Arm® Corstone™-102 Reference Package

Revision: r0p0

Technical Overview

Non-Confidential

Issue 01

101634_0000_01_en

Arm® Corstone™-102 Reference Package

Technical Overview

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	21 June 2019	Non-Confidential	First release
0000-01	29 July 2022	Non-Confidential	Second release

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.

Feedback

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

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

Inclusive language commitment

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

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

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

Contents

1. Introduction

1.1 Product revision status

The r*x*p*y* identifier indicates the revision status of the product described in this manual, for example, r*1*p*2*, where:

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

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

1.2 Intended audience

This book is written for hardware or software engineers who want an overview of the components and functionality in Corstone-102.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Typographic conventions

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

Convention	Use
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

Caution

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

Warning

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

Danger

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

Note

An important piece of information that needs your attention.

Tip

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

Remember

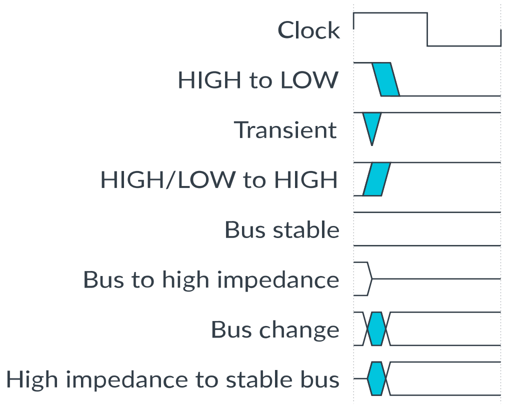
A reminder of something important that relates to the information you are reading.

Timing diagrams

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

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

1.4 Additional reading

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

Table 1-2: Arm publications

Document name	Document ID	Licensee only
Arm® Corstone™ SSE-123 Example Subsystem Technical Overview	101371	No
Arm® Corstone™ SSE-123 Example Subsystem Technical Reference Manual	101370	No
Arm® Corstone™ SSE-050 Subsystem Technical Reference Manual	100918	No
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No

Document name	Document ID	Licensee only
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	No
Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual	101484	No
Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual	101059	No
Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual	101150	No
Arm® CoreLink™ CG092 AHB Flash Cache Technical Reference Manual	DDI 0569	No
Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual	DDI 0224	No
Arm® Cortex®-M3 Processor Technical Reference Manual	100165	No
Arm® Cortex®-M23 Processor Technical Reference Manual	DDI 0550	No
PrimeCell µDMA Controller (PL230) Technical Reference Manual	DDI 0417	No
AMBA® APB Protocol Specification	IHI 0024	No
Arm®v8-M Architecture Reference Manual	DDI 0553	No
Arm® Corstone™ SSE-123 Example Subsystem Configuration and Integration Manual	101372	Yes
Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual	100919	Yes
Arm® Cortex®-M System Design Kit Example System Guide	DUI 0594	Yes
Arm® Cortex®-M0 and Cortex®-M0+ System Design Kit Example System Guide	DUI 0559	Yes
Arm® CoreLink™ SIE-200 System IP for Embedded Configuration and Integration Manual	DIT 0067	Yes
Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual	101485	Yes
Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual	101060	Yes
Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual	101151	Yes
Arm® CoreLink™ CG092 AHB Flash Cache Configuration and Integration Manual	DIT 0065B	Yes

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.

Note

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

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

2. Corstone™-102 Reference Package

Corstone™-102 Reference Package 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-102 provides you with a solid base for secure constrained devices.

Corstone-102 IP 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.

2.1 Corstone-102 IP components

Corstone-102 grants licenses to the following subsystems, security IP and system IP:

Subsystems

Corstone™ 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 3.1 SSE-123 Example Subsystem on page 15.

Corstone™ 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 3.2 SSE-050 Subsystem on page 16.

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 3.3 Cortex-M System Design Kit on page 19.

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 Arm®v8-M core architecture.
See 3.4 SIE-200 System IP for Embedded on page 21.

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 3.7 PCK-600 Power Control Kit on page 27.

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 3.5 GFC-200 Generic Flash Controller on page 22.

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 3.6 GFC-100 Generic Flash Controller on page 25.

CoreLink™ CG092 AHB Flash Cache

CG092 is an instruction cache that is instantiated between the bus interconnect and the *embedded Flash* (eFlash) controller.
See 3.8 CG092 AHB Flash Cache on page 29.

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 3.9 Real Time Clock on page 30.

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.

2.2 Using the Corstone components

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

The following examples show how you can use the components that are licensed by Corstone™-102:

- Use the SSE-123 or SSE-050 subsystem to build your IoT solution. Your solution might contain Cortex®-M23 or Cortex®-M3 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 to build your IoT solution. Your solution might contain 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-123.

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 Arm® Mbed™ *Operating System* (OS), 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.

2.3 Product deliverables

The Corstone-102 product package (BP316) 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-102 license:

- Corstone™ SSE-123 Example Subsystem (CG065)
- Corstone™ SSE-050 Subsystem (CG063)
- Cortex®-M System Design Kit (BP210)
- CoreLink™ SIE-200 System IP for Embedded (BP300)
- CoreLink™ GFC-200 Generic Flash Controller (CG094)
- CoreLink™ GFC-100 Generic Flash Controller (CG090)
- CoreLink™ PCK-600 Power Control Kit (PL608)
- CoreLink™ CG092 AHB Flash Cache (CG092)
- Real Time Clock (PL031)

See the *Arm® Corstone™-102 Reference Package Release Note* for the component versions.

2.4 Compliance

See the relevant component *Technical Reference Manuals* for more details about compliance that relates to the following areas:

- Arm® architecture
- CoreSight™ Debug
- Advanced Microcontroller Bus Architecture (AMBA)

2.5 Documentation

The following documents are supplied with the Corstone-102 product package:

Technical Overview

The *Technical Overview* (TO) describes the functionality of Corstone-102.

Release Note

The *Release Note* describes download and installation instructions for the IP products included in Corstone-102.

Note

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

---

3. Component IP overview

The following sections describe the IP products included in the Corstone-102 license.

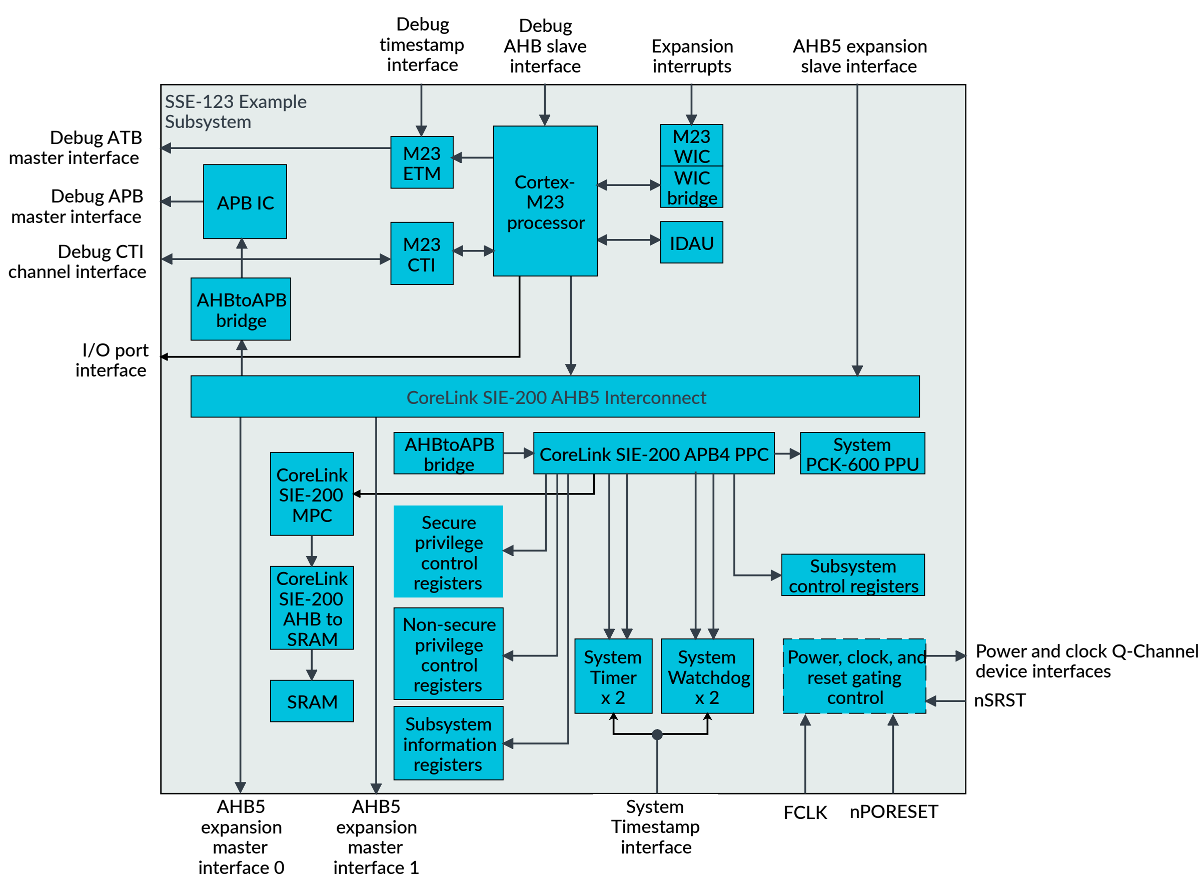
3.1 SSE-123 Example Subsystem

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 3-1: SSE-123 block diagram



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

3.1.1 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

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

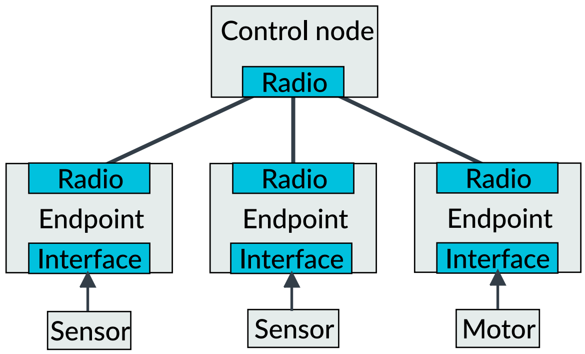
- *Arm® Corstone™ SSE-123 Example Subsystem Technical Overview*
- *Arm® Corstone™ SSE-123 Example Subsystem Technical Reference Manual*
- *Arm® Corstone™ SSE-123 Example Subsystem Configuration and Integration Manual*

3.2 SSE-050 Subsystem

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

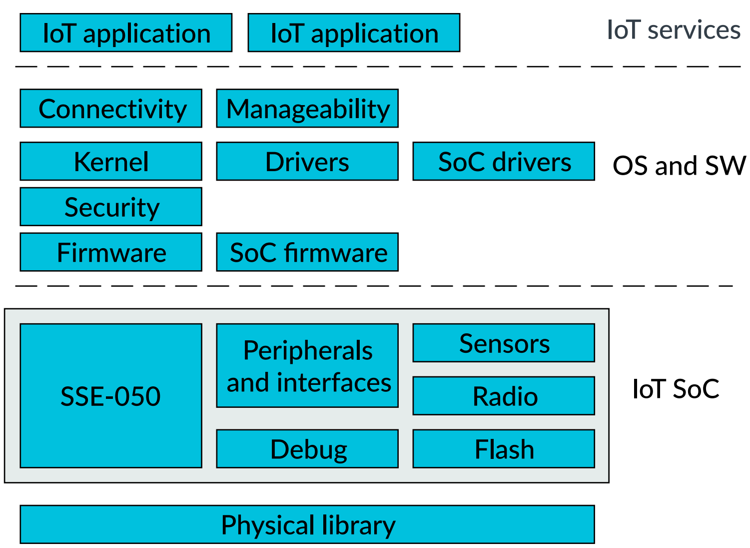
The following figure shows an IoT system consisting of several endpoints and a shared control node.

Figure 3-2: IoT endpoint HW and SW solution



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

Figure 3-3: IoT endpoint HW and SW solution



3.2.1 Features of 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® Corstone™ SSE-050 Subsystem Configuration and Integration Manual* for more information.
    - Little-endian memory addressing only for compatibility with typical eFlash controller and eFlash cache

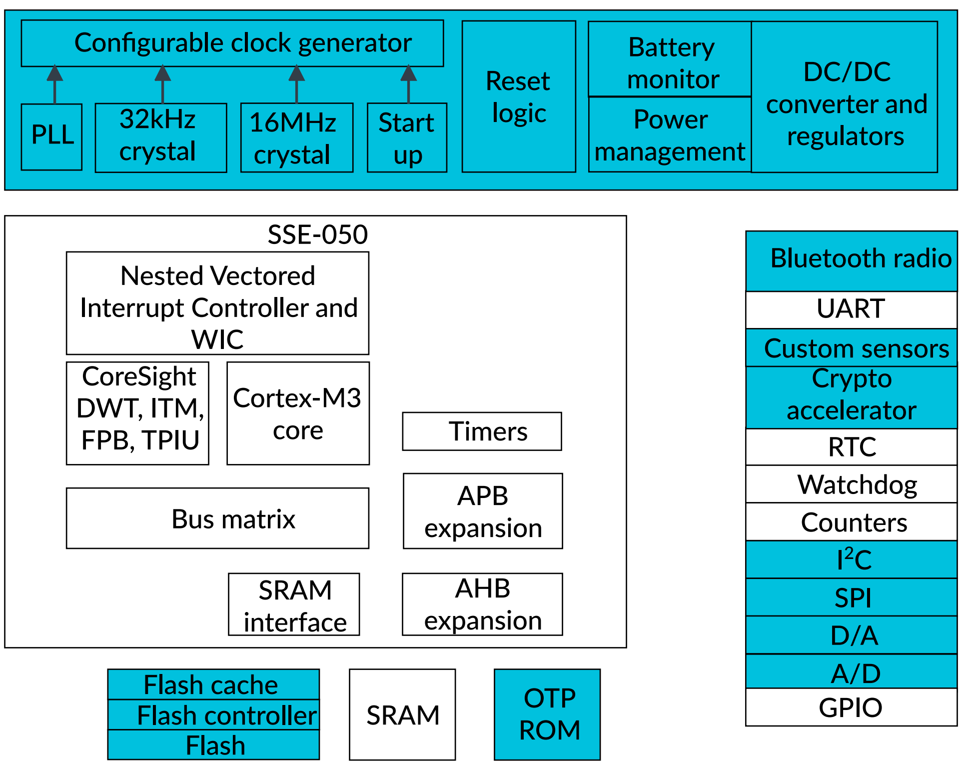
  For more information, see the *Arm® Cortex®-M3 Processor Technical Reference Manual*.
- Integrated debug and trace:
    - Standalone system with a *Trace Port Interface Unit* (TPIU) and a *Serial Wire or JTAG Debug Port* (SWJ - DP)
    - Supports instruction trace using an *Embedded Trace Macrocell* (ETM) if licensed
- Multilayer AMBA® AHB-Lite interconnect:
    - Low-latency interconnect bus matrix
    - Two AHB-Lite slave expansion ports for external AHB masters
    - Two AHB-Lite master expansion ports for external AHB slaves
    - Eleven APB4 master expansion ports (each with 4KB address space) to connect APB peripherals
- Memory system, consisting of:
    - A placeholder for embedded flash controller and optionally cache. The following flash controllers are compatible:
        - GFC-100 Generic Flash Controller
        - GFC-200 Generic Flash Controller
        - Any third-party flash controller that can be integrated to an AHB memory interface and up to two APB control interfaces. The address map is configurable for two banks of 128KB or two banks of 256KB.
    - Static memory (configurable as one to four 32KB banks) is provided in the example integration layer
    - A placeholder for representing a flash memory implementation in the integration layer
- Two APB timers:
    - Interrupt generation when the counter reaches 0
    - Each timer has an TIMERnEXTIN signal that can be used as an enable or external clock
    - Configurable privileged access mode
- Example integration for typical closely-coupled peripherals, using components from CMSDK:
    - Watchdog timer
    - UARTs
    - Application timers

- *Real Time Clock* (RTC)
- Optional radio solution integration capability:
    - AHB master and slave ports
    - Reserved interrupt ports

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 3-4: Example of an IoT endpoint SoC



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

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

3.3 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:

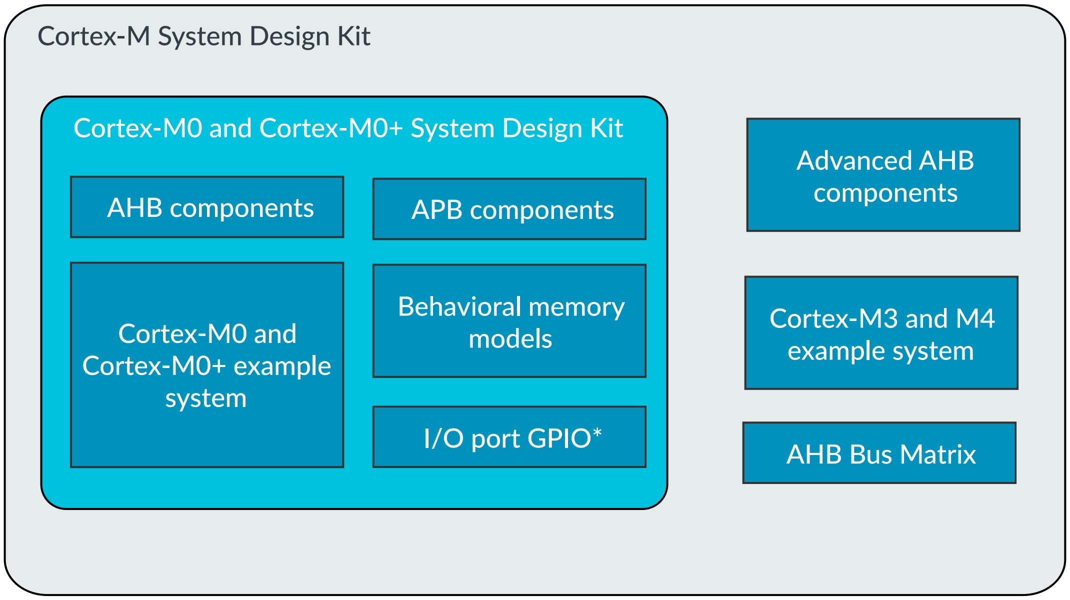
- A selection of AHB-Lite and APB components, including several peripherals such as GPIO, timers, watchdog, and UART
- Example systems for the Cortex-M0, Cortex-M0+, Cortex-M3, and Cortex-M4 cores
- Example synthesis scripts for the example systems
- Example compilation and simulation scripts for the 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 DS-5 Development Studio
  - 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, which supports Cortex-M0 and Cortex-M0+.
- Cortex-M System Design Kit, full version, which 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.

Figure 3-5: Difference between the two versions of the design kit



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

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*

3.4 SIE-200 System IP for Embedded

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 Arm®v8-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

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

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*

3.5 GFC-200 Generic Flash Controller

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.

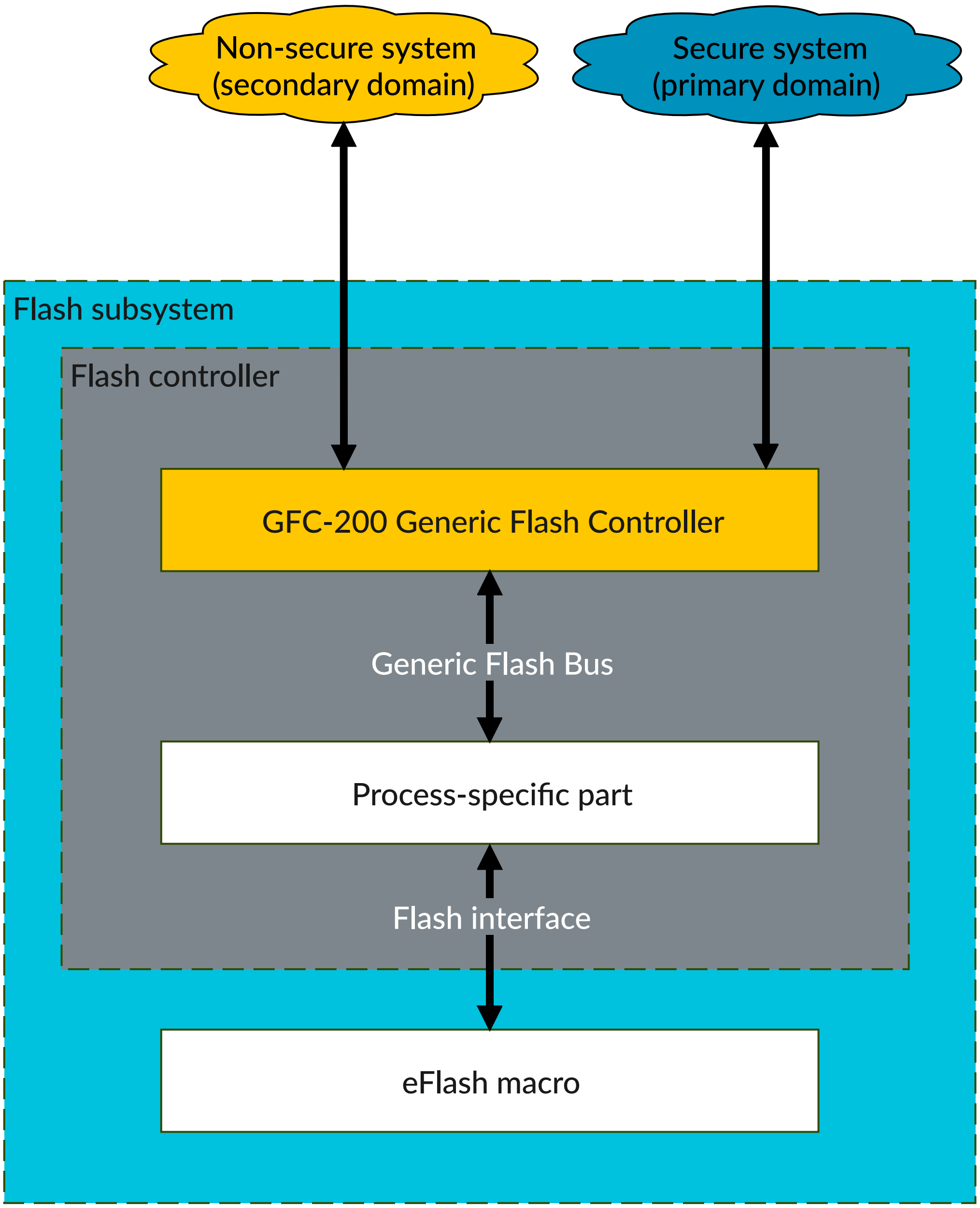
The GFC-200 provides functions that relate only to services for the system side of the Flash controller. The GFC-200 cannot communicate directly with the eFlash macro. Therefore, the GFC-200 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

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 3-6: GFC-200 in a Flash controller implementation



3.5.1 Features of GFC-200

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

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*

3.6 GFC-100 Generic Flash Controller

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.

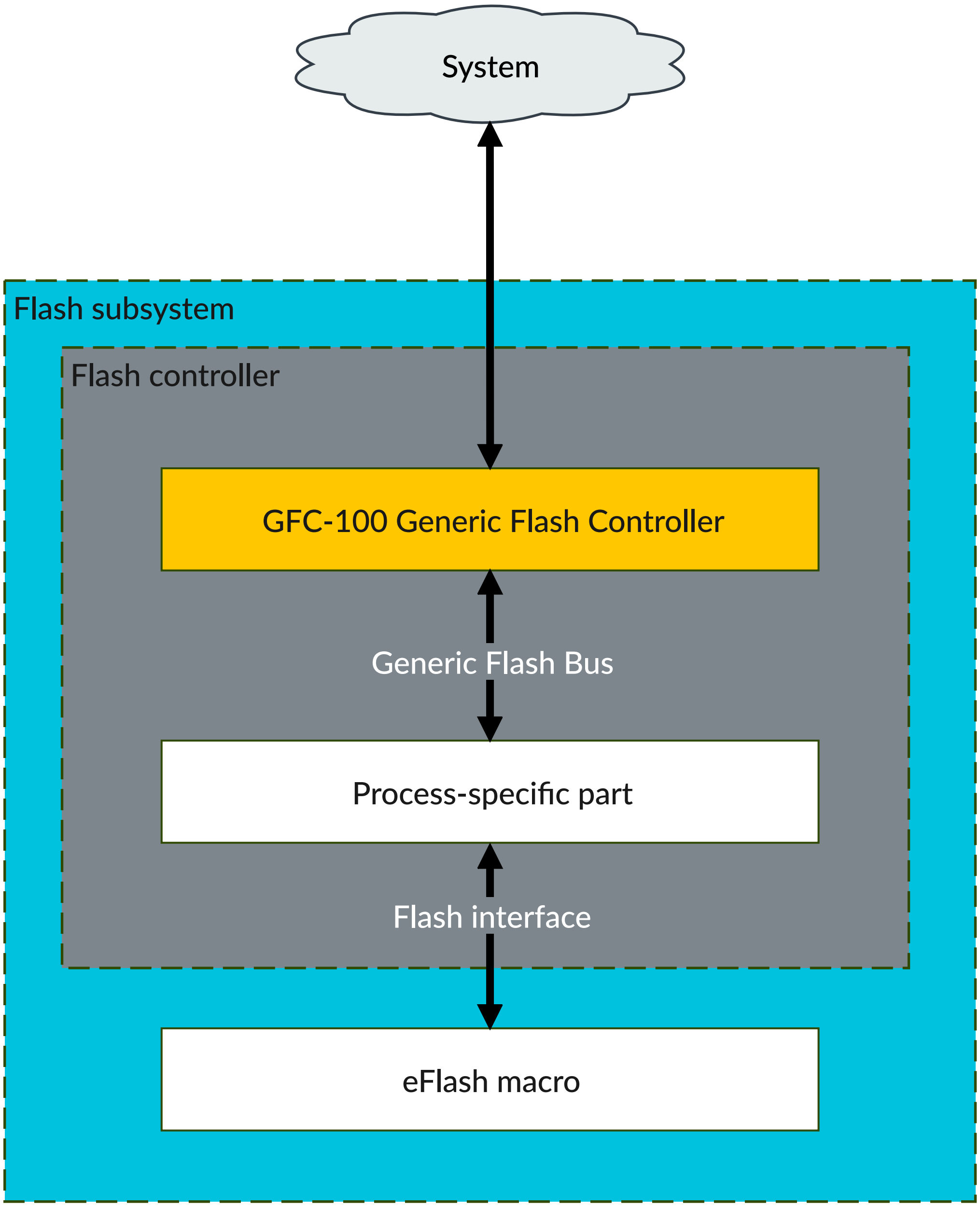
GFC-100 provides the functions that relate only to services for the system side of the Flash controller. GFC-100 cannot communicate directly with the eFlash macro. Therefore, GFC-100 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

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 3-7: GFC-100 in a Flash controller implementation



3.6.1 Features of GFC-100

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

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*

3.7 PCK-600 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)

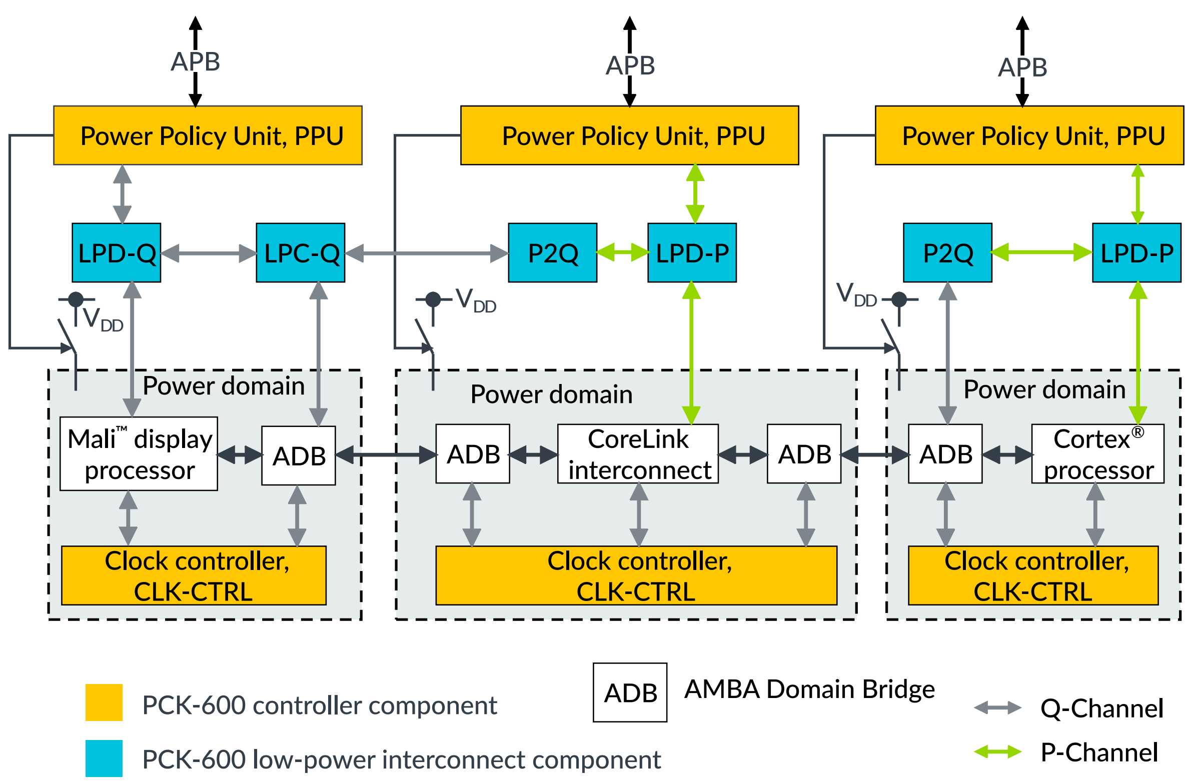
The CLK-CTRL component provides *High-level Clock Gating* (HCG) for a single clock domain.

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 PCK-600 components to manage three power domains. The PCK-600 components are shown in orange and blue.

Figure 3-8: Example system that contains PCK-600



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*

3.8 CG092 AHB Flash Cache

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

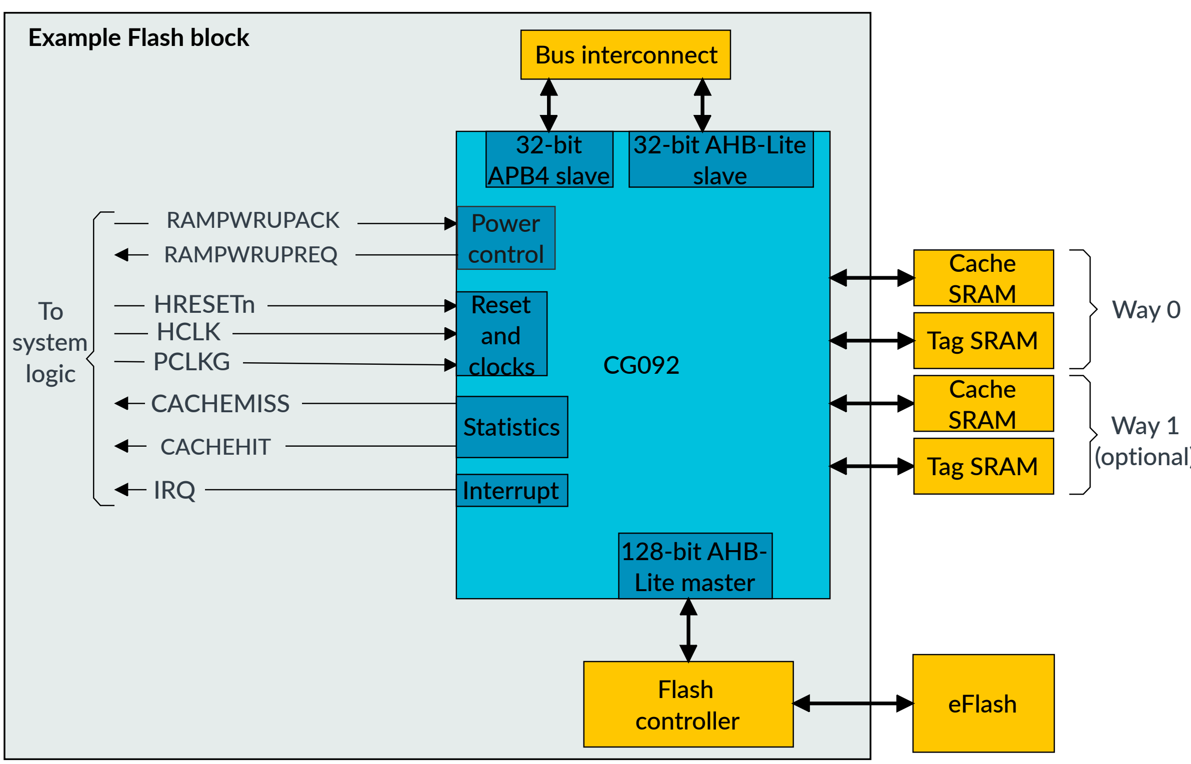
The CG092 is a simple cache for on-chip *embedded Flash* (eFlash). The CG092 design is optimized for fetching Cortex®-M3 or Cortex®-M4 instructions directly from an eFlash. The main benefit of the CG092 is improved power efficiency, but there are also improvements in code fetching performance.

Note

If the Flash controller is modified to fit, the AHB Flash Cache can also be used with external eFlash.

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

Figure 3-9: Example eFlash implementation



3.8.1 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 powerup 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 must not be invalidated on wake up. The software can therefore save energy by avoiding invalidating 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.

For more information, see the AHB Flash Cache documentation set:

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

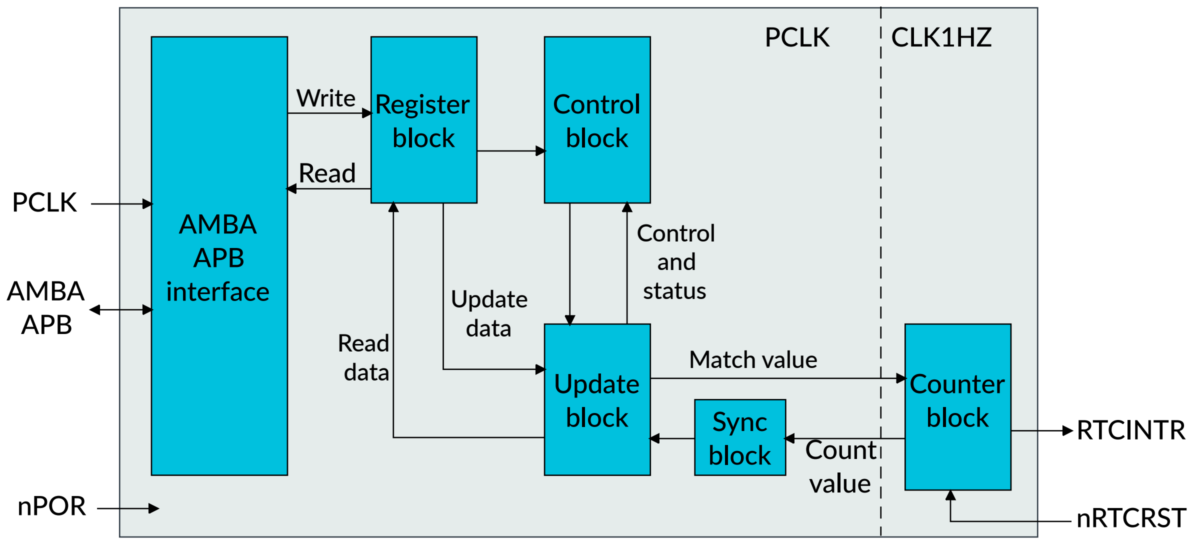
3.9 Real Time Clock

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

About Real Time Clock

The following figure shows the RTC block diagram.

Figure 3-10: RTC block diagram



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

Features of the RTC

The features of the RTC are:

- Compliance with the AMBA® 2 APB Specification for easy integration into SoC implementation. See the *AMBA® APB Protocol Specification*.
- 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.

For more information, see the RTC documentation:

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

Appendix A Revisions

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

Table A-1: Issue 0000-00

Change	Location
First release	-

Table A-2: Differences between issue 0000-00 and issue 0000-01

Change	Location
Removed references to *True Random Number Generator*	Throughout document
Changed branding of SSE-050 Subsystem from CoreLink™ to Corstone™.	Throughout document
Removed section on bus naming convention.	3.4 SIE-200 System IP for Embedded on page 21