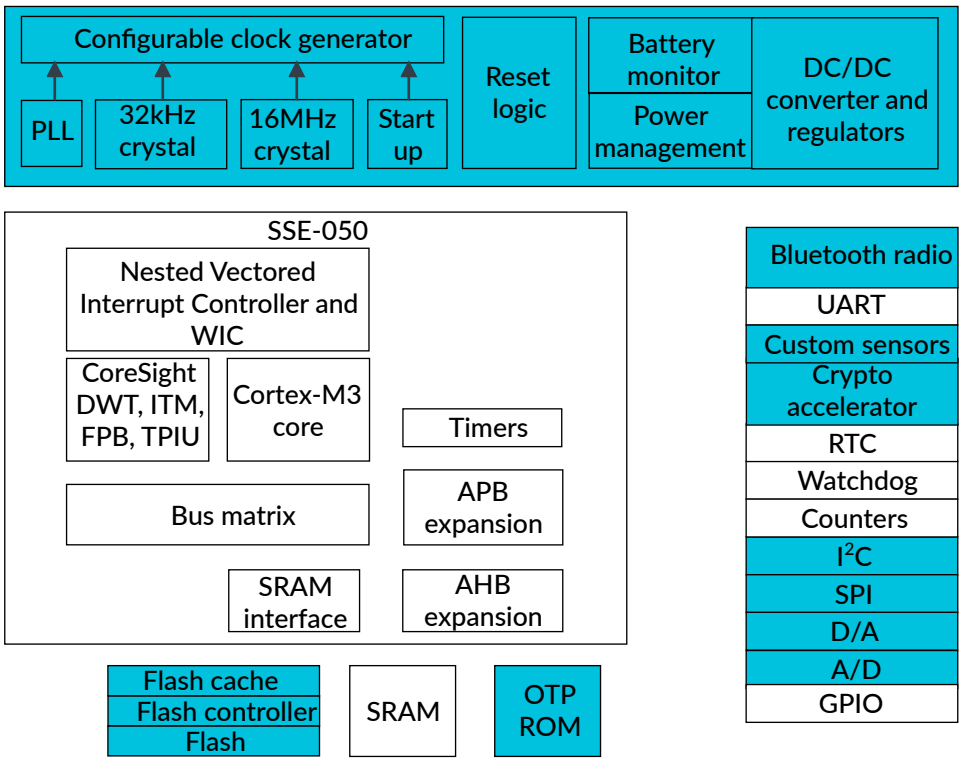
  For more information, see the *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- Integrated debug and trace:
  - Standalone system with a *Trace Port Interface Unit* (TPIU) and a *Serial Wire or JTAG Debug Port* (SWJ-DP)
  - Supports instruction trace using an *Embedded Trace Macrocell* (ETM) if licensed
- Multilayer AMBA® AHB-Lite interconnect:
  - Low-latency interconnect bus matrix
  - Two AHB-Lite slave expansion ports for external AHB masters
  - Two AHB-Lite master expansion ports for external AHB slaves
  - Eleven APB4 master expansion ports (each with 4KB address space) to connect APB peripherals
- Memory system, consisting of:
  - A placeholder for embedded flash controller and optionally cache. The following flash controllers are compatible:
    - GFC-100 Generic Flash Controller
    - GFC-200 Generic Flash Controller
    - Any third-party flash controller that can be integrated to an AHB memory interface and up to two APB control interfaces. The address map is configurable for two banks of 128KB or two banks of 256KB.
  - Static memory (configurable as one to four 32KB banks) is provided in the example integration layer
  - A placeholder for representing a flash-memory implementation in the integration layer

- Two APB timers:
  - Interrupt generation when the counter reaches 0
  - Each timer has an TIMERnEXTIN signal that can be used as an enable or external clock
  - Configurable privileged access mode
- Example integration for typical closely-coupled peripherals, using components from CMSDK:
  - Watchdog timer
  - UARTs
  - Application timers
  - *Real Time Clock* (RTC)
- Optional radio solution integration capability:
  - AHB master and slave ports
  - Reserved interrupt ports

A third-party Bluetooth solution can be connected to the AHB expansion ports. However, this requires customized software and firmware to support the product.

The reference system contains the peripherals that are required to support a *Real-Time Operating System* (RTOS), such as Arm® Keil® RTX. The components that are highlighted in the following figure are not provided by the SSE-050. Other peripherals not included in the SSE-050 might be required for specific application areas.

Figure 2-3: Example of an IoT endpoint SoC



2.3 Compliance

The SSE-050 complies with, or includes components that comply with, the following specifications:

- 2.3.1 Arm architecture on page 15.
- 2.3.2 Interrupt controller architecture on page 16.
- 2.3.3 Advanced Microcontroller Bus Architecture on page 16.

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

2.3.1 Arm architecture

The SSE-050 implements the Arm®v7-M architecture, which executes the Arm®v7M Thumb® instruction set.

See the *Arm®v7-M Architecture Reference Manual* for more information.

2.3.2 Interrupt controller architecture

The SSE-050 implements the Arm® *Nested Vectored Interrupt Controller* (NVIC).

See the *Arm® Cortex®-M3 Processor Technical Reference Manual* for more information.

2.3.3 Advanced Microcontroller Bus Architecture

The SSE-050 complies with the:

- *Advanced High Performance Bus* (AHB-Lite) protocol. See the *Arm® AMBA® 3 AHB-Lite Protocol Specification*.
- *Advanced Peripheral Bus* (APB) protocol. See the *Arm® AMBA® APB Protocol Specification, Version 2.0*.

2.4 Product revisions

This section describes the differences in functionality between product revisions:

r0p0	First release.
r0p0-r0p1	Values of revision and part number ID updated in system register.
r0p1-r0p1-01	Radio and stand-alone TRNG removed.

3. Functional description

This chapter describes the functionality of the SSE-050.

3.1 System top-level partitioning

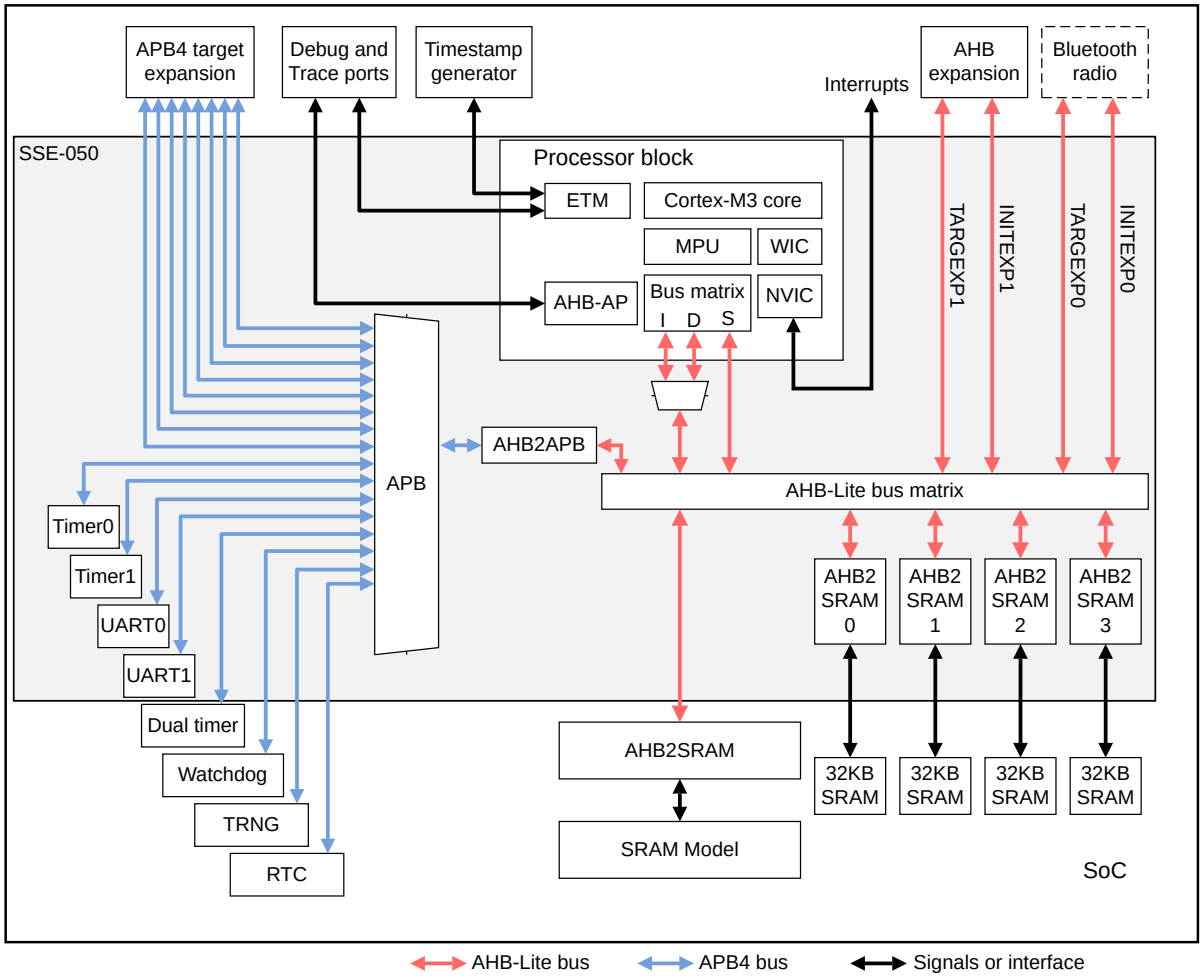
The SSE-050 consists of partitions of smaller sub-blocks. The SSE-050 is extended by additional components in the SoC integration layer.


Note

The provided example system is for information only and Arm expects system designers to customize it for their application requirements.

The following figure shows the top-level block diagram with the AHB-Lite and APB bus interconnections.

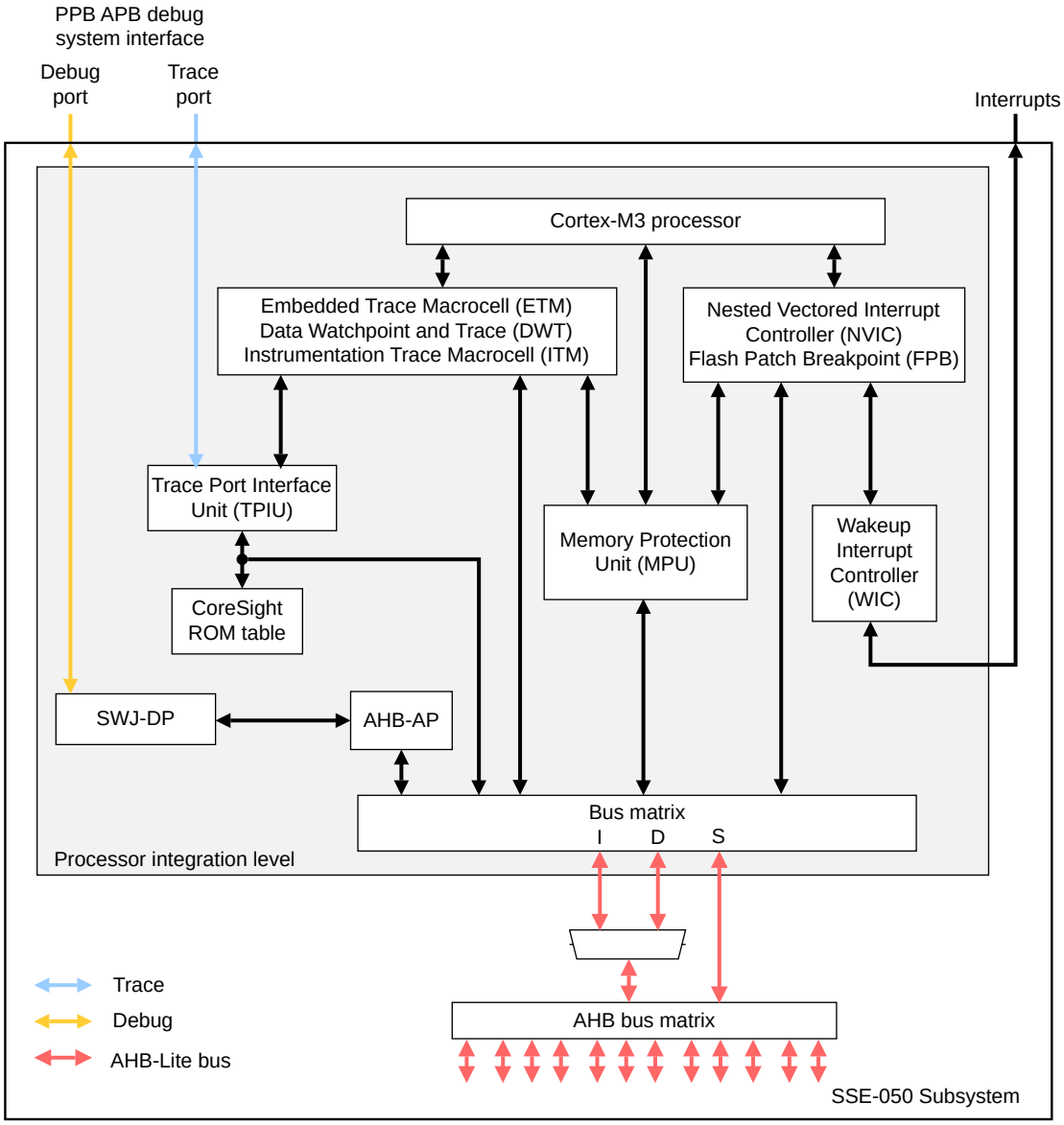
Figure 3-1: Bus interconnections



3.2 Cortex-M3 processor block

The following figure shows the block diagram for the Cortex®-M3 processor logic and interface.

Figure 3-2: Cortex®-M3 component



Note

The implementation integrates the TPIU and SWJ-DP from the Cortex®-M3 processor package in a similar way to the standard Cortex®-M3 integration level. For basic usage, there is no requirement to license the CoreSight™ SoC IP.

The system designer can however choose to design a system with the separately licensed CoreSight™ SoC and replace the integration level with the one provided by CoreSight™ SoC. This will require modification of the subsystem and is not supported as a validated configuration.

For more information on the Cortex®-M3 and the debug and trace logic, see the following documents:

- *Arm®v7-M Architecture Reference Manual*.
- *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- *Arm® CoreSight™ Components Technical Reference Manual*.
- *Arm® Debug Interface Architecture Specification, ADIv5.0 to ADIv5.2*.
- *Arm® Embedded Trace Macrocell Architecture Specification*.

3.3 Power management

Low-power operation is essential for most IoT endpoint devices that typically rely on a battery or on harvested energy.

The SSE-050 is a single power domain system and it does not support control of different power domains within the SoC. It does however define the necessary HW handshake signals that can be connected to a *Power Management Unit* (PMU) at the SoC level.

SRAM power management is not present in the SSE-050 because SRAM models are technology-dependent and are outside of the SSE-050 block. Memory power mode support can be implemented at SoC level.

Related information

CPU control, status, and power management signals on page 48

3.4 Clocks

The SSE-050 does not implement a clock control infrastructure. The SSE-050 is a single clock domain system that provides inputs for the clocks. Therefore all input clocks (except for the debug clocks) are identical in frequency and phase.

The target frequencies for the SSE-050 and associated components are:

- The typical configuration is 50MHz when using a 55nm process.
- Embedded flash might impose a minimum clock frequency.
- 20kHZ for the JTAG when using a 55nm process.
- 32kHZ low-power mode (optional). This clock can typically be integrated with an eFlash controller to reduce toggle rates.

3.4.1 Component clocks

The subsystem does not implement architectural clock gating other than the Cortex®-M3 and ETM internal gating. The SSE-050 is a single clock domain. The component clocks can however be gated by a custom implementation at SoC level.

For a full list of component clocks, see A.1 Clock and reset signals on page 41.

For a list of power-management related signals, see 3.3 Power management on page 20.

Related information

Clock and reset signals on page 41
Power management on page 20

3.5 Resets

The SSE-050 has no internal reset generation implemented, except that the Cortex®-M3 processor can be reset by the internal AIRCR.VECTRESET register bit of the NVIC.

All component resets in the SSE-050 are connected to the subsystem boundary and can therefore be reset using reset input signals.

The SSE-050 is not designed to handle arbitrary reset patterns. The SoC integration and software must ensure that all resets are cleanly released before functional operation and no software reset is triggered to functioning components.

All resets are active low and may be asserted asynchronously. External reset synchronization is required to guarantee the clean deassertion of the resets in synchronization with the corresponding clocks.

For a full list of component reset signals, see A.1 Clock and reset signals on page 41.

3.5.1 About boot after reset

There is one Cortex®-M3 processor that is integrated into the SSE-050. After a processor reset deassertion, the Cortex®-M3 processor starts fetching the addresses on the instruction AHB as follows:

1. `0x00000000`: Fetch the stack pointer to initialize the SP register.
2. `0x00000004`: Fetch the reset vector and jump to the reset vector value.
3. *Reset vector*: Start boot code execution.

Address `0x00000000` of the SSE-050 is mapped to the flash AHB master statically. It is therefore not possible to directly boot from ROM attached to the AHB expansion port.

When eFlash is integrated on a final device the initial reset vector, SP and boot code can be written using the debugger. Alternatively the eFlash can be pre-loaded using a DFT interface as part of the production process.

3.5.2 Events

The following table lists events that can be used for multiprocessor systems.

Table 3-1: Cortex®-M3 events

Name	Description
CPU0RXEV	RX event input of the Cortex®-M3. Causes a wakeup from a `WFE` instruction. Connect to TXEVs from other processors in a multiprocessor system. Input from ORing TXEV signals from other processors in the system. If different processors run at different frequencies then synchronizers must be used to guarantee that TXEV is synchronous to this processor. TXEV must also be a single-cycle pulse. Tie LOW if not used.
CPU0TXEV	TX event output of the Cortex®-M3. Event transmitted as a result of `SEV` instruction. This is a single-cycle pulse. You can use it to implement a more power efficient spin-lock in a multiprocessor system. In a multiprocessor system, TXEV from each processor can be broadcast to the RXEV input of the other processors.

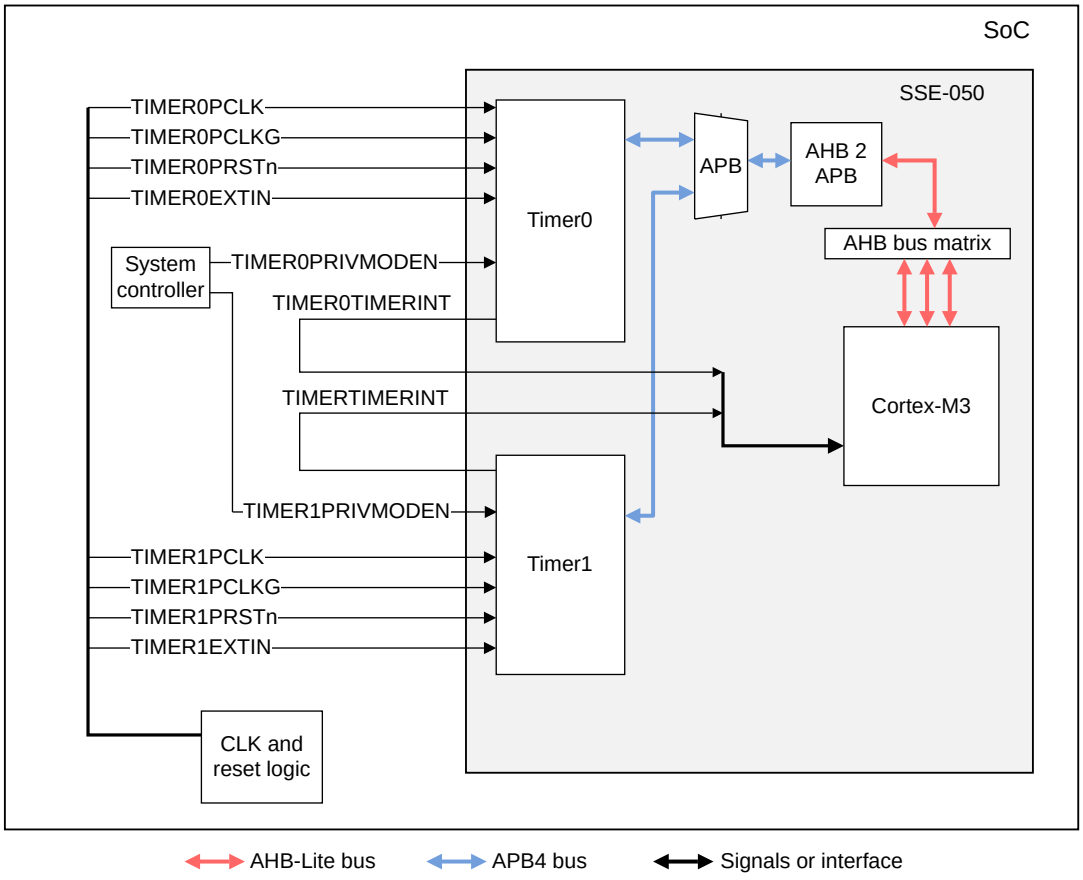

Note

Where a design contains a single Cortex®-M3 processor, the system designer can choose to connect the RX event port to DMA done signals, or another hardware accelerator unit.

3.6 Timer

The SSE-050 includes two instances of APB timers. These are required to satisfy the Mbed™ OS requirements.

Figure 3-3: Timer interfaces



See also A.4 Timer signals on page 43.

Related information

Timer signals on page 43

3.6.1 Security Extension

Privilege mode enable signals determine whether only privileged accesses or both privileged and non-privileged accesses can write to the timer registers.

Table 3-2: Privilege mode enable input signals

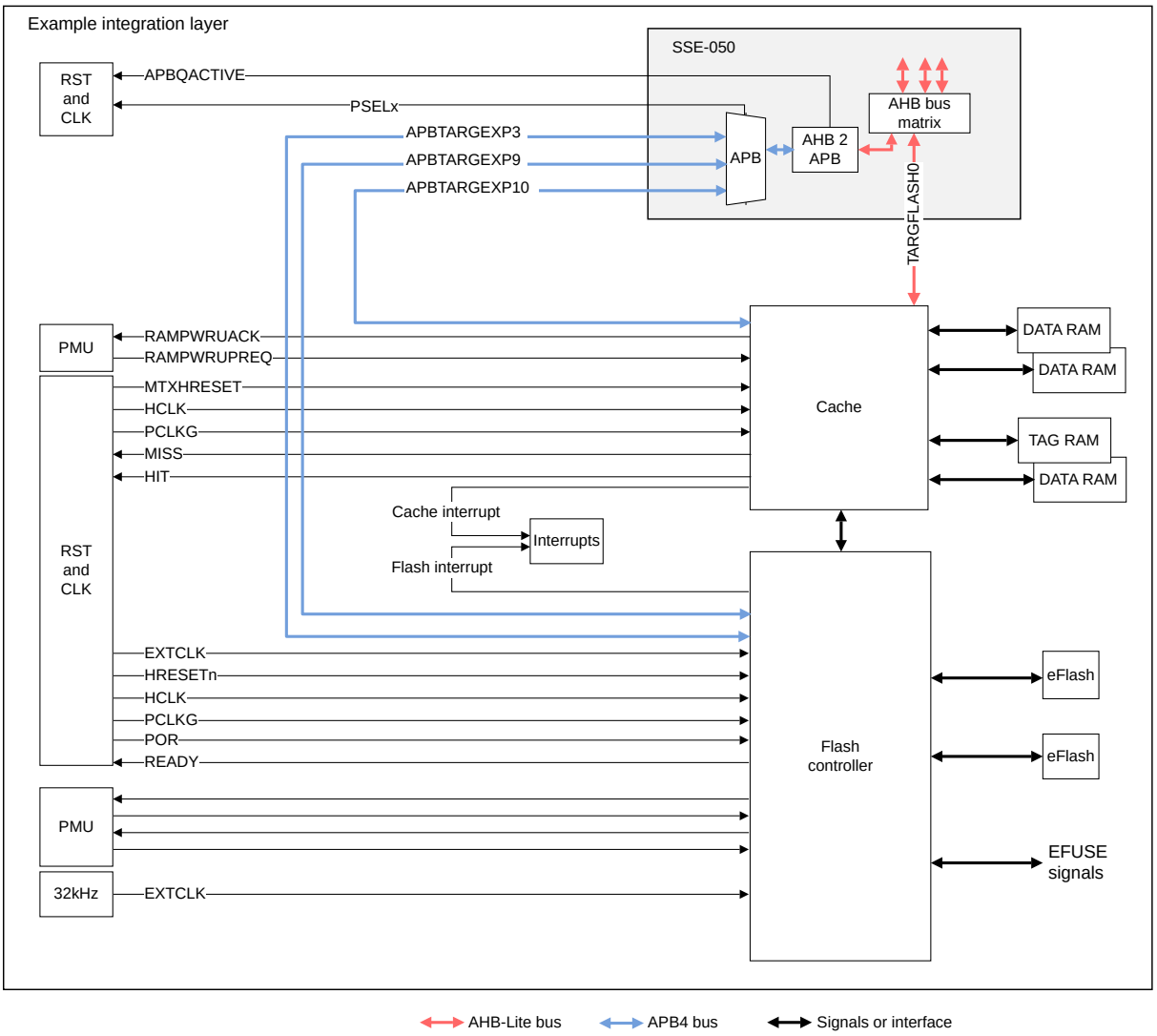
Signal	Clock	Description
TIMER0PRIVMODEN	TMER0PCLK	Defines if the timer memory mapped registers are writeable only by privileged access: • 0: Non-privileged access can write registers. • 1: Only privileged access can write registers. You must set this for Mbed™ compatibility.
TIMER1PRIVMODEN	TIMER1PCLK	Defines if the timer memory mapped registers are writeable only by privileged access: • 0: Non-privileged access can write registers. • 1: Only privileged access can write registers. You must set this for Mbed™ compatibility.

3.7 eFlash memory subsystem

In the example system that is delivered, cache, and flash controller are replaced by a simplified SRAM interface and model. These connections should be made in the level above the SSE-050.

The following figure shows the typical connections that you require for a full embedded flash and cache implementation.

Figure 3-4: Example flash system



3.7.1 Interfaces for flash memory integration

The SSE-050 includes an AHB master, two APB slaves, and interrupts for use with flash memory. These are the default, recommended connections. Any alternative implementation for code memory must take into account the Cortex®-M3 processor boot process, which always starts by fetching from address `0x00000000`.

AHB master port

An AHB master port permits an external flash memory subsystem to be connected to the subsystem. The signals of this port are prefixed with TARGFLASH0. This port maps to a 256KB or 512KB memory region, depending on the configuration.

APB master ports

Three APB master ports are reserved for connecting an external flash memory system to the subsystem. The signals of these ports are prefixed APBTARGEXP<3, 9, or 10>. Port APBTARGEXP3 should be used to control a cache if it is present in the memory system. Ports APBTARGEXP9 and APBTARGEXP10 should be used to access the flash which is normally arranged in two banks to permit over-the-air code download to a device.

Interrupt connections

There are no dedicated interrupt connections for the memory subsystem. As a result, all of the interrupt signals from the processor are exposed at the subsystem boundary. CPU0INTISR14 and CPU0INTISR15 are recommended for use by the memory subsystem. For more details, see 4.3.1 Interrupt signals on page 36 for the suggested interrupt connections.

3.8 Banked SRAM subsystem

The SSE-050 infrastructure supports up to four 32KB SRAMs. At least one 32KB SRAM must be implemented (SRAM bank 0).

If a SRAM bank is not implemented and the corresponding MTXREMAP bits are 1, then the corresponding address space is mapped to the AHB Slave expansion port of the interconnect.

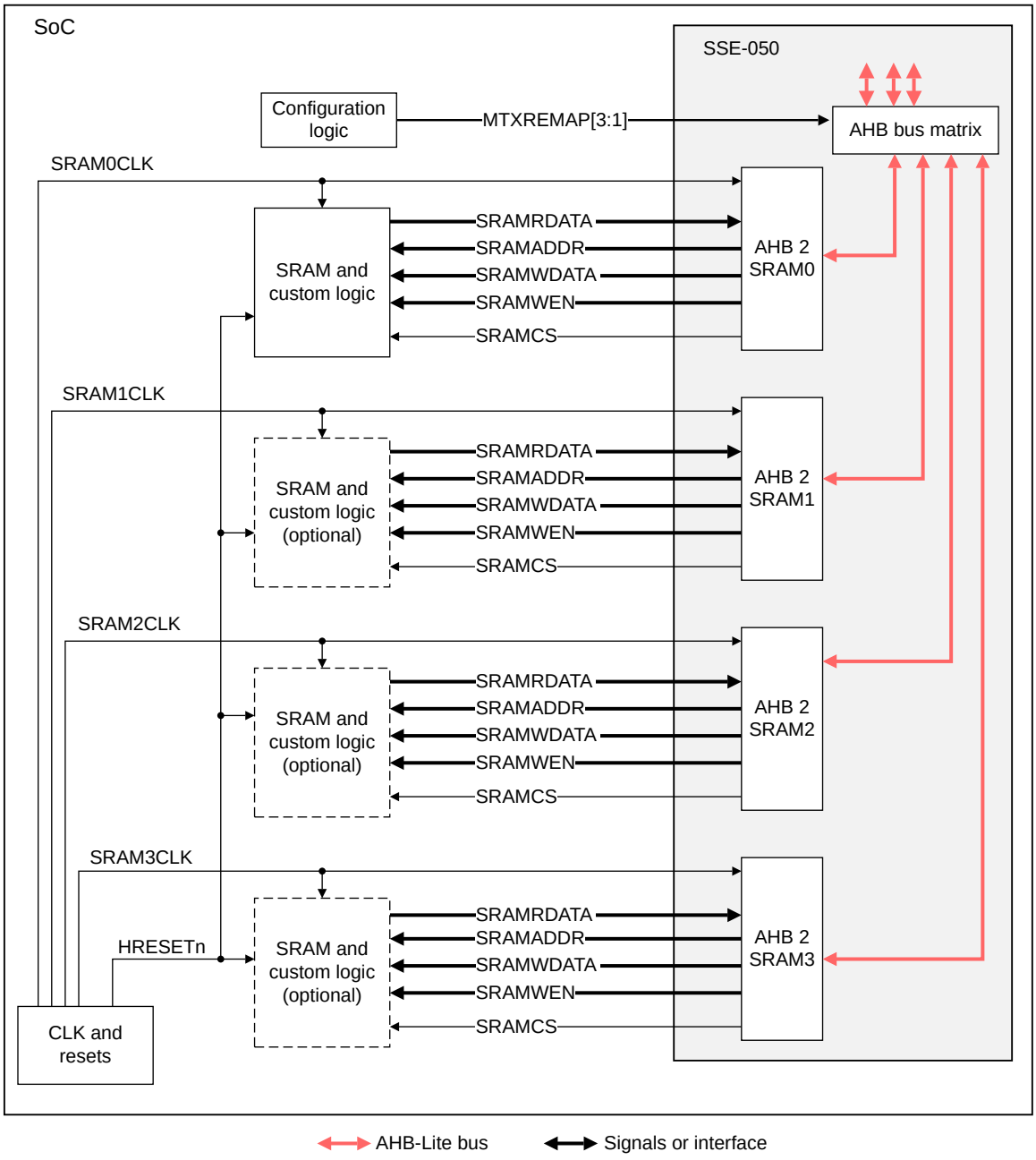
In the following figure, the MTXREMAP signals from the configuration logic must be static during functional operation.

The SRAM modules are outside of the SSE-050. The subsystem does not implement retention or powerdown supports for the SRAMs. It is the responsibility of the SoC integrator to implement power domains for the SRAM, and control the power modes of the SRAM banks with a SoC level PMU.

The AHB2SRAM bridge always responds with OKAY to all AHB accesses, even if the connected SRAM is not functional because for example it is powered-down or in retention mode.

It is the responsibility of the software to not read or write the SRAMs when they are not functional. If the software tries to access non-accessible SRAM, the AHB2SRAM bridge implementation ensures that system does not go into a deadlock state because of a non-responsive SRAM. The read data result is implementation-specific when SRAM is not powered.

Figure 3-5: SRAM interface



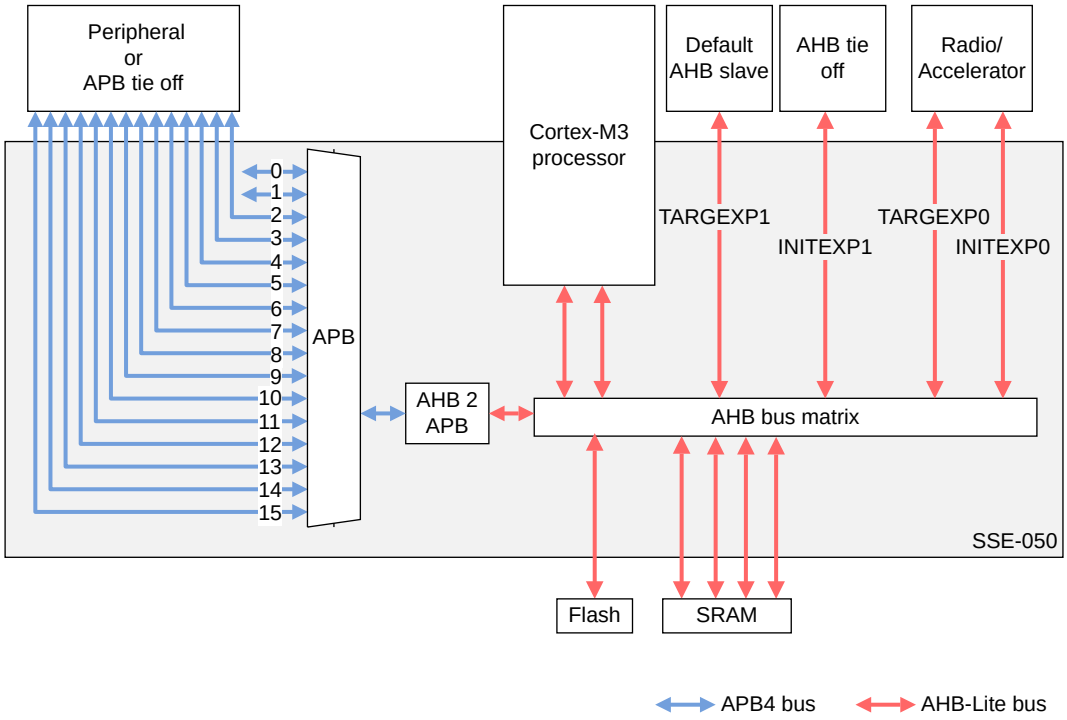
Related information

SRAM signals on page 42

3.9 AHB and APB expansion

The following figure shows the AHB and APB bus structure.

Figure 3-6: AHB and APB expansion buses



Any un-used AHB and APB buses must be tied off to the appropriate values. Synthesis flows will typically optimize away any redundant logic that remains.

See also A.5 Bus signals on page 43.

Related information

Bus signals on page 43

3.9.1 APB slave multiplexer

The APB slave multiplexer supports 16 APB slaves in the SSE-050. All ports are synchronous to the AHB expansion ports. Two APB ports are used for internal peripherals, the remaining APB ports are available to the subsystem boundary and can be connected to external peripherals.

The following table shows the expected usage of the APB ports in the SSE-050. Only the internal connections are made directly in the subsystem, the external connections are recommended, and match the example integration provided.

Table 3-3: APB ports

APB port	Connection	
	Peripheral	Location
PORT0	TIMER0	Internal
PORT1	TIMER1	Internal
PORT2	Dual timer	APBTARGEXP2
PORT3	Flash Cache if present	APBTARGEXP3
PORT4	UART0	APBTARGEXP4
PORT5	UART1	APBTARGEXP5
PORT6	RTC	APBTARGEXP6
PORT7	Spare	APBTARGEXP7
PORT8	Watchdog	APBTARGEXP8
PORT9	External Flash Controller if present	APBTARGEXP9
PORT10	Spare	APBTARGEXP10
PORT11	Spare	APBTARGEXP11
PORT12	Spare	APBTARGEXP12
PORT13	Spare	APBTARGEXP13
PORT14	Spare	APBTARGEXP14
PORT15	TRNG	APBTARGEXP15

If an APB expansion interface is not used to connect a peripheral, the port must be tied off properly at the SoC integration level.

3.9.2 AHB expansion

This section describes the AHB expansion features of the SSE-050.

3.9.2.1 AHB slave ports

Two AHB slave ports permit external AHB masters to be connected to the subsystem. These ports are prefixed with INITEXP<0..1>:

- INITEXP0 port is suitable for the AHB DMA master port of a communication interface.
- INITEXP1 port can be used to connect any additional AHB master to the system.


Note

If one of the slave ports is not used, then it must be tied off properly at the SoC integration level.

3.9.2.2 AHB master ports

Two AHB master ports permit external AHB masters to be connected to the subsystem. These ports are prefixed with TARGETEXP<0..1>:

- TARGETEXP0 port is suitable for the AHB slave port of a communication interface.
- TARGETEXP1 port can be used to connect any additional AHB slave to the system. Arm recommends that a secondary AHB interconnect is connected here for peripherals which are not latency critical.

If one of the master ports is not used, then the default slave must be connected to the port.

3.10 Debug and trace

In the Cortex®-M3 integration level, the SWJ-DP is a combined JTAG-DP and SW-DP that enables you to connect either an SWD or JTAG probe to a target. It is the standard CoreSight™ debug port.

To make efficient use of package pins, the JTAG pins use an auto-detect mechanism that switches between JTAG-DP and SW-DP depending on which probe is connected.

The Cortex®-M3 TPIU is an optional component that acts as a bridge between the on-chip trace data from the *Embedded Trace Macrocell* (ETM) and the *Instrumentation Trace Macrocell* (ITM), with separate IDs, to a data stream. The TPIU encapsulates IDs where required, and then the data stream is captured by a *Trace Port Analyzer* (TPA). The Cortex®-M3 TPIU is specially designed for low-cost debug, and should not be used in any different trace subsystem topologies.

The default implementation reuses the TPIU and SWJ-DP from the Cortex®-M3 package. This configuration is sufficient for single processor use, where an on-chip trace buffer is not required.

For more sophisticated multiprocessor debug solution, a full CoreSight™ SoC IP solution can be licensed and implemented. To implement a CoreSight™ derived system, the Cortex®-M3 integration level within the subsystem must be replaced with the version provided by CoreSight™ SoC.

A timestamp source is included in the example integration. This can be useful to correlate trace information between an ETM and an ITM, but is not a requirement for trace to function.

Related information

Debug and Trace signals on page 45

4. Programmers model

This chapter describes the SSE-050 Subsystem memory regions and registers, and provides information on how to program a SoC that contains an implementation of the subsystem.

4.1 About this programmers model

The following information applies to all registers:

- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in **UNPREDICTABLE** behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify register bits that are not defined.
  - On reads, ignore register bits that are not defined.
- The following describes the access type:

RW	Read and write.
RO	Read-only.
WO	Write-only.

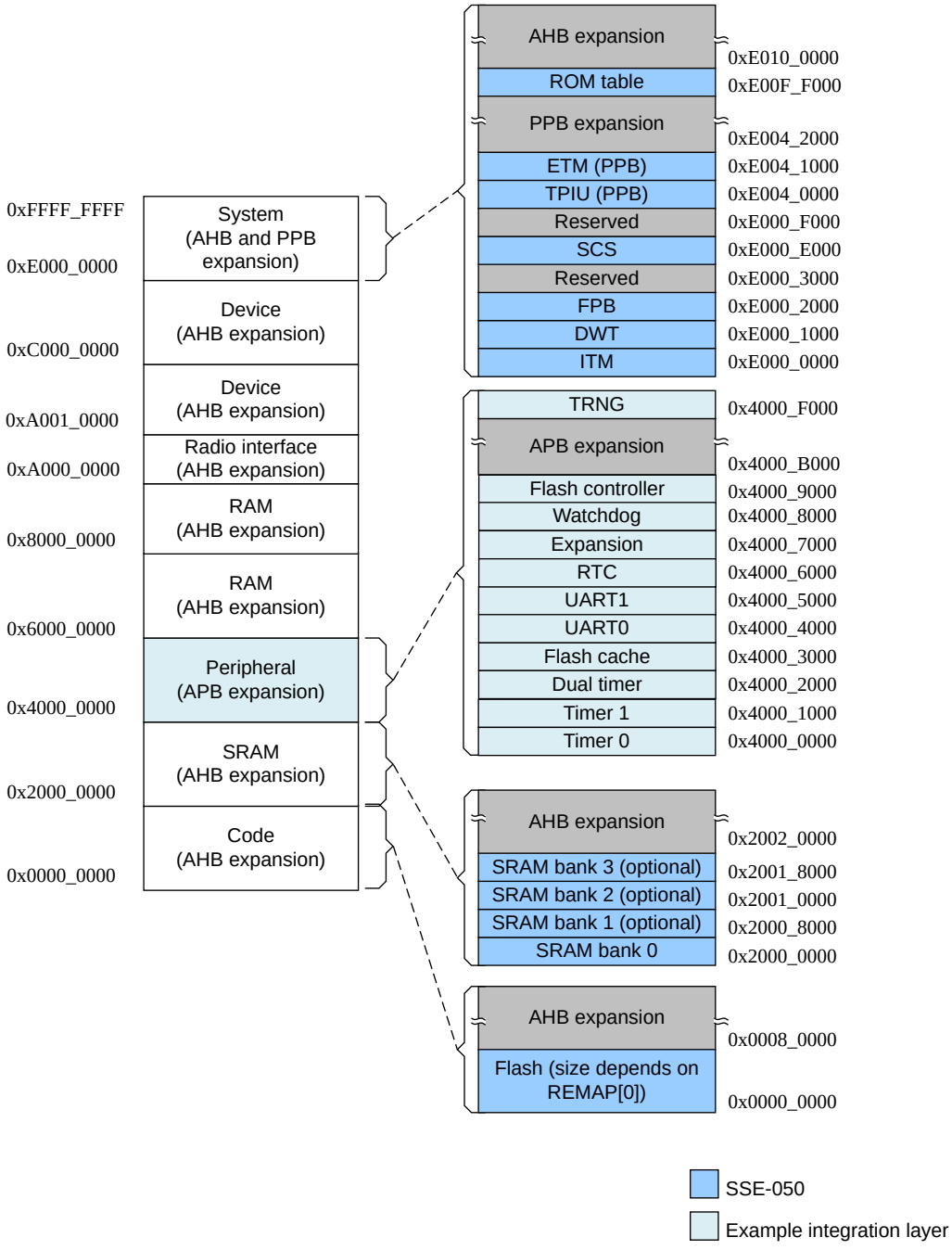

Note

The example peripherals that connect in the example integration are not documented here. For more information about the peripherals that are used in example integration, see the *Arm® Cortex®-M System Design Kit Technical Reference Manual*.

4.2 Memory map

The following figure shows the memory map for the SSE-050 and the example integration level.

Figure 4-1: Top-level memory map



4.2.1 Remap

REMAP[0-3] are the SSE-050 static configuration pins. Depending on the configured implementation, the memory map takes one of several compatible layouts.

Target software has to be aware of the specific memory map in use. The remapping feature of the AHB interconnect provides the following remapping options:

REMAP[0]

The Embedded eFlash memory region represents the maximum supported eFlash size: 512KB. If the size of the actual eFlash banks is 256KB, the upper 256KB can be remapped to the AHB expansion port as follows:

- 0: 512KB eFlash.
- 1: 256KB eFlash upper 256Kbytes mapped to AHB expansion port TARGETEXP1.

REMAP[1]

If SRAM1 is not present, the corresponding memory range can be mapped to AHB expansion port as follows:

- 0: SRAM1 present
- 1: SRAM1 not available, mapped to AHB expansion port TARGETEXP1

REMAP[2]

If SRAM2 is not present, the corresponding memory range can be mapped to AHB expansion port as follows:

- 0: SRAM2 present
- 1: SRAM2 not available, mapped to AHB expansion port TARGETEXP1

REMAP[3]

If SRAM3 is not present, the corresponding memory range can be mapped to AHB expansion port as follows:

- 0: SRAM3 present
- 1: SRAM3 not available, mapped to AHB expansion port TARGETEXP1

The PPB address region of the Cortex®-M3 memory map is not accessible from the INITEXP0 and INITEXP1 ports. Any access to this region returns a SLVERR response.

The following table shows the memory map for the SSE-050 Code and RAM regions.

Table 4-1: Code and SRAM regions

Region	Start	End	Peripheral	Size	AHB bus matrix	Remap				Comment
	Address	Address	name		port	3	2	1	0	
Code	0x00000000	0x0003FFFF	Code memory	256KB	TARG_FLASH0	-	-	-	-	-
	0x00040000	0x0007FFFF	Code memory	256KB	TARG_FLASH0	-	-	-	0	-
			AHB Expansion	256KB	TARG_EXP1	-	-	-	1	-

Region	Start	End	Peripheral	Size	AHB bus matrix	Remap				Comment
	Address	Address	name		port	3	2	1	0	
	0x00080000	0x1FFFFFFF	AHB Expansion	-	TARG_EXP1	-	-	-	-	-
SRAM	0x20000000	0x20007FFF	SRAM0	32KB	TARG_SRAM0	-	-	-	-	Bit band region
	0x20008000	0x2000FFFF	SRAM1	32KB	TARG_SRAM1	-	-	0	-	Bit band region
			AHB Expansion	32KB	TARG_EXP1	-	-	1	-	Bit band region
	0x20010000	0x20017FFF	SRAM2	32KB	TARG_SRAM2	-	0	-	-	Bit band region
			AHB Expansion	32KB	TARG_EXP1	-	1	-	-	Bit band region
	0x20018000	0x2001FFFF	SRAM	32KB	TARG_SRAM3	0	-	-		Bit band region
			AHB Expansion	32KB	TARG_EXP1	1	-	-		Bit band region
	0x20020000	0x21FFFFFF	AHB Expansion	32MB	TARG_EXP1	-	-	-	-	Bit band region 0x20020000 to 0x200FFFFF
	0x22000000	0x23FFFFFF	AHB Expansion	32MB	TARG_EXP1	-	-	-	-	Bit band alias
	0x24000000	0x3FFFFFFF	AHB Expansion	-	TARG_EXP1	-	-	-	-	-

Each bit in the 1MB bit band region can be accessed individually by making an access to the corresponding word in the 32MB bit band alias region. Only bit [0] is used when you make an access to the bit band alias region.

The base bit band region can be accessed directly in the same way as normal memory, using word, halfword, or byte accesses.

4.2.2 Peripheral, expansion, and system regions

The following table shows the memory map for the SSE-050 peripherals, expansion, and system regions.

Each bit in the 1MB bit band region can be accessed individually by making an access to the corresponding word in the 32MB bit band alias region. Only bit [0] is used when you make an access to the bit band alias region. The base bit band region can be accessed directly in the same way as normal memory, using word, halfword, or byte accesses.

Table 4-2: Expansion and system map

Type	Start	End	Peripheral	Size	AHB bus matrix	Bus fabric	Comment
Peripheral	0x40000000	0x40000FFF	Timer0	4KB	TARG_APB0	APB port 0	-
Peripheral	0x40001000	0x40001FFF	Timer1	4KB	TARG_APB0	APB port 1	-

Type	Start	End	Peripheral	Size	AHB bus matrix	Bus fabric	Comment
Peripheral	0x40002000	0x40002FFF	APB expansion	4KB	TARG_APB0	APB port 2	Use for dual timer
Peripheral	0x40003000	0x40003FFF	APB Expansion	4KB	TARG_APB0	APB port 3	-
Peripheral	0x40004000	0x40004FFF	APB expansion	4KB	TARG_APB0	APB port 4	Use for UART0
Peripheral	0x40005000	0x40005FFF	APB expansion	4KB	TARG_APB0	APB port 5	Use for UART1
Peripheral	0x40006000	0x40006FFF	APB expansion	4KB	TARG_APB0	APB port 6	Use for RTC
Peripheral	0x40007000	0x40007FFF	APB expansion	4KB	TARG_APB0	APB port 7	-
Peripheral	0x40008000	0x40008FFF	APB expansion	4KB	TARG_APB0	APB port 8	Use for watchdog
Peripheral	0x40009000	0x40009FFF	APB expansion	4KB	TARG_APB0	APB port 9	Use for flash bank 0
Peripheral	0x4000A000	0x4000AFFF	APB Expansion	2KB	TARG_APB0	APB port 10	Use for flash bank 1
Peripheral	0x4000B000	0x4000BFFF	APB expansion	4KB	TARG_APB0	APB port 11	-
Peripheral	0x4000C000	0x4000CFFF	APB expansion	4KB	TARG_APB0	APB port 12	-
Peripheral	0x4000D000	0x4000EFFF	APB expansion	4KB	TARG_APB0	APB port 13	-
Peripheral	0x4000E000	0x4000EFFF	APB expansion	4KB	TARG_APB0	APB port 14	-
Peripheral	0x4000F000	0x4000FFFF	APB expansion	4KB	TARG_APB0	APB port 15	Use for TRNG
Peripheral	0x40010000	0x400FFFFF	AHB expansion	1MB	TARG_EXP1	-	Bit band region
Peripheral	0x40100000	0x41FFFFFF	AHB expansion	32MB	TARG_EXP1	-	-
Peripheral	0x42000000	0x43FFFFFF	AHB expansion	32MB	TARG_EXP1	-	Bit band alias
RAM	0x60000000	0x7FFFFFFF	AHB Expansion	-	TARG_EXP1	-	-
RAM	0x80000000	0x9FFFFFFF	AHB Expansion	-	TARG_EXP1	-	-
Device	0xA0000000	0xA000FFFF	AHB Expansion	64KB	TARG_EXP0	-	Use for Radio ICCC
Device	0xA0010000	0xBFFFFFFF	AHB Expansion	-	TARG_EXP1	-	-
Device	0xC0000000	0xDFFFFFFF	AHB Expansion	-	TARG_EXP1	-	-
System	0xE0000000	0xE0000FFF	ITM (or reserved)	4KB	Default slave	PPB-Internal	Implementation defined
	0xE0001000	0xE0001FFF	DWT (or reserved)	4KB	Default slave	PPB-Internal	Implementation defined
	0xE0002000	0xE0002FFF	FPB (or reserved)	4KB	Default slave	PPB-Internal	Implementation defined
	0xE0003000	0xE000DFFF	Reserved	-	Default slave	PPB-Internal	-
	0xE000E000	0xE000EFFF	SCS (or reserved)	4KB	Default slave	PPB-Internal	Implementation defined
	0xE000F000	0xE003FFFF	Reserved	-	Default slave	PPB-Internal	-
	0xE0040000	0xE0040FFF	TPIU (or reserved)	4KB	Default slave	PPB-External	Implementation defined
	0xE0041000	0xE0041FFF	ETM (or reserved)	4KB	Default slave	PPB-External	Implementation defined
	0xE0042000	0xE00FEFFF	Reserved	-	Default slave	PPB-External	-
	0xE00FF000	0xE00FFFFF	ROM table	4KB	Default slave	PPB-External	-
	0xE0100000	0xFFFFFFFF	AHB expansion	-	TARG_EXP1	-	-

4.3 Interrupts

This section describes the (*Nested Vectored Interrupt Controller*) NVIC and the interrupt signal map.

The NVIC supports:

- An implementation-defined number of interrupts, in the range 4-240 interrupts.
- A programmable priority level of 0-255 for each interrupt. A higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and subpriority fields.
- Interrupt tail-chaining.
- An external *Non-Maskable Interrupt* (NMI).
- Optional WIC, providing ultra-low power sleep mode support.

4.3.1 Interrupt signals

This section describes interrupts and exceptions that the Cortex®-M3 processor handles in the SSE-050.

Table 4-3: Exceptions

No.	Exception type	Priority	Description
1	Reset	-3	Reset is invoked on powerup or a Warm reset.
2	NMI	-2	CPU0INTNMI port input. Arm recommends that a watchdog drives this interrupt if present.
3	Hard Fault	-1	A Hard Fault is an exception that occurs because of an error during exception processing.
4	Memory manage fault	Programmable	Exception from MPU.
5	Bus fault	Programmable	Bus error of bus access to the area that is not controlled by the MPU.
6	Use fault	Programmable	Error about operating instruction including undefined instruction.
7-10	Reserved	-	-
11	SVCall	Programmable	Call.
12	Debug Monitor	Programmable	Exception that is triggered by the SVC instruction.
13	Reserved	-	
14	PendSV	Programmable	PendSV is an interrupt-driven request for system-level service.
15	SysTick	Programmable	SysTick exception is an exception the system timer generates when it reaches zero.

Exceptions from 16 to the maximum that is configured in the processor are defined by the implementation. Arm recommends that you re-use the following connections where your peripheral types are aligned with the examples. For systems that have fewer interrupts, the low numbered interrupt connections should be maintained if possible.

The following table shows the reference interrupts.

Table 4-4: Interrupts

No.	Name	Source	Description
16	CPU0INTISR0	UART0	UART0 TX and RX combined
17	CPU0INTISR1	Not used	-
18	CPU0INTISR2	UART1	UART1 TX and RX combined

No.	Name	Source	Description
19	CPU0INTISR3	Reserved	Reserved for APB slaves
20	CPU0INTISR4	Reserved	Reserved for APB slaves
21	CPU0INTISR5	RTC	RTC interrupt
22	CPU0INTISR6	GPIO0	Combined
23	CPU0INTISR7	Reserved	Combined
24	CPU0INTISR8	Timer0	Timer interrupt
25	CPU0INTISR9	Timer1	Timer interrupt
26	CPU0INTISR10	Dual timer	Combined timer interrupt
27	CPU0INTISR11	Reserved	Reserved for APB slaves
28	CPU0INTISR12	All UARTs	UART combined overflow
29	CPU0INTISR13	Reserved	Reserved for APB slaves
30	CPU0INTISR14	Reserved	Reserved for APB Slaves
31	CPU0INTISR15	Reserved	Reserved for APB Slaves
32	CPU0INTISR16	GPIO0-0	-
33	CPU0INTISR17	GPIO0-1	-
34	CPU0INTISR18	GPIO0-2	-
35	CPU0INTISR19	GPIO0-3	-
36	CPU0INTISR20	GPIO0-4	-
37	CPU0INTISR21	GPIO0-5	-
38	CPU0INTISR22	GPIO0-6	-
39	CPU0INTISR23	GPIO0-7	-
40	CPU0INTISR24	GPIO0-8	-
41	CPU0INTISR25	GPIO0-9	-
42	CPU0INTISR26	GPIO0-10	-
43	CPU0INTISR27	GPIO0-11	-
44	CPU0INTISR28	GPIO0-12	-
45	CPU0INTISR29	GPIO0-13	-
46	CPU0INTISR30	GPIO0-14	-
47	CPU0INTISR31	GPIO0-15	-
48	CPU0INTISR32	Cache controller	Cache or memory errors
49	CPU0INTISR33	Embedded flash controller	Combined interrupts from flash controller
60	CPU0INTISR44	TRNG	TRNG interrupt
61-255	CPU0INTISR45-239	Not used	-

The interrupt signals from the peripherals are not connected directly to the interrupt controller. All signals go to the integration layer and can be connected differently by the system designer. Arm recommends that you reuse the mapping shown here for easier software reuse between similar devices.

4.3.2 Registers

A summary of the interrupt controller registers is listed in the following table.

Table 4-5: Summary of interrupt controller registers

Address	Name	Type	Reset value	Description
0xE000E004	ICTR	RO	-	Interrupt Controller Type Register
0xE000E100-0xE000E11C	NVIC_ISER0-NVIC_ISER7	RW	0	Interrupt Set Enable Registers
0xE000E180-0xE000E19C	NVIC_ICER0-NVIC_ICER7	RW	0	Interrupt Clear Enable Registers
0xE000E200-0xE000E21C	NVIC_ISPR0-NVIC_ISPR7	RW	0	Interrupt Set Pending Registers
0xE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	0	Interrupt Clear Pending Registers
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IABR7	RO	0	Interrupt Active Bit Registers
0xE000E400-0xE000E41F	NVIC_IPR0-NVIC_IPR7	RW	0	Interrupt Priority Registers

For more information on the interrupt controller, see the following documents:

- *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- *Arm®v7-M Architecture Reference Manual*.

4.4 Wakeup Interrupt Controller (WIC)

The *Wakeup Interrupt Controller* (WIC) is a peripheral that can detect an interrupt and wake the processor from deep sleep mode. The WIC is enabled only when the system is in deep sleep mode.

The WIC is not programmable, and does not have any registers or user interface. It operates entirely from hardware signals.

When the WIC is enabled and the processor enters deep sleep mode, the power management unit in the system might powerdown the Cortex®-M3 processor while retaining its software context. Wakeup operation requires the WIC to be powered up. When the WIC receives an interrupt, it takes several clock cycles to wakeup the processor and restore its state to enable it to process the interrupt. Therefore, the interrupt latency is increased in deep sleep mode.

For more information on the WIC, see the following documents:

- *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- *Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual*.

4.5 Timer

A summary of the registers in the timer is provided in the following table.

Table 4-6: Summary of timer registers

Name	Type	Width	Reset value	Address offset	Description
CTRL	RW	4	0x0	0x000	3: Interrupt enable. 2: Select external input as clock. 1: Select external input as enable. 0: Enable.
VALUE	RW	32	0x00000000	0x004	Current value
RELOAD	RW	32	0x00000000	0x008	Reload value. A write to this register sets the current value.
INTSTATUS	RO	1	0x0	0x00C	Timer interrupt. A read returns the timer interrupt status.
INTCLEAR	WO	1	0x0	0x00C	0: No effect. 1: Clears the timer interrupt.
PID4	RO	8	0x04	0xFD0	Peripheral ID register 4
PID5	RO	8	0x00	0xFD4	Peripheral ID register 5
PID6	RO	8	0x00	0xFD8	Peripheral ID register 6
PID7	RO	8	0x00	0xFDC	Peripheral ID register 7
PID0	RO	8	0x22	0xFE0	Peripheral ID register 0 [7:0] Part number[7:0].
PID1	RO	8	0xB8	0xFE4	Peripheral ID register 1 [7:4] jep106_id_3_0. [3:0] Part number[11:8].
PID2	RO	8	0x0B	0xFE8	Peripheral ID register 2 [7:4] Revision. [3] jedec_used. [2:0] jep106_id_6_4.
PID3	RO	8	0x00	0xFEC	Peripheral ID register 3 [7:4] ECO revision number. [3:0] Customer modification number.
CID0	RO	8	0x0D	0xFF0	Component ID register 0
CID1	RO	8	0xF0	0xFF4	Component ID register 1
CID2	RO	8	0x05	0xFF8	Component ID register 2
CID3	RO	8	0xB1	0xFFC	Component ID register 3

4.6 System registers

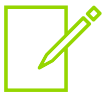
For information on the Cortex®-M3 registers, see the following documents:

- *Arm® Cortex®-M System Design Kit Technical Reference Manual*.
- *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- *Arm®v7-M Architecture Reference Manual*.

The following table lists the subsystem ROM table ID register (identical with the generic Cortex-M3 ROM table).

Table 4-7: Part number ID values

Class	Part Number	Part	Rev	ID0	ID1	ID2	ID3	ID4
0x1	0x741	SSE-050	r0p1	0xC3	0xB4	0x1B	0x00	0x04


Note

The peripheral ID in the highest-level ROM table of a system should be unique to that system. This means that you should replace this Arm default ID value with your own ID as described in the CoreSight architecture specification. The only exceptions to this rule are when either:

- The processor is configured with no debug.
- There is a higher level ROM table connected. In this case, it is correct to use the ROM table ID value without modification.

Appendix A Signal descriptions

This chapter summarizes the interface signals present in the SSE-050.

A.1 Clock and reset signals

All reset signals are active-LOW. They can be asserted asynchronously, but must be deasserted synchronous to their respective clocks. All clocks, except for DAPSWCLKTCK and TRACECLKIN, must be synchronous and balanced.

The following table lists clock and reset signals in the SSE-050 interface.

Table A-1: Clocks and resets

Name	Direction	Width	Description
AHB2APBHCLK	Input	1	AHB to APB bridge clock.
CPU0DAPCLK	Input	1	A clock signal for the debug bus interface from the *Debug Port* (DP) component, for example SWJ-DP. This can be asynchronous to other clock signals.
CPU0DAPCLKEN	Input	1	An enable signal for CPU0DAPCLK.
CPU0DAPRESETn	Input	1	Resets the debug bus connected to the AHB-AP inside the Cortex®-M3 processor.
CPU0FCLK	Input	1	A free-running clock. This must be active when the processor is running, for debugging, and for the *Nested Vectored Interrupt Controller* (NVIC) to detect interrupts. The SSE-050 implements the WIC with LATCH thus during WIC-based sleep it can be gated.
CPU0HCLK	Input	1	A system clock. This must be the same as CPU0FCLK when active. You can gate this off when the processor is in sleep.
CPU0PORESETn	Input	1	Power-on reset. Resets the entire Cortex®-M3 system and debug components.
CPU0STCALIB	Input	26	Calibration signal for alternative clock source of SysTick timer.
CPU0STCLK	Input	1	Clock enable of the alternative clock source of SysTick timer. CPU0FCLK is gated with this CPU0STCLK.
CPU0SYSRESETn	Input	1	System Reset. Resets the processor and the WIC excluding debug logic in the NVIC, FPB, DWT, and ITM.
DAPNTRST	Input	1	Debug nTRST reset initializes the state of the SWJ-DP TAP controller. This reset is typically used by the RealView ICE module for hot-plug connection of a debugger to a system. It initializes the SWJ-DP controller without affecting the normal operation of the processor.
DAPSWCLKTCK	Input	1	Serial wire clock and JTAG Test clock. Can be asynchronous.
DAPNPOTRST	Input	1	Debug port power on reset. Deassertion must be synchronized to SWCLKTCK.
MTXHCLK	Input	1	AHB matrix interconnect clock.
MTXHRESETn	Input	1	AHB matrix interconnect reset.
SRAM<0..3>HCLK	Input	1	AHB clock of the SRAM bridges.
TIMER0PCLK	Input	1	The free running clock used for timer operation. This must be the same frequency as, and synchronous to, the TIMER0PCLKG signal.
TIMER0PCLKG	Input	1	This clock is for register reads and writes of APB TIMER0. It can be gated off if there is no APB activity.

Name	Direction	Width	Description
TIMER0PRESETn	Input	1	Asynchronous active low reset for APB TIMER0. Deassert the reset synchronously to TIMER0PCLK.
TIMER1PCLK	Input	1	The free running clock used for timer operation. This must be the same frequency as, and synchronous to, the TIMER1PCLKG signal.
TIMER1PCLKG	Input	1	This clock is for register reads and writes of APB TIMER1. It can be gated off if there is no APB activity.
TIMER1PRESETn	Input	1	Asynchronous active low reset for APB TIMER1. Deassert the reset synchronously to TIMER1PCLK.
TPIUCLK	Input	1	APB and ATB interface clock.
TPIUTRACECLKIN	Input	1	Trace out port source clock.
TPIUCLKEN	Input	1	Clock enable for TPIUCLK.
TPIUTRESETn	Input	1	Trace out port active-LOW reset. This is asynchronously asserted and must be synchronously deasserted to TPIUTRACECLKIN.

A.2 Interrupt signals

The following table lists clock and reset signals in the SSE-050 interface.

Table A-2: Interrupt signals

Name	Direction	Width	Description
CPU0INTISR[239:0]	Input	240	External interrupt signals. The number of functional interrupt signals depends on your implementation. The Cortex®-M3 processor does not implement synchronizers for the CPU0INTISR input. To use asynchronous interrupts, you must implement external synchronizers to reduce the possibility of metastability issues.
CPU0INTNMI	Input	1	Non-maskable Interrupt. The number of functional interrupt signals depends on your implementation. The Cortex®-M3 processor does not implement synchronizers for the CPU0INTNMI input. To use asynchronous interrupts, you must implement external synchronizers to reduce the possibility of metastability issues. If the input is connected to an I/O pad, a noise filter must be applied.
CPU0CURRPRI[7:0]	Output	8	Indicates what priority interrupt, or base boost, is being used. CURRPRI represents the preemption priority, and does not indicate secondary priority.
CPU0AUXFAULT[31:0]	Input	32	Auxiliary fault status information from the system.

A.3 SRAM signals

The following table lists the interface signals for the AHB2SRAM subsystem.

Table A-3: AHB2SRAM Interfaces

Name	Direction	Width	Description
SRAM<0..3>RDATA	Input	32	SRAM read data bus.
SRAM<0..3>ADDR	Output	13	SRAM address (word address).
SRAM<0..3>WREN	Output	4	SRAM byte write enable. Active HIGH.

Name	Direction	Width	Description
SRAM<0..3>WDATA	Output	32	SRAM write data.
SRAM<0..3>CS	Output	1	SRAM chip select. Active HIGH.

A.4 Timer signals

The following table lists the interface signals for the timer subsystem.

Table A-4: Timer

Name	Direction	Width	Description
TIMER0EXTIN	Input	1	Timer0 external input. The external clock. This must be slower than half of the TMER0PCLK clock because it is sampled by a double flip-flop and then goes through edge-detection logic when the external inputs act as a clock.
TIMER1EXTIN	Input	1	Timer1 external input. The external clock, must be slower than half of the TMER1PCLK clock because it is sampled by a double flip-flop and then goes through edge-detection logic when the external inputs act as a clock.
TIMER0PRIVMODEN	Input	1	Defines if the timer memory mapped registers are writeable only by privileged access.
TIMER1PRIVMODEN	Input	1	Defines if the timer memory mapped registers are writeable only by privileged access.
TIMER0TIMERINT	Output	1	Timer0 interrupt output.
TIMER1TIMERINT	Output	1	Timer1 interrupt output.

A.5 Bus signals

The following table lists the signals for the three AHB master interfaces. Not shown in the list is the TARGEXP<0..1> prefix before each signal name for the expansion ports, and TARGFLASH0 suffix for the flash memory AHB master interface.

Table A-5: External AHB master port signals

Name	Direction	Width	Description
HSEL	Output	1	Slave Select.
HADDR	Output	32	Address bus.
HTRANS	Output	2	Transfer type.
HWRITE	Output	1	Transfer direction.
HSIZE	Output	3	Transfer size.
HBURST	Output	3	Burst type.
HPROT	Output	4	Protection control.
HMASTER	Output	4	Master select.
HWDATA	Output	32	Write data.
HMASTLOCK	Output	1	Locked sequence.
HREADYMUX	Output	1	Transfer done.
HAUSER	Output	1	Address USER signals (Not used by the subsystem Cortex®-M3 processor).

Name	Direction	Width	Description
EXREQ	Output	1	Exclusive request.
MEMATTR	Output	2	Memory attributes.
HWUSER	Output	4	Write-data USER signals (Not used by the subsystem Cortex®-M3 processor).
HRDATA	Input	32	Read data bus.
HREADYOUT	Input	1	HREADY feedback.
HRESP	Input	1	Transfer response.
HRUSER	Input	3	Read-data USER signals (Not used by the subsystem Cortex®-M3 processor).
EXRESP	Input	1	Exclusive response.

The following table lists the signals for the two AHB slave interfaces. Not shown in the list is the INITEXP<0..1> prefix before each signal name.

Table A-6: External AHB slave port signals

Name	Direction	Width	Description
HSEL	Input	1	Slave Select.
HADDR	Input	32	Address bus.
HTRANS	Input	2	Transfer Type.
HWRITE	Input	1	Transfer Direction.
HSIZE	Input	3	Transfer Size.
HBURST	Input	3	Burst type.
HPROT	Input	4	Protection Control.
HMASTER	Input	4	Master Select.
HWDATA	Input	32	Write Data.
HMASTLOCK	Input	1	Locked Sequence.
HAUSER	Input	1	Address USER signals (Not used by the subsystem Cortex®-M3 processor).
EXREQ	Input	1	Exclusive Request signal.
MEMATTR	Input	2	Memory Attribute signals.
HWUSER	Input	4	Write-data USER signals (Not used by the subsystem Cortex®-M3 processor).
HRDATA	Output	32	Read data bus.
HREADY	Output	1	HREADY feedback.
HRESP	Output	1	Transfer response.
HRUSER	Output	3	Read-data USER signals (Not used by the subsystem Cortex®-M3 processor).
EXRESP	Output	1	Exclusive Response.

The following table lists the signals for the two APB slave interfaces. Not shown in the list is the APBTARGETEXP<n> prefix before each signal name, where n is in the range 2-15.

Table A-7: External APB target port signals

Name	Direction	Width	Description
PSEL	Output	1	Slave select signal.
PENABLE	Output	1	Strobe to time all accesses. Used to indicate the second cycle of an APB transfer.

Name	Direction	Width	Description
PADDR	Output	12	Address bus.
PWRITE	Output	1	APB transfer direction.
PWDATA	Output	32	32-bit write data bus.
PRDATA	Input	32	32-bit read data bus.
PREADY	Input	1	Driven LOW if extra wait states are required to complete the transfer.
PSLVERR	Input	1	Indicates SLVERR response.
PSTRB	Output	1	Write strobes. This signal indicates which byte lanes to update during a write transfer. There is one write strobe for each eight bits of the write data bus. PSTRB[n] corresponds to PWDATA[(8n + 7):(8n)]. Write strobes must not be active during a read transfer.
PPROT	Output	3	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.

A.6 Debug and Trace signals

This section list signals related to debug and trace.

A.6.1 JTAG and SWD signals

The following table lists the interface signals for the JTAG and SWD subsystem.

Table A-8: JTAG and SW Debug access functional signals

Name	Direction	Width	Description
DAPSWDITMS	Input	1	Debug TMS.
DAPSWDO	Output	1	Serial Wire Data Out.
DAPSWDOEN	Output	1	Serial Wire Output Enable.
DAPTDI	Input	1	Debug TDI.
DAPTDO	Output	1	Debug TDO.
DAPNTDOEN	Output	1	TDO output pad control signal.
DAPJTAGNSW	Output	1	JTAG or Serial-Wire selection: 1 - JTAG mode. 0 - SW mode.
DAPJTAGTOP	Output	1	JTAG-DP TAP controller in one of four top states TLR, RTI, Sel-DR, Sel-IR.

A.6.2 CPU debug signals

The following table lists the interface signals related to CPU debug.

Table A-9: CPU debug signals

Name	Direction	Width	Description
CPU0EDBGRQ	Input	1	External debug request. This is combined internally with the ETM debug request. This signal must be synchronous to CPU0FCLK.
CPU0DBGRESTART	Input	1	External restart request. The processor exits the halt state when the CPU0DBGRESTART signal is deasserted during 4-phase handshaking.
CPU0DBGRESTARTED	Output	1	Handshake for CPU0DBGRESTART. Devices driving CPU0DBGRESTART must observe this signal to generate the required 4-phase handshaking.

A.6.3 Debug authentication control

The following table lists the interface signals related to debug authentication and security.

Table A-10: Debug authentication and security signals

Port name	Direction	Width	Description
CPU0DBGEN	Input	1	External debug enable. If CPU0DBGEN is deasserted, the halt debugging feature of the processor is disabled. If CPU0DBGEN is asserted, you can use debug features, but you must set other enables, C_DEBUGEN for example, to enable debug events such as halt to occur. Either tie HIGH or connect to a debug access controller if required.
CPU0NIDEN	Input	1	Non-Invasive debug enable. NIDEN must be HIGH to enable the ETM trace unit to trace instructions.
CPU0DAPEN	Input	1	Input AHB-AP enable. Enables the AHB-AP memory access functionality. In a typical arrangement, you can tie this signal HIGH. Connect HIGH when enabling debug accesses. If this signal is LOW, the debugger can still access registers inside the AHB-AP module inside the Cortex®-M3 processor, but cannot access the memory map using the AHB interface.
CPU0FIXMASTERTYPE	Input	1	The AHB-AP can issue AHB transactions with a HMASTER value of either 1, to indicate DAP, or 0, to indicate processor data side, depending on how the AHB-AP is configured using the MasterType bit in the AHB-AP Control and Status Word Register. You can use FIXMASTERTYPE to prevent this if required. If it is tied HIGH, then the HMASTER that the AHB-AP issues is always 1, to indicate DAP, and it cannot imitate the processor. If it is tied LOW, then HMASTER can be issued as either 0 or 1.

A.6.4 Trace signals

The following table lists the miscellaneous signals for the Trace subsystem.

Table A-11: CPU trace signals

Name	Direction	Width	Description
CPU0TSVALUEB	Input	48	Global timestamp value.
CPU0ETMINTNUM	Output	9	Marks the interrupt number of the current execution context.
CPU0ETMINTSTAT	Output	3	Interrupt status.
CPU0ETMEXTIN	Input	2	ETM external inputs.

The following table lists the interface signals for the Trace Port Interface Unit.

Table A-12: External TPIU interface

Name	Direction	Width	Description
TPIUTRACECLK	Output	1	Output clock, used by the TPA to sample the other pins of the trace out port. This runs at half the speed of TPIUTRACECLKIN, and data is valid on both edges of this clock (clock derived from TPIUTRACECLKIN).
TPIUTRACEDATA	Output	4	Output data. A system might not connect all the bits of this signal to the trace port pins, depending on the number of pins available and the bandwidth required to output trace.
TPIUTRACESWO	Output	1	Serial Wire Viewer data.
TPIUSWOACTIVE	Output	1	Controls the multiplexor if SWO shared with TRACEDATA[0]. Arm recommends implementing SWO shared with JTAG-TDO, therefore this pin shall be left unconnected.

A.6.5 HTM signals

The following table lists the signals for the *AHB Trace Macrocell* (HTM).

Table A-13: HTM interface signals

Name	Direction	Description	Connection information
HTMDHADDR[31:0]	Output	32-bit address.	Connect to HTM if implemented.
HTMDHTRANS[1:0]	Output	Indicates the type of the current data transfer. Can be IDLE, NONSEQUENTIAL, or SEQUENTIAL.	Connect to HTM if implemented.
HTMDHSIZE[1:0]	Output	Indicates the size of the access. Can be 8 bits, 16 bits, or 32 bits.	Connect to HTM if implemented.
HTMDHBURST[2:0]	Output	Indicates if the transfer is part of a burst.	Connect to HTM if implemented.
HTMDHPROT[3:0]	Output	Provides information on the access.	Connect to HTM if implemented.
HTMDHWDATA[31:0]	Output	32-bit write data bus.	Connect to HTM if implemented.
HTMDHWRITE	Output	Write not read.	Connect to HTM if implemented.
HTMDHRDATA[31:0]	Output	Read data bus.	Connect to HTM if implemented.
HTMDHREADY	Output	Ready signal.	Connect to HTM if implemented.

Name	Direction	Description	Connection information
HTMDHRESP[1:0]	Output	The transfer response status. Can be OKAY or ERROR.	Connect to HTM if implemented.

A.7 CPU control, status, and power management signals

This section describes power-management and control signals. Power management requires design choices made at the SoC level.

A.7.1 CPU status and control signals

The following table lists the status and control signals for the CPU subsystem.

Table A-14: CPU control and status

Name	Direction	Width	Description
CPU0HALTED	Output	1	In halting mode debug. HALTED remains asserted while the core is in debug.
CPU0MPUDISABLE	Input	1	If asserted the MPU is invisible and unusable. Tie HIGH to disable the MPU. Tie LOW to enable the MPU, if present.
CPU0SLEEPING	Output	1	Indicates that the processor is in sleep mode.
CPU0SLEEPDEEP	Output	1	Indicates that the processor is in deep sleep mode.
CPU0SLEEPHOLDREQn	Input	1	Request to extend sleep. Can only be asserted when CPU0SLEEPING is HIGH.
CPU0SLEEPHOLDACKn	Output	1	Acknowledge for CPU0SLEEPHOLDREQn.
CPU0WAKEUP	Output	1	Signal to the PMU that indicates that a wake-up event has occurred and the processor system domain requires its clocks and power to be restored.
CPU0WICSENSE	Output	Implementation defined	These indicate the input lines that cause the WIC to generate the CPU0WAKEUP signal (optional, for testing).
CPU0WICENREQ	Input	1	Request for deep sleep to be WIC-based deep sleep. The PMU drives this signal.
CPU0WICENACK	Output	1	Acknowledge signal for CPU0WICENREQ. If you do not require PMU, then tie this signal HIGH to enable the WIC if the WIC is implemented.
CPU0TRCENA	Output	1	Indicates that Trace is enabled, can be used to gate TPIUCLK.
CPU0ETMEN	Output	1	Indicates ETM is enabled, can be used to gate TPIUCLK.
CPU0SYSRESETREQ	Output	1	Processor control - system reset request. The AIRC.SYSRESETREQ register controls this signal.
CPU0LOCKUP	Output	1	Indicates that the core is in lockup state.
CPU0BRCHSTAT	Output	4	Branch status in decode.
CPU0ETMTRIGOUT	Output	1	Driven HIGH when the ETM trigger condition matches.

A.7.2 Power management signals

The following table lists the power management signals for the debug subsystem.

See also A.6 Debug and Trace signals on page 45.

Table A-15: Debug power-management signals

Signal name	Direction	Width	Clock	Description
DAPCDBGPWRUPREQ	Output	1	DAPSWCLKTCK	Indicates an external debugger request to the PMU to power up the debug domain. This signal must be synchronized before use.
DAPCDBGPWRUPACK	Input	1	Asynchronous	Indicates that the debug domain is powered up in response to DAPCDBGPWRUPREQ being HIGH. This signal is resynchronized internally to DAPSWCLKTCK.
DAPCSYSPWRUPREQ	Output	1	DAPSWCLKTCK	Indicates an external debugger request to the PMU to power up the debug domain. This signal must be synchronized before use.
DAPCSYSPWRUPACK	Input	1	Asynchronous	Indicates that the debug domain is powered up in response to DAPCSYSPWRUPREQ being HIGH. This signal is resynchronized internally to DAPSWCLKTCK.
DAPNCDBGPWRDN	Input	1	Asynchronous	Debug infrastructure powerdown control. Controls the clamps of the DAP APB interface.
DAPCDBGRSTREQ	Output	1	DAPSWCLKTCK	Indicates that an external debugger requested a debug reset to reset the controller. This signal must be synchronized before use.
DAPCDBGRSTACK	Input	1	Asynchronous	Indicates that the debug domain reset has completed. This signal is resynchronized internally by DAPSWCLKTCK.

The following table lists the power-management signals for the interconnect subsystem. See also A.5 Bus signals on page 43.

Table A-16: Interconnect power-management signals

Signal name	Direction	Width	Clock	Description
APBQACTIVE	Output	1	MTXHCLK	APB bus active signal for global clock gating control of all APB peripherals attached to the SSE-050 that supports gated APB clock.
TIMER0PCLKQACTIVE	Output	1	MTXHCLK	APB bus active signal for clock gating control of Timer 0 APB clock TIMER0PCLKG (for example PSEL).
TIMER1PCLKQACTIVE	Output	1	MTXHCLK	APB bus active signal for clock gating control of Timer 1 APB clock TIMER1PCLKG (that is, PSEL).
APBTARGEXPnPSEL	Output	1	MTXHCLK	External APB TARGET SEL signals from the AHB slave interface can be reused for power management.

Related information

Debug and Trace signals on page 45
Bus signals on page 43

A.8 Memory remap signals

The MTXREMAP signals control the remapping of the boot memory range.

When reset is not asserted, you must ensure that the memory remap signals remain static.

Table A-17: Memory remap signals

Name	Direction	Width	Description
MTXREMAP	Input	4	REMAP[0]: Determines the size of the eFlash memory region: **1** 256KB region for flash. **0** 512KB region for flash. REMAP[1]: If SRAM1 is not present, the corresponding memory range can be mapped to the AHB expansion port by tying this signal HIGH. REMAP[2]: If SRAM2 is not present, the corresponding memory range can be mapped to AHB expansion port by tying this signal HIGH. REMAP[3]: If SRAM3 is not present, the corresponding memory range can be mapped to AHB expansion port by tying this signal HIGH.

A.9 DFT signals

The following table list signals related to test mode.

Table A-18: DFT signals

Name	Direction	Width	Description
DFTSCANMODECPU0	Input	1	Reset bypass to disable internal generated reset for testing, for example ATPG.
DFTCGENCPU0	Input	1	Clock gating bypass to disable internal clock gating for testing. This signal is used to ensure safe shift where the clock is forced on during the shift mode.
DFTSE	Input	1	Scan mode enable for ETM, if present.

Appendix B Revisions

This appendix describes the technical changes between released issues of this book.

B.1 Revisions

Table B-1: Issue 0000-00

Change	Location	Affects
First release	-	-

Table B-2: Differences between issue 0000-00 and issue 0001-00

Change	Location	Affects
System register revision and part number ID values changed	System registers	Issue 0001-00

Table B-3: Differences between issue 0001-00 and issue 0001-01

Change	Location	Affects
Radio reference removed	Communication interface	Issue 0001-01
Stand-alone TRNG reference removed	Features	Issue 0001-01