

ARM® Cortex®-M3 DesignStart™ Eval

Revision: r0p0

FPGA User Guide



ARM® Cortex®-M3 DesignStart™ Eval

FPGA User 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 FPGA User Guide

Preface

<i>About this book</i>	7
<i>Feedback</i>	10

Chapter 1

Introduction

1.1	<i>About Cortex®-M3 DesignStart™ Eval</i>	1-12
1.2	<i>About the ARM Versatile Express Cortex-M Prototyping System (V2M-MPS2+) . .</i>	1-14
1.3	<i>Using the documentation</i>	1-15
1.4	<i>FPGA Evaluation Flow directory structure</i>	1-17
1.5	<i>Limitations</i>	1-18

Chapter 2

Using the prebuilt FPGA image

2.1	<i>Setting up the MPS2+ FPGA platform</i>	2-20
2.2	<i>Running the self-test program</i>	2-21
2.3	<i>Connecting to a debugger</i>	2-23

Chapter 3

FPGA platform overview

3.1	<i>System overview</i>	3-25
3.2	<i>Memory map</i>	3-26
3.3	<i>Block RAM instances</i>	3-27
3.4	<i>External Zero Bus Turnaround SSRAM</i>	3-28
3.5	<i>External PSRAM</i>	3-29

3.6	Arduino adapter board	3-30
3.7	Embedded Trace Macrocell interface	3-31
3.8	CMSDK APB subsystem	3-32
3.9	AHB GPIO	3-33
3.10	Serial Peripheral Interface	3-34
3.11	Color LCD parallel interface	3-35
3.12	Ethernet	3-36
3.13	VGA	3-37
3.14	Audio I ² S	3-38
3.15	Audio configuration	3-40
3.16	FPGA system control and I/O	3-41
Chapter 4	Clocks	
4.1	Source clocks	4-43
4.2	Derived clocks	4-44
Chapter 5	Serial Communication Controller	
5.1	SCC interface overview	5-46
5.2	SCC memory map	5-47
Chapter 6	FPGA build	
6.1	Build flow	6-50
6.2	Build requirements	6-52
Chapter 7	Integrating with mbed™ OS	
7.1	Compatibility with mbed™ OS	7-54
Chapter 8	Performance and utilization	
8.1	Performance and clocks	8-56
8.2	Utilization of default system	8-57
Appendix A	Revisions	
A.1	Revisions - Cortex®-M3 DesignStart™ Eval	Appx-A-59

Preface

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

It contains the following:

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

About this book

This book describes how to use the FPGA platform in the ARM *Versatile™ Express Cortex®-M Prototyping System* to evaluate a design developed using 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, software engineers, system integrators, and system designers, who might not have previous experience of ARM products, but want to run a complete example of a working system.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This chapter introduces Cortex-M3 DesignStart Eval and gives an overview of the FPGA Evaluation Flow, its directory structure, and limitations.

Chapter 2 Using the prebuilt FPGA image

Cortex-M3 DesignStart Eval includes a prebuilt FPGA image file of the Cortex-M3 DesignStart Eval example system. This chapter describes how to set up the MPS2+ platform to load the prebuilt file and run a self-test program.

Chapter 3 FPGA platform overview

This section gives an overview of the FPGA components that are used in Cortex-M3 DesignStart Eval.

Chapter 4 Clocks

This chapter describes the source and derived clocks for the FPGA design.

Chapter 5 Serial Communication Controller

This chapter describes the *Serial Communication Controller* (SCC) used in the Cortex-M3 DesignStart Eval FPGA image.

Chapter 6 FPGA build

This chapter describes the steps that are required to build an FPGA bit file from the supplied source code.

Chapter 7 Integrating with mbed™ OS

This chapter describes the support available for integrating the FPGA system with mbed OS.

Chapter 8 Performance and utilization

This chapter describes the performance, resources, and utilization for the default system of the FPGA design in 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.

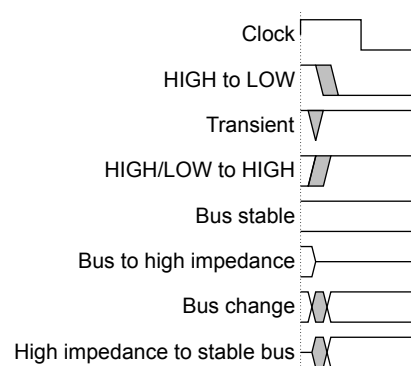


Figure 1 Key to timing diagram conventions

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 RTL and FPGA Quick Start Guide* (ARM 100895).
 - *ARM® Cortex®-M3 DesignStart™ Eval Customization Guide* (ARM 100897).
- 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 FPGA User Guide*.
- The number ARM 100896_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 and gives an overview of the FPGA Evaluation Flow, its directory structure, and limitations.

It contains the following sections:

- *1.1 About Cortex®-M3 DesignStart™ Eval* on page 1-12.
- *1.2 About the ARM Versatile Express Cortex-M Prototyping System (V2M-MPS2+)* on page 1-14.
- *1.3 Using the documentation* on page 1-15.
- *1.4 FPGA Evaluation Flow directory structure* on page 1-17.
- *1.5 Limitations* on page 1-18.

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 About the ARM Versatile Express Cortex-M Prototyping System (V2M-MPS2+)

The MPS2+ platform is a small FPGA development board that contains:

- A *Motherboard Configuration Controller* (MCC).
- An Altera Cyclone V FPGA.
- Various memories and debug connectors.
- A color LCD touch screen.
- Connectors for Ethernet, VGA, Audio, and serial interfaces.
- User switches and LEDs.
- An SPI to microSD card adapter.

You can program the FPGA with the FPGA image built using ARM Cortex-M3 DesignStart Eval.

The MPS2+ platform enables hardware and software developers to rapidly design and test hardware and software components as part of a Cortex-M3 processor ecosystem.

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

This section contains the following subsection:

- [1.2.1 Decryption Key on page 1-14](#).

1.2.1 Decryption Key

ARM supplies the MPS2+ platform with a decryption key that is programmed into the FPGA.

The decryption key is required to enable loading of the prebuilt images, which are encrypted.

User images do not require the decryption key.

Note

A battery supplies power to the key storage area of the FPGA. Any keys stored in the FPGA are lost when battery power is lost. If battery power is lost, you must return the board to ARM for reprogramming of the key.

1.3 Using the documentation

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

Scope of this document

The *ARM® Cortex®-M3 DesignStart™ Eval FPGA User Guide* describes how to build an FPGA image and evaluate software running on the *Versatile Express Cortex-M Prototyping System (V2M-MPS2+)* platform.

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 information required for system design and RTL simulation. ————— Note ————— This is the main document for Cortex-M3 DesignStart Eval. —————
<i>ARM® Cortex®-M3 DesignStart™ Eval RTL and FPGA Quick Start Guide</i>	Describes how to run basic tests using an RTL simulator and an 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. —————
<i>ARM® Cortex®-M3 DesignStart™ Eval Customization Guide</i>	Describes the high-level steps to integrate your own peripherals, and make other modifications to the Cortex-M3 DesignStart Eval system.

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+ 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*.

1.4 FPGA Evaluation Flow directory structure

The following diagram and table describe the main directories of the Cortex-M3 DesignStart Eval FPGA Evaluation Flow:

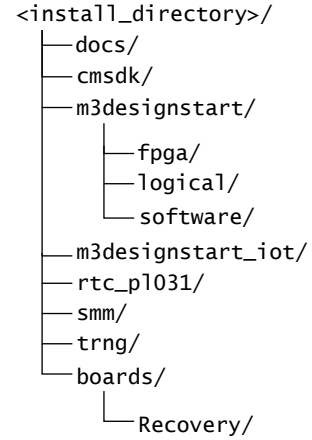


Figure 1-1 FPGA Evaluation Flow main directories

Table 1-2 Directory descriptions

Directory	Description
docs/	Contains documentation for Cortex-M3 DesignStart Eval.
cmsdk/	Contains the RTL for: <ul style="list-style-type: none"> ARM <i>Cortex-M System Design Kit</i> (CMSDK) components. Some CMSDK components are used in the example system in Cortex-M3 DesignStart Eval.
m3designstart/	Contains the following: <ul style="list-style-type: none"> Scripts for building an FPGA image. Example DesignStart system. Testbench in <code>testbench/execution_tb/</code>. Integration tests in <code>testbench/testcodes/</code>. ARM <i>Cortex Microcontroller Software Interface Standard</i> (CMSIS) support files for the Cortex-M3 DesignStart Eval.
m3designstart_iot/	Cortex-M3 DesignStart Eval version of ARM CoreLink SSE-050 subsystem.
rtc_pl031/	Real Time Clock peripheral.
smm/	Peripherals and support code for the MPS2+ FPGA platform.
trng/	Stand-alone True Random Number Generator.
boards/Recovery/	Contains the files required to be loaded onto the microSD card of the MPS2+ platform, in order to program and run the prebuilt FPGA image and software.

1.5 Limitations

This section describes the limitations of the Cortex-M3 DesignStart Eval FPGA Evaluation Flow.

You should not use the processor technology or the supporting deliverables as an indicator of what is received under a full technology license of the ARM Cortex-M3 processor.

This section contains the following subsections:

- [1.5.1 Deliverables on page 1-18.](#)
- [1.5.2 Processor support on page 1-18.](#)

1.5.1 Deliverables

Cortex-M3 DesignStart Eval does not contain the EDA tools used for simulation or compilation.

You must obtain the software tools separately.

The following table shows the supported software tools for Cortex-M3 DesignStart Eval:

Table 1-3 Software tools

Tool	Supported software
Compile test code	The recommended minimum versions are: <ul style="list-style-type: none">• ARM Keil <i>Microcontroller Development Kit</i> (MDK) version 5.22.• ARM <i>Development Studio 5</i> (DS-5) version 5.06.409.• GNU <i>Tools for ARM Embedded Processors</i> (ARM GCC) version 5-2016q2.
Compile RTL and FPGA build software	The recommended minimum version is: <ul style="list-style-type: none">• Intel Quartus version 16.1. <p style="text-align: center;">————— Note —————</p> <p>The Cyclone device on the ARM MPS2+ platform is also supported by the free Lite Edition of Quartus Prime. No Quartus license is required to build the Cortex-M3 DesignStart Eval FPGA Evaluation Flow.</p>

For more information on how to compile and simulate the RTL, see the *ARM® Cortex®-M3 DesignStart™ Eval RTL and Testbench User Guide*.

1.5.2 Processor support

The FPGA Evaluation Flow supports the Cortex-M3 processor from Cortex-M3 DesignStart Eval.

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

Chapter 2

Using the prebuilt FPGA image

Cortex-M3 DesignStart Eval includes a prebuilt FPGA image file of the Cortex-M3 DesignStart Eval example system. This chapter describes how to set up the MPS2+ platform to load the prebuilt file and run a self-test program.

It contains the following sections:

- [2.1 Setting up the MPS2+ FPGA platform on page 2-20.](#)
- [2.2 Running the self-test program on page 2-21.](#)
- [2.3 Connecting to a debugger on page 2-23.](#)

2.1 Setting up the MPS2+ FPGA platform

To set up the MPS2+ platform with the provided prebuilt FPGA image file of the Cortex-M3 DesignStart Eval example system, follow these steps:

1. Connect a USB lead from your computer to the USB-B connector on the MPS2+ platform.
2. Connect the 12V power adapter to the power input connector on the MPS2+ platform. Your computer should recognize the MPS2+ platform as an external USB drive, named V2M_MPS2.
3. Load an Application Note by locating the `boards/Recovery` directory in the Cortex-M3 DesignStart Eval bundle. Copy the following files to the MPS2+ platform directory, which has a similar directory structure:

`/MB/HBI0263C/AN511/`

Copy the complete directory.

`/MB/HBI0263C/board.txt`

Copy the following two lines in the `board.txt` file:

```
[MCCS]
MBBIOS: mbb_v221.ebf          ; MB BIOS IMAGE. Supports RTC with time updates via
MCC
APPFILE: AN511\an511_v1.txt   ; - Cortex-M3 DesignStart
```

————— **Note** —————

Ensure that the `APPFILE:an511_v1.txt` line is uncommented, and that all other `APPFILE` lines are commented. Only one `APPFILE` line may be enabled.

`/MB/HBI0263C/mbb_v221.ebf`

Copy the file.

`/SOFTWARE/iot_test.axf`

Copy the file.

`/config.txt`

Copy the file.

————— **Note** —————

In this file, `RTC = TRUE`. This setting is different from other Application Notes. The *Real Time Clock* (RTC) is enabled because Cortex-M3 DesignStart Eval supports RTC, which is updated at boot-up by the *Motherboard Configuration Controller* (MCC).

If you are starting with a blank microSD card in the MPS2+ motherboard card slot, you can copy the entire contents of the `boards/Recovery` directory (not including the `Recovery` part of the path) into the root of the microSD card.

4. Power up the MPS2+ platform.

When the platform is first powered up with a new BIOS file `mbb_v221.ebf`, the BIOS file is copied internally. This is indicated by the rapid flashing of **LED[0]**. After the new BIOS has been successfully copied, the platform resumes with its standard FPGA loading process, which is indicated by a count sequence on **LED[7:0]**. After the FPGA is fully loaded, then the color LCD screen displays a self-test splash screen.

If the MPS2+ platform does not boot correctly, then refer to the `log.txt` in the root directory of the MPS2+ platform, which provides a log of the files loaded at bootup.

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

2.2 Running the self-test program

The MPS2+ FPGA platform uses a self-test program to test its main components. This self-test program is specific to the prebuilt image, which is based on the Cortex-M3 DesignStart Eval example system.

If you extend the example system, then you may need to modify the self-test program.

If you need to modify and recompile the self-test image, you are required to install legacy support in Keil® MDK. This provides the low-level interface which is used by the self-test. Source code for the self-test is located at `m3designstart/selftest/Build_Keil/` in the deliverables.

To run a self-test program, follow these steps:

1. Connect an RS232 serial lead to the General UART connector (FPGA), on the MPS2+ platform.
2. Configure the UART according to the following settings:

Table 2-1 UART configuration

Configuration	Value
Baud rate	38 400
Data bits	8
Stop bits	1
Parity	None
Flow control	XON/XOFF protocol

3. Select the self-test program that you want to run based on the following options available that are displayed by the UART console window when the MPS2+ platform has booted up:

```
Versatile Express Cortex-M Prototyping System (V2M-MPS2) Test Suite
Version 1.0.0 Build date: Mar  8 2017
Copyright (C) ARM Ltd 2015. All rights reserved.

V2M-MPS2 revision C

Application Note AN511, FPGA build 1

CPU: Cortex-M3 r2p1

Summary of results
=====
1 AACI (Audio)           : Not Run
2 CLCD (Video)           : Not Run
3 TSC (touchscreen)      : Not Run
4 LEDs/Switches/Buttons  : Not Run
5 SSP (eeprom)           : Not Run
6 Ethernet               : Not Run
7 Memory                 : Not Run
8 Timer                  : Not Run
9 RTC                    : Not Run

Select the test you wish to run. (X - Exit)

Choice:
```

————— Note —————

To run Test 1, AACI (Audio), you must connect a 3.5mm stereo jack lead from the audio output connector to the audio input connector.

When the test runs, the console window displays the self-test status:

```
Versatile Express Cortex-M Prototyping System (V2M-MPS2) Test Suite
Version 1.0.0 Build date: Mar  8 2017
Copyright (C) ARM Ltd 2015. All rights reserved.

V2M-MPS2 revision C
```

Application Note AN511, FPGA build 1

CPU: Cortex-M3 r2p1

Summary of results

=====

1 AACI (Audio)	: PASS
2 CLCD (Video)	: PASS
3 TSC (touchscreen)	: PASS
4 LEDs/Switches/Buttons	: PASS
5 SSP (eeprom)	: PASS
6 Ethernet	: PASS
7 Memory	: PASS
8 Timer	: PASS
9 RTC	: PASS

Select the test you wish to run. (X - Exit)

Choice:

You can select 'X' to exit the self-test program.

2.3 Connecting to a debugger

The MPS2+ platform supports debug over USB, using CMSIS-DAP. This is a standard interface that is supported by most debug tools.

For details on connecting and configuring specific debug toolchains, see the *ARM® Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual*.

Chapter 3

FPGA platform overview

This section gives an overview of the FPGA components that are used in Cortex-M3 DesignStart Eval.

It contains the following sections:

- [3.1 System overview](#) on page 3-25.
- [3.2 Memory map](#) on page 3-26.
- [3.3 Block RAM instances](#) on page 3-27.
- [3.4 External Zero Bus Turnaround SSRAM](#) on page 3-28.
- [3.5 External PSRAM](#) on page 3-29.
- [3.6 Arduino adapter board](#) on page 3-30.
- [3.7 Embedded Trace Macrocell interface](#) on page 3-31.
- [3.8 CMSDK APB subsystem](#) on page 3-32.
- [3.9 AHB GPIO](#) on page 3-33.
- [3.10 Serial Peripheral Interface](#) on page 3-34.
- [3.11 Color LCD parallel interface](#) on page 3-35.
- [3.12 Ethernet](#) on page 3-36.
- [3.13 VGA](#) on page 3-37.
- [3.14 Audio I²S](#) on page 3-38.
- [3.15 Audio configuration](#) on page 3-40.
- [3.16 FPGA system control and I/O](#) on page 3-41.

3.1 System overview

The *Soft Macro Model* (SMM) is an FPGA implementation of an ARM processor or subsystem. The SMM is based on the SSE-050 Subsystem. Extra peripherals that are required by the FPGA are placed on the expansion of APB and AHB ports.

The following figure shows the block diagram of the FPGA design, indicating the Cortex-M3 processor from DesignStart, the CMSDK components (peripherals and interconnect), and the interfaces to the MPS2+ platform peripherals.

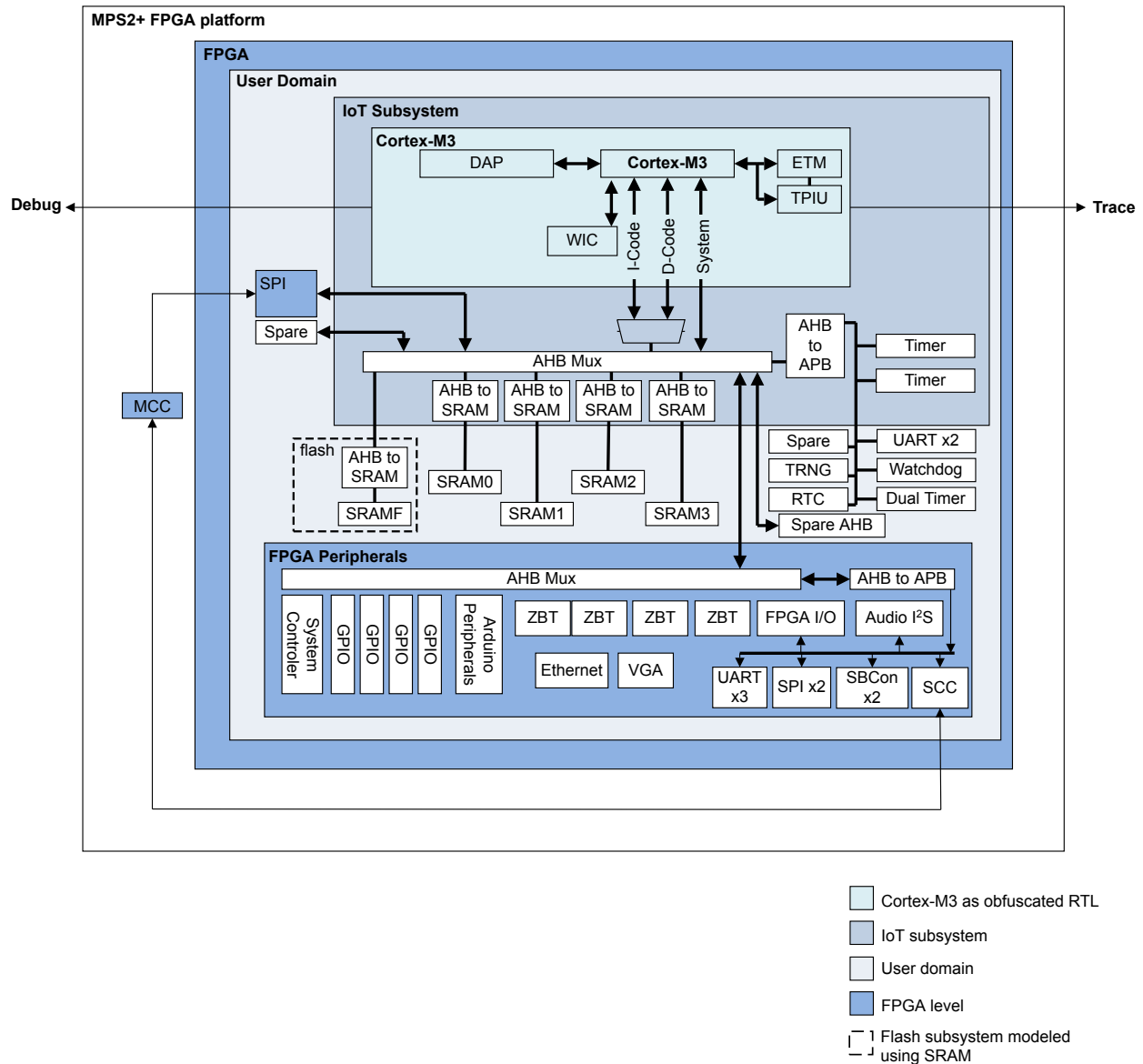


Figure 3-1 System overview

Note

This block diagram shows the functional hierarchy of the FPGA design.

3.2 Memory map

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

3.3 Block RAM instances

The *Soft Macro Model* (SMM) implements 256KB of FPGA internal block RAM as 32-bit AHB SRAM, as an emulation of flash. The boot code must be placed in this location. If your FPGA image file is large, you can transfer execution to the code stored in the ZBT SSRAM1.

The SMM implements four regions of 32KB internal block RAM as individual 32-bit AHB SRAM, intended for use as program RAM.

3.4 External Zero Bus Turnaround SSRAM

This section describes the *Zero Bus Turnaround* (ZBT) SSRAM in the FPGA platform.

This section contains the following subsections:

- [3.4.1 ZBT SSRAM1 on page 3-28.](#)
- [3.4.2 ZBT SSRAM2 and ZBT SSRAM3 on page 3-28.](#)

3.4.1 ZBT SSRAM1

This section describes the *Zero Bus Turnaround* (ZBT) SSRAM in the code region, which is an additional area in memory which can be used to store executable code.

This interface consists of two external 32-bit ZBT SSRAMs in parallel, forming a 64-bit ZBT SSRAM. This makes 4MB available (each ZBT SSRAM is 2MB).

This ZBT SSRAM1 is connected through the AHB.

3.4.2 ZBT SSRAM2 and ZBT SSRAM3

ZBT SSRAM2 and ZBT SSRAM3 are two external 32-bit ZBT SSRAMs that are connected to two independent ZBT interfaces.

Each ZBT SSRAM occupies 2MB of memory space. When combined, the ZBT SSRAMs occupy 4MB of memory space. This provides an additional RAM region. The addresses of the two ZBT SSRAMs are interleaved as shown in the following diagram:

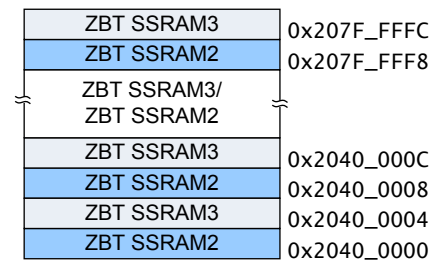


Figure 3-2 32-bit ZBT SSRAM memory space

The ZBT SSRAM2 and ZBT SSRAM3 are connected through the AHB.

3.5 External PSRAM

An external PSRAM is available in the FPGA platform.

A 16MB 16-bit PSRAM area is available and the memory map allocates the address-range 0x21000000 - 0x21FFFFFF. This PSRAM space enables large test programs to be used in the SRAM region of the Cortex-M3 memory space.

Note

The internal bus structure is such that the accesses to the PSRAM and ZBT SSRAM take longer than the accesses to the internal block RAM used for flash memory and SRAM regions. Therefore, any program using the PSRAM for code or data accesses suffers a corresponding performance degradation.

3.6 Arduino adapter board

The MPS2+ platform supports Arduino shields by using the ARM adapter for Arduino board. This is an expansion board which plugs into the GPIO connectors on the MPS2+ platform and allows you to connect up to two Arduino shields.

If you want to use the adapter, you can purchase this from ARM.

For more details on how to use the adapter, see the *Application Note AN502 Adapter for Arduino for the Cortex-M Prototyping System (MPS2 and MPS2+)*.

3.7 Embedded Trace Macrocell interface

If you want to use the MPS2+ platform to capture *Embedded Trace Macrocell* (ETM) trace, then you are required to use a suitable debug adapter, such as DSTREAM or ULINK*pro*. This can be connected using the 20-pin debug connector.

Serial Wire trace (instrumentation trace only) can be captured using a DSTREAM, ULINK*pro*-D, ULINK*plus*, or ULINK2 debug adapter.

3.8 CMSDK APB subsystem

The *Soft Macro Model* (SMM) uses a CMSDK-based APB subsystem for timers, UARTs, watchdog, and APB expansion ports.

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

3.9 AHB GPIO

The *Soft Macro Model* (SMM) uses six instances of the CMSDK GPIO.

GPIO0 through GPIO3 map to the expansion ports.

GPIO4 and GPIO5 have no external connections.

Where the GPIO expansion is shared with other peripherals, the pin muxing is controlled using the GPIO alternate function registers. Setting the alternate function bit to '1' disables the GPIO connection for the pin and enable the other peripheral connection.

For more information on the CMSDK GPIO registers, see the *ARM® Cortex®-M System Design Kit Technical Reference Manual*.

3.10 Serial Peripheral Interface

The *Soft Macro Model* (SMM) implements five PL022 *Serial Peripheral Interface* (SPI) modules:

- One general-purpose SPI module that connects to the general-purpose SPI connector, J21. An adapter board is available from ARM to mount a microSD card on this connector.
- One color LCD touch screen module control.
- One SPI interface for an ADC on the Arduino shield expansion adapter board.
- Two SPI interfaces for the Arduino shield expansion.

All these SPI interfaces are configured for master mode only.

If you are using the mbed OS, then it is required to connect the microSD card over SPI to provide the file system for the target. For more information on a supported adapter, see the *Application Note AN531 uSDCARD SPI Adapter for the Cortex-M Prototyping System (MPS2+)*.

To purchase the adapter, visit www.arm.com/mps.

3.11 Color LCD parallel interface

The color LCD module has two interfaces:

- SPI for LCD module that is used for sending image data to the LCD.
- I²C to transfer data input from the touch screen.

These interfaces are connected to a STMicroelectronics STMPE811QTR Port Expander with Advanced Touch Screen Controller on the KEIL MCBSTM32C display board. This display board contains an Ampire AM-240320LG 2.4" Touch Panel. Schematics for this board are listed in the *ARM® Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2 and V2M-MPS2+) Technical Reference Manual*.

Self-test that is provided with the MPS2+ platform includes example code for both of these interfaces.

3.12 Ethernet

The FPGA design connects SMSC LAN9220 through the AHB to the external memory block.

The *Soft Macro Model* (SMM) self-test code includes an example code for a simple loopback operation.

The Ethernet interface is supported in mbed.

Related references

[2.2 Running the self-test program on page 2-21.](#)

3.13 VGA

The following table shows the memory map for controlling a screen using the VGA interface:

Table 3-1 VGA memory map

Address	Description
0x41000000 - 0x4100FFFF	Writes to the current location of the cursor.
0x41100000 - 0x4110FFFF	512x128 image area at the top right of the screen. 0x41100000 is the top left of the image area and 0x4110FFFF is the bottom right. Pixels are mapped into memory from the base address using an offset address of {YYYYYYXXXXXXXX,0b00}, where X and Y are the horizontal and vertical pixel offset respectively.

For the image data, each pixel requires one 32-bit word, therefore, a total of 256KB are needed. The values in the data buffer are packed as 4 bits per channel in the format 0x00000RGB.

The pixel in the top left corner of the display occupies address 0x41100000 with each successive row using an offset of 0x00000400 from the previous row. For example: the *Left-Most Pixel* (LMP) of the second row is at 0x41100400 and the LMP of the third row is at 0x41100800.

3.14 Audio I²S

A simple FIFO interface generates and receives I²S audio.

The following table describes the memory map for I²S audio registers:

Table 3-2 Audio I²S memory map

Address	Name	Description
0x40024000	CONTROL	Control register [31:18] Reserved [17] Audio codec reset control (output pin) [16] FIFO reset [15] Reserved [14:12] Rx Buffer IRQ Water Level - Default 2 (IRQ triggers when less than two word space is available). [11] Reserved [10:8] TX Buffer IRQ Water Level - Default 2 (IRQ triggers when more than two word space is available). [7:4] Reserved [3] Rx Interrupt Enable [2] Rx Enable [1] Tx Interrupt Enable [0] Tx Enable
0x40024004	STATUS	Status register [31:6] Reserved [5] Rx Buffer Full [4] Rx Buffer Empty [3] Tx Buffer Full [2] Tx Buffer Empty [1] Rx Buffer Alert (Depends on Water level) [0] Tx Buffer Alert (Depends on Water level)
0x40024008	ERROR	Error status register [31:2] Reserved [1] Rx overrun. Set this bit to clear. [0] Tx overrun or underrun. Set this bit to clear.
0x4002400C	DIVIDE	Divide ratio register (for left or right clock) [31:10] Reserved [9:0] LRDIV (Left/Right). The default value is 0x80. $12.288\text{MHz} / 48\text{kHz} / 2*(L+R) = 128$.

Table 3-2 Audio I²S memory map (continued)

Address	Name	Description
0x40024010	TXBUF	Transmit Buffer FIFO Data Register. This is a write-only register. [31:16] Left channel [15:0] Right channel
0x40024014	RXBUF	Receive Buffer FIFO Data Register. This is a read-only register. [31:16] Left channel [15:0] Right channel
0x40024018 - 0x400242FC	RESERVED	-
0x40024300	ITCR	Integration Test Control Register [31:1] Reserved [0] ITCR
0x40024304	ITIP1	Integration Test Input Register 1 [31:1] Reserved [0] SDIN
0x40024308	ITOP1	Integration Test Output Register 1 [31:4] Reserved [3] IRQOUT [2] LRCK [1] SCLK [0] SDOUT

3.15 Audio configuration

The FPGA design implements a simple serial interface that is based on I²C.

The following table describes the registers for the audio I²C control interface:

Table 3-3 I²C control interface registers

Address	Name	Description
0x40023000	CONTROL	Reads from this register return: 1 <i>Serial Data</i> (SDA) input 0 <i>Serial Clock</i> (SCL) output
0x40023000	SET[1:0]	Bits written as 0b1 set the appropriate output bit: 1 SDOUTEN_n 0 SCL
0x40023004	CLEAR[1:0]	Bits written as 0b1 clear the appropriate output bit: 1 SDOUTEN_n 0 SCL Reads from this register return 0b00.

The serial data is driven LOW when **SDOUTEN_n** is driven high, otherwise it is configured as an input pin.

The audio I²C control interface drives the Cirrus Logic CS42L52 codec chip on the baseboard.

3.16 FPGA system control and I/O

The FPGA design implements an FPGA system control block.

The following table shows the system control and I/O memory map:

Table 3-4 System control and I/O memory map

Address	Name	Description
0x40028000	LED0	LED connections [31:2] Reserved [1:0] LED
0x40028004	-	Reserved
0x40028008	BUTTON	Buttons [31:2] Reserved [1:0] Buttons
0x4002800C	-	Reserved
0x40028010	CLK1HZ	1Hz up counter
0x40028014	CLK100HZ	100Hz up counter
0x40028018	COUNTER	Cycle Up Counter Increments when the 32-bit prescale counter reaches zero.
0x4002801C	PRESCALE	Contains the reload value for the prescale counter.
0x40028020	PSCNTR	32-bit prescale counter – current value of the prescaler counter. The Cycle Up Counter increments when the prescale down counter reaches zero. The prescaler counter is reloaded with PRESCALE after reaching zero.
0x40028024	-	Reserved
0x4002804C	MISC	Miscellaneous control [31:7] Reserved [6] CLCD_BL_CTRL [5] CLCD_RD [4] CLCD_RS [3] CLCD_RESET [2] Reserved [1] SPI_nSS [0] CLCD_CS

Chapter 4

Clocks

This chapter describes the source and derived clocks for the FPGA design.

It contains the following sections:

- [4.1 Source clocks on page 4-43.](#)
- [4.2 Derived clocks on page 4-44.](#)

4.1 Source clocks

There are several source clocks for the FPGA design.

The following table shows the source clocks and their respective frequency for the system:

Table 4-1 Source clocks

Name	Purpose	Frequency
OSCCLK[0]	Source clock for system PLL. This generates the system processor clock and ZBT phase shifted clock.	50MHz
OSCCLK[1]	Source clock for audio PLL. This generates two audio clocks.	24.576MHz
OSCCLK[2]	Source clock to the peripherals PLL. The output from this PLL is unused.	25MHz
CFGCLK	Used by the <i>Motherboard Configuration Controller</i> (MCC) to load the <i>Serial Communication Controller</i> (SCC). For more information, see the 5.1 SCC interface overview on page 5-46 .	0.5MHz
CS_TCK	Debugger clock	Determined by the debugger.
SPICFGCLK	Used by MCC to load the software to the ZBT memories.	7.5MHz

4.2 Derived clocks

There are several derived clocks for the FPGA design.

The following table shows the derived clocks and respective frequencies for the system:

Table 4-2 Derived clocks

Name	Frequency	Division Factor	Multiplication Factor	Derived From
SYSCLK	25MHz	2	1	OSCCLK[0]
DBGCLK	25MHz	2	1	OSCCLK[0]
SPICLCD	25MHz	2	1	OSCCLK[0]
SPICON	25MHz	2	1	OSCCLK[0]
I2CCLCD	25MHz	2	1	OSCCLK[0]
I2CAUD	25MHz	2	1	OSCCLK[0]
AUDMCLK	12.29MHz	2	1	OSCCLK[1]
AUDSCLK	3.07MHz	8	1	OSCCLK[1]

The system clock is **SYSCLK** in the [Table 4-2 Derived clocks on page 4-44](#).

The clock frequencies are controlled by the configuration file `an511_v1.txt` located in the following directory:

```
MPS2+/MB/HBI0263C/AN511
```

As **OSCCLK[0]** controls the Cortex-M3 DesignStart Eval system clock (**SYSCLK**), it is possible to increase the clock by changing the **OSC0** value. This value can be set from 2MHz to 230MHz, with a 1% accuracy, subject to the constraints of FPGA timing performance.

Note

If the source clocks are changed, you must modify the `AN511_SMM_CM3DS.sdc` in the `<install_directory>/m3designstart/fpga/AN511_SMM_CM3DS/synthesis/` directory to reflect the new clock frequencies.

Chapter 5

Serial Communication Controller

This chapter describes the *Serial Communication Controller* (SCC) used in the Cortex-M3 DesignStart Eval FPGA image.

It contains the following sections:

- [5.1 SCC interface overview on page 5-46.](#)
- [5.2 SCC memory map on page 5-47.](#)

5.1 SCC interface overview

The FPGA design implements communication between the microcontroller and the FPGA system through an SCC interface.

The following diagram provides an overview of the SCC interface:

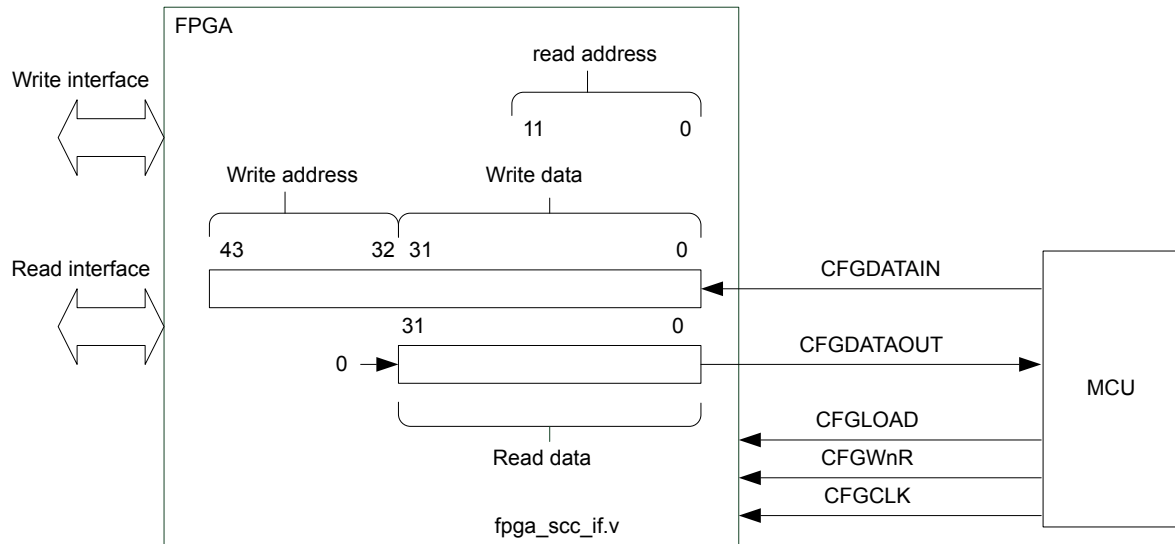


Figure 5-1 SCC interface

The read addresses and write addresses of the SCC interface do not use bits [1:0].

All address words are word-aligned.

5.2 SCC memory map

The following table shows the SCC register memory map:

Table 5-1 SCC register memory map

Address	Name	Description
0x00000000	CFG_REG0	[31:0] Reserved
0x00000004	CFG_REG1	[31:8] Reserved [7:0] Motherboard Configuration Controller (MCC) LEDs 0 Off 1 On
0x00000008	CFG_REG2	Reserved
0x0000000C	CFG_REG3	[31:8] Reserved [7:0] MCC LEDs 0 Off 1 On
0x00000010	CFG_REG4	[31:4] Reserved [3:0] Board revision
0x00000014	CFG_REG5	Reserved
0x00000018	CFG_REG6	Reserved
0x0000001C	CFG_REG7	Reserved
0x00000020 - 0x0000009C	-	Reserved
0x000000A0	SYS_CFGDATA_RTN	32-bit serial configuration data. This is a read/write register.
0x000000A4	SYS_CFGDATA_OUT	32-bit APB configuration data. This is a read/write register.
0x000000A8	SYS_CFGCTRL	[31] Start (generates interrupt on write to this bit) [30] Read/write access [29:26] Reserved [25:20] Function value [19:12] Reserved [11:0] Device (value of 0/1/2 for supported clocks)
0x000000AC	SYS_CFGSTAT	[31:2] Reserved [1] Error [0] Complete
0x000000AD – 0x000000FC	-	Reserved

Table 5-1 SCC register memory map (continued)

Address	Name	Description
0x00000100	SCC_DLL	<p>DLL lock register</p> <p>[31: 24] Indicate that the DLL lock is masked. These bits are mapped to DLL LOCK MASK[7:0].</p> <p>[23:16] Indicate whether the DLLs are locked or unlocked. These bits are mapped to DLL LOCK MASK[7:0].</p> <p>[15:1] Reserved</p> <p>[0] Indicates whether all enabled DLLs are locked.</p>
0x00000104 – 0x00000FF4	-	Reserved
0x00000FF8	SCC_AID	<p>SCC AID register is a read-only register.</p> <p>[31: 24] FPGA build number.</p> <p>[23:20] V2M-MPS2+ target board revision (A = 0, B = 1).</p> <p>[19:8] Reserved</p> <p>[7:0] Number of SCC configuration registers.</p>
0x00000FFC	SCC_ID	<p>SCC ID register is a read-only register.</p> <p>[31: 24] Implementer ID: 0x41 = ARM</p> <p>[23:20] Application note IP variant number</p> <p>[19:16] IP Architecture: 0x4 = AHB</p> <p>[15:12] Reserved</p> <p>[11:4] Primary part number: 511 = AN511</p> <p>[3:0] Reserved</p>

Chapter 6

FPGA build

This chapter describes the steps that are required to build an FPGA bit file from the supplied source code.

It contains the following sections:

- [6.1 Build flow on page 6-50.](#)
- [6.2 Build requirements on page 6-52.](#)

6.1 Build flow

Cortex-M3 DesignStart Eval includes a prebuilt FPGA image file that is based on an example system. However, if your design is different from the example system, then you are required to build your own FPGA image file.

The build flow involves two stages, which are creating a new user bit file, and loading the bit file onto the MPS2+ platform.

This section contains the following subsections:

- [6.1.1 Creating a user bit file on page 6-50.](#)
- [6.1.2 Loading a new bit file onto the MPS2+ platform on page 6-50.](#)

6.1.1 Creating a user bit file

To create a user bit file, follow these steps:

1. Open the Intel Quartus software.
 - a. Select **File > Open Project**. Navigate to the project file:
2. Open a terminal window (Linux) or a command prompt (Microsoft Windows).
3. Navigate to the following directory:

```
/m3designstart/fpga/AN511_SMM_CM3DS/synthesis/AN511_SMM_CM3DS.qpf
```

- b. Select **Processing > Start Compilation** to compile the design. This compiles the standard design and creates various report files in the `output_files` folder. After the compilation is completed, an SOF file, titled `AN511_SMM_CM3DS.sof`, is created in the `output_files` folder.

4. Execute the following command:

```
/m3designstart/fpga/AN511_SMM_CM3DS/synthesis/
```

```
make convert
```

This converts the SOF file to an RBF file, titled `an511_01.rbf`, in the current directory.

————— **Note** —————

To allow for a degree of traceability between different user bit files, the final digit can be used as a version number.

The *Motherboard Configuration Controller* (MCC) for the MPS2+ platform only supports a format of 8:3 characters for the filename, so only one digit is possible as the suffix. To change this digit when generating the RBF file, modify the `REV` variable in the header of the makefile with any text editor. If you provide an RBF file that uses a long file name, the MPS2+ platform will not boot.

6.1.2 Loading a new bit file onto the MPS2+ platform

To load a new user bit file onto the MPS2+ platform, execute the following instructions:

1. Power up the MPS2+ platform, connect a USB cable to your computer and the USB-B port (labeled USB) on the board. The computer recognizes a new USB device called `V2M_MPS2`.
2. Access the MPS2+ platform directory and copy the `an511_XX.rbf` file to the folder `MB\HBI0263C\AN511`.
3. Change which image is being programmed into the FPGA. In the USB device, the `board.txt` file allows you to select which hardware image is programmed into the FPGA. This affects which project to run. The board file is located in the directory `MB\HBI0263C\board.txt`. The image to be loaded is selected with the line starting with `APPPFILE:`. A selection of images has been preloaded onto the USB device. All but one has been commented out with a `'` character.

To select an image, remove the comment ';' in front of the appropriate line and ensure that all other APPFILE: references are preceded by the comment character. In the following example, the board.txt selects Application Note 511 which is the Cortex-M3 DesignStart Eval.

```
BOARD: HBI0263
TITLE: Motherboard configuration file
[MCCS]
MBBIOS: mbb_v217.ebf ; MB BIOS IMAGE

processor [APPLICATION NOTE] ; Please select the required
;APPFILE: AN382/an382_v2.txt ; - Cortex-M0
;APPFILE: AN383/an383_v2.txt ; - Cortex-M0+
;APPFILE: AN384/an384_v2.txt ; - Cortex-M1
;APPFILE: AN385/an385_v2.txt ; - Cortex-M3
;APPFILE: AN386/an386_v2.txt ; - Cortex-M4
;APPFILE: AN399/an399_v2.txt ; - Cortex-M7
;APPFILE: AN400/an400_v2.txt ; - Cortex-M4 with coresight
;APPFILE: AN387/an387_v3.txt ; - Cortex-M0 DesignStart
APPFILE : AN511/an511_v1.txt ; - Cortex M3 DesignStart
```

- Specify the bit file to use. Modify the file an511_v1.txt located in the MB\HBI0263C\AN511 folder. Use the following line to select which bit file to load:

F0FILE: an511_v1.rbe ;FPGA0 Filename

————— **Note** —————

ARM supplies an encrypted prebuilt bit file (.rbe) with the MPS2+ platform. If you are rebuilding the bit file, the file produced is not encrypted (.rbf). Therefore, if you are not using the prebuilt FPGA image and you are modifying an511_v1.txt, it is necessary to change the file extension from .rbe to .rbf.

- When you have completed steps 3 and 4, save and close both files. Eject the V2M_MPS2 volume using Windows Explorer.
- Press the ON button on the MPS2+ platform. When the bit file completes loading, either the board LEDs stop flashing, or a splash screen is displayed on the LCD display. The board is then ready to use.

————— **Note** —————

If the FPGA does not seem to have loaded correctly, then check the log.txt in the root directory of the MPS2+ platform. This log file contains the details of which files were loaded at boot-up.

6.2 Build requirements

To build the FPGA files, use the Intel Quartus software, version 16.1 onwards.

You can use the Lite Edition of Quartus Prime, which does not require a license.

Chapter 7

Integrating with mbed™ OS

This chapter describes the support available for integrating the FPGA system with mbed OS.

It contains the following section:

- [7.1 Compatibility with mbed™ OS on page 7-54.](#)

7.1 Compatibility with mbed™ OS

The MPS2+ platform is supported as a target in the mbed online compiler.

See <https://developer.mbed.org/platforms/ARM-CM3DS/> for more information on:

- Details of the platform.
- Links to the example code.
- mbed Cloud Quick Start application.

Chapter 8

Performance and utilization

This chapter describes the performance, resources, and utilization for the default system of the FPGA design in Cortex-M3 DesignStart Eval.

It contains the following sections:

- [8.1 Performance and clocks on page 8-56.](#)
- [8.2 Utilization of default system on page 8-57.](#)

8.1 Performance and clocks

There are several source and derived clocks in the FPGA board default system.

For details on the clocks used in the default system and their respective frequencies, see [Table 4-2 Derived clocks on page 4-44](#).

ARM recommends that you do not modify the clock frequencies. However, if you require modification of these clock frequencies, see [4.2 Derived clocks on page 4-44](#).

8.2 Utilization of default system

The FPGA on the MPS2+ platform is an Altera Cyclone V 5CEBA9F31C8.

The following table shows the resources of the FPGA:

Table 8-1 FPGA resources

Resources	Value
LUTs	113 560
DFFs	454 240
Memory	3.9Mbit
DSP	342
PLLs	7

The following table shows the Cortex-M3 DesignStart Eval code utilization of FPGA resources:

Table 8-2 Resource utilization

Resources	Utilization
LUTs	20%
DFFs	6%
Memory	32%
DSP	<1%
PLLs	42%

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-59.*

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	-	-