

ARM® Cortex®-M3 DesignStart™ Eval

Revision: r0p0

Customization Guide



ARM® Cortex®-M3 DesignStart™ Eval

Customization Guide

Copyright © 2017 ARM Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	14 June 2017	Non-Confidential	First release for r0p0

Non-Confidential Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

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

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

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited or its affiliates in the EU and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow ARM’s trademark usage guidelines at <http://www.arm.com/about/trademark-usage-guidelines.php>

Copyright © 2017, ARM Limited or its affiliates. All rights reserved.

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

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

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

<http://www.arm.com>

Contents

ARM® Cortex®-M3 DesignStart™ Eval Customization Guide

Preface

About this book	7
Feedback	10

Chapter 1

Introduction

1.1 About Cortex®-M3 DesignStart™ Eval	1-12
1.2 Using the documentation	1-14

Chapter 2

Typical customizations

2.1 Replacing existing peripherals	2-17
2.2 Adding peripherals to memory map	2-20
2.3 Low-power optimizations	2-21
2.4 Multiple processor designs	2-22
2.5 Migrating from Cortex®-M3 DesignStart™ Eval to Cortex®-M3 DesignStart™ Pro	2-23

Chapter 3

Using ARM Socrates™ to update the system

3.1 ARM Socrates™ tools	3-25
-------------------------------	------

Chapter 4

Validation and ASIC implementation

4.1 Validation of design changes	4-27
4.2 Key points for ASIC implementation	4-28

Chapter 5	Customizations in the FPGA Evaluation Flow	
5.1	FPGA flow	5-30
5.2	Increasing the processor clock speed	5-31
5.3	Porting to a different platform	5-32
Appendix A	Revisions	
A.1	Revisions - Cortex®-M3 DesignStart™ Eval	Appx-A-35

Preface

This preface introduces the *ARM® Cortex®-M3 DesignStart™ Eval Customization Guide*.

It contains the following:

- *About this book* on page 7.
- *Feedback* on page 10.

About this book

This book describes how to integrate peripherals, and make other modifications to the system delivered with Cortex®-M3 DesignStart™ Eval.

Product revision status

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

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

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

Intended audience

This book is written for hardware engineers, system integrators, and system designers, who might not have previous experience of ARM products, but want to understand the process of generating their own customized system.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This chapter introduces Cortex-M3 DesignStart Eval, its features, and its documentation structure.

Chapter 2 Typical customizations

This chapter describes the typical customizations that can be made on the example system in Cortex-M3 DesignStart Eval.

Chapter 3 Using ARM Socrates™ to update the system

This chapter describes how to use Socrates™ to update the system.

Chapter 4 Validation and ASIC implementation

This chapter describes the validation necessary for design changes, and the factors you must consider for ASIC implementation.

Chapter 5 Customizations in the FPGA Evaluation Flow

This chapter describes the typical customizations and the customization issues that might occur in the FPGA Evaluation Flow for Cortex-M3 DesignStart Eval.

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.

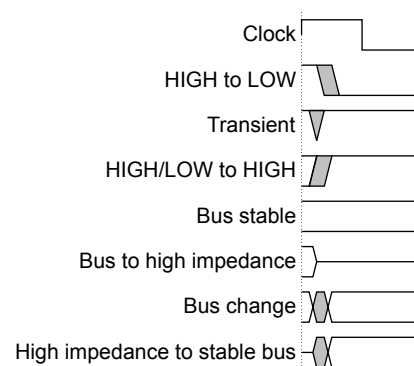


Figure 1 Key to timing diagram conventions

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

ARM publications

- Cortex®-M3 DesignStart™ Eval publications:
 - *ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide* (ARM 100894).
 - *ARM® Cortex®-M3 DesignStart™ Eval FPGA User Guide* (ARM 100896).
 - *ARM® Cortex®-M3 DesignStart™ Eval RTL and FPGA Quick Start Guide* (ARM 100895).
- Other ARM publications:
 - *ARM® Cortex®-M System Design Kit Technical Reference Manual* (ARM DDI0479).
 - *ARM® TrustZone® TRNG True Random Number Generator Technical Reference Manual* (ARM 1009676).
 - *ARM® PrimeCell™ Real Time Clock (PL031) Technical Reference Manual* (ARM DDI 0224).
 - *ARM® PrimeCell® Synchronous Serial Port (PL022) Technical Reference Manual* (ARM DDI 0194).
 - *ARM® Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual* (ARM 100112).
 - *Application Note AN531 uSDCARD SPI Adapter for the Cortex-M Prototyping System (MPS2+)* (ARM DAI 0531).
 - *Application Note AN502 Adapter for Arduino for the Cortex-M Prototyping System (MPS2 and MPS2+)* (ARM DAI 0502).
 - *ARM® AMBA® 3 AHB-Lite Protocol Specification (v1.0)* (ARM IHI 0033).
 - *ARM® Architecture Reference Manual ARMv7, for ARMv7-M architecture profile* (ARM DDI0403).
 - *ARM® Cortex®-M3 Technical Reference Manual* (ARM 100165).
 - *ARM® Cortex®-M3 Devices Generic User Guide* (ARM DUI0552).

Other publications

None.

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 Cortex-M3 DesignStart Eval Customization Guide*.
- The number ARM 100897_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

Introduction

This chapter introduces Cortex-M3 DesignStart Eval, its features, and its documentation structure.

It contains the following sections:

- [*1.1 About Cortex®-M3 DesignStart™ Eval*](#) on page 1-12.
- [*1.2 Using the documentation*](#) on page 1-14.

1.1 About Cortex®-M3 DesignStart™ Eval

Cortex-M3 DesignStart Eval provides developers an easy way to develop and simulate SoC designs based on the ARM Cortex-M3 processor. It allows a system designer to design and test on a simulator and then proceed with hardware prototyping using an FPGA.

The Cortex-M3 DesignStart Eval package is aimed at developers who are new to ARM or have limited soft IP system design experience. The package includes the following:

- [1.1.1 RTL on page 1-12.](#)
- [1.1.2 Execution Testbench on page 1-13.](#)
- [1.1.3 FPGA Evaluation Flow on page 1-13.](#)

Cortex-M3 DesignStart Eval provides an easy entry into the ARM ecosystem, rather than a complete solution for all Cortex-M processor design scenarios.

The hardware ecosystem in Cortex-M3 DesignStart Eval is built around the CoreLink™ SSE-050 Subsystem and includes the use of the *Cortex-M System Design Kit* (CMSDK) standard library of *Advanced High-performance Bus* (AHB) and *Advanced Peripheral Bus* (APB) components. For more information on the CMSDK, see the *ARM® Cortex®-M System Design Kit Technical Reference Manual*.

The software ecosystem in Cortex-M3 DesignStart Eval uses the ARM *Cortex Microcontroller Software Interface Standard* (CMSIS) software standard library.

The use of CMSDK and CMSIS, coupled with a reprogrammable FPGA, allows for a fast turnaround and prototyping of Cortex-M3 processor-based hardware and software.

Cortex-M3 DesignStart Eval does not support the implementation of the Cortex-M3 processor into silicon. Any implementation of the Cortex-M3 processor into silicon requires you to obtain Cortex-M3 DesignStart Pro, or take a full Cortex-M3 processor license from ARM.

A Cortex-M3 DesignStart Pro license offers the following:

- The Cortex-M3 processor.
- The SDK-100 *System Design Kit* (SDK), which includes:
 - The CoreLink SSE-050 Subsystem.
 - The CMSDK components.
 - A *Real Time Clock* (RTC).
 - A stand-alone *True Random Number Generation* (TRNG).

An *Embedded Trace Macrocell* (ETM) is not included in Cortex-M3 DesignStart Pro, and requires a separate license.

If you are working on ASIC implementation, then ARM recommends that you license Cortex-M3 DesignStart Pro as early as possible.

1.1.1 RTL

The RTL in Cortex-M3 DesignStart Eval includes the components and peripherals that are required to implement a complete example system in an FPGA.

The example system is intended to provide a reference starting point for a typical IoT endpoint application and is a supported ARM mbed™ platform when implemented on the ARM *Versatile Express Cortex-M Prototyping System* (V2M-MPS2+) platform.

The Cortex-M3 DesignStart Eval RTL provides an example system that includes:

- A Cortex-M3 processor in a fixed configuration (obfuscated but synthesizable).
- A modified CoreLink SSE-050 subsystem supporting a single Cortex-M3 processor with support for debug and trace.
- A memory subsystem supporting *Execute In Place* (XIP). The MPS2+ platform preloads a code file at powerup.
- Two timers for Operating System use (privileged access only).

- Peripherals for:
 - Application use, including Timers, UART, Watchdog, *Real Time Clock* (RTC), *True Random Number Generator* (TRNG).
 - MPS2+ platform, including Color LCD, Audio, and Ethernet.
 - Arduino Shield expansion using the adapter for the Arduino board.
- SPI interface supporting application persistent storage on microSD card.
- Reusable ARM *Advanced Microcontroller Bus Architecture* (AMBA) SoC interconnect components for system level development.

You must not modify the obfuscated Cortex-M3 processor (`cortexm3ds_logic.v`).

You are only permitted to redistribute the following files (modified or original), with the original headers unchanged, and any modifications clearly identified:

- `fpga_top.v`
- `m3ds_user_partition.v`
- `m3ds_peripherals_wrapper.v`

1.1.2 Execution Testbench

The Execution Testbench in Cortex-M3 DesignStart Eval is an RTL package that allows system design and simulation with a suitable Verilog simulator.

The Cortex-M3 DesignStart Eval Execution Testbench includes:

- A simulation model of the processor that includes register visibility and instruction execution tracing.
- Memory models that match the FPGA target.
- ARM CoreSight™ debug test engine that is preconfigured for a single fixed debug and trace implementation.
- Integration tests for memories and internal peripherals.

You are expected to modify the test code to support any modifications you make to your design. You must not redistribute any test code or binaries from these deliverables unless it is developed using mbed source code.

You are only permitted to redistribute the following files (modified or original), with the original headers unchanged, and any modifications clearly identified:

- `tb_fpga_shield.v`
- `cmsdk_uart_capture_ard.v`

1.1.3 FPGA Evaluation Flow

The Cortex-M3 DesignStart Eval FPGA Evaluation Flow allows developers to build an image file of the simulation system that can be used with the ARM *Versatile Express Cortex-M Prototyping System* (V2M-MPS2+). The FPGA image can be customized to the user system requirements.

The Cortex-M3 DesignStart Eval FPGA Evaluation Flow requires the purchase of the MPS2+ FPGA platform.

The MPS2+ FPGA platform includes a *Motherboard Configuration Controller* (MCC) on the baseboard, which provides the following features that are necessary to emulate an ARM mbed compliant system:

- Target application code. The target has no flash memory. The SRAM is instead initialized at powerup by the MCC using information stored on the configuration microSD card.
- DAPLink implementing CMSIS-DAP over USB for debug access.
- UART access is provided by a serial connector (and included serial to USB cable).
- *Real Time Clock* (RTC) initialization from baseboard processor on powerup.

For more information on how to use the MPS2+ FPGA platform, see the ARM® *Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual*.

You must not redistribute any FPGA bit files or other representations of the design that are produced from Cortex-M3 DesignStart Eval.

1.2 Using the documentation

There are several documents that are provided with Cortex-M3 DesignStart Eval.

Scope of this document

The *ARM® Cortex®-M3 DesignStart™ Eval Customization Guide* provides a high-level description on how to integrate your own peripherals and make other modifications in Cortex-M3 DesignStart Eval.

Some of the flows described in this document require you to license Cortex-M3 DesignStart Pro if they involve manufacture.

Other documents

The following table shows the documents that relate to the design flow processes for Cortex-M3 DesignStart Eval:

Table 1-1 Other Cortex-M3 DesignStart Eval documents

Document name	Purpose
<i>ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide</i>	Describes the information required for a system design and RTL simulation. ————— Note ————— This is the main document for Cortex-M3 DesignStart Eval. —————
<i>ARM® Cortex®-M3 DesignStart™ Eval FPGA User Guide</i>	Describes how to build an FPGA image and evaluate software running on the <i>Versatile Express Cortex-M Prototyping System</i> (V2M-MPS2+) platform.
<i>ARM® Cortex®-M3 DesignStart™ Eval RTL and FPGA Quick Start Guide</i>	Describes how to run basic tests using an RTL simulator and the MPS2+ FPGA platform. ————— Note ————— This is a procedural user-level document that gives a complete example of a working system. This document is highly recommended for users who do not have previous experience of ARM products. —————

For more information about:

- Programming the Cortex-M3 processor, see the *ARM® Cortex®-M3 Technical Reference Manual*.
- Software development on a Cortex-M3 device, see the *ARM® Cortex®-M3 Devices Generic User Guide*. This is a generic device user-level reference document.
- The ARM architecture that the Cortex-M3 processor complies with, and the instruction set and exception model it uses, see the *ARM® Architecture Reference Manual ARMv7, for ARMv7-M architecture profile*.
- The AHB-Lite master interface that the Cortex-M3 processor implements, see the *ARM® AMBA® 3 AHB-Lite Protocol Specification (v1.0)*.
- Peripherals and interconnect components, see the *ARM® Cortex®-M System Design Kit Technical Reference Manual*.
- The *Real Time Clock* (RTC), see the *ARM® PrimeCell™ Real Time Clock (PL031) Technical Reference Manual*.
- The *Serial Peripheral Interface* (SPI), see the *ARM® PrimeCell® Synchronous Serial Port (PL022) Technical Reference Manual*.

- The MPS2+ FPGA platform, see the *ARM® Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual*.
- The True Random Number Generator (TRNG), see the *ARM® TrustZone® TRNG True Random Number Generator Technical Reference Manual*.

Chapter 2

Typical customizations

This chapter describes the typical customizations that can be made on the example system in Cortex-M3 DesignStart Eval.

It contains the following sections:

- *2.1 Replacing existing peripherals* on page 2-17.
- *2.2 Adding peripherals to memory map* on page 2-20.
- *2.3 Low-power optimizations* on page 2-21.
- *2.4 Multiple processor designs* on page 2-22.
- *2.5 Migrating from Cortex®-M3 DesignStart™ Eval to Cortex®-M3 DesignStart™ Pro* on page 2-23.

2.1 Replacing existing peripherals

The Cortex-M3 DesignStart Eval system is representative of a fully functioning SoC design without any advanced power-saving functionality, and includes peripherals of several levels in hierarchy. You can replace existing peripherals to add design features or enhance performance.

The following diagram shows an overview of the example system in Cortex-M3 DesignStart Eval, and typical customizations:

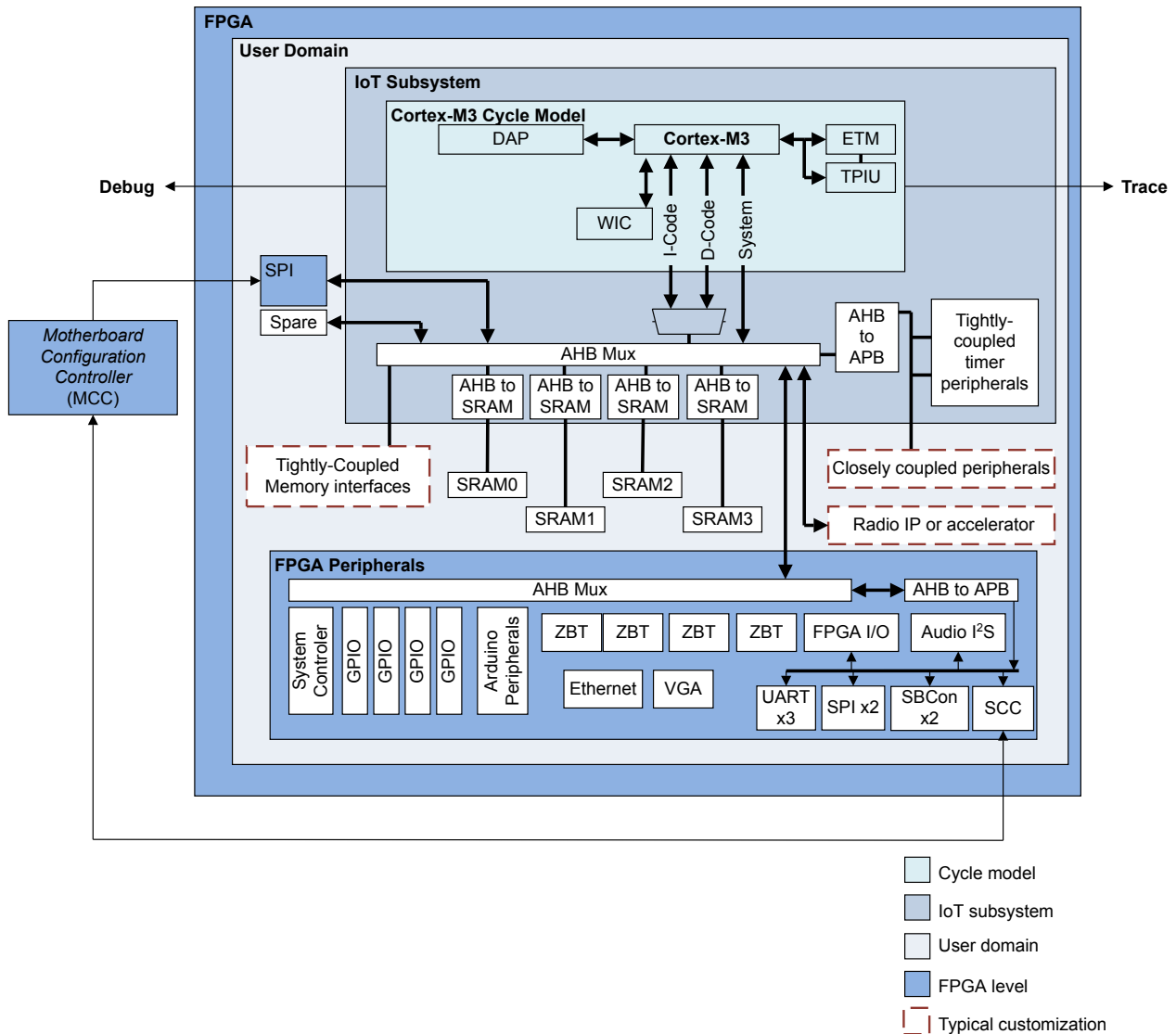


Figure 2-1 Cortex-M3 DesignStart Eval example system

This section contains the following subsections:

- [2.1.1 Tightly-coupled timer peripherals on page 2-18.](#)
- [2.1.2 Tightly-Coupled Memory interfaces on page 2-18.](#)
- [2.1.3 Primary code memory on page 2-18.](#)
- [2.1.4 Closely coupled peripherals on page 2-18.](#)
- [2.1.5 MPS2+ platform-specific peripherals on page 2-19.](#)
- [2.1.6 Integrating ARM radio IP or an accelerator on page 2-19.](#)

2.1.1 Tightly-coupled timer peripherals

There are two *Advanced Peripheral Bus* (APB) timer modules that are instantiated within the subsystem hierarchy in Cortex-M3 DesignStart Eval.

These APB timers are reserved for the operating system. ARM recommends that you do not modify these timers or their connections to the subsystem.

For more information on the subsystem in Cortex-M3 DesignStart Eval, see the *ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide*.

2.1.2 Tightly-Coupled Memory interfaces

There are four *Advanced High-performance Bus* (AHB) to SRAM interface modules that are instantiated within the subsystem hierarchy in the Cortex-M3 DesignStart Eval. This multi-bank memory configuration gives the fastest possible access.

The IoT subsystem can be configured to provide 1, 2, 3 or 4 banks of SRAM. When you use the Cortex-M3 DesignStart Eval, any configuration is required to be made manually. However, there is limited value in modifying these SRAM interfaces when targeting an FPGA platform.

2.1.3 Primary code memory

In the subsystem included with the Cortex-M3 DesignStart Eval, there is a dedicated AHB interface that provides access to flash storage. This is connected in `m3ds_user_partition` using an AHB to SRAM interface to an FPGA block RAM.

The block RAM is preloaded in simulation with the executable code. When using the MPS2+ platform, code can also be executed from the 64-bit ZBT SSRAM1. If your development is focused on peripherals, then you can choose not to modify this code memory architecture. However, for an eventual SoC implementation, it will be necessary to replace the block RAM with a suitable equivalent, either SRAM or a flash interface.

If you want to optimize the performance of the flash interface using a cache, or intend to interface to an external flash memory, then you should replace the example components with your own design. There are three APB master interfaces on the subsystem which are reserved for memory expansion:

- **APB[3]** is recommended to control a cache, if present.
- **APB[9]** is recommended to control a flash interface, if present.
- **APB[10]** is recommended to control a second bank of the flash interface, if present.

There are also two interrupts in the recommended mapping which a code memory interface can use. ARM recommends the following:

- **INTISR[32]** to indicate a memory system fault.
- **INTISR[33]** for the flash memory controller.

2.1.4 Closely coupled peripherals

The subsystem in Cortex-M3 DesignStart Eval provides APB master expansion ports which are used in the example system for UARTs, additional timers, Watchdog, *Real Time Clock* (RTC), and *True Random Number Generator* (TRNG).

The remaining five APB master expansion ports can be used for communications interfaces, or other peripherals with a low latency requirement. There are six reserved interrupt lines that are suitable for additional peripherals.

Peripherals can be connected to the APB expansion ports without any modifications to the interconnect within the subsystem. The normal behavior for an unused APB location is to return a read value of `0x00000000`, and the expansion ports are fully decoded in the memory map. For more information on the memory map, see the *ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide*.

The TRNG provides a minimum level of functionality that is required to support secure communication. If you require acceleration of encryption functions, then ARM TrustZone CryptoCell can be licensed separately.

2.1.5 MPS2+ platform-specific peripherals

There is a hierarchy of AHB and APB interconnect with an array of peripherals that are designed to match the MPS2+ platform resources and the Arduino shield expansion. This whole subsystem would typically be replaced when designing a custom device.

However, replacing a subset of the peripherals in this hierarchy is sufficient for development purposes. Since the replacement is design-specific, there are no recommendations that are relevant for compatibility.

The MPS2+ platform peripheral subsystem is connected to the 'background region' of the AHB expansion from the subsystem in DesignStart. This means that the peripherals added here are subjected to increased latency, but also provide a power benefit by reducing the interconnect activity when these peripherals are not in use.

Note

When you use the mbed Cloud client, your device requires a persistent storage device that the file system supports. The simplest implementation uses an microSD card, which can be connected using the SPI.

2.1.6 Integrating ARM radio IP or an accelerator

The subsystem included with Cortex-M3 DesignStart Eval is suitable for integration with the baseband part of the ARM Cordio® radio IP. If you license the ARM Cordio radio IP, you might want to contact ARM for details on the options for evaluation and prototyping.

The interface provided for the radio IP is also suitable for adding dedicated hardware accelerators that require an AHB memory-mapped control interface, and DMA access to the memory system.

One of the subsystem master expansion ports maps to a 64KB memory region that can be used to control the radio IP or accelerator.

The radio IP or accelerator can use one of the slave expansion ports on the subsystem to transfer data by DMA.

The ARM Cordio radio also requires some GPIO connections for control and monitoring, and several interrupt lines. If you are using the ARM Cordio radio IP, ARM recommends that you use interrupt lines 34-43.

It is also possible to interface to an external analogue radio although this would require many expansion pins and a different FPGA platform (the MPS2+ platform does not expose enough I/O pins).

Note

Cortex-M3 DesignStart Pro, which can be licensed for full production, is also suitable for integration with the baseband part of the ARM Cordio radio IP.

2.2 Adding peripherals to memory map

In the IoT subsystem included with the Cortex-M3 DesignStart Eval, both the APB expansion and the FPGA peripheral APB interfaces have unused ports that can be used to add peripherals, without any modifications. If these ports are not sufficient and an existing peripheral cannot be replaced, you are required to modify at least one of the AHB interconnects in the system.

Note

ARM recommends that:

- The subsystem is left unmodified.
- Only the interconnect within the `fpga_peripherals` hierarchy is replaced.

You can use *Cortex-M System Design Kit* (CMSDK) to generate an interconnect, but it is not possible to modify the subsystem. It is possible to generate a new subsystem using Cortex-M3 DesignStart Eval as an example, but you must validate any modifications to the subsystem.

Note

Although an APB master port can be terminated using tie offs, an AHB master port must be connected to a default slave component when it is unused.

2.3 Low-power optimizations

The IoT subsystem in Cortex-M3 DesignStart Eval supports low-power operation. However, the example system that is provided with Cortex-M3 DesignStart Eval is intended to provide a straightforward FPGA implementation flow and does not implement any low-power operation.

Note

Cortex-M3 DesignStart Pro, which can be licensed for full production, also supports low-power operation.

This section contains the following subsections:

- [2.3.1 Multiple clock frequencies on page 2-21.](#)
- [2.3.2 Latch-based WIC on page 2-21.](#)
- [2.3.3 Power gating on page 2-21.](#)

2.3.1 Multiple clock frequencies

A typical design implements support for different clock frequencies, coupled with the ability to gate clocks to most of the design when using the **WFI** and **WFE** instructions. Different peripheral subsystems might also use clocks that are appropriate to the performance requirements of each peripheral.

If you implement multiple clock frequencies, you might need to:

- Use synchronous or asynchronous bridges.
- Consider the synchronization of control signals such as interrupts.

Note

The CoreLink SSE-050 Subsystem provided with Cortex-M3 DesignStart Pro provides bus activity signals which can be used to help with managing the control of clocks for different parts of a design.

2.3.2 Latch-based WIC

The standard *Wake-up Interrupt Controller* (WIC) included with Cortex-M3 DesignStart Eval (and integrated within the flattened Cortex-M3 integration level of hierarchy) requires a free-running clock.

If you license the Cortex-M3 processor, you can choose to replace this design with a latch-based version that improves the standby power in the lowest power state. It is not possible to replace the WIC or observe its internal state when you use Cortex-M3 DesignStart Eval.

2.3.3 Power gating

In a more complex design, it can be beneficial to gate the power and clocks in inactive regions of the design. Power gating reduces leakage power at the expense of a more complex design.

If you implement power gating, then you are required to:

- Manage the sequencing of enable and disable cycles.
- Consider how to save and restore state in any logic which does not remain powered up.
- Ensure that the logic is properly idle before it is disabled.
- Determine the appropriate response on an interface that connects to a component which might be inactive.

2.4 Multiple processor designs

Cortex-M3 DesignStart Eval does not include support for designs with multiple processors.

However, it is possible to implement several Cortex-M3 processors in a single device by:

- Making significant changes to the subsystem to support:
 - Shared access to code.
 - Identity discovery for each processor.
- Updating the debug and trace infrastructure.

The CoreSight SoC deliverable can be licensed separately from ARM. It includes an alternative CORTEXM3INTEGRATIONCS level that has support for cross triggering, where the debug logic, *Embedded Trace Macrocell* (ETM) and *Cross Trigger Interface* (CTI) can interact with each other and with other CoreSight components. It also supports a full 32-pin *Trace Port Interface Unit* (TPIU), on-chip trace buffers, and trace capture into system memory with the appropriate IP.

If the Cortex-M3 processor does not match your application performance requirement, then you can consider licensing the full Cortex-M0 processor for lower power, or Cortex-M4 processor for more performance and DSP capabilities.

2.5 Migrating from Cortex®-M3 DesignStart™ Eval to Cortex®-M3 DesignStart™ Pro

ARM recommends that you perform this migration as early as possible in your development process.

The Cortex-M3 DesignStart Eval product is packaged to provide a working FPGA targeted system out of the box. Although it is closely based on the SSE-050 Subsystem, which is included as part of the Cortex-M3 DesignStart Eval product, the Cortex-M3 DesignStart Pro deliverables offer more flexibility, and do not specifically target the MPS2+ FPGA platform. This results in a smaller set of peripherals integrated in the 'platform-specific' group. In Cortex-M3 DesignStart Pro, the example integration uses a more simplified AHB peripheral subsystem connected to the **TARGEXP1** master port. This replaces the `mps2_peripherals_wrapper`, which includes all the FPGA peripherals provided with the Cortex-M3 DesignStart Eval system.

When you develop a system, you may initially choose to retain some peripherals specific to the MPS2+ FPGA platform. To do this, you need to plan to partition your extensions to the system so that they can be easily integrated into the final and simpler platform-specific extension peripheral subset. You can then branch from your Cortex-M3 DesignStart Eval system by using the working Cortex-M3 DesignStart Pro Execution Testbench and modifying that example system.

If you plan to build an FPGA view of your system developed from Cortex-M3 DesignStart Pro instead, then you should replace every module instantiated within the `m3ds_user_partition` with the equivalent modules from Cortex-M3 DesignStart Pro. When you do this, the closely coupled peripherals and the platform-specific extension peripherals are imported as two separate modules. You will need to rewire all the peripheral I/O to suit your new design.

When you do the migration, ARM recommends the following:

- Work in a disk area that is populated with Cortex-M3 DesignStart Pro deliverables.
- Copy only the minimum necessary Cortex-M3 DesignStart Eval components. Most of these components are in the `m3designstart/logical` and `m3designstart/fpga` directories.

Chapter 3

Using ARM Socrates™ to update the system

This chapter describes how to use Socrates™ to update the system.

It contains the following section:

- [3.1 ARM Socrates™ tools on page 3-25.](#)

3.1 ARM Socrates™ tools

The ARM Socrates *Design Environment* (DE) is a software tool for SoC integration. The tool standardizes, configures, and intelligently integrates IP with ARM IP to create a SoC.

Socrates uses an IP-XACT design environment to standardize IP in the IP-XACT standard.

An IP-XACT view of subsystem in Cortex-M3 DesignStart Eval is also included with the deliverables.

Contact ARM if you are interested in using the Socrates DE to develop a system that is derived from the example system in Cortex-M3 DesignStart Eval.

Chapter 4

Validation and ASIC implementation

This chapter describes the validation necessary for design changes, and the factors you must consider for ASIC implementation.

It contains the following sections:

- [4.1 Validation of design changes](#) on page 4-27.
- [4.2 Key points for ASIC implementation](#) on page 4-28.

4.1 Validation of design changes

Any changes with the integration within the example system in the Cortex-M3 DesignStart Eval RTL require validation.

You can use the integration tests included with Cortex-M3 DesignStart Eval as an example of how to validate the integration in a structured way. For more information on integration tests, see the *ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide*.

Typically, a full stand-alone validation would have been performed on the peripherals that you integrate into a design. The focus at system level should be in testing the integration of all interfaces in all major operating modes, and demonstrating system-level functionality.

Any components that operate mainly through providing a system-level function, such as the *Wake-up Interrupt Controller* (WIC) and clock control logic, should be exercised in as many different system states as deemed practical.

If the components have specific throughput or latency requirements, you must demonstrate that they inter-operate as expected, and the worst-case performance is acceptable. For example, if the FPGA peripheral subsystem connects to the main subsystem AHB interconnect using an AHB matrix, it is important to demonstrate that they behave as expected. When only one peripheral is accessed at a time, usually from the processor, only a few cycles of latency is added. However, if there are also accesses coming from the subsystem AHB slave ports that are accessing different peripherals on the same matrix, then there will be additional stalls due to the port contention.

4.2 Key points for ASIC implementation

Cortex-M3 DesignStart Eval provides access to an RTL simulation and FPGA implementation flow only.

If you plan to take a design to an ASIC implementation, there are several factors to consider, including:

- Cortex-M3 DesignStart Eval is not representative of the full CoreLink SSE-050 and Cortex-M3 processor.
- Moving to an ASIC flow requires additional IPs, which include power management, oscillators, memories, and the standard cells for the particular target process.
- A complete SoC requires the full power and reset sequence, which involves several phases as power rails and oscillators become stable.
- The FPGA flow requires timing closure, but does not target an aggressive implementation result. When a full clock and reset control structure is in place, the process to achieve timing closure with satisfactory performance is likely to be more complex.

Chapter 5

Customizations in the FPGA Evaluation Flow

This chapter describes the typical customizations and the customization issues that might occur in the FPGA Evaluation Flow for Cortex-M3 DesignStart Eval.

It contains the following sections:

- [5.1 FPGA flow on page 5-30.](#)
- [5.2 Increasing the processor clock speed on page 5-31.](#)
- [5.3 Porting to a different platform on page 5-32.](#)

5.1 FPGA flow

The FPGA flow in Cortex-M3 DesignStart Eval does not rely on any particular FPGA or FPGA vendor features. Therefore, you can easily add functional blocks to the design, within the constraints of the MPS2+ FPGA platform (in particular, the 52 pins available for I/O expansion).

ARM does not support any modifications to the *Motherboard Configuration Controller* (MCC) firmware. The MCC preloads a code image using the SPI interface, configures the *Real Time Clock* (RTC), and accesses the SCC using the serial interface.

ARM recommends the following:

- Keep all existing peripherals. The existing peripherals are relevant to the correct operation of the processor and the MPS2+ platform peripherals.
- Do not change the PLL or clocking structure without careful consideration.
- Do not change the FPGA I/O because it is matched to the peripherals on the MPS2+ platform. If you do not need to connect to the Arduino shield and have your own interface board that connects to the Expansion Port pins, then you can change the function of the Expansion Port I/O pins.
- Retain the RTC and SCC peripherals. The SPI code download interface can be modified if you provide an alternative mechanism for code storage (for example, external flash).

The implementation of Cortex-M3 DesignStart Eval on the MPS2+ platform only uses approximately 20% to 30% of the FPGA resources.

If you modify the FPGA design and include more peripheral logic, then the routing congestion increases. Depending on the specific design, if your FPGA resource utilization is approximately 90%, then it might start to become problematic. This can be offset by removing any unnecessary peripherals from the design.

The example system in Cortex-M3 DesignStart Eval does not include any clock domain bridges on the AHB, although each level of interconnect introduces a register slice. In a more complex design, it might be necessary to reduce the clock speed for some peripherals, or add further register slices.

Although the MPS2+ platform includes pins for expansion connectors, these pins are not designed for routing high-speed signals, which would be required for the expansion of the AHB interconnect. This might limit the expansion capabilities of the platform.

You are required to add any additional peripherals to the search path defined in the following file:

```
<install_directory>/m3designstart/fpga/AN511_SMM_CM3DS/synthesis/  
AN511_SMM_CM3DS.qsf
```

The Verilog structures used in the design for the FPGA should match the simulation environment, except for a few FPGA primitives used for IP pins and clocking.

5.2 Increasing the processor clock speed

You can increase the Cortex-M3 processor clock (**SYSCLK**) speed to meet your application requirements.

If you increase **SYSCLK**, then you are required to modify the following timing constraints file to reflect the new clock period constraints:

```
<install_directory>/m3designstart/fpga/AN511_SMM_CM3DS/synthesis/  
AN511_SMM_CM3DS.sdc
```

For more information on how to increase the core clock speed, see the *ARM® Cortex®-M3 DesignStart™ Eval FPGA User Guide*.

There are possible consequences to increasing the processor clock speed, such as meeting timing requirements. If it is not possible to meet timing at the new speed, then it is necessary to place register slices in the failing paths. However, these slices increase latency. The trade-off between clock speed and latency should be considered for optimum system performance.

5.3 Porting to a different platform

Cortex-M3 DesignStart Eval has limited use of specific FPGA primitives. Therefore it is possible to port the Cortex-M3 DesignStart Eval design to other Intel-based FPGA platforms, and also to FPGA platforms from other vendors.

If you decide to port the design to another platform, then there are several factors to consider, including:

- [Clocks on page 5-32.](#)
- [Code memory on page 5-32.](#)
- [Real Time Clock on page 5-32.](#)
- [Debug on page 5-33.](#)

5.3.1 FPGA platform clocks

Cortex-M3 DesignStart Eval directly instantiates three Altera Cyclone V PLLs, which convert the input board clocks to the specific clocks needed.

For details of the input source clocks and the derived clocks from the PLLs, see the *ARM® Cortex®-M3 DesignStart™ Eval FPGA User Guide*.

The PLL instantiation is contained within the following file:

```
<install_directory>/smm/logical/smm_common_fpga/verilog/fpga_pll_speed.v
```

It is possible that the source clocks on a new platform do not match the source clocks on the MPS2+ platform. In this case, it is necessary to ensure that the newly selected PLLs are able to generate the required derived clock frequencies from the new platform's source clocks. The minimum requirements are as follows:

- **PLL[0]** is required to generate **SYSCCLK** and the debug, SPI, and I²C clocks.
- **PLL[1]** is required if there are audio peripherals.
- **PLL[2]** is not currently used and so can be removed.

5.3.2 Code memory

The Cortex-M3 processor begins **code execution from address 0x00000000, which corresponds to the TARGETFLASH0 block RAM memory within the FPGA.**

The *Motherboard Configuration Controller* (MCC) in the MPS2+ platform loads the code into the block RAM using SPI during the start-up sequence. If you are porting your design to a new platform, then either have a mechanism whereby a similar method can be used, or have the FPGA compiled with the appropriate block RAM initialized with the code contents.

Possible mechanisms for loading the code include:

- Initializing the block RAM with a boot loader that **redirects code execution** to an external flash memory.
- **Remapping address 0x00000000** to an external flash device on the new platform.
- Initializing the block RAM using a debugger every time the platform is initialized.

5.3.3 Real Time Clock

Cortex-M3 DesignStart Eval includes a *Real Time Clock* (RTC) module. This is necessary for the correct operation of the mbed operating system.

The RTC value is retained on the MPS2+ platform during power down by an on-board battery and the *Motherboard Configuration Controller* (MCC).

During the start-up sequence, the MCC initializes the FPGA RTC module with the correct time.

If an RTC is required on the new platform, then a mechanism to initialize the FPGA RTC value after initial configuration is required.

5.3.4 Debug

Cortex-M3 DesignStart Eval has multiple methods of connection for debugging the operation of the Cortex-M3 processor and the FPGA.

ARM recommends that JTAG or SWD is made available, so that there is access to the processor debug functionality.

JTAG can be connected to the external pins, or connected as an extension of the FPGA debug infrastructure.

If you export the SWD interface from the FPGA, you can use any of the following:

- **Standard CoreSight debug connectors.**
- CMSIS-DAP adapter to interface to a USB port. For example, the DIPDAP-LPC11U35 debug probe from mbed. For more information on the debug probe, see <https://developer.mbed.org/dipdap>.

Appendix A

Revisions

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

It contains the following section:

- [*A.1 Revisions - Cortex®-M3 DesignStart™ Eval on page Appx-A-35.*](#)

A.1 Revisions - Cortex®-M3 DesignStart™ Eval

This section describes the technical changes between released issues of this document.

Table A-1 Issue 00

Change	Location	Affects
First release	-	-