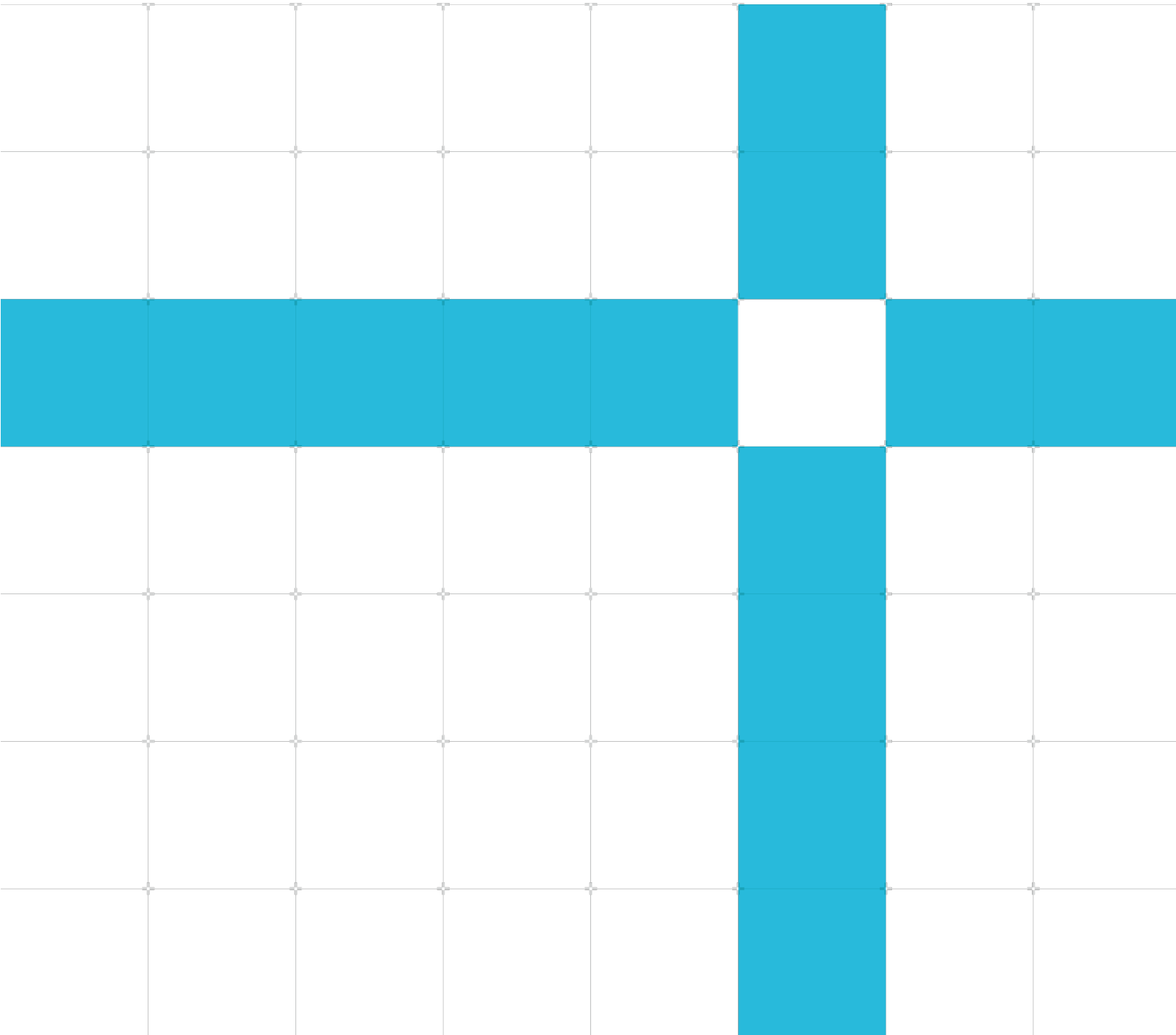
Arm® Corstone™-300 Foundation IP Technical Overview

Revision: r0p0

EAC

Non-Confidential



Arm® Corstone™-300 Foundation IP Technical Overview

Release information

Document history

Issue	Date	Confidentiality	Change
01	08 May 2020	Non-Confidential	First release for r0p0 EAC.

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

1 Preface

1.1 About this document

This Technical Overview is for the Arm® Corstone™-300 Foundation IP. It describes Corstone™-300 Foundation IP and gives a summary of the included products.

1.2 Product revision status

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

rm
: Identifies the major revision of the product, for example, r1.

pn
: Identifies the minor revision or modification status of the product, for example, p2.

1.3 Intended audience

This book is written for hardware or software engineers who want an overview of the components and functionality in Corstone™-300 Foundation IP.

1.4 Conventions

The following subsections describe conventions used in Arm documents.

1.4.1 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.

1.4.2 Typographical conventions

Convention	Use
italic	Introduces 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` **`bold`**	Denotes language keywords when used outside example code.
`monospace underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	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.
	This represents a recommendation which, if not followed, might lead to system failure or damage.
	This represents a requirement for the system that, if not followed, might result in system failure or damage.
	This represents a requirement for the system that, if not followed, will result in system failure or damage.
	This represents an important piece of information that needs your attention.
	This represents a useful tip that might make it easier, better or faster to perform a task.
	This is a reminder of something important that relates to the information you are reading.

1.5 Additional reading

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

- Arm® Corstone SSE-300 Example Subsystem Technical Reference Manual (101773).
- 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® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual (101526).
- Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual (101150).
- Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual (101375).
- Arm® CoreLink™ NIC-400-Lite Network Interconnect Technical Reference Manual (100824).
- Arm® CoreLink™ ADB-400 AMBA® Domain Bridge User Guide (DUI 0615).
- Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual (101059).
- Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual (101484).
- Arm® CoreLink™ CG092 AHB Flash Cache Technical Reference Manual (DDI 0569).
- Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual (DDI 0224).
- Arm® True Random Number Generator (TRNG) Technical Reference Manual (100976).
- Arm® Cortex®-M3 Processor Technical Reference Manual (100165).
- Arm® Cortex®-M55 Processor Technical Reference Manual (101051).
- Arm® v8-M Architecture Reference Manual (DDI 0553).

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

- Arm® Corstone™ SSE-300 Example Subsystem Configuration and Integration Manual (101774).
- 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™ SIE-300 AXI5 System IP Configuration and Integration Manual (101527).
- Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual (101151).
- Arm® CoreLink™ XHB-500 Bridge Configuration and Integration Manual AXI5 to AHB5 bridge and AHB5 to AXI5 bridge (101376).

- Arm® CoreLink™ NIC-400-Lite Network Interconnect Integration Manual (100825).
- Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual (101060).
- Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual (101485).
- Arm® CoreLink™ CG092 AHB Flash Cache Configuration and Integration Manual (DIT 0065B).
- Arm® True Random Number Generator (TRNG) Configuration and Integration Manual (100977).

See www.arm.com/cmsis for embedded software development resources including the Cortex Microcontroller Software Interface Standard (CMSIS).

See Arm Mbed™ platform, www.mbed.com for information on the Mbed tools including Mbed OS and online tools.

1.6 Feedback

Arm welcomes feedback on this product and its documentation.

1.6.1 Feedback on this product

If you have any comments or suggestions about this Development product, send and email to **support-subsystem-iot@arm.com** 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.

1.6.2 Feedback on content

If you have comments on content, send an e-mail to errata@arm.com and give:

- The title Arm® Corstone™-300 Foundation IP Technical Overview.
- The number 101772_0000_00.
- 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.

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.

2 Introduction

This chapter gives an overview of Arm® Corstone™-300 Foundation IP and its features.

2.1 About Corstone™-300 Foundation IP

Corstone™-300 Foundation IP makes an ideal starting point for creating Internet of Things (IoT) System on Chip (SoC) designs based on the Cortex®-M55 processor cores.

Corstone-300 example subsystems are configurable, and modifiable, and pre-integrate cores. Corstone™ SSE-300 Example Subsystem pre-integrates security IP with the most relevant Arm CoreLink and Arm CoreSight components.

2.2 Corstone™-300 Foundation IP components

Corstone-300 grants licenses to the following subsystems, security IP and system IP:

- Subsystems
  - Arm® Corstone™ SSE-300 Example Subsystem

    The SSE-300 is a collection of pre-assembled elements to use as the basis of an IoT SoC. SSE-300 provides a high-performance and low-power computing subsystem for Cortex-M55 processor cores. You can use it as the foundation of a secure system because of system-level support for TrustZone technology.
  - Arm® 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.
- Security and System IP
  - Arm® CoreLink™ SIE-200 System IP for Embedded

    SIE-200 is a collection of interconnect, peripheral, and TrustZone components for use with a processor that complies with the Armv8-M architecture.
  - Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded

    SIE-300 provides a set of configurable AXI5 security-aware components that comply with the TrustZone for Armv8-M processor architecture in addition to an SRAM controller, clock synchronizing bridges and an access control gate.
  - Arm® CoreLink™ PCK-600 Power Control Kit

    PCK-600 provides a set of configurable RTL components that provide a power control methodology. The components use the Arm Q-Channel and P-Channel low-power interfaces.
  - Arm® CoreLink™ XHB-500 Bridge

XHB-500 Bridge consists of an AMBA® AXI5 to AHB5 bridge and an AHB5 to AXI5 bridge.
The AXI5 to AHB5 bridge translates AXI5 transactions into the corresponding AHB5 burst transfers. The bridge has an AXI5 slave interface and an AHB5 master interface.
The AHB5 to AXI5 bridge translates AHB5 transfers into the corresponding AXI5 transactions. The bridge has an AHB5 slave interface and an AXI5 master interface.

- Arm® CoreLink™ NIC-400-Lite Network InterConnect

  NIC-400-Lite enables to create a complete high performance, optimized, and AMBA®-compliant network infrastructure.

- Arm® CoreLink™ ADB-400 AMBA® Domain Bridge

  ADB-400 is an asynchronous bridge between two components or systems that can be in different power, clock, or voltage domains. The bridge supports being put into clock or power isolation states to enable low-power system design.

- Arm® 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.

- Arm® 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.

- Arm® CoreLink™ CG092 AHB Flash Cache

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

- Arm® PrimeCell™ Real Time Clock

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

- Arm® TrustZone™ True Random Number Generator

  The True Random Number Generator (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 seeding approved deterministic random numbers.

- Arm® Socrates™

  The Socrates IP Tooling platform is an environment for the configuration of Arm IP.

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.

2.3 Using the Corstone components

The Corstone components form only part of the finished SoC. 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-300:

- Use the SSE-300 Example Subsystem as a foundation for your own IoT solution that is based around the Cortex-M55 core.
  Use the security and system IP components to add bus and controller IP to create secure TrustZone system.
- 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 Corstone-300 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 core and associated bus, debug, controller, peripherals, and interface logic supplied by Arm.

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-300.

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.

Sensors and actuators

The reference design is typically extended by adding sensors or actuator 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, 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.

For information on how to use the components that are licensed by Corstone™-300 Foundation IP, see the relevant component IP product documentation, starting with each IP product's Technical Reference Manual.

2.4 Product deliverables

The Corestone-300 product bundle (BP313) 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-300 license:

- SSE-300 Example Subsystem (NTBSS) (CG067)
- Cortex-M System Design Kit (NTBSS) (BP210)
- Cortex-M0_M0+ System Design Kit (NTBSS) (BP200)
- CoreLink SIE-200 System IP (BP300)
- CoreLink SIE-300 (BP301)
- CoreLink XHB-500 (PL417)
- (NTBSS) CoreLink GFC-200 Flash Cntl (CG094)
- CoreLink GFC-100 Flash Cntl (CG090)
- AHB Flash Cache (CG092)
- PCK-600 (PL608)
- NIC-400-Lite (PL410)
- ADB-400 Domain Bridge (PL405)
- PL031 RTC (PL031)
- True Random Number Generator (CC003)
- Socrates main product (SYSOC)

See the Arm® Corstone™-300 Foundation IP Release Note for the component versions.

2.5 Compliance

See the component ***Technical Reference Manuals*** for more details about compliance with, or implementation of, the following specifications:

- Arm architecture
- CoreSight Debug
- Advanced Microcontroller Bus Architecture

2.6 Documentation

The following documents are supplied with the Corstone-300 product bundle:

Technical Overview

The Technical Overview (TO) describes the functionality of Corstone-300.

Release Note

The Release Note describes download and installation instructions for the IP products included in Corstone-300.

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

3 Corstone™-300 Foundation IP component IP overview

This chapter describes the IP products included in the Corstone™-300 Foundation IP license.

3.1 Corstone™ SSE-300 Example Subsystem

This section is an extract from the SSE-300 Example Subsystem technical reference manual. It gives an overview of the product and its features.

For more information, see the SSE-300 Example Subsystem documentation set:

- Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual.
- Arm® Corstone™ SSE-300 Example Subsystem Configuration and Integration Manual.

3.1.1 About SSE-300 Example Subsystem

SSE-300 is a subsystem that integrates key components available from Arm that can be integrated into a larger system. SSE-300 integrates the following:

- Cortex-M CPU cores with optional MVE , FPU, DSP extensions, Caches, TCMs, and ETM. Corstone™ SSE-300 Example Subsystem supports only one Cortex M-55 processor.
- Multiple banks of System Volatile Memory, for example SRAMs. Corstone™ SSE-300 Example Subsystem supports two Volatile Memory bank.
- Memory Protection Controllers (MPC).
- Exclusive Access Monitor (EAM).
- System interconnect.
- Implementation Defined Attribution Unit (IDAU).
- CMSDK Timers and Watchdog timers.
- Timestamp-based System Timers and Watchdog timers.
- Subsystem Controllers for security and general system control.
- Power Policy Units, Clock Controller, and Low Power Interface interconnect components (PCK-600).

3.1.2 System Block Diagram

For a representative system block diagram of a Corstone™ SSE-300 Example Subsystem based IoT Subsystem, see Section 3.1 System Block Diagram in the Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual.

3.1.3 Hardware components

Arm® Corstone™ SSE-300 Example Subsystem contains the following components:

- One Arm® Cortex-M55 processor with M-Profile Vector Extension (MVE):
  - Optional Floating-Point Unit (FPU) (configurable).
  - Optional Embedded Trace Macrocell (ETM) (configurable).
  - 32kB Instruction Cache.
  - 32kB Data Cache.

For more information, see the Arm® Cortex-M55 Processor Technical Reference Manual.

- Tightly Coupled Memories (TCM):
  - 256kB ITCM.
  - 256kB DTCM.
- Secure AMBA® AXI interconnect:
  - AXI4 NIC-400 Lite interconnect.
  - AXI5 TrustZone® Memory Protection Controller (MPC).
  - AXI5 Access Control Gates (ACG).
  - AXI5 SRAM Controller (SMC) including Exclusive Access Monitor (EAM).
  - AXI5 to AHB5 bridge (XHB).
  - One Expansion AXI5 slave bus.
  - Three AXI5 Expansion master buses.
- Secure AMBA® AHB5 interconnect:
  - Advanced High-Performance Bus (AHB5) Bus Matrix.
  - AMBA® AHB5 TrustZone® Peripheral Protection Controller (PPC).
  - AMBA® AHB5 Access Control Gates (ACG).
  - AMBA® AHB5 to Advanced Peripheral Bus (APB) Bridges.
  - AMBA® APB TrustZone® Peripheral Protection Controller (PPC).
  - Expansion AHB5 master and slave buses (two each).
- Memory system:
  - AXI5 slave bus to access ITCM and DTCM memories.
  - AXI5 master bus to external code memory.
  - AXI5 master bus to external static memory.
  - AXI5 Static memory controllers.
  - Two banks of SRAM – 256kB each.
- Security components:
  - Implementation Defined Attribution Unit (IDAU).
  - Secure expansion ports.

- System Security Controller.
- System Controller.

- APB peripherals with security support:
  - One always-on Secure Watchdog in the SLOWCLK domain.
  - One always-on general-purpose timer with configurable security in the SLOWCLK domain.
  - One always-on Timestamp based Secure Watchdog in the CNTCLK domain.
  - One always-on Timestamp based Non-Secure Watchdog in the CNTCLK domain.
  - Four Timestamp based timers with configurable security in the CNTCLK domain.
- Power control components:
  - Power Dependency Control Matrix (PDCM).
  - PCK-600 components:
    - Power Policy Units (PPU).
    - Low Power P-channel distributors (LPD_P).
    - Low Power Q-channel distributors (LPD_Q) and combiners (LPC_Q).
    - P-channel to Q-channel converters (P2Q).
    - Clock Controllers (CLK-CTRL).
  - Cortex-M55 External Wake-up Interrupt Controller (EWIC).
- Example expansion integration:
  - AMBA® AHB5 to AXI5 bridge (HXB).
  - AMBA® AHB5 to AHB5 and APB asynchronous bridge.
  - AMBA® AHB5 to APB synchronous bridge.
  - System Timestamp generator (generic counter).
  - PCK-600 components.
  - Arm® CoreSightTM DAP-Lite 2.
  - Cortex-M55 Trace Port Interface Unit (TPIU).
  - Cortex-M55 MCU ROM table.
  - Debug Timestamp generator.

3.1.4 Configuration Options

The Corstone™ SSE-300 Example Subsystem is configurable, thus a system based on this specification can scale across the performance, power, and area requirement of the market.

3.2 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.

3.2.1 About Cortex-M System Design Kit

The Cortex-M System Design Kit helps you design products using Arm Cortex-M3 and Cortex-M4 processors.

The design kit contains the following:

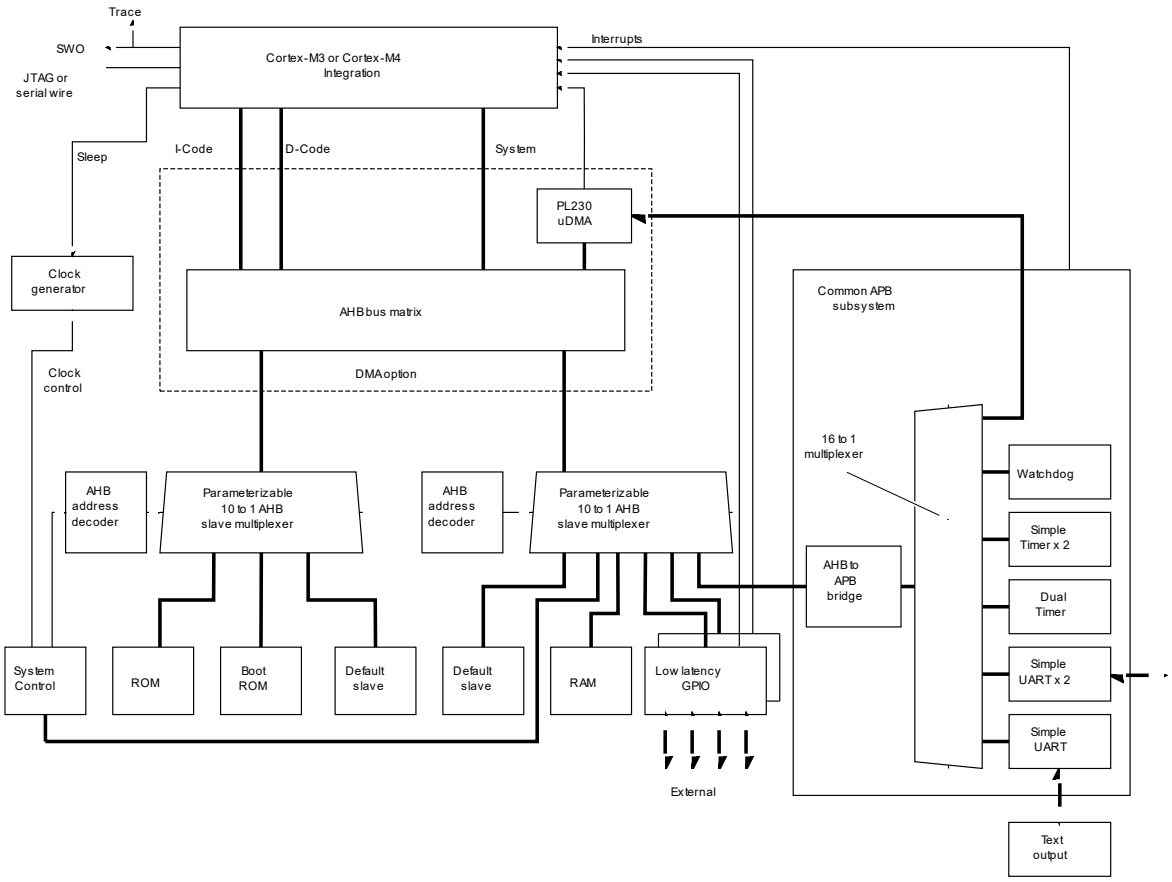
- A selection of AHB-Lite and APB components, including several peripherals such as GPIO, timers, watchdog, and UART. These components are used in the CMSDK example system, but you can also use the components to create your own custom system.
- An example system for supported processor products.
- Example synthesis scripts for the example system.
- 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).
- Documentation including:
  - Arm® Cortex® -M System Design Kit Technical Reference Manual.
  - Arm® Cortex® -M0 and Cortex® -M0+ System Design Kit Example System Guide.
  - Arm® Cortex® -M System Design Kit Example System Guide.

For details of the CMSDK components, see the Arm® Cortex® -M System Design Kit Technical Reference Manual.

3.2.1.1 Example system

The following figure shows the block diagram of the CMSDK example system:

Figure 3-1 CMSDK example system




Note

The µDMA Controller (PL230) is not included in the Corstone™-300 Foundation IP license and, if instantiated, must be licensed separately. See the Arm® PrimeCell µDMA Controller (PL230) Technical Reference Manual for more information.

3.2.1.2 Components

The CMSDK example system consists of the following components and models:

- Basic AHB-Lite components.
- APB components.
- Advanced AHB-Lite components.
- Behavioral memory models.

3.2.1.3 Cortex-M Software Design Kit software

The Cortex-M System Design Kit includes the following software:

- CMSIS-compliant drivers.

- Device-specific header files, startup code, and example drivers including retargeting code for the `printf()` and `puts()` functions.
- Platform hardware adaptation layer code that is required in addition to the open-source code and generic Cortex-M processor header files.
- Mbed OS driver support.
  Extra Cortex-M code is available on the Mbed website.
- Shell scripts to sync, build, and run the software.

3.3 Cortex-M0 and M0+ System Design Kit

This section is an extract from the CM0SDK 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.

3.3.1 About Cortex-M0 and M0+ System Design Kit

The Cortex-M0 and Cortex-M0+ System Design Kit provides:

- An example system-level design for the Arm Cortex-M0 and Cortex-M0+ processors.
- Reusable AMBA components for system-level development from the CMSDK.

For information on the AMBA components that the design kit uses, see the Arm® Cortex®-M System Design Kit Technical Reference Manual.

3.3.1.1 About the example system

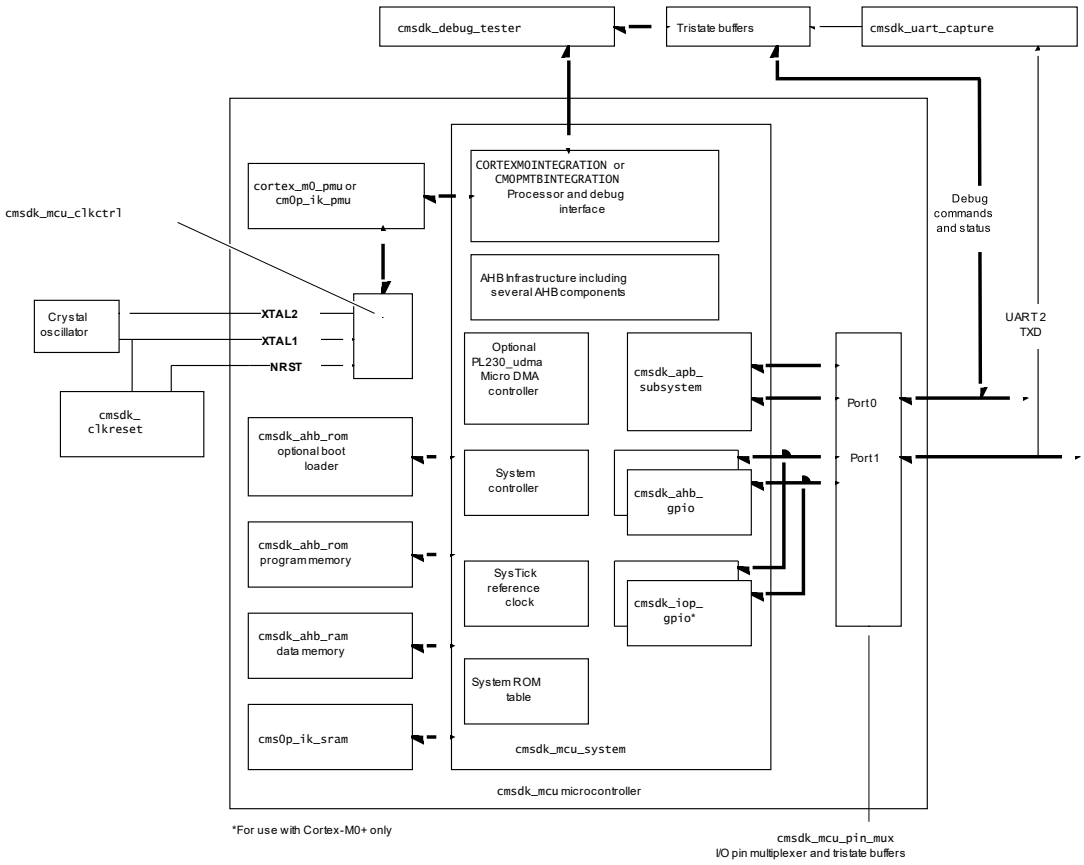
The Arm Cortex-M0 and M0+ System Design Kit Example System Guide describes an example system for the Cortex-M0 and Cortex-M0+ processors.

The Figure 3-1 CMSDK example system in section 3.2.1.1 Example system shows the example system block diagram. The following figure shows the testbench of the example system.

The example system is a simple microcontroller design that contains the following:

- A single Cortex-M0 or Cortex-M0+ processor.
- Internal program memory.
- SRAM data memory.
- Boot loader.
- The following peripherals:
  - Several timers.
  - General-Purpose input/output (GPIO).
  - Universal Asynchronous Receiver Transmitter (UART).
  - Watchdog timer.
- Debug connection.

Figure 3-2 CMSDK example system testbench




Note

The optional µDMA Controller (PL230) is not included in the Corstone™-300 Foundation IP license and, if instantiated, must be licensed separately. See the Arm® PrimeCell µDMA Controller (PL230) Technical Reference Manual for more information.

3.3.1.2 Cortex-M0 and Cortex-M0+ software

The Cortex-M0 and M0+ System Design Kit products include the following software:

- CMSIS-compliant drivers.
- Device-specific header files, startup code, and example drivers including retargeting code for the `printf()` and `puts()` functions.
- Platform hardware adaptation layer code that is required in addition to the open-source code and generic Cortex-M processor header files.
- Mbed OS driver support.
  Extra Cortex-M0 and Cortex-M0+ code is available on the Mbed website.
- Shell scripts to sync, build, and run the software.

3.4 CoreLink SIE-200 System IP

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.

3.4.1 About CoreLink SIE-200 System IP

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 and the AMBA 5 AHB5 protocol.

The SIE-200 components are used in the SSE-300 product, but you can also use the SIE-200 components to create your own custom system.

The CoreLink SIE-200 System IP for Embedded consists of the following components and models that support the AHB5 standard:

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

3.5 CoreLink SIE-300 AXI5 System IP

This section is an extract from the SIE-300 technical reference manual. It gives an overview of the product and its features.

For more information, see the SIE-300 documentation set:

- *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual.*
- *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Configuration and Integration Manual.*

3.5.1 About CoreLink SIE-300 AXI5 System IP

The SIE-300 AXI5 System IP for Embedded provides a set of configurable AXI5 security-aware components. The components comply with the TrustZone for Armv8-M processor architecture and can protect peripherals and memories that are unaware of security, so that a peripheral or memory is only accessible to trusted software. The SIE-300 also provides an AXI5 SRAM controller, clock synchronizing bridges and an access control gate.

The SIE-300 consists of the following components:

Master Security Controller (MSC)

The MSC acts as security gate for AXI transactions, and it can transform the security attribute.

Memory Protection Controller (MPC)

The MPC acts as security gate for AXI transactions that target a memory interface. The security checks operate on block or page level, and are programmable by using the APB slave interface.

Peripheral Protection Controller (PPC)

The PPC gates AXI5 transactions to, and responses from, peripherals when a security violation occurs.

Access Control Gate (ACG)

The ACG component can be placed on a clock or power domain boundary to pass or block AXI5 transactions whenever the downstream component cannot accept the transaction, or is explicitly asked not to do so. The transaction is latched internally and the ACG generates automatic responses when necessary.

Sync-Down Bridge (SDB)

The SDB synchronizes AXI5 interfaces where the upstream side is faster than the downstream side and the clocks are synchronous, in phase and have an N:1 frequency ratio.

Sync-Up Bridge (SUB)

The SUB synchronizes AXI5 interfaces where the upstream side is slower than the downstream side and the clocks are synchronous, in phase, and have a 1:N frequency ratio.

SRAM Memory Controller (SMC)

The SMC enables on-chip synchronous RAM blocks to attach to an AXI5 interface. The SMC supports 32, 64, 128, or 256-bit SRAM with byte writes.

3.6 CoreLink 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.

3.6.1 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 QChannel 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 PChannel 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.

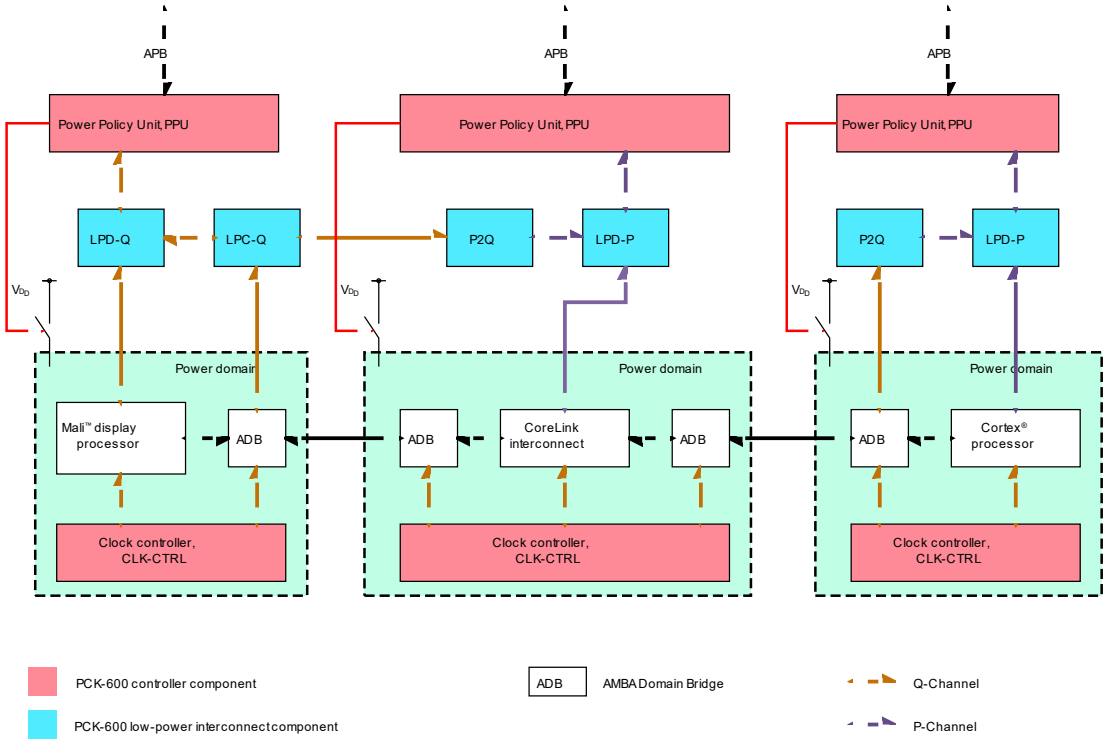
P-Channel to Q-Channel Converter (P2Q)
: The P2Q component converts a P-Channel to a Q-Channel.

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

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 3-3 Example system that contains PCK-600



3.7 CoreLink XHB-500 Bridge

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

For more information, see the XHB-500 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.*

3.7.1 About CoreLink XHB-500 Bridge

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.

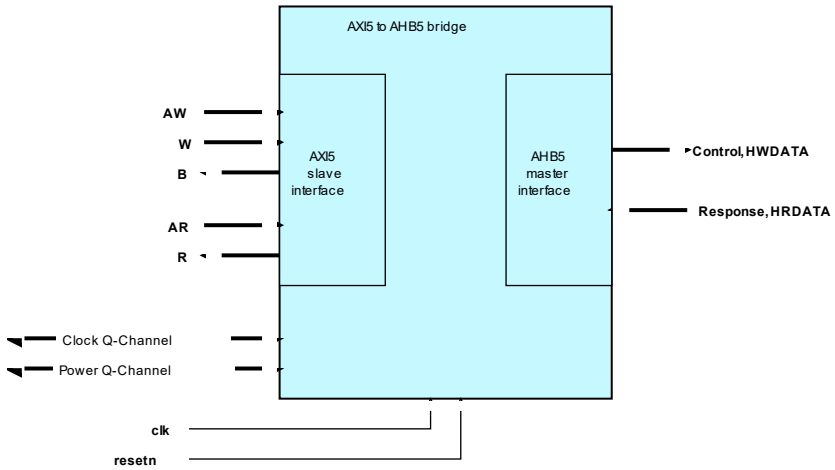
The AHB5 to AXI5 translates AHB5 transfers into the corresponding AXI transactions. The bridge has an AHB5 slave interface and an AXI5 master interface.

3.7.1.1 AXI5 to AHB5 bridge overview

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

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

Figure 3-4 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.

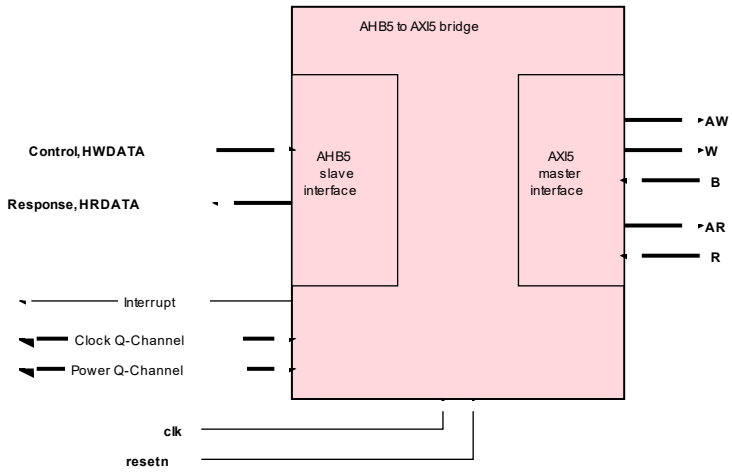
The bridge does not support endian conversion.

3.7.1.2 AHB5 to AXI5 bridge overview

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

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

Figure 3-5 AHB5 to AXI5 bridge interfaces



The main features are:

- Single power domain.
- Single clock domain.
- Configurable data width.
- AHB5 slave interface features:
  - AHB5 protocol support.
  - Fixed address width.
  - Registered or unregistered interface.
  - Support for early write response
  - Supports all burst types.
- AXI5 master interface features:
  - AXI5 support.
  - Fixed address width.
  - Registered or unregistered interface.
  - RAW hazard checking for early write response.
- Buffered write error interrupt.
- Q-Channel interface for clock control.
- Q-Channel interface for power control.

The bridge does not support endian conversion.

3.8 CoreLink NIC-400-Lite Network InterConnect

This section is an extract from the NIC-400 Lite technical reference manual. It gives an overview of the product and its features.

For more information, see the NIC-400 Lite documentation set:

- *Arm® CoreLink™ NIC-400-Lite Network Interconnect Technical Reference Manual.*
- *Arm® CoreLink™ NIC-400-Lite Network Interconnect Integration Manual.*

3.8.1 About CoreLink NIC-400 Lite Network Interconnect

The CoreLink NIC-400 Lite Network Interconnect is highly configurable and enables you to create a complete high performance, optimized, and AMBA-compliant network infrastructure.

There are many possible configurations for the CoreLink NIC-400 Lite Network Interconnect. They allow for any topology of up to 6 slave interfaces and 12 master interfaces. Bridges are not counted against the limits of 6 slave interfaces and 12 master interfaces.

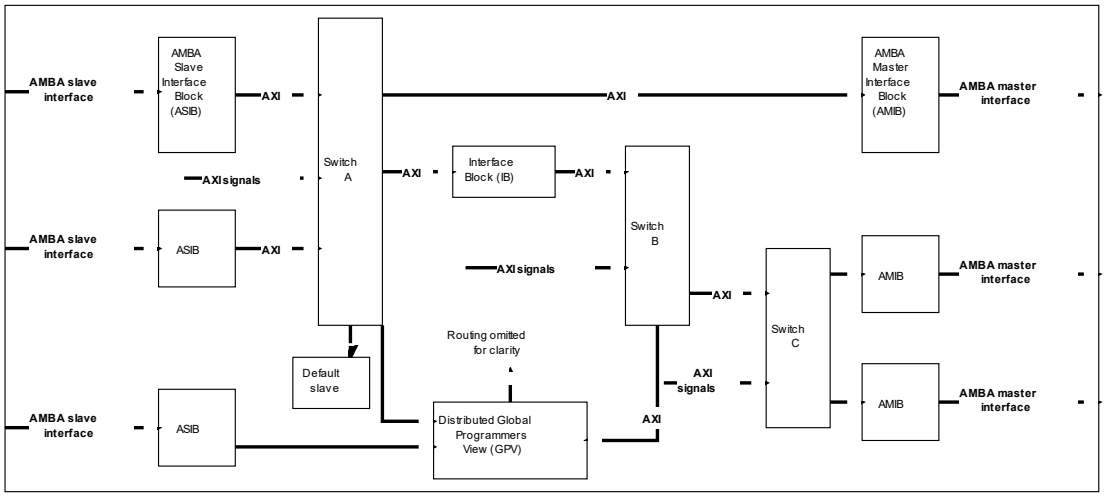

Note

For configurations containing multiple bridges, the total number of interfaces cannot be more than 64 slave interfaces and 128 master interfaces.

The NIC-400 Lite configuration can consist of many switches and topology options. The following figure shows a top-level block diagram of the NIC-400 Lite that contains:

- Multiple switches.
- Multiple ***AMBA Slave Interface Blocks*** (ASIBs).
- Multiple ***AMBA Master Interface Blocks*** (AMIBs).

Figure 3-6 NIC-400 Lite block diagram



The NIC-400-Lite is designed for applications ranging from simple single cores systems based on Cortex-M55 or Cortex®-M7 processors to more complex multi-core SoCs. For an enhanced feature set, upgrade to the NIC-400 license.

3.9 CoreLink ADB-400 AMBA Domain Bridge

This section is an extract from the ADB-400 user guide. It gives an overview of the product and its features.

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

- *Arm® CoreLink™ ADB-400 AMBA® Domain Bridge User Guide*.

3.9.1 About CoreLink ADB-400 AMBA Domain Bridge

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

The ADB-400 supports:

- An optional configurable destination register for the payload of each channel.
- Simple reset requirements.
- A power management interface.
- *Dynamic Voltage and Frequency Scaling* (DVFS).
- *Quality of Service* (QoS) *Virtual Network* (QVN).
- Clock status indication.

The ADB-400 consists of a slave domain and a master domain. The slave domain received transfers from the AMBA® master and the master domain transmits transfers to an AMBA® slave.


Note

The ADB-400 does not perform protocol translation.

3.10 CoreLink 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*.

3.10.1 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 (eFlash) 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.

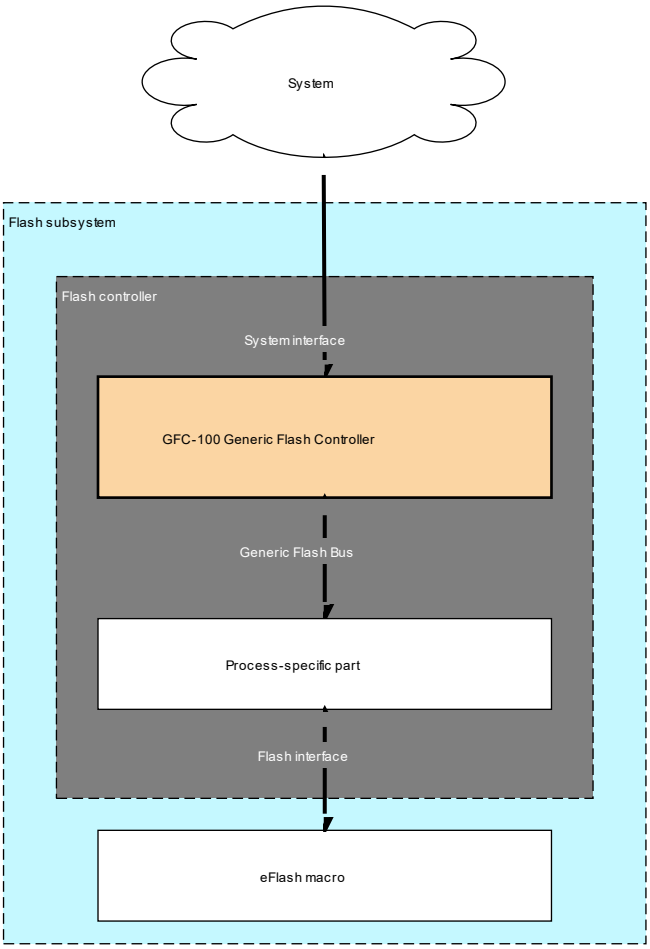
GFC-100 provides the functions that relate only to services for the system side of the Flash controller. GFC-100 cannot communicate directly with the eFlash macro. Therefore, GFC-100 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

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 3-7 GFC-100 in a Flash controller implementation



3.10.2 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.

3.11 CoreLink 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*.

3.11.1 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.

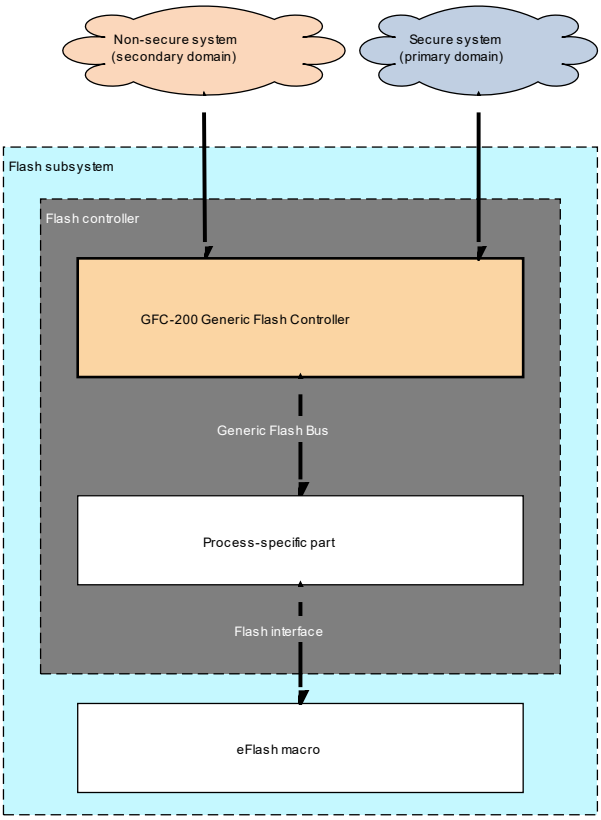
The GFC-200 provides functions that relate only to services for the system side of the Flash controller. The GFC-200 cannot communicate directly with the eFlash macro. Therefore, the GFC-200 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

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 Nonsecure 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 3-8 GFC-200 in a Flash controller implementation



3.11.2 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 attached 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.

3.12 CoreLink 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.

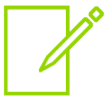
For more information, see the CG092 documentation set:

- Arm® CG092 AHB Flash Cache Technical Reference Manual.
- Arm® CG092 AHB Flash Cache Configuration and Integration Manual.

3.12.1 About CG092

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

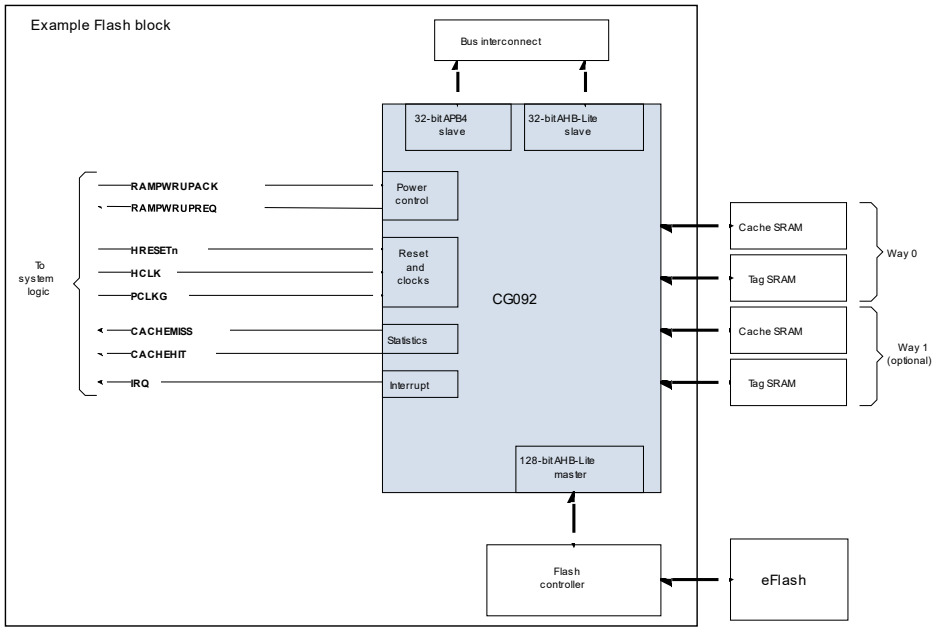
The CG092 is a simple cache for on-chip embedded Flash (eFlash). The CG092 design is optimized for fetching Cortex-M3 or Cortex-M4 instructions directly from an eFlash. The main benefit of the CG092 is improved power efficiency, but there are also improvements in code fetching performance.


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 3-9 Example eFlash implementation



3.12.2 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.

3.13 PrimeCell Real Time Clock

This section is an extract from the Real Time Clock (RTC) technical reference manual. It gives an overview of the product and its features.

For more information, see the RTC documentation set:

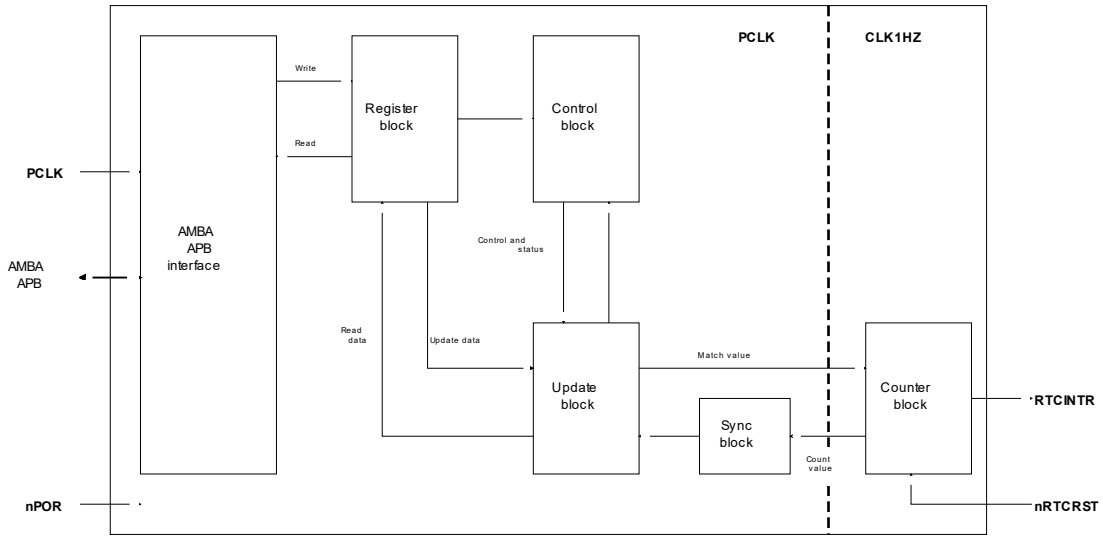
- Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual.

3.13.1 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 3-10 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.

3.13.2 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.

3.14 True Random Number Generator

This section is an extract from the True Random Number Generator (TRNG) technical reference manual. It 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.*

3.14.1 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.

3.14.2 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

Table A-1 Issue 01

Change	Location	Affects
First release for EAC.	-	-