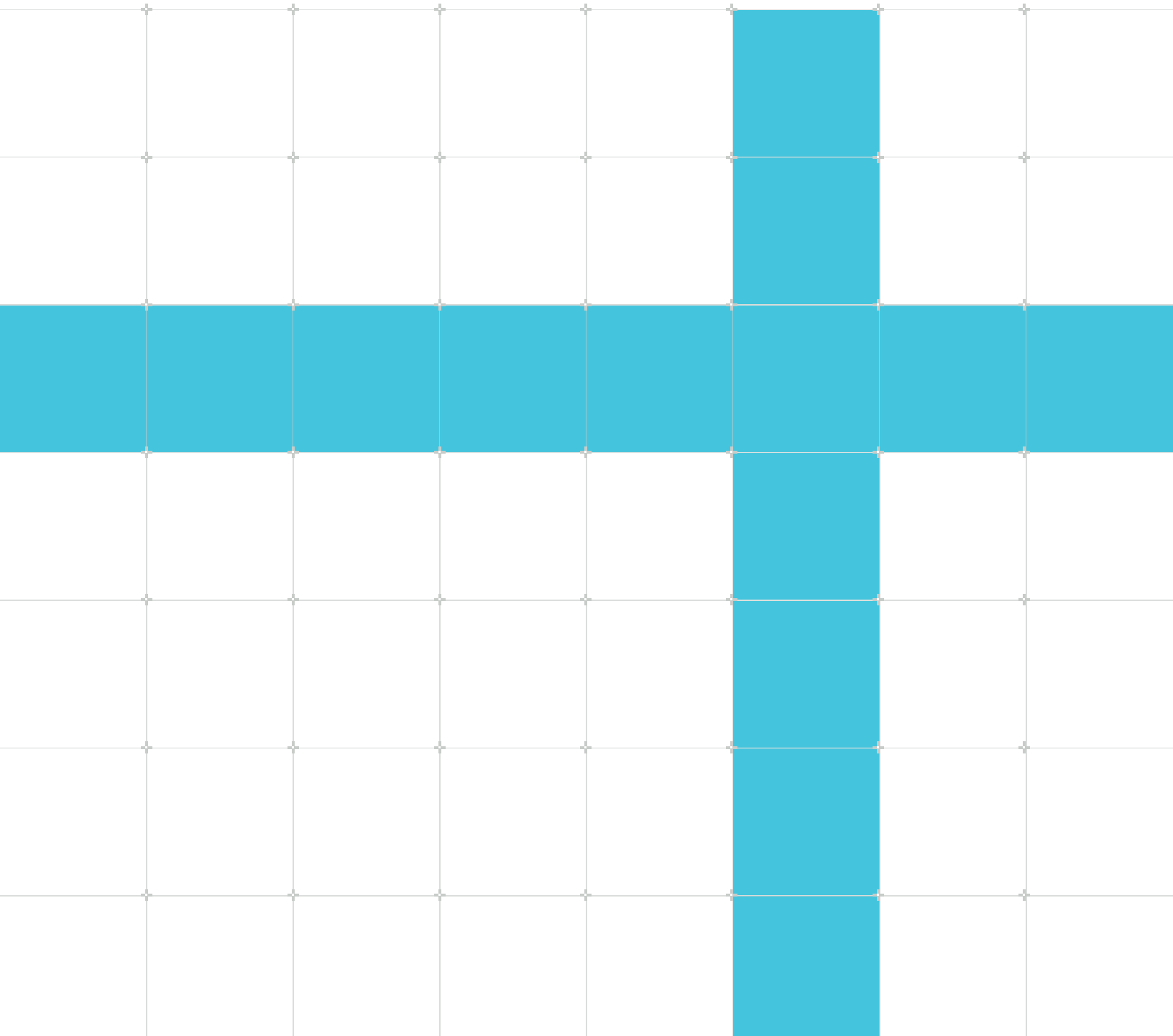
Arm® Corstone™-101 Reference Package

Revision: r0p1

Technical Overview

Non-Confidential

Issue 01
101147_0001_01_en



Arm® Corstone™-101 Reference Package

Technical Overview

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	13 October 2017	Non-Confidential	First release for r0p0
0001-00	5 October 2018	Non-Confidential	First release for r0p1
0001-01	29 July 2022	Non-Confidential	Second release for r0p1

Proprietary Notice

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.

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.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

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

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

Inclusive language commitment

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

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

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

Contents

1. Introduction

1.1 Product revision status

The r*x*p*y* identifier indicates the revision status of the product described in this manual, for example, r1p2, where:

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

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

1.2 Intended audience

This book is written for hardware or software engineers who want an overview of the functionality in the Corstone-101 Reference Package.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Typographic conventions

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

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

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

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

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

An important piece of information that needs your attention.

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

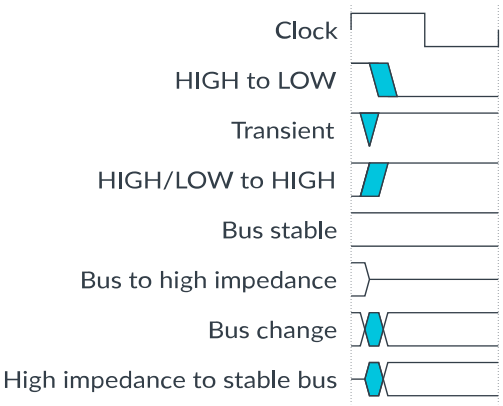
A reminder of something important that relates to the information you are reading.

Timing diagrams

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

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

1.4 Additional reading

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

Table 1-2: Arm publications

Document name	Document ID	Licensee only
Arm® AMBA® APB Protocol Specification, Version 2.0	IHI 0024	No
Arm® CoreLink™ CG092 AHB Flash Cache Technical Reference Manual	DDI 0569	No
Arm® CoreLink™ GFC-100 Generic Flash Controller Configuration and Integration Manual	101060	Yes
Arm® CoreLink™ GFC-100 Generic Flash Controller Technical Reference Manual	101059	Yes
Arm® Corstone-101 Reference Package Release Note	EPM-136363	No
Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual	100919	Yes
Arm® Corstone™ SSE-050 Subsystem Technical Reference Manual	100918	No

Document name	Document ID	Licensee only
Arm® Cortex®-M System Design Kit Example System Guide	DUI 0594	Yes
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® Cortex®-M0 and M0+ System Design Kit Example System Guide	DUI 0559	Yes
Arm® Cortex®-M3 Processor Technical Reference Manual	100165	No
Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual	DDI 0224	No

Confidential documents are available to licensees only, through a product bundle.

- See www.arm.com/cmsis for embedded software development resources including the *Cortex Microcontroller Software Interface Standard* (CMSIS).
- See Arm® Mbed™ platform, https://www.mbed.com for information on the Mbed™ tools including Mbed™ OS and online tools.

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

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

2. Overview

This chapter introduces the Arm® Corstone™-101 reference package (Corstone-101).

2.1 About the Corstone-101

The Corstone-101 provides a subsystem architecture, a reference platform, and a collection of IP products that can be used to create an IoT endpoint system. To provide this functionality, the Corstone-101 product grants licenses to the following subsystem and component IP products:

Corstone™ SSE-050 Subsystem

The SSE-050 provides a starting point for a product in the *Internet of Things* (IoT) and embedded market segments. The SSE-050 provides a process and technology agnostic reference, pre-integrated, validated, hardware and software subsystem for Arm® Cortex®-M3 processors that can be extended to provide an IoT endpoint system.

Cortex®-M System Design Kit

The Cortex®-M System Design Kit provides an example system for the Arm® Cortex®-M processors and reusable AMBA® AHB-Lite and APB components for low-power designs.

Cortex®-M0 and M0+ System Design Kit

The Cortex®-M0 and Cortex®-M0+ System Design Kit provides an example system-level design for the Arm® Cortex-M0 and Cortex-M0+ processors and reusable AMBA® interconnect components for system-level development.

CoreLink CG092 AHB Flash Cache

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

PrimeCell Real Time Clock

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

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

GFC-100 Generic Flash Controller

The GFC-100 Generic Flash Controller enables embedded Flash macros to be integrated easily into any system.

2.1.1 Using the Corstone IP products

The Corstone-101 licensed IP can be used in the following ways:

- Use the Corstone™ SSE-050 subsystem as a verified foundation for your own IoT solution that is based around the Cortex®-M3 processor.
- Use the *Cortex-M System Design Kit* (CMSDK) and the example system as a starting point for your own IoT solution that is based around the Cortex-M0, Cortex-M0+, Cortex-M3, or Cortex-M4 processors.
- Use the Cortex-M0 and Cortex-M0+ System Design Kit and the example system as a foundation for your own IoT solution that is based around the Cortex-M0 or Cortex-M0+ processors.
- Use the system IP provided with the SSE-050, CMSDK, CG092, 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.

- The SSE-050 and CMSDK build scripts include interconnections to a processor, but a processor must be separately licensed and installed.
- See the *Arm® Corstone-101 Reference Package Release Note* for details about how to download and install the Corstone-101 components that you require.

2.2 Product deliverables

The Corstone-101 product bundle does not have hardware or software deliverables. Its subsystems and IP component products include these deliverables.

The hardware deliverables must be downloaded separately for the following IP products that are included in the Corstone-101 license:

- Corstone™ SSE-050 Subsystem (CG063).
- Cortex®-M System Design Kit (BP210).
- Cortex®-M0 and M0+ System Design Kit (BP200).
- CoreLink™ CG092 AHB Flash Cache (CG092).
- PrimeCell Real Time Clock (PL031).
- GFC-100 Generic Flash Controller (CG090).

See the *Arm® Corstone-101 Reference Package Release Note* for the component versions.

2.3 Compliance

See the *Technical Reference Manuals* for more details of the product's compliance to the following specifications:

- Arm® architecture.
- CoreSight™ debug.
- Interrupt controller architecture.
- Advanced Microcontroller Bus Architecture.
- AMBA® Generic Flash Bus.

2.4 Documentation

The following documents are supplied with the Corstone-101 product bundle:

Technical Overview

The *Technical Overview* (TO) describes the functionality of the Corstone-101.

Release Note

The *Release Note* describes download and installation instructions for the IP products included in the Corstone-101.

- 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. Functional overview

This chapter describes the IP products included in the Corstone-101 license.

3.1 Supported processors

The following table lists the processors supported by the products in the Corstone-101 bundle.

Table 3-1: Supported processors

Corstone-101 products						
Processor	**CM0SDK**	**CMSDK**	**AHB Flash Cache**	**SSE-050 Subsystem**	**RTC**	**GFC-100**
Cortex-M0	Yes	-	Yes	-	Yes	Yes
Cortex-M0+	Yes	-	Yes	-	Yes	Yes
Cortex-M23	-	Yes	Yes	-	Yes	Yes
Cortex-M3	-	Yes	Yes	Yes	Yes	Yes
Cortex-M4	-	Yes	Yes	-	Yes	Yes

3.2 Corstone™ SSE-050 Subsystem

The Arm® Corstone™ SSE-050 Subsystem is a subsystem that provides a starting point for a product in the *Internet of Things* (IoT) and embedded market segments.

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

The SSE solution consists of hardware, software, and software tools to enable the rapid development of IoT *System on Chip* (SoC) solutions.

The SSE-050 contains the following components:

- A Cortex®-M3 processor:
  - Bit-banding enables using standard instructions to read or modify individual bits. The default implementation does not include bit banding.
  - Eight MPU regions (optional).
  - NVIC providing deterministic, high-performance interrupt handling with a configurable number of interrupts.
  - *Wakeup Interrupt Controller* (WIC) with configurable number of WIC lines (optional). This is a latch-based WIC implementation, and not the standard Cortex®-M3 WIC. See the *Arm® Corstone™ SSE-050 Subsystem Configuration and Integration Manual* for more information.

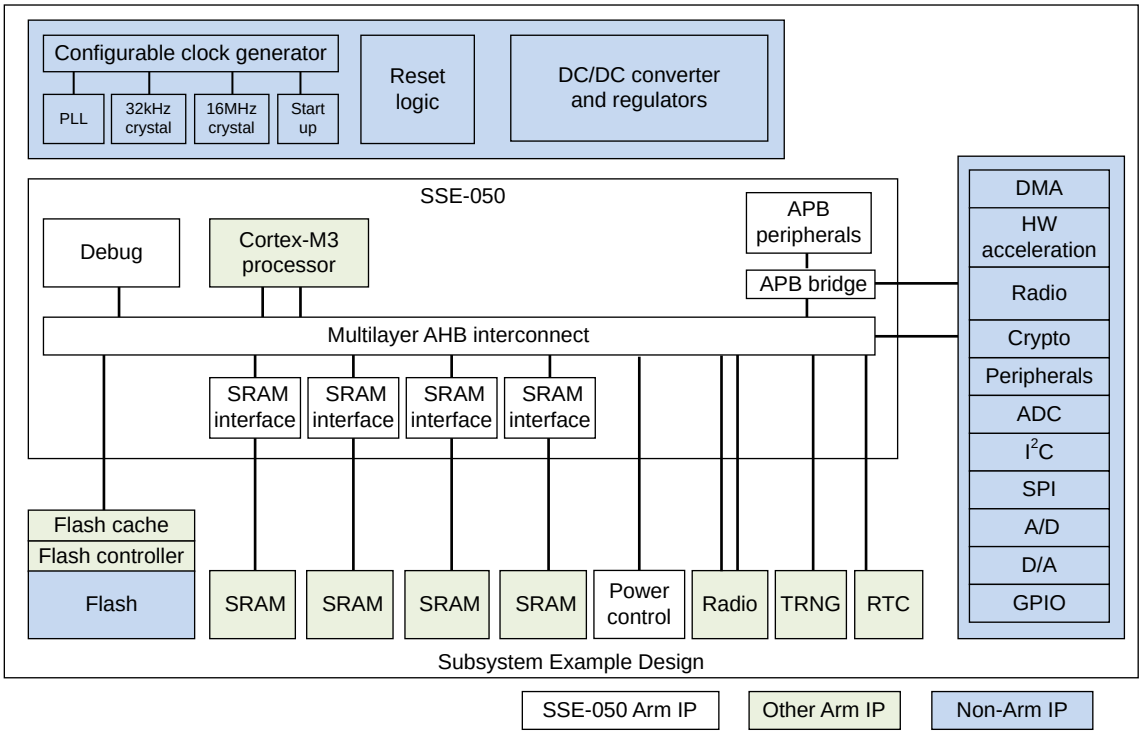
    - Little-endian memory addressing only (for compatibility with the eFlash cache).

    For more information, see the *Arm® Cortex®-M3 Processor Technical Reference Manual*.

    The Cortex®-M3 processor has a *Processor Integration Layer* (PIL) to simplify integration of the SSE-050 Subsystem into a multiprocessor system with a SoC-level CoreSight™ subsystem.
- Configurable Debug and Trace as either:
    - Standalone system with a *Trace Port Interface Unit* (TPIU) and an SWJ-DP.
    - Full CoreSight integration over a DAP and the ATB buses.
- Multilayer AMBA AHB-Lite interconnect:
    - Low-latency interconnect bus matrix.
    - Two AHB-Lite initiator expansion ports for external AHB masters.
    - Two AHB-Lite target expansion ports for external AHB slaves.
    - 11 APB4 target expansion ports (each with 4KB address space) to connect APB peripherals.
- Memory system, consisting of:
    - Placeholder for eFlash controller and optional cache.
    - Static memory (configurable as one to four 32KB banks) is provided in the example integration layer.
- Two APB timers:
    - Interrupt generation when the counter reaches 0.
    - Each timer has an TIMERnEXTIN signal that can be used as an enable or external clock.
    - Configurable privileged access mode.

The following figure shows the SSE-050 Subsystem, with other IP, in an example design.

Figure 3-1: SSE-050 Subsystem example design



3.2.1 Subsystem for Embedded software

Application processor firmware, which is available separately, consists of the code that is required to boot the subsystem up to the point where the OS execution starts. Contact your Arm representative for details on the software and its location.

3.3 Cortex-M System Design Kit

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

The design kit contains the following:

- A selection of AHB-Lite and APB components, including several peripherals such as GPIO, timers, watchdog, and UART.

  These components are used in the CMSDK example system, but you can also use the components to create your own custom system.
- An example system for supported processor products.
- Example synthesis scripts for the example system.

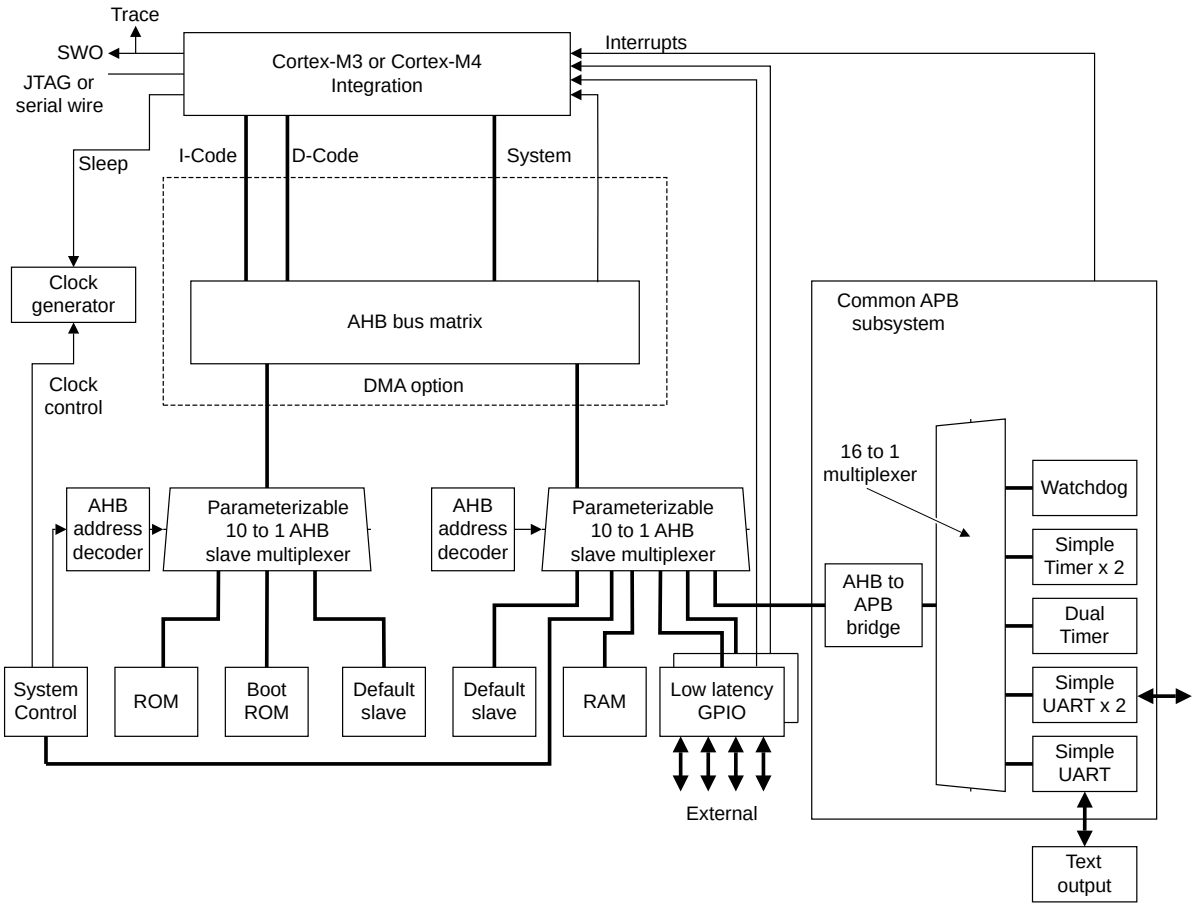
- Example compilation and simulation scripts for the Verilog environment that supports ModelSim, VCS, and NC Verilog.
- Example code for software drivers.
- Example test code to demonstrate various operations of the systems.
- Example compilation scripts and example software project files that support:
  - Arm Development Studio 5 (DS-5).
  - Arm RealView Development Suite.
  - Keil® *Microcontroller Development Kit* (MDK).
  - GNU tools for Arm embedded processors (Arm GCC).
- Documentation including:
  - *Arm® Cortex®-M System Design Kit Technical Reference Manual*.
  - *Arm® Cortex®-M0 and Cortex®-M0+ System Design Kit Example System Guide*.
  - *Arm® Cortex®-M System Design Kit Example System Guide*.

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

3.3.1 Example system

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

Figure 3-2: CMSDK example system



3.3.2 Components

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

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

3.3.3 Cortex-M Software Design Kit software

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

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

3.4 Cortex-M0 and M0+ System Design Kit

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

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

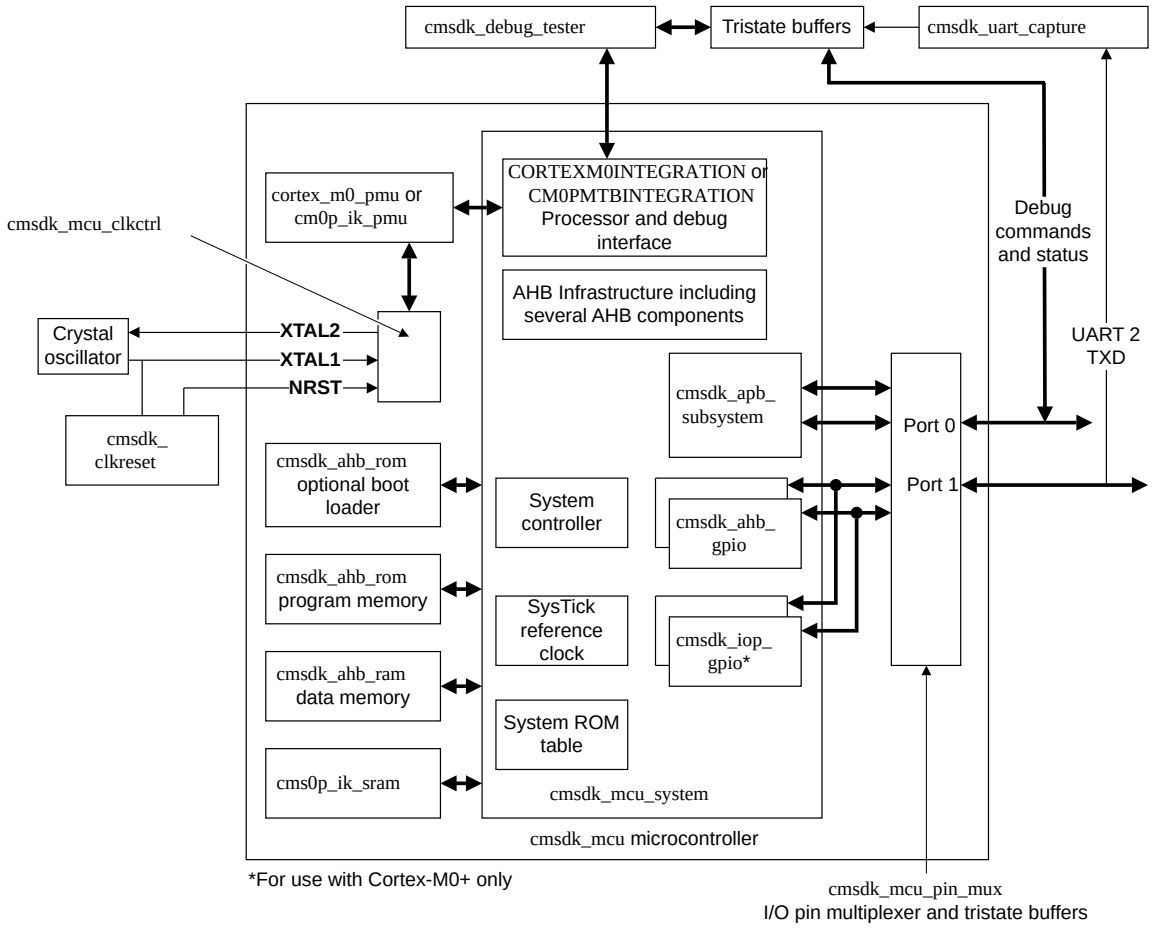
For information on the AMBA® components that the design kit uses, see the *Arm® Cortex®-M System Design Kit Technical Reference Manual*.

3.4.1 About the example system

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

The following figure shows the block diagram for the example system.

Figure 3-3: CMSDK example system



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

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


Note

The optional µDMA Controller (PL230) is not included in the Corstone-101 license and, if instantiated, must be licensed separately. See the *Arm® PrimeCell µDMA Controller (PL230) Technical Reference Manual* for more information.

3.4.2 Cortex-M0 and Cortex-M0+ software

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

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

3.5 About CoreLink CG092 AHB Flash Cache

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

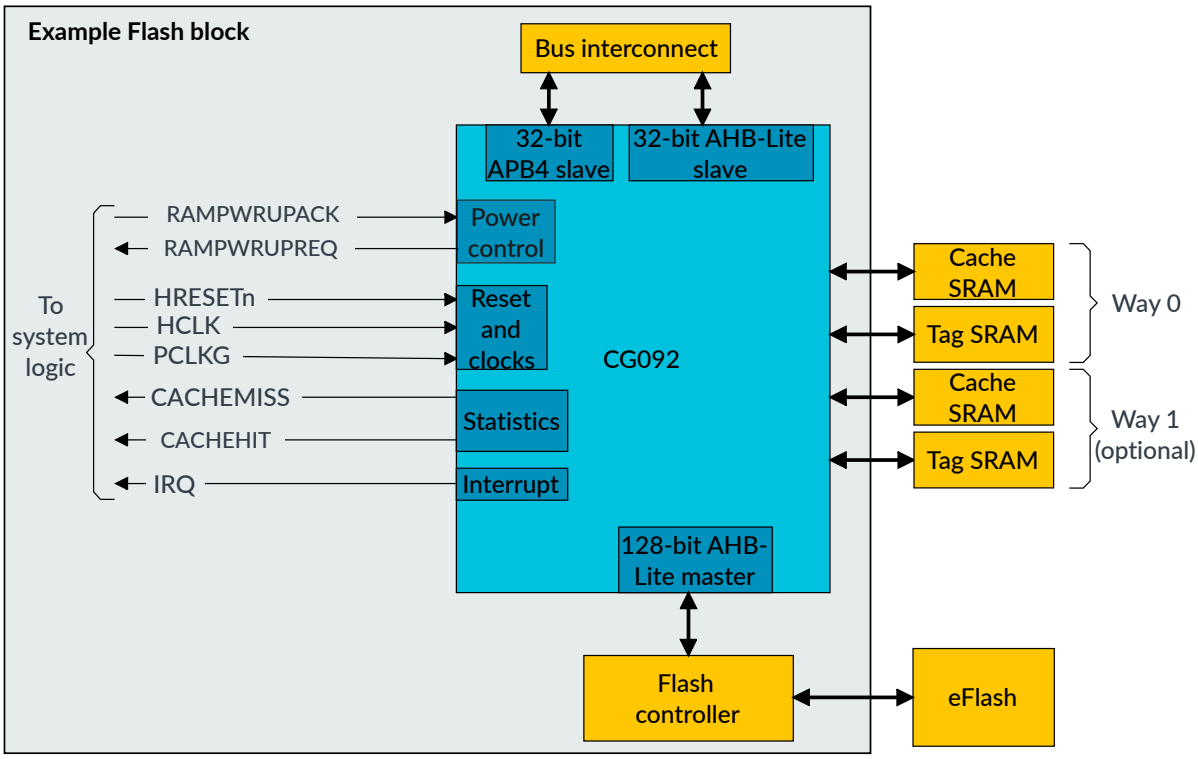
The CG092 is a simple cache for on-chip *embedded Flash* (eFlash). The CG092 design is optimized for fetching Cortex®-M3 or Cortex®-M4 instructions directly from an eFlash. The main benefit of the CG092 is improved power efficiency, but there are also improvements in code fetching performance.


Note

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 3-4: Example eFlash implementation



3.6 GFC-100 Generic Flash Controller

The Arm® CoreLink™ 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.

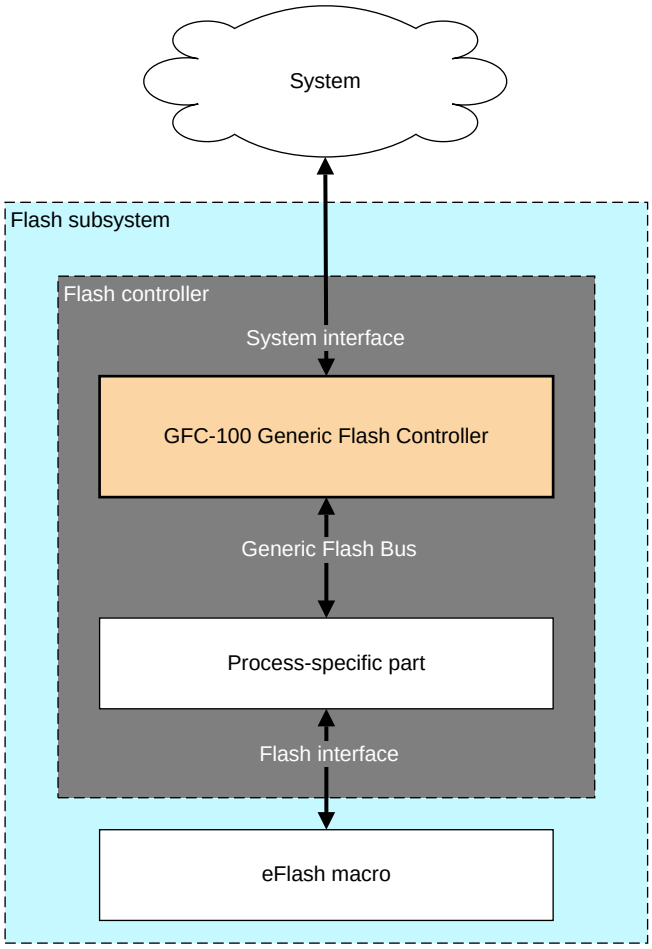
GFC-100 provides the functions that relate only to services for the system side of the Flash controller. GFC-100 cannot communicate directly with the eFlash macro. Therefore, GFC-100 must be integrated with a process-specific part that connects to, and communicates with, the eFlash macro.

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



Note

The process-specific part is not included and must be either licensed or designed separately.

GFC-100 provides several interfaces and test features:

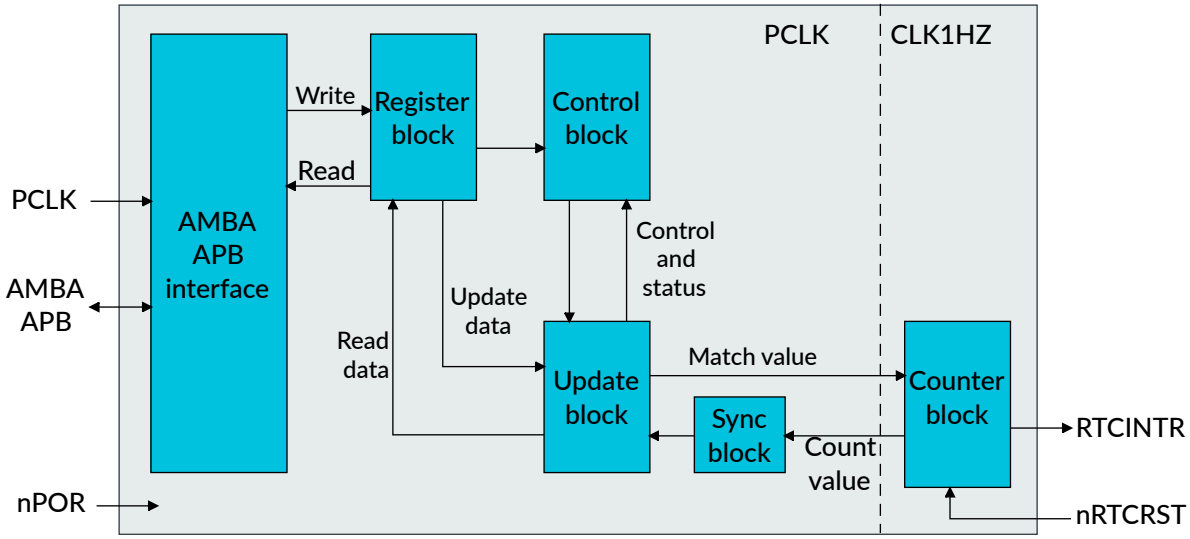
- *Advanced High-performance Bus* (AHB) interface.
- *Advanced Peripheral Bus* (APB) slave interface.
- APB register master interface.
- Q-Channel interface.
- P-Channel master interface.
- *Generic Flash Bus* (GFB)

3.7 Real Time Clock

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

The following figure shows the RTC block diagram.

Figure 3-6: RTC block diagram



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

Appendix A Revisions

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

A.1 Revisions

Table A-1: Issue 0000-00

Change	Location	Affects
First release	-	-

Table A-2: Differences between issue 0000-00 and issue 0001-00

Change	Location	Affects
Name of Gereric Flash Controller changed to *CoreLink™ GFC-100*.	2.1 About the Corstone-101 on page 10 2.2 Product deliverables on page 11 3.6 GFC-100 Generic Flash Controller on page 21	Issue 0001-00

Table A-3: Differences between issue 0001-00 and issue 0001-01

Change	Location	Affects
Stand-alone TRNG content removed.	2.1 About the Corstone-101 on page 10 3.1 Supported processors on page 13 3. Functional overview on page 13	Issue 0001-01