ARM® CoreLink™ SDK-100 System Design Kit

Revision: r0p0

Technical Overview

ARM 101062_0000_00_en

ARM® CoreLink™ SDK-100 System Design Kit

Technical Overview

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	16 June 2017	Non-Confidential	First release for r0p0

Non-Confidential Proprietary Notice

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

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

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

http://www.arm.com

Contents
ARM® CoreLink™ SDK-100 System Design Kit Technical Overview

Preface

This preface introduces the *ARM® CoreLink™ SDK-100 System Design Kit Technical Overview*.

It contains the following:

About this book

This book is for the ARM® CoreLink™ SDK-100 System Design Kit (SDK-100). It describes the hardware and software for the system.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

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

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

Intended audience

This book is written for hardware or software engineers who want an overview of the functionality in the CoreLink™ SDK-100 System Design Kit.

Using this book

This book is organized into the following chapters:

Chapter 1 SDK-100 overview
This chapter introduces the ARM CoreLink SDK-100 System Design Kit (SDK-100).

Chapter 2 Functional overview
This chapter describes the IP products included in the SDK-100 licence.

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

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.

Typographic conventions

italic
Introduces special terminology, denotes cross-references, and citations.

bold
Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`
Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`monospace`
Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`
Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`
Denotes language keywords when used outside example code.

`<and>`
Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS
: Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

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.

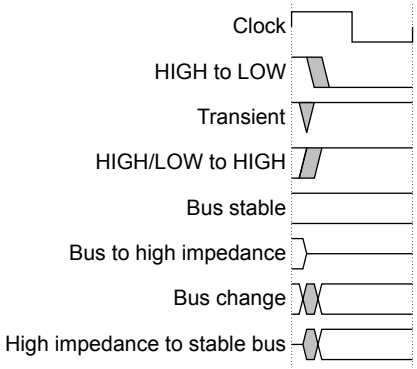

Figure 1 Key to timing diagram conventions

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 denotes an active-LOW signal.

Additional reading

This section lists publications by ARM and by third parties.

See *Infocenter*, for access to ARM documentation.

ARM publications

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

- *ARM® Cortex® -M3 Processor Technical Reference Manual* (ARM 100165).
- *ARM® Cortex®-M System Design Kit Technical Reference Manual* (ARM DDI 0479).
- *ARM® CoreLink™ CG092 AHB Flash Cache Technical Reference Manual* (ARM DDI 0569).
- *ARM® PrimeCell Real Time Clock (PL031) Technical Reference Manual* (ARM DDI 0224).
- *ARM® TrustZone® True Random Number Generator Technical Reference Manual* (ARM 100976).
- *ARM® CoreLink™ SDK-100 Release Note* (ARM EPM 132828).
- *ARM® AMBA® APB Protocol Specification* (ARM IHI 0024).

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

- *ARM® CoreLink™ SSE-050 Subsystem Configuration and Integration Manual* (ARM 100919).
- *ARM® Cortex®-M0 and M0+ System Design Kit Example System Guide* (ARM DUI 0559).
- *ARM® Cortex®-M System Design Kit Example System Guide* (ARM DUI 0594).

Note

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

Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to *errata@arm.com*. Give:

- The title *ARM CoreLink SDK-100 System Design Kit Technical Overview*.
- The number ARM 101062_0000_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Chapter 1
SDK-100 overview

This chapter introduces the ARM CoreLink SDK-100 System Design Kit (SDK-100).

It contains the following sections:

1.1 About the SDK-100

The SDK-100 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 SDK-100 product grants licensees to the following subsystem and component IP products:

CoreLink 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 components for system-level development.

CoreLink CG092 AHB Flash Cache
: The CG092 is an instruction cache 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.

TrustZone True Random Number Generator
: The *True Random Number Generator* (TRNG) provides an assured level of entropy (as analyzed by Entropy Estimation logic). The output from the TRNG can be used to seed deterministic random bit generators.

1.1.1 Using the SDK IP products

The SDK-100 licensed IP can be used in the following ways:

- Use the CoreLink SSE-050 Subsystem as a verified foundation for your own IoT solution based around the Cortex-M3.
- Use the *Cortex-M System Design Kit* (CMSDK) and the example system as a starting point for your own IoT solution 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 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.

——— **Note** ———

- The SSE-050 and CMSDK build scripts include interconnections to a processor, but a processor must be separately licensed and installed.
- See the *ARM® CoreLink™ SDK-100 Release Notes* for details on how to download and install the SDK-100 components you require.

———————————

1.2 Product deliverables

The CoreLink SDK-100 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 SDK-100 licence:

- CoreLink 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).
- TrustZone True Random Number Generator (CC003).

See the *ARM® CoreLink™ SDK-100 Release Note* for the component versions.

1.3 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.

1.4 Documentation

The following documents are supplied with the CoreLink SDK-100 product bundle:

Technical Overview
: The *Technical Overview* (TO) describes the functionality of the SDK-100 System Design Kit.

Release Note
: The *Release Note* describes download and installation instructions for the IP products included in the SDK-100.

Note

- 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.

1.5 Product revisions

This section describes the differences in functionality between product revisions.

r0p0

First release.

Chapter 2
Functional overview

This chapter describes the IP products included in the SDK-100 licence.

It contains the following sections:

2.1 Supported processors

The following table lists the processors supported by the products in the CoreLink SDK-100 bundle.

Table 2-1 Supported processors

CoreLink SDK-100 products						
Processor	**CM0SDK**	**CMSDK**	**AHB Flash Cache**	**SSE-050 Subsystem**	**RTC**	**TRNG**
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

2.2 CoreLink SSE-050 Subsystem

The SSE-050 Subsystem 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 solution consists of hardware, software, and software tools to enable the rapid development of IoT *System on Chip* (SoC) solutions.

The SSE-050 Subsystem contains the following components:

- A Cortex-M3 processor:
  - Bit-banding enables using standard instructions to read or modify of 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® CoreLink™ 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.

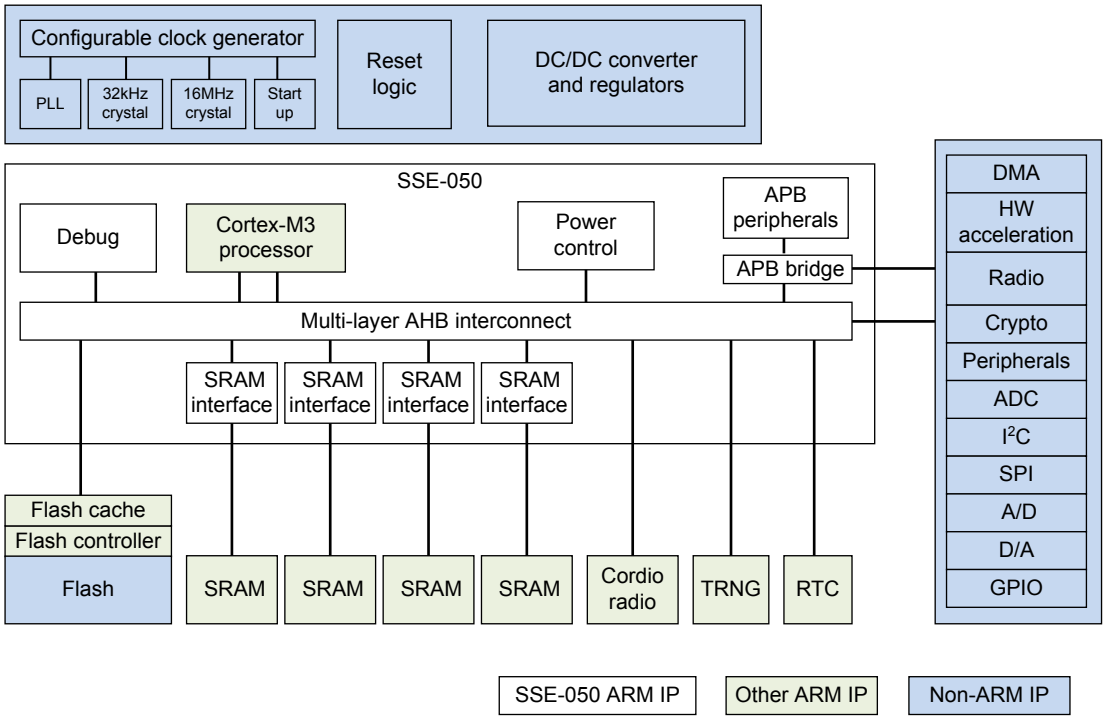

Figure 2-1 SSE-050 Subsystem example design

2.2.1 SSE-050 software

Application processor firmware, which is available separately, consists of the code 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.

The firmware contains:

- *Cortex Microcontroller Software Interface Standard* (CMSIS) compliant drivers.
- Flash programming support code, which is separate from mbed OS.
- Separately ported mbed OS that includes uVisor ported onto the SSE-050 system.
- mbed OS driver support and code for example IO peripherals.
- Code required to load mbed from boot media and set up the initial security environment.

2.3 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 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*.

This section contains the following subsections:

- *2.3.1 Example system* on page 2-20.
- *2.3.2 Components* on page 2-21.
- *2.3.3 Cortex-M Software Design Kit software* on page 2-21.

2.3.1 Example system

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

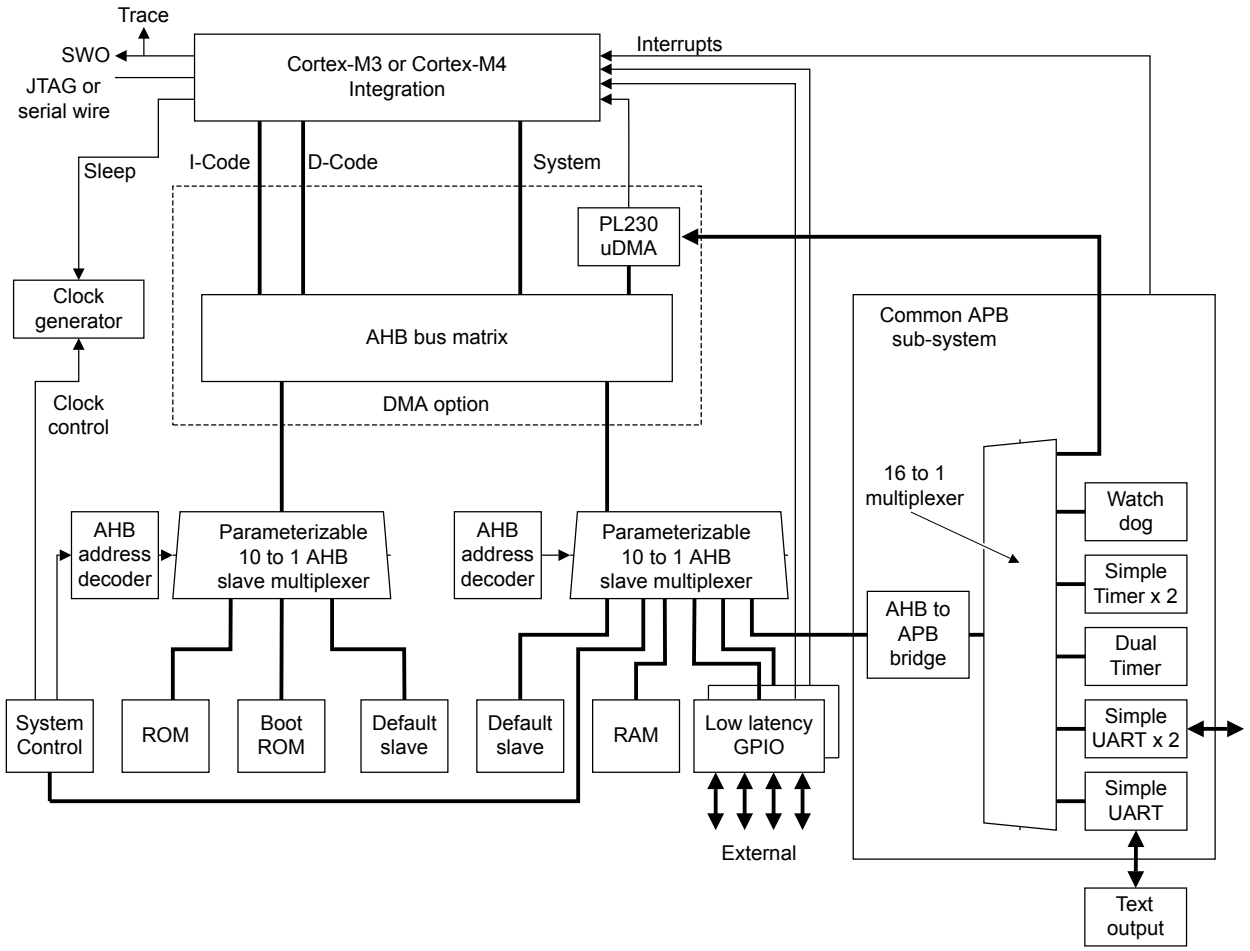

Figure 2-2 CMSDK example system

2.3.2 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.

2.3.3 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 retargetting code for the `printf()` and `puts()` functions.
- Platform hardware adaptation layer code that is required in addition to the open-source code and generic Cortex-M processor header files.
- mbed OS driver support.

  Additional Cortex-M code is available on the mbed website.
- Shell scripts to sync, build, and run the software.

2.4 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*.

This section contains the following subsections:

- *2.4.1 About the example system* on page 2-22.
- *2.4.2 Cortex-M0 and Cortex-M0+ software* on page 2-23.

2.4.1 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.

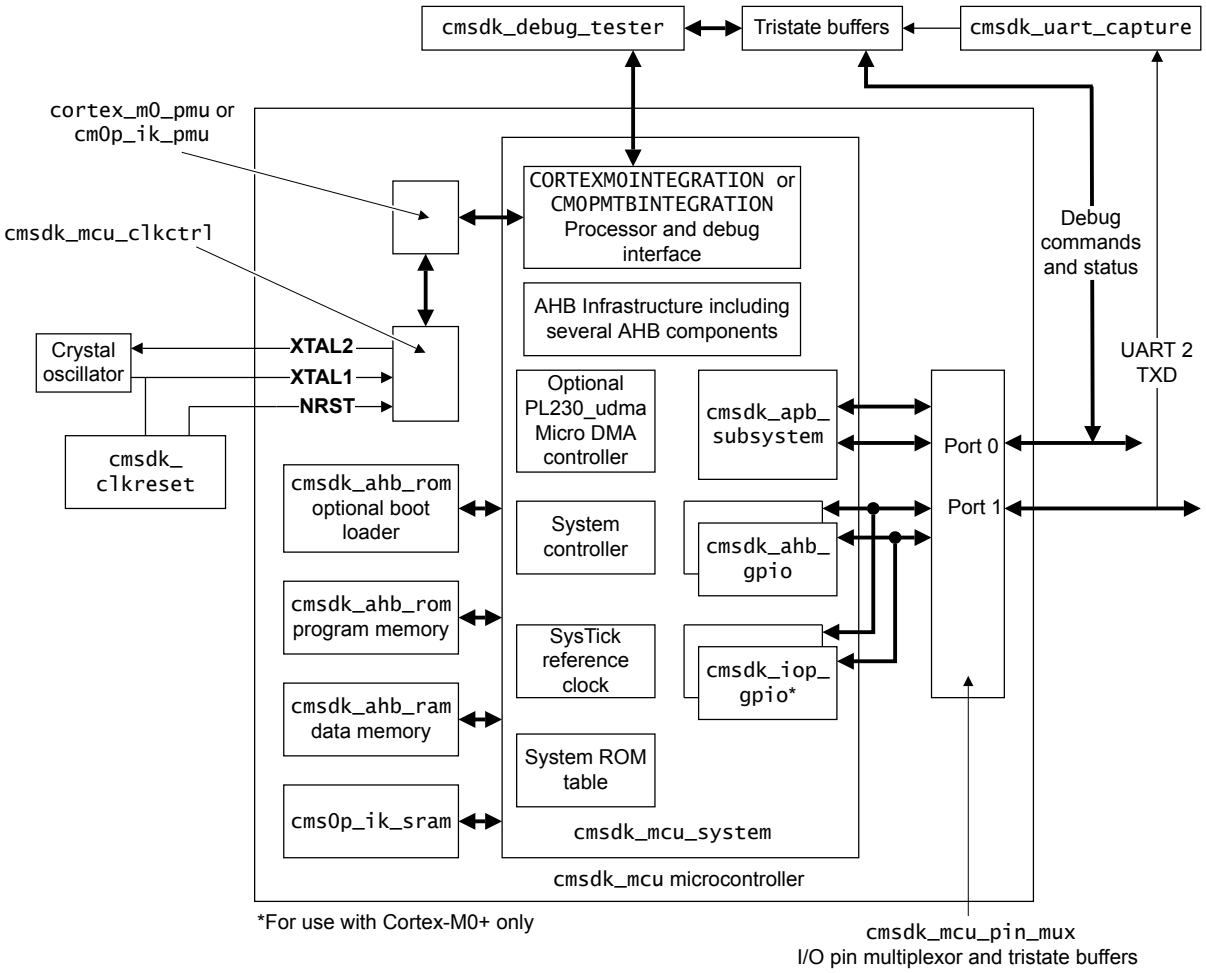

Figure 2-3 CMSDK example system

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.

2.4.2 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 retargetting code for the `printf()` and `puts()` functions.
- Platform hardware adaptation layer code that is required in addition to the open-source code and generic Cortex-M processor header files.
- mbed OS driver support.

  Additional Cortex-M0 and Cortex-M0+ code is available on the mbed website.
- Shell scripts to sync, build, and run the software.

2.5 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.

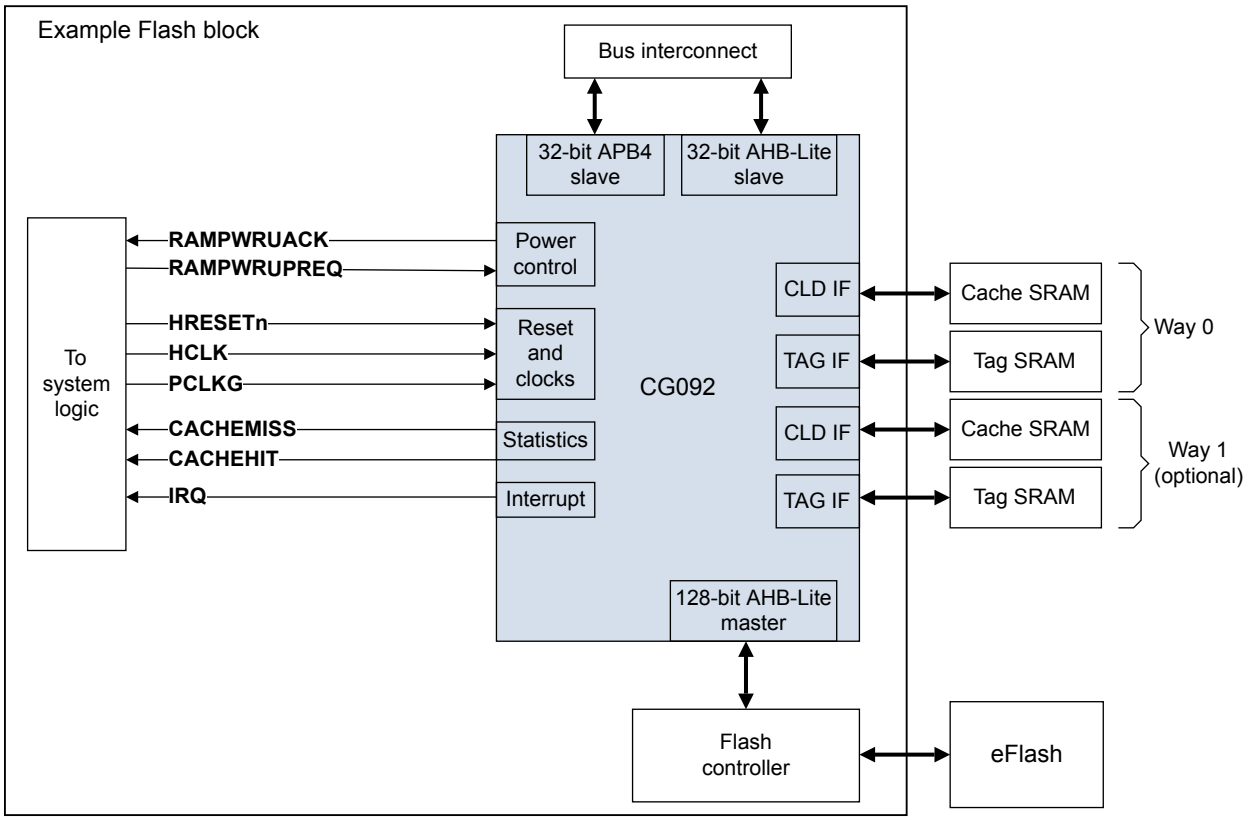

Figure 2-4 Example eFlash implementation

2.6 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.

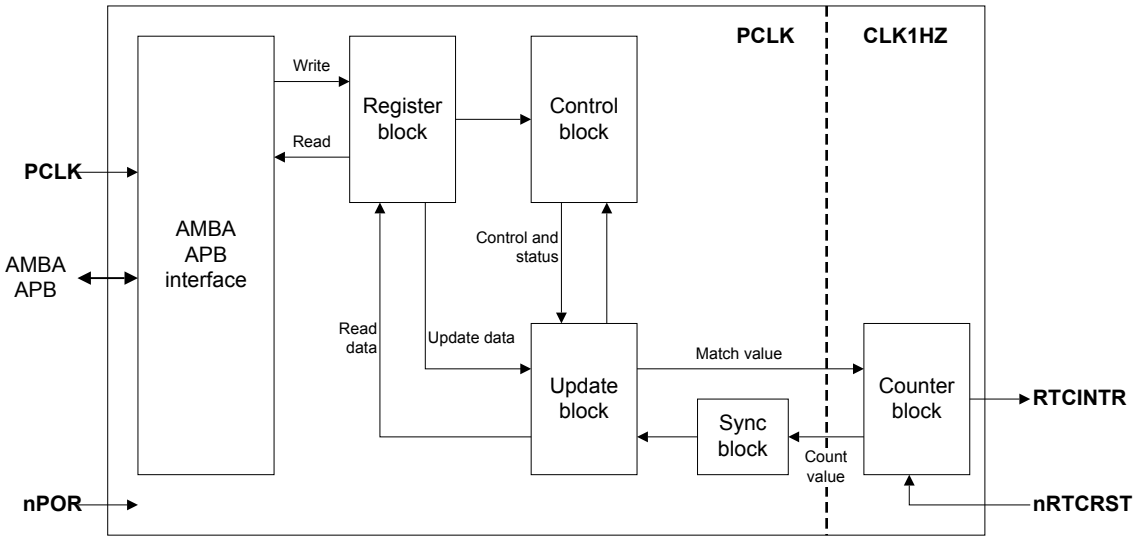

Figure 2-5 RTC block diagram

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.

2.7 True Random Number Generator

The *True Random Number Generator* (TRNG) provides an assured level of entropy (as analyzed by Entropy Estimation logic). The output from the TRNG can be used to seed deterministic random bit generators.

The following figure shows the TRNG.

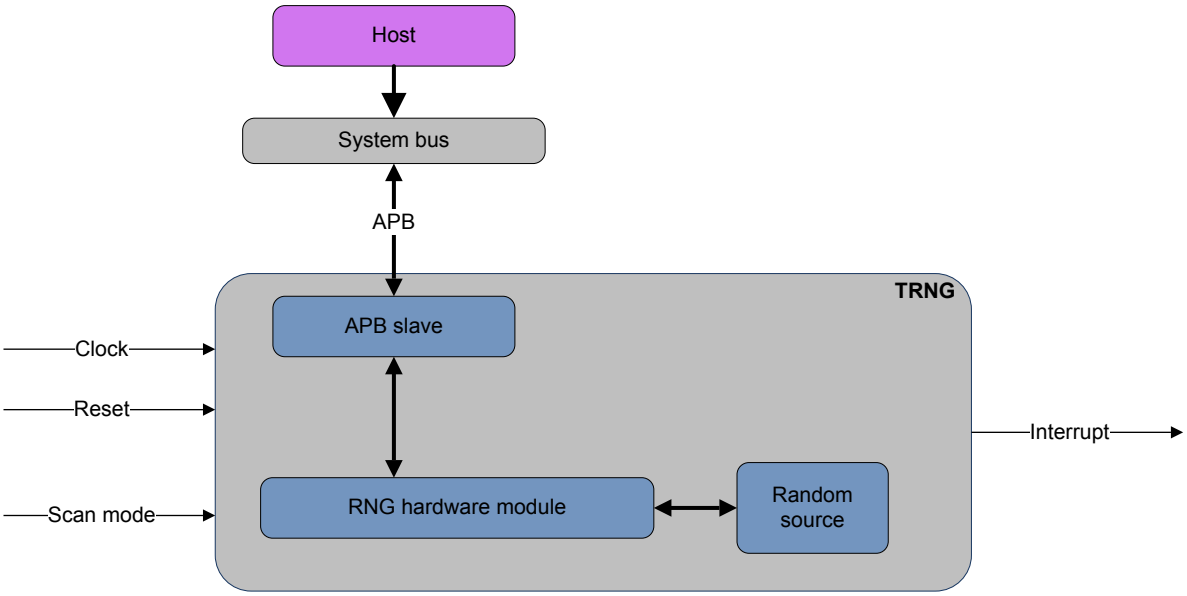

Figure 2-6 TRNG hardware overview

Appendix A
Revisions

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

It contains the following section:

- *A.1 Revisions* on page Appx-A-28.

A.1 Revisions

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

Table A-1 Issue 0000-00

Change	Location	Affects
First release	-	-