



# Arm® Corstone™ SSE-310 with Cortex®-M85 and Ethos™-U55 : Example Subsystem for MPS3

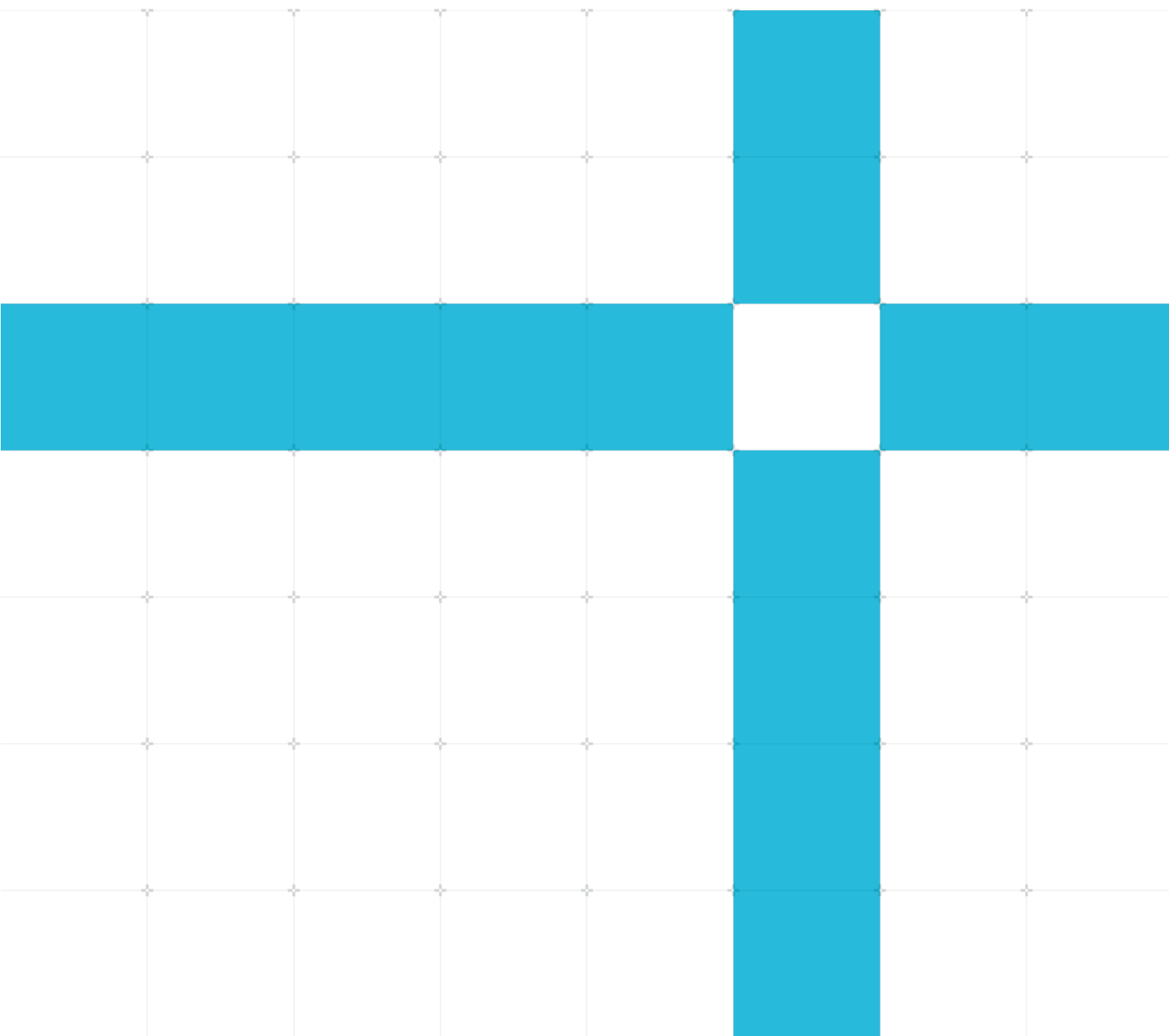
## Application Note AN555

Non-Confidential

Copyright © 2022 Arm Limited (or its affiliates).  
All rights reserved.

**Issue 1**

107642 (DAI0555)



# Arm® Corstone™ SSE-310 with Cortex®-M85 and Ethos™-U55 : Example Subsystem for MPS3

## Application Note AN555

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
1	12 Dec 2022	Non-Confidential	First Issue

## 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 click through or signed written agreement covering this document with Arm, then the click through or 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.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US 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/company/policies/trademarks>.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.  
110 Fulbourn Road, Cambridge, England CB1 9NJ.  
(LES-PRE-20348)

## 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.

## Product Status

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

## Web Address

[developer.arm.com](http://developer.arm.com)

## Inclusive language commitment

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

We believe that this document contains no offensive language. To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

## LICENCE GRANTS

THE END USER LICENCE AGREEMENT FOR THE ARM SYSTEM OR SUBSYSTEM FOR AN ARM FPGA PROTOTYPING BOARD ("THE LICENCE"), LES-PRE-21902 SP-Version: 3.0, DEFINES THE LICENCE GRANTS.

### DELIVERABLES

#### Part A

##### Hardware Binaries:

Encrypted FPGA bitstream file containing the SSE-310 product and other Arm technology.

##### Software Binaries:

Motherboard Configuration Controller binary, including Arm® Keil® USB and SD card drivers, and Analog Devices FMC EEPROM reader.  
Self-test binary.

##### Documentation:

Documentation, provided as PDF

#### Part B

##### Example Code:

Platform initialisation source code  
Platform specific libraries and source code  
Selftest example source code  
Arm source code portions of the Selftest software project

#### Part C

None

#### Part D

None

# Contents

<b>1.</b>	<b>Introduction.....</b>	<b>8</b>
1.1.	Purpose of this application note.....	8
1.2.	Intended audience.....	8
1.3.	Conventions.....	8
1.4.	Additional Reading.....	10
1.5.	Feedback.....	10
1.6.	Terms and Abbreviations.....	12
1.7.	Subsystem Version Details.....	13
1.8.	Encryption key.....	13
1.9.	Additional required hardware.....	14
<b>2.</b>	<b>Overview.....</b>	<b>16</b>
2.1.	System Block Diagram.....	16
2.2.	SSE-310 Configuration.....	17
2.3.	SIE-300 components.....	20
2.4.	SIE-200 components.....	20
2.5.	XHB-500.....	20
2.6.	Memory protection.....	20
2.7.	Memory Map Overview.....	21
2.8.	Expansion System peripherals.....	26
2.9.	FPGA Utilization.....	31
<b>3.</b>	<b>Programmer's model.....</b>	<b>32</b>
3.1.	Instruction Tightly Coupled Memory (ITCM).....	32
3.2.	FPGA SRAM.....	32
3.3.	Data Tightly Coupled Memory (DTCM).....	32
3.4.	QSPI.....	33
3.5.	DDR4.....	34
3.6.	SIE-300 MPC.....	34
3.7.	AHB GPIO.....	35
3.8.	SPI.....	35
3.9.	SBCon (I <sup>2</sup> C).....	36

3.10.	UART .....	38
3.11.	Colour LCD parallel interface.....	39
3.12.	Ethernet .....	41
3.13.	USB .....	41
3.14.	Real Time Clock.....	41
3.15.	Audio I <sup>2</sup> S.....	42
3.16.	Audio Configuration .....	43
3.17.	FPGA system control and I/O.....	44
3.18.	FPGA Serial Configuration Controller (SCC).....	46
3.19.	DMA .....	48
<b>4.</b>	<b>Clock architecture.....</b>	<b>50</b>
4.1.	Source clocks.....	50
4.2.	Clocks generated within the FPGA .....	51
4.3.	SSE-310 Clocks.....	51
<b>5.</b>	<b>Reset architecture.....</b>	<b>52</b>
5.1.	FPGA Resets.....	52
<b>6.</b>	<b>FPGA Secure Privilege Control .....</b>	<b>54</b>
<b>7.</b>	<b>Interrupts.....</b>	<b>58</b>
7.1.	FPGA interrupt map.....	58
7.2.	UART interrupts.....	60
<b>8</b>	<b>Shield Support.....</b>	<b>61</b>
<b>9.</b>	<b>QSPI Flash.....</b>	<b>63</b>
<b>10.</b>	<b>Debug Infrastructure .....</b>	<b>66</b>
10.1.	DAP-Lite2.....	66
10.2.	Debug timestamp generator.....	66
10.3.	TPIU-M.....	66
<b>11.</b>	<b>ZIP Bundle Description .....</b>	<b>67</b>
11.1.	Overall Structure .....	67
11.2.	Documentation.....	67
11.3.	MPS3 Board Revision and Support.....	68

11.4. Bundle Directory Tree/Structure .....	69
<b>12. Software and Debug .....</b>	<b>70</b>
12.1. Overview .....	70
12.2. Building the Software.....	71
12.3. Debug Support.....	73
12.4. Establishing a Debug Session.....	79
<b>13. Using AN555 on the MPS3 Board .....</b>	<b>81</b>
13.1. Pre-Requisites.....	81
13.2. Loading the Boardfiles .....	82
13.3. UART Serial ports .....	83
13.4. UART serial port settings.....	83
13.5. MPS3 USB Serial Port Drivers .....	83
13.6. MCC Memory mapping.....	84
13.7. MCC Debug UART – Serial Port 0 .....	85
13.8. Memory Preload .....	89
<b>14. Other Software Applications .....</b>	<b>91</b>

# 1. Introduction

## 1.1. Purpose of this application note

This document describes the features and functionality of AN555, which is an example FPGA implementation. AN555 is based on the Arm® Corstone™ SSE-310 Example Subsystem, which is extended to support FPGA and hardware peripherals.

## 1.2. Intended audience

This Application Note is written for experienced hardware, System-on-Chip (SoC) and software engineers who might or might not have experience with Arm products.

## 1.3. Conventions

The following subsections describe conventions used in Arm documents.

### 1.3.1. 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: <https://developer.arm.com/glossary>.

### 1.3.2. Typographical conventions

Convention	Use
<i>italic</i>	Introduces citations.
<b>bold</b>	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
<code>monospace</code>	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
<code>monospace</code> <b>bold</b>	Denotes language keywords when used outside example code.



Convention	Use
monospace <u>underline</u>	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <code>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</code>
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.
 Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
 Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
 Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.
 Note	This represents an important piece of information that needs your attention.
 Tip	This represents a useful tip that might make it easier, better or faster to perform a task.
 Remember	This is a reminder of something important that relates to the information you are reading.

## 1.4. Additional Reading

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

**Table 1-1: Arm publications**

Document name	Document ID	Licensee only
Arm® MPS3 FPGA Prototyping Board Technical Reference Manual	100765	No
Arm® MPS3 FPGA Prototyping Board Getting Started Guide	107789	No
Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual	102778	No
Arm® Ethos™-U55 NPU Technical Reference Manual	102420	No
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	107080	No
Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual	101526	No
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® CoreLink™ XHB-500 Bridge Technical Reference Manual	101375	No
Arm® Debug Interface Architecture Specification ADIv6.0.	IHI 0074	No
PrimeCell® Single Master DMA Controller (PL081) Technical Reference Manual	DDI 0218E	No

## 1.5. Feedback

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

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

### 1.5.1. 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.

## 1.5.2. Feedback on content

If you have comments on content, send an email to [errata@arm.com](mailto:errata@arm.com) and give:

- The title Arm® Corstone™ SSE-310 with Cortex®-M85 and Ethos™-U55 : Example Subsystem for MPS3 - Application Note AN555.
- The number DAI 0555 Issue 1
- If viewing a PDF version of a document, the page number(s) to which your comments refer.
- If viewing online, the topic names to which your comments apply.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.



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.

---

## 1.5.3. Other information

Arm also welcomes general suggestions for additions and improvements

- Arm Documentation, <https://developer.arm.com/documentation/>
- Arm Technical Support Knowledge Articles, <https://www.arm.com/support/technical-support>
- Arm Support, <https://www.arm.com/support>
- Arm Glossary, <https://developer.arm.com/documentation/aeg0014/g>
- MCBQVGA-TS-Display-v12 – Keil MCBSTM32F200 display board schematic <https://www.keil.com/mcbstm32c/mcbstm32c-display-board-schematics.pdf>

## 1.6. Terms and Abbreviations

AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
AXI	Advanced Extensible Interface
BRAM	FPGA Block RAM
CMSDK	Cortex-M System Design Kit
CPU	Central Processing Unit
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
KB	Kilobyte
LAR	Long Address Range (preloading mechanism)
LUTs	Lookup tables
MB	Megabyte
MCC	Motherboard Configuration Controller
MPC	Memory Protection Controller
MPS3	Microcontroller Prototyping System 3
MSC	Manager Security Controller
NIC	Network Interconnect
NVM	Non-Volatile Memory
Pmod™	Peripheral module
PPC	Peripheral Protection Controller
QSPI	Quad SPI
RAM	Random Access Memory
RAZ	Read as Zero
ROM	Read Only Memory
RTC	Real Time Clock
SCC	Serial Configuration Controller
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
NVM	Non-volatile Memory
WI	Write Ignored
W0	Write Zero

## 1.7. Subsystem Version Details

Table 1-2: IP Versions

Version	Descriptions
r0p0 <sup>1</sup>	Arm® Corstone™ SSE-310 Example Subsystem <sup>1</sup>
r0p2 <sup>2</sup>	Arm® Cortex®-M85 Processor <sup>2</sup>
r2p0	Arm® Ethos™-U55 NPU
r1p1	Arm® Cortex®-M System Design Kit
r3p2	Arm® CoreLink™ SIE-200
r1p2	Arm® CoreLink™ SIE-300
r0p5	Arm® CoreLink™ PCK-600
r0p1	Arm® CoreLink™ XHB-500
r2p1	Arm® CoreSight™ DAP-Lite2
r0p1	Arm® CoreSight™ TPIU-M
r1p2	Arm® CoreLink™ NIC-400 Network Interconnect



Remember

1. – The Arm® Corstone™ SSE-310 Example Subsystem NIC400 configuration for the Main Interconnect has been changed from the default configuration which would be found in the released Example Subsystem.
2. – The Arm® Cortex®-M85 Processor has been patched, this relates to an Errata. A Link to the Errata can be found in the `Release_Notes.txt`.

## 1.8. Encryption key

Arm supplies the MPS3 prototyping board with a decryption key programmed into the FPGA. This key is needed to enable loading of prebuilt encrypted images.



Note

The FPGA programming file that is supplied as part of the bundle is encrypted.



Caution

A battery supplies power to the key storage area of the FPGA. Any keys stored in the FPGA might be lost when battery power is lost. If this happens you must return the board to Arm for reprogramming of the key.

## 1.9. Additional required hardware



One or two 32 MB QSPI flash Pmod module are required to support the External QSPI memory (see [QSPI](#)) in this implementation. If this is not fitted, the `an555_v2.txt` file on the MPS3 SD card needs to be checked and potentially modified.

The following lines need to be set to FALSE or TRUE depending on the number of QSPI modules that are fitted to the Board.

```
FPGA_QSPI_PMOD0: TRUE      ;QSPI Pmod™ board inserted on shield0 / pmod0
FPGA_QSPI_PMOD1: TRUE      ;QSPI Pmod™ board inserted on shield0 / pmod1
```

These lines are used by the MCC on the MPS3 Board to enable or disable the QSPI controllers inside the design.

The table below shows the Pmod module required for this implementation:

**Table 1-3: Additional Hardware**

Description	Manufacturer	PN	Quantity
Pmod 32MB SF3 QSPI Flash Board	Digilent	410-340	2

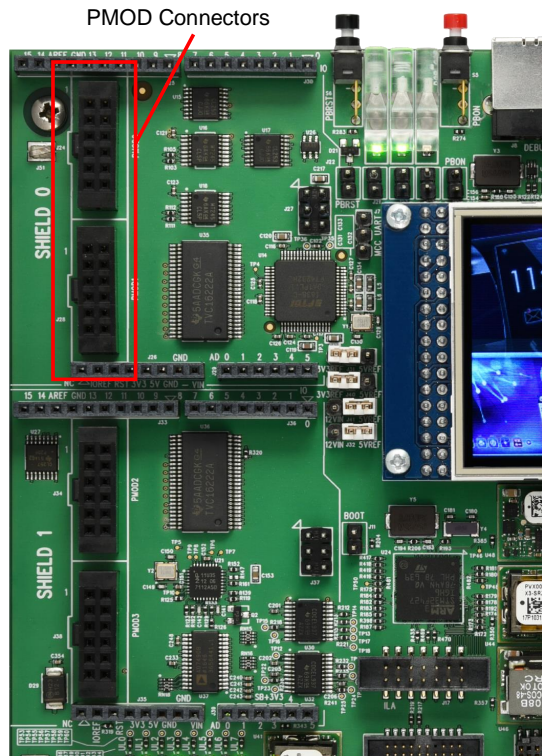
The AN555 implementation supports 64MB External QSPI Flash memory, this is provided by 2 32MB Pmod QSPI Flash memory modules.

The Pmod module is not distributed as part of the orderable MPS3 board product.

### 1.9.1. Installing the Pmod's

If Pmod's are being used, they need to be installed on the Shield 0 headers as shown in the figure below:

**Figure 1-1: MPS3 Pmod location and installed QSPI Pmod**



This document outlines the implementation of Corstone SSE-310 Subsystem in FPGA for use on the MPS3 development board. It details the implementation and the configuration options that are used in the design.

The following high-level block diagram shows the full MPS3 FPGA system:

[illegible]

Page 16 of 91



## 2.2. SSE-310 Configuration

The following section details the configuration values of the SSE-310 that are used in the FPGA Subsystem. Please see *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual* for the details of each configuration.



Any values that are in **bold** in the table below have been changed from the default values defined in the SSE-310 example subsystem, due to FPGA system implementation requirements.

**Table 2-1: SSE-310 Configuration**

Configuration Define	SSE-310 Default Value
NUMCPU	0
PILEVEL	1
CPU0TYPE	4
NUMNPU	1
NPU0TYPE	1
DEBUGLEVEL	2
HASCSS	0
NUMVMBANK	2
HASCRYTO	0
NUMDMA	0
DMATYPE	0
SECEXT	1
BUSPROT_PRESENT	0
ECC_PRESENT	0
INITTCMEN	0b11
INITPAHBEN	1
LOCKDCAIC	0
CFGBIGEND	0
CFGMEMALIAS	0b10000
CPU0_RAR	1
CPU0_LOCKSTEP	0
CPU0_FLOPPARITY	0
CPU0_PACBTI	1
CPU0_INITNSVTOR_ADDR_INIT	0x00000000
CPU0_IRQLVL	3
CPU0EXPNUMIRQ	100

Configuration Define	SSE-310 Default Value
CPU0EXPIRQDIS	100'b0
CPU0_DBGLVL	2
CPU0_ITM_PRESENT	1
CPU0_ETM_PRESENT	1
CPU0_FPU_PRESENT	1
CPU0_MVE_CONFIG	2
CPU0_MPU_S	8
CPU0_MPU_NS	8
CPU0_SAUDISABLE	0
CPU0_NUM_SAU_CONFIG	8
CPU0_IDCACHEID	0
CPU0_INSTR_CACHE_SIZE	0b01111
CPU0_DATA_CACHE_SIZE	0b01111
CPU0_CFGITCMSZ	0b0110
CPU0_CFGDTCMSZ	0b0110
CPU0_ITGUBLKSZ	6
CPU0_DTGUBLKSZ	6
HASCPUOCPIF	1
CPU0MCUROMADDR	0xE00FE
CPU0MCUROMVALID	1
CPU0_PMC_PRESENT	0
HASCPU0IWIC	0
CPU0CPUIDRST	0
CPU0_CTI_PRESENT	1
CPU0_INITECCEN	0
CPU0_CFGPAHBZ	0b010
CPU0_LOCKPAHB	1
NPU0_NUM_MACS	256
NPU0_CUSTOM_DMA_PRESENT	0
NPU0PORSLRST	0
NPU0PORPLRST	1
NUM_AXI_SUBORDINATES_EXP_MI	2
NUM_AHB_SUBORDINATES_EXP_P1HL	1
NUM_AHB_SUBORDINATES_EXP_P1LL	1
S_MID_WIDTH	5
TCM_ID_WIDTH	5

Configuration Define	SSE-310 Default Value
XS_ID_WIDTH	6
S_HMANAGER_WIDTH	5
VMADDRWIDTH	21
VMMPCBLKSIZE	10
PERIPHERAL_INTERCONNECT_ARBITRATION_SCHEME	"round"
XOM_USER_SIGNAL_PRESENT	0
<b>NSMSCEXPST</b>	<b>0xFFE0</b>
MPCEXPDIS	0xFFF8
<b>MSCEXPDIS</b>	<b>0xFFE0</b>
BRGEXPDIS	0x5A5A
PERIPHPPCEXP3DIS	0x5A5A
PERIPHPPCEXP2DIS	0xF000
PERIPHPPCEXP1DIS	0xFE00
<b>PERIPHPPCEXP0DIS</b>	<b>0x1FF8</b>
MAINPPCEXP3DIS	0x5A5A
MAINPPCEXP2DIS	0x5A5A
MAINPPCEXP1DIS	0xFFF0
<b>MAINPPCEXP0DIS</b>	<b>0x1E00</b>
UARCH_CONFIG	2
PDCMQCHWIDTH	4
LOGIC_RETENTION_PRESENT	0
COLDRESET_MODE	0
SOCVAR	0x0
SOCREV	0x0
SOCPRID	0x7E1
SOCIMPLID	0x43B
IMPLVAR	0x0
IMPLREV	0x0
IMPLPRID	0x74E
IMPLID	0x43B
EXPLOGIC_PRESENT	1
PERFORM_CONFIGCHECK	1
SOCRATES_CONFIG	""
CDC_SIMMODEL	1
CFG_FILE_NAME	sse310_user_cfg1.yaml

## 2.3. SIE-300 components

The AN555 FPGA implementation uses the following SIE-300 components:

- AXI5 Memory Protection Controller (MPC).

## 2.4. SIE-200 components

The AN555 FPGA implementation uses the following SIE-200 components:

- Arm® TrustZone® AHB5 Peripheral Protection Controller (PPC)
- Arm® TrustZone® AHB5 Manager Security Controller (MSC)
- AHB5 bus matrix
- AHB5 to AHB5 synchronous bridge
- AHB5 to APB synchronous bridge
- Arm® TrustZone® APB4 Peripheral Protection Controller (PPC)
- AHB5 default subordinate

## 2.5. XHB-500

The AN555 FPGA implementation uses one XHB-500, which is configured in AHB to AXI mode.

## 2.6. Memory protection

The SIE-300 MPC, and SIE-200 PPC components can affect memory and I/O security management and must be configured as required for your application.

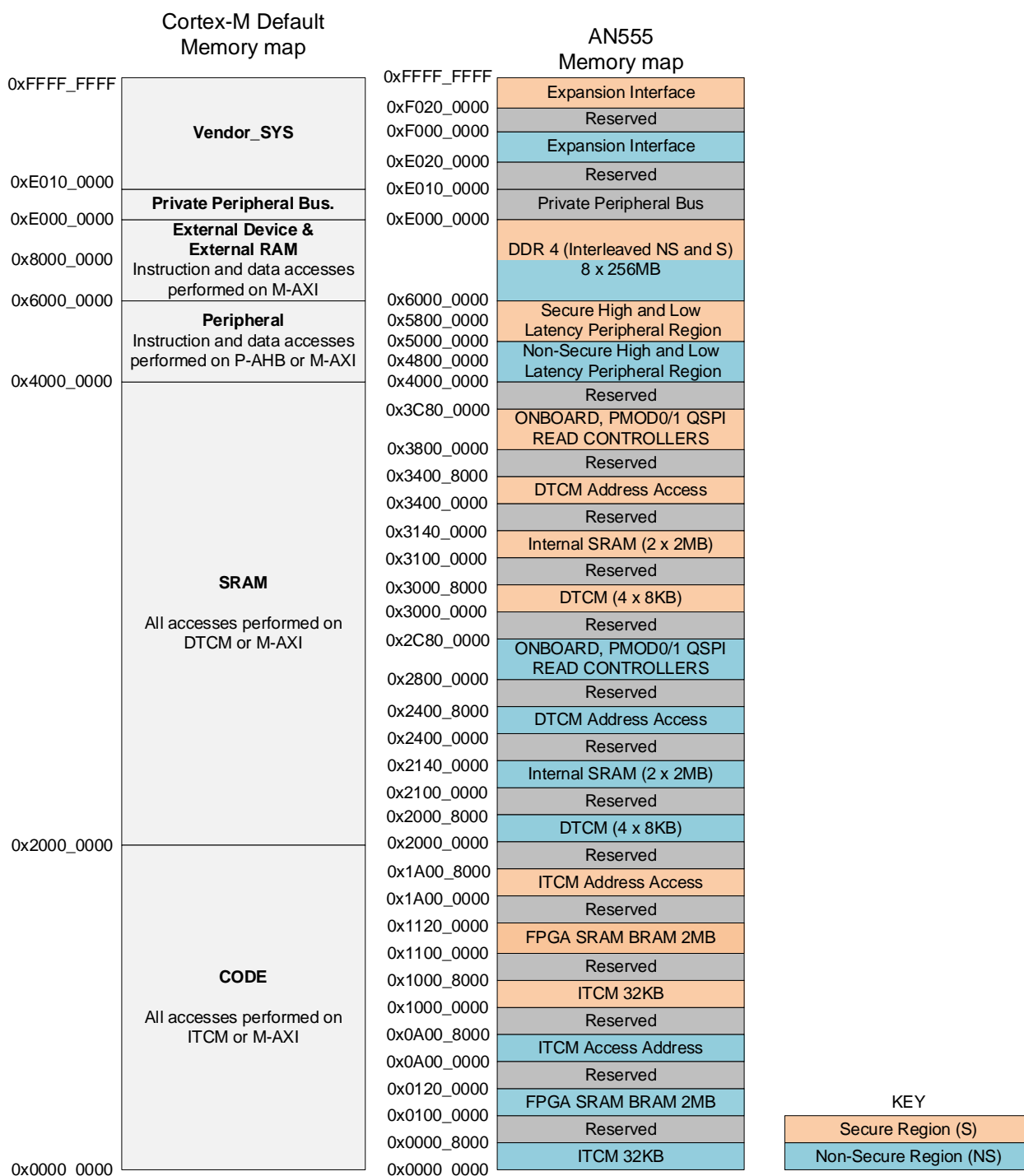
For more information, see:

- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*
- *Arm® CoreLink™ SIE-300 AXI5 System IP for Embedded Technical Reference Manual*

## 2.7. Memory Map Overview

The following figure shows the AN555 memory map and how it relates to the Armv8-M reference memory map. For information on the SSE-310 Subsystem peripherals, see the *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual*.

Figure 2-2: Memory map



**Figure 2-3: Low Latency Peripheral Memory map**

	0x5800_0000	Reserved
	0x5700_3000	DDR4 Memory Protection Controller (MPC)
	0x5700_2000	QSPI Memory Protection Controller (MPC)
	0x5700_1000	SRAM Memory Protection Controller (MPC)
	0x5700_0000	Reserved
0x4180_3000	0x5180_3000	PMOD1 QSPI Read/Write Controller
0x4180_2000	0x5180_2000	PMOD0 QSPI Read/Write Controller
0x4180_1000	0x5180_1000	Onboard QSPI Read/Write Controller
0x4180_0000	0x5180_0000	Reserved
0x4170_3000	0x5170_3000	Timing Adapter APB 2 – DDR4
0x4170_2000	0x5170_2000	Timing Adapter APB 1 - QSPI
0x4170_1000	0x5170_1000	Timing Adapter APB 0 – FPGA SRAM
0x4170_0000	0x5170_0000	Reserved
0x4160_0000	0x5160_0000	USB
0x4150_0000	0x5150_0000	Ethernet
0x4140_0000	0x5140_0000	Reserved
0x4120_4000	0x5120_4000	DMA 3
0x4120_3000	0x5120_3000	DMA 2
0x4120_2000	0x5120_2000	DMA 1
0x4120_1000	0x5120_1000	DMA 0
0x4120_0000	0x5120_0000	Reserved
0x4110_4000	0x5110_4000	GPIO 3
0x4110_3000	0x5110_3000	GPIO 2
0x4110_2000	0x5110_2000	GPIO 1
0x4110_1000	0x5110_1000	GPIO 0
0x4110_0000	0x5110_0000	Reserved
0x4010_0000	0x5010_0000	SSE-310 Subsystem Peripherals
0x4000_0000	0x5000_0000	
Non-Secure Low Latency Peripheral Region	Secure Low Latency Peripheral Region	

**Figure 2-4: High Latency Peripheral Memory map**

0x5000_0000	0x6000_0000	Reserved
0x4930_C000	0x5930_C000	RTC
0x4930_B000	0x5930_B000	CLCD Configuration register
0x4930_A000	0x5930_A000	Reserved
0x4930_9000	0x5930_9000	UART5 - FPGA_UART3
0x4930_8000	0x5930_8000	UART4 - UART Shield 1
0x4930_7000	0x5930_7000	UART3 - UART Shield 0
0x4930_6000	0x5930_6000	UART2 - FPGA_UART2
0x4930_5000	0x5930_5000	UART1 - FPGA_UART1
0x4930_4000	0x5930_4000	UART0 - FPGA_UART0
0x4930_3000	0x5930_3000	FPGA - IO (System Ctrl + I/O)
0x4930_2000	0x5930_2000	FPGA - I2S (Audio)
0x4930_1000	0x5930_1000	FPGA - SCC registers
0x4930_0000	0x5930_0000	Reserved
0x4920_9000	0x5920_9000	FPGA - SBCon I2C (DDR4 EEPROM)
0x4920_8000	0x5920_8000	Reserved
0x4920_7000	0x5920_7000	SBCon (I2C – Shield1)
0x4920_6000	0x5920_6000	SBCon (I2C - Shield0)
0x4920_5000	0x5920_5000	FPGA - PL022 (SPI Shield1)
0x4920_4000	0x5920_4000	FPGA - PL022 (SPI Shield0)
0x4920_3000	0x5920_3000	FPGA - PL022 (SPI ADC)
0x4920_2000	0x5920_2000	FPGA - SBCon I2C (Audio Conf)
0x4920_1000	0x5920_1000	FPGA - SBCon I2C (Touch)
0x4920_0000	0x5920_0000	Reserved
0x4810_0000	0x5810_0000	SSE-310 Subsystem Peripherals
0x4800_0000	0x5800_0000	

Non-Secure High Latency Peripheral Region	Secure High Latency Peripheral Region
---	---------------------------------------

## 2.7.1. SSE-310 FPGA memory map

The memory map implementation aligns with SSE-310 Subsystem host system memory map. The memory map is expanded to show the supported MPS3 peripherals in the Expansion Manager Regions and their mapping. For information on the SSE-310 Subsystem host system memory map, see *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual*.

**Table 2-2: Memory map overview**

ROW ID	Address		Size	Region Name	Description	Alias with Row ID	IDAU Region Values		
	From	To					Security	IDAU ID	NSC
1.	0x0000_0000	0x0000_7FFF	32KB	Code	ITCM <sup>3</sup>	7	NS	0	0
2.	0x0000_8000	0x00FF_FFFF	15.968MB	Reserved	Reserved				
3.	0x0100_0000	0x011F_FFFF	2MB	Code	FPGA SRAM (2MB) <sup>1</sup>	9			
4.	0x0120_0000	0x09FF_FFFF	142MB	Reserved	Reserved				
5.	0x0A00_0000	0x0A00_7FFF	32KB	Code	ITCM Access Address	11			
6.	0x0A00_8000	0x0FFF_FFFF	95.968MB	Reserved	Reserved				
7.	0x1000_0000	0x1000_7FFF	32KB	Code	ITCM <sup>3</sup>	1	S	1	CODE NSC
8.	0x1000_8000	0x10FF_FFFF	15.968MB	Reserved	Reserved				
9.	0x1100_0000	0x111F_FFFF	2MB	Code	FPGA SRAM (2MB) <sup>1</sup>	3			
10.	0x1120_0000	0x19FF_FFFF	142MB	Reserved	Reserved				
11.	0x1A00_0000	0x1A00_7FFF	32KB	Code	ITCM Access Address	5			
12.	0x1A00_8000	0x1FFF_FFFF	95.968MB	Reserved	Reserved				
13.	0x2000_0000	0x2000_7FFF	32KB	SRAM	DTCM (4 x banks of 8KB) <sup>3</sup>	23	NS	2	0
14.	0x2000_8000	0x20FF_FFFF	15.968MB	Reserved	Reserved				
15.	0x2100_0000	0x213F_FFFF	4MB	SRAM	Internal SRAM Area (SSE-310 implements 2x2MB)	25	NS	2	0
16.	0x2140_0000	0x23FF_FFFF	44MB	Reserved	Reserved				
17.	0x2400_0000	0x2400_7FFF	32KB	SRAM	DTCM Access Address	27	NS	2	0
18.	0x2400_8000	0x27FF_FFFF	63.968MB	Reserved	Reserved				
19.	0x2800_0000	0x287F_FFFF	8MB	SRAM	QSPI (8MB) <sup>1</sup>	29	NS	2	0
20.	0x2880_0000	0x2A7F_FFFF	32MB	SRAM	QSPI PMOD0 (32MB) <sup>1 4</sup>	30	NS	2	0
21.	0x2A80_0000	0x2C7F_FFFF	32MB	SRAM	QSPI PMOD1 (32MB) <sup>1 4</sup>	31	NS	2	0
22.	0x2C80_0000	0x2FFF_FFFF	56MB	Reserved	Reserved				
23.	0x3000_0000	0x3000_7FFF	32KB	SRAM	DTCM (4 x banks of 8KB) <sup>3</sup>	13	S	3	RAM NSC
24.	0x3000_8000	0x30FF_FFFF	15.968MB	Reserved	Reserved				
25.	0x3100_0000	0x313F_FFFF	4MB	SRAM	Internal SRAM Area (SSE-310 implements 2x2MB)	15	S	3	RAM NSC
26.	0x3140_0000	0x33FF_FFFF	44MB	Reserved	Reserved				



ROW ID	Address		Size	Region Name	Description	Alias with Row ID	IDAU Region Values		
	From	To					Security	IDAU ID	NSC
27.	0x3400_0000	0x3400_7FFF	32KB	SRAM	DTCM Access Address	17	S	3	RAM NSC
28.	0x3400_8000	0x37FF_FFFF	63.968MB	Reserved	Reserved				
29.	0x3800_0000	0x387F_FFFF	8MB	SRAM	QSPI (8MB) <sup>1</sup>	19	S	3	RAM NSC
30.	0x3880_0000	0x3A7F_FFFF	32MB	SRAM	QSPI PMOD0 (32MB) <sup>1,4</sup>	20	S	3	RAM NSC
31.	0x3A80_000	0x3C7F_FFFF	32MB	SRAM	QSPI PMOD1 (32MB) <sup>1,4</sup>	21	S	3	RAM NSC
32.	0x3C80_0000	0x3FFF_FFFF	56MB	Reserved	Reserved				
33.	0x4000_0000	0x47FF_FFFF	128MB	Peripheral	Non-Secure Low Latency Peripheral Region	35	NS	4	0
34.	0x4800_0000	0x4FFF_FFFF	128MB	Peripheral	Non-Secure High Latency Peripheral Region	36	NS	4	0
35.	0x5000_0000	0x57FF_FFFF	128MB	Peripheral	Secure Low Latency Peripheral Region	33	S	5	0
36.	0x5800_0000	0x5FFF_FFFF	128MB	Peripheral	Secure High Latency Peripheral Region	34	S	5	0
37.	0x6000_0000	0x6FFF_FFFF	256MB	External RAM	DDR4 <sup>1</sup>		NS	6	0
38.	0x7000_0000	0x7FFF_FFFF	256MB	External RAM	DDR4 <sup>1</sup>		S	7	0
39.	0x8000_0000	0x8FFF_FFFF	256MB	External RAM	DDR4 <sup>1</sup>		NS	8	0
40.	0x9000_0000	0x9FFF_FFFF	256MB	External RAM	DDR4 <sup>1</sup>		S	9	0
41.	0xA000_0000	0xAFFF_FFFF	256MB	External device	DDR4 <sup>1</sup>		NS	A	0
42.	0xB000_0000	0xBFFF_FFFF	256MB	External device	DDR4 <sup>1</sup>		S	B	0
43.	0xC000_0000	0xCFFF_FFFF	256MB	External device	DDR4 <sup>1</sup>		NS	C	0
44.	0xD000_0000	0xDFFF_FFFF	256MB	External device	DDR4 <sup>1</sup>		S	D	0
45.	0xE000_0000	0xE00F_FFFF	1MB	EPPB	External Private Peripheral Bus		Exempt		
46.	0xE010_0000	0xE01F_FFFF	1MB	Vendor_SYS	Reserved		NS	E	0
47.	0xE020_0000	0xEFFF_FFFF	254MB	Vendor_SYS	Maps to HMSTEXPILL Expansion Interface <sup>2</sup>		NS	E	0
48.	0xF000_0000	0xF01F_FFFF	2MB	Vendor_SYS	Reserved		S	F	0

ROW ID	Address		Size	Region Name	Description	Alias with Row ID	IDAU Region Values		
	From	To					Security	IDAU ID	NSC
49.	0xF020_0000	0xFFFF_FFFF	254MB	Vendor_SYS	Maps to HMSTEXPPILL Expansion Interface <sup>2</sup>		S	F	0



Note

<sup>1</sup> – Security Access is controlled by MPC.

<sup>2</sup> – Accesses to these addresses results in an AHB5 error response.

<sup>3</sup> – For security settings, controls and features. See the *Arm® Corstone™ SSE-310 Example Subsystem Technical Reference Manual*.

<sup>4</sup> – Memory map region changes based on the presence or absence of QSPI memory blocks. See [QSPI Flash](#) for further details.

## 2.8. Expansion System peripherals

All FPGA peripherals are mapped to four areas of the memory map. The addresses and interfaces to access the four regions are:

Non-Secure Low Latency region:

- 0x4000\_0000 - 0x47FF\_FFFF
- Manager Peripheral Expansion Low Latency Interface (HMSTEXPPILL)

Non-Secure High Latency region:

- 0x4800\_0000 - 0x4FFF\_FFFF
- Manager Peripheral Expansion High Latency Interface (HMSTEXPPIHL)

Secure Low Latency region:

- 0x5000\_0000 - 0x57FF\_FFFF
- Manager Peripheral Expansion Low Latency Interface (HMSTEXPPILL)

Secure High Latency region:

- 0x5800\_0000 - 0x5FFF\_FFFF
- Manager Peripheral Expansion High Latency Interface (HMSTEXPPIHL)

To support TrustZone for Armv8-M and allow software to map these peripherals to Secure or Non-Secure address space, all peripherals are mapped twice and either an APB PPC or an AHB PPC gates the access to these peripherals.

## 2.8.1. Manager Peripheral Expansion Low Latency Interface Memory Map (HMSTEXPPILL)

The following table shows the FPGA peripheral mapping to the Non-Secure and Secure Low Latency region.



Reserved regions respond with RAZ/WI when accessed.

**Table 2-3: MSTEXPPILL Non-Secure Peripheral Map**

ROW ID	Address From	To	Size	Description	Alias with ROW ID	Port
1.	0x4000_0000	0x400F_FFFF		Subsystem peripherals	24.	
2.	0x4010_0000	0x410F_FFFF		Reserved	25.	
3.	0x4110_0000	0x4110_0FFF	4KB	GPIO 0	26.	AHB
4.	0x4110_1000	0x4110_1FFF	4KB	GPIO 1	27.	
5.	0x4110_2000	0x4110_2FFF	4KB	GPIO 2	28.	
6.	0x4110_3000	0x4110_3FFF	4KB	GPIO 3	29.	
7.	0x4110_8000	0x411F_FFFF		Reserved	30.	
8.	0x4120_0000	0x4120_0FFF	4KB	DMA 0	31.	AHB
9.	0x4120_1000	0x4120_1FFF	4KB	DMA 1	32.	
10.	0x4120_2000	0x4120_2FFF	4KB	DMA 2	33.	
11.	0x4120_3000	0x4120_3FFF	4KB	DMA 3	34.	
12.	0x4120_4000	0x413F_FFFF		Reserved	35.	
13.	0x4140_0000	0x414F_FFFF	1MB	Ethernet	36.	AHB
14.	0x4150_0000	0x415F_FFFF	1MB	USB	37.	
15.	0x4160_0000	0x416F_FFFF		Reserved	38.	
16.	0x4170_0000	0x4170_0FFF	4KB	Timing Adapter APB 0 – FPGA SRAM	39.	APB (Mem)
17.	0x4170_1000	0x4170_1FFF	4KB	Timing Adapter APB 1 - QSPI	40.	
18.	0x4170_2000	0x4170_2FFF	4KB	Timing Adapter APB 2 – DDR4	41.	
19.	0x4170_4000	0x417F_FFFF		Reserved	42.	
20.	0x4180_0000	0x4180_0FFF	4KB	Onboard QSPI Read/Write Controller	43.	AHB
21.	0x4180_1000	0x4180_1FFF	4KB	PMOD0 QSPI Read/Write Controller	44.	
22.	0x4180_2000	0x4180_2FFF	4KB	PMOD1 QSPI Read/Write Controller	45.	
23.	0x4180_3000	0x47FF_FFFF		Reserved	46.	

**Table 2-4: MSTEPPILL Secure Peripheral Map**

ROW ID	Address		Size	Description	Alias with ROW ID	Port
	From	To				
24.	0x5000_0000	0x500F_FFFF		Subsystem peripherals	1.	
25.	0x5010_0000	0x510F_FFFF		Reserved	2.	
26.	0x5110_0000	0x5110_0FFF	4KB	GPIO 0	3.	AHB
27.	0x5110_1000	0x5110_1FFF	4KB	GPIO 1	4.	
28.	0x5110_2000	0x5110_2FFF	4KB	GPIO 2	5.	
29.	0x5110_3000	0x5110_3FFF	4KB	GPIO 3	6.	
30.	0x5110_8000	0x511F_FFFF		Reserved	7.	
31.	0x5120_0000	0x5120_0FFF	4KB	DMA 0	8.	AHB
32.	0x5120_1000	0x5120_1FFF	4KB	DMA 1	9.	
33.	0x5120_2000	0x5120_2FFF	4KB	DMA 2	10.	
34.	0x5120_3000	0x5120_3FFF	4KB	DMA 3	11.	
35.	0x5120_4000	0x513F_FFFF		Reserved	12.	
36.	0x5140_0000	0x514F_FFFF	1MB	Ethernet	13.	AHB
37.	0x5150_0000	0x515F_FFFF	1MB	USB	14.	
38.	0x5160_0000	0x516F_FFFF		Reserved	15.	
39.	0x5170_0000	0x5170_0FFF	4KB	Timing Adapter APB 0 – FPGA SRAM	16.	APB (Mem)
40.	0x5170_1000	0x5170_1FFF	4KB	Timing Adapter APB 1 - QSPI	17.	
41.	0x5170_2000	0x5170_2FFF	4KB	Timing Adapter APB 2 – DDR4	18.	
42.	0x5170_4000	0x517F_FFFF		Reserved	19.	
43.	0x5180_0000	0x5180_0FFF	4KB	Onboard QSPI Read/Write Controller	20.	AHB
44.	0x5180_1000	0x5180_1FFF	4KB	PMOD0 QSPI Read/Write Controller	21.	
45.	0x5180_2000	0x5180_2FFF	4KB	PMOD1 QSPI Read/Write Controller	22.	
46.	0x5180_3000	0x56FF_FFFF		Reserved	23.	
47.	0x5700_0000	0x5700_0FFF	4KB	SRAM Memory Protection Controller (MPC)		APB (Mem)
48.	0x5700_1000	0x5700_1FFF	4KB	QSPI Memory Protection Controller (MPC)		
49.	0x5700_2000	0x5700_2FFF	4KB	DDR4 Memory Protection Controller (MPC)		
50.	0x5700_3000	0x57FF_FFFF		Reserved	23.	

## 2.8.2. Manager Peripheral Expansion High Latency Interface Memory Map (HMSTEXPPIHL)

The following table shows the FPGA peripheral mapping to the Non-Secure and Secure High Latency region.



Reserved regions respond with RAZ/WI when accessed.

**Table 2-5: MSTEXPPIHL Non-Secure Peripheral Map**

ROW ID	Address From	To	Size	Description	Alias with ROW ID	Port
1.	0x4800_0000	0x480F_FFFF		Subsystem peripherals	26.	
2.	0x4810_0000	0x491F_FFFF		Reserved	27.	
3.	0x4920_0000	0x4920_0FFF	4KB	FPGA - SBCon I2C (Touch)	28.	APB0
4.	0x4920_1000	0x4920_1FFF	4KB	FPGA - SBCon I2C (Audio Conf)	29.	
5.	0x4920_2000	0x4920_2FFF	4KB	FPGA - PL022 (SPI ADC)	30.	
6.	0x4920_3000	0x4920_3FFF	4KB	FPGA - PL022 (SPI Shield0)	31.	
7.	0x4920_4000	0x4920_4FFF	4KB	FPGA - PL022 (SPI Shield1)	32.	
8.	0x4920_5000	0x4920_5FFF	4KB	SBCon (I2C - Shield0)	33.	
9.	0x4920_6000	0x4920_6FFF	4KB	SBCon (I2C - Shield1)	34.	
10.	0x4920_7000	0x4920_7FFF		Reserved	35.	
11.	0x4920_8000	0x4920_8FFF	4KB	FPGA - SBCon I2C (DDR4 EEPROM)	36.	
12.	0x4920_9000	0x492F_FFFF		Reserved	37.	
13.	0x4930_0000	0x4930_0FFF	4KB	FPGA - SCC registers	38.	APB1
14.	0x4930_1000	0x4930_1FFF	4KB	FPGA - I2S (Audio)	39.	
15.	0x4930_2000	0x4930_2FFF	4KB	FPGA - IO (System Ctrl + I/O)	40.	
16.	0x4930_3000	0x4930_3FFF	4KB	UART0 - FPGA_UART0	41.	
17.	0x4930_4000	0x4930_4FFF	4KB	UART1 - FPGA_UART1	42.	
18.	0x4930_5000	0x4930_5FFF	4KB	UART2 - FPGA_UART2	43.	
19.	0x4930_6000	0x4930_6FFF	4KB	UART3 - UART Shield 0	44.	
20.	0x4930_7000	0x4930_7FFF	4KB	UART4 - UART Shield 1	45.	
21.	0x4930_8000	0x4930_8FFF	4KB	UART5 - FPGA_UART3	46.	
22.	0x4930_9000	0x4930_9FFF		Reserved	47.	
23.	0x4930_A000	0x4930_AFFF	4KB	CLCD Configuration register	48.	
24.	0x4930_B000	0x4930_BFFF	4KB	RTC	49.	
25.	0x4930_C000	0x4FFF_FFFF		Reserved	50.	

**Table 2-6: MSTEPPPIHL Secure Peripheral Map**

ROW ID	Address From	To	Size	Description	Alias with ROW ID	Port
26.	0x5800_0000	0x5810_1FFF		Subsystem peripherals	1.	
27.	0x5810_2000	0x591F_FFFF		Reserved	2.	
28.	0x5920_0000	0x5920_0FFF	4KB	FPGA - SBCon I2C (Touch)	3.	APB0
29.	0x5920_1000	0x5920_1FFF	4KB	FPGA - SBCon I2C (Audio Conf)	4.	
30.	0x5920_2000	0x5920_2FFF	4KB	FPGA - PL022 (SPI ADC)	5.	
31.	0x5920_3000	0x5920_3FFF	4KB	FPGA - PL022 (SPI Shield0)	6.	
32.	0x5920_4000	0x5920_4FFF	4KB	FPGA - PL022 (SPI Shield1)	7.	
33.	0x5920_5000	0x5920_5FFF	4KB	SBCon (I2C - Shield0)	8.	
34.	0x5920_6000	0x5920_6FFF	4KB	SBCon (I2C - Shield1)	9.	
35.	0x5920_7000	0x5920_7FFF		Reserved	10.	
36.	0x5920_8000	0x5920_8FFF	4KB	FPGA - SBCon I2C (DDR4 EEPROM)	11.	
37.	0x5920_9000	0x592F_FFFF		Reserved	12.	
38.	0x5930_0000	0x5930_0FFF	4KB	FPGA - SCC registers	13.	APB1
39.	0x5930_1000	0x5930_1FFF	4KB	FPGA - I2S (Audio)	14.	
40.	0x5930_2000	0x5930_2FFF	4KB	FPGA - IO (System Ctrl + I/O)	15.	
41.	0x5930_3000	0x5930_3FFF	4KB	UART0 - FPGA_UART0	16.	
42.	0x5930_4000	0x5930_4FFF	4KB	UART1 - FPGA_UART1	17.	
43.	0x5930_5000	0x5930_5FFF	4KB	UART2 - FPGA_UART2	18.	
44.	0x5930_6000	0x5930_6FFF	4KB	UART3 - UART Shield 0	19.	
45.	0x5930_7000	0x5930_7FFF	4KB	UART4 - UART Shield 1	20.	
46.	0x5930_8000	0x5930_8FFF	4KB	UART5 - FPGA_UART3	21.	
47.	0x5930_9000	0x5930_9FFF		Reserved	22.	
48.	0x5930_A000	0x5930_AFFF	4KB	CLCD Configuration register	23.	
49.	0x5930_B000	0x5930_BFFF	4KB	RTC	24.	
50.	0x5930_C000	0x5FFF_FFFF		Reserved	25.	

## 2.9. FPGA Utilization

The AN555 FPGA example image is designed for the MPS3 board, which uses a Xilinx Kintex Ultrascale XCKU115 FPGA. The FPGA features up to 8MB BRAM (2160 Block RAM tiles) and up to 663,360 LUTs. The FPGA full part number is XCKU115-FLVB1760-1-C.

The following table shows the numbers of LUTs and BRAMs available in the FPGA, and the numbers used by the AN555 FPGA image.

**Table 2-7: AN555 FPGA total utilization**

Site Type	FPGA total	Used	Utilization %
LUTs	663,360	401,633	60.55%
Block RAM Tile	2160	1624	75.19%



These numbers relate to the full AN555 FPGA example image. The values cannot be used to infer IP size, or the relative sizes of different IP blocks, because the implementation and system design can significantly differ depending on the target technology.

## 3. Programmer's model

This programmer's model is supplemental to the CMSDK, SIE-200 and SIE-300 documentation which describes many of the included components in more detail. The connectivity of the system is shown in the [System Block Diagram](#).

### 3.1. Instruction Tightly Coupled Memory (ITCM)

The ITCM is implemented with 32KB of FPGA SRAM connected to the SSE-310 TCM interconnect.

The Cortex-M85 processor inside the Subsystem accesses the ITCM with the following addresses. The base memory addresses are:

- Non-Secure Address Range: 0x0000\_0000 - 0x0000\_7FFF
- Secure Address Range: 0x1000\_0000 - 0x1000\_7FFF

Other Managers in the system access the ITCM with the following addresses. The base memory addresses are:

- Non-Secure Address Range: 0x0A00\_0000 - 0x0A00\_7FFF
- Secure Address Range: 0x1A00\_0000 - 0x1A00\_7FFF

### 3.2. FPGA SRAM

The code memory is implemented with 2MB of internal FPGA SRAM.

The base memory addresses are:

- Non-Secure Address Range: 0x0100\_0000 - 0x011F\_FFFF
- Secure Address Range: 0x1100\_0000 - 0x111F\_FFFF

### 3.3. Data Tightly Coupled Memory (DTCM)

The DTCM is implemented as 4 x 8KB of internal FPGA SRAM connected to the SSE-310 TCM interconnect.

The Cortex-M85 processor inside the Subsystem accesses the DTCM with the following addresses. The base memory addresses are:

- Non-Secure Address Range: 0x2000\_0000 - 0x2000\_7FFF
- Secure Address Range: 0x3000\_0000 - 0x3000\_7FFF

Other Managers in the system access the DTCM with the following addresses. The base memory addresses are:



- Non-Secure Address Range: 0x2400\_0000 – 0x2400\_7FFF
- Secure Address Range: 0x3400\_0000 - 0x3400\_7FFF

## 3.4. QSPI

The AN555 FPGA implementation can provide access to a maximum of 72MB external QSPI Flash memory. This is divided into a single onboard Flash memory providing 8MB, and 2 optional blocks of QSPI Flash memory (PMOD0 and PMOD1) providing 32MB each. For further information see [QSPI Flash](#).

Onboard QSPI can be accessed through a read-only interface directly. It can also be accessed indirectly through a read/write interface with the use of a software driver.

The base memory addresses for the QSPI Read only interface are:

- Non-Secure Address Range: 0x2800\_0000 - 0x287F\_FFFF
- Secure Address Range: 0x3800\_0000 - 0x387F\_FFFF

The base memory addresses for the QSPI Read/Write interface are:

- Non-Secure Address Range: 0x4180\_0000 - 0x4180\_0FFF
- Secure Address Range: 0x5180\_0000 - 0x5180\_0FFF

PMOD0 memory is implemented using an external Pmod module and can be accessed through a read-only interface directly. It can also be accessed indirectly through a read/write interface with the use of a software driver.

The base memory addresses for the QSPI Read only interface are:

- Non-Secure Address Range: 0x2880\_0000 - 0x2A7F\_FFFF
- Secure Address Range: 0x3880\_0000 - 0x3A7F\_FFFF

The base memory addresses for the QSPI Read/Write interface are:

- Non-Secure Address Range: 0x4180\_1000 - 0x4180\_1FFF
- Secure Address Range: 0x5180\_1000 - 0x5180\_1FFF

PMOD1 memory is implemented using an external Pmod module and can be accessed through a read-only interface directly. It can also be accessed indirectly through a read/write interface with the use of a software driver.

The base memory addresses for the QSPI Read only interface are:

- Non-Secure Address Range: 0x2A80\_0000 - 0x2C7F\_FFFF
- Secure Address Range: 0x3A80\_0000 - 0x3C7F\_FFFF

The base memory addresses for the QSPI Read/Write interface are:

- Non-Secure Address Range: 0x4180\_2000 - 0x4180\_2FFF
- Secure Address Range: 0x5180\_2000 - 0x5180\_2FFF

## 3.5. DDR4

The AN555 FPGA implementation provides access to 2GB of external DDR4 memory through the DDR4 controller. This is split into 8 x 256MB regions which are mapped as interleaved Non-Secure and Secure Regions.

The base addresses for the Non-Secure Address regions are:

- Non-Secure Address Range 1 0x6000\_0000 - 0x6FFF\_FFFF
- Non-Secure Address Range 2 0x8000\_0000 - 0x8FFF\_FFFF
- Non-Secure Address Range 3 0xA000\_0000 - 0xAFFF\_FFFF
- Non-Secure Address Range 4 0xC000\_0000 - 0xCFFF\_FFFF

The base addresses for the Secure Address regions are:

- Secure Address Range 1 0x7000\_0000 - 0x7FFF\_FFFF
- Secure Address Range 2 0x9000\_0000 - 0x9FFF\_FFFF
- Secure Address Range 3 0xB000\_0000 - 0xBFFF\_FFFF
- Secure Address Range 4 0xD000\_0000 - 0xDFFF\_FFFF

## 3.6. SIE-300 MPC

There are three MPCs implemented in the AN555 FPGA model and these are configured with the following block sizes:

- SRAM MPC, 16KB block size.
- QSPI MPC, 256KB block size.
- DDR4 MPC, 1MB block size.

The Configuration interfaces of these modules are mapped to secure address ranges and can be accessed at the following addresses:

- SRAM MPC address range 0x5700\_0000 - 0x5700\_0FFF
- QSPI MPC address range 0x5700\_1000 - 0x5700\_1FFF
- DDR4 MPC address range 0x5700\_2000 - 0x5700\_2FFF

## 3.7. AHB GPIO

The AN555 FPGA implementation uses four CMSDK AHB GPIO blocks, each providing 16 bits of I/O. The four modules are connected to the two Arduino compatible headers Shields 0 and 1 as follows:

**Table 3-1: GPIO mapping**

Shield	GPIO
SH0_IO [15:0]	GPIO0 [15:0]
SH0_IO [17:16]	GPIO2 [1:0]
SH1_IO [15:0]	GPIO1 [15:0]
SH1_IO [17:16]	GPIO2 [3:2]

The GPIO alternative function lines select whether peripherals or GPIOs are available on each pin.



For further information on the pin out on the MPS3 Shield and Pmod headers please refer to section 2.16 Arduino Shield and Pmod interfaces of the *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual*.

## 3.8. SPI

The AN555 FPGA model implements three PL022 SPI modules:

- One general purpose SPI module (SPI ADC) communicates with an onboard ADC. The analog pins of the Shield headers connect to the input channels of the ADC. Chip Select is managed by FPGA system control module.
- Two general purpose SPI modules connect to the Shield headers and provide an SPI interface on each header. These are alternative functions on the GPIO ports. See [Shield Support](#) for mappings. Select is managed by FPGA system control module.

## 3.9. SBCon (I<sup>2</sup>C)

The AN555 FPGA model implements five SBCon serial modules which are mapped to the high latency peripheral region:

- One SBCon module for use by the Colour LCD touch interface.

The base memory addresses are:

- Non-Secure Address Range                      0x4920\_0000 - 0x4920\_0FFF
- Secure Address Range                            0x5920\_0000 - 0x5920\_0FFF

- Two general purpose SBCon modules that connect to Shield 0 and Shield 1 and provide an I<sup>2</sup>C interface on each header. These are alt-functions on the GPIO ports. See [Shield Support](#) for mappings.

The base memory addresses for Shield 0 are:

- Non-Secure Address Range                      0x4920\_5000 - 0x4920\_5FFF
- Secure Address Range                            0x5920\_5000 - 0x5920\_5FFF

The base memory addresses for Shield 1 are:

- Non-Secure Address Range                      0x4920\_6000 - 0x4920\_6FFF
- Secure Address Range                            0x5920\_6000 - 0x5920\_6FFF

- One SBCon module, used to read EEPROM from DDR4 SODIMM.

The base memory addresses are:

- Non-Secure Address Range                      0x4920\_8000 - 0x4920\_8FFF
- Secure Address Range                            0x5920\_8000 - 0x5920\_8FFF

- One SBCon is used for the audio configuration controller for details please see [Audio Configuration](#).

The selftest software provided with AN555 includes example code for the colour LCD module control and audio interfaces.

The following table shows the two-wire SBCon registers in address offset order from the base memory address.

**Table 3-2: SBCon registers**

Offset	Register Name	Type	Reset	Description
0x000	SB_CONTROL	RAZ/WI	0x0	Bits[31:2]: Reserved
		RW	0b00	Read and set serial control bits: Bit[1] is SDA Bit[0] is SCL
0x004	SB_CONTROLC	RAZ/WI	0x0	Bits[31:2]: Reserved
		W0	0b00	Clear serial control bits: Bit[1] is SDA Bit[0] is SCL

## 3.10. UART

The AN555 FPGA model implements six CMSDK UART's:

- UART 0 - FPGA\_UART0

The base memory addresses are:

- Non-Secure Address Range 0x4930\_3000 - 0x4930\_3FFF
- Secure Address Range 0x5930\_3000 - 0x5930\_3FFF

- UART 1 - FPGA\_UART1

The base memory addresses are:

- Non-Secure Address Range 0x4930\_4000 - 0x4930\_4FFF
- Secure Address Range 0x5930\_4000 - 0x5930\_4FFF

- UART 2 - FPGA\_UART2

The base memory addresses are:

- Non-Secure Address Range 0x4930\_5000 - 0x4930\_5FFF
- Secure Address Range 0x5930\_5000 - 0x5930\_5FFF

- UART 3 - Shield 0

The base memory addresses are:

- Non-Secure Address Range 0x4930\_6000 - 0x4930\_6FFF
- Secure Address Range 0x5930\_6000 - 0x5930\_6FFF

- UART 4 - Shield 1

The base memory addresses are:

- Non-Secure Address Range 0x4930\_7000 - 0x4930\_7FFF
- Secure Address Range 0x5930\_7000 - 0x5930\_8FFF

- UART 5 - FPGA\_UART3

The base memory addresses are:

- Non-Secure Address Range 0x4930\_8000 - 0x4930\_8FFF
- Secure Address Range 0x5930\_8000 - 0x5930\_8FFF

UART's 3 and 4 are alternative functions on the GPIO ports. See [Shield Support](#) for mappings.



All UART's are driven from FPGA generated clock **PERIF\_CLK**. For details of the clocks please see [Clocks generated within the FPGA](#).

## 3.11. Colour LCD parallel interface

The colour LCD module has two interfaces:

- Parallel bus for sending image data to the LCD.
- I<sup>2</sup>C to transfer input data from the touch screen.

The colour LCD module is a custom peripheral that provides an interface to a STMicroelectronics STMPE811QTR Port Expander with Advanced Touch Screen Controller on the Keil MCBSTM32C display board. The schematic is listed in the reference section, see [Additional Reading](#). The Keil display board contains an AM240320LG display panel and uses a Himax HX8347-D LCD controller.

The selftest software provided with the application note includes drivers and example code for both interfaces.

The base memory addresses of the colour LCD parallel interface are:

- Non-Secure Address Range 0x4930\_A000 - 0x4930\_AFFF
- Secure Address Range 0x5930\_A000 - 0x5930\_AFFF

The following table shows the memory map of the colour LCD registers in address offset order from the base memory address.

**Table 3-3: LCD and data registers**

Offset	Register Name	Type	Reset	Description
0x000	CHAR_COM	RW	0x0	Write command: A write to this address causes a write to the LCD command register. Read busy status: A read from this address causes a read from the LCD busy register.
0x004	CHAR_DAT	RW	0x0	Write data RAM. A write to this address causes a write to the LCD data register. Read data RAM. A read from this address causes a read from the LCD data register.
0x008	CHAR_RD	RAZ/WI	0x0	Bits[31:8]: Reserved.
		RO	0x0	Bits[7:0]: Captured data from an earlier read command. These bits contain the data from the last request read. Valid only when CHAR_RAW[0] is set.
0x00C	CHAR_RAW	RAZ/WI	0x0	Bits[31:1]: Reserved
		RW	0x0	Bit[0]: Access complete status. 1 indicates Access Complete and data in CHAR_RD is valid. Write 0 to clear access complete flag.
0x010	CHAR_MASK	RAZ/WI	0x0	Bits[31:1]: Reserved
		WO	0x0	Bit[0]: Write interrupt mask. Write 0 to enable Access Complete to generate an interrupt.

Offset	Register Name	Type	Reset	Description
0x014	CHAR_STAT	RAZ/WI	0x0	Bits[31:1]: Reserved
		RO	0x0	Bit[0]: is the state of Access Complete ANDed with the CHAR_MASK.
0x04C	CHAR_MISC	RAZ/WI	0x0	Bits[31:7]: Reserved Bits[2]: Reserved
		RW	0b1 0b1 0b1 0b1 0b1 0b1	CLCD control lines: Bit[6]: CLCD_BL Bit[5]: CLCD_RD Bit[4]: CLCD_RS Bit[3]: CLCD_RESET Bit[1]: CLCD_WR Bit[0]: CLCD_CS



## 3.12. Ethernet

The AN555 FPGA design connects to a Microchip LAN9220 device through a static memory interface.

The base memory addresses of the Ethernet static memory interface are:

- Non-Secure Address Range 0x4140\_0000 - 0x414F\_FFFF
- Secure Address Range 0x5140\_0000 - 0x514F\_FFFF

The selftest software includes example code for an internal loopback operation.

## 3.13. USB

The AN555 FPGA design connects to a Hi-Speed USB controller (ISP1763) device through a static memory interface.

The base memory addresses of the USB static memory interface are:

- Non-Secure Address Range 0x4150\_0000 - 0x415F\_FFFF
- Secure Address Range 0x5150\_0000 - 0x515F\_FFFF

The selftest software includes example code for an internal loopback operation.

## 3.14. Real Time Clock

The AN555 FPGA design uses the PL031 PrimeCell Real Time Clock Controller (RTC). A 1Hz counter in the RTC enables it to be used as a basic alarm function or long time-based counter.

The base memory addresses of the Real Time Clock controller are:

- Non-Secure Address Range 0x4930\_B000 - 0x4930\_BFFF
- Secure Address Range 0x5930\_B000 - 0x5930\_BFFF

The I<sup>2</sup>S interface supports transfer of digital audio to and from the Audio codec. The base memory addresses of the I<sup>2</sup>S audio interface are:

- The following table shows the I<sup>2</sup>S audio registers in address offset order from the base memory address.

Offset	Register Name	Type	Reset	Description
0x000	CONTROL	RAZ/WI	0x0	Bits[31:18], Bit[15], Bit[11], Bits[7:4]: Reserved.
		RW	0b0	Control lines:
			0b0	Bit[17]: Audio codec reset control (output pin)
			0b10	Bit[16]: FIFO reset
			0b10	Bits[14:12]: Rx Buffer IRQ Water Level Default 0b10, IRQ triggers when there is less than a two-word space available.
			0b10	Bits[10:8]: Tx Buffer IRQ Water Level Default 0b10, IRQ triggers when there is more than a two-word space available.
			0b0	Bit[3]: Rx Interrupt Enable
			0b0	Bit[2]: Rx Enable
			0b0	Bit[1]: Tx Interrupt Enable
			0b0	Bit[0]: Tx Enable
0x004	STATUS	RAZ/WI	0x0	Bits[31:6]: Reserved
		RO	0b0	Status Register
			0b1	Bit[5]: Rx Buffer Full
			0b0	Bit[4]: Rx Buffer Empty
			0b1	Bit[3]: Tx Buffer Full
			0b1	Bit[2]: Tx Buffer Empty
			0b0	Bit[1]: Rx Buffer Alert (Depends on Water level)
			0b1	Bit[0]: Tx Buffer Alert (Depends on Water level)
0x008	ERROR	RAZ/WI	0x0	Bits[31:2]: Reserved
		R/W1C	0b0	Error Status Register
			0b0	Bit[1]: Rx overrun. Set this bit to clear. Bit[0]: Tx overrun or underrun. Set this bit to clear.
0x00C	DIVIDE	RAZ/WI	0x0	Bits[31:10]: Reserved
		RW	0x20	Clock Divide Ratio Register (for left or right clock) Bits[9:0]: LRDIV (Left/Right).
0x010	TXBUF	W	0x0	Transmit Buffer FIFO Data Register. This is a write-only register. Bits[31:16]: Left channel
		W	0x0	Bits[15:0]: Right channel

Offset	Register Name	Type	Reset	Description
0x014	RXBUF	RO	0x0	Receive Buffer FIFO Data Register. Bits[31:16]: Left channel
		RO	0x0	Bits[15:0]: Right channel
0x018-0x2FF	RESERVED	RAZ/WI	0x0	Reserved
0x300	ITCR	RAZ/WI	0x0	Bits[31:1]: Reserved
		RW	0b0	Integration Test Control Register Bit[0]: ITCR
0x304	ITIP1	RAZ/WI	0x0	Bits[31:1]: Reserved
		RO	0b0	Integration Test Input Register 1 Bit[0]: SDIN
0x308	ITOP1	RAZ/WI	0x0	Bits[31:4]: Reserved
		RW	0b0 0b0 0b0 0b0	Integration Test Output Register 1 Bit[3]: IRQOUT Bit[2]: LRCK Bit[1]: SCLK Bit[0]: SDOUT

## 3.16. Audio Configuration

The AN555 FPGA model implements a simple SBCon interface based on I<sup>2</sup>C. It configures the Cirrus Logic Low Power Codec with Class D Speaker Driver, CS42L52 part on the MPS3 board.

The base memory addresses are:

- Non-Secure Address Range 0x4920\_1000 - 0x4920\_1FFF
- Secure Address Range 0x5920\_1000 - 0x5920\_1FFF

## 3.17. FPGA system control and I/O

The AN555 implements an FPGA system and I/O control block. The base memory addresses of the control block are:

- Non-Secure Address Range                      0x4930\_2000 - 0x4930\_2FFF
- Secure Address Range                          0x5930\_2000 - 0x5930\_2FFF

The following table shows the FPGA system and I/O control registers in address offset order from the base memory address.

**Table 3-5: System control and I/O registers**

Offset	Register Name	Type	Reset	Description
0x000	LEDO	RAZ/WI	0x0	LED connections Bits[31:10]: Reserved
		RW	0x0	Bits[9:0]: LED
0x004	M85DBGCTRL	RAZ/WI	0x0	Bits[31:4]: Reserved
		RW	0b1	Cortex-M85 control signals Bit[3]: SPNIDEN
		RW	0b1	Bit[2]: SPIDEN
		RW	0b1	Bit[1]: NIDEN
		RW	0b1	Bit[0]: DBGEN
0x008	BUTTON	RAZ/WI	0x0	Bits[31:2]: Reserved
		R	2b0	Buttons Bits[1:0]: Buttons
0x00C	GPIOALT2 <sup>1</sup>	RAZ/WI	0x0	GPIO Alt Function 2 select: Bits[31:16]: Reserved
		RW	0b0	Bit[15]: PMOD1 IO2
		RW	0b0	Bit[14]: PMOD1 IO3
		RW	0b0	Bit[13]: PMOD0 SCK
		RW	0b0	Bit[12]: PMOD0 IO1
		RW	0b0	Bit[11]: PMOD0 IO0
		RW	0b0	Bit[10]: PMOD0 SS
		RW	0b0	Bit[9]: PMOD1 SCK
		RW	0b0	Bit[8]: PMOD0 IO3
		RW	0b0	Bit[7]: PMOD0 IO2
		RAZ/WI	0x0	Bits[6:4]: Reserved
		RW	0b0	Bit[3]: PMOD1 SS
		RAZ/WI	0x0	Bit[2]: Reserved
		RW	0b0	Bit[1]: PMOD1 IO0
		RW	0b0	Bit[0]: PMOD1 IO1
0x010	CLK1HZ	RW	0x0	Bits[31:0]: 1Hz up counter
0x014	CLK100HZ	RW	0x0	Bits[31:0]: 100Hz up counter

Offset	Register Name	Type	Reset	Description
0x018	COUNTER	RW	0x0	Bits[31:0]: Cycle Up Counter - Increments when 32-bit prescale counter equals zero and automatically reloads.
0x01C	PRESCALE	RW	0x0	Prescale Reload Value Bits[31:0]: Reload value for prescale counter.
0x020	PSCNTR	RW	0x0	Prescale Counter Value Bits[31:0]: Current value of the prescale counter. The prescale counter is reloaded with PRESCALE after reaching 0.
0x024	RESERVED	RAZ/WI	0x0	Reserved
0x028	SWITCH	RAZ/WI	0x0	Switches Bits[31:8]: Reserved
		R	0x0	Bits[7:0]: Switches
0x04C	MISC	RAZ/WI	0x0	Bits[31:3]: Reserved
		RW	0b1	Miscellaneous control bits Bit[2]: SHIELD1_SPI_nCS
		RW	0b1	Bit[1]: SHIELD0_SPI_nCS
		RW	0b1	Bit[0]: ADC_SPI_nCS



All counters are driven from FPGA generated clock **PERIF\_CLK**. For details of the clocks please see [Clocks generated within the FPGA](#).

<sup>1</sup> Further detail about the GPIOALT2 usage can be found in [Shield Support](#).

## 3.18. FPGA Serial Configuration Controller (SCC)

The AN555 FPGA implements a communication channel between the MCC and the FPGA system through a Serial Configuration Controller (SCC) interface.

The Cortex-M85 Processor can access the SCC registers through the following memory mapped addresses:

- Non-Secure Address Range                      0x4930\_0000 - 0x4930\_0FFF
- Secure Address Range                            0x5930\_0000 - 0x5930\_0FFF

The following table shows the SCC registers in address offset order from the base memory address.



The read-addresses and write-addresses of the SCC interface do not use bits [1:0]. All address words are word-aligned.

**Table 3-6: SCC registers**

Offset	Register Name	Type	Reset	Description
0x000	CFG_REG0	RAZ/WI	0x0	Bits[31:2]: Reserved
		RW	1b1	Bit[1]: CPU_WAIT control bit. 1 - CPU held in wait state, 0 - CPU is released from wait state.
		RAZ/WI	0b0	Bit[0]: Reserved
0x004	CFG_REG1	RAZ/WI	0x0	Bits[31:2]: Reserved
0x008	CFG_REG2 <sup>1</sup>	RAZ/WI	0x0	Bits[31:3]: Reserved
		RW	0x0	Bit[2]: PMOD1 Read / Write select signal Bit[1]: PMOD0 Read / Write select signal Bit[0]: QSPI Read / Write select signal Read = 0, Write = 1
0x00C	CFG_REG3	RAZ/WI	0x0	Bits[31:0]: Reserved
0x010	CFG_REG4	RAZ/WI	0x0	Bits[31:4]: Reserved
		RW	0x0	Bits[3:0]: Board Revision [r]
0x014	CFG_REG5	RW	0x0	Bits[31:0]: ACLK Frequency in Hz
0x018	CFG_REG6	RAZ/WI	0x0	Bits[31:2]: Reserved
		R	0x0	Bit[1]: PMOD1 IN (defines if PMOD1 present) <sup>1</sup> Bit[0]: PMOD0 IN (defines if PMOD0 present) <sup>1</sup>
0x01C – 0x09C	RESERVED	RAZ/WI	0x0	-
0x0A0	SYS_CFGDATA_RTN	R		Bits[31:0]: DATA R
0x0A4	SYS_CFGDATA_OUT	W		Bits[31:0]: DATA W

Offset	Register Name	Type	Reset	Description
0x0A8	SYS_CFGCTRL	R	1b0	Bit[31]: Start (generates interrupt on write to this bit)
		RW	1b0	Bit[30]: RW access
		RAZ/WI	0x0	Bits[29:26]: Reserved
		RW	0x0	Bits[25:20]: Function value
		RAZ/WI	0x0	Bits[19:12]: Reserved
		RW	0x0	Bits[11:0]: Device (value of 0/1/2 for supported clocks)
0x0AC	SYS_CFGSTAT	RAZ/WI		Bits[31:2]: Reserved
		R		Bit[1]: Error Bit[0]: Complete
0x0B0 – 0xFF4	RESERVED	RAZ/WI		
0xFF8	SCC_AID	R	0x02	SCC AID register is read only Bits[31:24]: FPGA build number, (decimal)
		R	0x1	Bits[23:20]: MPS3 target board revision (A = 0, B = 1, C = 2)
		RAZ/WI	0x7	Bits[19:8]: Reserved
		R	0x8	Bits[7:0]: Number of SCC configuration register
0xFFC	SCC_ID	R	0x41	SCC ID register is read only Bits[31:24]: Implementer ID: 0x41 = Arm
		RAZ/WI	0x0	Bits[23:20]: Reserved
		R	0x5	Bits[19:16]: IP Architecture: 0x5 = AXI
		R	0x555	Bits[15:4]: Primary part number in Binary Coded Decimal, (BCD): 0x555
		RAZ/WI	0x1	Bits[3:0]: Reserved



<sup>1</sup> - Details of the Pmod configuration are provided in [QSPI Flash](#).

## 3.19. DMA

The AN555 FPGA implements 4 single Manager DMA controllers (PL081). Each DMA is connected to a separate Subordinate interface of the subsystem through an MSC.

The following details the address mapping and access restrictions of each of the four DMA's that are implemented.

- DMA0

The base memory addresses are:

- Non-Secure Address Range                      0x4120\_0000 - 0x4120\_0FFF
- Secure Address Range                              0x5120\_0000 - 0x5120\_0FFF

DMA0 is connected to the **HSLVEXPMI1** interface of the subsystem and therefore can access most of the memory mapped regions in the system, except for those that are private to the processor.

- DMA1

The base memory addresses are:

- Non-Secure Address Range                      0x4120\_1000 - 0x4120\_1FFF
- Secure Address Range                              0x5120\_1000 - 0x5120\_1FFF

DMA1 is connected to the **HSLVEXPPILL** interface of the subsystem and therefore can access the following peripheral memory mapped regions except for regions which are private to the processor.

- Non-Secure Address Range                      0x4000\_0000 - 0x47FF\_FFFF
- Secure Address Range                              0x5000\_0000 - 0x57FF\_FFFF
- Non-Secure Address Range                      0xE010\_0000 - 0xEFFF\_FFFF
- Secure Address Range                              0xF010\_0000 - 0xFFFF\_FFFF

- DMA2

The base memory addresses are:

- Non-Secure Address Range                      0x4120\_2000 - 0x4120\_2FFF
- Secure Address Range                              0x5120\_2000 - 0x5120\_2FFF

DMA2 is connected to the **HSLVEXPIHL** interface of the subsystem and therefore can access the following peripheral memory mapped regions.

- Non-Secure Address Range                      0x4800\_0000 - 0x4FFF\_FFFF
- Secure Address Range                              0x5800\_0000 - 0x5FFF\_FFFF



- DMA3

The base memory addresses are:

- Non-Secure Address Range                      0x4120\_3000- 0x4120\_3FFF
- Secure Address Range                              0x5120\_3000 - 0x5120\_3FFF

DMA1 is connected to the **XSLVTCM** interface of the subsystem and therefore can only access the TCM's that are internal to the processor. These are access through the following addresses:

ITCM

- Non-Secure Address Range                      0x0A00\_0000 - 0x0A00\_7FFF
- Secure Address Range                              0x1A00\_0000 - 0x1A00\_7FFF

DTCM

- Non-Secure Address Range                      0x2000\_0000 - 0x2000\_7FFF
- Secure Address Range                              0x3000\_0000 - 0x3000\_7FFF



For further information please refer to *PrimeCell® Single Master DMA Controller (PL081) Technical Reference Manual*

---

## 4. Clock architecture

The following sections describe:

- Input clocks from the MPS3 board to the FPGA
- Clocks generated internally within the FPGA

### 4.1. Source clocks

The following table shows clocks generated on the MPS3 board which are input clocks to the FPGA:

**Table 4-1 : Source clocks**

AN555 clock	Input pin	Frequency	Note
REFCLK24MHZ	OSCCLK[0]	24MHz	24MHz reference
ACLK	OSCCLK[1]	25MHz	Programmable oscillator
MCLK	OSCCLK[2]	30MHz	Programmable oscillator
GPUCLK	OSCCLK[3]	50MHz	Programmable oscillator
AUDCLK	OSCCLK[4]	24.576MHz	Programmable oscillator
HDLCDCLK	OSCCLK[5]	23.75MHz	Programmable oscillator
DBGCLK	CS_TCK	Variable	JTAG input. Frequency set by debugger
CFGCLK	CFG_CLK	Variable	SCC register clock. Frequency set by MCC
DDR4_REF_CLK	c0_sys_clk_p/n	100MHz	Differential input clock to DDR4 controller
SMBM_CLK	SMBM_CLK	24MHz	SMB clock. Frequency defined by MCC



Note

The frequency of ACLK has been reduced to 25MHz from 32MHz as seen in other Applications Notes on MPS3

## 4.2. Clocks generated within the FPGA

The following table shows clocks generated within the FPGA from source clocks inputs from the MPS3 board.

**Table 4-2: Clocks generated within FPGA**

FPGA generated clock	MPS3 source clock	Frequency	Note
MAINCLK	OSCCLK[1]	25MHz	Clock source for SSE-310 and all non- APB peripherals in the design
FASTCLK	OSCCLK[1]	50MHz	Fast Clock source for SSE-310 required due to the clock structure of the SSE-310 which allows different CPU clock speeds
EXTCLK	OSCCLK[1]	30MHz	Used for QSPI Pmod controllers
PERIF_CLK	OSCCLK[3]	25MHz	Clock source for APB peripherals
AUDMCLK	AUDCLK	12.288MHz	-
AUDSCLK	AUDCLK	3.072MHz	-
SDMCLK	REFCLK24MHZ	24MHz	-
CLK32KHZ	REFCLK24MHZ	32kHz	-
CLK100HZ	REFCLK24MHZ	100Hz	-
CLK1HZ	REFCLK24MHZ	1Hz	-
CFGCLK	CFG_CLK	Set by MCC	SCC register clock from MCC

## 4.3. SSE-310 Clocks

The following table shows clocks within FPGA which are inputs to SSE-310 Subsystem.

**Table 4-3: SSE-310 clocks**

SSE-310 clock input	FPGA generated clock	Frequency	Note
SYSCLK	MAINCLK	25MHz	Main System clock
CPU0CLK	MAINCLK	25MHz	CPU clock
AONCLK	MAINCLK	25MHz	Always On clock
CNTCLK	MAINCLK	25MHz	Counter clock
SLOWCLK	CLK32KHZ	32KHz	Slow clock
SYSPLLCLK	FASTCLK	50MHz	Fast System clock

## 5. Reset architecture

### 5.1. FPGA Resets

The following tables list the top-level reset signals based on their direction to the FPGA.

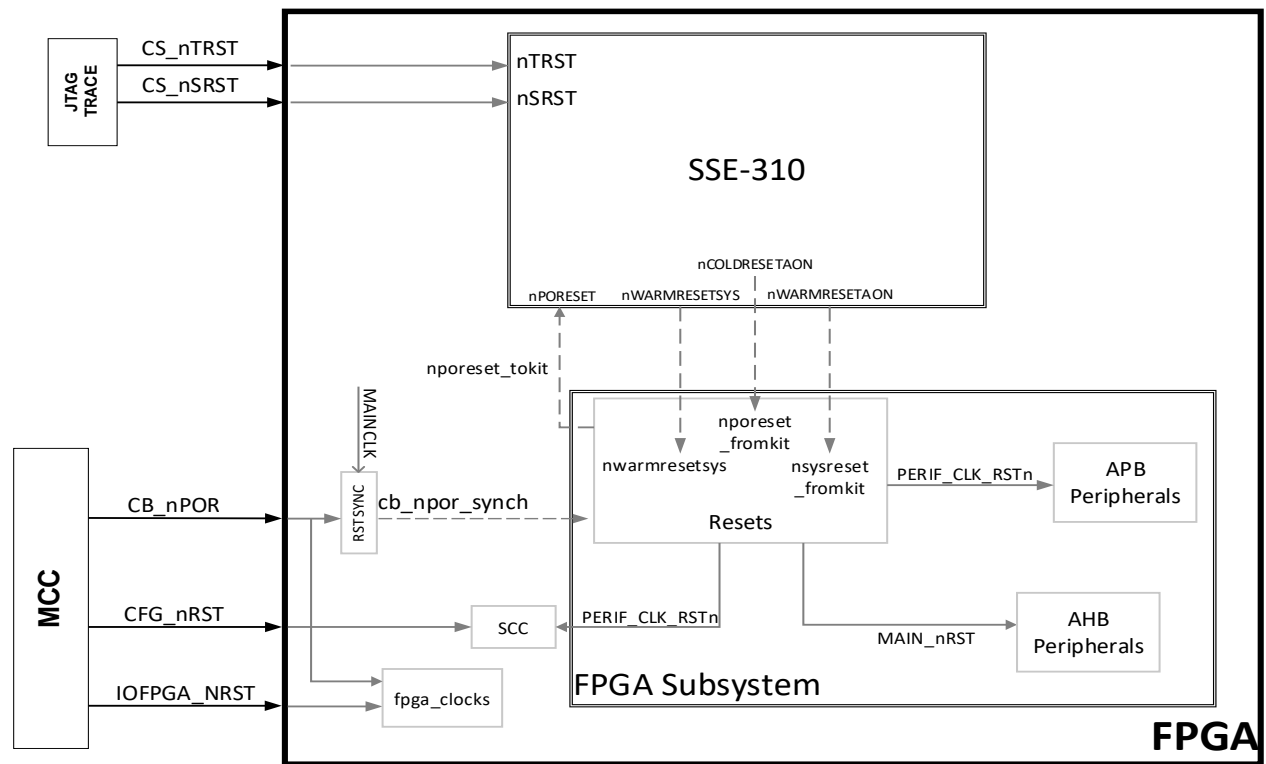
**Table 5-1: FPGA Input Resets List**

Input Source Reset	Note	SSE-310 Example Design Destination Reset	Reset control
CB_nPOR	Power-on Reset Reset signal to all FPGA	nPORESET	Embedded in MCC firmware
IOFPGA_NRST	Auxiliary reset signal Connected to Audio clock generator	Not connected	
CFG_nRST	SCC Register block Reset	Not connected	
CS_nTRST	JTAG Reset	nTRST	Debugger, please refer to debugger user manual
CS_nSRST	Warm SSE-310 Subsystem reset	nSRST	

**Table 5-2: FPGA Subsystem Resets List**

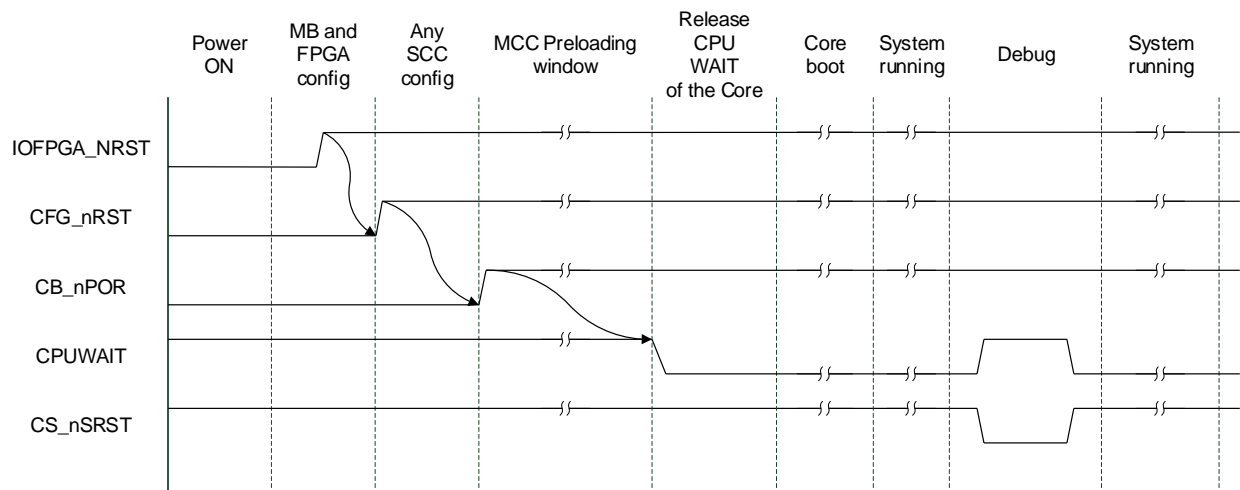
Output Source Reset	Note
MAIN_nRST	AHB Peripherals Reset
PERIPH_CLK_RSTn	APB Peripherals Reset

Figure 5-1 : FPGA Resets



The following timing diagram shows a typical reset de-assertion sequence after power on.

Figure 5-2 : Reset Timing



CPUWAIT is controlled by SCC CFG\_REG0, see [FPGA Serial Configuration Controller \(SCC\)](#)

## 6. FPGA Secure Privilege Control

The SSE-310 Subsystem Secure Privilege and Non-Secure Privilege control block provides expansion security control signals to control the security gating units outside the Subsystem.

The following table lists the connectivity of the system security extension signals.

**Table 6-1: Security expansion signals connectivity.**

Component name	Components signal	Security expansion signal
USER MSC	msc_irq	SMSCEXPSTATUS[4:0]
	msc_irq_clear	SMSCEXP CLEAR[4:0]
	cfg_nonsec	NSMSCEXP[4:0]
APB PPC EXP 0	apb_ppc_irq	SPERIPHPPCEXPSTATUS[0]
	apb_ppc_clear	SPERIPHPPCEXP CLEAR[0]
	cfg_sec_resp	SECRES PCFG
	cfg_non_sec	PERIPHNSPPCEXP0[15:0]
	cfg_ap	PERIPHPPPCEXP0[15:0]
APB PPC EXP 1	apb_ppc_irq	SPERIPHPPCEXPSTATUS[1]
	apb_ppc_clear	SPERIPHPPCEXP CLEAR[1]
	cfg_sec_resp	SECRES PCFG
	cfg_non_sec	PERIPHNSPPCEXP1[15:0]
	cfg_ap	PERIPHPPPCEXP1[15:0]
APB PPC EXP 2	apb_ppc_irq	SPERIPHPPCEXPSTATUS[2]
	apb_ppc_clear	SPERIPHPPCEXP CLEAR[2]
	cfg_sec_resp	SECRES PCFG
	cfg_non_sec	PERIPHNSPPCEXP2[15:0]
	cfg_ap	PERIPHPPPCEXP2[15:0]
AHB PPC EXP 0	ahb_ppc_irq	SMAINPPCEXPSTATUS[0]
	ahb_ppc_clear	SMAINPPCEXP CLEAR[0]
	cfg_sec_resp	SECRES PCFG
	cfg_non_sec	MAINNSPPCEXP0[15:0]
	cfg_ap	MAINPPPCEXP0[15:0]

Component name	Components signal	Security expansion signal
AHB PPC EXP 1	ahb_ppc_irq	SMAINPPCEXPSTATUS[1]
	ahb_ppc_clear	SMAINPPCEXPCLR[1]
	cfg_sec_resp	SECRESPCFG
	cfg_non_sec	MAINNSPPCEXP1[15:0]
	cfg_ap	MAINPPPCEXP1[15:0]
MPC FPGA SRAM	secure_error_irq	SMPCEXPSTATUS[2]

The following table lists the peripherals that are controlled by the SMSCEXP port.

Each MSC interface is controlled by **SMSCEXPSTATUS[n]** and **SMSCEXPCLR[n]**.

**Table 6-2: Peripheral mapping of AHB MSC EXP**

AHB user MSC interface number <n>	Name
0	DMA 0
1	DMA 1
2	DMA 2
3	DMA 3
4	Memory Preload
15:3	Reserved

The following table lists the peripherals that are controlled by PERIPHERAL PPC EXP 0.

Each APB <n> interface is controlled by **PERIPHNSPPCEXP0[n]** and **PERIPHPPPCEXP0[n]**.

**Table 6-3: Peripheral mapping of APB PPC EXP 0**

APB PPC EXP 0 interface number <n>	Name
0	Timing Adapter APB 0 – FPGA SRAM
1	Timing Adapter APB 1 - QSPI
2	Timing Adapter APB 2 – DDR4
15:3	Reserved

The following table lists the peripherals that are controlled by APB PPC EXP 1.

Each APB <n> interface is controlled by PERIPHNSPPCEXP1[n] and PERIPHPPPCEXP1[n].

**Table 6-4: Peripheral mapping of APB PPC EXP 1**

APB PPC EXP 1 interface number <n>	Name
0	FPGA - SBCon I2C (Touch)
1	FPGA - SBCon I2C (Audio Conf)
2	FPGA - PL022 (SPI ADC)
3	FPGA - PL022 (SPI Shield 0)
4	FPGA - PL022 (SPI Shield 1)
5	SBCon (I2C – Shield 0)
6	SBCon (I2C – Shield 1)
7	Reserved
8	FPGA - SBCon I2C (DDR4 EEPROM)
15:9	Reserved

The following table lists the peripherals that are controlled by APB PPC EXP 2.

Each APB <n> interface is controlled by PERIPHNSPPCEXP2[n] and PERIPHPPPCEXP2[n].

**Table 6-5: Peripheral mapping of APB PPC EXP 2**

APB PPC EXP 2 interface number <n>	Name
0	FPGA - SCC registers
1	FPGA - I2S (Audio)
2	FPGA – I/O (System Ctrl + I/O)
3	UART0 - FPGA_UART0
4	UART1 - FPGA_UART1
5	UART2 - FPGA_UART2
6	UART3 - UART Shield 0
7	UART4 - UART Shield 1
8	UART5 - FPGA_UART3
9	Reserved
10	CLCD Configuration register
11	RTC
15:12	Reserved



The following table lists the peripherals that are controlled by AHB PPC EXP 0.

Each APB <n> interface is controlled by MAINNSPPCEXP0[n] and MAINPPPCEXP0[n].

**Table 6-6: Peripheral mapping of AHB PPC EXP 0**

AHB PPC EXP 0 interface number <n>	Name
0	GPIO 0
1	GPIO 1
2	GPIO 2
3	GPIO 3
7:4	Reserved
8	Ethernet and USB
12:9	Reserved
13	Onboard QSPI Read/Write Controller
14	PMOD0 QSPI Read/Write Controller
15	PMOD1 QSPI Read/Write Controller

The following table lists the peripherals that are controlled by AHB PPC EXP 1.

Each APB <n> interface is controlled by MAINNSPPCEXP1[n] and MAINPPPCEXP1[n].

**Table 6-7: Peripheral mapping of AHB PPC EXP 1**

AHB PPC EXP 1 interface number <n>	Name
0	DMA 0
1	DMA 1
2	DMA 2
3	DMA 3
15:4	Reserved

# 7. Interrupts

## 7.1. FPGA interrupt map

The following table shows how Interrupts in the FPGA Subsystem extend the SSE-310 Interrupt map by adding to the expansion area.

**Table 7-1: SSE-310 expansion interrupt map.**

Interrupt input	Interrupt source
IRQ[0]	Non-Secure Watchdog reset Request
IRQ[1]	Non-Secure Watchdog Interrupt
IRQ[2]	SLOWCLK Timer
IRQ[3]	Timer 0
IRQ[4]	Timer 1
IRQ[5]	Timer 2
IRQ[6]	Reserved
IRQ[7]	Reserved
IRQ[8]	Reserved
IRQ[9]	MPC Combined (Secure)
IRQ[10]	PPC Combined (Secure)
IRQ[11]	MSC Combined (Secure)
IRQ[12]	Bridge Error Combined Interrupt (Secure)
IRQ[13]	Reserved
IRQ[14]	PPU Combined (Secure)
IRQ[15]	Reserved
IRQ[16]	NPU0
IRQ[26:17]	Reserved
IRQ[27]	TIMER 3 AON
IRQ[28]	CPU0CTIIRQ0 (local CPU CTI only)
IRQ[29]	CPU0CTIIRQ01 (local CPU CTI only)
IRQ[31:30]	Reserved
IRQ[32]	SYSCOUNTER

**Table 7-2: FPGA expansion interrupt map.**

Interrupt input	Interrupt source
IRQ[33]	UART 0 Receive Interrupt
IRQ[34]	UART 0 Transmit Interrupt
IRQ[35]	UART 1 Receive Interrupt
IRQ[36]	UART 1 Transmit Interrupt
IRQ[37]	UART 2 Receive Interrupt
IRQ[38]	UART 2 Transmit Interrupt
IRQ[39]	UART 3 Receive Interrupt
IRQ[40]	UART 3 Transmit Interrupt
IRQ[41]	UART 4 Receive Interrupt
IRQ[42]	UART 4 Transmit Interrupt
IRQ[43]	UART 0 Combined Interrupt
IRQ[44]	UART 1 Combined Interrupt
IRQ[45]	UART 2 Combined Interrupt
IRQ[46]	UART 3 Combined Interrupt
IRQ[47]	UART 4 Combined Interrupt
IRQ[48]	UART Overflow (0, 1, 2, 3, 4 & 5)
IRQ[49]	Ethernet
IRQ[50]	Audio I <sup>2</sup> S
IRQ[51]	Touch Screen
IRQ[52]	USB
IRQ[53]	SPI ADC
IRQ[54]	SPI (Shield 0)
IRQ[55]	SPI (Shield 1)
IRQ[56]	Reserved
IRQ[57]	DMA 0 Error Interrupt
IRQ[58]	DMA 0 Terminal Count Interrupt
IRQ[59]	DMA 0 Combined Interrupt
IRQ[60]	DMA 1 Error Interrupt
IRQ[61]	DMA 1 Terminal Count Interrupt
IRQ[62]	DMA 1 Combined Interrupt
IRQ[63]	DMA 2 Error Interrupt
IRQ[64]	DMA 2 Terminal Count Interrupt
IRQ[65]	DMA 2 Combined Interrupt
IRQ[66]	DMA 3 Error Interrupt
IRQ[67]	DMA 3 Terminal Count Interrupt

Interrupt input	Interrupt source
IRQ[68]	DMA 3 Combined Interrupt
IRQ[69]	GPIO 0 Combined Interrupt
IRQ[70]	GPIO 1 Combined Interrupt
IRQ[71]	GPIO 2 Combined Interrupt
IRQ[87:72]	GPIO 3 Combined Interrupt
IRQ[103:88]	GPIO 0 individual interrupts
IRQ[119:104]	GPIO 1 individual interrupts
IRQ[123:120]	GPIO 2 individual interrupts
IRQ[124]	GPIO 3 individual interrupts
IRQ[125]	UART 5 Receive Interrupt
IRQ[126]	UART 5 Transmit Interrupt
IRQ[127]	UART 5 Combined Interrupt

## 7.2. UART interrupts

There are six CMSDK UART's in the system, and each UART has the following interrupt pins:

- TXINT
- RXINT
- TXOVRINT
- RXOVRINT
- UARTINT (combined interrupt)

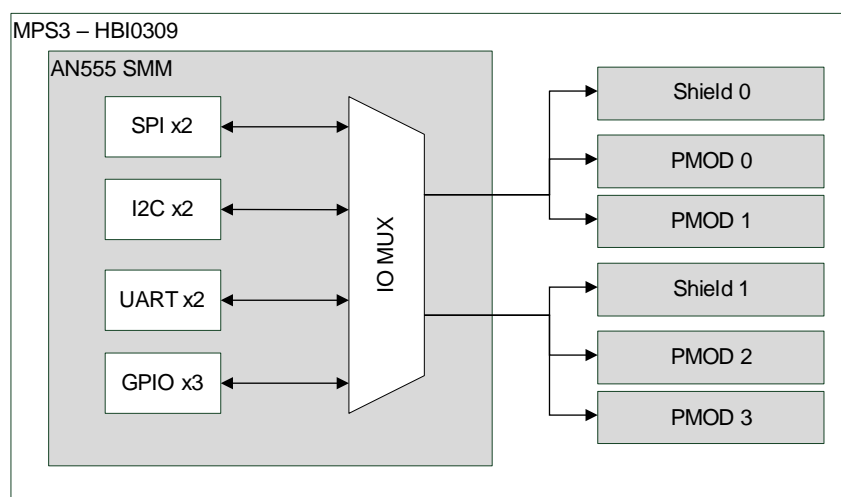
The TXINT, RXINT and UARTINT interrupt signal of each UART drive a separate interrupt input of the Cortex-M85 processor.

In addition, the TXOVRINT and RXOVRINT interrupt signals of all six UART's, twelve signals in total, are logically OR'd together to drive IRQ[48].

## 8 Shield Support

This AN555 FPGA implementation supports external shield devices. To enable the Shield support, two SPI, two UART and two I<sup>2</sup>C interfaces plus secondary alternate function of Pmod QSPI, are multiplexed with GPIO over the Shield Headers.

**Figure 8-1: Shield device expansion**



Pmod 2 & 3 are not used in this design and are here only for reference purposes.

Multiplexing is controlled by the alternative function (ALTF1) output from the associated GPIO Register. A second alternative function (ALTF2) is multiplexed for certain pins of Shield 0, and these are controlled through GPIOALT2 in the FPGAIO Registers at address offset 0x0C.

The following table shows the multiplexed output functions ALTF1 and ALTF2.

**Table 8-1: Shield alternative function pinout**

Shield pin	GPIO	ALTF1	ALTF2
SH0_IO0	GPIO0_0	UART3 RXD - SH0_RXD	PMOD1 IO1
SH0_IO1	GPIO0_1	UART3 TXD - SH0_TXD	PMOD1 IO0
SH0_IO2	GPIO0_2	-	-
SH0_IO3	GPIO0_3	-	PMOD1 SS
SH0_IO4	GPIO0_4	-	-
SH0_IO5	GPIO0_5	-	-
SH0_IO6	GPIO0_6	-	-

Shield pin	GPIO	ALTF1	ALTF2
SH0_IO7	GPIO0_7	-	PMOD0 IO2
SH0_IO8	GPIO0_8	-	PMOD0 IO3
SH0_IO9	GPIO0_9	-	PMOD1 SCK
SH0_IO10	GPIO0_10	SPI3 SS – SH0_nCS	PMOD0 SS
SH0_IO11	GPIO0_11	SPI3 MOSI – SH0_DO	PMOD0 IO0
SH0_IO12	GPIO0_12	SPI3 MISO – SH0_DI	PMOD0 IO1
SH0_IO13	GPIO0_13	SPI3 SCK – SH0_CLK	PMOD0 SCK
SH0_IO14	GPIO0_14	SBCON2 SDA – SH0_SDA	PMOD1 IO3
SH0_IO15	GPIO0_15	SBCON2 SCL – SH0_SCL	PMOD1 IO2
SH1_IO0	GPIO1_0	UART4 RXD – SH1_RXD	-
SH1_IO1	GPIO1_1	UART4 TXD – SH1_TXD	-
SH1_IO2	GPIO1_2	-	-
SH1_IO3	GPIO1_3	-	-
SH1_IO4	GPIO1_4	-	-
SH1_IO5	GPIO1_5	-	-
SH1_IO6	GPIO1_6	-	-
SH1_IO7	GPIO1_7	-	-
SH1_IO8	GPIO1_8	-	-
SH1_IO9	GPIO1_9	-	-
SH1_IO10	GPIO1_10	SPI4 SS – SH1_nCS	-
SH1_IO11	GPIO1_11	SPI4 MOSI – SH1_DO	-
SH1_IO12	GPIO1_12	SPI4 MISO – SH1_DI	-
SH1_IO13	GPIO1_13	SPI4 SCK – SH1_CLK	-
SH1_IO14	GPIO1_14	SBCON3 SDA – SH1_SDA	-
SH1_IO15	GPIO1_15	SBCON3 SCL – SH1_SCL	-



Note

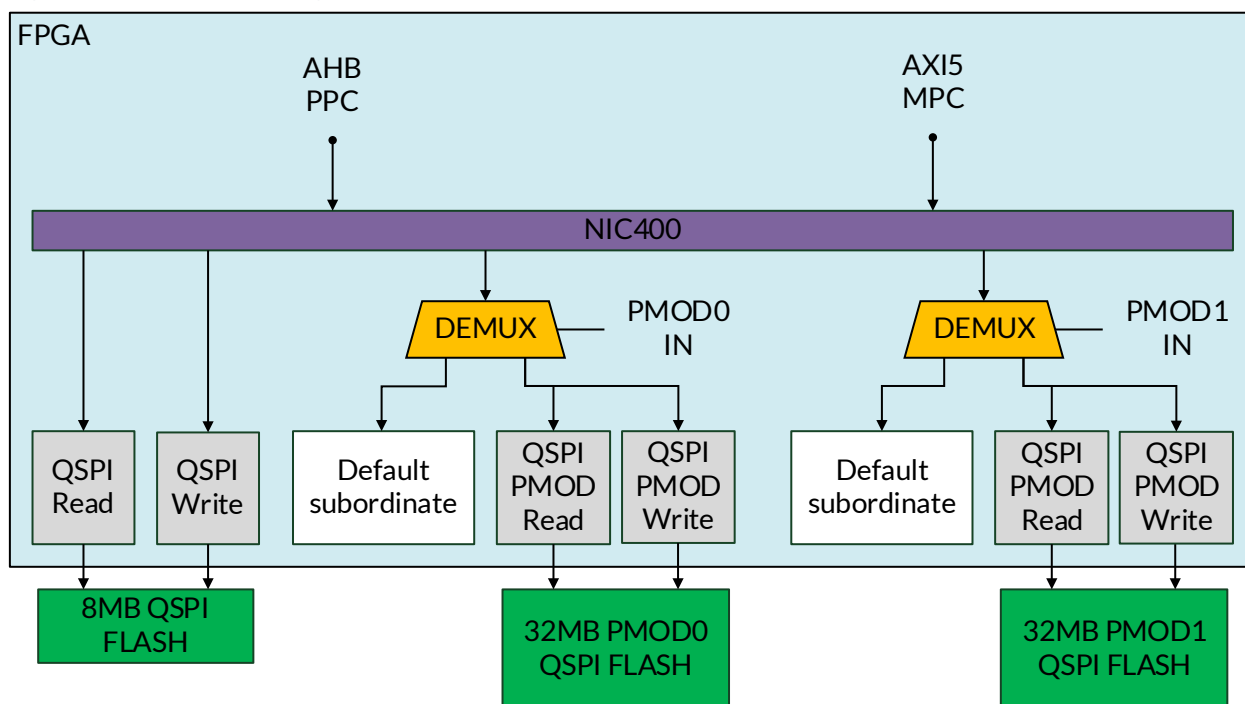
During boot up, ALTF2 is automatically set for Pmod peripherals if Pmod's are enabled, this is to allow preloading of the QSPI Flash. Once CPUWAIT is released, the Shield pins will default to GPIO with both ALTF1 and ALTF2 disabled. Therefore, to use ALTF1/2, software will need to enable each alternate function individually.

## 9. QSPI Flash

This AN555 FPGA implementation supports 3 Quad Serial Peripheral Interface (QSPI) flash memories, 1 onboard Microchip SST26VF064B and 2 optional blocks of QSPI memory that are implemented using external QSPI flash Pmod modules (see [Additional required hardware](#) for further details).

All the memories are accessed over Xilinx AXI QSPI controller interfaces. The Onboard QSPI Flash provides 8MB of memory and PMOD0 and PMOD1 provide 32MB of memory each. The presence of the optional QSPI Flash in the design is controlled by the SCC CFG\_REG6 further described in [Table 3-6](#), these bits are set by the MCC at boot time, depending on the settings described in [Additional required hardware](#).

Figure 9-1: QSPI block diagram



If one or both QSPI Flash memory Pmod modules is absent, the PMODx IN bit will be set to 0 and all transactions will be routed to an AXI default subordinate, which will respond with a bus fault error, this error needs to be handled by software using a fault handler. By default, both Pmod modules will be disabled.

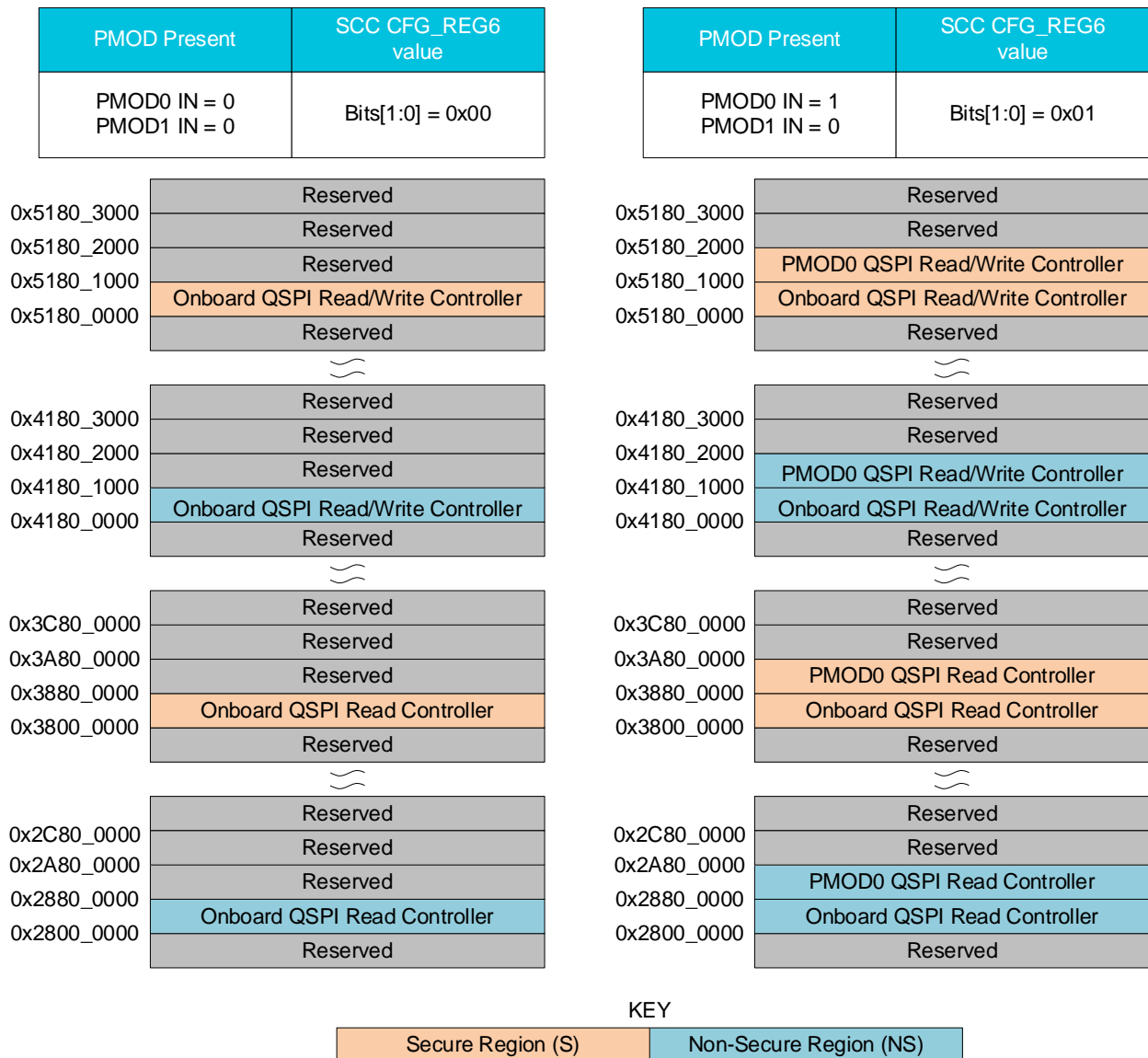
When one or both QSPI Flash memory modules (Pmod's) are absent, the full system memory map shown in [Memory Map Overview](#) changes. The relevant memory map regions for all possible combinations of PMOD0 and PMOD1 are shown in the following tables:

The figures below show the changes to the memory mapping (see [Memory Map Overview](#) for the full mapping), depending on which of the QSPI Pmod modules are installed and enabled.

The first diagram ([Figure 9-2: Memory map with no Pmod's present and with PMOD0 present](#)) covers the situations when:

- No QSPI Pmod's are installed and enabled
- Only QSPI PMOD0 is installed and enabled.

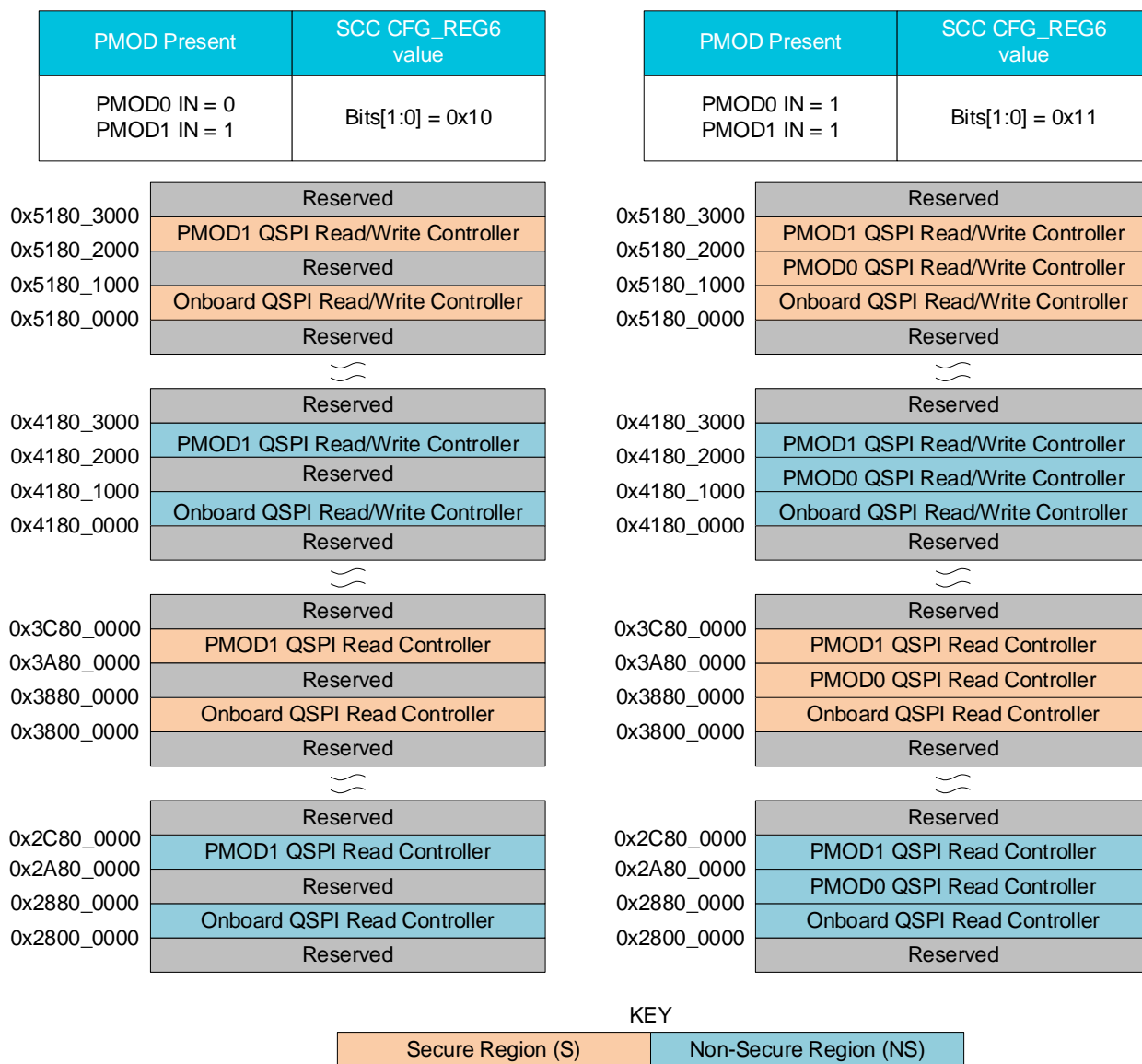
**Figure 9-2: Memory map with no Pmod's present and with PMOD0 present**





- Only QSPI PMOD1 is installed and enabled
- Both QSPI PMOD0 and QSPI PMOD1 are installed and enabled.

**Figure 9-3: Memory map with PMOD1 present and with both PMOD0/1 present**



# 10. Debug Infrastructure

The render of the SSE-310 Subsystem used in this FPGA includes an example debug infrastructure that instances DAP-Lite2, a debug timestamp generator and TPIU-M.

The following subsections describe the example debug infrastructure.

## 10.1. DAP-Lite2

Arm® CoreSight™ DAP-Lite2 enables an off-chip debugger to connect to a target system using a low pin-count JTAG or Serial Wire interface. It is compliant with Arm® Debug Interface Architecture Specification ADIv6.0. DAP-Lite2 which supports AMBA AHB5 debug access interface.

## 10.2. Debug timestamp generator

The debug timestamp generator is implemented as a 64-bit binary up counter.

## 10.3. TPIU-M

The TPIU-M is a Trace Port Interface Unit (TPIU) that is designed for use in single-processor systems based on Arm Cortex-M processors.

The TPIU-M bridges between the on-chip trace data from the Embedded Trace Macrocell (ETM) and the Instrumentation Trace Macrocell (ITM), with separate IDs, to a data stream.

# 11. ZIP Bundle Description

## 11.1. Overall Structure

The accompanying .zip bundle contains:

- AN555 FPGA Document (this document).
- An example Keil® MDK Version 5.37 software project, that can be run on the Cortex-M85 to test supported board peripherals and interfaces.
- `Boardfiles/` directory contains the directory structure and files to be loaded onto the MPS3 SD Card. This is required to configure the MPS3 board to load and run this implementation.

## 11.2. Documentation

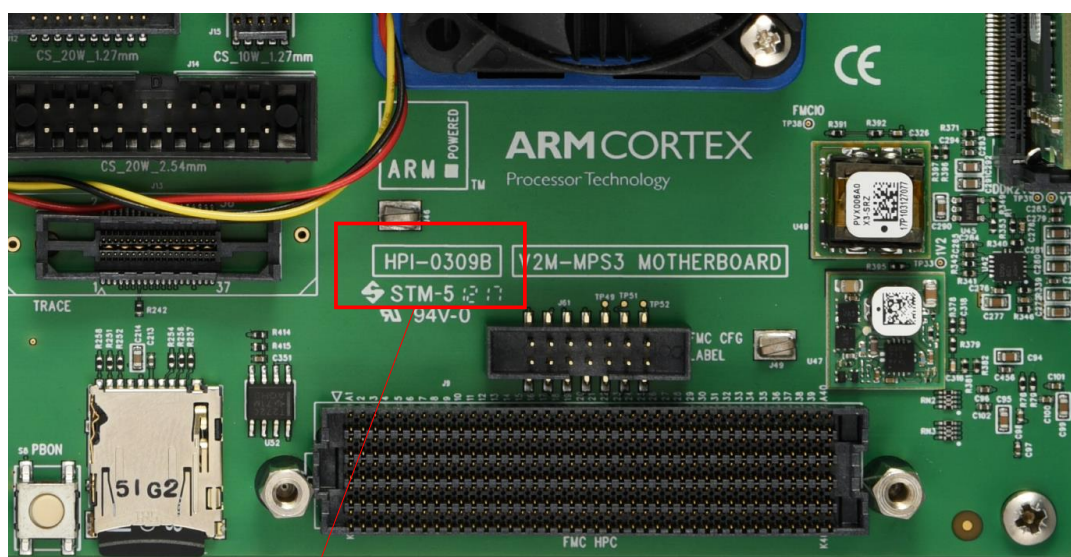
AN555 FPGA Document is in the `Docs/` folder of the bundle.

## 11.3. MPS3 Board Revision and Support

### 11.3.1. Identifying the MPS3 Board Revision

The bundle supports MPS3 board revisions B and C. The board revision, if not known can be identified from the silk screen text, inside a marked box, on the board as shown in the diagram below:

Figure 11-1: MPS3 Board Revision Identifier



Board Part Number and Revision

In this example the part number is “HBI0309B”. The last letter at the end of the part number denotes the board revision. The illustration shows a revision B board.

### 11.3.2. Bundle Support for Specific MPS3 Board Revisions

There are two subdirectories in the `Boardfiles/MB/` directory that correspond to the two supported revisions:

- HBI0309B
- HBI0309C

The contents of each of these directories, within the provided bundle, are identical but the MCC only uses the contents from the directory name that matches the board part number and revision in use. See [Identifying the MPS3 Board Revision](#) for further details on how to identify the board part number and revision.



Only files modified within the directory name that align with the MPS3 board part number and revision are used by the MCC. Care must be taken to ensure the correct directory contents are modified if modifications are required.

## 11.4. Bundle Directory Tree/Structure

The directory structure of the bundle is shown below.

```
`-- AN555_SSE-310_for_MPS3
    |-- Boardfiles
    |-- Docs
    |-- SOFTWARE
    |-- License.pdf
    |-- Release_notes.txt
    |-- revision_history.txt
```

The directory structure of the Boardfiles which need to be loaded onto the sd card is shown below.

```
`-- Boardfiles
    |-- MB
    |   |-- BRD_LOG.TXT
    |   |-- HBI0309B
    |       |-- AN555
    |           |-- an555_v2.bit
    |           |-- an555_v2.txt
    |           |-- images.txt
    |       |-- board.txt
    |       |-- mbb_v220.ebf
    |   |-- HBI0309C
    |       |-- AN555
    |           |-- an555_v2.bit
    |           |-- an555_v2.txt
    |           |-- images.txt
    |       |-- board.txt
    |       |-- mbb_v220.ebf
    |-- SOFTWARE
    |   |-- an555_st.axf
    |-- config.txt
```

# 12. Software and Debug

## 12.1. Overview

It is strongly recommended to use Keil uVision 5.37, which is the minimum edition needed to support the processor in this implementation and is the version supported by AN555 FPGA implementation. Additionally the ARM::V2M\_MPS3\_SSE\_310\_BSP 1.1.0 (2022-09-30) Keil pack must be installed.

An example software project that can be built using Keil uVision 5.37 (targeting the Cortex-M85) which is provided as part of the bundle. The test software uses the UART 1 to implement a menu-driven software interface, controllable from a host terminal emulator. The test software, which executes on the Cortex-M85, offers the following tests:

- Audio
- Display
- Touchscreen
- LEDs/Switches/Buttons
- Ethernet
- USB
- UART
- Memory (FPGA SRAM, Internal SRAM and DDR4)
- Timers
- RTC
- QSPI
- QSPI PMOD0
- QSPI PMOD1

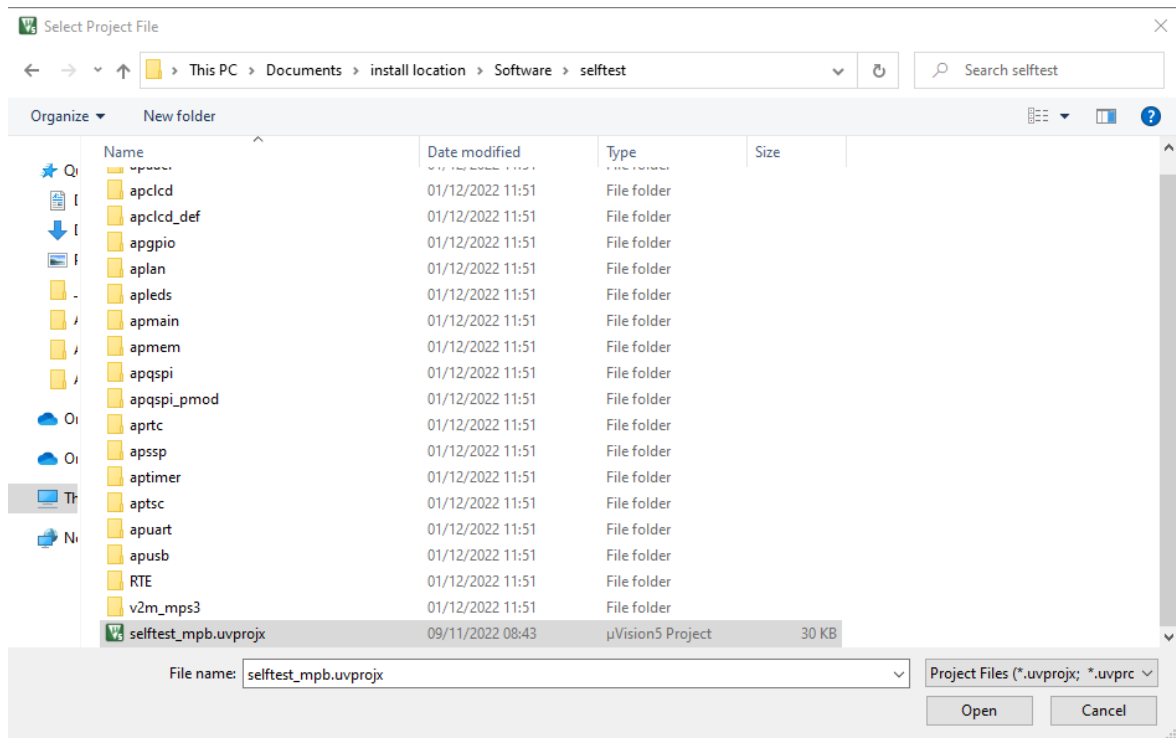
## 12.2. Building the Software

Requirements:

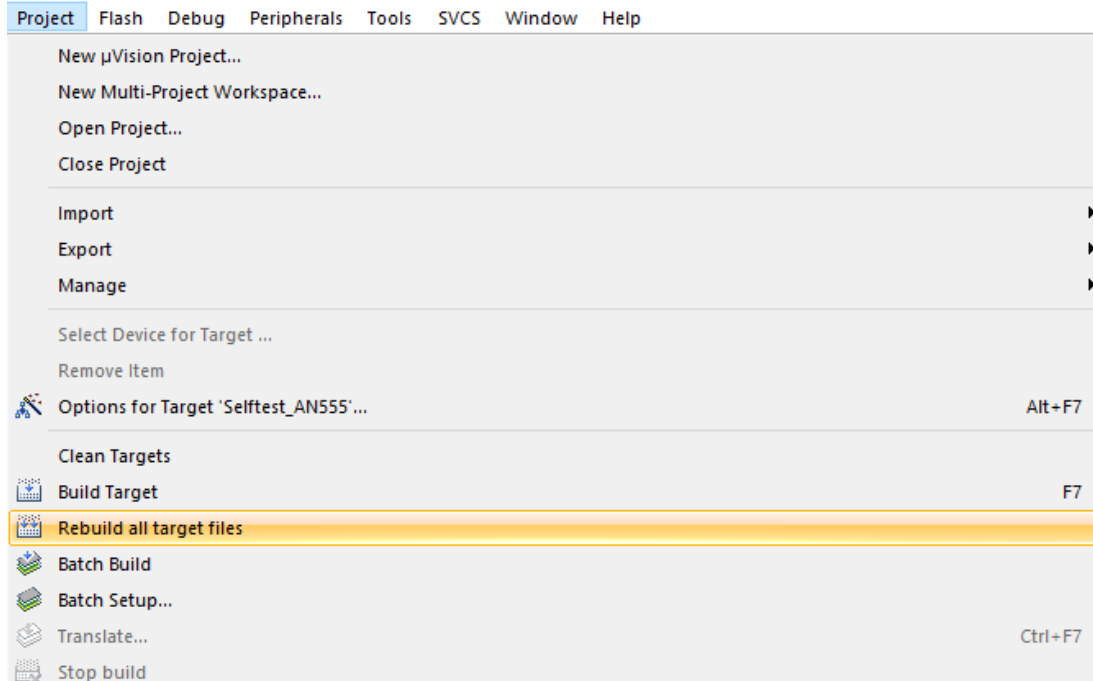
- The software directory from the supplied bundle **SOFTWARE/selftest**.
- Proficiency with Keil uVision 5.37.

Steps:

1. Open Keil uVision.
2. Select **Project > Open Project...**, navigate to the <install location>**Software/selftest/** directory in the bundle.
3. Select **selftest\_mpb.uvprojx** and click **Open**



4. To rebuild the project, **Project > Rebuild all target files.**



5. Ensure you have the Build Output window viewable where you can observe verbose output as the project builds:

```
Build Output
compiling apuart.c...
compiling aprtc.c...
compiling gspl_flash_micron.c...
compiling ETH_MPS3.c...
compiling gspl_test.c...
compiling apqspi.c...
compiling GLCD_MPS3.c...
compiling apmem.c...
compiling dma350_ch_drv.c...
compiling device_definition.c...
compiling system_SSE310MPS3.c...
compiling dma350_address_remap.c...
compiling syscounter_armv8-m_read_drv.c...
compiling dma350_drv.c...
compiling syscounter_armv8-m_cntrl_drv.c...
compiling systimer_armv8-m_drv.c...
compiling startup_SSE310MPS3.c...
compiling systimer_armv8-m_timeout.c...
compiling dma350_lib.c...
linking...
Program Size: Code=36546 RO-data=313770 RW-data=204 ZI-data=51284
FromELF: creating hex file...
After Build - User command #1: move.bat
C:\Users\petcon01\OneDrive - ARM\Documents\install location\Software\selftest>call copy objects\an555_st.axf an555_st.axf
1 file(s) copied.
After Build - User command #2: fromelf --bin --output=an555_st.bin .\objects\an555_st.axf
".\Objects\an555_st.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:03
```

6. The project has now built and there is a newly built .axf file in the SOFTWARE/selftest/ directory, titled an555\_st.axf .



## 12.3. Debug Support

This AN555 FPGA implementation is supported by Arm Development Studio version 2022.1. A platform configuration is achieved using the auto discovery feature of Arm Development Studio. Instructions for using this feature can be found in [Creating a Development Studio Debug Configuration](#), which also gives locations of the debug connectors on the MPS3 board.

### 12.3.1. DSTREAM Debug Hardware

The Arm DSTREAM-PT Debug Probe is required for this AN555 FPGA implementation and all the instructions and guidance in this application note assume the use of the DSTREAM-PT debugger.

### 12.3.2. Debug Connectivity

The following table shows the supported connectivity between the supported MPS3 Board debug connectors and supported debug in the FPGA implementation.

**Table 12-1: Debug Connectivity and Support**

Debug Connector Type	P-JTAG Debug	SWD	4-bit Trace	16-bit Trace
Coresight 20-pin	Cortex-M85	Cortex-M85	Cortex-M85 <sup>1</sup>	-
Mictor 38	Cortex-M85	Cortex-M85	Cortex-M85	Cortex-M85



<sup>1</sup> When using the Coresight 20-pin connector, the DSTREAM-ST streaming trace must be selected as the Trace capture method, when Trace capture is required.

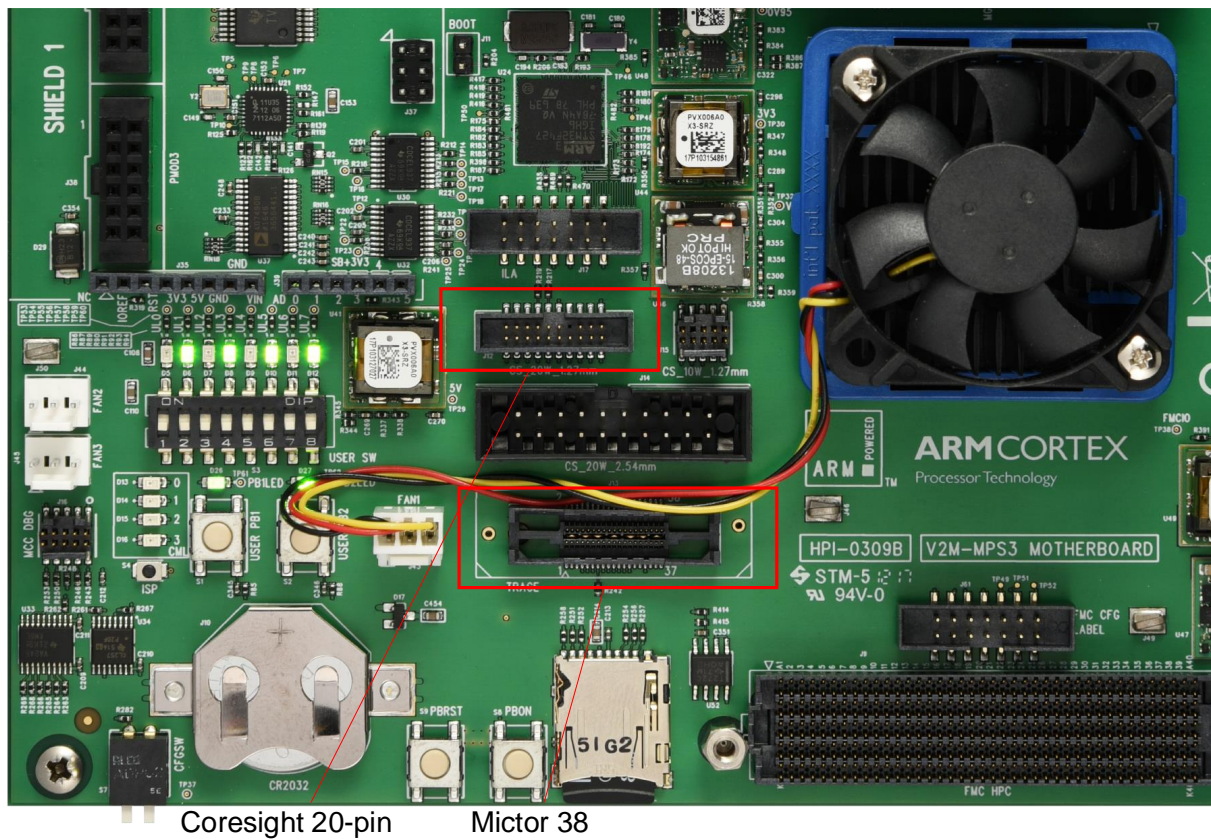
---

### 12.3.3. Creating a Development Studio Debug Configuration

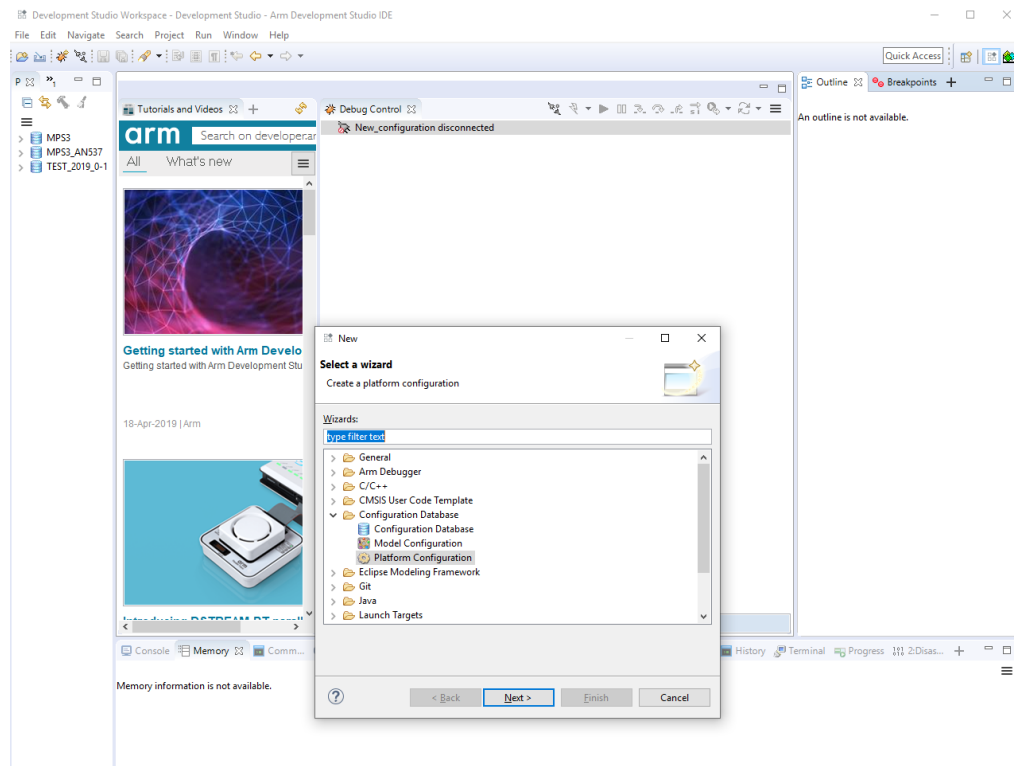
The steps below show the procedure to create a debug configuration:

1. Ensure the DSTREAM debugger is:
  - a. Powered, and connected to the host running the Development Studio software.
  - b. Connected to the MPS3 using the Coresight 20-pin / Mictor 38 connector on the MPS3 as shown below:

Figure 12-1: MPS3 Board Debug Connector Locations

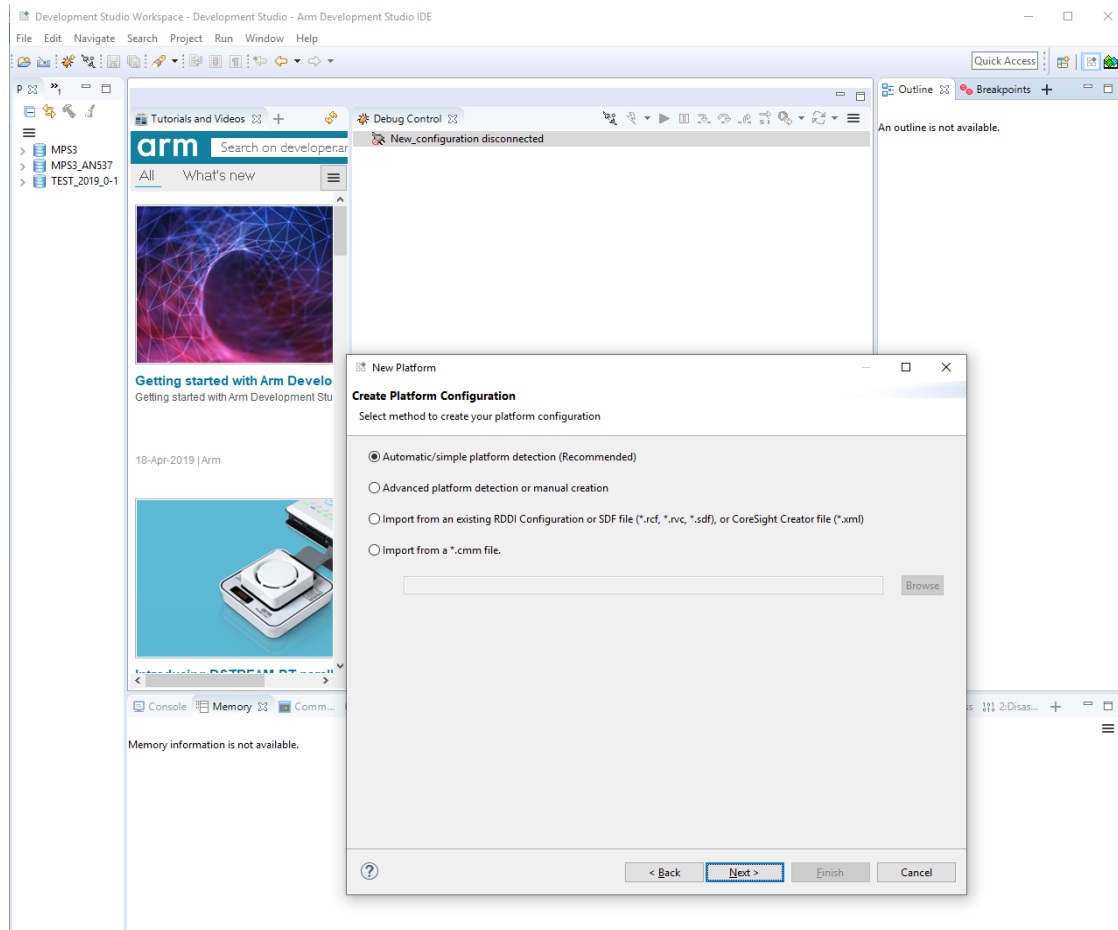


2. Open the Development Studio application on the host machine.
3. Press either Ctrl+N, or from the Menu select **File > New > Other**, to open the wizard to create a Platform Configuration as shown below.

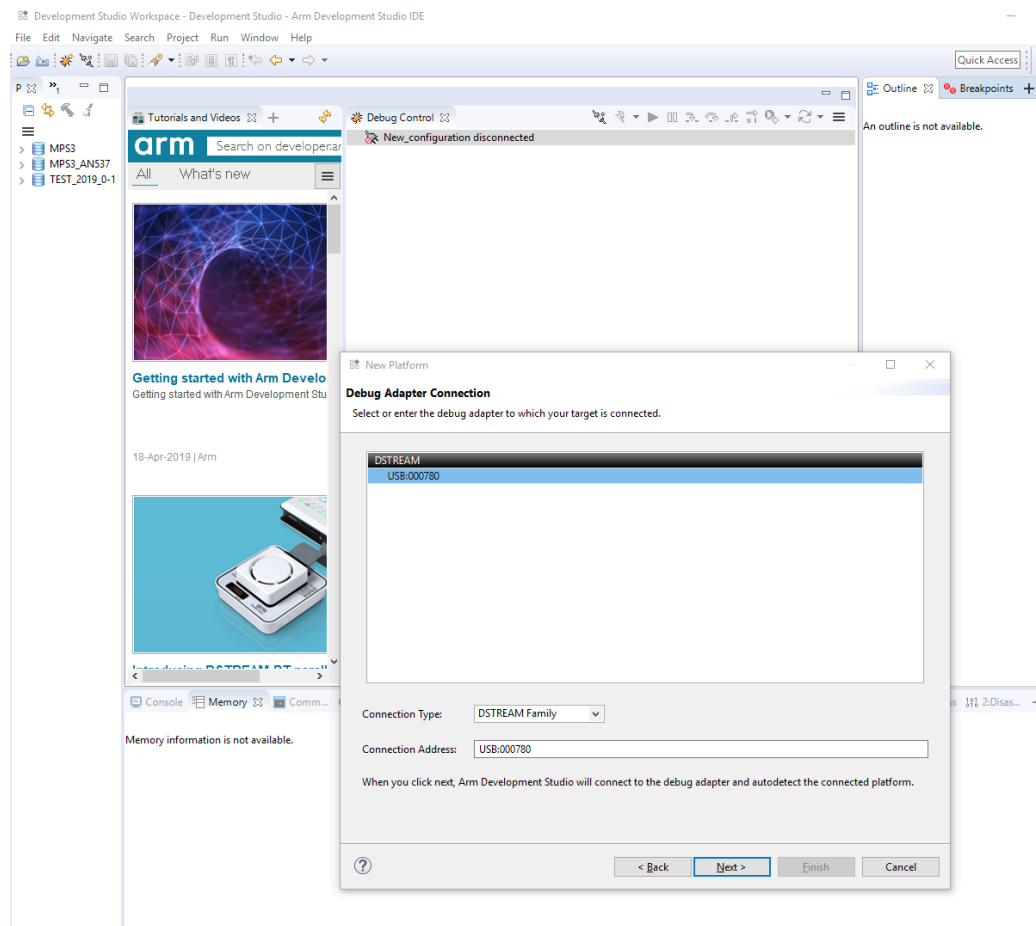


4. Under the **Configuration Database** folder, choose **Platform Configuration** (which highlights when selected) and click **Next**.

5. Select Automatic/simple platform detection and click Next as shown below.



6. Select your connection type and connection address for your target debugger. The figure below shows the case for a DSTREAM debugger with a USB connection.

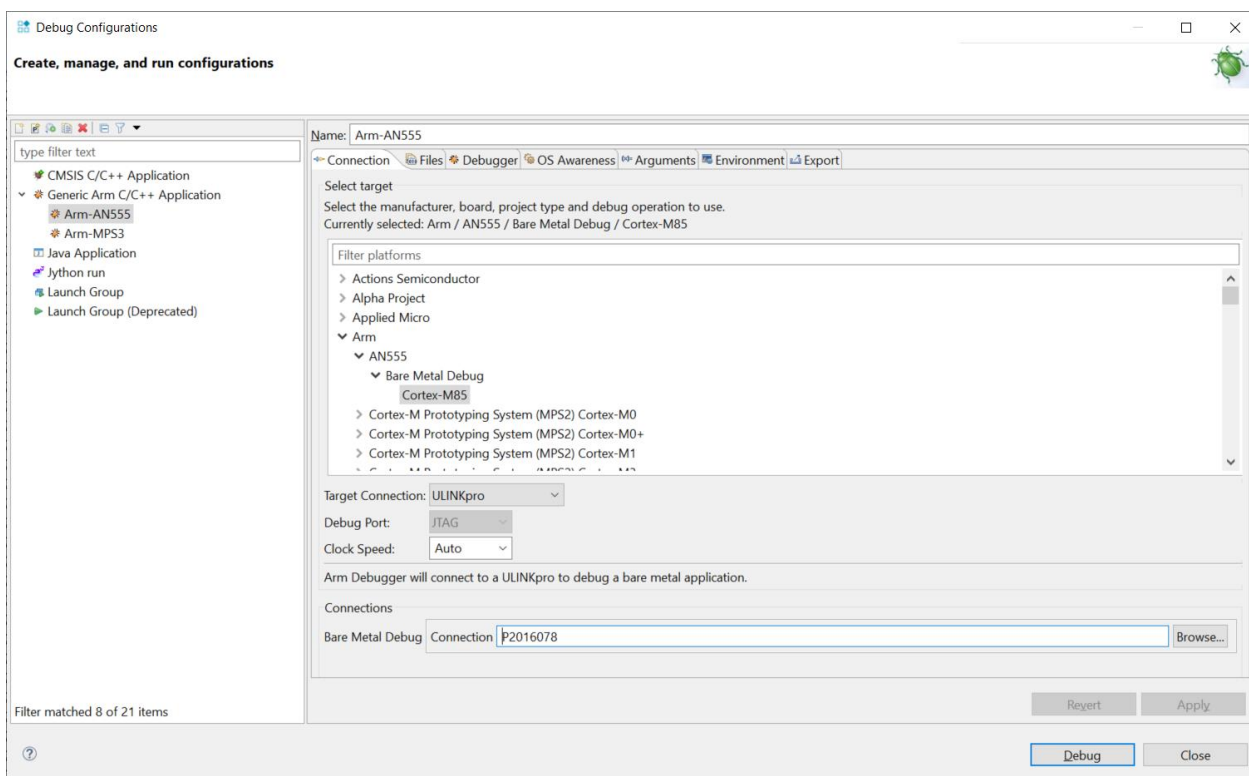


7. Ensure that the debugger is connected to the target (the MPS3 board in this case) and that the MCC has loaded the FPGA and the implementation has booted. Click **Next**.

8. Auto Discovery traverses the debug infrastructure to build a configuration for this target. The console should show something similar to the verbose output shown in the example below.

```
PCE Console
[12/05/22 16:23:19] End of ROM table
[12/05/22 16:23:19] --- --- ---
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Acquiring device info for CSETM (0xE0041000):
[12/05/22 16:23:20]   Device supports timestamps
[12/05/22 16:23:20]   Device does not support context IDs
[12/05/22 16:23:20]   Device supports cycle accurate trace
[12/05/22 16:23:20]   Device does not support data address trace
[12/05/22 16:23:20]   Device does not support data value trace
[12/05/22 16:23:20]   Device does not support trace range
[12/05/22 16:23:20]   Version: 4.5
[12/05/22 16:23:20] Acquiring device info for CSCTI (0xE0042000):
[12/05/22 16:23:20]   Number of supported triggers: 8
[12/05/22 16:23:20] Acquiring device info for CSTPIU (0xE0040000):
[12/05/22 16:23:20]   Supported Port Sizes: 1,2,4,8,12,16
[12/05/22 16:23:20]   Clock prescaler enabled for parallel trace
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Detecting topology ... (may take several seconds)
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Testing ATB topology:
[12/05/22 16:23:20] MASTER = CSITM (0xE0000000) (0)   SLAVE = CSTPIU (0xE0040000) (0)
[12/05/22 16:23:20] MASTER = CSETM (0xE0041000) (0)   SLAVE = CSTPIU (0xE0040000) (0)
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Testing core-source topology:
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Testing CTI topology:
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Asserting M-class topology using device base addresses:
[12/05/22 16:23:20] MASTER = Cortex-M85 (0xE000E000)   SLAVE = CSCTI (0xE0042000)   Trigger = 1
[12/05/22 16:23:20] MASTER = CSETM (0xE0041000)   SLAVE = CSCTI (0xE0042000)   Trigger = 4
[12/05/22 16:23:20] MASTER = Cortex-M85 (0xE000E000)   SLAVE = CSETM (0xE0041000)
[12/05/22 16:23:20] MASTER = CSCTI (0xE0042000)   SLAVE = CSTPIU (0xE0040000)   Trigger = 3
[12/05/22 16:23:20] --- --- ---
[12/05/22 16:23:20] Asserting missing topology using existing link information:
[12/05/22 16:23:20] --- --- ---
Debug server shutdown application
Debug server shut down attempted
Done
[12/05/22 16:23:21] Autodetection Complete
[12/05/22 16:23:42] Creating database entry...
[12/05/22 16:23:42] Platform "Arm - AN555" built successfully
```

9. Select the target configuration database or create a new one depending on your requirements. In this example a new database **Example** has been created which is used.
10. Click **Finish** to complete the setup process.
11. The newly created debug configuration is now available for selection, similar to the example below:



## 12.4. Establishing a Debug Session

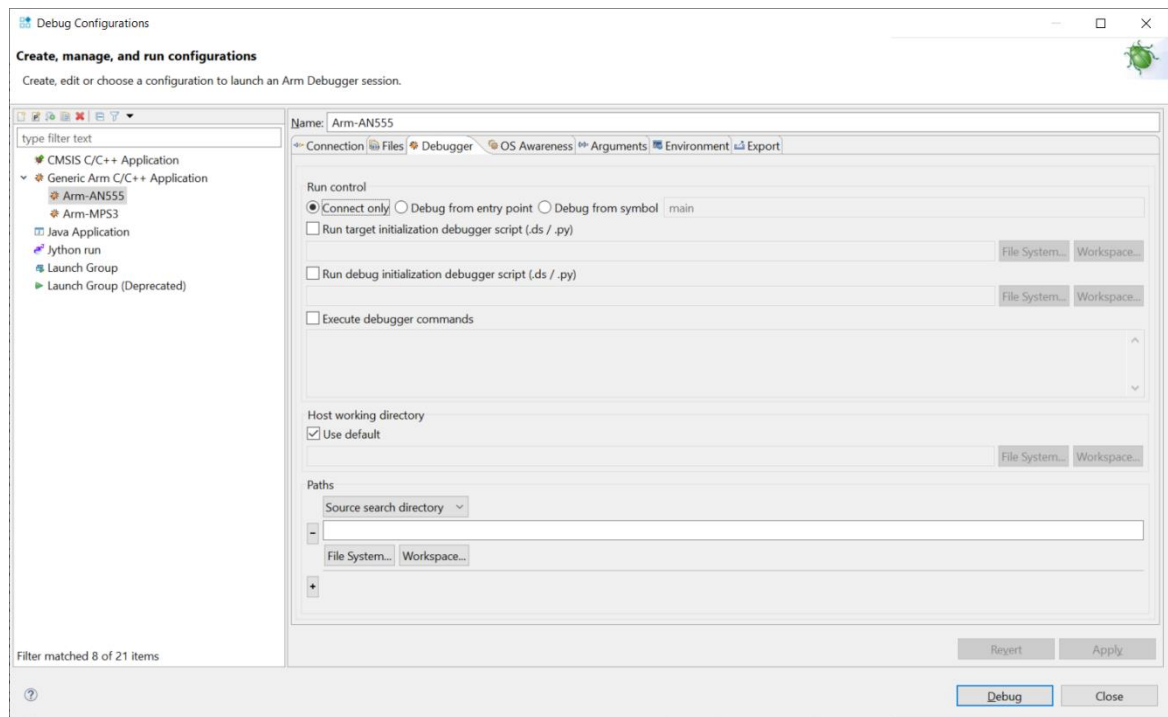
In this example we connect to the Cortex-M85.

Steps:

1. Ensure the Development Studio debugger is:
  - a. Powered, and connected to the host running the Development Studio software.
  - b. Connected to the MPS3 using the Coresight 20-pin / Mictor 38 port on the MPS3 as shown on **Figure 12-1: MPS3 Board Debug Connector Locations**
2. Open the Debug Configurations dialog box (by right-clicking on the newly created debug session in section [Creating a Development Studio Debug Configuration](#) (in the Debug Control Tab)) and ensure that Cortex-M85 target is selected under the **Connection** tab.
3. Click the **Debugger** tab, select **Connect only** under **Run control**.



An example dialog box is shown below. This example is pointing to the example made in the workspace.



4. Click **Debug**.



# 13. Using AN555 on the MPS3 Board

## 13.1. Pre-Requisites

Before attempting to use the board you must:

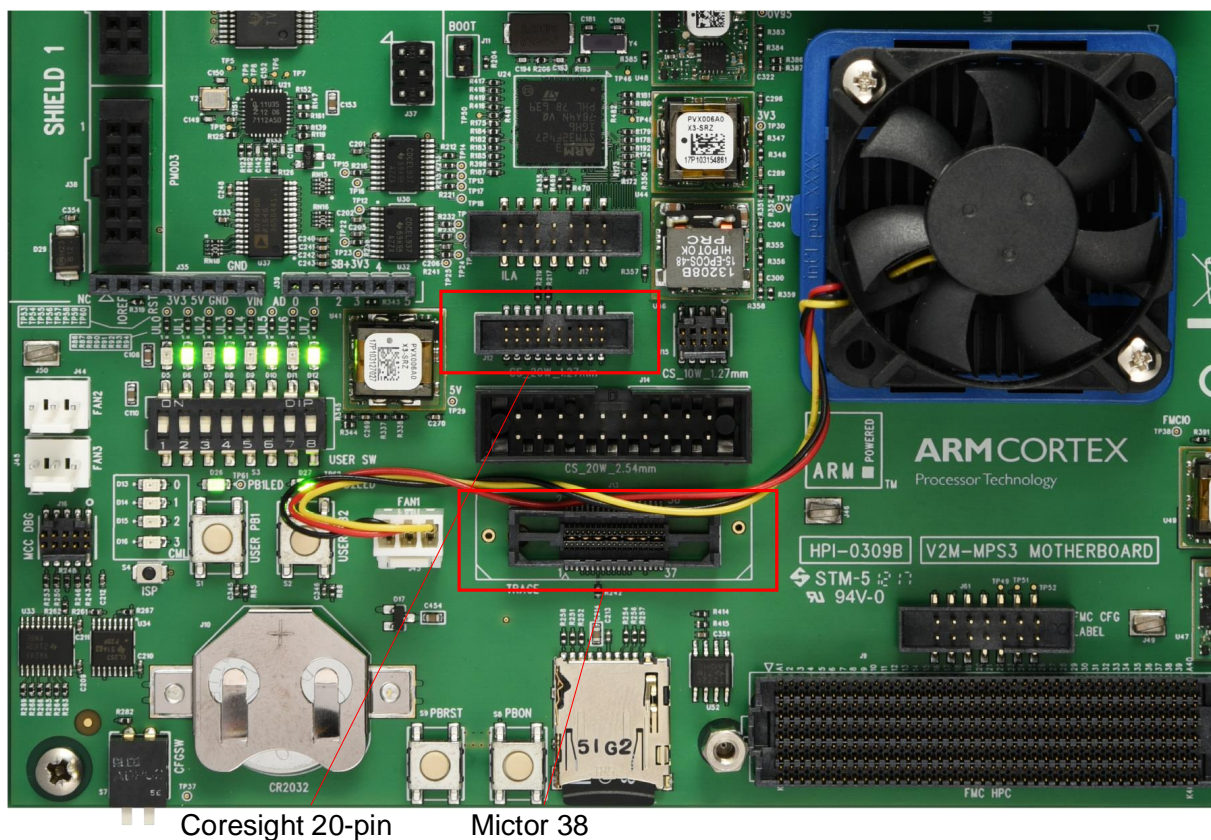
- Read the *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual*. In particular, become familiar with the description of the configuration and boot flow.

You must be able to:

- Connect a PC to the MPS3 board using a USB connection (which is required to load files onto the MPS3 board SD card).
- Power the MPS3 board. The MPS3 board appears as a mapped drive named V2M-MPS3.
- Understand how to power up, reset and establish a serial terminal over the USB connection to a host PC.

The following picture shows the location of the debug connectors on the MPS3 board.

Figure 13-1: MPS3 Prototyping Board debug connector locations



## 13.2. Loading the Boardfiles

To load the Boardfiles onto the MPS3 board SD Card, follow the following steps.

1. Open the bundle ZIP file
2. Navigate into the `Boardfiles/` directory
3. Copy its contents into the root directory of the attached MPS3 drive (the MPS3 SD Card).  
The require content to be copied is:
  - `MB/`
  - `SOFTWARE/`
  - `config.txt`



It is recommended to delete any previous contents on the SD card in favour of these new contents. However, you might want to manually modify and merge the contents for certain configuration files.

---

The configuration files that can be modified can be found on the SD card here:

- `config.txt`
- `MB/HBI0309{<boardrev>}/board.txt`
- `MB/HBI0309{<boardrev>}/AN555/images.txt`

## 13.3. UART Serial ports

The AN555 FPGA supports four serial ports which are accessible through the MPS3 board Debug USB port:

- Serial Port 0 is connected to the MCC and outputs debug information about the status of the MCC.
- Serial Port 1 is connected to UART 0.
- Serial Port 2 is connected to UART 1.
- Serial Port 3 is connected to UART 2.



The logical < > physical mapping of the serial ports on a host PC can be confusing due to the way the driver may allocate the port numbers. The serial port presented with the lowest number aligns to Serial Port 0 above.

---

## 13.4. UART serial port settings

All serial ports on the AN555 FPGA implementation use the following terminal/serial port settings:

- Baud Rate: 115200 bps
- New-Line: CR (Serial port 0) And LF (Serial Port 1,2 and 3 Only)
- Data: 8 bits
- Parity: none
- Stop: 1 bit
- Flow control: none

## 13.5. MPS3 USB Serial Port Drivers

For information on installing drivers to support USB serial port on MPS3:

<https://community.arm.com/oss-platforms/w/docs/589/accessing-mps3-serial-ports-in-windows-10>

## 13.6. MCC Memory mapping

The MCC on the MPS3 has visibility of certain areas of the system memory map. This is to enable preload of boot memory and data memories and configuring peripherals if needed.

The following table shows the memory map as viewed from the MCC. It also gives example addresses for each region for use in `an555_v2.txt` and `images.txt`. Preloading can only target secure memory locations.

**Table 13-1 : MCC memory map**

Address	Size	IOFPGA	Text file format example
0x1100_0000 - 0x111F_FFFF	2MB	FPGA SRAM	0x01_00_1100_0000
0x1A00_0000 - 0x1A00_7FFF	32KB	ITCM Access	0x01_00_1A00_0000
0x2400_0000 - 0x2400_7FFF	32KB	DTCM Access	0x01_00_3400_0000
0x3100_0000 - 0x313F_FFFF	4MB	Internal SRAM	0x01_00_3100_0000
0x5000_0000 - 0x57FF_FFFF	128 MB	Low Latency Peripherals	0x00_00_5000_0000 or 0x01_00_5000_0000
0x5800_0000 - 0x5FFF_FFFF	128 MB	High Latency Peripherals	0x00_00_5800_0000 or 0x01_00_5800_0000
0x7000_0000 - 0x7FFF_FFFF 0x9000_0000 - 0x9FFF_FFFF 0xB000_0000 - 0xBFFF_FFFF 0xD000_0000 - 0xDFFF_FFFF	2GB	DDR4	0x01_00_7000_0000 0x01_00_9000_0000 0x01_00_B000_0000 0x01_00_D000_0000

## 13.7. MCC Debug UART – Serial Port 0

The MPS3 board supports an MCC debug UART interface on serial port 0. This interface has the following functionality:

- Bi-directional CLI interface
- Verbose status output of MPS3 during the power on and boot stages
- Support to reboot the MCC using command-line syntax

### 13.7.1. MCC Debug UART – Verbose Output

It is possible to enter debug mode on the MCC UART. To do so

1. Change `images.txt` file TOTALIMAGES parameter to 0 and change `config.txt` UARTMODE from 2 to 1.
2. Reboot the board.
3. Once the initialization process is completed and you have received output on all four UART's press PBON to perform a software reset. An example output from the MCC UART can be seen below.

```
ARM V2M-MPS3 Firmware v2.2.0
Build Date: Oct 25 2022

Powering up system...
Switching on main power...
Configuring motherboard (rev A, var A)...

Reading Board File \MB\HBI0309A\AN555\an555_v2.txt

Configuring FPGA from file \MB\HBI0309A\AN555\an555_v2.bit
Address: 0x02840000
FPGA configuration complete.

OSCCLK0 : 25.000000MHz
OSCCLK1 : 25.000000MHz
OSCCLK2 : 30.000000MHz
OSCCLK3 : 50.000000MHz
OSCCLK4 : 24.576000MHz
OSCCLK5 : 23.750000MHz

OSCCLK setup: PASSED
GTXCLK (100): PASSED

Writing SCC 0x010 with board revision A
Writing SCC 0x014 with ACLK 25000000 Hz
Setting QSPI PMOD0 FALSE & PMOD1 FALSE configuration
QSPI PMOD status: 0x0
Reading SCC 0xFF8 with 0x02100708
Reading SCC 0xFFC with 0x41055551

Long Address Range Version 1.0 Ports 2
PORT0 AHB 0x0000_00000000 -1KB
```

```
PORT1 AHB 0x0000_00000000 -1KB

Releasing CB_nRST
Initialise QSPI ON-BOARD
QSPI On-board Init: PASSED
Reading images file \MB\HBI0309A\AN555\images.txt

Image 0 LAR port 0 no offset added
Writing ELF file \SOFTWARE\an555_st.axf
Writing segment 0 to LAR address 0x0100_11000000
.....
File \SOFTWARE\an555_st.axf written to memory address 0x0100_11000000
Image loaded from \SOFTWARE\an555_st.axf

Setting QSPI to XIP (QSPI) mode...
XIP Controller SCC Reg: 0x0
SMSC9220 was identified successfully.
MAC addr test: PASSED

DDR SPD EEPROM detected...

UART0: MCC, UART1: FPGA0
Clearing SCC CPUWAIT (0xFFFFFFFF80000000:0x00000000)
Enabling debug USB.
USB Serial Number = 500510
```

The verbose output above shows:

- The MCC booting
- The MCC reading the configuration files from the SD CARD
- The MCC configuring the FPGA with the target `.bit` file



If the MPS3 board does not boot correctly, refer to the log file `LOG.TXT` on the SD Card.

### 13.7.2. MCC Debug UART – Reboot Control

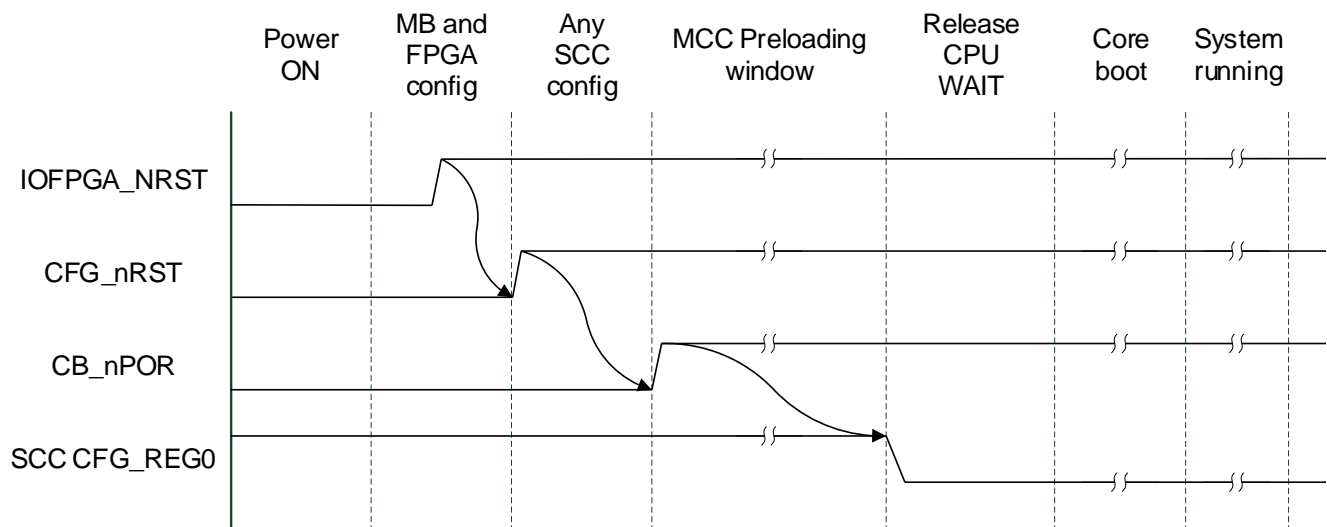
The command `reboot` on the terminal emulator console initiates MCC reboot.

### 13.7.3. The Preload Sequence

The preload sequence is:

1. MPS3 is powered up.
2. The PBON button on the MPS3 board is pressed.
3. The MCC configures the FPGA with the FPGA image, which is specified in the FPGA configuration file in `MB/HBI0309{<boardrev>}/AN555/an555_v2.txt`.
4. The FPGA power on reset is released.
5. The Cortex-M85 is held in wait state by driving CPUWAIT from the SCC CFG\_REG0, which resets to high at boot up. The rest of the system is out of reset to allow preloading.
6. The MCC can then preload the target memories, using the specified images (if directed to do so in the `images.txt` file settings (see [Preload Image Configuration \(images.txt\)](#) for more details).
7. The MCC then clears the SCC CFG\_REG0 which de-asserts the Cortex-M85 CPUWAIT.
8. The Cortex-M85 boots from the FPGA SRAM, booting into the code preloaded from the designated image file.

**Figure 13-2 : Preload reset sequence**



## 13.7.4. Preload Image Configuration (images.txt)

Preload image configuration:

Images to be preloaded are selected/configured by modifying the following file on the SDCARD  
MB/HBI0309{<boardrev>}/AN555/images.txt.

For convenience, the file in the bundle has been templated with the syntax to load the FPGA SRAM, with the prebuilt selftest image, also supplied in the bundle. This can be observed below:

```
TITLE: Arm MPS3 FPGA prototyping board Images Configuration File

[IMAGES]
TOTALIMAGES: 1 ;Number of Images (Max: 32)

IMAGE0ADDRESS: 0x01_00_1100_0000 ;Software image load address
IMAGE0UPDATE: RAM ;Image Update type
IMAGE0FILE: \SOFTWARE\an555_st.axf ;Software image filename SFN 8.3 format
```



A “;” preceding any text on the same line in the `images.txt` file denotes a comment and is ignored by the MPS3 MCC.

The `images.txt` can be modified by the user with the following restrictions:

- **TOTALIMAGES** - Can have a value from 0-32. Each additional image to be loaded must be described with an **IMAGExADDRESS**, **IMAGExUPDATE** and **IMAGExFILE** entry and follow the syntax and format shown in the example above.
- **IMAGExADDRESS** - Will be the address that the image needs to be loaded to. The addresses for available preload regions in AN555 FPGA can be found in [MCC Memory mapping](#), for QSPI preload addresses see [Preloading QSPI Memories](#).
- **IMAGExUPDATE** - Informs the MCC how it should load the data. The options for this are:
  - NONE | RAM | FORCEQSPI | FORCEQSPI\_PMOD0 | FORCEQSPI\_PMOD1

It is advised that any new entries use **RAM** except when targeting QSPI memories when **FORCEQSPI** / **FORCEQSPI\_PMOD0** / **FORCEQSPI\_PMOD1** should be used.

- **RAM** - Informs the MCC to load the specified file to the specified address.
  - **FORCEQSPI\*** - Informs the MCC to load QSPI memory using the required driver to the base address specified in `an555_v2.txt` plus the offset specified by **IMAGExADDRESS**
- **IMAGExFILE** - This is the location on the SD card of the software image to load and it must use the SFN 8.3 format for the filename.





Note

The `TOTALIMAGES` defines how many of the entries are read by the MCC, no matter how many entries are present in the `images.txt` file. E.g., if the `TOTALIMAGES` parameter is set to 1, the MCC will only read in the file for `"IMAGE0xxxx"` and ignore `"IMAGE1xxxx"`.

You can modify the file whilst the MCC/system is booted and running, the file is only read by the MCC during the boot up sequence.

To reboot the MCC after an image file configuration change, you must do one of the following:

- Press the **PBRST** button on the MPS3 board, followed by a pressing the **PBON** button to reboot the MCC.
- Issue a `reboot` command on the MCC console.

## 13.8. Memory Preload

### 13.8.1. Supported Memory Preload File Types

The MPS3 MCC supports preload of the following file types:

- AXF – Object file, with an `.axf` filename extension.
- Binary – Binary file, with a `.bin` filename extension.

### 13.8.2. Supported Memory Preload File Naming Format

The files that can be preloaded by the MCC must use the following file naming format:

- The MPS3 MCC uses the SFN (short filename) 8.3 format.
- The filename must have a maximum length of 8 characters.
- The filename must have a three-character extension (`.axf` or `.bin`).

### 13.8.3. Preloading QSPI Memories

To preload any of the QSPI memories a driver is used by the MCC. The driver automatically uses the QSPI read/write controller address specified in `an555_v2.txt`.

Therefore, for the `IMAGExADDRESS` in `images.txt` only the offset from the QSPI base address needs to be specified.

`IMAGExUPDATE` also needs to use one of the QSPI specific options.

Below is an example `images.txt` entry for preloading to QSPI `PMOD0`.

```
TITLE: Arm MPS3 FPGA prototyping board Images Configuration File

[IMAGES]
TOTALIMAGES: 1                      ;Number of Images (Max: 32)

IMAGE0ADDRESS: 0x00_00_0000_0000    ;Software image load address
IMAGE0UPDATE:  AUTOQSPI_PMOD0       ;Image Update type
IMAGE0FILE:  \SOFTWARE\pmodcode.bin  ;Software image filename SFN 8.3 format
```

# 14. Other Software Applications

The following Machine Learn (ML) Software is compatible with this FPGA image:

- <https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ml-embedded-evaluation-kit/+/refs/heads/main/Readme.md>.
- <https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ethos-u-core-platform/>.