Arm® Corstone™-700

Revision: r1p0

Technical Overview

101893_0100_02_en

Arm® Corstone™-700

Technical Overview

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-01	13 December 2019	Non-Confidential	First LAC release for r0p0
0100-02	03 July 2020	Non-Confidential	First EAC release for r1p0

Non-Confidential Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

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

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

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Web Address

www.arm.com

Contents

Arm® Corstone™-700 Technical Overview

Preface

Chapter 1 Introduction

Chapter 2 Corstone-700 IP descriptions

Preface

This preface introduces the *Arm® Corstone™-700 Technical Overview*.

It contains the following:

About this book

This Technical Overview is for Arm® Corstone™-700. It describes Corstone-700 and gives a summary of the included products.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r1p2, where:

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

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

Intended audience

This book is written for hardware or software engineers who want an overview of the components and functionality in Corstone-700.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
This chapter describes Corstone-700, which includes two packages: the Corstone-700 System Package (BP312) and Corstone-700 Secure Enclave (BP315).

Chapter 2 Corstone-700 IP descriptions
This chapter describes the IP products included in the Corstone-700.

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

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.

Typographic conventions

italic
Introduces special terminology, denotes cross-references, and citations.

bold
Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`
Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`monospace`
Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`
Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`
Denotes language keywords when used outside example code.

`<and>`

Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS

Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Additional reading

Arm publications

This document contains information that is specific to this product. See the following documents for other relevant information. See *https://developer.arm.com*, for Arm documentation.

- *Arm® Corstone™ SSE-700 Subsystem Technical Reference Manual* (101418)
- *Arm® Corstone™ SSE-700 Secure Enclave Technical Reference Manual* (101870)
- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Overview* (101123)
- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual* (101104)
- *Arm® SSE-123 Example Subsystem Technical Overview* (101371)
- *Arm® SSE-123 Example Subsystem Technical Reference Manual* (101370)
- *Arm® CoreLink™ SSE-050 Subsystem Technical Reference Manual* (100918)
- *Arm® Cortex®-M System Design Kit Technical Reference Manual* (DDI 0479)
- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* (DDI 0571)
- *Arm® CoreSight™ SoC-400 Technical Reference Manual* (100536)
- *Arm® CoreSight™ System-on-Chip SoC-600 Technical Reference Manual* (100806)
- *Arm® CoreSight™ STM-500 System Trace Macrocell Technical Reference Manual* (DDI 0528)
- *Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual* (101484)
- *Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual* (101059)
- *Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual* (101150)
- *Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual* (101130)
- *Arm® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual* (100361)
- *Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview* (100459)
- *Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual* (101375)
- *Arm® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual* (100361)
- *Arm® CoreLink™ CG092 AHB Flash Cache Technical Reference Manual* (DDI 0569)
- *PrimeCell UART (PL011) Technical Reference Manual* (DDI 0183)
- *Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual* (DDI 0224C)
- *Arm® True Random Number Generator (TRNG) Technical Reference Manual* (100976)
- *Arm® Cortex®-M3 Processor Technical Reference Manual* (100165)
- *Arm® Cortex®-M33 Processor Processor Technical Reference Manual* (100230)
- *Arm® Cortex®-M23 Processor Technical Reference Manual* (DDI 0550)
- *Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview* (100459)
- *Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (DDI 0475)
- *Arm® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (DSU 0026)
- *Arm® CoreLink™ QVN-400 Network Interconnect Advanced QoS for Virtual Networks Supplement to Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (DSU 0027)
- *Arm® CoreLink™ TLX-400 Network Interconnect Thin Links Supplement to Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (DSU 0028)
- *Arm® CoreLink™ AXI4 to AHB-Lite XHB-400 Bridge Technical Reference Manual* (DDI 0523)
- PrimeCell *µDMA Controller (PL230) Technical Reference Manual* (DDI 0417)
- *AMBA® APB Protocol Specification* (IHI 0024)
- *PrimeCell Infrastructure AMBA® 3 AXI Internal Memory Interface (BP140) Technical Overview* (DTO 0009)
- *Arm® v8-M Architecture Reference Manual* (DDI 0553)
- *Arm® Debug Interface Architecture Specification ADIv6.0* (IHI 0074)
- *Arm® CoreSight™ Architecture Specification v3.0* (IHI 0029)

The following confidential books are only available to licensees or require registration with Arm:

- *Arm® Corstone™ SSE-700 Subsystem Configuration and Integration Manual* (101419)
- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Configuration and Integration Manual* (100224)
- *Arm® CoreLink™ NIC-400 Network Interconnect Implementation Guide* (DII 0273)
- *Arm® CoreLink™ NIC-400 Network Interconnect Integration Manual* (DII 0269)
- *Arm® SSE-123 Example Subsystem Configuration and Integration Manual* (101372)
- *Arm® CoreLink™ SSE-050 Subsystem Configuration and Integration Manual* (100919)
- *Arm® Cortex®-M System Design Kit Example System Guide* (DUI 0594)
- *Arm® Cortex®-M0 and Cortex®-M0+ System Design Kit Example System Guide* (DUI 0559)
- *Arm® CoreSight™ SoC-400 User Guide* (100490)
- *Arm® CoreSight™ SoC-400 System Design Guide* (100495)
- *Arm® CoreSight™ SoC-400 Implementation Guide* (100487)
- *Arm® CoreSight™ SoC-400 Integration Manual* (100491)
- *Arm® CoreLink™ SIE-200 System IP for Embedded Configuration and Integration Manual* (DIT 0067)
- *Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual* (101485)
- *Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual* (101060)
- *Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual* (101151)
- *Arm® CoreSight™ SDC-600 Secure Debug Channel Configuration and Integration Manual* (101131)
- *Arm® CoreLink™ LPD-500 Low Power Distributor Integration and Implementation Manual* (100362)
- *Arm® CoreLink™ CG092 AHB Flash Cache Configuration and Integration Manual* (DIT 0065B)
- *Arm® True Random Number Generator (TRNG) Configuration and Integration Manual* (100977)
- *PrimeCell Infrastructure AMBA® 3 AXI Internal Memory Interface (BP140) Design Manual* (DDI 0334)
- *Arm® CoreLink™ ADB-400 AMBA® Domain Bridge User Guide* (DUI 0615)
- *Arm® CoreLink™ XHB-500 Configuration and Integration Manual AXI5 to AHB5 bridge and AHB5 to AXI5 bridge* (101376)
- *Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual* (101151)
- *Arm® CoreSight™ System-on-Chip SoC-600 Configuration and Integration Manual* (100807).
- *Arm® Corstone™ SSE-700 Subsystem Release Note* (PJDOC-1779577084-27432).
- *Arm® Corstone™-700 Release Note* (PJDOC-1779577084-27950).
- *Arm® Corstone™-700 Secure Enclave Release Note* (PJDOC-1779577084).

- See *www.arm.com/cmsis* for embedded software development resources including the *Cortex Microcontroller Software Interface Standard* (CMSIS).
- See Arm Mbed™ platform, *https://www.mbed.com* for information on the Mbed tools including Mbed OS and online tools.
- For *Continuous Random Number Generation Testing* (CRNGT) see:
  - — FIPS 140-2, *Security Requirements for Cryptographic Modules*
  - — AIS-31, *Functionality Classes and Evaluation Methodology for True Random Number Generators*

Feedback

Feedback on this product

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

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

Feedback on content

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

- The title *Arm Corstone-700 Technical Overview*.
- The number 101893_0100_02_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

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

Chapter 1
Introduction

This chapter describes Corstone-700, which includes two packages: the Corstone-700 System Package (BP312) and Corstone-700 Secure Enclave (BP315).

It contains the following sections:

1.1 Corstone-700

Corstone-700 includes these packages:

———— **Note** ————

Each package grants licenses to a set of IP product components. These components must be downloaded separately.

————————————

Corstone-700 System Package

- Corstone SSE-700 subsystem, providing a flexible compute architecture that combines Cortex-A and Cortex-M processors and allows expansion for sensors, connectivity, video, audio, and machine learning at the edge
- Other subsystems, security, and system IPs to build secure SoCs for rich IoT nodes, gateways, and embedded applications

See *Arm® Corstone™-700 Release Note* for the component versions.

Corstone-700 Secure Enclave

- Corstone SSE-700 Secure Enclave subsystem, referred to in this guide as the Corstone SSE-700 SE subsystem, which provides hardware *Root of Trust* (RoT) and cryptographic functions that are integrated into the Corstone-700.
- Cortex
- Associated peripherals, such as timers and watchdogs

———— **Note** ————

The silicon integrator must add a Crypto Accelerator.

————————————

See *Arm® Corstone™-700 Secure Enclave Release Note* for the component versions.

1.2 Product structure

Corstone-700 includes licenses to the following subsystems, security IP, and system IP.

The following table lists IP that enables extension and modification of the Corstone SSE-700 subsystem. The table is followed by a descriptive list of the Subsystems, Security, and System IP available.

Table 1-1 Corstone-700 System Package

Description	Used in subsystem	Used in integration layer	Useful for extension	Notes
CoreLink NIC-450 network interconnect for AXI with options	Yes	Yes	Yes	Main interconnect
CoreLink XHB-500 – Heaney AXI to AHB bridge	No	Yes	Yes	Bridging component
CoreSight STM-500 System Trace Macro	Yes	No	No	Part of debug
CoreLink PCK-600 (Kratos) Power Control Kit	Yes	Yes	Yes	Power control elements. Includes LPD-500 Low Power InterfaceDistributor.
CoreSight SDC-600 (Chaucer) Secure Debug Channel	Yes	No	No	Secure debug interface
CoreSight SoC-600	Yes	Yes	Yes	Debug infrastructure
CoreSight SoC-400M	No	No	Yes	Debug infrastructure for Cortex-M
IntMemAxi	No	Yes	No	SRAM controller
UART	Yes	No	Yes	Serial port
CM0SDK / CMSDK Cortex-M System Design Kit	Yes	Yes	Yes	System IP for Cortex-M
AHB Flash Cache	No	No	Yes	Useful if using an eFlash
TRNG True Random Number Generator	No	No	Yes	For additional security
RTC Real Time Clock	No	Yes	No	Reference time
CoreLink SIE-200 System IP for AHB5	Yes	No	Yes	Necessary for Armv8-M systems
SSE-050 subsystem for Cortex-M3	No	No	Yes	Example Cortex-M3 system
SSE-123 subsystem for Cortex-M23	No	No	Yes	Example Cortex-M23 system
SSE-200 subsystem for Cortex-M33	No	No	Yes	Reference Armv8-M subsystem
SSE-700 subsystem including firewall and MHU	Yes	No	No	Main subsystem
CoreLink GFC-100 Flash Controller (single port)	No	Yes	No	Useful if using an eFlash
CoreLink GFC-200 Flash Controller (secure dual port)	No	No	Yes	Useful if using an eFlash

IP not included in Corstone-700 System Package

The Secure Enclave, Cortex processors, ISP, video, NPUs and so on are not included in the Corstone-700 System Package and are licensed separately. The following table shows the IP that enables extension and modification with the Corstone SSE-700 Secure Enclave subsystem. Contact your licensing manager to learn more about your licensing options.

Table 1-2 Corstone-700 Secure Enclave

Description	Used in subsystem	Used in integration layer	Useful for extension	Notes
Cortex-M0+	Yes	No	Yes	-
Secure enclave subsystem	Yes	No	No	Need to merge with a crypto engine, not provided, for example CryptoCell™-312

Corestone-700 SP subsystems

SSE-200

SSE-200 provides a high-performance and low-power computing subsystem for Cortex-M33 cores. It can be used as the foundation of a secure system because of system-level support for TrustZone® technologies.

See *2.3 SSE-200 subsystem* on page 2-26.

SSE-123

SSE-123 integrates an example subsystem for Cortex-M23 with key Arm components to give the core functionality of a system targeting IoT SoC designs. The susbsystem can be implemented as a standalone single core system or as part of a cluster system.

See *2.4 SSE-123 example subsystem* on page 2-28.

SSE-050

SSE-050 provides a starting point for a product in the IoT and embedded market segments using the Cortex-M3 cores. The integrator can extend the subsystem to provide an IoT endpoint system.

See *2.5 SSE-050 subsystem* on page 2-31.

Cortex-M System Design Kit

The CMSDK provides example systems for the Cortex-M0, Cortex-M0+, Cortex-M3, and Cortex-M4 cores, with reusable AMBA components for system-level development.

See *2.6 Cortex-M System Design Kit* on page 2-35.

Security and System IP

CoreLink SIE-200 System IP for Embedded

SIE-200 is a collection of interconnect, peripheral, and TrustZone controller components for use with a core that complies with the Armv8-M core architecture.

See *2.8 SIE-200 System IP for Embedded* on page 2-37.

CoreLink NIC-450 Network Interconnect

NIC-450 is a library of IP that supports the creation of optimized, multi-domain, AMBA interconnect solutions.

See *2.9 NIC-450 Network Interconnect* on page 2-38.

CoreSight System-on-Chip SoC-600

SoC-600 is a member of the Arm embedded debug and trace component family. It supports the Arm Debug Interface v6 and CoreSight v3 Architectures that enable you to build debug and trace functionality into your systems and to support debug and trace over existing functional interfaces.

See *2.10 SoC-600* on page 2-40.

CoreSight SoC-400M

SoC-400M enables customization of complex debug and trace capabilities for Cortex-M designs. It combines SoC-400 with the LIB-400M library that contains a configurable *Processor Integration Layer* (PIL) for multi-core design and IP-XACT models of the PIL.

See *2.11 SoC-400 and SoC-400M* on page 2-42.

CoreSight SDC-600 Secure Debug Channel

SDC-600 provides a dedicated channel for authentication between an external debugger and a debug target platform by using an unlocking mechanism.

See *2.12 SDC-600 Secure Debug Channel* on page 2-44.

CoreSight STM-500 System Trace Macrocell

STM-500 is a trace source that is integrated into a CoreSight system, and that is designed primarily for high-bandwidth trace of instrumentation embedded into software.

See *2.13 STM-500 System Trace Macrocell* on page 2-45.

CoreLink PCK-600 Power Control Kit

PCK-600 provides a set of configurable RTL components so you can create SoC clock and power control infrastructure. The components use the Arm Q-Channel and P-Channel low-power interfaces.

See *2.14 PCK-600 Power Control Kit* on page 2-48.

CoreLink GFC-200 Generic Flash Controller

GFC-200 comprises the generic part of a Flash controller in a SoC, so you can easily integrate an embedded Flash macro into your system. The GFC-200 supports accesses from two masters that can operate in separate domains, such as a Non-secure domain and a Secure domain.

See *2.15 GFC-200 Generic Flash Controller* on page 2-50.

CoreLink GFC-100 Generic Flash Controller

GFC-100 comprises the generic part of a Flash controller in a SoC. GFC-100 enables an embedded Flash macro to be integrated easily into your system.

See *2.16 GFC-100 Generic Flash Controller* on page 2-53.

CoreLink LPD-500 Low Power Distributor

LPD-500 is a standalone configurable component to distribute Q-Channel interfaces to multiple devices and subsystems. You can use Q-Channels to manage clock gating and power control. In the Corstone-700 system, LPD-500 is used in the SSE-200 Subsystem only. For any other clock gating and power control purposes, PCK-600 should be used.

See *Arm® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual*.

CoreLink CG092 AHB Flash Cache

CG092 is an instruction cache that is instantiated between the bus interconnect and the *embedded Flash* (eFlash) controller.

See *2.17 CG092 AHB Flash Cache* on page 2-56.

PrimeCell Infrastructure AMBA 3 AXI™ Internal Memory Interface (BP140)

You can use the `IntMemAxi` component to interface to a synthesized SRAM. It can also be connected to ROM.

See *2.18 AXI Internal Memory Interface (BP140)* on page 2-58.

CoreLink XHB-500 Bridge

The XHB-500 product provides an AMBA AXI5 to AHB5 bridge and an AHB5 to AXI5 bridge.

See *2.19 XHB-500 bridge* on page 2-59.

UART (PL011)

The UART is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by Arm.

See *2.20 UART* on page 2-62.

Real Time Clock (PL031)

The *Real Time Clock* (RTC) is an AMBA slave module that connects to the *Advanced Peripheral Bus* (APB). A 1Hz clock input to the RTC generates counting in one second intervals. The RTC provides an alarm function or long time base counter by generating an interrupt signal after counting a programmed number of cycles of the clock input.

See *2.21 Real Time Clock* on page 2-63.

True Random Number Generator

The *True Random Number Generator* (TRNG) provides an assured level of entropy (as analyzed by Entropy Estimation logic). You can use the output from the TRNG to seed deterministic random bit generators.

See *2.22 True Random Number Generator* on page 2-64.

1.3 Features

Corstone-700 is a flexible subsystem designed for secure, rich *Internet of Things* (IoT) applications. It integrates Arm Cortex-A and Arm Cortex-M processors together, with built-in security as one of its primary features.

The Corstone-700 contains a Secure Enclave subsystem. The Secure Enclave is a Cortex-M0+ based security subsystem that acts as the RoT for the system. It holds and generates keys, and provides cryptographic services and security controls to the host system.

Note

The Secure Enclave IP product components are licensed separately from Corstone SSE-700 subsystem and are available in the Corstone-700 Secure Enclave.

The following diagram shows the Corstone-700 subsystem design.

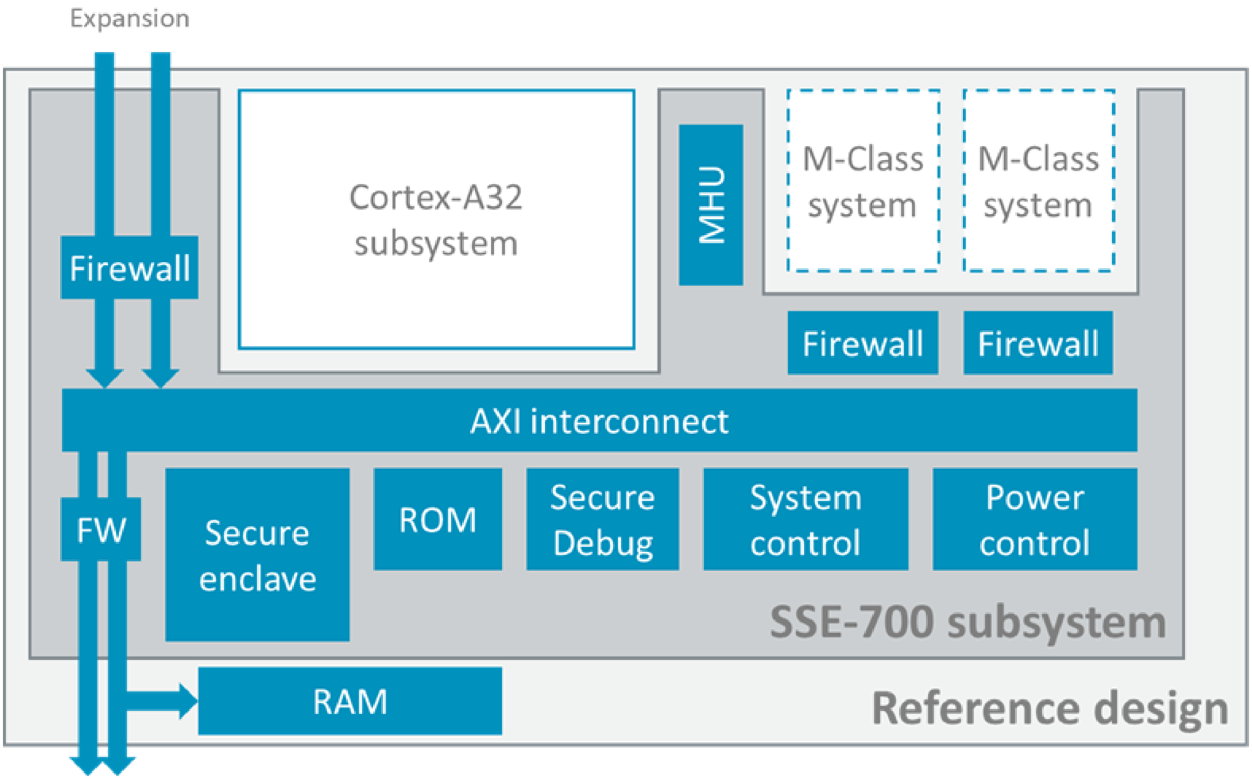


Figure 1-1 Corstone-700 subsystem design

Key features of the Corstone-700 are:

- Host system based on Linux-capable Cortex-A32 processor
- Up to 2 external systems, M-Class or other, for application-specific processing
- Dedicated integrated Secure Enclave based on Cortex-M0+ to provide root of trust and cryptographic acceleration functions
- Interrupt routing logic to distribute interrupts to the all the processing elements
- Inter-processor communication handled by the *Message Handling Unit* (MHU), supporting for both Secure and Non-secure messages among all the processing elements
- Extensive system security, with full TrustZone support and hardware compartmentalization via the firewall

- Pre-built advanced power management for hardware control of power and clock domains
- Secure and authenticated system debug for all processing elements

The Corstone SSE-700 subsystem components form just part of the SoC. You must extend and customize the subsystems for your application requirements.

To create a SoC, extend the Corstone SSE-700 subsystem. A complete system typically contains the following components:

Compute subsystem

The Corstone SSE-700 subsystem consists of one to four Cortex-A32 processor cores and associated bus, debug, controller, peripherals, and interface logic.

Memory and peripherals

An extended SoC requires extra memory, control, and peripheral components beyond the minimum subsystem components. Flash memory, for example, is not provided with the Corstone SSE-700 subsystem, but can be added through implemented interfaces.

Sensors and actuators

The reference design can be extended by adding sensors or actuator logic, such as temperature input or motor control output.

Software development environment

A reference Open Source software stack is available for Corstone-700. This enables product software development and helps bootstrap a variety of Arm Partner software and product solutions.

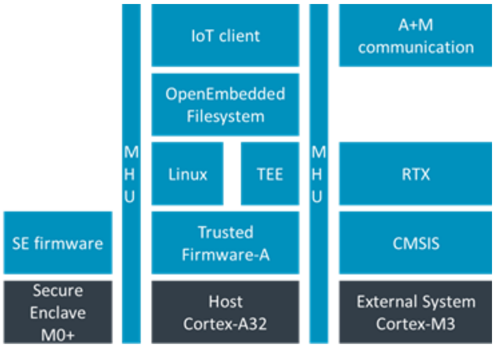


Figure 1-2 Reference Open Source software stack

The same software stack runs on the Corstone-700 *Fixed Virtual Platform* (FVP) model and Corstone-700 for MPS3 (AN543) FPGA. These are freely and publicly available.

This software stack provides:

- A reference OS/RTOS port for each of the compute components, including a Cortex-A hosted Linux stack with a reduced memory footprint
- Example drivers for key IP components

Supporting code is available, either directly in the relevant upstream projects or via public-facing git repositories hosted by Linaro.

For information on the software stack, FPGA, and FVP, see *www.community.arm.com/developer*.

For information on how to use the components that are licensed by Corstone-700 System Package, see the relevant component IP product documentation, starting with their Technical Reference Manual.

Chapter 2
Corstone-700 IP descriptions

This chapter describes the IP products included in the Corstone-700.

It contains the following sections:

2.1 SSE-700 overview

This section gives an overview of the product and its features.

For more information, see the SSE-700 documentation set:

- *Arm® Corstone™ SSE-700 Subsystem Technical Reference Manual*
- *Arm® Corstone™ SSE-700 Subsystem Configuration and Integration Manual*

This section contains the following subsections:

- *2.1.1 About SSE-700* on page 2-21.
- *2.1.2 Features of SSE-700* on page 2-22.

2.1.1 About SSE-700

The SSE-700 is a flexible subsystem designed to provide a secure solution for rich *Internet of Things* (IoT) applications based on Arm Cortex-A32 and Cortex-M3 processors.

SSE-700 is designed to cover a range of *Power, Performance, and Area* (PPA) applications, and enable extension for use-case specific applications. For example, sensors, cloud connectivity, or edge compute.

Connected embedded devices are exposed to many different security threats. SSE-700 implements the reference subsystem for a SoC that targets secure rich IoT applications with built-in security as one of its key features.

SSE-700 consists of:

- A reference subsystem
- An example integration layer
- An example Cortex-A32 and CoreLink GIC-400 instantiation
- A set of documentation

The following figure shows the topology of the SSE-700.

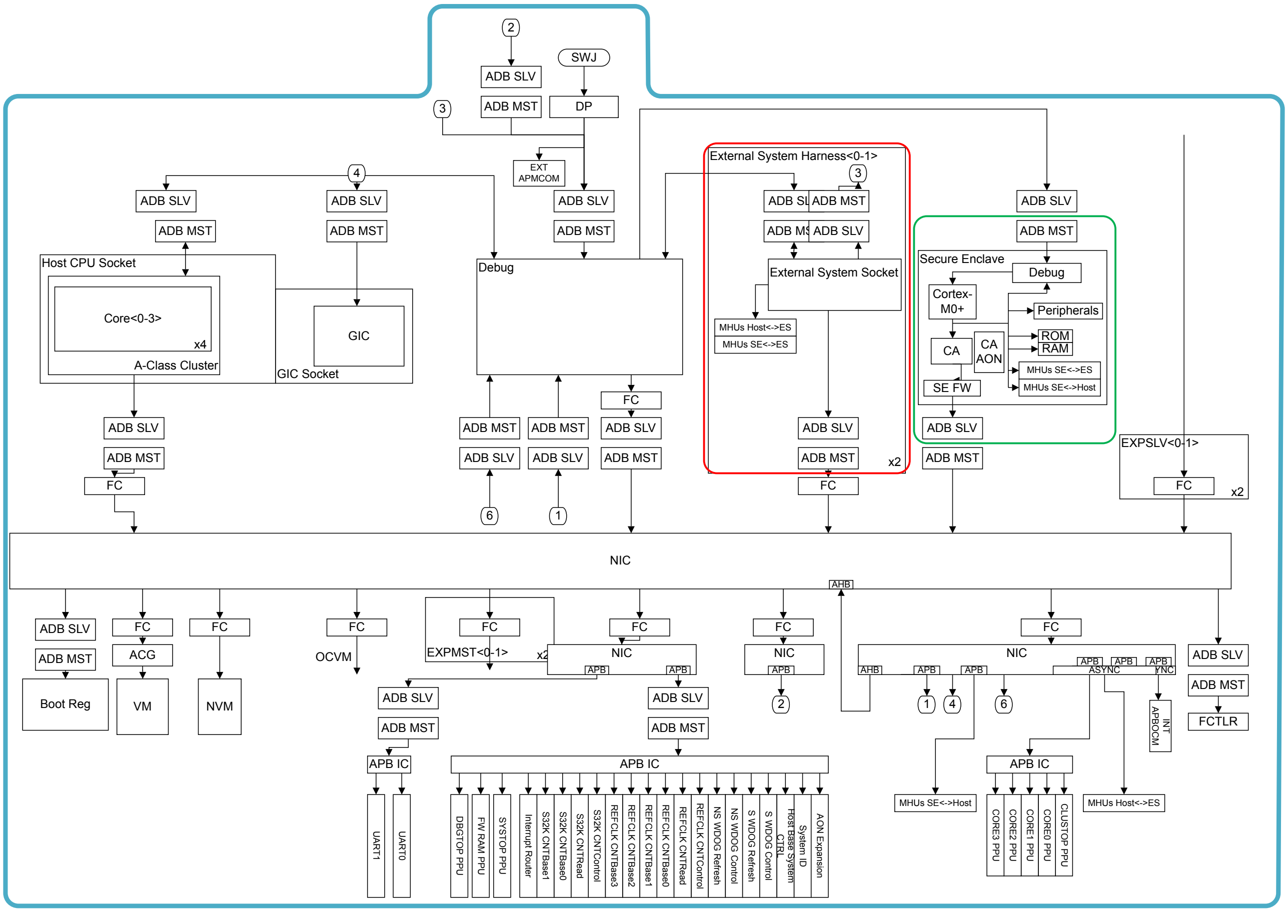


Figure 2-1 SSE-700 topology

An SSE-700-based SoC is formed of three types of systems:

- Secure Enclave, green box
- External Systems, red box
- Host System, the rest of SSE-700 subsystem

The Secure Enclave socket and External System Harnesses are connected, along with all other master and peripherals added by the integrator, to the interconnect inside the Host System.

2.1.2 Features of SSE-700

SSE-700 provides the reference design for the heart of a rich IoT SoC.

You can configure certain aspects of the SSE-700 design to meet the specific requirements of your SoC and intended application. For example, the number of shared interrupt inputs, or Host System Firewalls.

SSE-700 standardizes the following product features:

- Reset Controller to handle all of the system reset requirements including servicing reset requests and distributing resets to the subsystem
- Memory and interrupt maps of the Host System and Secure Enclave system
- Hardware address compartmentalization
- Boot and security, including a hardware isolated Root of Trust
- Communication between different parts of the SoC
- Timer and watchdog infrastructure
- Debug requirements
- Power, clock, and reset control

- Requirements for adding new or existing External Systems for use-case specific applications
- Requirements for adding use-case specific masters and peripherals to the Host System

The SSE-700 includes the following key components:

- A Cortex-A processor core and *Generic Interrupt Controller* (GIC) socket to support up to four Cortex-A32 cores and GIC-400.
- Secure Enclave socket, based on a Cortex-M0+ processor with dedicated SRAM, ROM, and peripherals, such as timers and watchdogs, into which a Crypto Accelerator can be integrated.
- *Message Handling Units* (MHUs) for communication between the different systems in the SSE-700. An MHU provides a unidirectional communication channel. Therefore, MHUs are provided in pairs to allow for bidirectional communication:
  - Two pairs of MHUs for communication between Cortex-A32 and Secure Enclave
  - Two pairs of MHUs for communication between Cortex-A32 and each External System
  - Two pairs of MHUs for communication between Secure Enclave and each External System
- Interrupt Router to handle routing interrupts from shared peripherals to the interrupt controller of a specific system.
- Firewall to provide:
  - Hardware compartmentalization of the Host System address space
  - Translation between the address space of the Secure Enclave system and the External System into the address space of the Host System
- Common debug infrastructure supporting single and multi-system debug, by self-hosted and external debug agents. Certificate base debug authentication is supported by SDC-600.
- Two Harnesses for integration of External Systems into SSE-700
- Advanced hardware autonomous power control design
- Shared peripherals, such as counters, timers, and watchdog, that can be used by any system in the SSE-700

2.2 Corstone SSE-700 Secure Enclave subsystem

This section gives an overview of the product and its features.

Note

The Corstone SSE-700 SE subsystem is included in the Corstone-700 Secure Enclave (BP315), but not in the Corstone-700 System Package (BP312).

For more information, see the SSE-700 Secure Enclave documentation set:

- *Arm® Corstone™ SSE-700 Secure Enclave Technical Reference Manual*
- *Arm® Corstone™ SSE-700 Secure Enclave Configuration and Integration Manual*

The Secure Enclave within the Corstone SSE-700 subsystem, is a Cortex-M0+ based security subsystem that acts as the root-of-trust for the system. It holds and generates keys, and provides cryptographic services and security controls to the Host System. For example:

- Authenticating the firmware of the Secure Enclave itself, the Host System and the External System.
- Enabling/disabling debug capabilities based on the secure state of the device.

At power-on reset, the Corstone SSE-700 SE subsystem is the first system to boot. It performs initial configuration of its own system and other components of the Corstone SSE-700 subsystem, such as the Boot Register and the Host System Firewall.

The Corstone SSE-700 SE subsystem consists of:

- An Arm-v6 Cortex-M0+ Processor, with an in-built *Nested Vectored Interrupt Controller* (NVIC)
- Dedicated ROM and SRAM
- PL011 UART
- *Message Handling Units* (MHUs) for communication with other systems.
- 2 CMSDK timers
- 2 CMSDK watchdogs:
  - Secure Enclave watchdog
  - SoC watchdog
- Secure Enclave Base System Control registers controlling the Corstone SSE-700 subsystem
- Secure Enclave System Control registers controlling the Corstone SSE-700 SE subsystem
- Independent clock and power control infrastructure
- *Security Control Bits* (SCB), controlling security access across the Corstone SSE-700 subsystem
- A Firewall, permitting the Secure Enclave to access any location in the Host System address space

The following figure is a high-level block diagram of the components within the Secure Enclave.

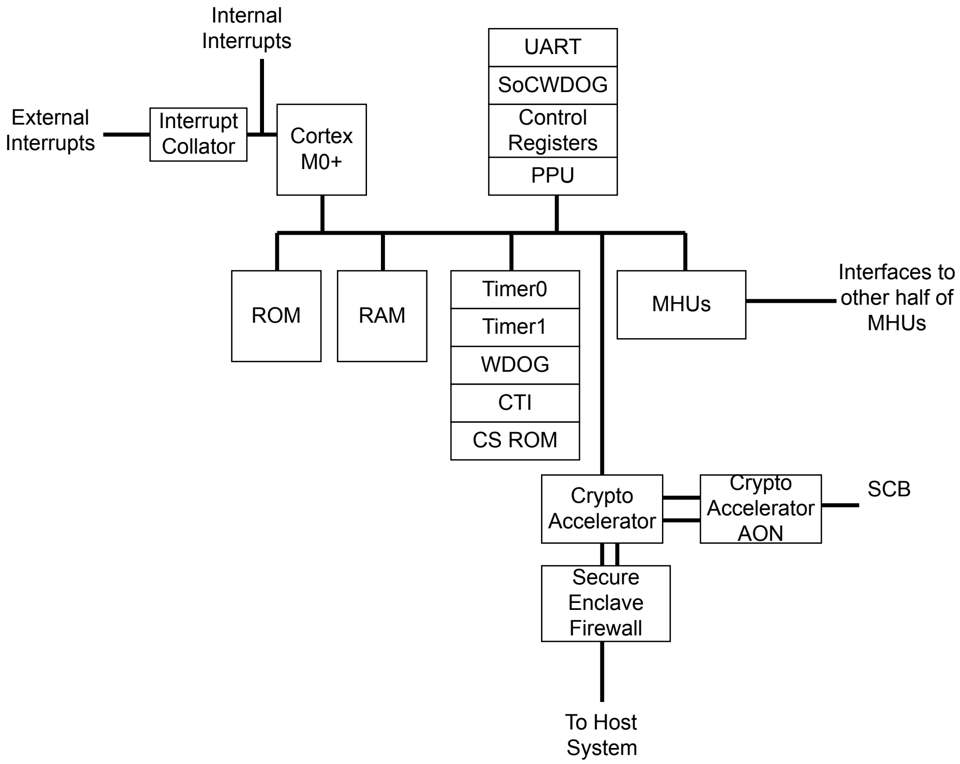


Figure 2-2 Secure Enclave block diagram

Note

The integrator must add a Crypto Accelerator to the Corstone SSE-700 SE subsystem. Therefore the algorithms supported by the Crypto Accelerator are IMPLEMENTATION DEFINED.

To reduce complexity in security reviews, software running on the Corstone SSE-700 SE subsystem is isolated by hardware. MHUs manage the communication between the Host System, External Systems, and the Secure Enclave. For more information about using MHUs, see the Secure Enclave MHU section in the *Arm® Corstone™ SSE-700 Secure Enclave Technical Reference Manual*.

The Secure Enclave consists of two sections:

- An always-on section, which resides in the AONTOP power domain of Corstone SSE-700 subsystem. This contains the following components:
  - SoC Watchdog
  - Secure Enclave UART
  - SECENCTOP PPU and PCSM
  - Crypto Accelerator Always-on
  - Secure Enclave System and Base System Control
- A switchable part, which resides in the SECENCTOP power domain of the Corstone SSE-700 subsystem. For more information see the SECENCTOP power domain section in the *Arm® Corstone™ SSE-700 Secure Enclave Technical Reference Manual*.

2.3 SSE-200 subsystem

This section gives an overview of the product and its features.

For more information, see the SSE-200 documentation set:

- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Overview*
- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual*
- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Configuration and Integration Manual*

This section contains the following subsections:

- *2.3.1 About SSE-200* on page 2-26.
- *2.3.2 Features of SSE-200* on page 2-27.

2.3.1 About SSE-200

SSE-200 is a collection of pre-assembled elements to use as the basis of an *Internet of Things* (IoT) *System on Chip* (SoC).

It is complemented by software libraries that are integrated with the Mbed operating system. The SSE-200 is part of the Corstone foundation IP which also contains the CoreLink SIE-200 product and other components. SSE-200 provides components to quickly create systems that are based on Cortex-M33 processors.

The following figure shows the major blocks present in SSE-200.

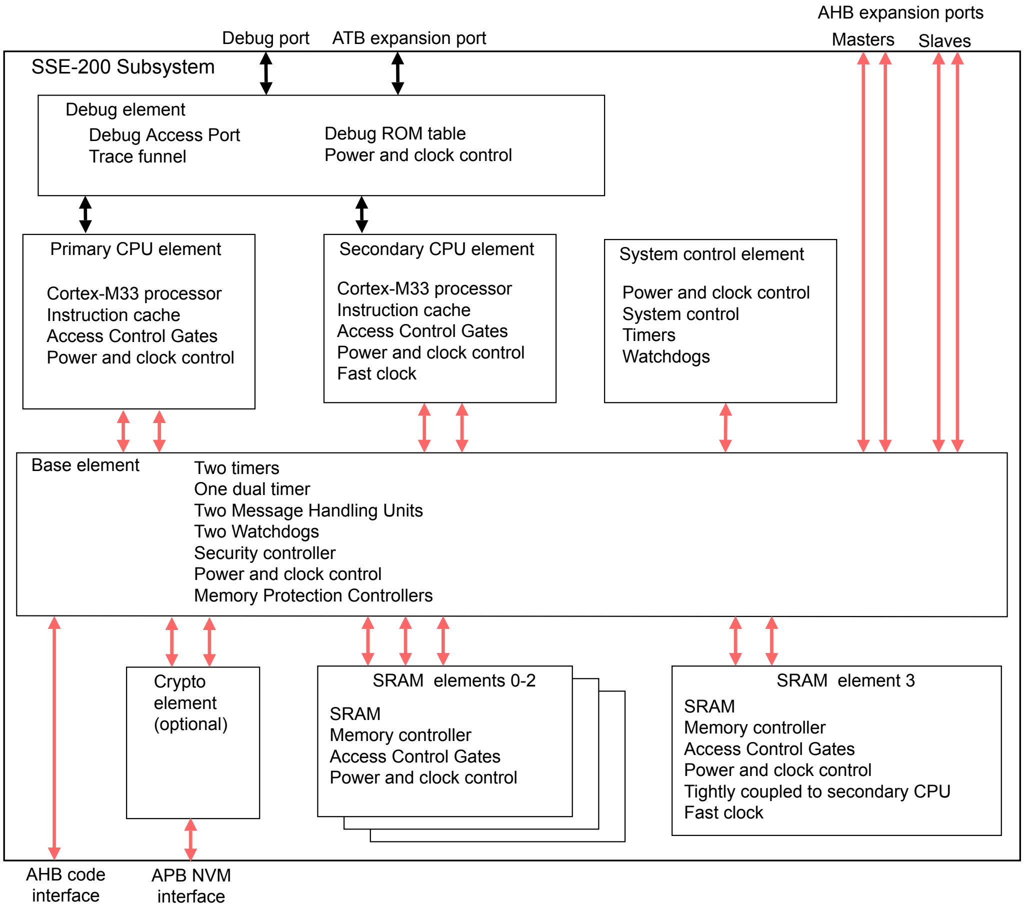


Figure 2-3 SSE-200 subsystem elements

2.3.2 Features of SSE-200

The SSE-200 contains the following components:

- Two Cortex-M33 processors:
  - Optional *Floating-Point Unit* (FPU) and *Digital Signal Processor* (DSP) extensions (configurable)
  - *Embedded Trace Macrocell* (ETM)

  For more information, see the *Arm Cortex-M33 Processor Technical Reference Manual*.
- CoreSight debug system with configurable Secure Debug and Trace.
- Secure AMBA interconnect:
  - *Advanced High Performance Bus* (AHB5) Bus Matrix
  - AHB5 TrustZone *Memory Protection Controller* (MPC)
  - AHB5 TrustZone *Peripheral Protection Controller* (PPC)
  - AHB5 *Exclusive Access Monitor* (EAM)
  - AHB5 *Access Control Gates* (ACG)
  - AHB5 to *Advanced Peripheral Bus* (APB) Bridges
  - Expansion AHB5 master and slave buses (two each)
- Memory system:
  - AHB5 master bus to external code memory
  - Static memory controllers
  - Multiple banks of SRAM. One bank of SRAM functions as *Tightly Coupled Memory* (TCM)
  - Instruction caches
- Security components:
  - CryptoCell-312 (optional)
  - *Implementation Defined Attribution Unit* (IDAU)
  - Secure expansion ports
  - System Security Controller
  - System Controller
- APB peripherals with security support:
  - Three general-purpose timers with configurable security. One timer is on the 32KHz domain and two are on the SYSCLK PD_SYS domain.
  - A *Cortex-M System Design Kit* (CMSDK) dual timer with configurable security
  - Three Watchdog timers with fixed security. One Secure watchdog is on the 32KHz domain and one Secure and one Non-Secure is on the SYSCLK PD_SYS domain.
  - Two *Message Handling Units* (MHUs) allow software to raise interrupts and facilitate cross processor messaging
- Power-control components:
  - *Power Dependency Control Matrix* (PDCM)
  - *Power Policy Units* (PPU)
  - CoreLink LPD-500 Low Power Distributor
  - Wakeup on interrupt from *External Wakeup Controllers* (EWC) and *Wakeup Interrupt Controllers* (WIC)

2.4 SSE-123 example subsystem

This section gives an overview of the product and its features.

For more information, see the SSE-123 documentation set:

- *Arm® SSE-123 Example Subsystem Technical Overview*
- *Arm® SSE-123 Example Subsystem Technical Reference Manual*
- *Arm® SSE-123 Example Subsystem Configuration and Integration Manual*

This section contains the following subsections:

- *2.4.1 About the SSE-123 example subsystem* on page 2-28.
- *2.4.2 Features of SSE-123* on page 2-29.

2.4.1 About the SSE-123 example subsystem

The SSE-123 integrates a subsystem of key Arm components that implement core functionality of a system targeting *Internet of Things* (IoT) *System on Chip* (SoC) designs.

The subsystem can be implemented as a standalone single core system or as part of a multiprocessor system.

The following figure shows a block diagram of the SSE-123.

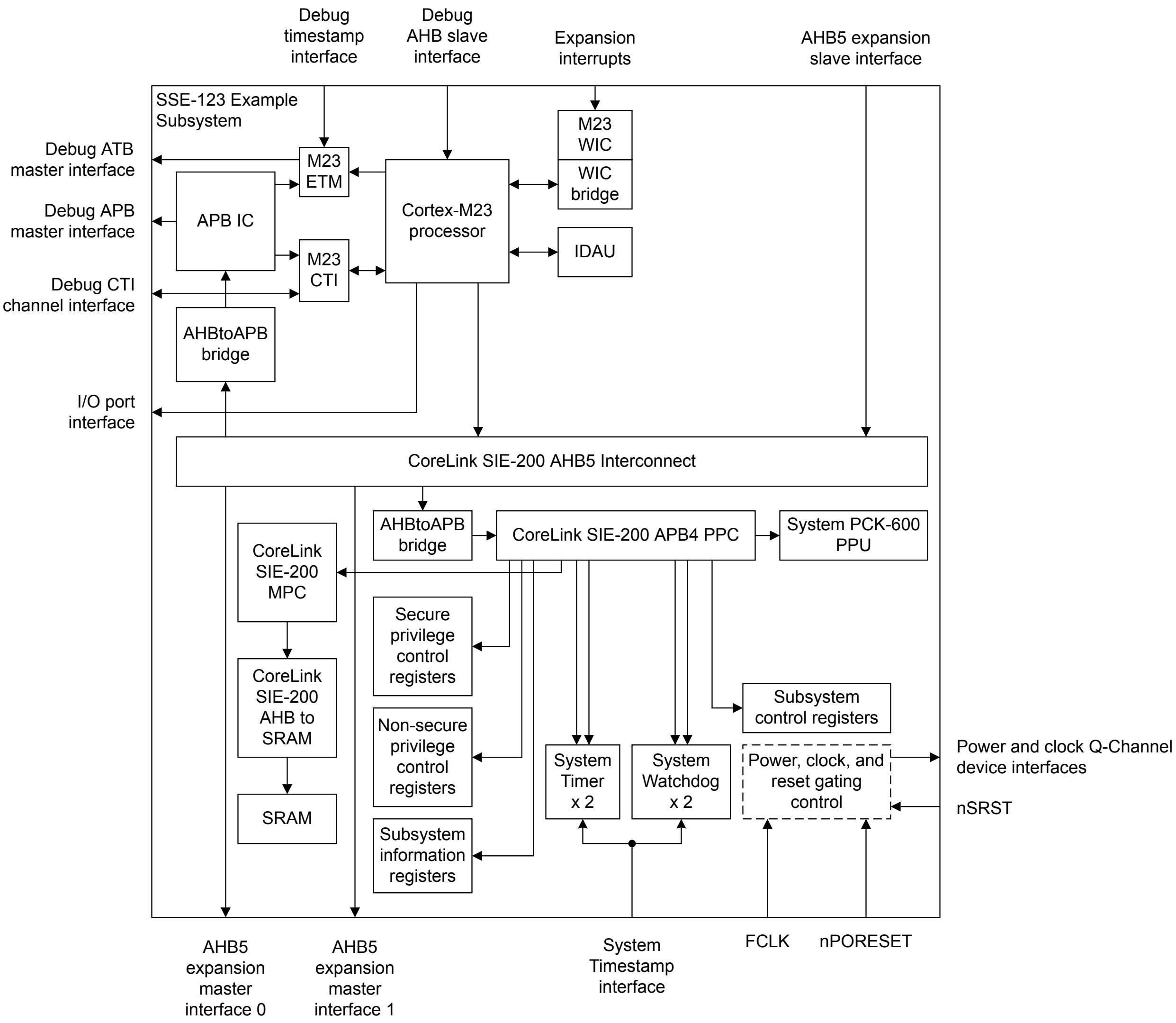

Debug timestamp interface
Debug AHB slave interface
Expansion interrupts
AHB5 expansion slave interface
SSE-123 Example Subsystem
Debug ATB master interface
Debug APB master interface
Debug CTI channel interface
I/O port interface
M23 ETM
APB IC
M23 CTI
Cortex-M23 processor
M23 WIC
WIC bridge
IDAU
AHBtoAPB bridge
CoreLink SIE-200 AHB5 Interconnect
AHBtoAPB bridge
CoreLink SIE-200 APB4 PPC
System PCK-600 PPU
CoreLink SIE-200 MPC
CoreLink SIE-200 AHB to SRAM
SRAM
Secure privilege control registers
Non-secure privilege control registers
Subsystem information registers
System Timer x 2
System Watchdog x 2
Subsystem control registers
Power, clock, and reset gating control
Power and clock Q-Channel device interfaces
nSRST
AHB5 expansion master interface 0
AHB5 expansion master interface 1
System Timestamp interface
FCLK
nPORESET

Figure 2-4 SSE-123 block diagram

The block diagram shows all the key integrated components and interfaces.

2.4.2 Features of SSE-123

The SSE-123 provides the following features:

- A Cortex-M23 processor, including Armv8-M Security Extensions
- A single bank of system SRAM
- CoreLink SIE-200 System IP for Embedded:
  - — AHB5 bus matrix
  - — *Memory Protection Controller* (MPC)
  - — *Peripheral Protection Controller* (PPC)
  - — AHB5 to APB4 bridge
  - — AHB5 to SRAM controller
- CoreLink PCK-600 Power Control Kit:
  - — *Power Policy Unit* (PPU)
  - — Clock controller
  - — Low-Power Distributor Q-Channel (LPD-Q)
- *Implementation Defined Attribution Unit* (IDAU)

- Cortex-M23 processor *Wakeup Interrupt Controller* (WIC)
- System Timer and Watchdog
- System Control and Security Control Registers
- Optional Cortex-M23 processor Debug components:
  - *Embedded Trace Macrocell* (ETM)
  - *Cross Trigger Interface* (CTI)
  - Debug APB interconnect

2.5 SSE-050 subsystem

This section gives an overview of the product and its features.

For more information, see the SSE-050 documentation set:

- *Arm® CoreLink™ SSE-050 Subsystem Technical Reference Manual*
- *Arm® CoreLink™ SSE-050 Subsystem Configuration and Integration Manual*

This section contains the following subsections:

- *2.5.1 About IoT endpoints* on page 2-31.
- *2.5.2 Features* on page 2-33.

2.5.1 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.

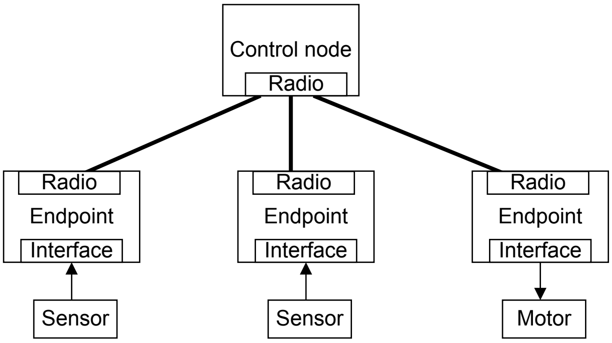


Figure 2-5 An IoT endpoint as part of a larger control system

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

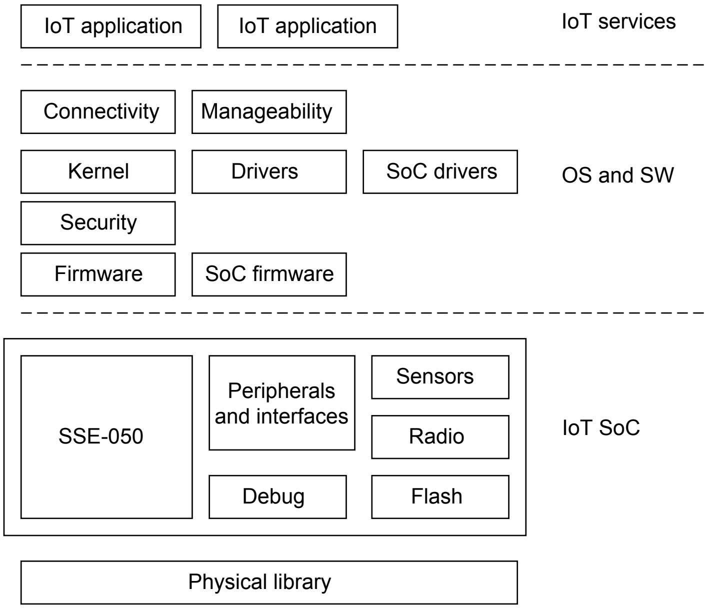


Figure 2-6 IoT endpoint HW and SW solution

A complete endpoint system typically contains the following components:

Compute subsystem

The SSE-050 consists of the CoreLink 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.

The Arm Cordio® BT4 radio IP is available as an option for the SSE-050. The subsystem expansion ports are however technology independent and other radio devices could be used instead of the Cordio radio IP. 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 the Arm Mbed operating system, 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.

2.5.2 Features

The SSE-050 contains the following components:

- A CoreLink 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® CoreLink™ 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:
  - Placeholder for embedded flash controller and optionally cache

    Note

    The SSE-050 can support the integration of any 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
  - 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
  - *True Random Number Generator* (TRNG)
  - *Real Time Clock* (RTC)
- Optional radio solution integration capability:
  - AHB master and slave ports
  - Reserved interrupt ports

Note

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 rich OS. 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.

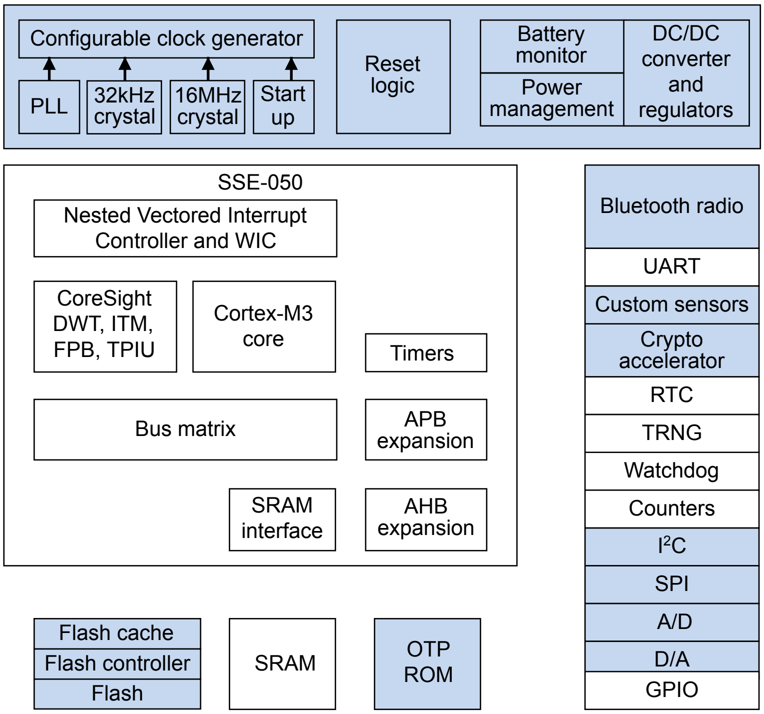


Figure 2-7 Example of an IoT endpoint SoC

2.6 Cortex-M System Design Kit

This section gives an overview of the product and its features.

For more information, see the CMSDK documentation set:

- *Arm® Cortex®-M System Design Kit Technical Reference Manual*
- *Arm® Cortex®-M System Design Kit Example System Guide*
- *Arm® Cortex®-M0 and Cortex®-M0+ System Design Kit Example System Guide*

This section contains the following subsection:

- *2.6.1 About the Cortex-M System Design Kit* on page 2-35.

2.6.1 About the Cortex-M System Design Kit

The Cortex-M System Design Kit helps you design products using Arm Cortex-M processors.

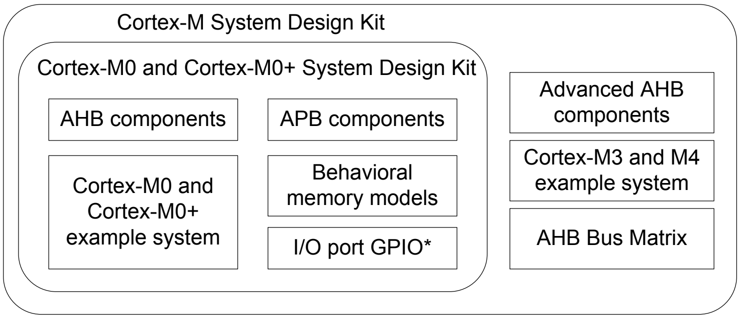
The design kit contains the following:

- A selection of AHB-Lite and APB components, including several peripherals such as GPIO, timers, watchdog, and UART
- Example systems for the Cortex-M0, Cortex-M0+, Cortex-M3, and Cortex-M4 cores
- Example synthesis scripts for the example systems
- Example compilation and simulation scripts for the Verilog environment that supports ModelSim, VCS, and NC Verilog
- Example code for software drivers
- Example test code to demonstrate various operations of the systems
- Example compilation scripts and example software project files that support:
  — Arm Development Studio 5 (DS-5)
  — Arm RealView Development Suite
  — Keil® *Microcontroller Development Kit* (MDK)
  — GNU tools for Arm embedded processors (Arm GCC).

The Cortex-M System Design Kit is available as:

- Cortex-M0 and Cortex-M0+ System Design Kit. This supports Cortex-M0 and Cortex-M0+.
- Cortex-M System Design Kit, full version. This supports Cortex-M0, Cortex-M0+, Cortex-M3, and Cortex-M4.

The other differences between the Cortex-M0 and Cortex-M0+ version, and the Cortex-M version of the design kit are the example systems, and the components provided. See the following figure.


Cortex-M System Design Kit
Cortex-M0 and Cortex-M0+ System Design Kit
AHB components
APB components
Cortex-M0 and Cortex-M0+ example system
Behavioral memory models
I/O port GPIO*
Advanced AHB components
Cortex-M3 and M4 example system
AHB Bus Matrix

* For use with the Cortex-M0+ directly, or as a subcomponent within AHB GPIO module.

2.7 Cortex-M0+ Processor

This section gives an overview of the product and its features.

For more information, see:

- *Arm® Cortex®-M0 and Cortex®-M0+ System Design Kit Example System Guide* (DUI 0559)

This section contains the following subsection:

- *2.7.1 About the Cortex-M0+ processor* on page 2-36.

2.7.1 About the Cortex-M0+ processor

The Cortex-M0+ processor is a very low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications that require an area optimized, low-power processor.

The processor features and benefits are:

- Tight integration of system peripherals reduces area and development costs
- Thumb instruction set combines high code density with 32-bit performance
- Support for single-cycle I/O access
- Power control optimization of system components
- Integrated sleep modes for low power consumption
- Fast code execution enables running the processor with a slower clock or increasing sleep mode time
- Optimized code fetching for reduced flash and ROM power consumption
- Hardware multiplier
- Deterministic, high-performance interrupt handling for time-critical applications
- Deterministic instruction cycle timing
- Support for system level debug authentication
- Serial Wire Debug reduces the number of pins required for debugging
- Support for optional instruction trace

2.8 SIE-200 System IP for Embedded

This section gives an overview of the product and its features.

For more information, see the SIE-200 documentation set:

- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*
- *Arm® CoreLink™ SIE-200 System IP for Embedded Configuration and Integration Manual*

This section contains the following subsections:

- *2.8.1 About SIE-200* on page 2-37.
- *2.8.2 Features of SIE-200* on page 2-37.

2.8.1 About SIE-200

The CoreLink SIE-200 System IP for Embedded product is a collection of interconnect, peripheral, and TrustZone controller components for use with a processor that complies with the Armv8-M processor architecture.

Bus architecture

SIE-200 supports the following bus protocols:

- AMBA 5 AHB5 Protocol
- AMBA 4 APB4 Protocol
- AMBA 3 APB3 Protocol
- AMBA 3 AHB-Lite Protocol

Bus naming convention

It is important to always view each AMBA point-to-point connection as a master to slave connection. To distinguish between external AMBA masters or slaves and the conceptual masters or slaves on the component, masters and slaves on the interconnect are referred to as master ports or slave ports. External masters and slaves are referred to as masters and slaves.

2.8.2 Features of SIE-200

SIE-200 consists of the following components and models that support the AHB5 standard:

- AHB5 system components
- AHB5 bridge components
- TrustZone protection controllers
- Verification components

2.9 NIC-450 Network Interconnect

This section gives an overview of the product and its features.

For more information, see the NIC-450 and associated products documentation sets:

- *Arm CoreLink NIC-450 Network Interconnect Technical Overview*
- *Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual*
- *Arm CoreLink NIC-400 Network Interconnect Integration Manual*
- *Arm CoreLink NIC-400 Network Interconnect Implementation Guide*
- *Arm CoreLink QoS-400 Network Interconnect Advanced Quality of Service Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual*
- *Arm CoreLink QVN-400 Network Interconnect Advanced QoS for Virtual Networks Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual*
- *Arm CoreLink TLX-400 Network Interconnect Thin Links Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual*
- *Arm CoreLink ADB-400 AMBA Domain Bridge User Guide*
- *Arm CoreLink AXI4 to AHB-Lite XHB-400 Bridge Technical Reference Manual*

This section contains the following subsection:

- *2.9.1 About NIC-450* on page 2-38.

2.9.1 About NIC-450

Arm NIC-450 is a library of highly configurable and multi-power domain tools.

NIC-450 is a library of key interconnect IP that enables you to build a scalable and configurable network interconnect. NIC-450 includes:

NIC-400 Network Interconnect
NIC-400 is a cascading, routing interconnect component. NIC-400 is a hierarchical, low latency, and low-power connection for various other components.

QoS-400 Network Interconnect Advanced Quality of Service
QoS-400 provides programmable QoS facilities for any attached masters.

QVN-400 Advanced Quality of Service for Virtual Networks
QVN-400 provides a mechanism to avoid head-of-line blocking and cross-path blocking between different data flows.

DPE-400 Data Parity Extension

TLX-400 Network Interconnect Thin Links
TLX-400 provides a mechanism to reduce the number of signals in an AXI point-to-point connection and enable it to be routed over a longer distance.

ADB-400 AMBA Domain Bridge
ADB-400is an asynchronous bridge between two components or systems that can be in a different power, clock, or voltage domains.

AXI4 to AHB-Lite XHB-400 Bridge
XHB-400 converts AXI4 protocol to AHB-Lite protocol, and has an AXI4 slave interface and an AHB-Lite master interface.

LPD-500 Low Power Distributor
LPD-500 is a standalone configurable component to distribute Q-Channel interfaces to multiple devices and subsystems.

You can integrate the NIC-400 with the ADB-400 AMBA Domain Bridge or TLX-400 Network Interconnect Thin Links bridges into a single interconnect. You can utilize the high level of configurability of NIC-450 for optimization and tuning.

The benefits of using the NIC-450 are:

- Unified low-power interfaces when applicable
- Single design environment to configure IP blocks and connect them together

Use NIC-450 with Socrates™, a tool that employs algorithms to aid the creation of valid configurations that are based on your specific design requirements.

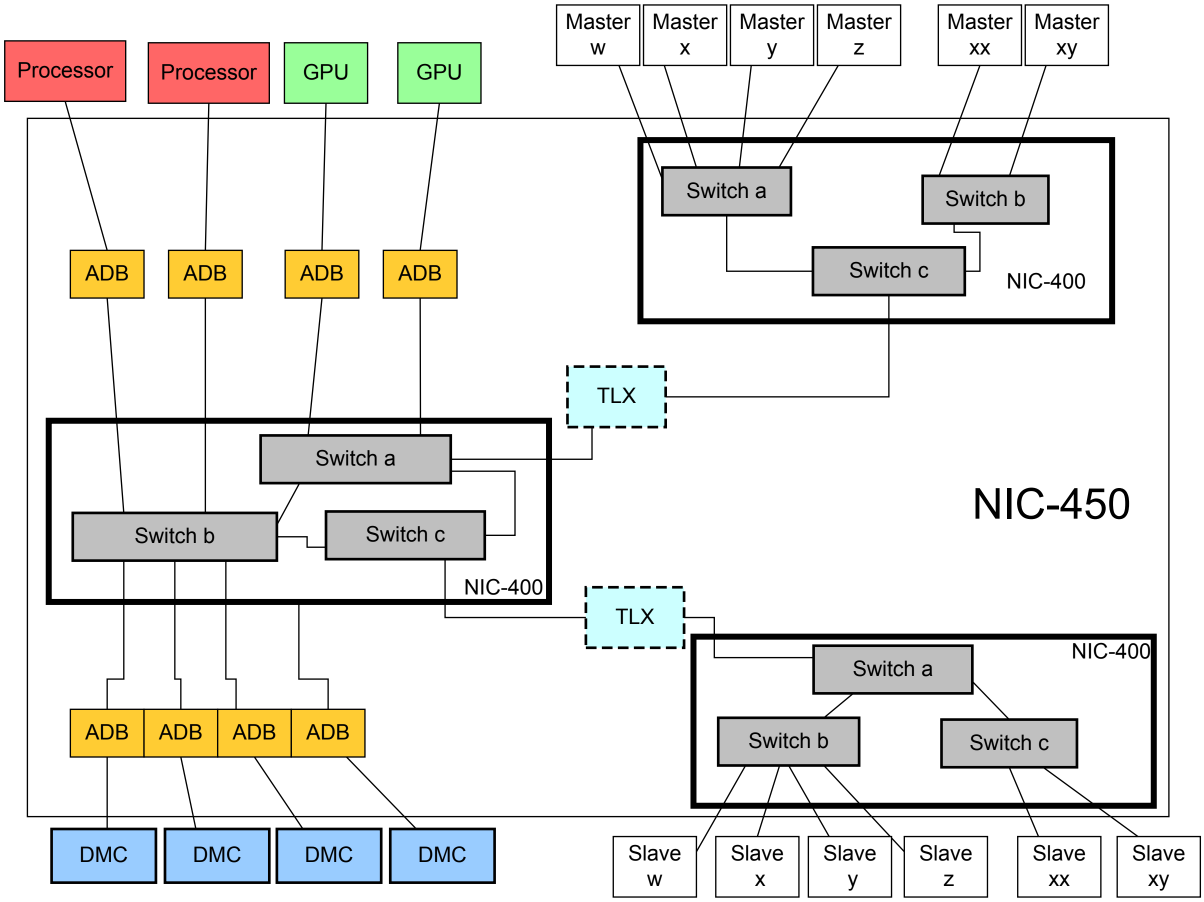


Figure 2-8 NIC-450 block diagram

2.10 SoC-600

This section gives an overview of the product and its features.

For more information, see the SoC-600 documentation set:

- *Arm® CoreSight™ System-on-Chip SoC-600 Technical Reference Manual*
- *Arm® CoreSight™ System-on-Chip SoC-600 Configuration and Integration Manual*

This section contains the following subsections:

- *2.10.1 About this product* on page 2-40.
- *2.10.2 Features* on page 2-40.

2.10.1 About this product

CoreSight SoC-600 is a member of the Arm embedded debug and trace component family that is based on the Cortex-M processor.

CoreSight SOC-600M is a subset of CoreSight SOC-600. Therefore, component names and filepaths refer to `css600` not `css600m`.

Some of the features that CoreSight SoC-600 provides are:

- Components that can be used for debug and trace of Arm SoCs. These SoCs can be simple single-processor designs to complex multiprocessor and multi-cluster designs that include many heterogeneous processors.
- Support for the *Arm® Debug Interface* (ADI) v6 and CoreSight v3 Architectures that enable you to build debug and trace functionality into your systems. It supports debug and trace over existing functional interfaces.
- Components that support the development of low-power system implementations through architected fine-grained power control.
- Q-Channel interfaces for clock and power quiescence.
- Can be integrated with the Arm CoreLink LPD-500 as part of a full-chip power and clock control methodology.
- The Arm CoreSight SDC-600 can be integrated with CoreSight SoC-600, with an applicable licence, as part of a certificate-based authenticated debug solution.

The CoreSight SoC-600 bundle includes:

- A library of configurable CoreSight components that are written in Verilog, and that are compliant with the *Verilog-2001 Standard* (IEEE Std 1364-2001).
- Example timing constraint files for each component in SDC format.

Note

- CoreSight was previously configurable in Socrates System Builder.
- Socrates System Builder is now in maintenance. There will be no functionality updates to Socrates System Builder, although there might be maintenance updates.
- The new IP Tooling platform, Socrates, enables configuration and build of individual CoreSight components. However, there is no CoreSight creation flow, and no system-stitching capability.
- Socrates functionality can be enabled using a legacy Socrates System Builder license.

2.10.2 Features

Features and capabilities that the SoC-600 provides include:

Debug

- *Arm® Debug Interface Architecture Specification ADIv6.0*-compliant debug port. This debug port supports JTAG and Serial Wire protocols for connection to an off-chip debugger. This connection is achieved using a low-pin-count connection that is suitable for bare-metal debug and silicon bring-up.
- *Arm® CoreSight™ Architecture Specification v3.0* compliance enables debug over functional interfaces, suitable for application development and in-field debug without a dedicated debug interface.
- Infrastructure components supporting system identification and integration with other CoreSight IP.

Trace

- Versatile *Trace Memory Controller* (TMC) supporting local on-chip storage, and buffering of trace data.
- TMC router configuration supports efficient hand-off of trace data to other system masters. This feature enables trace over functional interfaces, suitable for application development and in-field debug without a dedicated debug and trace interface.
- TMC streaming configuration supports integration to third-party *High Speed Serial Trace Port*s (HSSTP) for high bandwidth, low pin count trace solutions.
- Infrastructure components supporting filtering and routing of trace data on chip.

Trace

- Versatile *Trace Memory Controller* (TMC) supporting local on-chip storage, and buffering of trace data.
- Infrastructure components supporting filtering and routing of trace data on chip.

Embedded Cross Triggering

- *Cross Trigger Interface* (CTI) supports up to 32 trigger inputs and outputs with a single component instance.
- *Cross Trigger Matrix* (CTM) supports up to 33 CTI or CTM connections without cascading.

Power

- *Arm® CoreSight™ Architecture Specification v3.0*-compliant *Granular Power Requester* (GPR) enables fine-grained debug and system power control at all levels of debug hierarchy.
- Components are designed for low-power implementation, supporting clock and power quiescence and wakeup signaling where necessary.
- Components support Q-Channel *Low-Power Interfaces* (LPI) for integration with power controllers to support system-level clock and power gating where necessary.
- Infrastructure components support implementation across multiple clock and power domains.

Miscellaneous

- Some components, such as the bridges and *Serial Wire Debug Port* (SW-DP), use two Verilog modules to span clock and power domains. This design can ease implementation in complex SoC designs that have multiple clock and power domains.
- Infrastructure components support integration with legacy IP including *Arm® CoreSight™ Architecture Specification v2.0*-compliant, and JTAG components.

2.11 SoC-400 and SoC-400M

This section gives an overview of the product and its features. The SoC-400M license enables you to use SoC-400 functionality with Cortex-M cores.

Note

This IP is included to enable full rendering of SSE-200, because the debug and trace infrastructure is built with components from this IP. SSE-700 debug and trace is built using SoC-600.

For more information, see the SoC-400 documentation set:

- *Arm® CoreSight™ SoC-400 Technical Reference Manual*
- *Arm® CoreSight™ SoC-400 User Guide*
- *Arm® CoreSight™ SoC-400 System Design Guide*
- *Arm® CoreSight™ SoC-400 Implementation Guide*
- *Arm® CoreSight™ SoC-400 Integration Manual*

This section contains the following subsections:

- *2.11.1 About SoC-400* on page 2-42.
- *2.11.2 Features* on page 2-42.

2.11.1 About SoC-400

SoC-400 is a solution for debug and trace of complex SoCs.

SoC-400 includes:

- A library of configurable CoreSight components, written in Verilog, and scripts to render configured instances of the CoreSight components based on your parameter choices.
- An optional flow to graphically configure, integrate, and stitch the supplied components and Arm processors using IP Tooling and supplied IP-XACT component views.
- Support for the *System Trace Macrocell* (STM) and *Trace Memory Controller* (TMC), which are licensed separately.

2.11.2 Features

The SoC-400 provides many features to enable rapid and efficient debugging.

Some of the features provided by SoC-400 are:

- Access to debug features and on-chip AXI, AHB, APB, and JTAG buses through a JTAG or *Serial Wire Debug* (SWD) interface
- Merging of multiple trace sources into a single trace stream
- Configurable trace bus widths between 8 bits and 128 bits, with upsizing and downsizing between different widths
- Capture of trace streams on-chip or off-chip
- Cross-triggering between different debug and trace components
- Timestamp generation and system-wide compressed timestamp distribution, including local interpolation to provide local high-resolution timestamps synchronized to a global low-resolution timestamp
- Support for inserting synchronous and asynchronous clock domain boundaries and power domain boundaries across internal interfaces
- Improved configurability of components to better optimize area and power consumption
- Integration with supported Arm processors
- Integration of STM and TMC, licensed separately
- IP-XACT views of all components, defining interfaces, signals, configurability, and programmers models
- Power intent for all components in *Unified Power Format* (UPF), including definitions of how signals must be clamped when parts of the system are powered down
- Synthesis flow

- Flow to verify correct CoreSight system integration
- Optional support for IP Tooling, enabling graphical component configuration, system stitching, and verification
- Full compliance with the CoreSight architecture, enabling integration of third-party IP and comprehensive tools support

2.12 SDC-600 Secure Debug Channel

This section gives an overview of the product and its features.

For more information, see the SDC-600 documentation set:

- *Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual*
- *Arm® CoreSight™ SDC-600 Secure Debug Channel Configuration and Integration Manual*

This section contains the following subsection:

- *2.12.1 About SDC-600* on page 2-44.

2.12.1 About SDC-600

Arm CoreSight SDC-600 provides a dedicated channel for authentication between an external debugger and a debug target platform by using an unlocking mechanism.

The SDC-600-based architecture provides an interface through which secure debug certificates can be injected to the platform. This is done in a standard way through the *Debug Access Port* (DAP), which is normally used to debug the platform. It eliminates the need for OEM proprietary delivery mechanisms for such certificates.

SDC-600 performs the following tasks:

- Requests power and optionally reboots the servicing agent.
- Establishes and maintains a link between a port on the external side, which is serviced by the debugger, and a port on the internal side, which is serviced by an agent on the target system.
- Transports messages from an external debugger to a hardware or software agent on a target system through a point-to-point link.

The debugged target and the servicing agent are typically the same processor or processor subsystem, but they can be separate entities as well.

The authentication process can involve a hardware- or software-based cryptographic engine on the target. The cryptographic engine verifies the debug certificate that is passed to the servicing agent through the SDC-600. The debugger and the servicing agent run a protocol on top of the SDC-600, which:

1. Identifies the SoC (SoC_ID).
2. Injects the appropriate debug certificate to the debug target for processing by the cryptographic engine.

The following is a high-level description of a sample authentication process:

1. The debugger wants to access the target's debug resources.
2. The debugger uses the CoreSight ID registers and discovery process to identify the SDC-600's external interface.
3. The debugger accesses the SDC-600 to start the unlocking process.
4. The SDC-600 requests the powerup of the rest of its functional blocks.
5. The debugger asks for a SoC_ID from the servicing target to identify the target system.
6. A certificate is generated by the debugger for the SoC_ID that is transmitted to the servicing target.
7. The servicing agent decides whether the debugger has the rights to access the debug target based on the provided certificate.
8. If access is granted, the target agent drives the authentication signals accordingly on the Access Ports so that the connected devices can be accessed by the debugger.

2.13 STM-500 System Trace Macrocell

This section gives an overview of the product and its features.

For more information, see the STM-500 documentation set:

- *Arm® CoreSight™ STM-500 System Trace Macrocell Technical Reference Manual*

This section contains the following subsections:

- *2.13.1 About STM-500* on page 2-45.
- *2.13.2 Features of STM-500* on page 2-46.

2.13.1 About STM-500

STM-500 is a trace source that is integrated into a CoreSight system, and is designed primarily for high-bandwidth trace of instrumentation embedded into software. This instrumentation is made up of memory-mapped writes to the STM Advanced eXtensible Interface (AXI) slave, which carry information about the behavior of the software.

STM-500 is a natural successor to the CoreSight *Instrumentation Trace Macrocell* (ITM) in mid- to high-performance applications. The STM provides the following advantages over the ITM for software instrumentation:

- It has a dedicated AXI slave interface for receiving the instrumentation information. The AXI slave interface is in addition to the *Advanced Peripheral Bus* (APB) interface that you can use for programming the STM-500 registers. The AXI slave interface has significantly higher performance than the APB interface of the ITM.
- Multiple processors and processes can share and directly access STM-500 without being aware of each other, by being allocated different pages in the STM stimulus space. 128 masters, each supporting 65,536 stimulus ports. Each 4KB page of the STM stimulus space provides 16 stimulus ports. Stimulus ports are also known as channels.
- STM-500 can optionally stall the AXI when its FIFO becomes full, ensuring that no data is lost because of overflow, without having to poll the FIFO status in software. This behavior depends on the address written to, and can therefore be controlled by each stimulus port independently.
- An improved, configurable FIFO, supporting up to 32 transactions, reduces the likelihood of the FIFO becoming full.
- Timestamping can be requested for each write independently, based on the address written to. You can also optimize the bandwidth by requesting a timestamp for only one write transaction in a message made up of several writes.
- Timestamps are automatically correlated with other timestamping trace sources in the CoreSight system, enabling automatic correlation with, for example, processor execution trace.

In addition to the AXI slave, STM-500 provides a hardware event interface. STM-500 generates trace when signals are asserted on this interface. Alternatively, you can implement advanced custom system tracing features by generating AXI write accesses directly to the AXI slave.

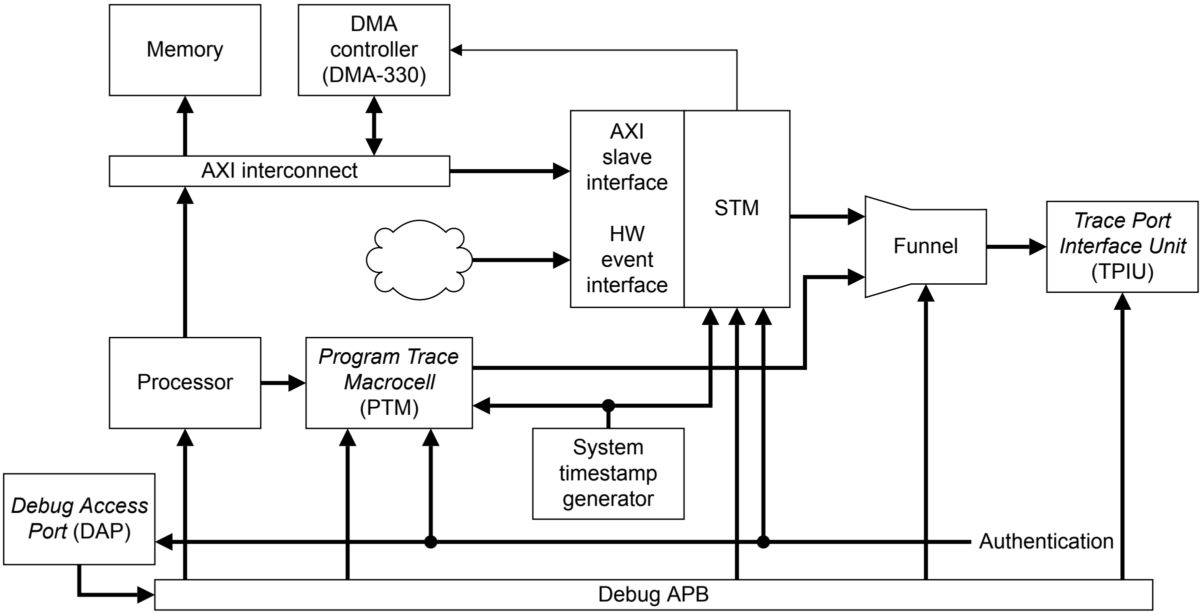


Figure 2-9 System integration

STM-500 AXI slave is connected to a system interconnect that enables all system masters, such as processors and *Direct Memory Access* (DMA) controllers, to generate trace by writing to the STM-500 stimulus ports.

For interaction with DMA controllers, STM-500 provides a DMA request interface that is compatible with the AMBA DMA Controller DMA-330.

For configuration purposes, STM-500 is connected to a Debug APB interconnect. This enables off-chip and on-chip debug agents to access STM-500.

STM-500 uses CoreSight authentication signals to control debug permissions.

The STM-500 trace stream is output through the *Advanced Trace Bus* (ATB) interface and it is integrated with the rest of the CoreSight trace infrastructure.

2.13.2 Features of STM-500

STM-500 has the following features:

- A fully synchronous design with one clock and two resets

  One 64-bit AXI slave interface for extended stimulus port inputs
- One hardware event observation interface for tracing 64 hardware events
- One 32-bit debug APB slave interface for configuration and status
- One 64-bit ATB master interface for trace output

  One DMA peripheral request interface that is compatible with the AMBA DMA Controller DMA-330

  Two depth-configurable FIFO buffers for usage-optimized configurability:
  — Data FIFO
  — Channel information FIFO
- A fully memory-mapped software stimulus supporting 65,536 stimulus ports and 128 masters
- Leading zero data compression
- Full support for guaranteed and invariant timing software stimulus writes
- Support for single-shot and multi-shot triggers with a cross-trigger port, trigger packet insertion, and ATB trace triggers
- An internal and an external source for STPv2 synchronization

- Timestamping of trace events
- Two low-power interfaces

The STM-500 architecture has many IMPLEMENTATION DEFINED options. *Table 2-1 STM-500 configuration* on page 2-47 shows the configuration implemented by STM-500.

Table 2-1 STM-500 configuration

Feature	Configuration
Trace protocol	STPv2
Timestamping	Absolute
STMTSFREQR	Read-write
STMTSSTIMR	Implemented
STMSYNCR	Implemented
Claim tags	Four
TRACEID	CoreSight ATB and ATB trigger
Trigger control	Multi-shot and single-shot
STMTCSR.TSPRESCALE	Not implemented
STMTCSR.HWTEN	Not implemented
STMTCSR.SYNCEN	Always reads as `0b1`
STMTCSR.SWOEN	Not implemented
Number of stimulus ports	65536
Number of masters	Minimum of two
Stimulus port types	Extended only
Fundamental data size	64
Transaction Types	Invariant timing and guaranteed
STMSPER	Implemented
STMSPTER	Implemented
STMPRIVMASKR	Not implemented
STMSPOVERRIDER and STMSPMOVERRIDER	Implemented
STMSPSCR and STMSPMSCR	Implemented
Data compression on stimulus ports	Programmable
Hardware event tracing	Implemented
Number of hardware events	64
STMHETER	Implemented
Hardware error detection	Implemented
STMHEMASTR	Read only
Data compression on hardware event trace	Programmable
Hardware event multiplexing	The STMHEEXTMUXR is 8-bits wide

2.14 PCK-600 Power Control Kit

This section gives an overview of the product and its features.

For more information, see the PCK-600 documentation set:

- *Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual*
- *Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual*

This section contains the following subsection:

- *2.14.1 About the Power Control Kit* on page 2-48.

2.14.1 About the Power Control Kit

The PCK-600 provides a set of configurable RTL components for the creation of SoC clock and power control infrastructure. The components use the Arm Q-Channel and P-Channel low power interfaces.

The PCK-600 consists of the following components:

Low Power Distributor Q-Channel (LPD-Q)
The LPD-Q component distributes a Q-Channel from one Q-Channel controller to up to 32 Q-Channel devices.

Low Power Distributor P-Channel (LPD-P)
The LPD-P component distributes a P-Channel from one P-Channel controller to up to 8 P-Channel devices.

Low Power Combiner Q-Channel (LPC-Q)
The LPC-Q component combines the Q-Channels from multiple Q-Channel controllers to multiple Q-Channel devices with common control requirements.

P-Channel to Q-Channel Converter (P2Q)
The P2Q component converts a P-Channel to a Q-Channel.

Clock Controller (CLK-CTRL)
The CLK-CTRL component provides *High-level Clock Gating* (HCG) for a single clock domain.

Power Policy Unit (PPU)
The PPU component is a configurable and programmable P-Channel and Q-Channel power domain controller.

The following figure shows an example system that uses the components to manage three power domains. The components are shown in red and blue.

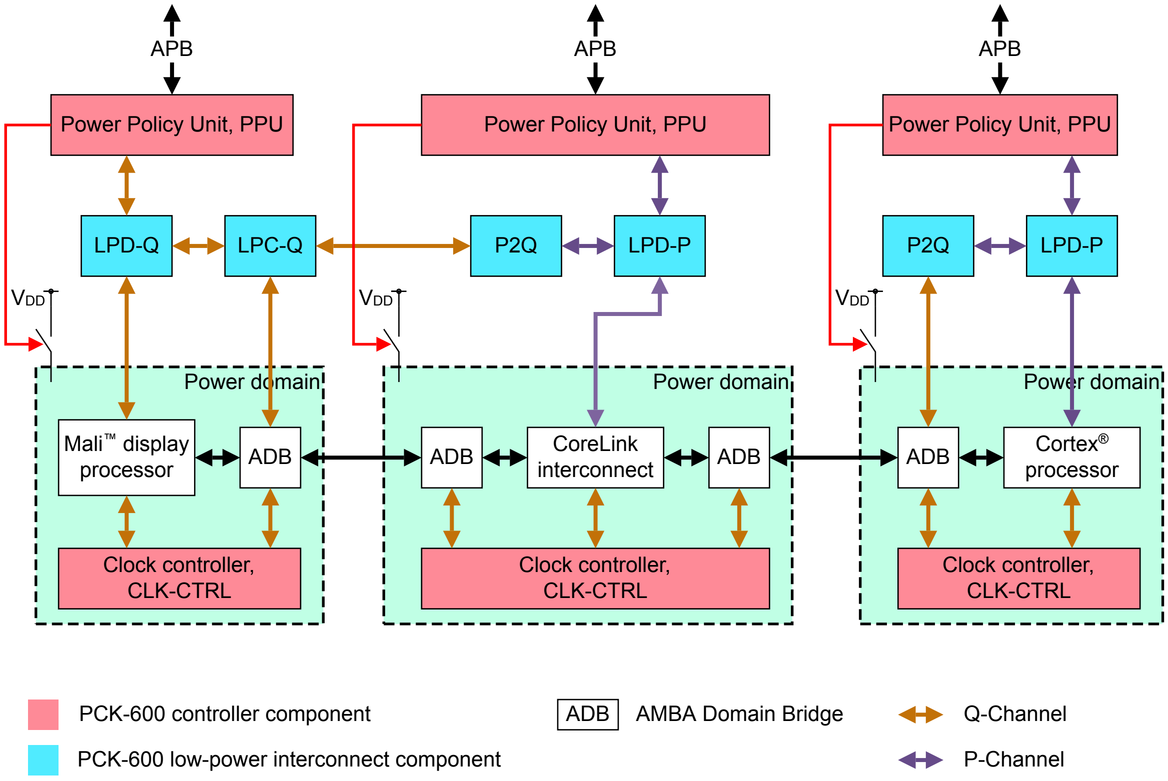


Figure 2-10 Example system that contains PCK-600

2.15 GFC-200 Generic Flash Controller

This section gives an overview of the product and its features.

For more information, see the GFC-200 documentation set:

- *Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual*
- *Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual*

This section contains the following subsections:

- *2.15.1 About the GFC-200* on page 2-50.
- *2.15.2 Features* on page 2-51.

2.15.1 About the GFC-200

The GFC-200 comprises the generic part of a Flash controller in a *System-on-Chip* (SoC). The GFC-200 enables an embedded Flash macro to be integrated easily into any system.

An eFlash macro enables a Flash controller to access eFlash memory. The eFlash macros produced by different foundries and processes can have different interfaces, timings, signal names, protocols, and features that are determined by the foundry processes that produced the eFlash memory.

The GFC-200 provides functions that relate only to services for the system side of the Flash controller. The GFC-200 cannot communicate directly with the eFlash macro. Therefore, the GFC-200 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

The process-specific part of the Flash controller is part of the Flash subsystem in your SoC. It communicates directly with the eFlash macro through a Flash interface.

The GFC-200 supports accesses from two masters that can operate in separate domains such as a Non-secure domain and a Secure domain. Communication between the system and eFlash memory is through a *Generic Flash Bus* (GFB) supplied with GFC-200.

The following figure shows how the GFC-200 is used in a Flash controller implementation.

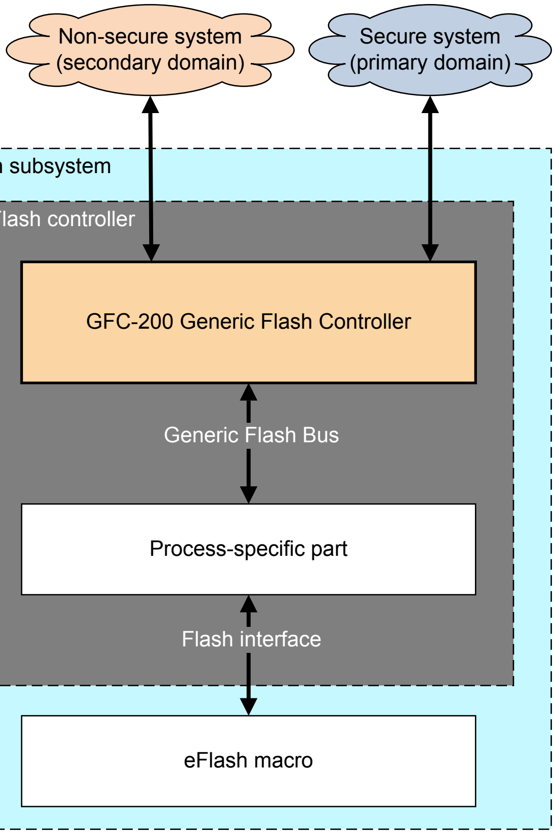


Figure 2-11 GFC-200 in a Flash controller implementation

2.15.2 Features

The GFC-200 provides several interfaces and features.

Flash memory partitioning:

- Ability to divide the available Flash memory space into several partitions and perform access control on a per partition basis
- Dynamically configurable access rights to partitions
- A configuration parameter controls the size of the partitions

AMBA AHB-Lite interface:

- Read-only access to the embedded Flash
- Configurable data width
- Burst support
- Low latency

Primary APB slave interface:

- Write and erase access to the embedded Flash
- Debug read access to the embedded Flash
- Control port for GFC-200 and the eFlash macro
- Interrupt capability for long running commands
- Access to internal registers and the control registers in the process-specific part

Secondary APB slave interface:

- Write and erase access to the embedded Flash
- Debug read access to the embedded Flash
- Control port for GFC-200

- Interrupt capability for long running commands
- Access to internal registers

APB register master interface:

- Enables access to the registers in the process-specific part

Q-Channel interface:

- Control port for system power
- Control port for the system clock

P-Channel controller interface:

- Control port for power to the process-specific part

Generic Flash Bus (GFB):

- Enables GFC-200 accesses to embedded Flash
- Simple command-based protocol
- Synchronous with the AHB clock
- Simplifies communication between GFC-200 and the attached process-specific part

2.16 GFC-100 Generic Flash Controller

This section gives an overview of the product and its features.

For more information, see the GFC-100 documentation set:

- *Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual*
- *Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual*

This section contains the following subsections:

- *2.16.1 About GFC-100* on page 2-53.
- *2.16.2 Features* on page 2-54.

2.16.1 About GFC-100

The GFC-100 comprises the generic part of a Flash controller in a *System-on-Chip* (SoC). GFC-100 enables an embedded Flash macro to be integrated easily into any system.

An eFlash macro enables a Flash controller to access eFlash memory. The eFlash macros produced by different foundries and processes can have different interfaces, timings, signal names, protocols and features that are determined by the foundry processes that produced the eFlash memory.

GFC-100 provides the functions that relate only to services for the system side of the Flash controller. GFC-100 cannot communicate directly with the eFlash macro. Therefore, GFC-100 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

The process-specific part of the Flash controller is part of the Flash subsystem in your SoC. It communicates directly with the eFlash macro through a Flash interface.

Communication between the system and eFlash memory is through a *Generic Flash Bus* (GFB) supplied with GFC-100.

The following figure shows how GFC-100 is used in a Flash controller implementation.

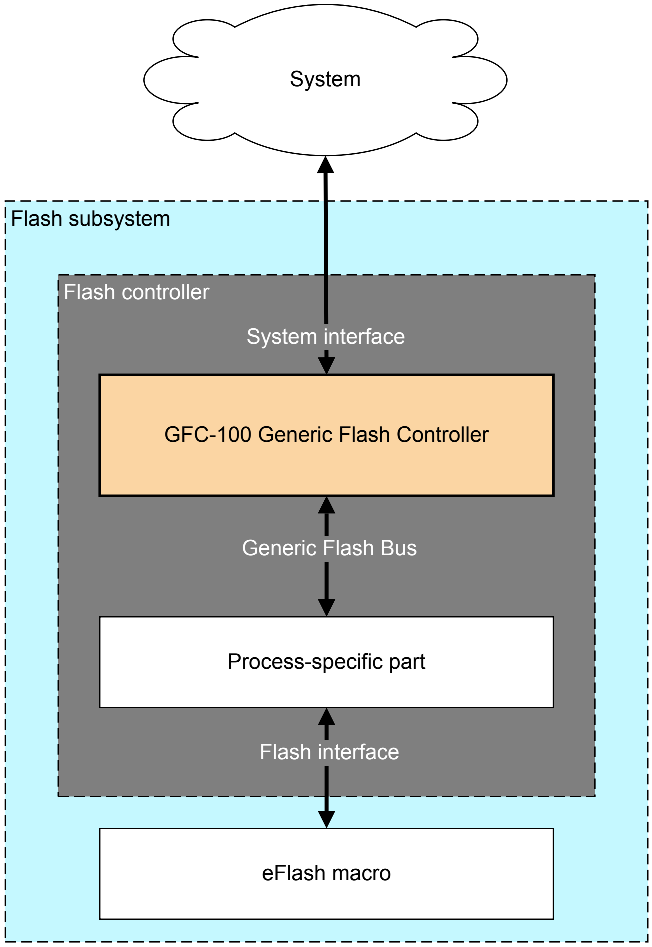


Figure 2-12 GFC-100 in a Flash controller implementation

2.16.2 Features

GFC-100 provides several interfaces and test features.

Advanced High-performance Bus (AHB-Lite) interface:

- Read access to the main and extended areas of embedded Flash
- Burst support
- Low latency

Advanced Peripheral Bus (APB) slave interface:

- Write and erase access to the main and extended areas of embedded Flash
- Debug read access to the main and extended areas of embedded Flash
- Control port for GFC-100 and the eFlash macro
- Interrupt capability for long running commands
- Access to internal and external registers

APB register master interface:

- Control port for attached process-specific registers

Q-Channel interface:

- Control port for system power
- Control port for the system clock

P-Channel controller interface:

- Control port for power to the attached process-specific part

Generic Flash Bus (GFB):

- Enables GFC-100 accesses to embedded Flash
- Simple command-based protocol
- Synchronous with the AHB clock
- Simplifies communication between GFC-100 and the attached process-specific part

2.17 CG092 AHB Flash Cache

This section gives an overview of the product and its features.

For more information, see the CG092 documentation set:

- *Arm® CG092 AHB Flash Cache Technical Reference Manual*
- *Arm® CoreLink™ CG092 AHB Flash Cache Configuration and Integration Manual*

This section contains the following subsections:

- *2.17.1 About the CoreLink CG092 AHB Flash Cache About CG092* on page 2-56.
- *2.17.2 Features of CG092* on page 2-57.

2.17.1 About the CoreLink CG092 AHB Flash Cache About CG092

The CG092 AHB Flash Cache is an instruction cache that is instantiated between the bus interconnect and the eFlash controller.

The CG092 is a simple cache for on-chip *embedded Flash* (eFlash). The CG092 design is optimized for fetching Cortex-M3 or Cortex-M4 instructions directly from an eFlash. The main benefit of the CG092 is improved power efficiency, but there are also improvements in code fetching performance.

Note

The AHB Flash Cache can also be used with external eFlash if the Flash controller is modified accordingly.

The following figure shows the connections in a typical Flash subsystem.

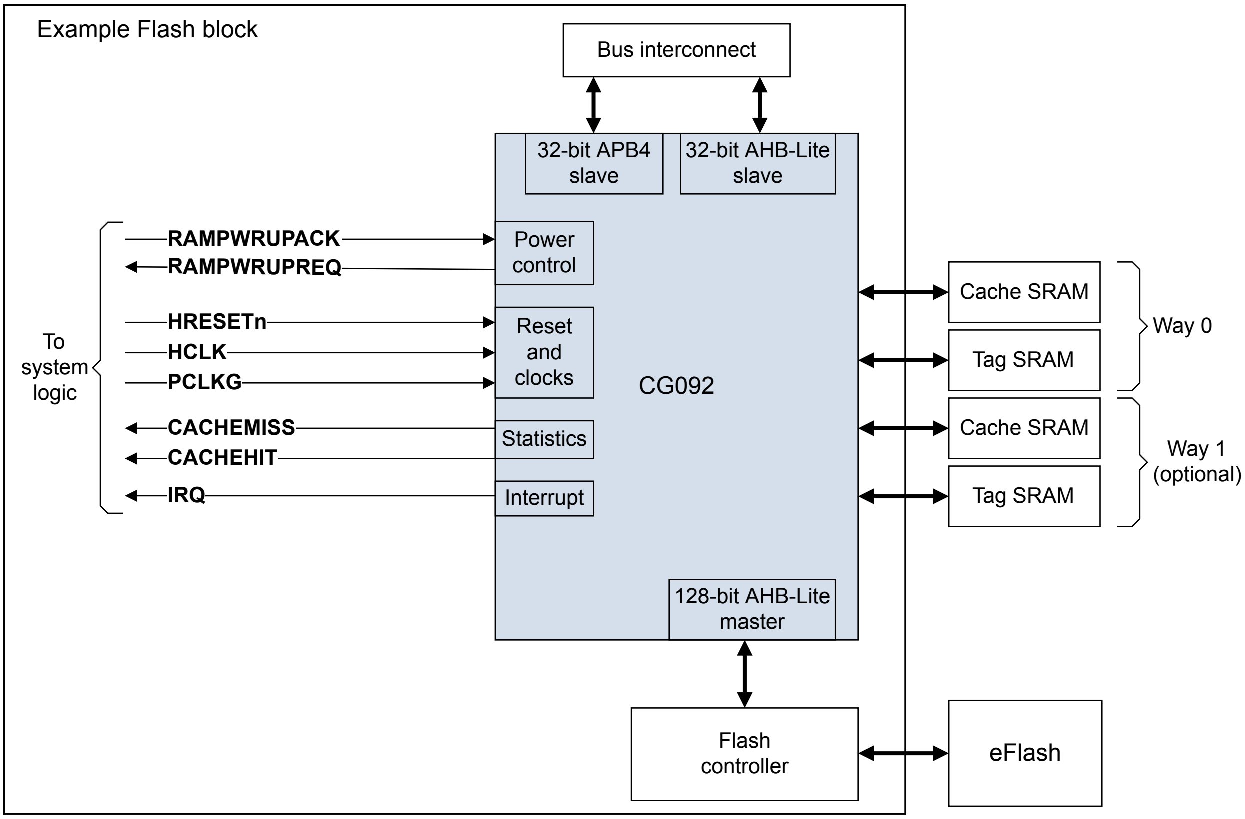


Figure 2-13 Example eFlash implementation

2.17.2 Features of CG092

The CG092 is an instruction cache designed to be instantiated between the bus interconnect and the eFlash controller.

The CG092 has the following features:

- Configurable cache size (minimum 256 bytes/way)
- Four words per cacheline
- Supports 2-way set associative cache, or 1-way fully associative cache
- Configurable address bus size (based on flash memory size) so that tag memory size can be minimized
- SRAM power-control handshaking to an external power management unit
- Supports automatic and manual SRAM power up and power down (with simple handshaking).

  If valid data is in the powered-down cache because the cache is in a low-power state, the cache contents should not be invalidated on wake up. The software can therefore save energy by avoiding invaliding the cache RAMs on wake up.
- Supports automatic or manual cache invalidate in the enabling sequence. This behavior can be overridden.
- 32 bit AHB slave interface to the AHB master in the system processor
- 32 bit APB slave interface to the memory-mapped registers of the CG092
- 128-bit AHB master interface to the eFlash
- Interrupt request generated on SRAM power or manual invalidation errors
- Optional run-time support for prefetch to improve performance when executing a sequence of code that has not been read before.

  The prefetching performance impact is application dependent and might have a negative impact on eFlash power consumption.
- Optional compile-time support configurable performance counters that measure cache hits and misses.

  Exported cache hit and cache miss status signals can be used by performance measurement logic implemented at SoC level.

Note

An eFlash controller is not part of the CG092 component.

2.18 AXI Internal Memory Interface (BP140)

This section gives an overview of the product and its features.

For more information, see the AXI Internal Memory Interface (BP140) documentation set:

- *Arm PrimeCell Infrastructure AMBA 3 AXI Internal Memory Interface (BP140) Technical Overview*
- *Arm PrimeCell Infrastructure AMBA 3 AXI Internal Memory Interface (BP140) Design Manual*

This section contains the following subsections:

- *2.18.1 About the internal memory interface* on page 2-58.
- *2.18.2 Features of the internal memory interface* on page 2-58.

2.18.1 About the internal memory interface

You can use the `IntMemAxi` component to interface to a synthesized SRAM. It can also be connected to ROM.

You can use the internal memory behavioral component, `IntMemBhavAxi`, as a memory slave for behavioral testbenches.

2.18.2 Features of the internal memory interface

The AXI internal memory interface, `IntMemAxi`, has the following features:

- It provides a single-port memory interface configurable for synchronous SRAM or ROM
- The HDL code is supplied as Verilog
- The memory footprint is that of the connected SRAM or ROM
- It accepts a single address for each of the read and write channels
- It uses round-robin arbitration between read and write transactions, the default is read
- For write transactions, the first data transfer can take place 2 cycles after the address is accepted. Subsequent data transfers can complete in consecutive cycles.
- For read transactions:
  - With one wait state: The first data transfer can take place 2 cycles after the address is accepted. Subsequent data transfers can complete in consecutive cycles.
  - With zero wait states: The first data transfer can take place 3 cycles after the address is accepted. Subsequent data transfers take 2 cycles.
- It supports:
  - All AMBA AXI channels except the low power channel
  - All AXI burst types
  - Aligned and unaligned transfer types
- You can configure the following parameters:
  - Data width of 64 bits or 32 bits
  - ID width
  - Wait states, single or none, for SRAM and ROM reads
  - Whether the interface is to be used with SRAM or ROM
  - Memory read access to be zero or one wait-state
  - `MEM_ADDR_WIDTH`, `MEM_INIT_FILE_0`, and `MEM_INIT_FILE_1` for `IntMemBhavAxi`

2.19 XHB-500 bridge

This section gives an overview of the product and its features.

For more information, see the XHB-500 Bridge documentation set:

- *Arm CoreLink XHB-500 Technical Reference Manual AXI5 to AHB5 bridge and AHB5 to AXI5 bridge*
- *Arm CoreLink XHB-500 Configuration and Integration Manual AXI5 to AHB5 bridge and AHB5 to AXI5 bridge*

This section contains the following subsection:

- *2.19.1 About the XHB-500 bridges* on page 2-59.

2.19.1 About the XHB-500 bridges

The product provides an AMBA AXI5 to AHB5 bridge and an AHB5 to AXI5 bridge.

The AXI5 to AHB5 bridge translates AXI5 transactions into the corresponding AHB transfers. The bridge has an AXI5 slave interface and an AHB5 master interface.

The AHB5 to AXI5 translates AHB5 transfers into the corresponding AXI transactions. The bridge has an AHB5 slave interface and an AXI5 master interface.

AXI5 to AHB5 overview

The AHB5 is a low-latency bridge that performs no transaction buffering.

The following figure shows the interfaces of the AHB5.

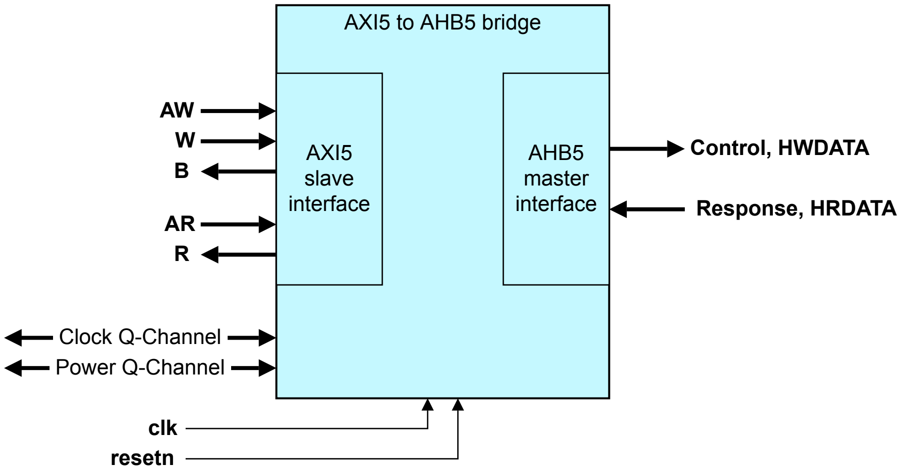


Figure 2-14 AHB5 interfaces

The main features are:

- Single power domain
- Single clock domain
- Configurable data width
- AXI5 slave interface features:
  - AXI5 protocol support
  - AXI4 protocol support
  - Fixed address width
  - Registered or unregistered interface
  - Single Exclusive accesses. Exclusive bursts are not supported
  - Unaligned accesses

- — Conversion of sparse write transactions, when the `HWSTRB_ENABLE` configuration parameter is set to OFF
- — Supports all burst types
- AHB5 master interface features:
  - — AHB5 support
  - — AHB-Lite support, which requires several signals to be tied off
  - — Fixed address width
  - — Registered or unregistered interface
  - — Exclusive accesses. For AHB-Lite, extra glue logic is required.
  - — Write strobe support using the **hwstrb** signal, when the `HWSTRB_ENABLE` configuration parameter is set to ON. The **hwstrb** signal is not present in the *Arm® AMBA® 5 AHB Protocol Specification*.
- Q-Channel interface for clock control
- Q-Channel interface for power control

The bridge does not support endian conversion.

AXI5 overview

The AXI5 to AHB5 bridge is a low-latency bridge that performs no transaction buffering.

The following figure shows the interfaces of the AXI5 to AHB5 bridge.

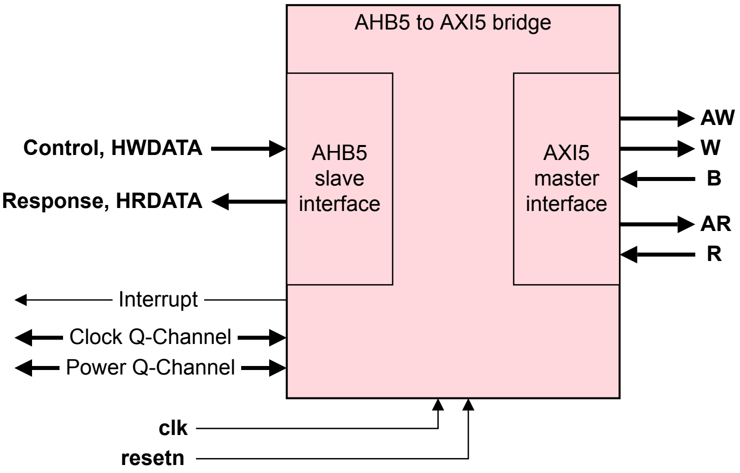


Figure 2-15 AXI5 to AHB5 interfaces

The main features are:

- Single power domain
- Single clock domain
- Configurable data width
- AXI5 slave interface features:
  - — AXI5 protocol support
  - — AXI4 protocol support
  - — Fixed address width
  - — Registered or unregistered interface
  - — Single Exclusive accesses. Exclusive bursts are not supported.
  - — Unaligned accesses
  - — Conversion of sparse write transactions, when the `HWSTRB_ENABLE` configuration parameter is set to OFF
  - — Supports all burst types
- AHB5 master interface features:
  - — AHB5 support
  - — AHB-Lite support, which requires several signals to be tied off
  - — Fixed address width
  - — Registered or unregistered interface

  - — Exclusive accesses. For AHB-Lite, extra glue logic is required.
  - — Write strobe support using the **hwstrb** signal, when the `HWSTRB_ENABLE` configuration parameter is set to ON. The **hwstrb** signal is not present in the *Arm® AMBA® 5 AHB Protocol Specification*.
- Q-Channel interface for clock control
- Q-Channel interface for power control

The bridge does not support endian conversion.

2.20 UART

This section gives an overview of the product and its features.

For more information, see the UART (PL011) documentation set:

- *PrimeCell UART (PL011) Technical Reference Manual*

This section contains the following subsections:

- *2.20.1 About UART* on page 2-62.
- *2.20.2 Features of UART* on page 2-62.

2.20.1 About UART

The UART is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) peripheral that is developed, tested, and licensed by Arm.

The UART is an AMBA slave module that connects to the *Advanced Peripheral Bus* (APB). The UART includes an *Infrared Data Association* (IrDA) *Serial InfraRed* (SIR) protocol *ENcoder/DECoder* (ENDEC).

2.20.2 Features of UART

The UART supports the following features:

- Compliance to the *AMBA Specification* (Rev 2.0) onwards for easy integration into SoC implementation
- Programmable use of UART or IrDA SIR input/output
- Separate 32×8 transmit and 32×12 receive First-In, First-Out (FIFO) memory buffers to reduce CPU interrupts
- Programmable FIFO disabling for 1-byte depth
- Programmable baud rate generator. This enables division of the reference clock by (1×16) to (65535×16) and generates an internal ×16 clock. The divisor can be a fractional number enabling you to use any clock with a frequency >3.6864MHz as the reference clock.
- Standard asynchronous communication bits (start, stop and parity). These are added prior to transmission and removed on reception.
- Independent masking of transmit FIFO, receive FIFO, receive timeout, modem status, and error condition interrupts
- Support for Direct Memory Access (DMA)
- False start bit detection
- Line break generation and detection
- Support of the modem control functions CTS, DCD, DSR, RTS, DTR, and RI
- Programmable hardware flow control
- Fully-programmable serial interface characteristics:
  - data can be 5, 6, 7, or 8 bits
  - even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - baud rate generation, dc up to **UARTCLK**/16
- IrDA SIR ENDEC block providing:
  - programmable use of IrDA SIR or UART input/output
  - support of IrDA SIR ENDEC functions for data rates up to 115200 bps half-duplex
  - support of normal 3/16 and low-power (1.41 - 2.23µs) bit durations
  - programmable division of the **UARTCLK** reference clock to generate the appropriate bit duration for low-power IrDA mode.
- Identification registers that uniquely identify the UART. These can be used by an operating system to automatically configure itself.

2.21 Real Time Clock

This section gives an overview of the product and its features.

For more information, see the RTC documentation set:

- *Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual*

This section contains the following subsections:

- *2.21.1 About Real Time Clock* on page 2-63.
- *2.21.2 Features of the RTC* on page 2-63.

2.21.1 About Real Time Clock

The RTC is an AMBA slave module that connects to the *Advanced Peripheral Bus* (APB).

The following figure shows the RTC block diagram.

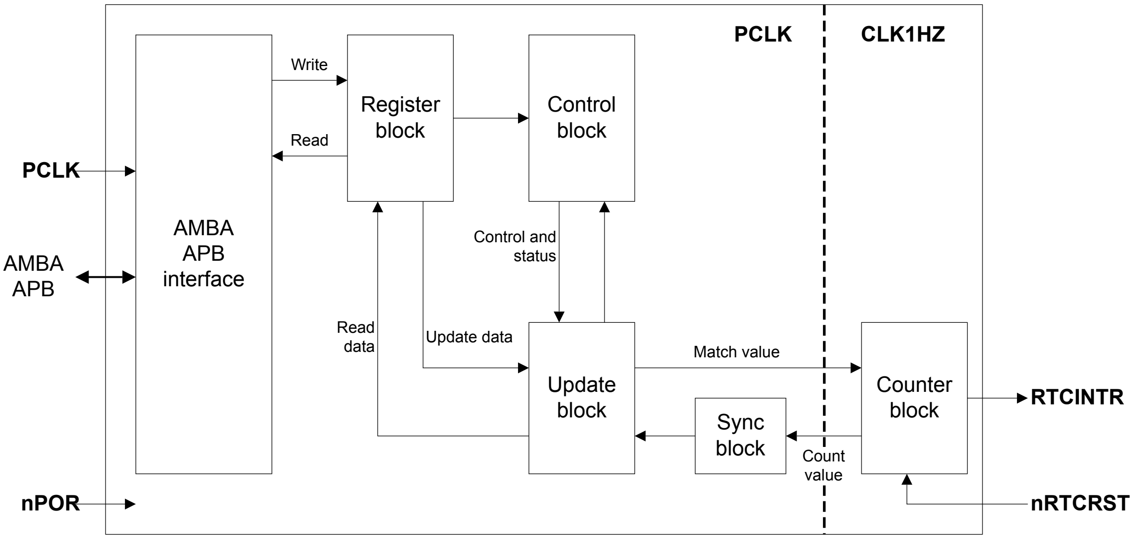


Figure 2-16 RTC block diagram

The RTC can be used to provide a basic alarm function or long time base counter. This is achieved by generating an interrupt signal after counting for a programmed number of cycles of a real-time clock input. Counting in one second intervals requires a 1Hz clock input to the RTC.

2.21.2 Features of the RTC

The features of the RTC are:

- Compliance to the Arm AMBA Specification (Rev 2.0) onwards for easy integration into SoC implementation
- 32-bit up counter (free-running counter)
- Programmable 32-bit match compare register
- Software maskable interrupt when counter and compare registers are identical

Additional test registers and modes are implemented for functional verification and manufacturing test.

2.22 True Random Number Generator

This section gives an overview of the product and its features.

For more information, see the TRNG documentation set:

- *Arm® True Random Number Generator (TRNG) Technical Reference Manual*
- *Arm® True Random Number Generator (TRNG) Configuration and Integration Manual*
- *Arm® TRNG Characterization Application Note*

This section contains the following subsections:

- *2.22.1 About the TRNG* on page 2-64.
- *2.22.2 Features* on page 2-64.

2.22.1 About the TRNG

The TRNG generates a random bit stream.

The entropy of the random bit stream complies with the standards that are described in *Arm® True Random Number Generator (TRNG) Technical Reference Manual*. The TRNG is designed for simple SoC integration. The typical usage of a TRNG is key generation or for seeding approved deterministic random number generators.

2.22.2 Features

The TRNG generates a random bit stream.

The TRNG core has the following key features:

- Produces 10K bits/second of entropy when core is running at 200MHz
- Includes an internal entropy source that is based on a chain of digital inverters. The inverter cells are taken from a standard cell library. No special cells are required.
  - Odd number of inverters, leading to continuous oscillation (while active)
- Built-in hardware tests for auto correlation and *Continuous Random Number Generation Testing* (CRNGT) as required by the following standards:
  - FIPS 140-2, *Security Requirements for Cryptographic Modules*
  - AIS-31, *Functionality Classes and Evaluation Methodology for True Random Number Generators*
- AMBA APB2 slave interface

Appendix A
Revisions

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

It contains the following section:

A.1 Revisions

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

Table A-1 Issue 0000-00

Change	Location	Affects
First release	-	-

Table A-2 Differences between issue 0000-00 and issue 0100-00

Change	Location	Affects
Document name and structure changed	Throughout	r1p0 EAC release
Additional reading list updated for non-confidential documents	*Additional reading* on page 9	
LPD-500 information clarified	*1.2 Product structure* on page 1-14	
Reset Controller information added	*2.1.2 Features of SSE-700* on page 2-22	