Arm® Corstone™-201 Foundation IP

Revision: r0p0

Technical Overview

arm

101633_0000_00_en

Arm® Corstone™-201 Foundation IP

Technical Overview

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	21 March 2019	Non-Confidential	First release

Non-Confidential Proprietary Notice

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

http://www.arm.com

Contents
Arm® Corstone™-201 Foundation IP Technical Overview

Preface

This preface introduces the *Arm® Corstone™-201 Foundation IP Technical Overview*.

It contains the following:

About this book

This Technical Overview is for Arm® Corstone™-201 foundation IP. It describes Corstone-201 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, r*1*p*2*, 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-201.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
This chapter gives an overview of Arm Corstone™-201 foundation IP and its features.

Chapter 2 Component IP overview
This chapter describes the IP products included in the Corstone-201 license.

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.

`mono`space
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.

Timing diagrams

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

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

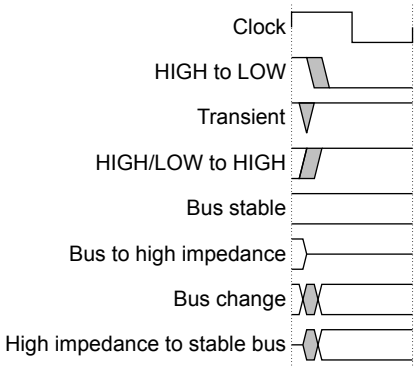


Figure 1 Key to timing diagram conventions

Signals

The signal conventions are:

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

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

Additional reading

This section lists publications by Arm and by third parties.

See *Infocenter*, for access to Arm documentation.

Arm publications

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

- *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* (101373).
- *Arm® SSE-123 Example Subsystem Technical Reference Manual* (101370).
- *Arm® CoreLink™ SSE-050 Subsystem Technical Reference Manual* (101918).
- *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® 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™ CG092 AHB Flash Cache Technical Reference Manual* (DDI 0569).
- *Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual* (DDI 0224).
- *Arm® TrustZone® True Random Number Generator Technical Reference Manual* (100976).
- *Arm® Cortex®-M3 Processor Technical Reference Manual* (100165).
- *Arm® Cortex®-M33 Processor Processor Technical Reference Manual* (100230).
- *Arm® PrimeCell µDMA Controller (PL230) Technical Reference Manual* (DDI 0417).
- *Arm® AMBA® APB Protocol Specification* (IHI 0024).
- *Arm® v8-M Architecture Reference Manual* (DDI 0553).

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

- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Configuration and Integration Manual* (100224)
- *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® 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® CG092 AHB Flash Cache Configuration and Integration Manual* (DIT 0065B).
- *Arm® TrustZone® TrustZone® True Random Number Generator Configuration and Integration Manual* (100977).

Note

- 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.

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-201 Foundation IP Technical Overview*.
- The number 101633_0000_00_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 gives an overview of Arm Corstone™-201 foundation IP and its features.

It contains the following sections:

1.1 About Corstone-201

Corstone foundation IP makes an ideal starting point for creating *Internet of Things* (IoT) *System on Chip* (SoC) designs based on the power-efficient Arm Cortex-M cores. Corstone-201 provides you with a solid base for secure mainstream or constrained devices.

Corstone-201 subsystems and example subsystems are pre-verified, configurable, and modifiable, and pre-integrate cores and security IP with the most relevant Arm CoreLink and Arm CoreSight components.

1.1.1 Corstone-201 IP components

Corstone-201 grants licenses to the following subsystems, security IP and system IP:

Subsystems

CoreLink SSE-200 Subsystem for Embedded
SSE-200 provides a high-performance and low-power computing subsystem for Cortex-M33 cores. You can use it as the foundation of a secure system because of system-level support for TrustZone technologies.

See *2.1 SSE-200 Subsystem* on page 2-19.

SSE-123 Example Subsystem
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. You can implement the subsystem as a standalone single core system or as part of a cluster system.

See *2.2 SSE-123 Example Subsystem* on page 2-21.

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

See *2.3 SSE-050 Subsystem* on page 2-24.

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 *2.4 Cortex-M System Design Kit* on page 2-27.

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 *2.5 SIE-200 System IP for Embedded* on page 2-28.

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 *2.6 SOC-400M* on page 2-29.

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 *2.7 GFC-200 Generic Flash Controller* on page 2-31.

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 *2.8 GFC-100 Generic Flash Controller* on page 2-34.

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 *2.9 PCK-600 Power Control Kit* on page 2-37.

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 *2.10 SDC-600 Secure Debug Channel* on page 2-39.

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.

See *2.11 LPD-500 Low Power Distributor* on page 2-41.

CoreLink CG092 AHB Flash Cache

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

See *2.12 CG092 AHB Flash Cache* on page 2-42.

PrimeCell 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 *2.13 Real Time Clock* on page 2-44.

TrustZone 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 *2.14 True Random Number Generator* on page 2-45.

Separately licensed IP

In order to provide optimum flexibility, all Cortex cores must be licensed separately.

See the individual release notes for instructions on downloading and installing the components that you require.

1.1.2 Using the Corstone components

The Corstone components only form part of the finished SoC and you must extend and customize the subsystems for your specific application requirements.

The following examples show you some of the ways you can use the components that are licensed by Corstone-201:

- Use the SSE-050 or SSE-200 subsystem as a foundation for your own IoT solution that is based around the Cortex-M3 or Cortex-M33 cores.
- Use the SIE-200 components to add bus and controller IP to create secure TrustZone systems.

- Use the *Cortex-M System Design Kit* (CMSDK) and the example systems as a starting point for your own IoT solution that is based around the Cortex-M0, Cortex-M0+, Cortex-M3, or Cortex-M4 cores.
- Use the system IP provided with the subsystems and your own IP to create a custom solution. You can use the example systems and software libraries as a reference for your system solution.

A complete system typically contains the following components:

Compute subsystem
A compute subsystem consisting of Cortex-M cores and associated bus, debug, controller, peripherals, and interface logic supplied by Arm.

Reference system memory and peripherals
SRAM is part of some of the subsystems, but a SoC requires extra memory, control, and peripheral components beyond the minimum subsystem components. Flash memory, for example, is not provided with the SSE-200.

Communication interface
The endpoint must have some way of communicating with other nodes or masters in the system. This interface could be WiFi, Bluetooth, or a wired connection.

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

Software development environment
Arm provides a complete software development environment which includes the Mbed operating system, Arm or GNU (GCC) compilers and debuggers, and firmware. Custom peripherals typically require corresponding third-party firmware that can be integrated into the software stack.

1.2 Product deliverables

The Corstone-201 product bundle (BP311) does not have hardware or software deliverables. Its subsystems and IP component products include these deliverables.

The hardware deliverables must be downloaded separately for the following IP products that are included in the Corstone-201 license:

- CoreLink SSE-200 Subsystem for Embedded (CG062).
- SSE-123 Example Subsystem (CG065).
- CoreLink SSE-050 Subsystem (CG063).
- Cortex-M System Design Kit (BP210).
- CoreLink SIE-200 System IP for Embedded (BP300).
- CoreSight SOC-400M (TM150).
- CoreLink GFC-200 Generic Flash Controller (CG094).
- CoreLink GFC-100 Generic Flash Controller (CG090).
- CoreLink PCK-600 Power Control Kit (PL608).
- CoreSight SDC-600 Secure Debug Channel (TM210).
- CoreLink LPD-500 Low Power Distributor (PL408).
- CoreLink CG092 AHB Flash Cache (CG092).
- PrimeCell Real Time Clock (PL031)
- True Random Number Generator (CC003).

See the *Arm® Corstone™-201 Foundation IP Release Note* for the component versions.

1.3 Compliance

See the component *Technical Reference Manuals* for more details about compliance to the following specifications:

- Arm architecture.
- CoreSight Debug.
- Advanced Microcontroller Bus Architecture.

1.4 Documentation

The following documents are supplied with the Corstone-201 product bundle:

Technical Overview

The *Technical Overview* (TO) describes the functionality of Corstone-201.

Release Note

The *Release Note* describes download and installation instructions for the IP products included in Corstone-201.

Note

- The separately downloaded product bundles also contain documentation such as *Technical Reference Manual*s or *Configuration and Integration Manual*s.
- See the individual product bundles for details of what documentation is provided for that IP bundle.

Chapter 2
Component IP overview

This chapter describes the IP products included in the Corstone-201 license.

It contains the following sections:

2.1 SSE-200 Subsystem

This section is an extract from the SSE-200 technical reference manual. It 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:

2.1.1 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. 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.

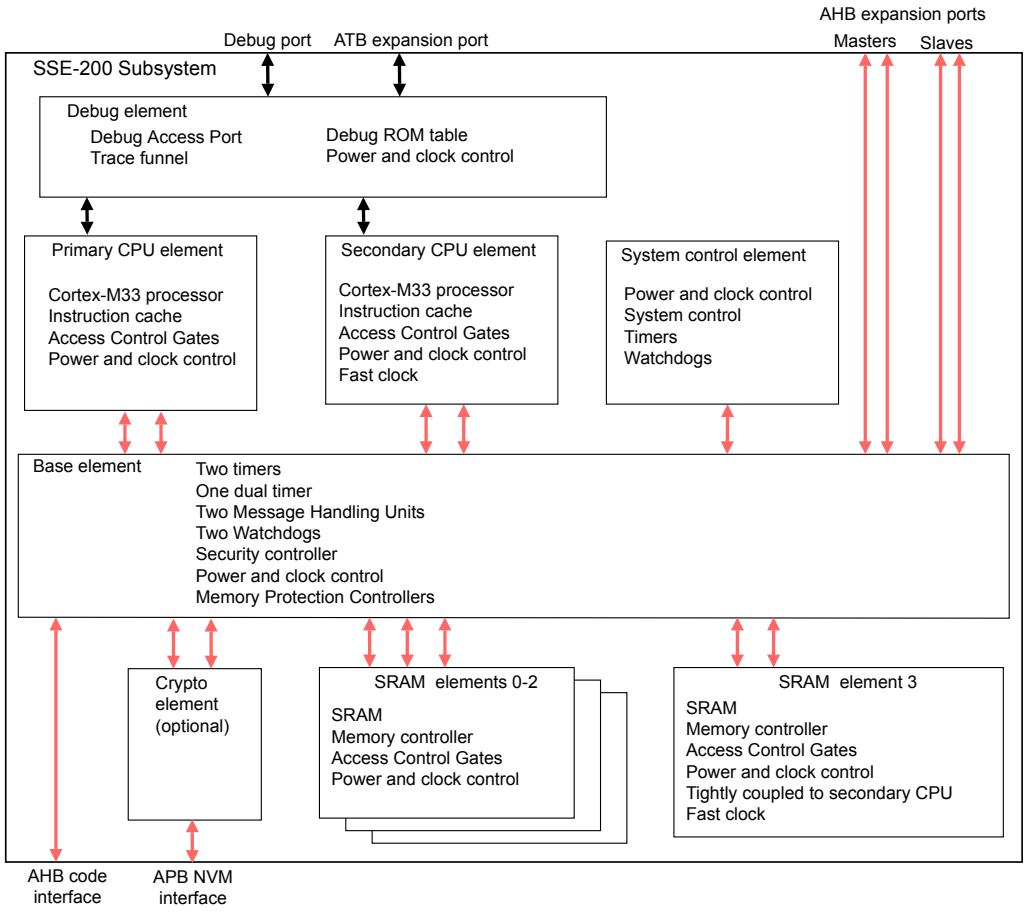


Figure 2-1 SSE-200 subsystem elements

2.1.2 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:
  - TrustZone 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).

2.2 SSE-123 Example Subsystem

This section is an extract from the SSE-123 technical reference manual. It 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:

2.2.1 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.

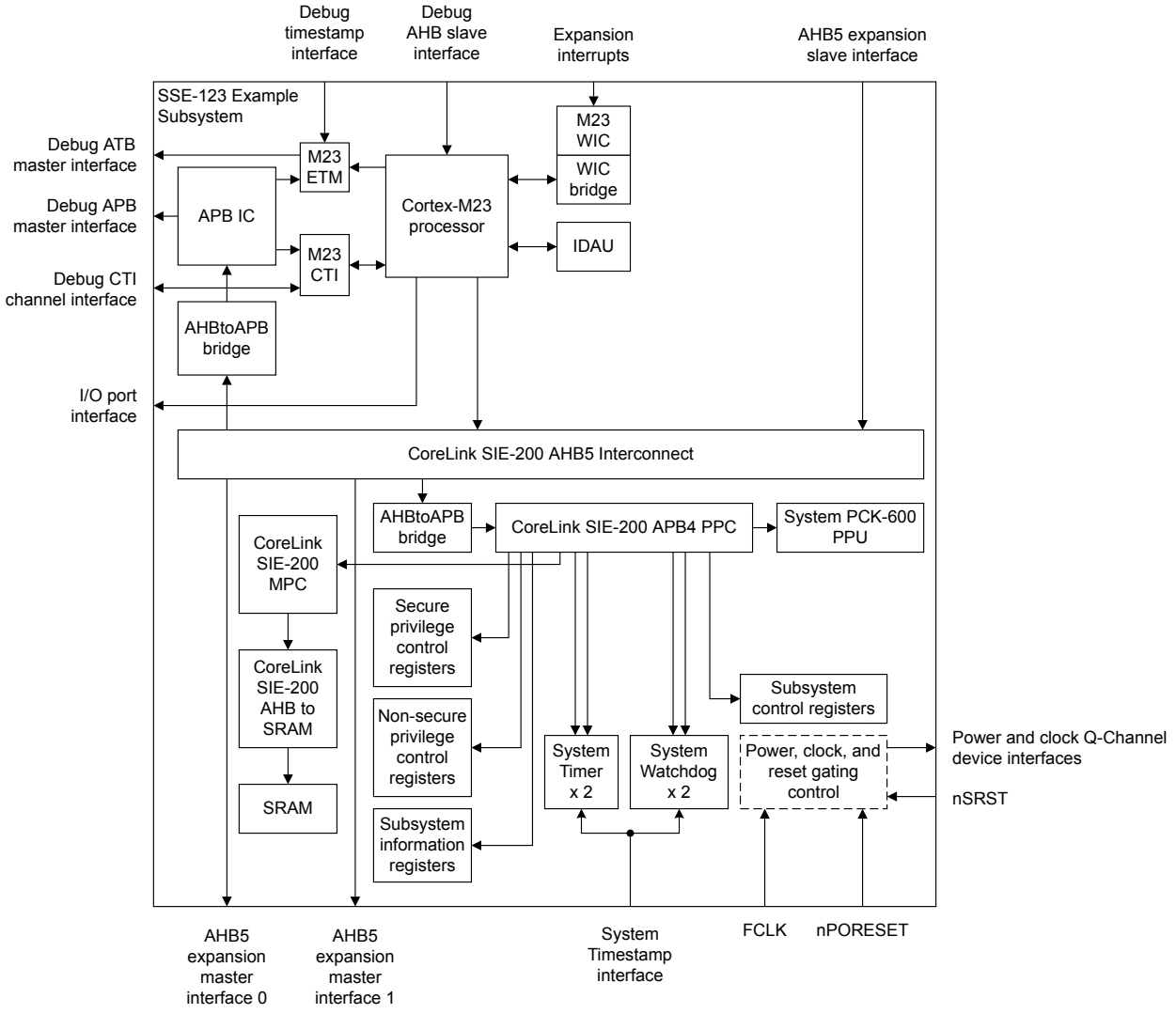


Figure 2-2 SSE-123 block diagram

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

2.2.2 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.

2.3 SSE-050 Subsystem

This section is an extract from the SSE-050 technical reference manual. It 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:

- *2.3.1 About SSE-050* on page 2-24.
- *2.3.2 Features of the SSE-050* on page 2-25.

2.3.1 About SSE-050

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.

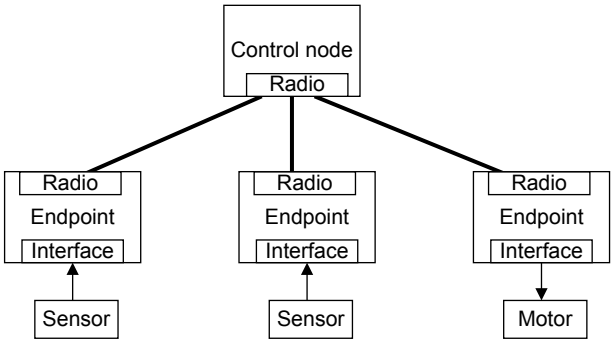


Figure 2-3 An IoT endpoint as part of a larger control system

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

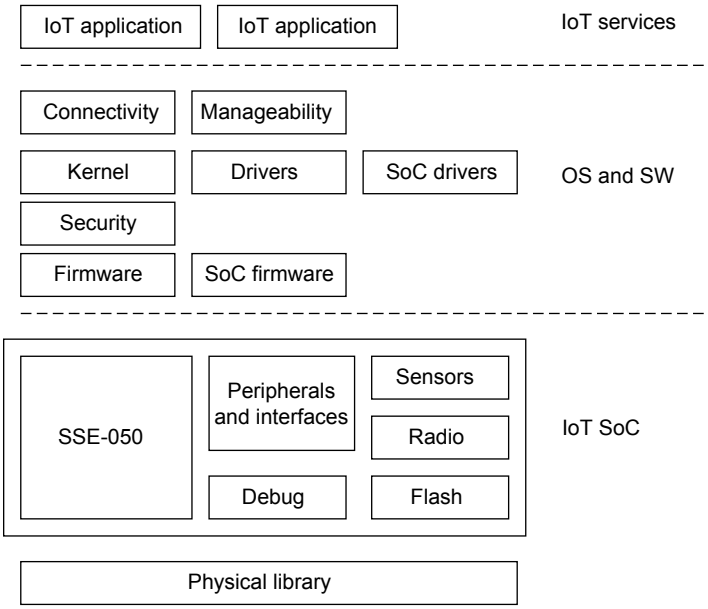


Figure 2-4 IoT endpoint HW and SW solution

2.3.2 Features of the SSE-050

The SSE-050 contains the following components:

- A 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 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.

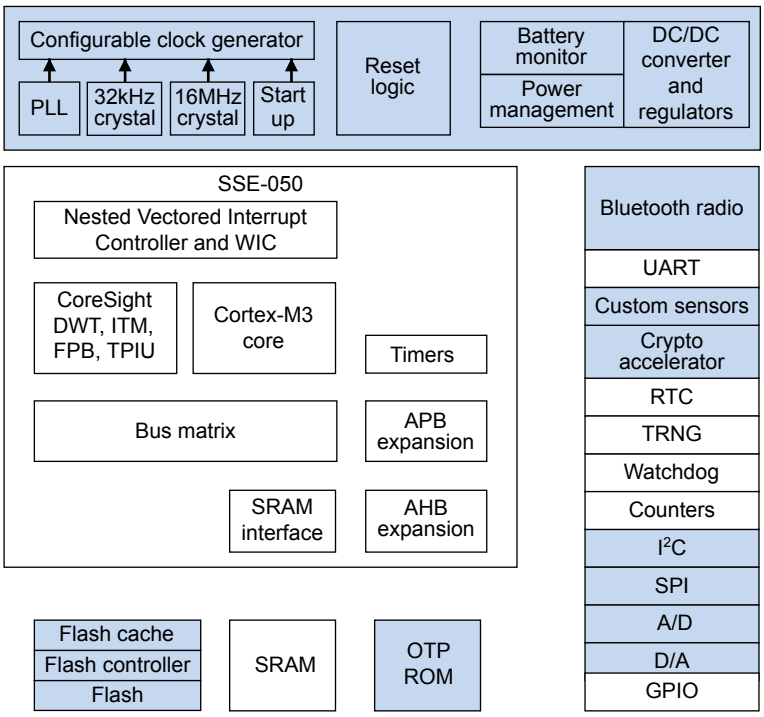


Figure 2-5 Example of an IoT endpoint SoC

2.4 Cortex-M System Design Kit

This section is an extract from the CMSDK technical reference manual. It 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:

- *2.4.1 About the Cortex-M System Design Kit* on page 2-27.

2.4.1 About the Cortex-M System Design Kit

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

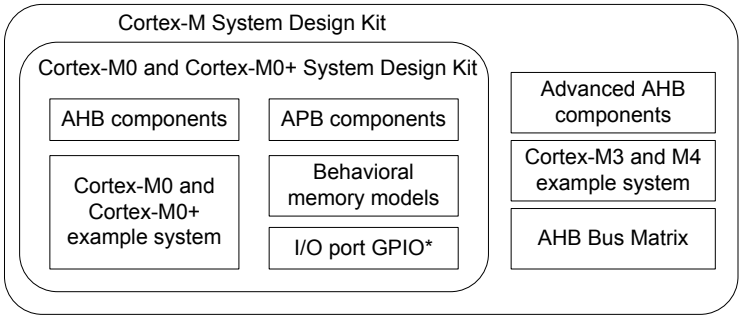
The design kit contains the following:

- 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 *Difference between the two versions of the design kit* on page 2-27.



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

2.5 SIE-200 System IP for Embedded

This section is an extract from the SIE-200 technical reference manual. It 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:

2.5.1 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.

2.5.2 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.

2.6 SOC-400M

The SOC-400M license enables you to use SOC-400 functionality with Cortex-M cores. This section is an extract from the SOC-400 technical reference manual. It gives an overview of the product and its features.

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:

2.6.1 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.

2.6.2 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.

2.7 GFC-200 Generic Flash Controller

This section is an extract from the GFC-200 technical reference manual. It 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:

2.7.1 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.

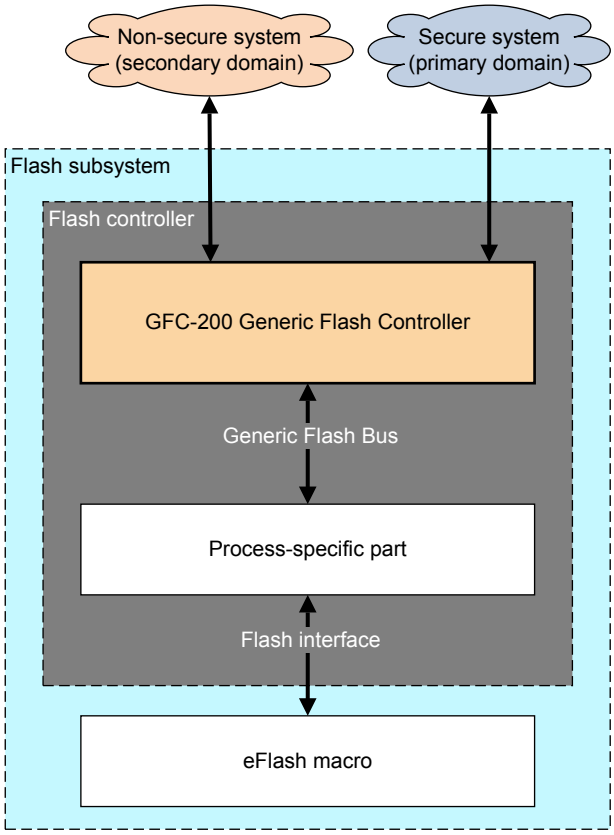


Figure 2-6 GFC-200 in a Flash controller implementation

2.7.2 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.

2.8 GFC-100 Generic Flash Controller

This section is an extract from the GFC-100 technical reference manual. It 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:

2.8.1 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.

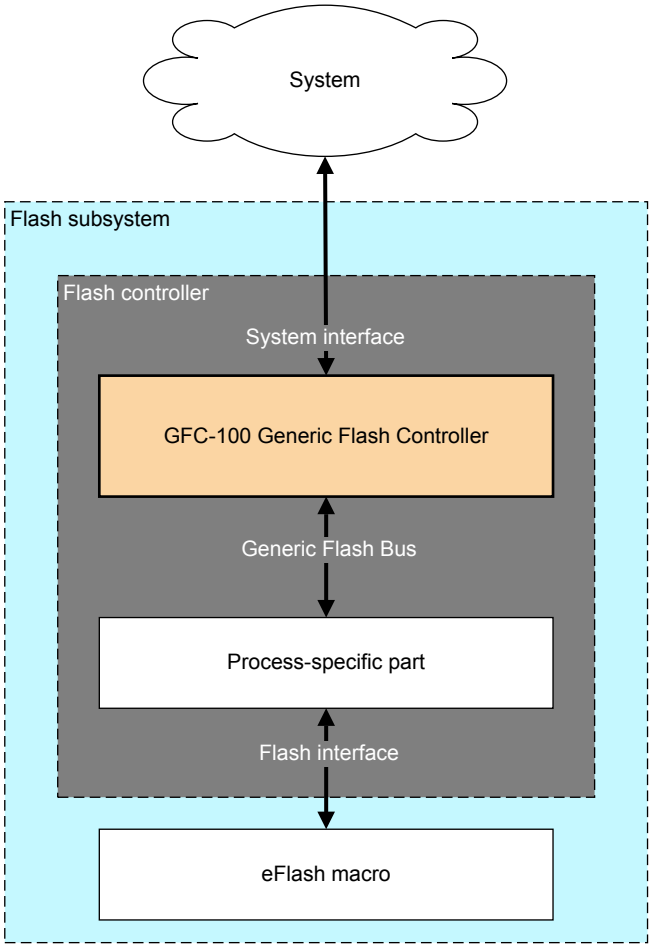


Figure 2-7 GFC-100 in a Flash controller implementation

2.8.2 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.

2.9 PCK-600 Power Control Kit

This section is an extract from the PCK-600 technical reference manual. It 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:

- *2.9.1 About the Power Control Kit* on page 2-37.

2.9.1 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.

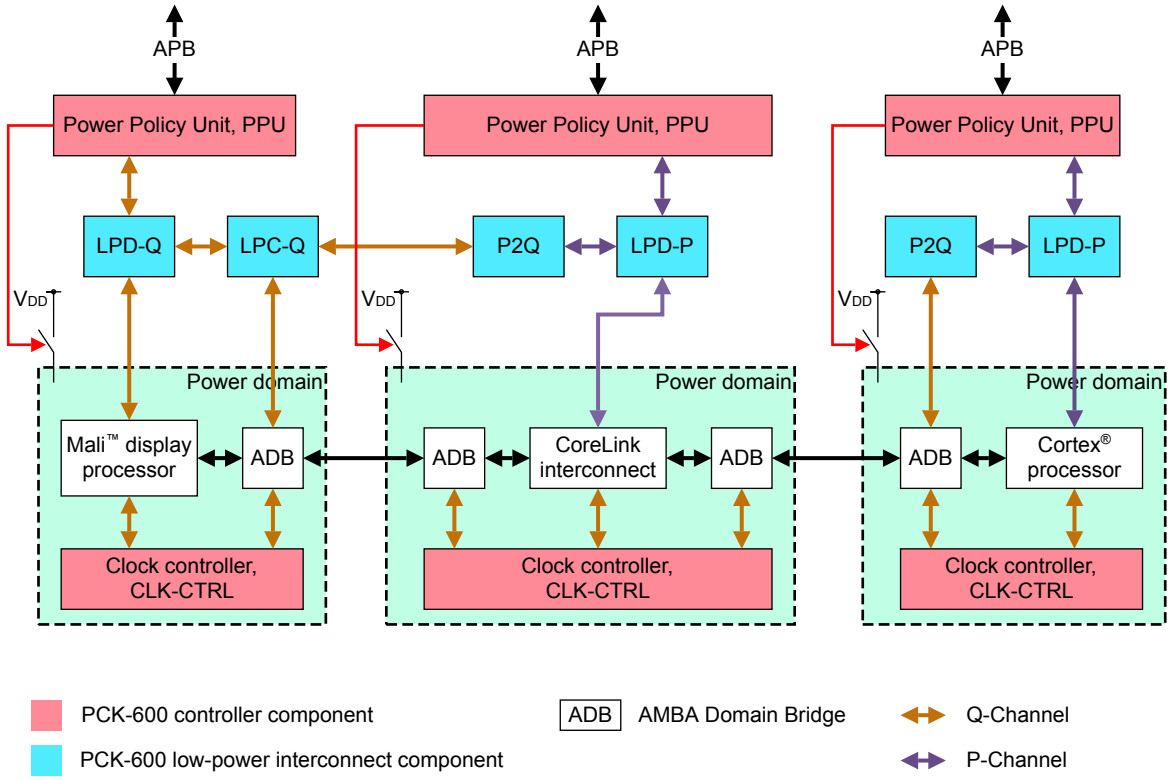


Figure 2-8 Example system that contains PCK-600

2.10 SDC-600 Secure Debug Channel

This section is an extract from the SDC-600 technical reference manual. It 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:

- *2.10.1 About SDC-600* on page 2-39.

2.10.1 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.

The following terminology is used throughout the document:

External
: The component or the end of the communication channel that is connected to the debugger through the Debug Port.

Internal

The component or the end of the communication channel that is connected to the servicing agent.

Servicing agent

The agent on the internal side that implements authentication by checking the certificate and controls the authentication signals in the target system. It communicates through the SDC-600 and services the interrupts that are generated by the internal COM Port component. The servicing agent can be implemented as software executing on the target processor, or on a separate processor in a secure island or subsystem.

Target

The debug target that is requesting debug authentication. In some systems, the servicing agent can be implemented as code which runs on the target processor.

2.11 LPD-500 Low Power Distributor

This section is an extract from the LPD-500 technical reference manual. It gives an overview of the product and its features.

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

- *Arm® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual*.
- *Arm® CoreLink™ LPD-500 Low Power Distributor Integration and Implementation Manual*.

This section contains the following subsections:

2.11.1 About the LPD-500

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

Q-Channels are used to manage quiescence in components of the system that allow the clock to be gated off or power to be removed. Gating off a clock or removing power is done to save power when not operational.

The LPD-500 supports use cases where not all signals of the Q-Channel are used by an attached device. See the *Arm® CoreLink™ LPD-500 Integration and Implementation Manual* for more information.

2.11.2 Features

The LPD-500 provides a low latency method of controlling multiple, device-level, *Low Power Interfaces* (LPIs) from a single controller.

The LPD-500 supports the following key features:

- Expands a single Q-Channel LPI from a power controller or a clock controller into multiple Q-Channel LPIs for controlled devices.
- Low latency to and from device channels.
- Up to 32 device control channels.
- Cascadable to multiple levels to expand beyond 32 devices.
- Optionally integrates synchronizers on request and accepts inputs for use in systems with different clock domains.
- Configurable as an expander, where all devices are controlled together, or as a sequencer, where all devices are controlled in a sequence.
- Optional active deny feature to allow a denial of quiescence that is based on a device signal.

2.12 CG092 AHB Flash Cache

This section is an extract from the CG092 technical reference manual. It 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® CG092 AHB Flash Cache Configuration and Integration Manual*.

This section contains the following subsections:

- *2.12.1 About CG092* on page 2-42.
- *2.12.2 Features of CG092* on page 2-43.

2.12.1 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.

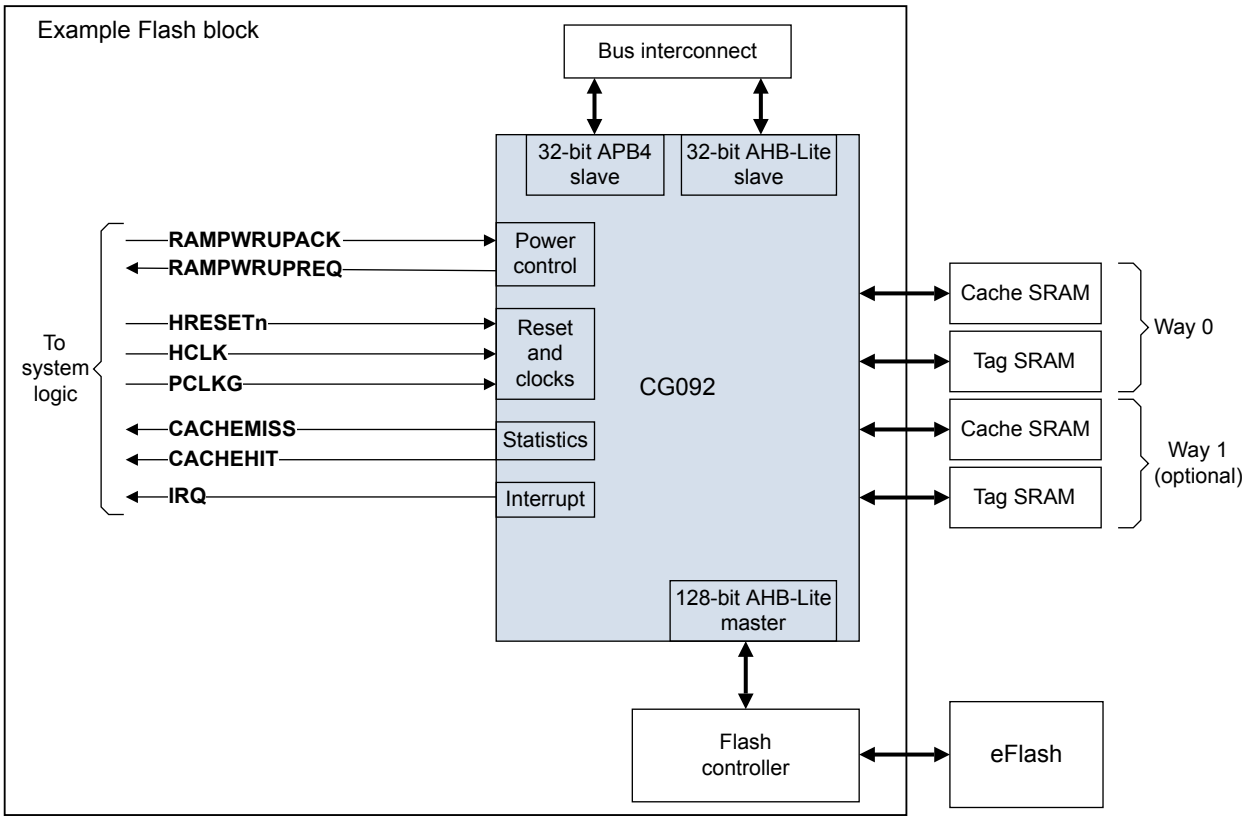


Figure 2-9 Example eFlash implementation

2.12.2 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.

2.13 Real Time Clock

This section is an extract from the RTC technical reference manual. It 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:

- *2.13.1 About Real Time Clock* on page 2-44.
- *2.13.2 Features of the RTC* on page 2-44.

2.13.1 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.

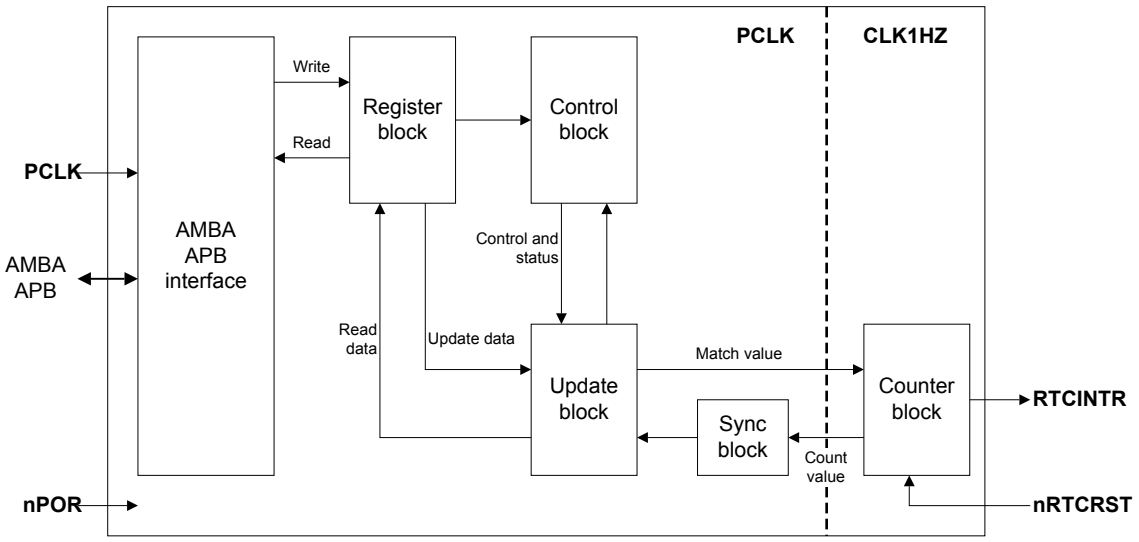


Figure 2-10 RTC block diagram

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.

2.13.2 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.

2.14 True Random Number Generator

This section is an extract from the TRNG technical reference manual. It gives an overview of the product and its features.

For more information, see the TRNG documentation set:

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

This section contains the following subsections:

2.14.1 About the TRNG

The TRNG enables generation and collection of a truly random bit stream from a digital logic. The TRNG is designed for simple SoC integration.

The typical usage of a TRNG is key generation or for seeding approved deterministic random numbers.

2.14.2 Features

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.
  - Odd number of inverters, leading to continuous oscillation (while active).
  - Inverter cells that are taken from a standard cells library.
- 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* on page Appx-A-47.

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	-	-