



Arm® Corstone™-310 Foundation IP

Revision: r0p0

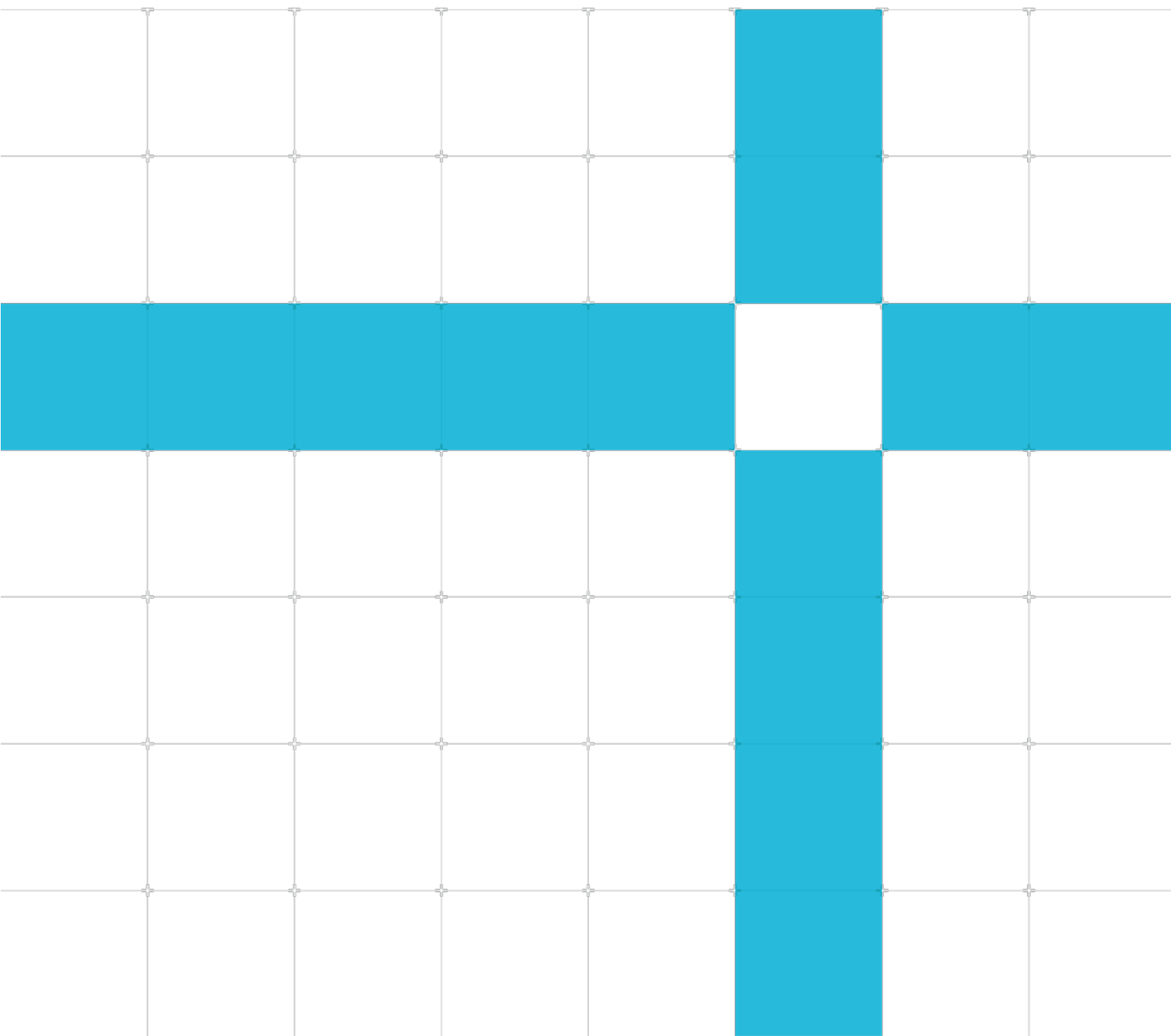
Technical Overview

Non-Confidential

Copyright © 2022 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

102792_0000_01_en



Arm® Corstone™-310 Foundation IP Technical Overview

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0000-01	09 May 2022	Non-Confidential	First release for r0p0 EAC.

Non-Confidential Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

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

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

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names

mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at <http://www.arm.com/company/policies/trademarks>.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

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 Corstone™-310 Foundation IP, 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.

We believe that this document contains no offensive terms. If you find offensive terms in this document, please email terms@arm.com.

Contents

1 Introduction	6
1.1 About this document	6
1.2 Product revision status	6
1.3 Intended audience	6
1.4 Conventions	6
1.4.1 Glossary	6
1.4.2 Typographical conventions	7
1.5 Additional reading	8
2 Overview of Corstone™-310 Foundation IP	10
2.1 Corstone™-310 Foundation IP components	10
2.2 Using the Corstone components	12
2.3 Compliance	13
2.4 Documentation	13
3 Corstone-310 component IP overview	14
3.1 Corstone™ SSE-310 Example Subsystem	14
3.1.1 About SSE-310 Example Subsystem	14
3.1.2 System block diagram	14
3.1.3 Hardware components	15
3.1.4 Configuration Options	16
3.2 Cortex-M System Design Kit	17
3.2.1 About Cortex-M System Design Kit	17
3.3 CoreLink SIE-200 System IP	18
3.3.1 About CoreLink SIE-200 System IP	18
3.4 CoreLink SIE-300 AXI5 System IP	19
3.4.1 About CoreLink SIE-300 AXI5 System IP	19
3.5 CoreLink PCK-600 Power Control Kit	20
3.5.1 About the Power Control Kit	20
3.6 CoreLink XHB-500 Bridge	22
3.6.1 About CoreLink XHB-500 Bridge	22
3.7 CoreLink NIC-400 Network InterConnect	26
3.7.1 About CoreLink NIC-400 Network Interconnect	26

3.8 CoreLink ADB-400 AMBA Domain Bridge.....	27
3.8.1 About CoreLink ADB-400 AMBA Domain Bridge	27
3.9 CoreLink GFC-100 Generic Flash Controller	27
3.9.1 About GFC-100.....	27
3.9.2 Features	29
3.10 CoreLink GFC-200 Generic Flash Controller	31
3.10.1 About the GFC-200	31
3.10.2 Features.....	32
3.11 CoreLink AHB Flash Cache	34
3.11.1 About AHB Flash Cache.....	34
3.11.2 Features of AHB Flash Cache	35
3.12 PrimeCell Real Time Clock.....	37
3.12.1 About Real Time Clock	37
3.12.2 Features of the RTC.....	37
3.13 CoreSight System-on Chip SoC-600M	39
3.13.1 About SoC-600M.....	39
3.13.2 SoC-600M features	39
3.14 CoreSight SDC-600 Secure Debug Channel	40
3.14.1 About SDC-600	41
Appendix A Revisions	42

1 Introduction

1.1 About this document

This Technical Overview is for the Arm® Corstone™-310 Foundation IP. It describes Corstone™-310 Foundation IP and gives a summary of the included products.

1.2 Product revision status

The *rm**pn* 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™-310 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
<i>italic</i>	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
<code>monospace</code>	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
<code>monospace bold</code>	Denotes language keywords when used outside example code.
<code>monospace <u>underline</u></code>	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: <pre>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></pre>
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.
 Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
 Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
 Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.
 Note	This represents an important piece of information that needs your attention.
 Tip	This represents a useful tip that might make it easier, better or faster to perform a task.
 Remember	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:

Table 1-1: Non-Confidential documents

Document	ID
Arm® Corstone SSE-310 Example Subsystem Technical Reference Manual	102778
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 Network Interconnect Technical Reference Manual	DDI 0475
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® Cortex®-M85 Processor Technical Reference Manual	101924
Arm®v8-M Architecture Reference Manual	DDI 0553
Arm® CoreSight™ System-on-Chip SoC-600M Technical Reference Manual	101883
Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual	101130
Arm® Ethos™-U55 NPU Technical Reference Manual	102420

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

Table 1-2: Confidential documents

Document	ID
Arm® Corstone™ SSE-310 Example Subsystem Configuration and Integration Manual	102779
Arm® Cortex®-M System Design Kit Example System Guide	DUI 0594
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 Network Interconnect Integration Manual	DII 0269
Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual	101060
Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual	101485

Document	ID
Arm® CoreLink™ CG092 AHB Flash Cache Configuration and Integration Manual	DIT 0065
Arm® Cortex®-M85 Integration and Implementation Manual	101925
Arm® CoreSight™ System-on-Chip SoC-600M Configuration and Integration Manual	101884
Arm® CoreSight™ SDC-600 Secure Debug Channel Configuration and Integration Manual	101131
Arm® Ethos™-U55 NPU Configuration and Integration Manual	101887

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

2 Overview of Corstone™-310 Foundation IP

Corstone™-310 Foundation IP makes an ideal starting point for creating Internet of Things (IoT) System on Chip (SoC) designs based on the Arm® Cortex® M85 processor cores.

Corstone™ SSE-310 example subsystem is configurable, and modifiable, and pre-integrate Cortex-M85 and Ethos-U55. Corstone™ SSE-310 Example Subsystem pre-integrates security IPs with the most relevant Arm CoreLink and Arm CoreSight components.

2.1 Corstone™-310 Foundation IP components

Corstone-310 grants licenses to the following subsystems, security IP, and system IP:

Subsystems

Arm® Corstone™ SSE-310 Example Subsystem

The SSE-310 is a collection of pre-assembled elements to use as the basis of an IoT SoC. SSE-310 provides a high-performance and low-power computing subsystem for the Cortex®-M85 processor and optional integration of Ethos-U55 NPU. 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 Cortex-M System Design Kit (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-aware System IPs

Arm® CoreLink™ SIE-200 System IP for Embedded

SIE-200 is a collection of security-aware 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. It also provides 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 subordinate interface and an AHB5 manager interface.

The AHB5 to AXI5 bridge translates AHB5 transfers into the corresponding AXI5 transactions. The bridge has an AHB5 subordinate interface and an AXI5 manager interface.

Arm® CoreLink™ NIC-400 Network InterConnect

NIC-400 enables to create a complete high performance, optimized, and AMBA® AXI-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® CoreSight™ SoC-600M

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

Arm® 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.

Arm® CoreLink™ GFC-200 Generic Flash Controller

GFC-200 comprises the generic part of a Flash controller in an SoC, so you can easily integrate an embedded Flash macro into your system. The GFC-200 supports accesses from two managers 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

AHB Flash Cache 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® subordinate 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.

Separately licensed IP

To provide optimum flexibility, all Cortex processors, Socrates™, and Ethos NPUs 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 form only part of the finished SoC. You must extend and customize the subsystems for your specific application requirements.

Arm provides the Total Solutions for the IoT developer platform, which includes 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.

The following examples show you some of the ways that you can use the components, which are licensed by Corstone-310:

- Use the SSE-310 Example Subsystem as a foundation for your own IoT solution that is based around the Cortex-M85 core and Ethos U55 NPU.
Use the security and system IP components to add bus and controller IP to create Secure TrustZone system.
- Use the Security-aware system IPs provided with Corstone-310 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 Ethos U55 NPU, with 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-310.

Communication interface

The endpoint must have some way of communicating with other nodes or managers in the system. This interface could be a wireless connection (for example WiFi or 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.

Arm provides a software development environment, which includes Total Solutions, Arm or GNU (GCC) compilers and debuggers, and firmware.

For information on how to use the components that are licensed by Corstone™-310 Foundation IP, see the relevant component IP product documentation, starting with the Technical Reference Manual of each IP product.

2.3 Compliance

See the relevant Technical Reference Manual of the various Corstone-310 Foundation IP components for more details about compliance with, or implementation of, the following specifications:

- Arm architecture
- CoreSight Debug
- Advanced Microcontroller Bus Architecture

2.4 Documentation

The following documents are supplied with the Corstone-310 product bundle:

Technical Overview

The Technical Overview (TO) describes the functionality of Corstone-310 components.

Release Note

The Release Note describes download and installation instructions for the IP products included in Corstone-310.



- 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-310 component IP overview

This chapter describes the IP products included in the Corstone™-310 Foundation IP license.

3.1 Corstone™ SSE-310 Example Subsystem

This section is an extract from the *SSE-310 Example Subsystem Technical Reference Manual*. It gives an overview of the product and its features.

For more information, see the SSE-310 example Subsystem documentation set:

- *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual*
- *Arm® Corstone™ SSE-310 Example Subsystem Configuration and Integration Manual*

3.1.1 About SSE-310 Example Subsystem

SSE-310 is a subsystem that integrates key components available from Arm that can be integrated into a larger system. SSE-310 integrates the following:

- Cortex-M85 processor core with optional M-class Vector Extension (MVE), Floating-Point Unit (FPU), Digital Signal Processing (DSP) extensions, Caches, Tightly Coupled Memory (TCM)s, and Embedded Trace Macrocell (ETM).
- Arm® Ethos™-U55 Neural Processing Unit (NPU)
- SSE-310 supports one Arm® Cortex-M85 processor
- SSE-310 supports two Volatile Memory (VM) banks, for example SRAMs.
- Memory Protection Controllers (MPC)
- Exclusive Access Monitor (EAM)
- System interconnect
- Implementation Defined Attribution Unit (IDAU)
- 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 component

3.1.2 System block diagram

For a representative system block diagram of a Corstone™ SSE-310 Example Subsystem based IoT subsystem, see the System block diagram section in the *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual*.

3.1.3 Hardware components

Arm® Corstone™ SSE-310 Example Subsystem contains the following components:

- One Arm® Cortex-M85 processor with M-Profile Vector extension (MVE):
 - Configurable optional Floating-Point Unit (FPU)
 - Configurable optional Embedded Trace Macrocell (ETM)
 - 32kB Instruction Cache
 - 32kB Data Cache

For more information, see the *Arm® Cortex-M85 Processor Technical Reference Manual*.
- (optional) One Ethos -U55 Neural Processing Unit
- Tightly Coupled Memories (TCM):
 - 32KB ITCM
 - 32KB DTCM
- Secure AMBA® AXI interconnect:
 - AXI4 NIC-400 interconnect
 - AXI5 TrustZone® Memory Protection Controller (MPC)
 - AXI5 Access Control Gates (ACG)
 - AXI5 Sync Down Bridge (SDB)
 - AXI5 SRAM Controller (SMC) including Exclusive Access Monitor (EAM)
 - AXI5 to AHB5 bridge (XHB)
 - One Expansion AXI5 subordinate bus
 - Three AXI5 Expansion manager 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 manager and subordinate buses (two each)
- Memory system:
 - AXI5 subordinate bus to access ITCM and DTCM memories.
 - AXI5 manager bus to external code memory
 - AXI5 manager bus to external static memory
 - AXI5 Static memory controllers
 - Two banks of SRAM – 2MB 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-M85 External Wakeup 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® CoreSight™ DAP-Lite 2
 - Trace Port Interface Unit for Cortex-M processors (TPIU-M)
 - Cortex-M85 MCU ROM table
 - Debug Timestamp generator

3.1.4 Configuration Options

The Corstone™ SSE-310 Example Subsystem is configurable, therefore 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 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
- Documentation including:
 - *Arm® Cortex®-M System Design Kit Technical Reference Manual*
 - *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 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

3.2.1.2 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
- Shell scripts to sync, build, and run the software

3.3 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*

3.3.1 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. It is designed 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-310 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 bus matrix
- AHB5 default subordinate
- AHB5 example subordinate
- AHB5 exclusive access monitor
- AHB5 GPIO
- AHB5 manager multiplexer
- AHB5 subordinate multiplexer
- AHB5 timeout monitor
- AHB5 to external SRAM interface
- AHB5 to ROM interface
- AHB5 to internal SRAM interface
- Cortex-M3/Cortex-M4 AHB5 adapter

AHB5 bridge components

- AHB5 access control gate
- AHB5 downsizer
- AHB5 to AHB5 and APB4 asynchronous bridge
- AHB5 to AHB5 sync-down bridge
- AHB5 to AHB5 low-latency sync-down bridge

- AHB5 to AHB5 synchronous bridge
- AHB5 to AHB5 sync-up bridge
- AHB5 to AHB5 low-latency sync-up bridge
- AHB5 to APB4 asynchronous bridge
- AHB5 to APB4 sync-down bridge
- AHB5 to APB4 low-latency sync-down bridge
- AHB5 upsizer

TrustZone Protection components

- AHB5 TrustZone manager security controller
- AHB5 TrustZone memory protection controller
- AHB5 TrustZone peripheral protection controller
- APB4 TrustZone peripheral protection controller

Verification components

- AHB5 File Reading Bus Manager
- Behavioral SRAM model with an AHB5 interface
- External asynchronous 8-bit SRAM model
- External asynchronous 16-bit SRAM model
- FPGA SRAM synthesizable model.
- RAM wrapper model
- ROM behavioral model
- ROM wrapper model

3.4 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*

3.4.1 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 Non-trusted software. The SIE-300 also provides an AXI5 SRAM controller, clock synchronizing bridges, and an access control gate.

The SIE-300 components are used in the SSE-310 product, but you can also use the SIE-300 components to create your own custom system.

The SIE-300 consists of the following components:

Manager 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, and it can transform the security attribute.

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.
- 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.

3.5 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*

3.5.1 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)

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Page 21 of 42

3.6 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*

3.6.1 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 subordinate interface and an AHB5 manager interface.

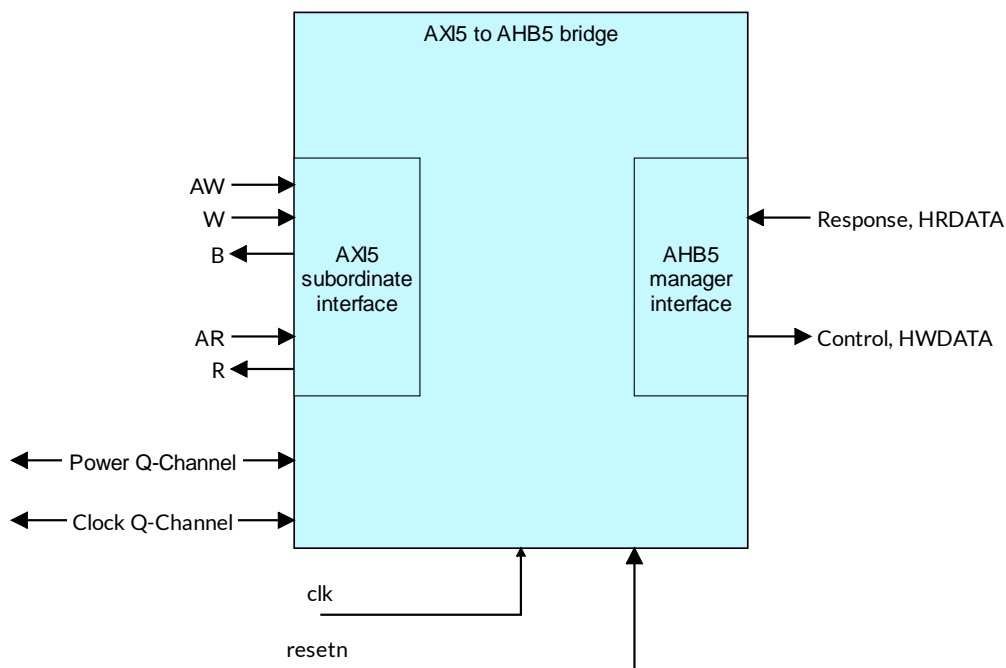
The AHB5 to AXI5 translates AHB5 transfers into the corresponding AXI transactions. The bridge has an AHB5 subordinate interface and an AXI5 manager interface.

3.6.1.1 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 3-2: AXI5 to AHB5 interfaces



The main features are:

- Single power domain
- Single clock domain
- Configurable data width
- AXI5 subordinate 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 manager 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

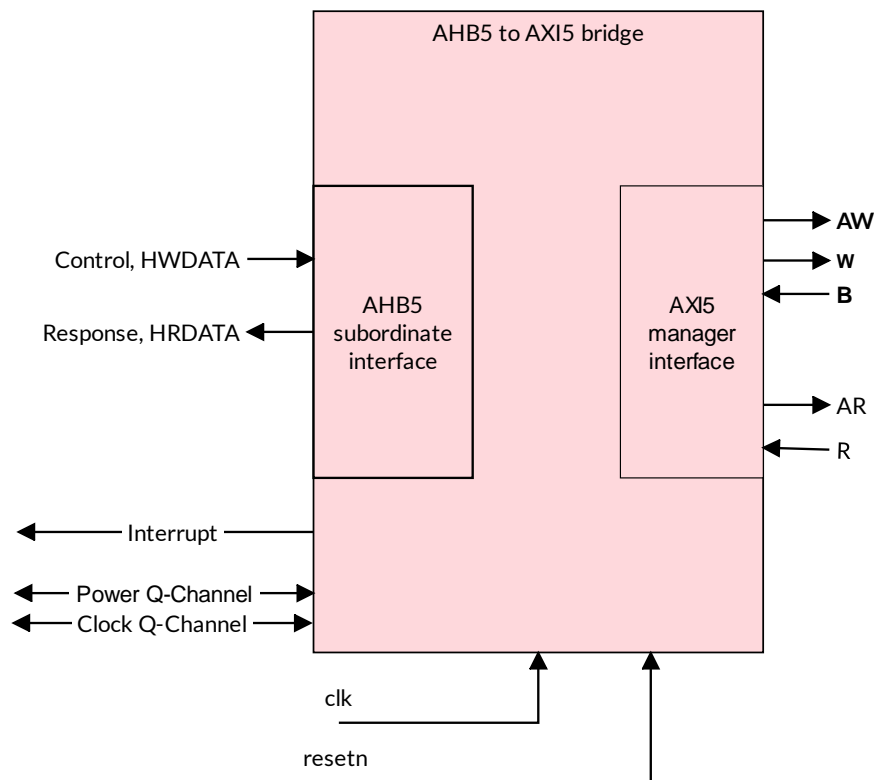
The bridge does not support endian conversion.

3.6.1.2 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 3-3: AHB5 to AXI5 bridge interfaces



The main features are:

- Single power domain
- Single clock domain
- Configurable data width
- AHB5 subordinate interface features:
 - AHB5 protocol support
 - Fixed address width
 - Registered or unregistered interface
 - Support for early write response
 - Supports all burst types
- AXI5 manager 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.

3.7 CoreLink NIC-400 Network InterConnect

This section is an extract from the *NIC-400 Technical Reference Manual*. It gives an overview of the product and its features.

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

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

3.7.1 About CoreLink NIC-400 Network Interconnect

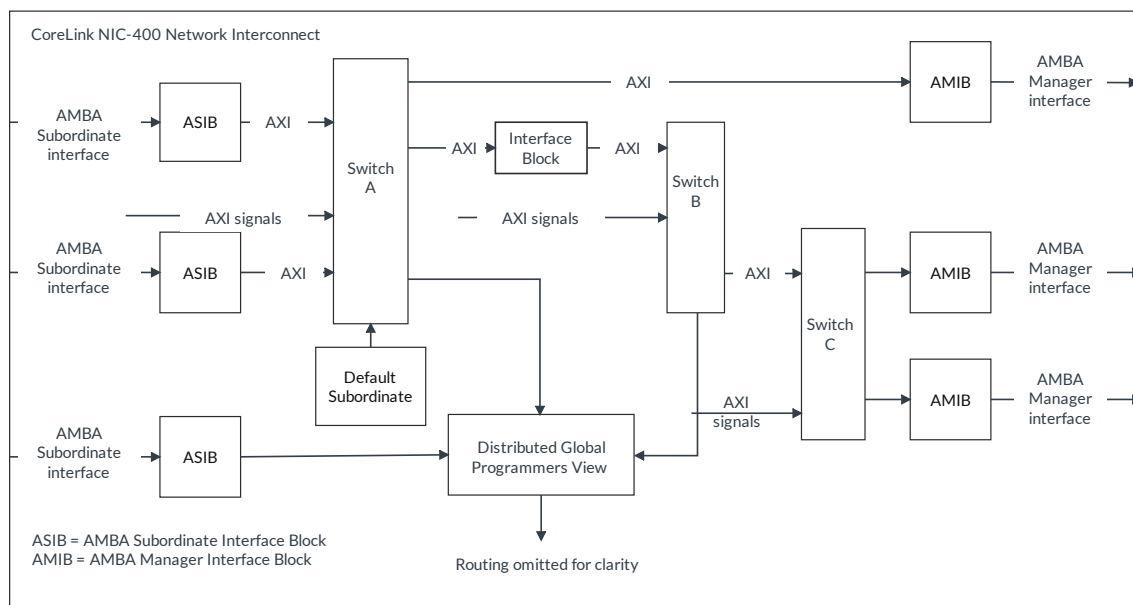
The CoreLink NIC-400 Network Interconnect is highly configurable and enables you to create a complete high performance, optimized, and AMBA-compliant network infrastructure.

There are many possible configurations for the CoreLink NIC-400 Network Interconnect. They can range from a single bridge component (for example an AHB to AXI protocol conversion bridge) to a complex interconnect that consists of up to 128 managers and 64 subordinates of AMBA protocols.

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

- Multiple switches
- Multiple AMBA Subordinate Interface Blocks (ASIBs)
- Multiple AMBA Manager Interface Blocks (AMIBs)

Figure 3-4: NIC-400 block diagram



3.8 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*

3.8.1 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 subordinate domain and a manager domain. The subordinate domain received transfers from the AMBA® manager and the manager domain transmits transfers to an AMBA® subordinate.



The ADB-400 does not perform protocol translation.

3.9 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*

3.9.1 About GFC-100

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

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The eFlash macros are produced by different foundries with different processes. The processes are determined by the foundry processes that produced the eFlash memory and 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. Therefore, GFC-100 must be integrated with a process-specific part that connects to, and communicates with, the eFlash.

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



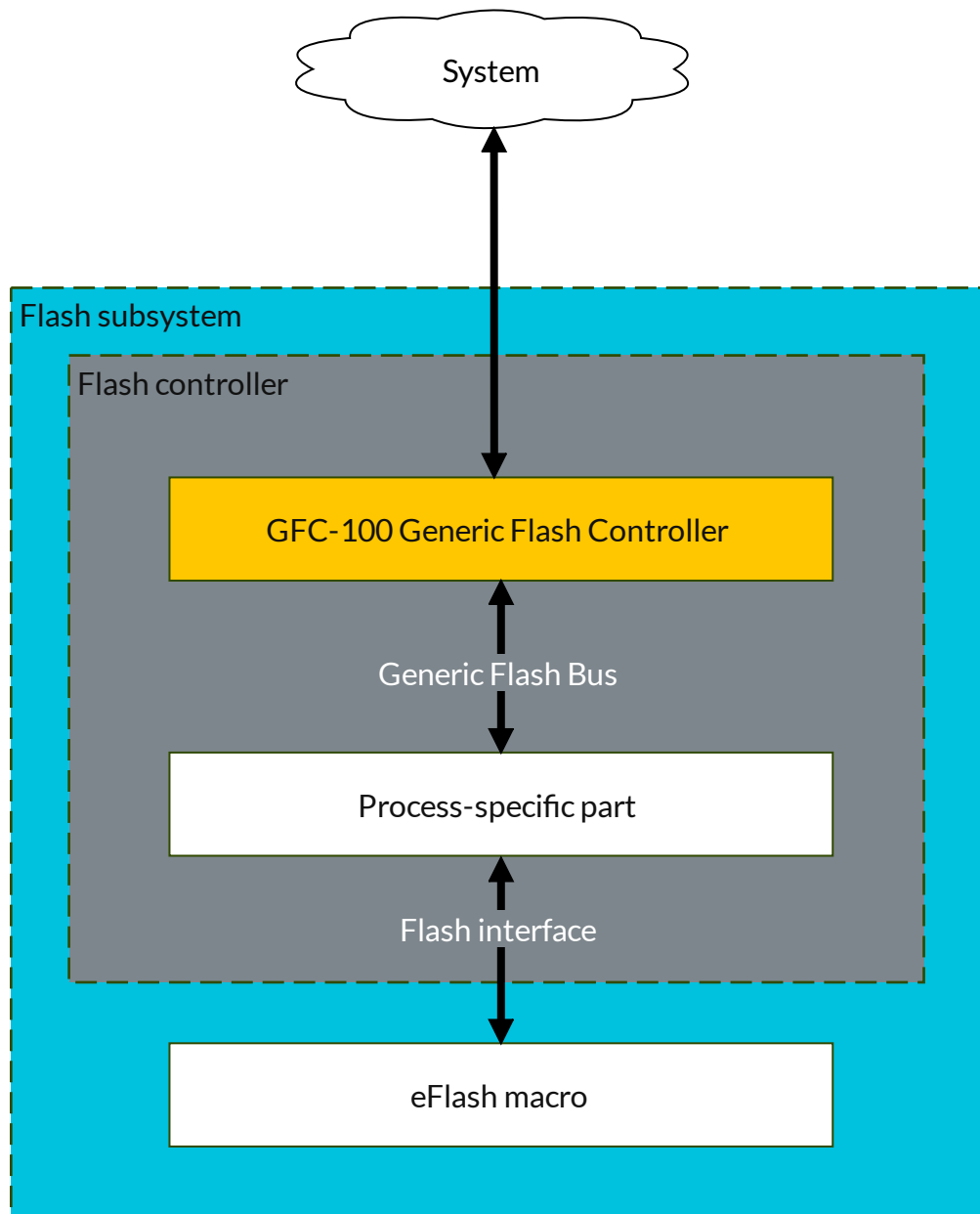
Note

The process-specific part of the controller is foundry and process specific, and therefore is not included in the SSE-310 package.

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 3-5: GFC-100 in a Flash controller implementation



3.9.2 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.

The APB Completer 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 requester 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

3.10 CoreLink GFC-200 Generic Flash Controller

This section is an extract from the *GFC-200 Generic Flash Controller 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*

3.10.1 About the GFC-200

The GFC-200 comprises the generic part of a Flash controller in a SoC. The GFC-200 enables an embedded Flash to be integrated easily into any system.

The eFlash macros are produced by different foundries with different processes. The processes are determined by the foundry processes that produced the eFlash memory and 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. Therefore, the GFC-200 must be integrated with a process-specific part that connects to, and communicates with, the eFlash.

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

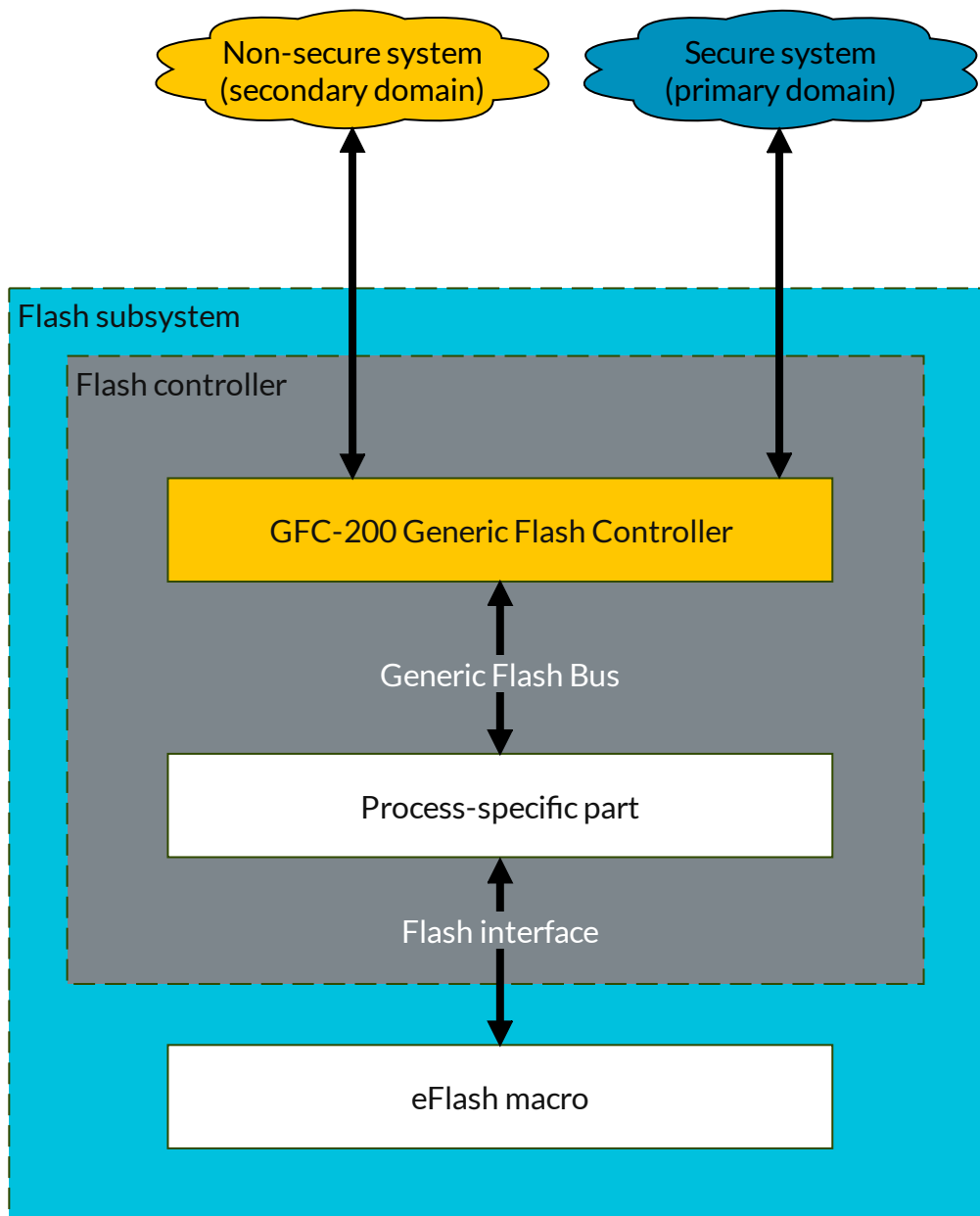


The process-specific part of the controller is foundry and process specific, and therefore is not included in the SSE-310 package.

The GFC-200 supports accesses from two managers 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



3.10.2 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

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

AMBA AHB-Lite interface:

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

Primary APB completer 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 completer 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 requester 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

3.11 CoreLink AHB Flash Cache

This section is an extract from the *AHB Flash Cache Technical Reference Manual*. It gives an overview of the product and its features.

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

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

3.11.1 About AHB Flash Cache

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

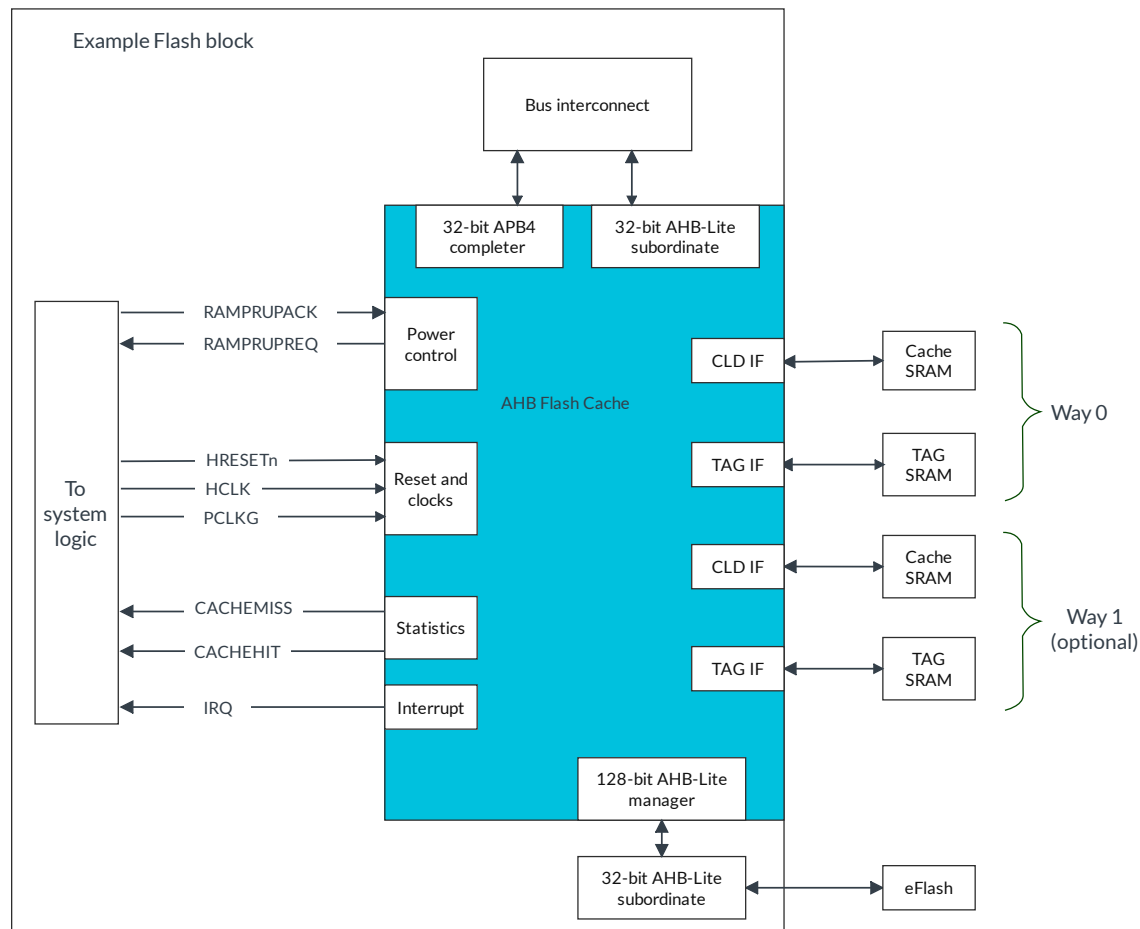
The AHB Flash Cache is a simple cache for on-chip embedded Flash (eFlash). The AHB Flash Cache design is optimized for fetching the processor instructions directly from an eFlash. The main benefit of the AHB Flash Cache is improved power efficiency, but there are also improvements in code fetching performance.



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 3-7: Example eFlash implementation



3.11.2 Features of AHB Flash Cache

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

The AHB Flash Cache 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 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 subordinate interface to the AHB manager in the system processor.
- 32-bit APB subordinate interface to the memory-mapped registers of the AHB Flash Cache.
- 128-bit AHB manager interface to the eFlash.
- Interrupt request generated on SRAM power or manual invalidation errors.
- Optional runtime 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.



An eFlash controller is not part of the AHB Flash Cache component.

3.12 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:

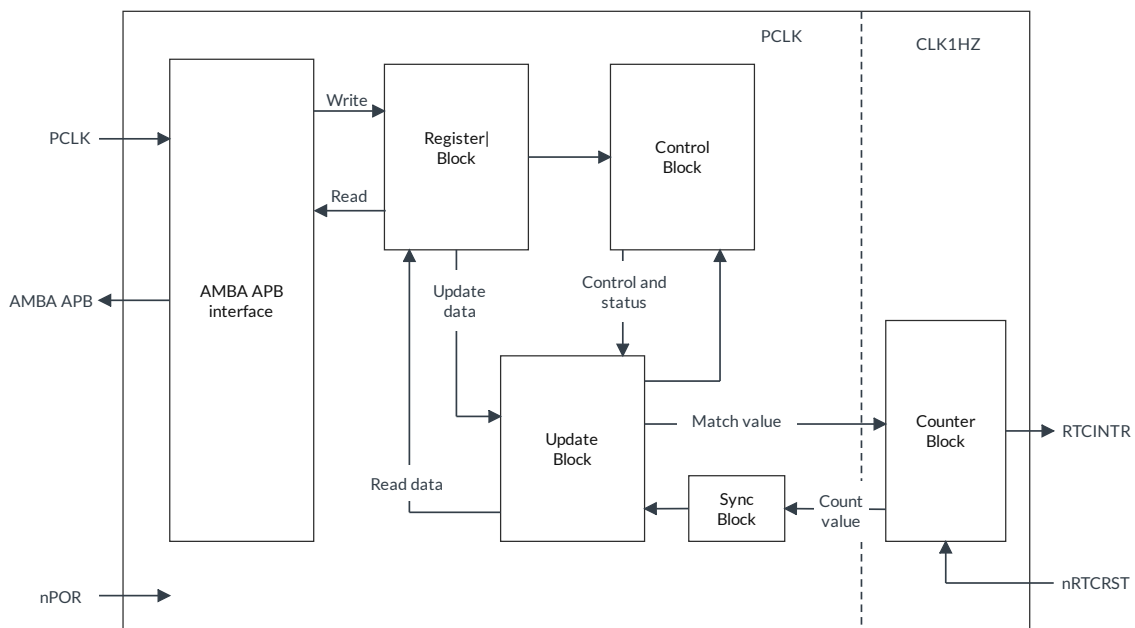
- *Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual*

3.12.1 About Real Time Clock

The RTC is an AMBA completer module that connects with the APB interface.

The following figure shows the RTC block diagram.

Figure 3-8: RTC block diagram



The RTC can be used to provide a basic alarm function or long-time base counter. These are provided 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.

3.12.2 Features of the RTC

The features of the RTC are:

- Compliance with 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

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

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

3.13 CoreSight System-on Chip SoC-600M

CoreSight System-on Chip SoC-600M (SOC-600) provides embedded debug and trace component capabilities.

For more information, see the SOC-600M documentation set:

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

3.13.1 About SoC-600M

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

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

The CoreSight™ SoC-600M bundle includes:

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



Arm Socrates supports SoC-600 component level configuration.

3.13.2 SoC-600M features

Features and capabilities that the SoC-600M provides include:

Debug

- Arm Debug Interface Architecture Specification ADIv6.0-compliant debug port.

This debug port supports JTAG and Serial Wire protocols for connection to an off-chip debugger. This connection is achieved using a low-pin-count connection that is suitable for bare-metal debug and silicon bring-up.

- *Arm CoreSight Architecture Specification v3.0* compliance enables debug over functional interfaces, suitable for application development and in-field debug without a dedicated debug interface.
- Infrastructure components supporting system identification and integration with other CoreSight IP.

Trace

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

Embedded Cross Triggering

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

Power

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

Miscellaneous

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

3.14 CoreSight SDC-600 Secure Debug Channel

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

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

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

3.14.1 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 provided 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 debug resources of the target.
2. The debugger uses the CoreSight™ ID registers and discovery process to identify the external interface of the SDC-600.
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 debugger can access connected devices.

Appendix A Revisions

Table A-1: Issue 0000-01

Change	Location	Affects
First release for EAC.	-	-