



Arm® Corstone™-320 Reference Package

Revision r0p0

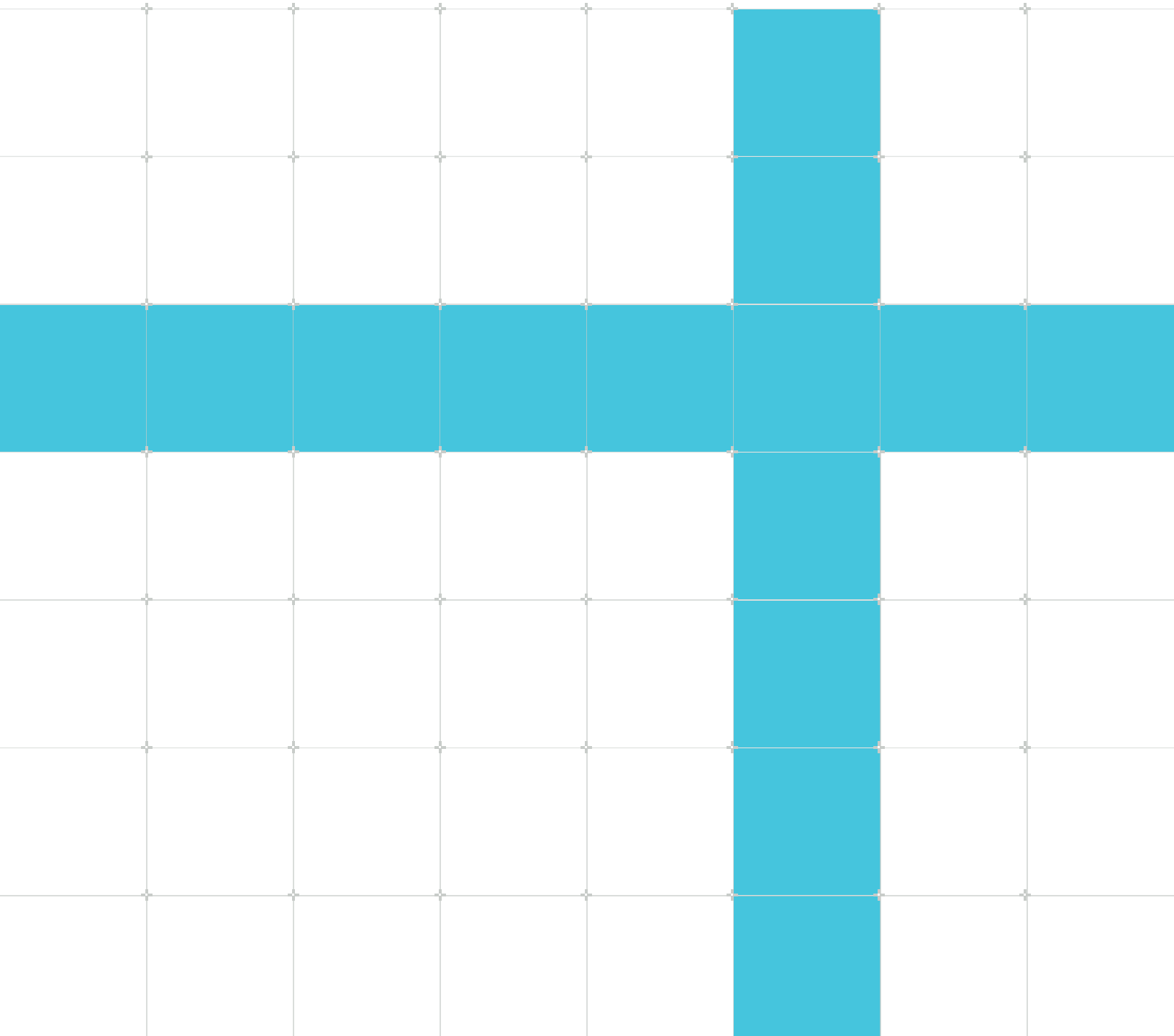
Technical Overview

Non-Confidential

Copyright © 2024 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

109761_0000_01_en



Arm® Corstone™-320 Reference Package Technical Overview

This document is Non-Confidential.

Copyright © 2024 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

This document (109761_0000_01_en) was issued on 2024-10-04. There might be a later issue at <https://developer.arm.com/documentation/109761>

The product revision is r0p0.

See also: [Proprietary notice](#) | [Product and document information](#) | [Useful resources](#)

Start reading

If you prefer, you can skip to [the start of the content](#).

Intended audience

This book is written for hardware or software engineers who want an overview of the components and functionality in Corstone-320.

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

Feedback

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

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

Contents

1. Overview of Corstone-320 Reference Package.....5

2. Overview of SSE-320 Subsystem.....7

3. CMSDK.....10

4. NIC-450.....12

5. SIE-200.....15

6. SIE-300.....16

6.1 Manager Security Controller.....17

6.2 Memory Protection Controller.....18

6.3 Peripheral Protection Controller.....20

6.4 SRAM Memory Controller.....21

6.5 Bridge components.....24

7. SoC-600M.....26

8. SDC-600.....27

9. PCK-600.....29

10. GFC-100.....31

11. GFC-200.....34

12. XHB-500 bridge.....38

13. RTC.....42

14. CG092 Flash Cache.....43

Proprietary notice.....45

Product and document information.....47

Product status.....47

Revision history.....47

Conventions.....48

Useful resources.....51

1. Overview of Corstone-320 Reference Package

Corstone-320 is a licensable package that contains the Arm® Corstone™ SSE-320 Example Subsystem, and a selection of compatible system IP. Corstone-320 also offers associated drivers, security software, and reference applications.

SSE-320 is designed to be the base of an SoC for low-power vision applications or the low-power, ambient, hard real-time compute island in a larger SoC. SSE-320 includes pre-integration support for the optional Arm® Mali™-C55 image signal processor, the Arm® CoreLink™ DMA-350 DMAC and the Arm® Ethos™-U85 NPU.

To build your own SoC, you use the provided system IP, along with the separately licensed Arm® Cortex®-M85 processor, the Mali™-C55 ISP and the Ethos-U85 NPU.

You can adapt the SoC to your specific requirements by using SSE-320 as supplied, modifying it, or not at all.

Corstone-320 grants licenses to the following subsystems and system IP. For details of the product versions, see *Arm® Corstone™-320 Reference Package Release Note*.

Table 1-1: Components included in Corstone-320 license

Component	Description	Used in subsystem	Used in subsystem integration layer	Useful for extension
Arm® Corstone™ SSE-320 Internet of Things (IoT) Example Subsystem (SSE-320)	Main subsystem	Yes	No	No
Arm® Cortex®-M System Design Kit (CMSDK)	Selection of AHB-Lite and APB components	Yes	No	Yes
Arm® CoreLink™ NIC-450 for AXI (NIC-450)	Main interconnect	Yes	Yes	Yes
Arm® CoreLink™ ADB-400 AMBA® Domain Bridge (part of the NIC-450)	AMBA domain bridge	Yes	No	No
Arm® CoreLink™ SIE-200 System IP for Embedded for AHB5 (SIE-200)	Necessary for Arm V8-M systems	Yes	No	No
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded (SIE-300)	AXI5 Security aware components	Yes	Yes	Yes
Arm® CoreSight™ System-on-Chip SoC-600M (SoC-600M)	Debug infrastructure	Yes	Yes	Yes
Arm® CoreSight™ SDC-600 Secure Debug Channel (SDC-600)	Secure Debug Channel	Yes	Yes	Yes
Arm® CoreLink™ PCK-600 Power Control Kit (PCK-600)	Power control elements	Yes	Yes	Yes
Arm® CoreLink™ GFC-100 Generic Flash Controller (GFC-100)	Flash controller	No	No	Yes
Arm® CoreLink™ GFC-200 Generic Flash Controller (GFC-200)	Flash controller	No	No	Yes

Component	Description	Used in subsystem	Used in subsystem integration layer	Useful for extension
Arm® CoreLink™ XHB-500 – AXI to AHB bridge (XHB-500 bridge)	Bridging component	No	Yes	Yes
Arm® PrimeCell™ Real Time Clock (RTC)	Real Time clock	No	No	Yes
Arm® CoreLink™ AHB CG092 Flash Cache (CG092 Flash Cache)	Instruction cache	No	No	Yes

Separately licensed IP

To provide optimum flexibility, the following IP must be licensed separately. These products are not described in detail in this document. For more information, see the relevant product page on Arm Developer:

- [Arm® Cortex-M85](#)
- [Socrates™](#)
- [Arm® Ethos-U85](#)
- [Arm® CoreLink™ DMA-350](#)
- [Arm® Mali™-C55 Image Signal Processor](#)

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

Table 1-2: Components not included in Corstone-320 license

Component	Description	Used in subsystem	Used in subsystem integration layer	Useful for extension
Cortex-M85	Central Processing Unit	Yes	No	No
Socrates	IP integration tool	NA	NA	NA
Ethos-U85	Neural Processing Unit	Yes	No	No
Corelink DMA-350	Direct memory access controller	Yes	No	Yes
Mali-C55	Image Signal Processor	No	Yes	No

2. Overview of SSE-320 Subsystem

A subsystem is self-contained pre-integrated System IP that is designed to perform a specific function, it is intended for use in a SoC.

An Example subsystem showcases the integration of various IP products. The Example subsystem design requires a processor that is licensed and downloaded separately. The Example subsystem can be used as is, or as a starting point for creating a custom subsystem. The reference package includes the components that you would require to create a custom subsystem similar in function the Example subsystem.

Arm® Corstone™ SSE-320 Internet of Things (IoT) Example Subsystem (SSE-320) is designed to be the base of a low-power SoC or the low-power, ambient, hard real-time compute island in a larger SoC.

The Example subsystem design also integrates the following Arm IP, which are optional and can be licensed and downloaded separately:

- Arm® Ethos™-U85
- Arm® CoreLink™ DMA-350

In addition, an example Arm® Mali™-C55 integration on the expansion bus is provided as a reference integration.

SSE-320 supports the [Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M](#) used for deploying secure IoT endpoints. The subsystem was designed with the Arm® Mali™-C55 ISP to target the wide range of IoT low-power applications. Object-recognition, keyword spotting and speech recognition software applications are available for this example subsystem, along with drivers for all the IP used, middleware, cloud integration, security software and the tools to build and debug the firmware.

An example subsystem design follows best practices for EAC, but validation ends at BETA quality (where EAC and BETA refer to Arm standards for the level of testing performed on RTL).

The Example Subsystem is licensed with full modification rights.

The expected partner use model for the Example Subsystem is as follows:

- Render their own specific configuration
- Modify the rendered RTL according to their specific requirements
- Integrate in the rest of the partner SoC
- Fully verify the SoC

SSE-320 integrates the following key Arm components:

- One Arm® Cortex® Cortex-M85 processor core with optional M-Profile Vector Extension (MVE), Floating Point Unit (FPU), Digital Signal Processing (DSP), extensions, caches, Tightly Coupled Memory (TCM)s and Embedded Trace Macrocell (ETM).

- (Optional) One Ethos-U85 neural processing unit (NPU)
- Four Volatile Memory (VM) banks, typically SRAMs.
- Support for two-way striping across two SRAM banks, and support for two-way striping across two DRAMs.
- Memory Protection Controllers (MPC)
- Exclusive Access Monitor (EAM)
- Arm® CoreLink™ NIC-400 System Interconnect
- Implementation Defined Attribution Unit (IDAU)
- Cortex®-M System Design Kit (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)
- (Optional) One Arm® CoreLink™ DMA-350 Direct Memory Access Controller (DMAC)
- Lifecycle Manager (LCM)
- Key Management Unit (KMU)
- Security Alarm Manager (SAM)

The previously listed components are integrated to implement SSE-320 with the following features:

- TrustZone® aware system with the system segregated into Secure and Non-secure worlds
- Configurability to allow several features within the system to be included or removed
- Power Control infrastructure with several pre-defined voltage and power domains
- Each switchable power domain has a local power policy control, and coordinates with other power domains through a centralized dependency control or power interfaces. Switchable power domains provide the system with autonomous dynamic power control infrastructure that, while being software configurable, aiming to minimize software interaction.
- Clock control infrastructure that supports high level clock control including dynamic clock gating and provides clock request handshakes to clock generators
- Comprehensive reset generation and control
- A CoreSight SoC-600M debug infrastructure that supports:
 - A shared Debug Access Port (without example expansion logic)
 - A JTAG/SW debug port (with example expansion logic)
 - Trace Port
 - Cross-triggering
- Secured Debug Channel (SDC-600)

The integrator can tailor the final implementation to the target use-cases by using:

- The supported configuration options.

- The sockets for the Host processor, Host GIC, and External Systems.
- The expansion interfaces of SSE-320.

For more information, see *Arm® Corstone™ SSE-320 Example Subsystem Reference Manual*

3. CMSDK

Corstone-320 includes the Cortex®-M System Design Kit (CMSDK). CMSDK helps you design products using Arm® Cortex-M processors.

CMSDK contains the following items:

- 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

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

Cortex-M Software Design Kit Software

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

CMSIS-compliant drivers

- 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 mode

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](#)

4. NIC-450

Corstone-320 includes the Arm® CoreLink™ NIC-450 Network Interconnect. NIC-450 is a library of highly configurable and multi-power domain tools.

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

NIC-400 Network Interconnect

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

CoreLink™ QoS-400

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

CoreLink™ QVN-400

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

CoreLink™ DPE-400 Data Parity Extension CoreLink™ TLX-400 Network Interconnect Thin Links

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

CoreLink™ AMBA® Domain Bridge

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

CoreLink™ AXI4 to AHB-Lite XHB-400 Bridge

CoreLink™ XHB-400 converts AXI4 protocol to AHB-Lite protocol, and has an AXI4 subordinate interface and an AHB-Lite manager interface.

CoreLink™ LPD-500 Low Power Distributor

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

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

The benefits of using the NIC-450 are:

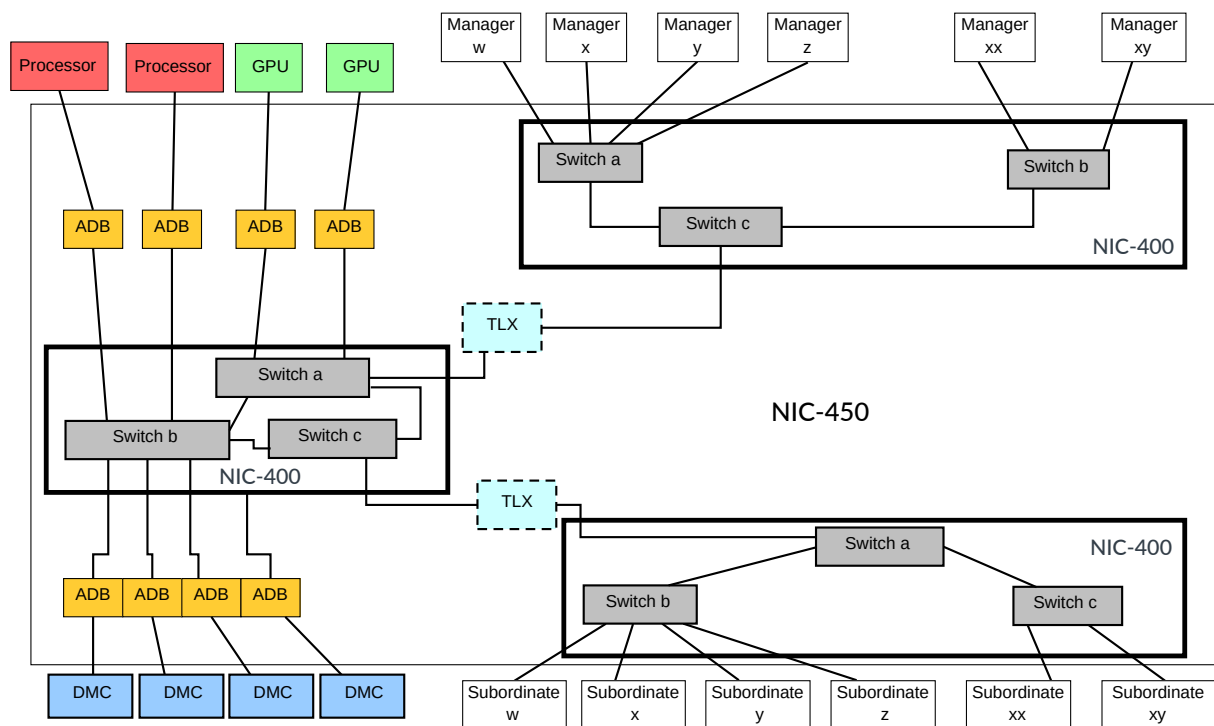
- Unified low-power interfaces when applicable
- Single design environment to configure IP blocks and connect them together

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

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

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

Figure 4-1: NIC-450 block diagram



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

- [Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview](#)
- [Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual](#)
- [Arm® CoreLink™ NIC-400 Network Interconnect Integration Manual](#)
- [Arm® CoreLink™ NIC-400 Network Interconnect Implementation Guide](#)
- [Arm® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual](#)
- [Arm® CoreLink™ QVN-400 Network Interconnect Advanced Quality of Service for Virtual Networks Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual](#)
- [Arm® CoreLink™ TLX-400 Network Interconnect Thin Links Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual](#)

- *Arm® CoreLink™ ADB-400 AMBA Domain Bridge User Guide*
- *Arm® CoreLink™ AXI4 to AHB-Lite XHB-400 Bridge Technical Reference Manual*

5. SIE-200

Corstone-320 includes the Arm® CoreLink™ SIE-200 System IP for Embedded. SIE-200 is a collection of interconnect, peripheral, and TrustZone™ controller components for use with a processor that complies with the ARMv8-M processor architecture.

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

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

For more information, see the SIE-200 and the associated products documentation sets:

- [Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual](#)
- [Arm® CoreLink™ SIE-200 System IP for Embedded Configuration and Integration Manual](#)

6. SIE-300

Corstone-320 includes the CoreLink™ SIE-300 AXI5 System IP for Embedded. SIE-300 provides a set of configurable AXI5 security-aware components. The components 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 clock synchronizing bridges and an access control gate.

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 that target a memory interface. The security checks operate on block or page level, and are programmable by using the APB controller 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.

For more information, see the SIE-300 and the associated products documentation sets:

- [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](#)

6.1 Manager Security Controller

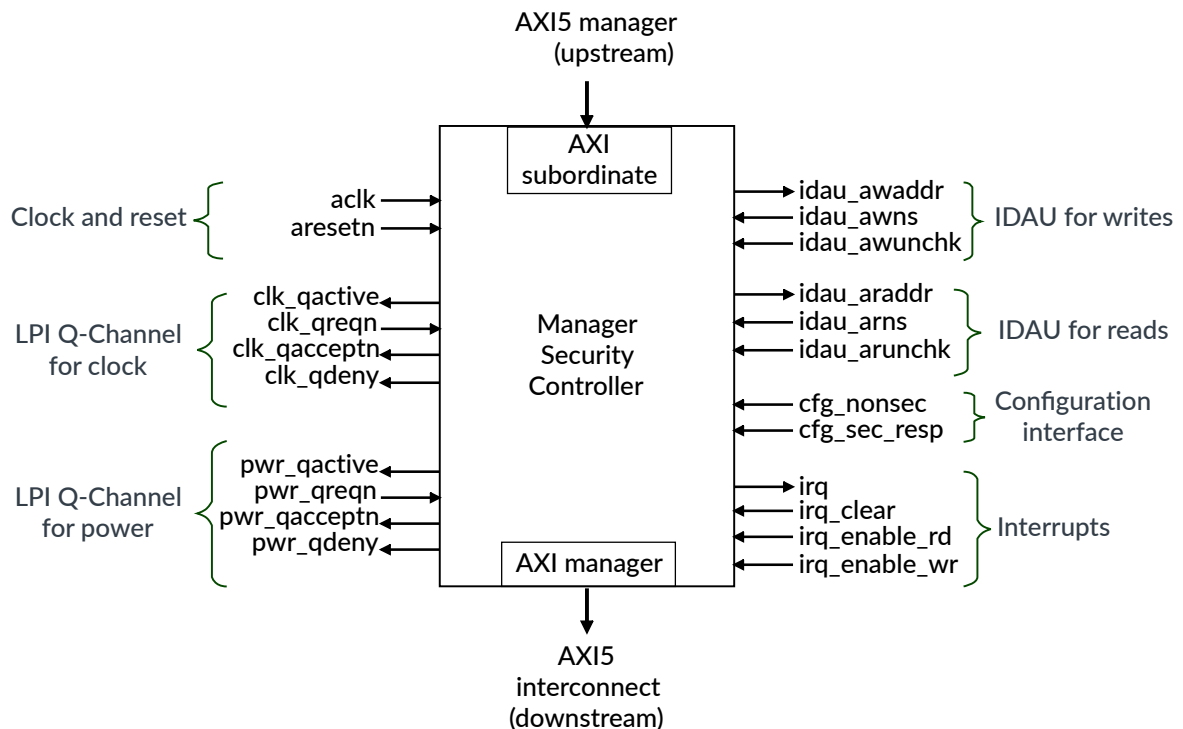
The Manager Security Controller (MSC) acts as security gate for AXI transactions, and it can transform the security attribute.

The MSC enables AXI managers that are designed for A-class systems to be inserted into M-class systems. Since A-class and M-class systems handle security differently, the MSC can transform the security attributes of a transaction to satisfy the M-class requirements.

The reference body describes and discusses the subject of the topic.

The following figure shows the MSC interfaces.

Figure 6-1: MSC interfaces



The AXI subordinate and AXI manager interfaces provide the AXI data path from the AXI manager to the interconnect.

To support low-power quiescence, the MSC has two Q-Channel interfaces. One Q-Channel is for clock quiescence and the other Q-Channel is for power quiescence. The AXI subordinate and AXI manager interfaces provide the AXI data path from the AXI manager to the interconnect.

Configuration interface

The `cfg_nonsec` input tells the MSC whether the AXI5 manager, which connects to its subordinate interface, is in the Secure state or the Non-secure state. The MSC uses this information to control whether it blocks a transaction from going downstream.

When the MSC blocks a transaction, the `cfg_sec_resp` controls whether the MSC:

- Responds with an AXI subordinate error (SLVERR).
- Ignores a write transaction or returns zero for a read transaction.

IDAU interfaces

The MSC has two Implementation Defined Attribution Unit (IDAU) interfaces that it uses to discover the Security state of an addressed region. One IDAU is for read transactions and the other IDAU is for write transactions.

When the MSC receives an AXI transaction, it accesses the corresponding IDAU and retrieves the Security state for that transaction address. By using the Security state information, the incoming AXI access permissions (AxPROT), and the state of `cfg_nonsec`, the MSC can do one of the following:

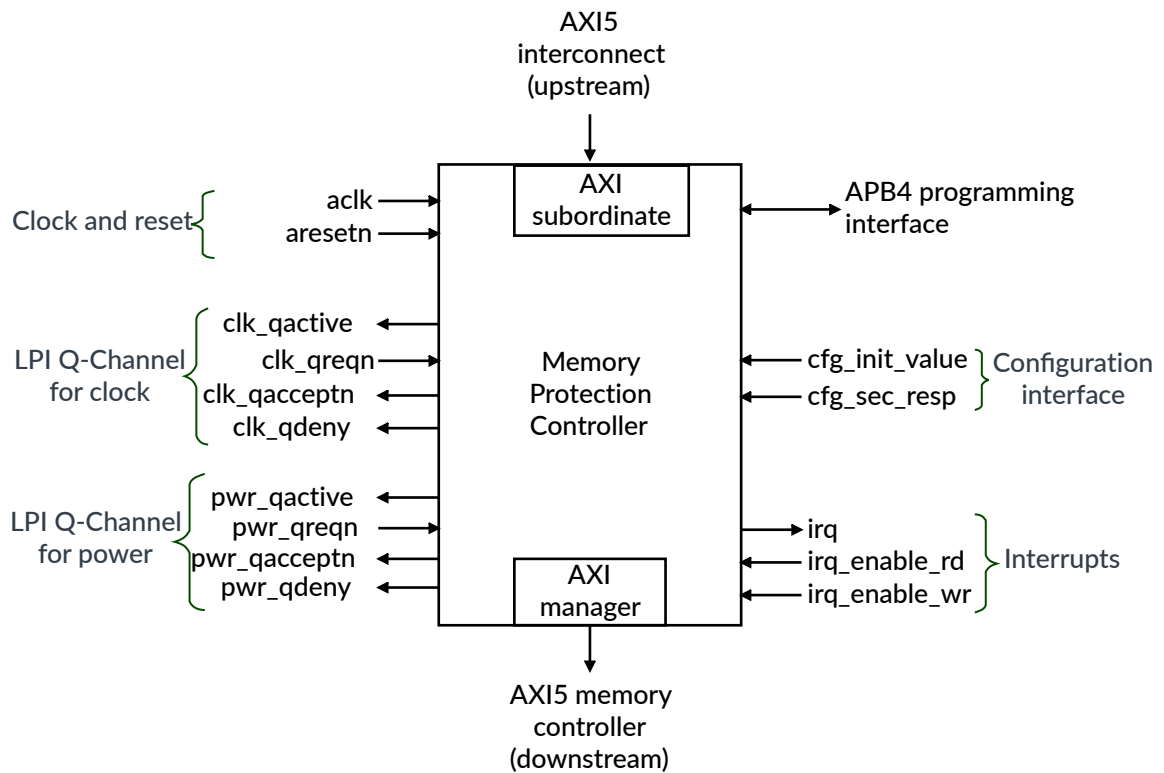
- Block the transaction from going downstream.
- Forward the transaction.
- Transform the security attributes and then forward the transaction.

6.2 Memory Protection Controller

The Memory Protection Controller (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 completer interface.

The following figure shows the MPC interfaces.

Figure 6-2: MPC interfaces



The AXI subordinate and AXI manager interfaces provide the AXI data path from the interconnect to the memory controller.

To support low-power quiescence, the MPC has two Q-Channel interfaces. One Q-Channel is for clock quiescence and the other Q-Channel is for power quiescence.

Configuration interface

At powerup, the MPC uses the value of the **cfg_init_value** input as the initialization value for the Look Up Table (LUT) to be Secure or Non-secure for the entire memory range that the MPC protects.

If a security violation occurs, the MPC generates an interrupt and the **cfg_sec_resp** controls whether the MPC:

- Responds with an AXI subordinate error (SLVERR).
- Ignores a write transaction or returns zero for a read transaction.



- When accessing all internal registers (except for the PID/CID registers) with a Non-secure APB transaction, the response is either an error or **RAZ/WI** depending on the value of the **cfg_sec_resp** input signal.

- When accessing all internal registers (except for the PID/CID registers) with a Secure but unprivileged APB transaction, the response is always **RAZ/WI**, regardless of the value of the `cfg_sec_resp` input signal.

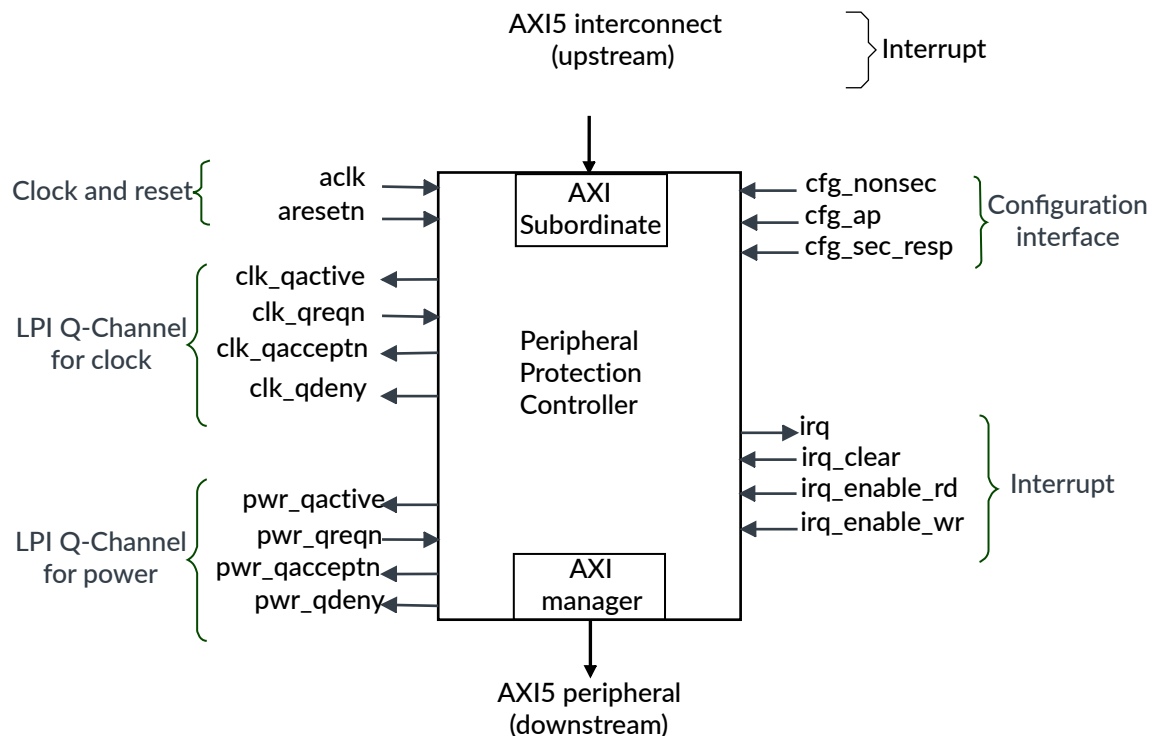
6.3 Peripheral Protection Controller

The Peripheral Protection Controller (PPC) provides security checks for AXI peripherals.

The PPC gates AXI transactions towards a peripheral when a security violation occurs. It can be instantiated in the system in connection to any non-security aware AXI5 peripheral. Security checking is performed against the state of the `cfg_ap` and `cfg_nonsec` signals, which indicate the privilege and Security state of the peripheral.

The following figure shows the PPC interfaces.

Figure 6-3: PPC interfaces



The AXI subordinate and AXI manager interfaces provide the AXI data path from the AXI manager to the attached peripheral.

To support low-power quiescence, the PPC has two Q-Channel interfaces. One Q-Channel is for clock quiescence and the other Q-Channel is for power quiescence.

Configuration interface

The `cfg_nonsec` signal controls the security settings of the attached peripheral:

If HIGH, only Non-secure accesses to the peripheral are allowed. If LOW, only Secure accesses to the peripheral are allowed. The `cfg_ap` signal controls the privilege settings of the attached peripheral:

- If HIGH, only privileged accesses to the peripheral are allowed.
- If LOW, the privilege attribute is ignored for security checks.

When the PPC blocks a transaction, the `cfg_sec_resp` signal controls whether the PPC

When the MSC blocks a transaction, the `cfg_sec_resp` controls whether the MSC:

- Responds with an AXI subordinate error (SLVERR).
- Ignores a write transaction or returns zero for a read transaction.

6.4 SRAM Memory Controller

The SRAM Memory Controller (SMC) is an AXI5 memory controller for static memory devices.

The SMC has the following features:

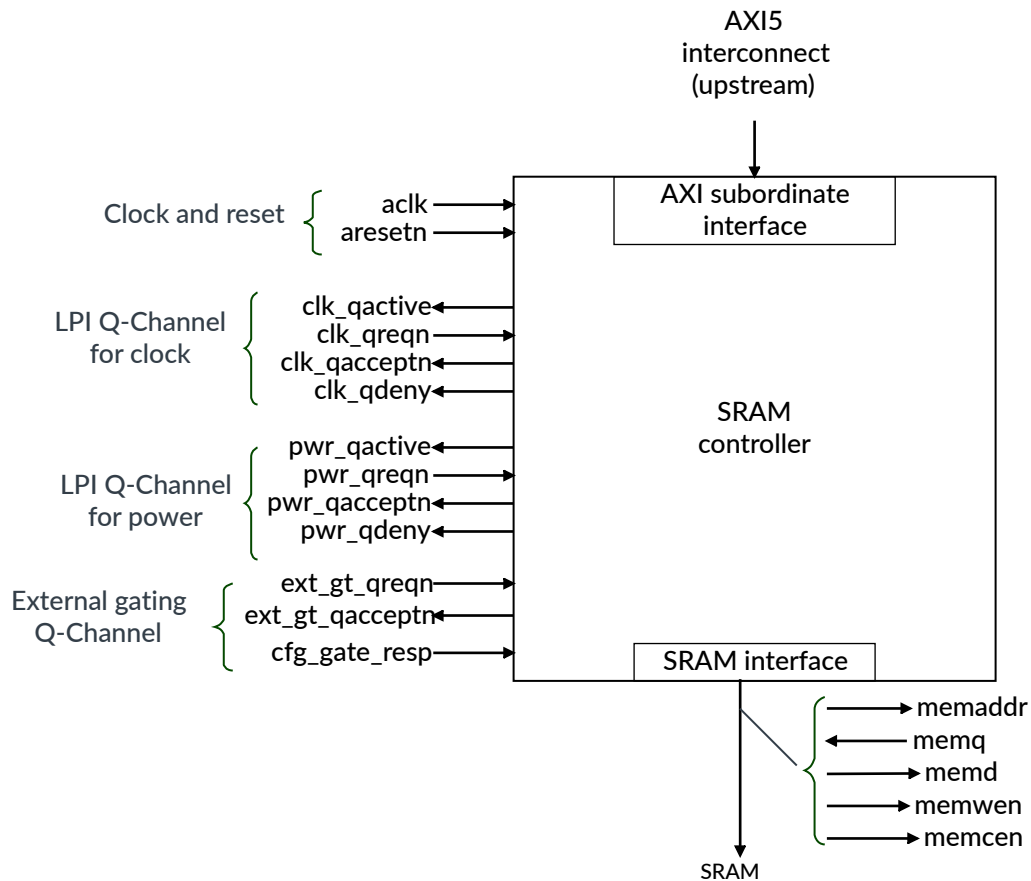
- A single clock and reset domain
- An AXI5 subordinate interface
- An SRAM interface
- Two Q-Channels for clock and power control
- No data width conversion
- An external-gating interface that prevents the SMC from issuing new transactions on the SRAM interface



The SRAM controller primarily supports memory macros that the Arm® SRAM Compiler generates.

The following figure shows the SMC interfaces

Figure 6-4: SMC interfaces



To support low-power quiescence, the SMC has two Q-Channel device interfaces. One Q-Channel is for clock quiescence and the other Q-Channel is for power quiescence.

The SMC also provides a partial Q-Channel device interface to support external gating, which stops the SMC from starting any new transactions on the SRAM interface.

Early write response

The SMC buffers the write transactions and for non-exclusive writes it returns an early write response. If the write buffers become full, then the SMC does not return an early write response.

Read and write transaction scheduling

The AXI has separate buses for reads and writes, but the SRAM interface is a single bus for reads and writes. Therefore, the SMC must arbitrate between the AXI channels. The SMC performs arbitration between read and write bursts.

If the write buffers become full, then the arbitration scheme uses the QoS value of the incoming read *arqos* and write *awqos* signals. For example, if the SMC receives a write with a QoS value that

is higher than the read QoS, it forwards the oldest transaction from the write queue to the SRAM to allow the new write into the write buffer.



Provided the write buffers (address or data) are not full, the SMC gives priority to read bursts.

Poison

The AXI5_POISON_EN configuration parameter controls whether the SMC supports data poisoning. When data poisoning is enabled, the SMC provides 1 bit of poison information for each 64 bits of data. The SMC uses the MSB of the memwen write enable bus on the SRAM memory side to control the writes to the storage element that holds the poison information.



When narrow writes are written to the SRAM, the poison information always gets updated with the new value, regardless of the previously stored content.

Table 6-1: Poison bit locations

Data width	AXI poison bits	Poison bits on the SRAM interface
32	wpoison[0] and rpoison[0]	memd[32] and memq[32]
64	wpoison[0] and rpoison[0]	memd[64] and memq[64]
128	wpoison[1:0] and rpoison[1:0]	memd[129:128] and memq[129:128]
256	wpoison[3:0] and rpoison[3:0]	memd[259:256] and memq[256:256]

Exclusive accesses

The SMC can contain up to 16 Exclusive Access Monitors (EAMs), depending on the setting of the EXCLUSIVE_MONITORS configuration parameter.

If EXCLUSIVE_MONITORS > 0, then Exclusive Load transactions always return an EXOKAY response and the SMC stores the transaction details (address, ID) in an internal TAG buffer.

For Exclusive Store transactions, the SMC checks if the address and ID are present in the TAG buffer, and if so the SMC forwards the write to the SRAM and returns an EXOKAY write response. If the check fails, the SMC ignores the write data and it returns an OKAY response, which indicates an exclusive access failure. If a non-exclusive write transaction accesses a location that is stored in a TAG buffer, then the SMC clears the TAG.

If all EAMs are occupied, and the SMC receives a new Exclusive Load transaction with an:

- ID that exists in the TAG table, then the new transaction replaces an old entry.
- ID that does not exist in the TAG table, then the SMC overwrites an entry in the TAG table that the round-robin algorithm selects, and returns an EXOKAY response.

If an SMC is configured to contain no EAMs (`EXCLUSIVE_MONITORS == 0`), then exclusive writes always fail. The SMC ignores the write and returns an OKAY response.

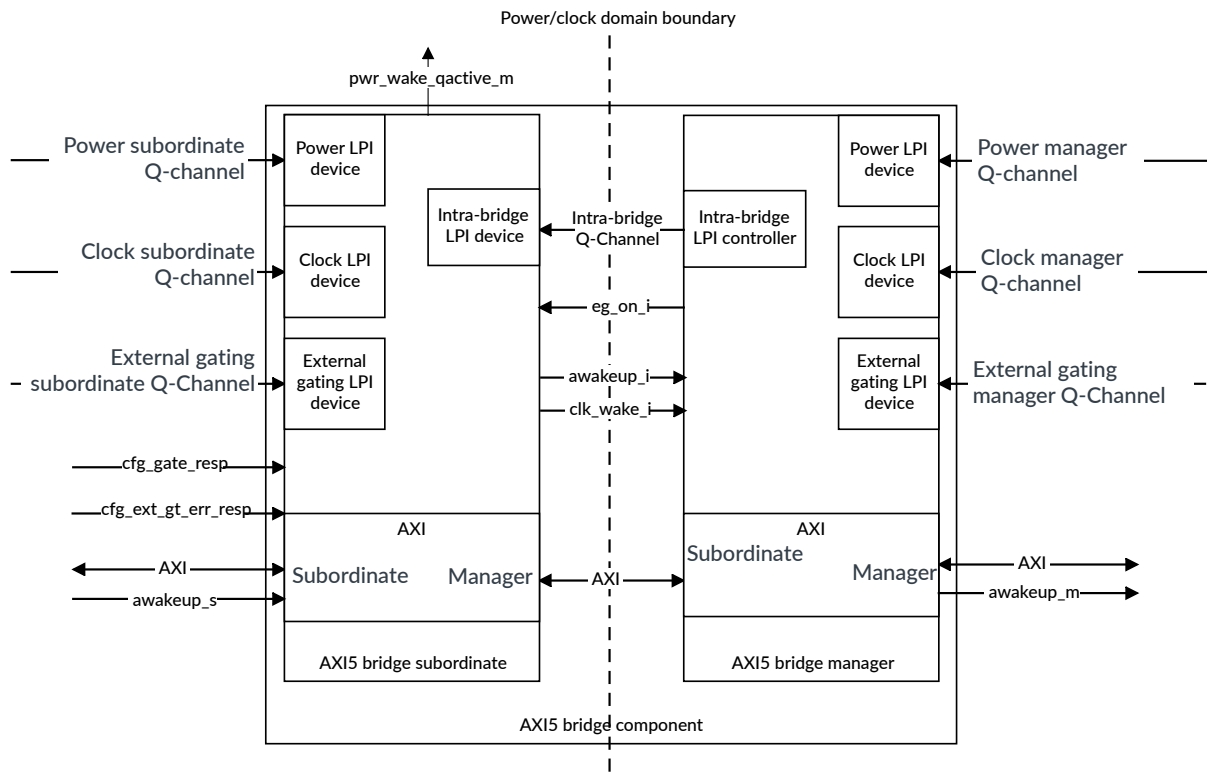
6.5 Bridge components

}

Bridge components provide low-power management and external gating on boundaries between clock and power domains along the AXI5 data bus. They also have configurable registering options to ease timing on long AXI5 paths.

The following figure shows the interfaces of a bridge component.

Figure 6-5: Bridge component interfaces



The following table lists the bridge components

Table 6-2: Supported bridge components

Bridge component	Upstream to downstream clock ratio
Access Control Gate (ACG)	One-to-one
Sync-Down Bridge (SDB)	N-to-one

Bridge component	Upstream to downstream clock ratio
Sync-Up Bridge (SUB)	One-to-N

Each bridge component consists of an upstream side and a downstream side. To allow communication across clock and power domains, each side of the bridge has one intra-bridge Q-Channel interface and one intra-bridge AXI interface. The `eg_on_i` signal provides the upstream side with information about the state of external gating on the downstream side. The intra-bridge uses a standard Q-Channel LPI interface.

Each half of a bridge component supports the following features:

- A single clock and reset domain
- An AXI5 subordinate interface
- An AXI5 manager interface
- Two Q-Channels for clock and power control to support low-power quiescence
- No data width conversion
- An external-gating interface that prevents the bridge component from issuing new transactions on the AXI interface. The external-gating interface is a Q-Channel implementation without the QDENY (if `otherprops` is 'g.signal.name') and QACTIVE signals.

Configuration interface

The `cfg_gate_resp` controls how the upstream side of the bridge component responds, when the bridge is closed by external gating or downstream power quiescence:

- Responds with an AXI subordinate error (SLVERR).
- The bridge component sets the relevant AXI ready signals LOW, which stalls any AXI transactions, until the bridge is able to forward the transfers to the downstream side.

The `cfg_ext_gt_err_resp` signal controls how the bridge component responds to AXI transactions, when the upstream external gating is in quiescence. However, if the `cfg_gate_resp` is set to error, then the bridge returns an error response. Therefore, when the upstream external gating is in quiescence, the bridge:

- Responds with an AXI subordinate error (SLVERR), when `cfg_gate_resp` or `cfg_ext_gt_err_resp` if `otherprops` is are HIGH.
- Stalls the transaction until the external gating request is released, that is, `ext_gt_qreqn_s` goes HIGH. This response behavior requires that `cfg_gate_resp` and `cfg_ext_gt_err_resp` are LOW.

7. SoC-600M

Corstone-320 includes the CoreSight™ SoC-600M. SoC-600M is a member of the Arm® embedded debug and trace component family that is based on the Cortex-M processor.

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

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

The SoC-600M package 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.

For more information, see the SOC-600M and the associated products documentation sets:

- [Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0](#)
- [Arm® Coresight™ System-on-Chip SoC-600M Configuration and Integration Manual, Version r1p0](#)

8. SDC-600

Corstone-320 includes the Arm® CoreSight™ SDC-600. SDC-600 provides a dedicated channel for authentication between an external debugger and a debug target platform by using an unlocking mechanism.

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

SDC-600 performs the following tasks:

- Requests power and optionally reboots the servicing agent.
- Establishes and maintains a link between a port on the external side, which is serviced by the debugger, and a port on the internal side, which is serviced by an agent on the target system.
- Transports messages from an external debugger to a hardware or software agent on a target system through a point-to-point link.
- The debugged target and the servicing agent are typically the same processor or processor subsystem, but they can be separate entities as well.

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

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

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

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

For more information, see SDC-600 documentation sets:

- [Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual](#)
- [Arm® CoreSight™ SDC-600 Secure Debug Channel Configuration and Integration Manual](#)

9. PCK-600

Corstone-320 includes the Arm® CoreLink PCK-600 Power Control Kit. PCK-600 is a library of highly configurable and multi-power domain tools.

The PCK-600 consists of the following components:

Low Power Distributor Q-Channel(LPD-Q)

The LPD-Q component distributes a Q-Channel from one Q-Channel controller to up to 32 Q-Channel devices.

Low Power Distributor P-Channel (LPD-P)

The LPD-P component distributes a P-Channel from one P-Channel controller to up to 8 P-Channel devices.

Low Power Combiner Q-Channel (LPC-Q)

The LPC-Q component combines the Q-Channels from multiple Q-Channel controllers to multiple Q-Channel devices with common control requirements.

P-Channel to Q-Channel Converter (P2Q)

The P2Q component converts a P-Channel to a Q-Channel.

Clock Controller (CLK-CTRL)

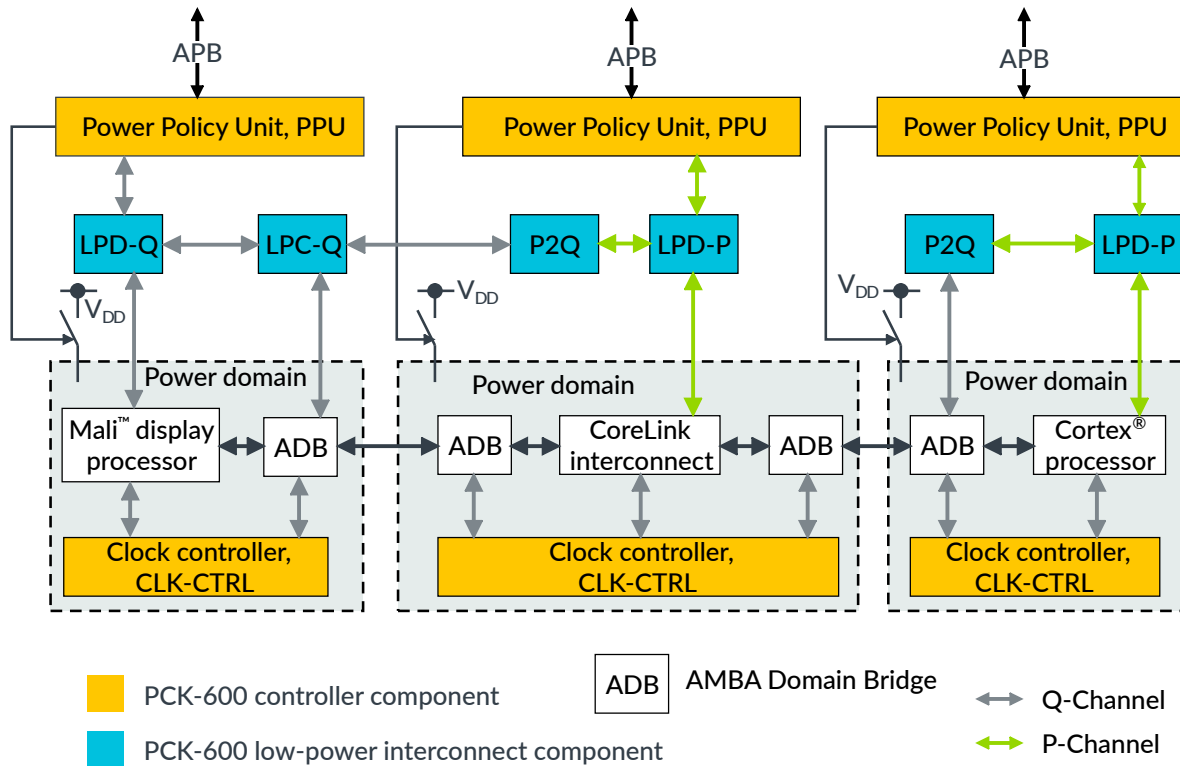
The CLK-CTRL component provides High-level Clock Gating (HCG) for a single clock domain.

Power Policy Unit (PPU)

The PPU component is a configurable and programmable P-Channel and Q-Channel power domain controller.

The following figure shows an example system that uses the PCK-600 components to manage three power domains. The PCK-600 components are shown in orange and blue.

Figure 9-1: Example system that contains PCK-600



For more information, see PCK-600 documentation sets:

- [Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual](#)
- [Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual](#)

10. GFC-100

Corstone-320 includes the Arm® CoreLink™ GFC-100 Generic Flash Controller. GFC-100 comprises the generic part of a Flash controller in a System-on-Chip (SoC). GFC-100 enables an embedded Flash macro to be integrated easily into any system. The GFC-100 comprises the generic part of a Flash controller in a System-on-Chip (SoC). GFC-100 enables an embedded Flash macro to be integrated easily into any system.

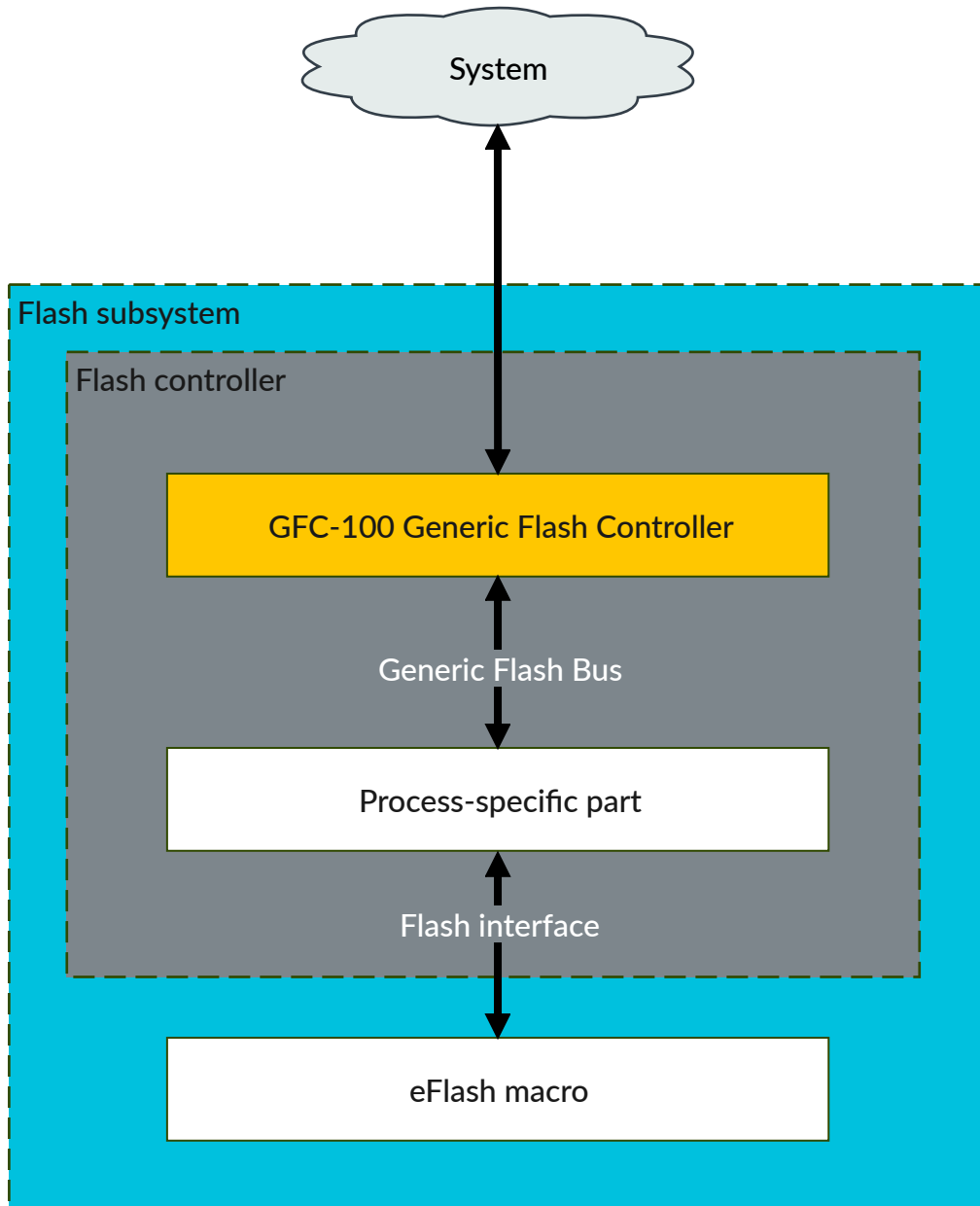
An eFlash macro enables a Flash controller to access eFlash memory. The eFlash macros produced by different foundries and processes can have different interfaces, timings, signal names, protocols and features that are determined by the foundry processes that produced the eFlash memory.

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

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

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

Figure 10-1: GFC-100 in a Flash controller implementation



GFC-100 features

The GFC-100 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 main and extended areas of embedded Flash
- Burst support
- Low latency

Primary APB completer interface :

- Write and erase access to to the main and extended areas of embedded Flash
- Debug read access to 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 registers and external registers

APB register requester interface :

- Control port for process-specific registers

Q-Channel interface :

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

P-Channel controller interface :

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

Generic Flash Bus (GFB) :

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

For more information, see GFC-100 documentation sets:

- [Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual](#)
- [Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual](#)

11. GFC-200

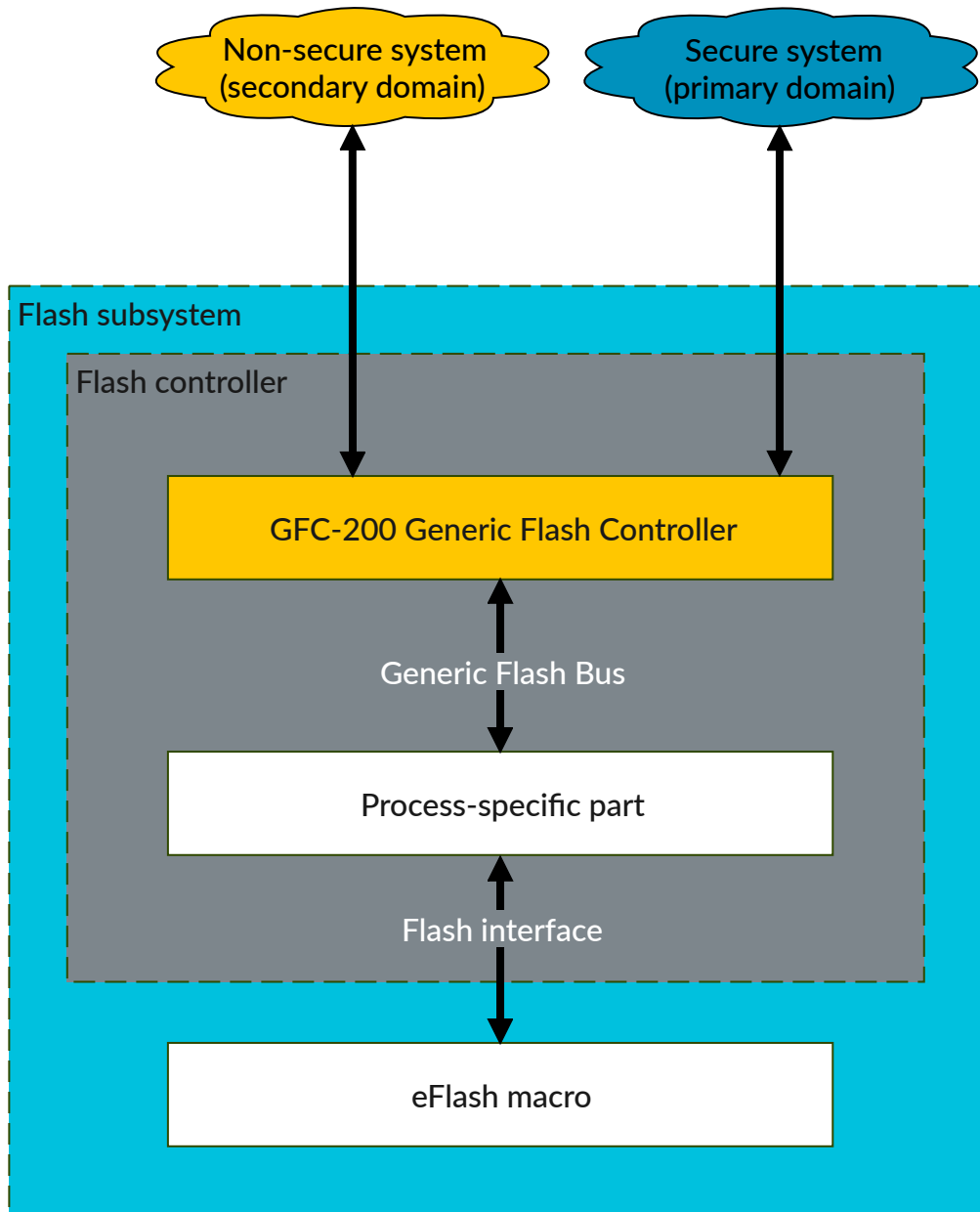
Corstone-320 includes the Arm® CoreLink™ GFC-200 Generic Flash Controller. The GFC-200 comprises the generic part of a Flash controller in a System-on-Chip (SoC). The GFC-200 enables an embedded Flash macro to be integrated easily into any system.

An eFlash macro enables a Flash controller to access eFlash memory. The eFlash macros produced by different foundries and processes can have different interfaces, timings, signal names, protocols, and features 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 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 11-1: GFC-200 in a Flash controller implementation



GFC-200 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 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 process-specific part

Generic Flash Bus (GFB):

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

For more information, see GFC-200 documentation sets:

- [Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual](#)
- [Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual](#)

12. XHB-500 bridge

Corstone-320 includes the AMBA® AXI5 to AHB5 bridge and the 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.

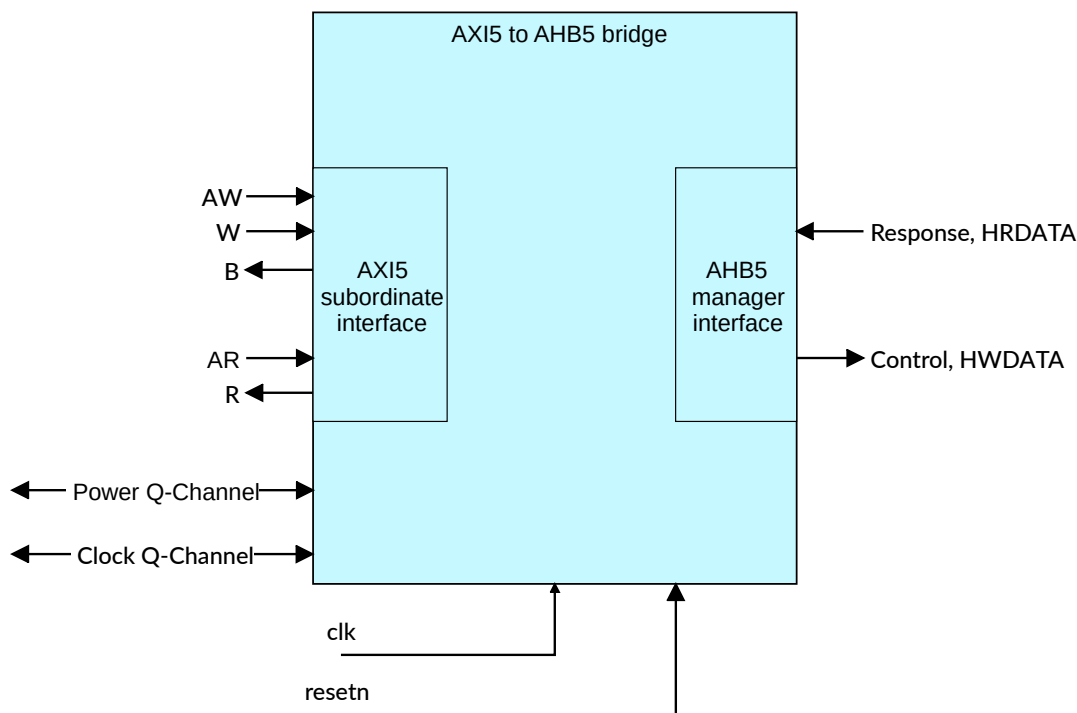
The AHB5 to AXI5 translates AHB5 transfers into the corresponding AXI transactions. The bridge has an AHB5 subordinate interface and an AXI5 manager interface.

AXI5 to AHB5 overview

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

The following figure shows the interfaces of the AHB5.

Figure 12-1: 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

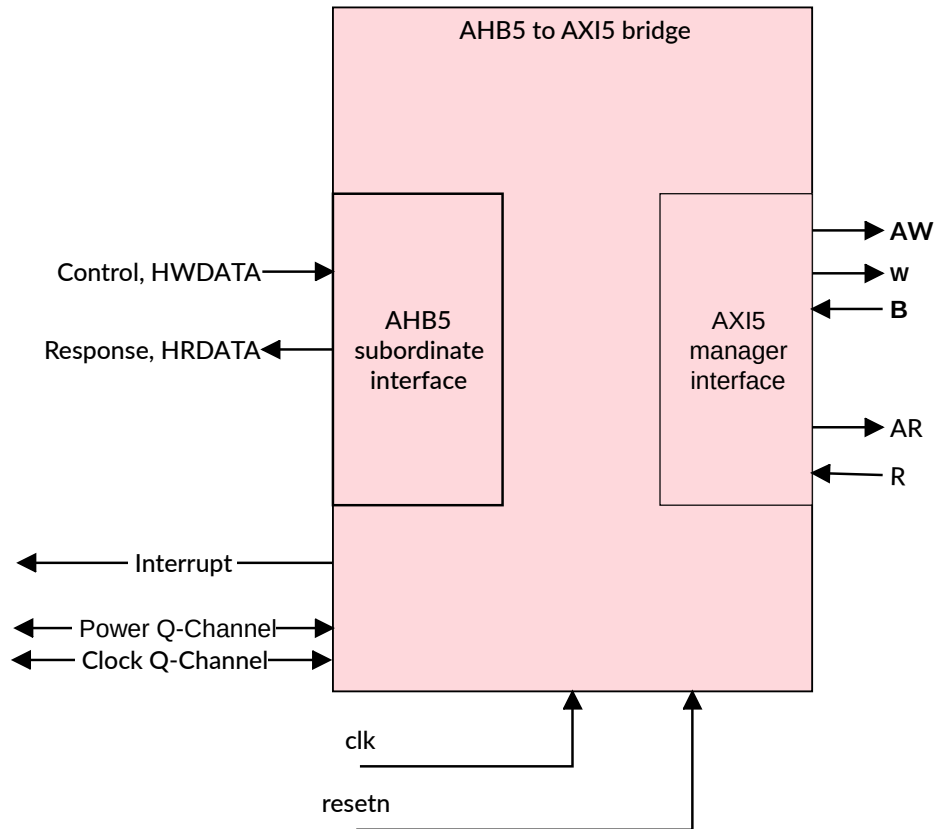
The bridge does not support endian conversion.

AXI5 overview

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

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

Figure 12-2: 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.

For more information, see the XXHB-500 and the associated products documentation sets:

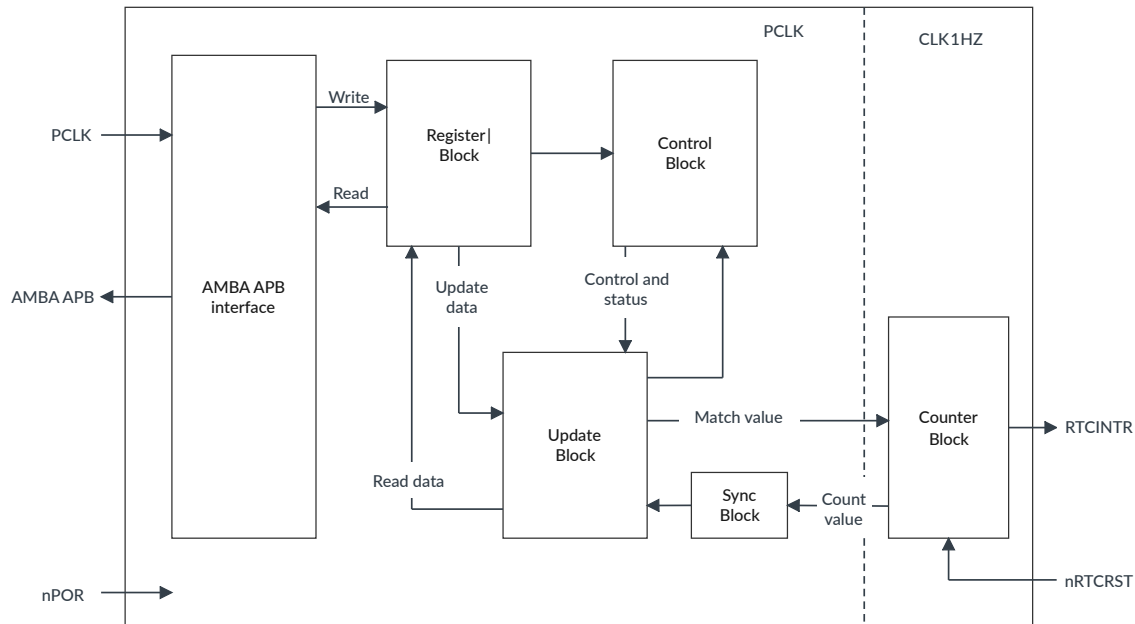
- [Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual](#)
- [Arm® CoreLink™ XHB-500 Bridge Configuration and Integration Manual](#)

13. RTC

Corstone-320 includes the Arm® PrimeCell Real Time Clock (RTC). The RTC is an AMBA completer module that connects with the APB interface.

The following figure shows the RTC block diagram.

Figure 13-1: 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.

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

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

For more information, see the Real Time Clock (RTC) documentation set:

- [Arm® PrimeCell Real Time Clock \(PL031\) Technical Reference Manual](#)

14. CG092 Flash Cache

Corstone-320 includes the Arm® CoreLink™ CG092 AHB Flash Cache. The CG092 Flash Cache is an instruction cache that is instantiated between the bus interconnect and the eFlash controller.

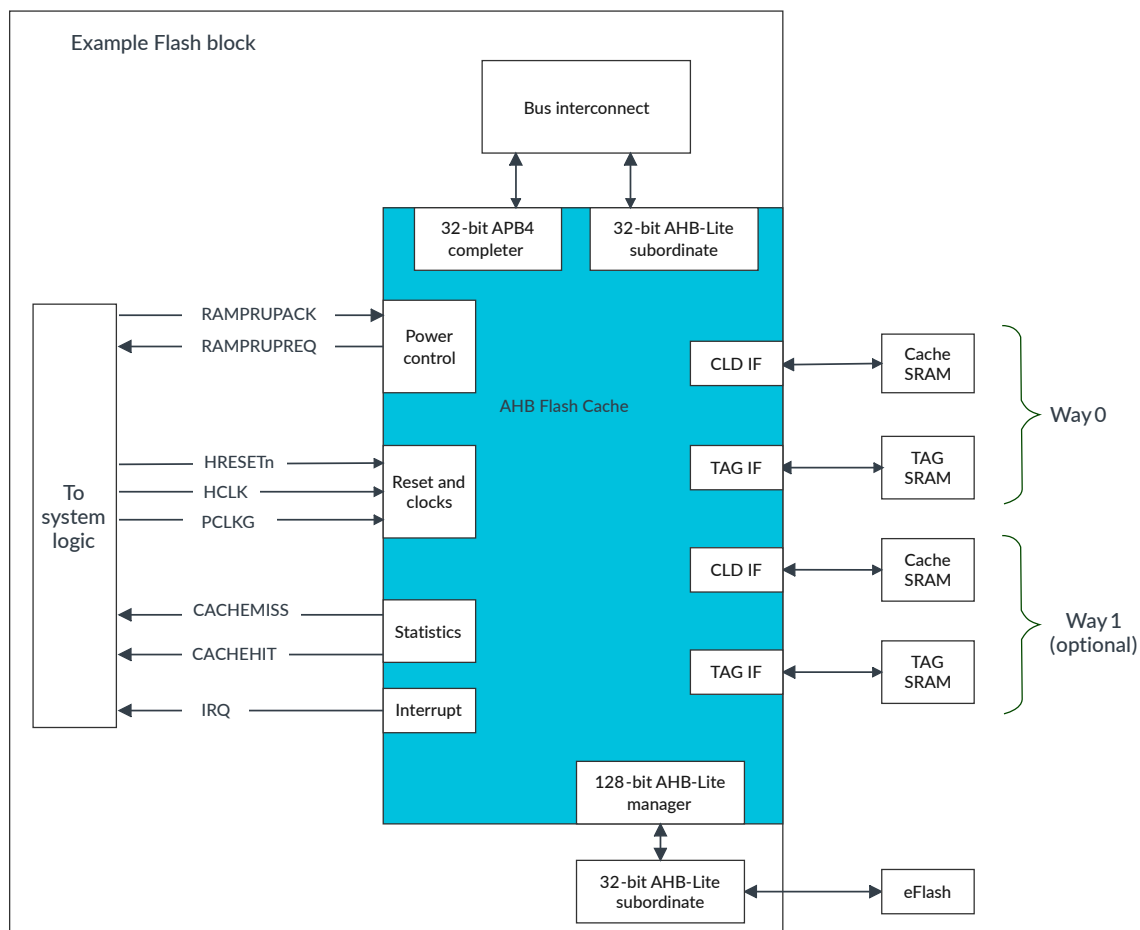
The CG092 Flash Cache is a simple cache for on-chip embedded Flash (eFlash). The AHB Flash Cache design is optimized for fetching the processor instructions directly from an eFlash. The main benefit of the AHB Flash Cache is improved power efficiency, but there are also improvements in code fetching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



The AHB CG092 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 14-1: Example eFlash implementation



The AHB CG092 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 powerup and power down (with simple handshaking). If valid data is in the powered-down cache because the cache is in a low-power state, the cache contents must not be invalidated on wake up. The software can therefore save energy by avoiding invalidating the cache RAMs on wake up.
- Supports automatic or manual cache invalidate in the enabling sequence. This behavior can be overridden.
- 32-bit AHB subordinate interface to the AHB manager in the system processor.
- 32-bit APB subordinate interface to the memory-mapped registers of the CG092.
- 128-bit AHB manager 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.



An eFlash controller is not part of the CG092 component.

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

- [Arm® Corelink™ CG092 AHB Flash Cache Technical Reference Manual](#)
- [Arm® Corelink™ CG092 AHB Flash Cache Configuration and Integration Manual](#)

Proprietary Notice

This document is protected by copyright and other related rights and the use 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 Limited ("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 the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. 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, any patents, copyrights, trade secrets, trademarks, or other rights.

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.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted 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.

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 this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

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. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

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

Product revision status

The rOp0 identifier indicates the revision status of the product described in this manual, where:

rx	Identifies the major revision of the product.
py	Identifies the minor revision or modification status of the product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0000-01	4 October 2024	Non-Confidential	Initial release

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in [Document release information](#) on page 47.

Table 2: Issue 0000-01

Change	Location
Initial issue of the document	-

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
bold	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	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: <div>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



Caution

We recommend the following. If you do not follow these recommendations your system might not work.



Warning

Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.



This information is important and needs your attention.



This information might help you perform a task in an easier, better, or faster way.



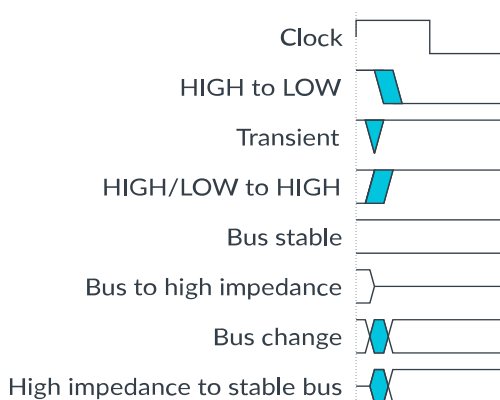
This information reminds you of something important relating to the current content.

Timing diagrams

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

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

Figure 1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ ADB-400 AMBA Domain Bridge User Guide	DUI 0615	Confidential
Arm® CoreLink™ AXI4 to AHB-Lite XHB-400 Bridge Technical Reference Manual	DDI 0523	Non-Confidential
Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual	101060	Confidential
Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual	101059	Non-Confidential
Arm® CoreLink™ GFC-200 Generic Flash Controller Configuration and Integration Manual	101485	Confidential
Arm® CoreLink™ GFC-200 Generic Flash Controller Technical Reference Manual	101484	Non-Confidential
Arm® CoreLink™ NIC-400 Network Interconnect Implementation Guide	DII 0273	Confidential
Arm® CoreLink™ NIC-400 Network Interconnect Integration Manual	DII 0269	Confidential
Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual	DDI 0475	Non-Confidential
Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview	100459	Non-Confidential
Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual	101151	Confidential
Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual	101150	Non-Confidential
Arm® CoreLink™ QVN-400 Network Interconnect Advanced Quality of Service for Virtual Networks Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual	DSU 0027	Non-Confidential
Arm® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual	DSU 0026	Non-Confidential
Arm® CoreLink™ SIE-200 System IP for Embedded Configuration and Integration Manual	DIT 0067	Confidential
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	Non-Confidential
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Configuration and Integration Manual	101527	Confidential
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual	101526	Non-Confidential
Arm® CoreLink™ TLX-400 Network Interconnect Thin Links Supplement to Arm CoreLink NIC-400 Network Interconnect Technical Reference Manual	DSU 0028	Non-Confidential
Arm® CoreLink™ XHB-500 Bridge Configuration and Integration Manual	101376	Confidential

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual	101375	Non-Confidential
Arm® CoreSight™ SDC-600 Secure Debug Channel Configuration and Integration Manual	101131	Confidential
Arm® CoreSight™ SDC-600 Secure Debug Channel Technical Reference Manual	101130	Non-Confidential
Arm® Corelink™ CG092 AHB Flash Cache Configuration and Integration Manual	DIT 0065	Confidential
Arm® Corelink™ CG092 AHB Flash Cache Technical Reference Manual	DDI 0569	Non-Confidential
Arm® Coresight™ System-on-Chip SoC-600M Configuration and Integration Manual, Version r1p0	101884	Confidential
Arm® Coresight™ System-on-Chip SoC-600M Technical Reference Manual, Version r1p0	101883	Non-Confidential
Arm® Corstone™ SSE-320 Example Subsystem Reference Manual	109758	Confidential
Arm® Corstone™-320 Reference Package Release Note	109765	Confidential
Arm® Cortex®-M System Design Kit Example System Guide	DUI 0594	Confidential
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	Non-Confidential
Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual	DDI 0224	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M	DEN 0083	Non-Confidential