Application Note AN545

Example Cortex-M33 Subsystem
with Custom Datapath Extension for MPS3

Non-Confidential

arm

Example Cortex-M33 Subsystem with Custom Datapath Extension for MPS3

Release Information

The following changes have been made to this Application Note.

Change History

Date	Issue	Confidentiality	Change
27 August 2020	A	Non-Confidential	First version

LICENCE GRANTS

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

DELIVERABLES

Part A

Hardware Binaries:
Encrypted FPGA bitstream file containing the following Arm technology:
SSE-200 Subsystem
Cortex-M33 processor

Hardware Source Code:
Hardware netlists of Arm CoreLink peripheral technology and components known as
NIC-400, BP136.
RTL Example designs of Arm components in the Cortex-M33 kit known as CDE and FPCDE
RTL of Arm CoreLink peripheral technology and components known as PL022 and
PL031, apb_i2s_top, CharLCDI, SBCon
RTL of Arm components in the Cortex-M System Design Kit (CMSDK) including:
cmsdk_sram, cmsdk_ahb_gpio, cmsdk_apb_uart, cmsdk_irq_sync, cmsdk_to_extmem

Software Binaries:
Motherboard Configuration Controller binary, including Keil USB and SD card
drivers, and Analog Devices FMC EEPROM reader

Documentation:
Documentation, provided as PDF

Part B

Wrapper:
Wrapper file(s) identified in the Documentation provided as hardware source
files and netlists.

Part C

Example Code:
Platform initialisation source code
Platform specific libraries and source code
Selftest example source code
Demo example source code
Arm source code portions of the Self-test

Software Binaries:
Demo binary
Self-test binary

Contents

1 Conventions and Feedback

The following describes the typographical conventions and how to give feedback:

Typographical conventions

The following typographical conventions are used:

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

<u>`mono`</u>`space`   Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

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

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

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

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

Feedback on this product

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

- Your name and company.

- The serial number of the product.

- Details of the release you are using.

- Details of the platform you are using, such as the hardware platform, operating system type and version.

- A small standalone sample of code that reproduces the problem.

- A clear explanation of what you expected to happen, and what actually happened.

- The commands you used, including any command-line options.

- Sample output illustrating the problem.

- The version string of the tools, including the version number and build numbers.

Feedback on documentation

If you have comments on the documentation, e-mail *errata@Arm.com.* Give:

- The title *Example Cortex-M33 Subsystem with Custom Datapath Extension for MPS3*

- The number, DAI 0545A.

- If viewing online, the topic names to which your comments apply.

- If viewing a PDF version of a document, the page numbers to which your comments apply.

- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Arm periodically provides updates and corrections to its documentation on the Arm's technical documentation Center.

Other information

- Arm's technical documentation Center, *https://developer.arm.com/documentation/*

- Arm Support and Maintenance, *https://www.arm.com/support*

2 Preface

2.1 Purpose of this application note

This document describes the features and functionality of application note AN545. AN545 is an FPGA implementation of the SSE-200 Subsystem that uses Cortex-M33 Core with Custom Datapath Extension, SIE-200 components together with CMSDK peripherals to provide an example design.

2.2 References

- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual (Arm 101104)*

- *Arm® Cortex®-M33 Processor Technical Reference Manual (Version 100230_0100_03_en)*

- *Arm Custom Instructions: Enabling Innovation and Greater Flexibility on Arm (White Paper from Feb 2020)*

- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual (Arm DDI 0571).*

- *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual* (Version 100765_0000_03_en)

- *Arm® Cortex®-M System Design Kit Technical Reference Manual (Arm DDI 0479)*

- *Arm® MPS3 FPGA Prototyping Board Getting Started Guide*

- *MCBQVGA-TS-Display-v12 – Keil MCBSTM32F200 display board schematic.*

- *Xilinx Vivado Design Suite User Guide UG909*

2.3 Terms and abbreviations

ACI	Arm Custom Instructions
ADB	AMBA Domain Bridge
AHB	Advanced High-performance Bus.
APB	Advanced Peripheral Bus.
AXI	Advanced Extensible Interface.
BRAM	FPGA Block RAM
CA	Cryptographic Accelerator
CDE	Custom Datapath Extension
CMSDK	Cortex-M System Design Kit
CPU	Central Processing Unit
CVM	On-Chip Volatile Memory
DPRAM	Dual Port RAM
EIS	Engineering Implementation Specification
EXPMST	Expansion Master

EXTSYS0	External System 0
FPGA	Field Programmable Gate Array
FPCDE	Floating-Point Unit Custom Datapath Extension
FPU	Floating-Point Unit
FW	Firewall
GIC	Generic Interrupt Controller
JTAG	Joint Test Action Group
KB	Kilo Byte.
MB	Mega Byte.
MCC	Motherboard Configuration Controller.
MHU	Message Handling Unit
MIG	Memory Interface Generator
MPS3	Microcontroller Prototyping System 3
NIC	Network Interconnect
OCVM	Off-Chip Volatile Memory
RAM	Random Access Memory.
RAZ	Read as Zero
ROM	Read Only Memory
RTC	Real time Clock
RTL	Register Transfer Level
SCB	Security Control Bits
SCC	Serial Configuration Controller
SE	Secure Enclave
SMM	Soft Macrocell Model
SoC	System on Chip
SWD	Serial Wire Debug
TBD	To Be Defined
TRM	Technical Reference Manual
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WI	Write Ignored
XNVM	eXecute-in-place Non-Volatile Memory (XNVM)

2.4 Subsystem version details

This SMM is generated using various packages and IP. These are detailed below:

Version	Descriptions
BP210	**Cortex-M System Design Kit** Full version of the design kit supporting Cortex-M0, Cortex-M0 DesignStart®, Cortex-M0+, Cortex-M3 and Cortex-M4. Also contains the AHB Bus Matrix and advanced AHB components.
r3p1	**SIE-200** SIE-200 is a system IP library to enable Armv8-M and TrustZone for v8-M ecosystem. All SIE-200 components have AHB5 interfaces to support Armv8-M processors.
r1p0 REL	**Cortex-M33** The Cortex-M33 processor is a low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications. The processor is based on the Armv8‑M architecture and is primarily for use in environments where security is an important consideration. It has support for *Arm Custom Instructions* (ACIs) and include implementation of the *Custom Datapath Extension* (CDE)
r2p0	**SSE-200** The SSE-200 is a collection of a pre-assembled elements to use as the basis of an IoT SoC.

Figure 2-1 : Module versions

2.5 Encryption key

Arm supplies the MPS3 Prototyping Board with a decryption key programmed into the FPGA. This key enables loading of prebuilt encrypted images.

Note

Some of the FPGA programming files that is supplied as part of the bundle are 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.

3 Overview

This SMM is based around the SSE-200 Subsystem which contains dual Cortex-M33 core, the system is then extended with interconnect and peripherals.

The SMM is implemented using Partial Reconfiguration which allows the user to modify the user partition shown below.

3.1 System block diagram

The diagram below shows the high level of the full MPS3 SSE-200 FPGA System.
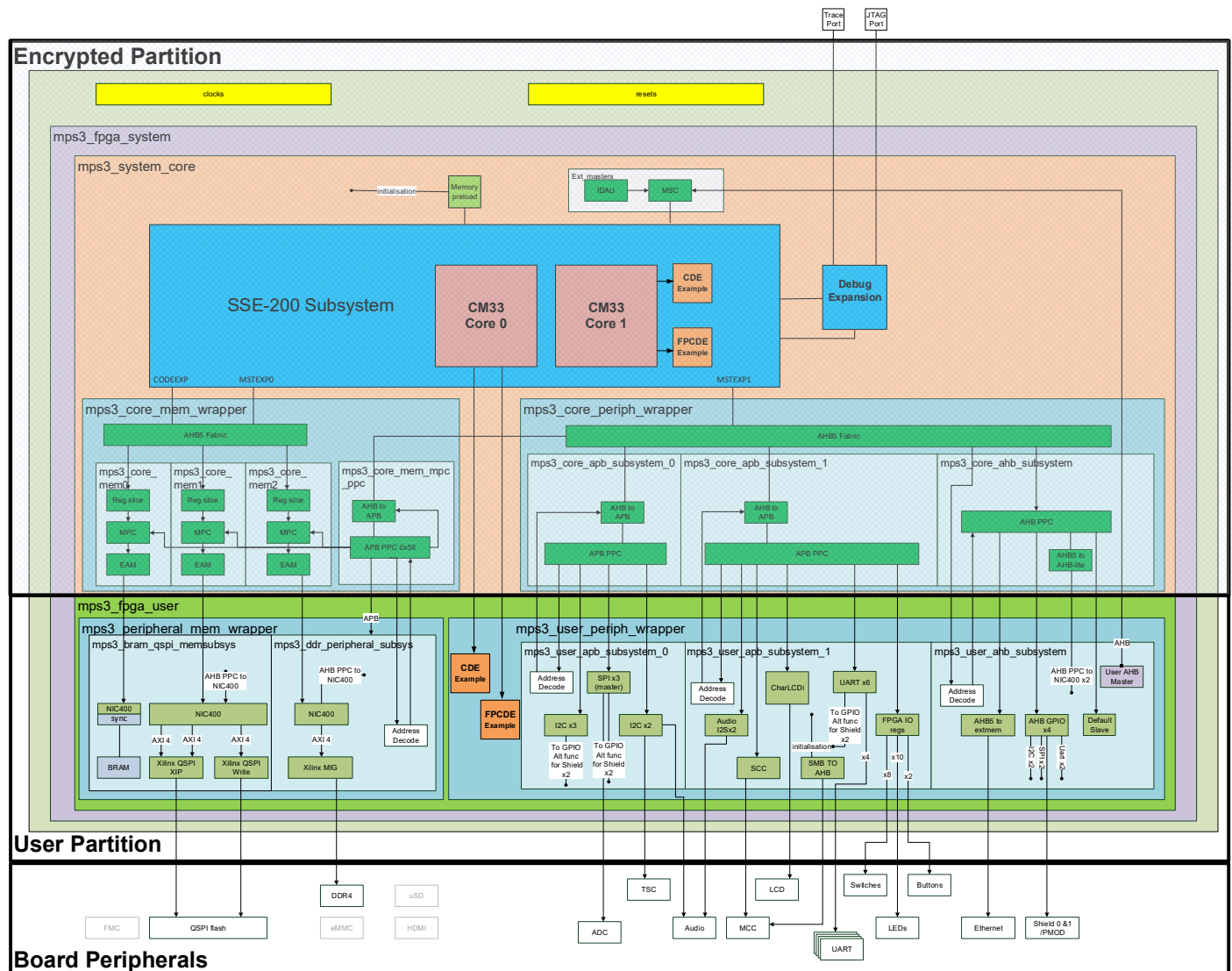
Figure 3-1 : System Overview

Note how the FPGA Subsystem extends the SSE-200 Subsystem by adding to its expansion interfaces.

3.2   SIE-200 components

The following SIE-200 components are used in this system:

- TrustZone AHB5 peripheral protection controller.

- TrustZone AHB5 master security controller.

- AHB5 bus matrix.

- AHB5 to AHB5 synchronous bridge.

- AHB5 to APB synchronous bridge.

- TrustZone APB4 peripheral protection controller.

- TrustZone AHB5 memory protection controller.

- AHB5 exclusive access monitor.

- AHB5 default slave.

3.3   Memory protection note

The SIE-200 MPC and PPC components can affect memory and IO security management and must be configured as required for your application. Please see *Arm® SIE-200 System IP Technical Reference Manual* (Arm DDI0571).

3.4 Memory Map Overview

This memory map includes information regarding IDAU security information for memory regions.

For more information on these, please refer to the SIE-200 components documentation.
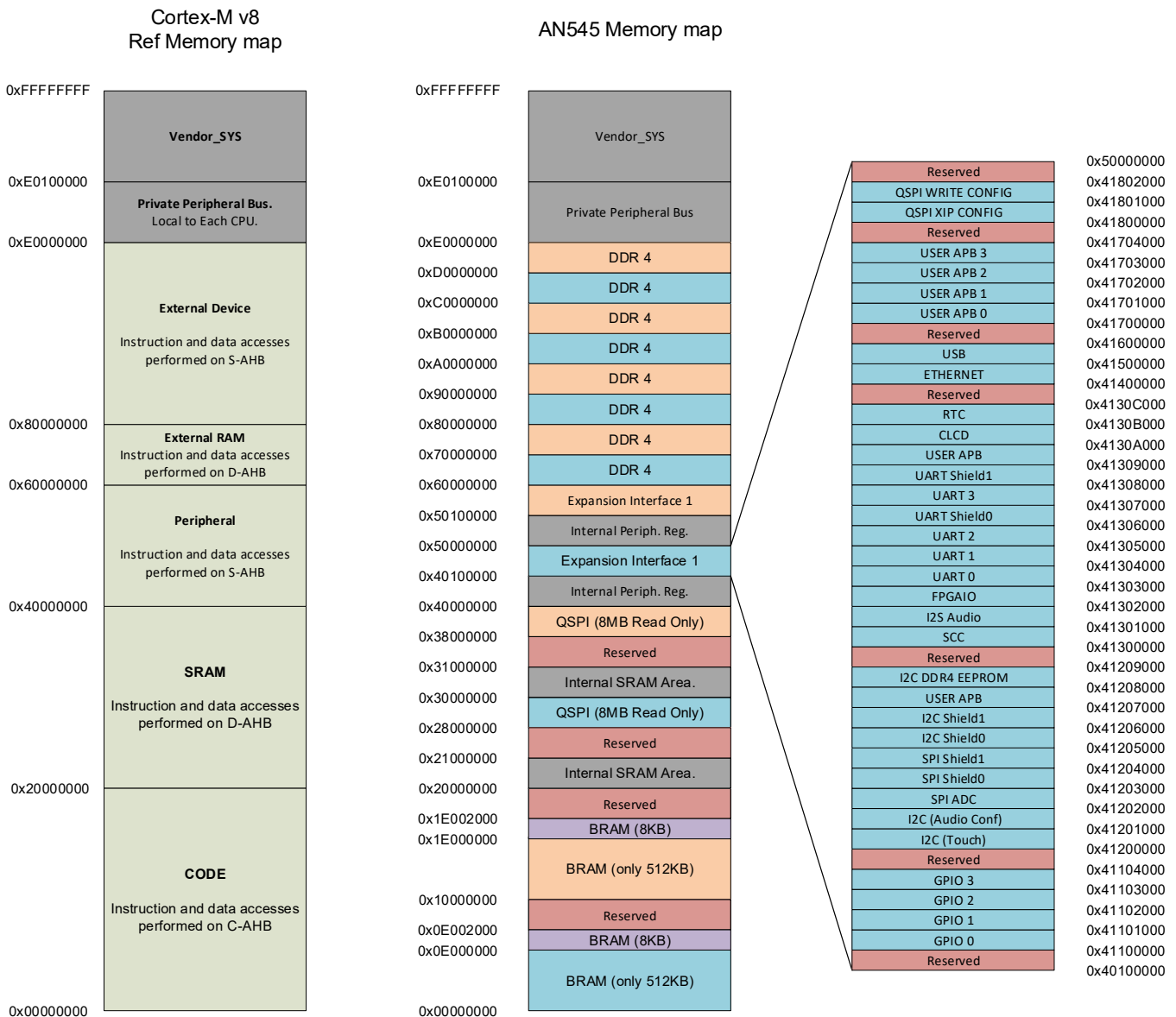
Figure 3-2 : Memory Map

ROW ID	Address From	To	Size	Region Name	Description	Alias With Row ID	IDAU Region Values Security	IDAUID	NSC
1	0x0000_0000	0x0007_FFFF	512KB	Code Memory	BRAM	5			
2	0x0008_0000	0x0DFF_FFFF	223MB	Reserved	Reserved		NS	0	0
3	0x0E00_0000	0x0E00_1FFF	8KB	NVM Code	CryptoCell APB code interface for NVM	7			
4	0x0E00_2000	0x0FFF_FFFF	32MB	Reserved	Reserved				
5	0x1000_0000	0x1DFF_FFFF	512KB	Code Memory	Alias to BRAM	1			
6	0x1008_0000	0x1DFF_FFFF	223MB	Reserved	Reserved		S	1	CODE NSC2
7	0x1E00_0000	0x1E00_1FFF	8KB	NVM Code	CryptoCell APB code interface for NVM	3			
8	0x1E00_2000	0x1FFF_FFFF	32MB	Reserved	Reserved				
9	0x2000_0000	0x2000_7FFF	32KB	Internal SRAM	Internal SRAM 0 Area	16			
10	0x2000_8000	0x2000_FFFF	32KB	Internal SRAM	Internal SRAM 1 Area	17			
11	0x2001_0000	0x2001_7FFF	32KB	Internal SRAM	Internal SRAM 2 Area	18			
12	0x2001_8000	0x2001_FFFF	32KB	Internal SRAM	Internal SRAM 3 Area	19	NS	2	0
13	0x2002_0000	0x27FF_FFFF	112MB	Reserved	Reserved				
14	0x2800_0000	0x287F_FFFF	8MB	Expansion 0	QSPI (Read Only)	21			
15	0x2880_0000	0x2FFF_FFFF	120MB	Reserved	Reserved				
16	0x3000_0000	0x3000_7FFF	32KB	Internal SRAM	Internal SRAM 0 Area	9			
17	0x3000_8000	0x3000_FFFF	32KB	Internal SRAM	Internal SRAM 1 Area	10			
18	0x3001_0000	0x3001_7FFF	32KB	Internal SRAM	Internal SRAM 2 Area	11			
19	0x3001_8000	0x3001_FFFF	32KB	Internal SRAM	Internal SRAM 3 Area	12	S	3	RAMNSC
20	0x3002_0000	0x37FF_FFFF	112MB	Reserved	Reserved				
21	0x3800_0000	0x387F_FFFF	8MB	Expansion 0	Alias to QSPI (Read Only)	14			
22	0x3880_0000	0x3FFF_FFFF	120MB	Reserved	Reserved				
23	0x4000_0000	0x4000_FFFF	64KB	Base Peripheral	Base Element Peripheral Region.	30			
24	0x4001_0000	0x4001_FFFF	64KB	Private CPU	CPU Element Peripheral Region.	31	NS	4	0
25	0x4002_0000	0x4002_FFFF	64KB	System Control	System Control Element Peripheral region.	32			

ROW ID	Address From	To	Size	Region Name	Description	Alias With Row ID	Security	IDAUID	NSC
26	0x4003_0000	0x4003_FFFF		Reserved	Reserved				
27	0x4004_0000	0x4007_FFFF		Reserved	Reserved				
28	0x4008_0000	0x400F_FFFF	512KB	Base Peripheral	Base Element Peripheral Region.	35			
29	0x4010_0000	0x4FFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	36			
30	0x5000_0000	0x5000_FFFF	64KB	Base Peripheral	Base Element Peripheral Region.	23			
31	0x5001_0000	0x5001_FFFF	64KB	Private CPU	CPU Element Peripheral Region.	24			
32	0x5002_0000	0x5002_FFFF	64KB	System Control	System Control Element Peripheral region.	25			
33	0x5003_0000	0x5003_FFFF		Reserved	Reserved		S	5	0
34	0x5004_0000	0x5007_FFFF		Reserved	Reserved				
35	0x5008_0000	0x500F_FFFF	512KB	Base Peripheral	Base Element Peripheral Region.	28			
36	0x5010_0000	0x5FFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	29			
37	0x6000_0000	0x6FFF_FFFF	256MB	Expansion 0	DDR4	38	NS	6	0
38	0x7000_0000	0x7FFF_FFFF	256MB	Expansion 0	DDR4	37	S	7	0
39	0x8000_0000	0x8FFF_FFFF	256MB	Expansion 1	DDR4	40	NS	8	0
40	0x9000_0000	0x9FFF_FFFF	256MB	Expansion 1	DDR4	39	S	9	0
41	0xA000_0000	0xAFFF_FFFF	256MB	Expansion 1	DDR4	42	NS	A	0
42	0xB000_0000	0xBFFF_FFFF	256MB	Expansion 1	DDR4	41	S	B	0
43	0xC000_0000	0xCFFF_FFFF	256MB	Expansion 1	DDR4	44	NS	C	0
44	0xD000_0000	0xDFFF_FFFF	256MB	Expansion 1	DDR4	43	S	D	0
45	0xE000_0000	0xE00F_FFFF	1MB	PPB	Private Peripheral Bus. Local to Each CPU.	47	Exempt		
46	0xE010_0000	0xEFFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	48	NS	E	0
47	0xF000_0000	0xF00F_FFFF	1MB	System Debug	System Debug.	45	Exempt		
48	0xF010_0000	0xFFFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	46	S	F	0

Table 3-1 : Memory map overview

3.5 REMAP

Memory remap function is controlled via SCC CFGREG0[0] register. It can be setup in the FPGA_REMAP section of a an545_v0.txt file.

AN545 Remap Options
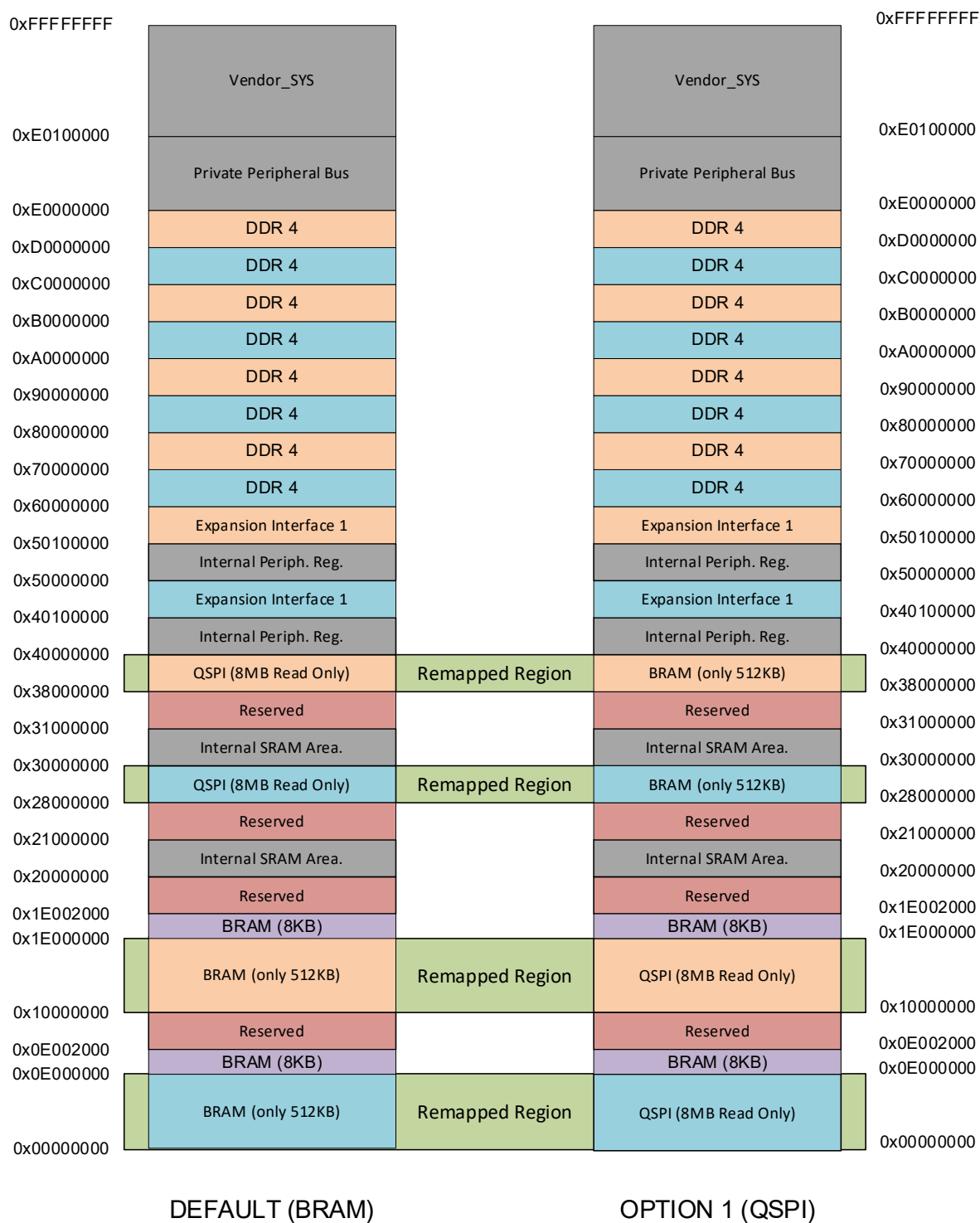
Figure 3-3 : Remap options

3.6 MCC Memory map for AN545

MCC should have some visibility into the memory for initiating boot memory areas and configuring peripherals if needed. MCC has a limited access to the design memory map (just 4x64MB), so it is unable to cover the whole map, hence only those regions which are necessary for the design functionality are visible.

The memory map as viewed from the MCC is below:

Address	Region	Chip Select
0x1000_0000	DDR 4 (2) S (only first 32MB)	CS4
0x0E00_0000	DDR 4 (2) NS (only first 32MB)	
0x0C00_0000	DDR 4 (1) S (only first 32MB)	CS3
0x0A00_0000	DDR 4 (1) NS (only first 32MB)	
0x0800_0000	Peripherals S	CS2
0x0500_0000	Peripherals NS	
0x0400_0000	Reserved	
0x0380_8000	QSPI S (8MB Read Only)	
0x0300_8000	Reserved	
0x0280_8000	QSPI NS (8MB Read Only)	
0x0200_8000	Reserved	CS1
0x0108_0000	BRAM S (only 512KB)	
0x0100_0000	Reserved	
0x0080_0000	BRAM NS (only 512KB)	
0x0000_0000		

CS	MCC SMB Address	MCC Internal	SSE-200 Address	Size	IOFPGA
1	0x0000_0000 - 0x0007_FFFF	0x6000_0000 - 0x607F_FFFF	0x0000_0000 - 0x0007_FFFF	512KB	BRAM NS
	0x0100_0000 – 0x0107_FFFF	0x6100_0000 - 0x617F_FFFF	0x1000_0000 - 0x0107_FFFF	512KB	BRAM S
	0x0200_8000 - 0x0280_7FFF	0x6200_8000 - 0x6280_7FFF	0x2800_0000 - 0x2880_7FFF	8 MB	QSPI NS
	0x0300_8000 - 0x0380_7FFF	0x6300_8000 - 0x6380_7FFF	0x3800_0000 - 0x3880_7FFF	8 MB	QSPI S
2	0x0400_0000 - 0x04FF_FFFF	0x6400_0000 - 0x64FF_FFFF	0x4000_0000 - 0x4FFF_FFFF	32 MB	Peripherals NS
	0x0500_0000 - 0x05FF_FFFF	0x6500_0000 - 0x65FF_FFFF	0x5000_0000 - 0x5FFF_FFFF	32 MB	Peripherals S
3	0x0800_0000 - 0x09FF_FFFF	0x6800_0000 - 0x69FF_FFFF	0x6000_0000 - 0x61FF_FFFF	32 MB	DDR 4 (1) NS
	0x0A00_0000 - 0x0BFF_FFFF	0x6A00_0000 - 0x6BFF_FFFF	0x7000_0000 - 0x71FF_FFFF	32 MB	DDR 4 (1) S
4	0x0C00_0000 - 0x0DFF_FFFF	0x6C00_0000 - 0x6DFF_FFFF	0x8000_0000 - 0x81FF_FFFF	32 MB	DDR 4 (2) NS
	0x0E00_0000 - 0x0FFF_FFFF	0x6E00_0000 - 0x6FFF_FFFF	0x9000_0000 - 0x91FF_FFFF	32 MB	DDR 4 (2) S

3.7 Expansion System peripherals

All FPGA peripherals that are extensions to the SSE-200 are mapped into two key areas of the memory map:

- 0x4010_0000 to 0x4FFF_FFFF Non-Secure region which maps to AHB Master Expansion 1 interface.

- 0x5010_0000 to 0x5FFF_FFFF Secure region which maps to AHB Master Expansion 1 interface

To support TrustZone-Armv8M and allow software to map these peripherals to secure or non-secure address space, many peripherals are mapped twice and either an APB PPC or AHB PPC is then used to gate access to these peripherals. An FPGA Secure Privilege Control block and a Non-Secure Privilege Control block then provide controls to these PPC's.

For expansion AHB Slaves within the system, there is a Master Security Controller (MSC) added to each slave with an associated IDAU. The user has access to one of these interfaces via AHB from the user peripheral area of the design.

ROW ID	Address From	To	Size	Description	Port
Non-Secure Region					
1	0x4110_0000	0x4110_0FFF	4K	GPIO 0	
2	0x4110_1000	0x4110_1FFF	4K	GPIO 1	**AHB**
3	0x4110_2000	0x4110_2FFF	4K	GPIO 2	
4	0x4110_3000	0x4110_3FFF	4K	GPIO 3	
	0x4110_4000	0x411F_FFFF		Reserved	
5	0x4120_0000	0x4120_0FFF	4K	FPGA - SBCon I2C (Touch)	
6	0x4120_1000	0x4120_1FFF	4K	FPGA - SBCon I2C (Audio Conf)	
7	0x4120_2000	0x4120_2FFF	4K	FPGA - PL022 (SPI ADC)	
8	0x4120_3000	0x4120_3FFF	4K	FPGA - PL022 (SPI Shield0)	
9	0x4120_4000	0x4120_4FFF	4K	FPGA - PL022 (SPI Shield1)	**APB0**
10	0x4120_5000	0x4120_5FFF	4K	SBCon (I2C - Shield0)	
11	0x4120_6000	0x4120_6FFF	4K	SBCon (I2C - Shield1)	
12	0x4120_7000	0x4120_7FFF	4K	USER APB	
13	0x4120_8000	0x4120_8FFF	4K	FPGA - SBCon I2C (DDR4 EEPROM)	
	0x4120_9000	0x412F_FFFF		Reserved	
14	0x4130_0000	0x4130_0FFF	4K	FPGA - SCC registers	
15	0x4130_1000	0x4130_1FFF	4K	FPGA - I2S (Audio)	
16	0x4130_2000	0x4130_2FFF	4K	FPGA - IO (System Ctrl + I/O)	
17	0x4130_3000	0x4130_3FFF	4K	UART0 - UART_F[0]	
18	0x4130_4000	0x4130_4FFF	4K	UART1 - UART_F[1]	
19	0x4130_5000	0x4130_5FFF	4K	UART2 - UART_F[2]	
20	0x4130_6000	0x4130_6FFF	4K	UART3 - UART Shield 0	**APB1**
21	0x4130_7000	0x4130_7FFF	4K	UART4 - UART Shield 1	
22	0x4130_8000	0x4130_8FFF	4K	UART5 - UART_F[3]	
23	0x4130_9000	0x4130_9FFF	4K	USER APB	
24	0x4130_A000	0x4130_AFFF	4K	CLCD Config Reg	
25	0x4130_B000	0x4130_BFFF	4K	RTC	
	0x4130_C000	0x413F_FFFF		Reserved	
26	0x4140_0000	0x414F_FFFF	1M	Ethernet	**EAM**
27	0x4150_0000	0x415F_FFFF	1M	USB	
	0x4160_2000	0x416F_FFFF		Reserved	
28	0x4170_0000	0x4170_0FFF	4K	User APB0	
29	0x4170_1000	0x4170_1FFF	4K	User APB1	**APB (Mem)**
30	0x4170_2000	0x4170_2FFF	4K	User APB2	
31	0x4170_3000	0x4170_3FFF	4K	User APB3	
	0x4170_4000	0x4800_6FFF		Reserved	
32	0x4800_7000	0x4800_7FFF	4K	FPGA Non-Secure Privilege Control	
	0x4800_8000	0x4FFF_FFFF		Reserved	

ROW ID	Address From	Address To	Size	Description	Port
				Secure Region	
1	0x5110_0000	0x5110_0FFF	4K	GPIO 0	
2	0x5110_1000	0x5110_1FFF	4K	GPIO 1	**AHB**
3	0x5110_2000	0x5110_2FFF	4K	GPIO 2	
4	0x5110_3000	0x5110_3FFF	4K	GPIO 3	
	0x5110_4000	0x511F_FFFF		Reserved	
5	0x5120_0000	0x5120_0FFF	4K	FPGA - SBCon I2C (Touch)	
6	0x5120_1000	0x5120_1FFF	4K	FPGA - SBCon I2C (Audio Conf)	
7	0x5120_2000	0x5120_2FFF	4K	FPGA - PL022 (SPI ADC)	
8	0x5120_3000	0x5120_3FFF	4K	FPGA - PL022 (SPI Shield0)	
9	0x5120_4000	0x5120_4FFF	4K	FPGA - PL022 (SPI Shield1)	**APB0**
10	0x5120_5000	0x5120_5FFF	4K	SBCon (I2C - Shield0)	
11	0x5120_6000	0x5120_6FFF	4K	SBCon (I2C - Shield1)	
12	0x5120_7000	0x5120_7FFF	4K	USER APB	
13	0x5120_8000	0x5120_8FFF	4K	FPGA - SBCon I2C DRR4 EEPROM	
	0x5120_9000	0x512F_FFFF		Reserved	
14	0x5130_0000	0x5130_0FFF	4K	FPGA - SCC registers	
15	0x5130_1000	0x5130_1FFF	4K	FPGA - I2S (Audio)	
16	0x5130_2000	0x5130_2FFF	4K	FPGA - IO (System Ctrl + I/O)	
17	0x5130_3000	0x5130_3FFF	4K	UART0 - UART_F[0]	
18	0x5130_4000	0x5130_4FFF	4K	UART1 - UART_F[1]	
19	0x5130_5000	0x5130_5FFF	4K	UART2 - UART_F[2]	
20	0x5130_6000	0x5130_6FFF	4K	UART3 - UART Shield 0	**APB1**
21	0x5130_7000	0x5130_7FFF	4K	UART4 - UART Shield 1	
22	0x5130_8000	0x5130_8FFF	4K	UART5 - UART_F[3]	
23	0x5130_9000	0x5130_9FFF	4K	USER APB	
24	0x5130_A000	0x5130_AFFF	4K	CLCD Config Reg	
25	0x5130_B000	0x5130_BFFF	4K	RTC	
	0x5130_C000	0x513F_FFFF		Reserved	
26	0x5140_0000	0x514F_FFFF	1M	Ethernet	**EAM**
27	0x5150_0000	0x515F_FFFF	1M	USB	
	0x5160_0000	0x516F_FFFF		Reserved	
28	0x5170_0000	0x5170_0FFF	4K	User APB0	
29	0x5170_1000	0x5170_1FFF	4K	User APB1	**APB (Mem)**
30	0x5170_2000	0x5170_2FFF	4K	User APB2	
31	0x5170_3000	0x5170_3FFF	4K	User APB3	
	0x5170_4000	0x5800_8FFF		Reserved	
32	0x5800_7000	0x5800_7FFF	4K	BRAM Memory Protection Controller (MPC)	
33	0x5800_8000	0x5800_8FFF	4K	QSPI Memory Protection Controller (MPC)	**APB (Mem)**
34	0x5800_9000	0x5800_9FFF	4K	DDR4 Memory Protection Controller (MPC)	
	0x5800_8000	0x5FFFF_FFFF		Reserved	

Table 3-2 : FPGA Expansion Peripheral Map

Note: Reserved regions should not be accessed.

4 Programmers Model

4.1 CMSDK and SIE-200 components

This programmers model is supplemental to the CMSDK, SSE-200 Subsystem and SIE-200 documentation which covers many of the included components in more detail. Figure 3-1 : System Overview shows the connectivity of the system.

4.2 BRAM

Primary memory is 512KB of Internal FPGA SRAM, which is the default option for boot memory.

Size: 512KB FPGA BRAM
Address Range: 0x00000000 - 0x0007FFFF
Alias Range: 0x10000000 - 0x1007FFFF

4.3 QSPI

The secondary memory is 8MB of external Flash memory which is accessed via a QSPI interface in Read Only mode.

Size: 8MB Flash
Address Range: 0x28000000 - 0x287FFFFF
Alias Range: 0x38000000 - 0x387FFFFF

4.4 DDR4

The SMM also includes 2GB of External DDR4 memory

Size: 2GB DDR4
Address Range: 0x60000000 - 0xDFFFFFFF

4.5 AHB GPIO

The SMM uses four CMSDK AHB GPIO blocks, each providing 16 bits of IO. These are connected to the two Arduino compatible headers shield 0 and 1 as follows:

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]

Table 4-1 : GPIO Mapping

The GPIO alternative function lines select whether or not peripherals or GPIOs are available on each pin. See section 8 - Shield Support for mappings.

4.6 SPI (Serial Peripheral Interface)

The SMM implements three PL022 SPI modules:

- One general purpose SPI module (SPI ADC) is used for communication with an onboard ADC. The analog pins of the Shield headers are connected to the input channels of the ADC.

- Two general purpose SPI modules connect to the Shield headers and provide an SPI interface on each header. These are alt-functions on the GPIO ports. See section 8 - Shield Support for mappings.

4.7 SBCon (I²C)

The SMM implements five SBCon serial modules:

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

- One SBCon module to configure the audio controller.

- Two general purpose SBCon modules that connect to the Shield0 and Shield1 and provide an I2C interface on each header. These are alt-functions on the GPIO ports. See section 8 - Shield Support for mappings.

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

The Self-test program provided with the MPS3 includes example code for the color LCD module control and Audio interfaces.

Here is a brief description of its operation, SBCon consists of 3 registers:

```
`define SB_CONTROL 6'b000000 // 0 status

`define SB_CONTROLS 6'b000000 // 0 set outputs SDA or SCL

`define SB_CONTROLC 6'b000001 // 4 clear outputs SDA or SCL
```

SCL is bit 0 of the SB_CONTROLS and SB_CONTROLC registers. SDA_OUT_ENABLE is bit 1.

The SB_CONTROL reads back SDA from bit 1 and SCL from bit 0. There is no automation, pipe-lining or flow control.

4.8 UART

The SMM implements six CMSDK UARTs:

- UART 0 – FPGA_UART0

- UART 1 – FPGA_UART1

- UART 2 – FPGA_UART2

- UART 3 - Shield 0

- UART 4 - Shield 1

- UART 5 - FPGA_UART3

UART 3 and 4 are alt-functions on the GPIO ports. See section 8 - Shield Support for mappings.

4.9 Color LCD parallel interface

The color LCD module has two interfaces:

- Parallel bus for sending image data to the LCD.

- I^{2}C to transfer data input from the touch screen.

This 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. (Schematic listed in the reference section). The Keil display board contains an AM240320LG display panel and uses a Himax HX8347-D LCD controller.

Self-test provided with the MPS3 includes drivers and example code for both of these interfaces.

The control and data registers for the CLCD interface are listed in Table 4-2.

Address	Name	Type	Information
0x4130A000	CHAR_COM	Write command, read busy status	A write to this address will cause a write to the LCD command register. A read from this address will cause a read from the LCD busy register.
0x4130A004	CHAR_DAT	Write data RAM, Read data RAM	A write to this address will cause a write to the LCD data register. A read from this address will cause a read from the LCD data register.
0x4130A008	CHAR_RD	Read captured data from an earlier read command	Bits [7:0] contain the data from last request read, valid only when bit 0 is set in CHAR_RAW. Bits [31:8] are reserved.
0x4130A00C	CHAR_RAW	Write to reset access complete flag, Read to determine if data in CHAR_RD is valid	Bit 0 indicates Access Complete (write 0 to clear). The bit is set if read data is valid. Bits [31:1] are reserved.
0x4130A010	CHAR_MASK	Write interrupt mask	Set bit 0 to 1 to enable Access Complete to generate an interrupt.
0x4130A014	CHAR_STAT	Read status	Bit 0 is the state of Access Complete ANDed with the CHAR_MASK

Table 4-2 : LCD control and data registers

4.10 Ethernet

The SMM design connects to an SMSC LAN9220 device through a static memory interface.

The self-test program includes example code for a simple loopback operation.

4.11 USB

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

The self-test program includes example code for a simple loopback operation.

4.12 Real Time Clock, RTC

The SMM uses PL031 PrimeCell *Real Time Clock Controller* (RTC). A counter in the RTC is incremented every second. The RTC can therefore be used as a basic alarm function or long time-base counter.

4.13 Audio I²S

The SMM has a single I2S module directly connected to the MPS3 back panel audio sockets.

4.14 Audio Configuration

The SMM implements a simple SBCon interface based on I²C. It is used to configure the Cirrus Logic Low Power Codec with Class D Speaker Driver, CS42L52 part on the MPS3 board.

4.15 FPGA system control and I/O

The SMM implements an FPGA system control block.

Address	Name	Information
0x41302000 0x51302000	FPGAIO->LED0	LED connections [31:10] : Reserved [9:0] : LED
0x41302004 0x51302004	RESERVED	
0x41302008 0x51302008	FPGAIO->BUTTON	Buttons [31:2] : Reserved [1:0] : Buttons
0x4130200C 0x5130200C	RESERVED	
0x41302010 0x51302010	FPGAIO->CLK1HZ	1Hz up counter
0x41302014 0x51302014	FPGAIO->CLK100HZ	100Hz up counter
0x41302018 0x51302018	FPGAIO->COUNTER	Cycle Up Counter Increments when 32-bit prescale counter reach zero and automatically reloads.
0x4130201C 0x5130201C	FPGAIO->PRESCALE	Bit[31:0] – reload value *for* prescale counter.
0x41302020 0x51302020	FPGAIO->PSCNTR	32-bit Prescale counter – current value of the pre-scaler counter. The Cycle Up Counter increment when the prescale down counter reach 0. The pre-scaler counter is reloaded with PRESCALE after reaching 0.
0x41302024 0x51302024	RESERVED	
0x41302028 0x51302028	FPGAIO->SWITCH	Switches [31:8] : Reserved [7:0] : Switches
0x4130204C 0x5130204C	FPGAIO->MISC	Misc control [31:3] : Reserved [2] : SHIELD1_SPI_nCS [1] : SHIELD0_SPI_nCS [0] : ADC_SPI_nCS

Table 4-3 : System Control and I/O Memory Map

4.16 Serial Communication Controller (SCC)

The SMM implements communication between the microcontroller and the FPGA system through an SCC interface.
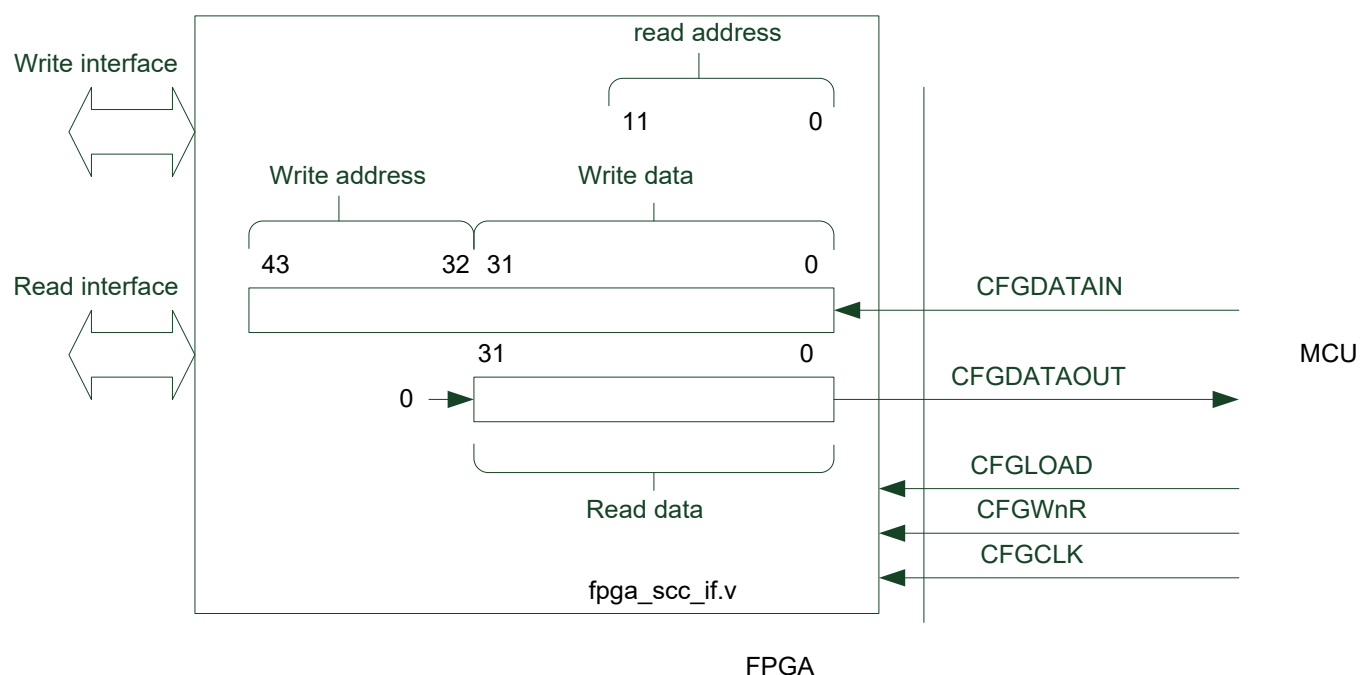
Figure 4-1 : Diagram of the SCC Interface

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

Address	Name	Information
0x000	CFG_REG0	Bits [31:1] Reserved
		Bits [0] Memory Remap (0-Default, 1 – Option 1)
0x004	CFG_REG1	32bit DATA [r/w]
0x008	CFG_REG2	Bits [31:1] Reserved
		Bits [0] QSPI Select signal
0x00C	CFG_REG3	Bits [31:0] Reserved
0x010	CFG_REG4	Bits [31:4] Reserved
		Bits [3:0] Board Revision [r]
0x014	CFG_REG5	Bits [31:0] ACLK Frequency in Hz
0x018	CFG_REG6	Bits [3:0] Clock divider for BRAM (1, 2, 4, 8)
0x01C – 0x09C	RESERVED	-
0x0A0	SYS_CFGDATA_RTN	32bit DATA [r/w]
0x0A4	SYS_CFGDATA_OUT	32bit DATA [r/w]
0x0A8	SYS_CFGCTRL	Bit[31] Start (generates interrupt on write to this bit)
		Bit[30] R/W access
		Bits[29:26] Reserved
		Bits[25:20] Function value
		Bits[19:12] Reserved
		Bits[11:0] Device (value of 0/1/2 for supported clocks)

Address	Name	Information
0x0AC	SYS_CFGSTAT	Bits[31:2] Reserved
		Bit[1] Error
		Bit[0] Complete
0x0B0 – 0xFF4	RESERVED	-
0xFF8	SCC_AID	SCC AID register is read only
		Bits[31:24] FPGA build number
		Bits[23:20] V2M-MPS3 target board revision (A = 0, B = 1, C = 2)
		Bits[19:8] Reserved
		Bits[7:0] Number of SCC configuration register
0xFFC	SCC_ID	SCC ID register is read only
		Bits[31:24] Implementer ID: 0x41 = Arm
		Bits[23:20] Reserved
		Bits[19:16] IP Architecture: 0x4 =AHB
		Bits[15:4] Primary part number: 545 = AN545
		Bits[3:0] Reserved

Table 4-4 : SCC Register memory map

5 Clock architecture

The following tables list clocks entering and generated by the SMM.

5.1 Source clocks

The following clocks are inputs to the system:

Input Pin	Frequency	Note
OSCCLK[0]	24MHz	Constant 24MHz reference, used for RTC and timers.
OSCCLK[1]	32MHz	ACLK, main clock used to clock SSE-200 subsystem. Frequency can be changed in an545_v2.txt
OSCCLK[2]	50MHz	Reserved
OSCCLK[3]	50MHz	GPUCLK, aux clock used to generate PERIPH_CLK for user space. Frequency can be changed in an545_v2.txt
OSCCLK[4]	24.576MHz	AUDCLK, clock used to clock I2S audio module. Frequency can be changed in an545_v2.txt
OSCCLK[5]	23.75MHz	HDLCDCLK, clock can be used to clock video module. Frequency can be changed in an545_v2.txt
c0_sys_clk_p/n	100MHz	DDR4_REF_CLK, Constant Differential input clock for DDR4 controller

Table 5-1 : Source clocks

5.2 User clocks

The following clocks are generated internally from the source clocks:

Clock	Source	Frequency	Note
MAINCLK	OSCCLK[1]	32MHz	Main clock, used to clock user's CMSDK subsystem
BRAMCLK	OSCCLK[1]	1:1/2/4/8 of MAINCLK	Synchronous clock used to clock BRAM
PERIF_CLK	OSCCLK[3]	50MHz	AUX clock.
AUDMCLK	AUDCLK	12.29MHz	Clock used to clock I2S audio module
AUDSCLK	AUDCLK	3.07MHz	Clock used to clock I2S audio module
SDMCLK	REFCLK24MHZ	50MHz	Additional clock for SDCard or eMMC controllers
CLK32KHZ	REFCLK24MHZ	32kHz	RTC clock
CLK100HZ	REFCLK24MHZ	100Hz	RTC clock
CLK1HZ	REFCLK24MHZ	1Hz	RTC clock
CFGCLK	CFG_CLK	Set by MCC	SCC register clock from MCC

Table 5-2 : Generated internal clocks

6 FPGA Secure Privilege Control

The SSE-200 Subsystem's Secure Privilege and Non-Secure Privilege Control Block can provide expansion security control signals to control the various security gating units within the subsystem. The following table lists the connectivity of system security extension signal.

Components Name	Components signals	Security Expansion Signals
USER MSC	msc_irq	S_MSCEXP_STATUS[0]
	msc_irq_clear	S_MSCEXP_CLEAR[0]
	cfg_nonsec	NS_MSCEXP[0]
APB PPC EXP 0	apb_ppc_irq	S_APBPPCEXP_STATUS[0]
	apb_ppc_clear	S_APBPPCEXP_CLEAR[0]
	cfg_sec_resp	SEC_RESP_CFG
	cfg_non_sec	APB_NS_PPCEXP0[15:0]
	cfg_ap	APB_P_PPCEXP0[15:0]
APB PPC EXP 1	apb_ppc_irq	S_APBPPCEXP_STATUS[1]
	apb_ppc_clear	S_APBPPCEXP_CLEAR[1]
	cfg_sec_resp	SEC_RESP_CFG
	cfg_non_sec	APB_NS_PPCEXP1[15:0]
	cfg_ap	APB_P_PPCEXP1[15:0]
APB PPC EXP 2	apb_ppc_irq	S_APBPPCEXP_STATUS[2]
	apb_ppc_clear	S_APBPPCEXP_CLEAR[2]
	cfg_sec_resp	SEC_RESP_CFG
	cfg_non_sec	APB_NS_PPCEXP2[15:0]
	cfg_ap	APB_P_PPCEXP2[15:0]
AHB PPC EXP 0	ahb_ppc_irq	S_AHBPPCEXP_STATUS[0]
	ahb_ppc_clear	S_AHBPPCEXP_CLEAR[0]
	cfg_sec_resp	SEC_RESP_CFG
	cfg_non_sec	AHB_NS_PPCEXP0[15:0]
	chg_ap	AHB_P_PPCEXP0[15:0]
AHB PPC EXP 1	ahb_ppc_irq	S_AHBPPCEXP_STATUS[1]
	ahb_ppc_clear	S_AHBPPCEXP_CLEAR[1]
	cfg_sec_resp	SEC_RESP_CFG
	cfg_non_sec	AHB_NS_PPCEXP1[15:0]
	chg_ap	AHB_P_PPCEXP1[15:0]
MPC SSRAM	secure_error_irq	S_MPCEXP_STATUS[2]

Table 6-1 : Security Expansion signals connectivity.

The following table lists the peripherals that are controlled by APB PPC EXP 0.
Each APB <n> interface is controlled by APB_NS_PPCEXP0[n] and APB_P_PPCEXP0[n].

APB PPC EXP 0 Interface Number <n>	Name
0	SSRAM Memory Protection Controller (MPC)
1	QSPI Memory Protection Controller (MPC)
2	DDR4 Memory Protection Controller (MPC)
15:3	Reserved

Table 6-2 : Peripherals Mapping of APB PPC EXP 0

The following table lists the peripherals that are controlled by APB PPC EXP 1.
Each APB <n> interface is controlled by APB_NS_PPCEXP1[n] and APB_P_PPCEXP1[n].

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 Shield0)
4	FPGA - PL022 (SPI Shield1)
5	SBCon (I2C - Shield0)
6	SBCon (I2C – Shield1)
7	Reserved
8	I2C DDR4 EPROM
15:9	Reserved

Table 6-3 : Peripherals Mapping of APB PPC EXP 1

The following table lists the peripherals that are controlled by APB PPC EXP 2.
Each APB <n> interface is controlled by APB_NS_PPCEXP2[n] and APB_P_PPCEXP2[n].

APB PPC EXP 0 Interface Number <n>	Name
0	FPGA - SCC registers
1	FPGA - I2S (Audio)
2	FPGA - IO (System Ctrl + I/O)
3	UART0 - UART_F[0]
4	UART1 - UART_F[1]
5	UART2 - UART_F[2]
6	UART3 - UART Shield 0
7	UART4 - UART Shield 1
8	UART5 - UART_F[3]
9	Reserved
10	CLCD
11	RTC
15:12	Reserved

Table 6-4 : Peripherals Mapping of APB PPC EXP 2

The following table lists the peripherals that are controlled by AHB PPC EXP 0.
Each APB <n> interface is controlled by AHB_NS_PPCEXP0[n] and AHB_P_PPCEXP0[n].

AHB PPC EXP 0 Interface Number <n>	Name
0	GPIO_0
1	GPIO_1
2	GPIO_2
3	GPIO_3
4	USB and Ethernet
5	User AHB interface 0
6	User AHB interface 1
7	User AHB interface 2
15:8	Reserved

Table 6-5 : Peripherals Mapping of AHB PPC EXP 0

7 Interrupt Map

The Interrupts in the FPGA subsystem extend the SSE-200 Interrupt map by adding to the expansion area as follows:

Interrupt Input	Interrupt Source
IRQ[32]	UART 0 Receive Interrupt
IRQ[33]	UART 0 Transmit Interrupt
IRQ[34]	UART 1 Receive Interrupt
IRQ[35]	UART 1 Transmit Interrupt
IRQ[36]	UART 2 Receive Interrupt
IRQ[37]	UART 2 Transmit Interrupt
IRQ[38]	UART 3 Receive Interrupt
IRQ[39]	UART 3 Transmit Interrupt
IRQ[40]	UART 4 Receive Interrupt
IRQ[41]	UART 4 Transmit Interrupt
IRQ[42]	UART 0 Combined Interrupt
IRQ[43]	UART 1 Combined Interrupt
IRQ[44]	UART 2 Combined Interrupt
IRQ[45]	UART 3 Combined Interrupt
IRQ[46]	UART 4 Combined Interrupt
IRQ[47]	UART Overflow (0, 1, 2, 3, 4 & 5)
IRQ[48]	Ethernet
IRQ[49]	FPGA Audio I2S
IRQ[50]	Touch Screen
IRQ[51]	Unused
IRQ[52]	SPI ADC
IRQ[53]	SPI (Shield 0)
IRQ[54]	SPI (Shield 1)
IRQ[67:55]	Unused
IRQ[68]	GPIO 0 Combined Interrupt
IRQ[69]	GPIO 1 Combined Interrupt
IRQ[70]	GPIO 2 Combined Interrupt
IRQ[71]	GPIO 3 Combined Interrupt
IRQ[87:72]	GPIO 0 individual interrupts
IRQ[103:88]	GPIO 1 individual interrupts
IRQ[119:104]	GPIO 2 individual interrupts
IRQ[123:120]	GPIO 3 individual interrupts
IRQ[124]	UART 5 Receive Interrupt
IRQ[125]	UART 5 Transmit Interrupt
IRQ[126]	UART 5 Combined Interrupt
IRQ[127]	HDCLCD Interrupt

Table 7-1 : FPGA Expansion Interrupt Map.

7.1 UARTS Interrupts

There are six CMSDK UARTs in the system, and each has the following interrupt pins:

- TXINT

- RXINT

- TXOVRINT

- EXOVRINT

- UARTINT

The TXINT, RXINT and UARTINT interrupt signal of each UART drive a single interrupt input of the Cortex-M33 CPU. In addition, the TXOVERINT and EXOVRINT interrupt signals of all six UARTs, twelve signals in all, are logically ORed together to drive IRQ[47].

8 Shield Support

This SMM support external shield devices. To enable the Shield support, two SPI, two UART and two I2C interfaces are multiplexed with GPIO over the Shields Headers.
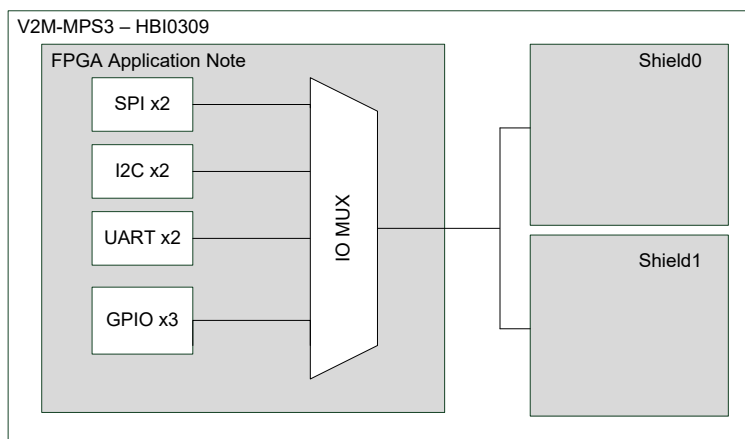
Figure 8-1 : Shield Device Expansion

Multiplexing is controlled by the alternative function output from the associated GPIO Register.

MPS3	AN545	Alt Function	Alt Peripheral	Alt Description
SH0_IO0	GPIO0_0	SH0_RXD	UART3	SH0 UART
SH0_IO1	GPIO0_1	SH0_TXD		
SH0_IO2	GPIO0_2	-	-	-
SH0_IO3	GPIO0_3	-	-	-
SH0_IO4	GPIO0_4	-	-	-
SH0_IO5	GPIO0_5	-	-	-
SH0_IO6	GPIO0_6	-	-	-
SH0_IO7	GPIO0_7	-	-	-
SH0_IO8	GPIO0_8	-	-	-
SH0_IO9	GPIO0_9	-	-	-
SH0_IO10	GPIO0_10	SH0_nCS	SPI3	SH0 SPI
SH0_IO11	GPIO0_11	SH0_DO		
SH0_IO12	GPIO0_12	SH0_DI		
SH0_IO13	GPIO0_13	SH0_CLK		
SH0_IO14	GPIO0_14	SH0_SDA	I2C2	SH0 I2C
SH0_IO15	GPIO0_15	SH0_SCL		
SH0_IO16	GPIO2_0	-	-	-
SH0_IO17	GPIO2_1	-	--	-

MPS3	AN545	Alt Function	Alt Peripheral	Alt Description
SH1_IO0	GPIO1_0	SH1_RXD	UART4	SH1 UART
SH1_IO1	GPIO1_1	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	SH1_nCS	SPI4	SH1 SPI
SH1_IO11	GPIO1_11	SH1_DO		
SH1_IO12	GPIO1_12	SH1_DI		
SH1_IO13	GPIO1_13	SH1_CLK		
SH1_IO14	GPIO1_14	SH1_SDA	I2C3	SH1 I2C
SH1_IO15	GPIO1_15	SH1_SCL		
SH1_IO16	GPIO2_2	-	-	-
SH1_IO17	GPIO2_3	-	-	-

Table 8-1 : Shield Alternative Function Pinout

9 Configurations

9.1 SSE-200 Subsystem

The SSE-200 Subsystem has configurable options. These options are documented in *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual*, section A.8 Top-level parameters. Where this application note uses a non-default value, the configuration settings used are listed below:

Parameter	Implemented Values	Default Values	Description
HAS_CRYPTO	No {0}	Yes {1}	Include CryptoCell 312
CPU0_FPU	Yes {1}	No {0}	CPU0 Floating Point Unit Present
CPU0_DSP	No {0}	No {0}	CPU0 DSP Extension instructions present
CPU0_ICACHESIZE	2KB {11}	2KB{11}	CPU0 Instruction cache size
CPU1_FPU	Yes {1}	Yes {1}	CPU1 Floating Point Unit Present
CPU1_DSP	Yes {1}	Yes {1}	CPU1 DSP Extension instructions present
CPU1_ICACHESIZE	2KB{11}	2KB{11}	CPU1 Instruction cache size
CPU0WAIT_RST	1	0	CPU wait at boot '0' boot normally, '1' wait at boot. The MCC controller releases CPU0WAIT by writing to a register after user code is loaded to system memory at startup.
CPU0_EXP_NUMIRQ	97	64	Specifies the number of expansion interrupt. This means that the M33 NVIC has 92+32 = 124 interrupts.
CPU1_EXP_NUMIRQ	97	64	Specifies the number of expansion interrupt. This means that the M33 NVIC has 92+32 = 124 interrupts.
CPU0_EXP_IRQDIS	0	CPU0_EXP_IRQDIS_DEF [CPU0_EXP_NUMIRQ−1:0]	When a bit is set to 1, it disables the corresponding interrupt logic on CPU element 0.
CPU1_EXP_IRQDIS	0	CPU1_EXP_IRQDIS_DEF [CPU1_EXP_NUMIRQ−1:0]	When a bit is set to 1, it disables the corresponding interrupt logic on CPU element 1.

Table 9-1 : SSE-200 configuration option

9.2 Cortex-M33

Refer to document *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual*, section A.8 Top-level parameters for the parameters used in SSE-200 Subsystem to configure the Cortex-M33 CPU cores.

10 Arm Custom Instructions

Both Cortex-M33 cores supports Arm Custom Instructions (ACIs) and implements the Custom Datapath Extension (CDE) for Armv8-M. For more information on ACI and CDE, please refer to Paragraph B7 of *Arm® Cortex®-M33 Processor Technical Reference Manual (Version 100230_0100_03_en),* and *Arm Custom Instructions: Enabling Innovation and Greater Flexibility on Arm (White Paper from Feb 2020).*

In the current design:

Core 0 – implements connection to user modifiable example design of CDE and FPCDE modules.

Core 1 – implements connection to user fixed example design of CDE and FPCDE modules.
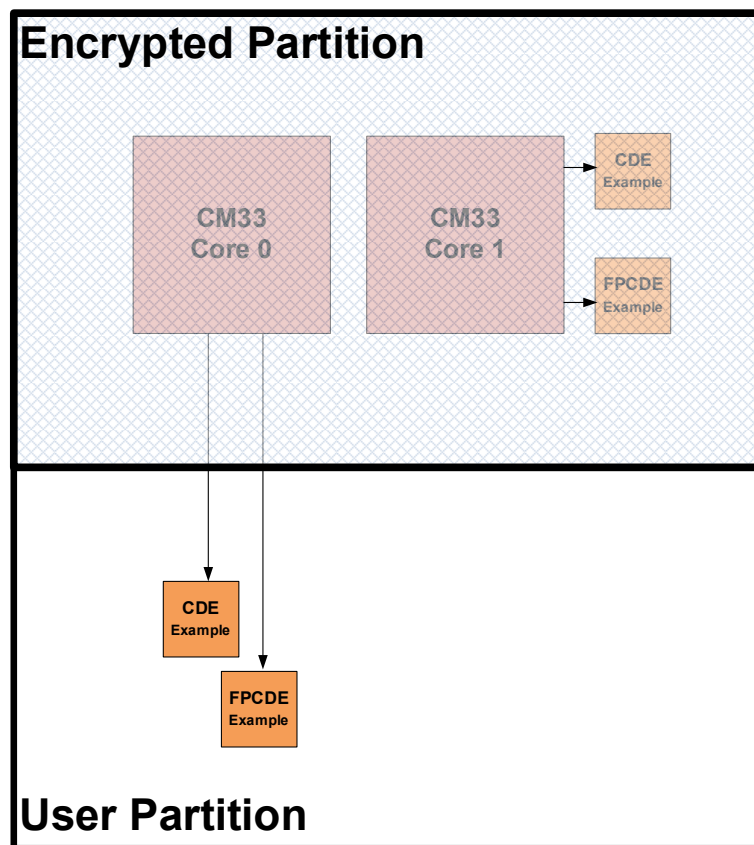
Figure 10-1 CDE modules in design

You will be able to find example implementations of CDE and FPCDE in the release bundle path:

Luna\Logical\Resources\mps3_user_peripheral\AN545\user_wrapper\core_cde.v and
Luna\Logical\Resources\mps3_user_peripheral\AN545\user_wrapper\fpu_cde.v

Please refer to page 12-45 for further instructions on how to implement modified design. Selftest and Demo software can provide you with code examples for interaction with current implementations of CDE and FPCDE modules.

11 ZIP Bundle Description

11.1 Overall Structure

The accompanying `.zip` bundle contains:

`Boardfiles/`	This 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.
`Docs/`	This Application Note Document.
`Luna/`	This Directory contains the RTL and scripts needed to rebuild User Partition
`Software/`	An example Keil software project, that can be run on the SE Cortex-M33 to test supported board peripherals and interfaces.
`Licence.pdf`	Generic EULA
`readme.txt`	Short description of the project
`revision_history.txt`	Revision history

11.2 MPS3 Board Revision and Support

11.2.1 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:

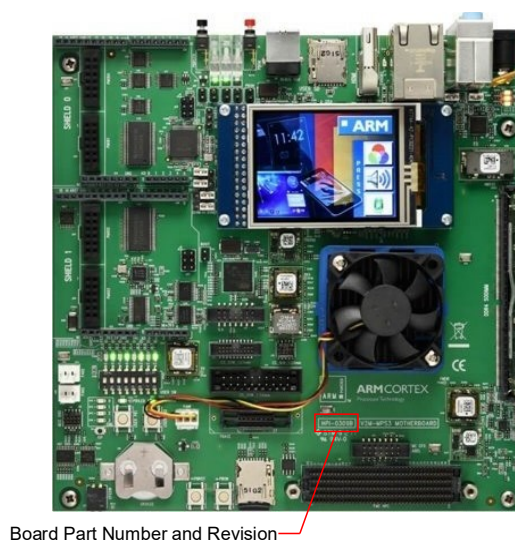Board Part Number and Revision

Figure 7-1 : MPS3 Board Revision Identifier

In this example the part number is "HBI0309**B".** The last letter at the end of the part number denotes the board revision. The illustration shows a revision **B** board.

11.2.2 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 section 11.2.1 for further details on how to identify the board part number and revision).

Note

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

11.3 Bundle Directory Tree/Structure

11.3.1 Boardfiles

```
|-- Boardfiles
|   |-- MB
|   |   |-- BRD_LOG.TXT
|   |   |-- HBI0309B
|   |   |   |-- AN545
|   |   |   |   |-- 545_t_0.bit
|   |   |   |   |-- 545_u_0.bit
|   |   |   |   |-- 545_uc_0.bit
|   |   |   |   |-- an545_v0.txt
|   |   |   |   `-- images.txt
|   |   |   |-- board.txt
|   |   |   `-- mbb_v141.ebf
|   |   `-- HBI0309C
|   |       |-- AN545
|   |       |   |-- 545_t_0.bit
|   |       |   |-- 545_u_0.bit
|   |       |   |-- 545_uc_0.bit
|   |       |   |-- an545_v0.txt
|   |       |   `-- images.txt
|   |       |-- board.txt
|   |       `-- mbb_v141.ebf
|   |-- SOFTWARE
|   |   |-- an545_dm.axf
|   |   `-- an545_st.axf
|   `-- config.txt
```

11.3.2 Docs

```
|-- Docs
|   `-- AN545
|       `-- DAI0545A_example_cortex_m33_subsystem_with_CDE_for_mps3.pdf
```

11.3.3 Luna/FPGA

```
|-- Luna
|   |-- FPGA
|   |   `-- AN545
|   |       |-- blackboxes
|   |       |   |-- AhbSToAxi.edf
|   |       |   |-- nic400_mem0.edf
|   |       |   |-- nic400_mem1.edf
|   |       |   `-- nic400_peripheral_user.edf
|   |       `-- smm_toplevel
|   |           `-- xilinx
|   |               |-- constraints
|   |               |   |-- mps3_fpga_top_placement.xdc
|   |               |   |-- mps3_fpga_top_timing.xdc
|   |               |   |-- mps3_fpga_top_timing_additional.xdc
|   |               |   `-- mps3_fpga_user.xdc
|   |               |-- netlist_user
|   |               |-- scripts
|   |               |   |-- user_pr_flow.bat
|   |               |   |-- user_pr_flow.scr
|   |               |   |-- user_pr_impl.tcl
|   |               |   `-- user_pr_synth.tcl
|   |               `-- user_pr_checkpoints
|   |                   |-- mps3_fpga_top_static_base.dcp
|   |                   `-- mps3_system_core_synth.dcp
```

11.3.4 Luna/Logical

```
|-- Luna
|   `-- Logical
|       |-- AN545_SMM_SSE200_CDE
|       |   `-- fpga_options_defs.vh
|       `-- Resources
|           |-- ip_libraries
|           |   `-- CMSDK_r1p1
|
|           |-- mps3_user_peripheral
|           |   `-- AN545
|           |       |-- memory_subsystem
|           |       |   |-- mps3_bram_qspi_memsubsys.v
|           |       |   |-- mps3_ddr_peripheral_subsys.v
|           |       |   |-- mps3_mem_apb_addr_decode.v
|           |       |   `-- mps3_peripheral_mem_wrapper.v
|           |       |-- peripheral_subsystem
|           |       |   |-- mps3_ahb_addr_decode.v
|           |       |   |-- mps3_apb_addr_decode_ss0.v
|           |       |   |-- mps3_apb_addr_decode_ss1.v
|           |       |   |-- mps3_gpio_altfunc_2_exp_mux.v
|           |       |   |-- mps3_interrupt_assignment.v
|           |       |   |-- mps3_usb_eth.v
|           |       |   |-- mps3_user_ahb_subsystem.v
|           |       |   |-- mps3_user_apb_subsystem_0.v
|           |       |   |-- mps3_user_apb_subsystem_1.v
|           |       |   `-- mps3_user_periph_wrapper.v
|           |       |-- small_blocks
|           |       |   `-- resets.v
|           |       `-- user_wrapper
|           |           |-- mps3_fpga_user.v
|           |           |-- core_cde.v
|           |           `-- fpu_cde.v
|           `-- smm_common
|               `-- AN545
|                   |-- apb_i2s
|                   |   |-- apb_i2s_top.v
|                   |   |-- i2s_apb_if.v
|                   |   |-- i2s_async_dest.v
|                   |   |-- i2s_async_src.v
|                   |   |-- i2s_fifo_4.v
|                   |   |-- i2s_if.v
|                   |   |-- i2s_pulse_sync.v
|                   |   `-- i2s_sync_cell.v
|                   |-- blackboxes_stubs
|                   |   |-- AhbSToAxi.v
```

```
|           |   |-- nic400_mem0.v
|           |   |-- nic400_mem1.v
|           |   `-- nic400_peripheral_user.v
|           |-- cmsdk_ahb_to_extmem16_psram
|           |   `-- verilog
|           |       |-- cmsdk_ahb_to_extmem16_ahb_fsm_psram.v
|           |       |-- cmsdk_ahb_to_extmem16_mem_fsm_psram.v
|           |       `-- cmsdk_ahb_to_extmem16_psram.v
|           |-- ds703_scc_r0p3
|           |   `-- scc_mps3.v
|           |-- fpga
|           |   |-- CharLCDI.v
|           |   |-- Clock_dev_1_n.v
|           |   |-- Clock_gen_1_100_32K.v
|           |   |-- MicroToAhb.v
|           |   |-- RESETsync.v
|           |   |-- RESETsyncp_dly.v
|           |   |-- SBCon.v
|           |   |-- clcd_reg.v
|           |   |-- fpga_io_regs_ard.v
|           |   `-- fpga_rst_sync.v
|           |-- pl022_ssp
|           |   `-- verilog
|           |       |-- Ssp.v
|           |       |-- SspApbif.v
|           |       |-- SspDMA.v
|           |       |-- SspDataStp.v
|           |       |-- SspDefs.v
|           |       |-- SspIntGen.v
|           |       |-- SspMTxRxCntl.v
|           |       |-- SspRegCore.v
|           |       |-- SspRevAnd.v
|           |       |-- SspRxFCntl.v
|           |       |-- SspRxFIFO.v
|           |       |-- SspRxRegFile.v
|           |       |-- SspSTxRxCntl.v
|           |       |-- SspScaleCntr.v
|           |       |-- SspSynctoPCLK.v
|           |       |-- SspSynctoSSPCLK.v
|           |       |-- SspTest.v
|           |       |-- SspTxFCntl.v
|           |       |-- SspTxFIFO.v
|           |       |-- SspTxLJustify.v
|           |       `-- SspTxRegFile.v
|           `-- rtc_pl031
|               `-- verilog
|                   |-- Rtc.v
|                   |-- RtcApbif.v
|                   |-- RtcControl.v
|                   |-- RtcCounter.v
|                   |-- RtcInterrupt.v
|                   |-- RtcParams.v
|                   |-- RtcRevAnd.v
|                   |-- RtcSynctoPCLK.v
|                   `-- RtcUpdate.v
```

11.3.5 **Software/Demo**

```
|-- Software
|   |-- demo
|   |   |-- Build_Keil
|   |   |   |-- Dbg_Ext_RAM.ini
|   |   |   |-- Dbg_Ext_RAM_eMMC.ini
|   |   |   |-- DebugConfig
|   |   |   |   `-- Selftest_AN524_SSE-200-MPS3.dbgconf
|   |   |   |-- EventRecorderStub.scvd
|   |   |   |-- an545_dm.axf
|   |   |   |-- demo.uvoptx
|   |   |   |-- demo.uvprojx
|   |   |   `-- move.bat
|   |   |-- CDE
|   |   |   |-- popcnt.c
|   |   |   |-- popcnt.h
|   |   |   |-- runcrc.c
|   |   |   `-- runcrc.h
|   |   |-- apclcd
|   |   |   `-- CDE_bitmap.c
|   |   |-- apmain
|   |   |   |-- common.c
|   |   |   |-- common.h
|   |   |   |-- main.c
|   |   |   |-- retarget.c
|   |   |   |-- sys_sri.c
|   |   |   |-- uart_stdout.c
|   |   |   `-- uart_stdout.h
|   |   |-- cmsis
|   |   |   |-- CMSIS
|   |   |   |   `-- Include
|   |   |   |       |-- arm_common_tables.h
|   |   |   |       |-- arm_const_structs.h
|   |   |   |       |-- arm_math.h
|   |   |   |       |-- cmsis_armcc.h
|   |   |   |       |-- cmsis_armclang.h
|   |   |   |       |-- cmsis_compiler.h
|   |   |   |       |-- cmsis_gcc.h
|   |   |   |       |-- core_armv8mbl.h
|   |   |   |       `-- core_armv8mml.h
|   |   |   `-- Device
|   |   |       |-- Include
|   |   |       |   |-- CMSDK_IoT
|   |   |       |   |   |-- CMSDK_IoT.h
|   |   |       |   |   `-- system_CMSDK_IoT.h
|   |   |       |   `-- CMSDK_driver.h
|   |   |       `-- Source
|   |   |           |-- CMSDK_IoT
|   |   |           |   |-- partition_CMSDK_ARMv8MBL.h
|   |   |           |   |-- startup_CMSDK_IoT.s
|   |   |           |   `-- system_CMSDK_IoT.c
|   |   |           `-- CMSDK_driver.c
|   |   `-- v2m_mps3
|   |       |-- Font_16x24_h.h
|   |       |-- Font_6x8_h.h
|   |       |-- GLCD_MPS3.c
|   |       |-- GLCD_MPS3.h
|   |       |-- SMM_MPS3.h
```

11.3.6 **Software/Selftest**

```
`-- selftest
    |-- Build_Keil
    |   |-- Dbg_Ext_RAM.ini
    |   |-- Dbg_Ext_RAM_eMMC.ini
    |   |-- DebugConfig
    |   |   `-- Selftest_AN524_SSE-200-MPS3.dbgconf
    |   |-- EventRecorderStub.scvd
    |   |-- an545_st.axf
    |   |-- move.bat
    |   |-- objects
    |   |   `-- selftest.axf
    |   |-- selftest_mpb.sct
    |   |-- selftest_mpb.uvoptx
    |   `-- selftest_mpb.uvprojx
    |-- CDE
    |   |-- core_CDE.c
    |   |-- core_CDE.h
    |   |-- fpu_CDE.c
    |   `-- fpu_CDE.h
    |-- apaaci
    |   |-- apaaci.c
    |   `-- apaaci.h
    |-- apclcd
    |   |-- CDE_image.c
    |   |-- apclcd.c
    |   |-- apclcd.h
    |   |-- flyer.c
    |   |-- intro.c
    |   `-- intro_AN524.c
    |-- apgpio
    |   |-- apgpio.c
    |   `-- apgpio.h
    |-- aplan
    |   |-- aplan.c
    |   `-- aplan.h
    |-- apleds
    |   |-- apleds.c
    |   `-- apleds.h
    |-- apmain
    |   |-- common.c
    |   |-- common.h
    |   |-- main.c
    |   |-- retarget.c
    |   |-- sys_sri.c
    |   |-- uart_stdout.c
    |   `-- uart_stdout.h
    |-- apmem
    |   |-- apmem.c
    |   |-- apmem.h
    |   `-- apmemsup.s
    |-- apqspi
    |   |-- apqspi.c
    |   `-- apqspi.h
    |-- aprtc
    |   |-- aprtc.c
    |   `-- aprtc.h
    |-- apssp
    |   |-- apssp.c
    |   `-- apssp.h
    |-- aptimer
    |   |-- aptimer.c
    |   `-- aptimer.h
    |-- aptsc
    |   |-- aptsc.c
    |   `-- aptsc.h
    |-- apuart
    |   |-- apuart.c
    |   `-- apuart.h
    |-- apusb
    |   |-- apusb.c
    |   `-- apusb.h
    |-- cmsis
    |   |-- CMSIS
```

```
|    |    |     `-- Include
|    |    |          |-- arm_common_tables.h
|    |    |          |-- arm_const_structs.h
|    |    |          |-- arm_math.h
|    |    |          |-- cmsis_armcc.h
|    |    |          |-- cmsis_armclang.h
|    |    |          |-- cmsis_compiler.h
|    |    |          |-- cmsis_gcc.h
|    |    |          |-- core_armv8mbl.h
|    |    |          `-- core_armv8mml.h
|    |    `-- Device
|    |         |-- Include
|    |         |    |-- CMSDK_IoT
|    |         |    |    |-- CMSDK_IoT.h
|    |         |    |    `-- system_CMSDK_IoT.h
|    |         |    `-- CMSDK_driver.h
|    |         `-- Source
|    |              |-- CMSDK_IoT
|    |              |    |-- partition_CMSDK_ARMv8MBL.h
|    |              |    |-- startup_CMSDK_IoT.s
|    |              |    `-- system_CMSDK_IoT.c
|    |              `-- CMSDK_driver.c
|    `-- v2m_mps3
|         |-- AAIC_I2C_MPS3.c
|         |-- AAIC_I2C_MPS3.h
|         |-- AAIC_I2S_MPS3.c
|         |-- AAIC_I2S_MPS3.h
|         |-- ETH_MPS3.c
|         |-- ETH_MPS3.h
|         |-- Font_16x24_h.h
|         |-- Font_6x8_h.h
|         |-- GLCD_MPS3.c
|         |-- GLCD_MPS3.h
|         |-- SMM_MPS3.h
|         |-- TSC_I2C_MPS3.c
|         |-- TSC_I2C_MPS3.h
|         |-- fpga.c
|         |-- led.c
|         `-- led.h
```

12 Modifying and building AN545

12.1 Partial reconfiguration

AN545 for MPS3 makes use of Xilinx's partial reconfiguration, (PR) flow. With partial reconfiguration, specific design blocks can be allocated to a PR partition. These partitions can then be compiled to independent bitstreams. The PR bitstreams can be loaded to the FPGA to change the functionality of the FPGA within the PR design block.

In this flow, the mps3_fpga_user subsystem is designed as a PR partition and the contents of that partition can be modified by the user. The remaining functionality, (SSE-200 subsystem), is delivered as a pre-compiled encrypted bitstream and cannot be modified.

A Xilinx DCP (Design Checkpoint) file is provided to allow the users to compile their modified versions of the mps3_fpga_user subsystem. This is a preplaced design file containing all placement and routing for the enclosing top-level functionality which wraps around the mps3_fpga_user subsystem.

Note : For further understanding of partial reconfiguration using the Xilinx PR flow, the user is directed to the Xilinx Vivado Design Suite User Guide 909 – Partial Reconfiguration.

Note : With reference to the Xilinx Partial Reconfiguration terminology; "static image" aligns with the top level encrypted bitstream, and Reconfigurable Module, (RM), aligns with PR partition.

12.2 Pre-requisites

To build the AN545 FPGA the user must have a licensed copy of Xilinx Vivado HLx Edition. Version 2019.1 has been used for this application note . The license must also support partial reconfiguration.

The Vivado executable must be included in the user's path.

12.3 Flow overview

The files provided to the user consists of:

- Top level static DCP

- Encrypted bitstream containing the top level and SSE-200 subsystem, (545_t_X.bit).

- Source files to build mps3_fpga_user

In overview the flow consists of:

1. User synthesizes mps3_fpga_user into a DCP file.

2. The top level static DCP is combined with mps3_fpga_user DCP, and a stub DCP for the system core.

3. Place and route are then run. Note that since the top level is preplaced and routed, only the mps3_fpga_user partition is placed and routed.

4. PR bitfile produced for the mps3_fpga_user PR partition. Two files are produced for any PR partition; a clearing bitstream to clear the appropriate part of FPGA configuration memory, and the programming bitstream. These two bitstreams are named 545_uc_X.bit, (clearing), and 545_u_X.bit, (programming).

5. Top level static encrypted bitfile downloaded to MPS3 board

6. Two user PR partition bitfiles downloaded to MPS3 board

7. MPS3 board booted.

12.4 Flow detail

The user partition code is located in *<install_dir>/Luna/Logical/Resources/mps3_user_peripheral/AN545*. The top-level file, *mps3_fpga_user.v* is further located in the *user_wrapper* directory.

The steps to build a new version of AN545 are detailed below:

- Modify the code in the hierarchy under *mps3_fpga_user.v* to include your new code. Note that the ports of *mps3_fpga_user.v* itself must not be changed as these matches the provided top level DCP. It is strongly recommended that the user add their code within one of the existing hierarchical layers rather than directly into *mps3_fpga_user.v*

- Navigate to *<install_dir>/Luna/FPGA/AN545/smm_toplevel/xilinx/scripts*

- If different version numbers are required for the planned bitfiles, then edit *user_pr_synth.tcl* and set the variable *FPGA_BUILD* to the desired single digit number

  Note : The version number of the supplied files is 0. The default value of FPGA_BUILD set in the user scripts is 0. Therefore, in order to avoid any new bitfiles overwriting the pre-compiled files it is suggested that the value of FPGA_BUILD is modified.

- For a Linux system, execute $ *./user_pr_flow.scr*
  For a Windows system execute > *user_pr_flow.bat* from Vivado HLS Command Prompt.

- When the flow has completed it will produce two bitfiles, *545_u_X.bit*, and *545_uc_X.bit*. These will be written to the *<install_dir>Luna/FPGA/AN545/smm_toplevel/Xilinx/netlist_user* directory. *The* "X" will equate to the value of *FPGA_BUILD* written into *user_pr_impl.tcl.*

- Copy the new bitfiles *545_u_X.bit*, and *545_uc_X.bit.* to the directory *<MPS3_dir>MB/HBI0309C/AN545/* on the MPS3 board.

- Edit the configuration file *an545_v0.txt* in the same directory to use the new files
  ```
  F1FILE: 545_uc_6.bit    ;FPGA1 Filename - clear system PR - change this line
  F1MODE: FPGA            ;FPGA1 Programming Mode
  F2FILE: 545_u_6.bit     ;FPGA2 Filename - write system PR- change this line
  F2MODE: FPGA            ;FPGA2 Programming Mode
  ```

- Power on the MPS3 board. Check using either the debug UART or *log.txt* file that the new files were successfully programmed.

The MPS3 board is now programmed with the user code.

13 Using AN545 on the MPS3 board

13.1 Loading a prebuilt image onto the MPS3 board

To load the pre-built AN545 images, follow these steps:

- Power up the MPS3 board using the PBON push button and wait for the V2M_MPS3 drive to appear.

- Format the V2M_MPS3 drive and copy all the contents of *<install_dir>/Boardfiles* and paste them into the root directory of the attached V2M_MPS3 drive

- Note :You might want to manually modify and merge the contents for certain configuration files. Alternatively, you can restore the existing configuration files from the /Boardfiles directory. The affected configuration files are:

- <install_dir>/Boardfiles/config.txt

- <install_dir>/Boardfiles/MB/HBI0309C/board.txt

- <install_dir>/Boardfiles/MB/HBI0309C/AN545/images.txt

- Eject the V2M_MPS3 volume from your computer to unmount the drive

- Power cycle the MPS3 board using the PBRST push button and then launch BIOS update and FPGA configuration by pressing PBON push button. The LEDs will flash rapidly to indicate that a new BIOS is being downloaded (this only occurs the first time when the BIOS is updated) and that the prebuilt image is being downloaded onto the board. If you have configured the images.txt file, so that the MCC loads the selftest program, then the color LCD touch screen shows Arm MPS3 splash screen. If you have configured the UARTMODE to its default value of "0" in the config.txt file, then the debug UART0 terminal will simultaneously show the self-test menu for Application Note AN545

- If the MPS3 board does not boot correctly, then refer to the log.txt in the root directory of the MPS3 board which provides a log file of the files loaded at bootup.

13.2 UART Serial ports

Please refer to the *Arm® MPS3 FPGA Prototyping Board Getting Started Guide* accompanying the MPS3 board and *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual* for more information.

13.3 MPS3 USB Serial port drivers for Windows

See the following information on installing drivers to support the USB serial port on MPS3:

https://community.arm.com/dev-platforms/w/docs/381/accessing-mps3-serial-ports-in-windows

14 Software

In the Keil uVision, under Projects>Manage>Pack Installer you can find the "ARM::V2M-
MPS3_SSE_200_BSP" pack which contains software components like peripheral drivers and example
software for the target platform.
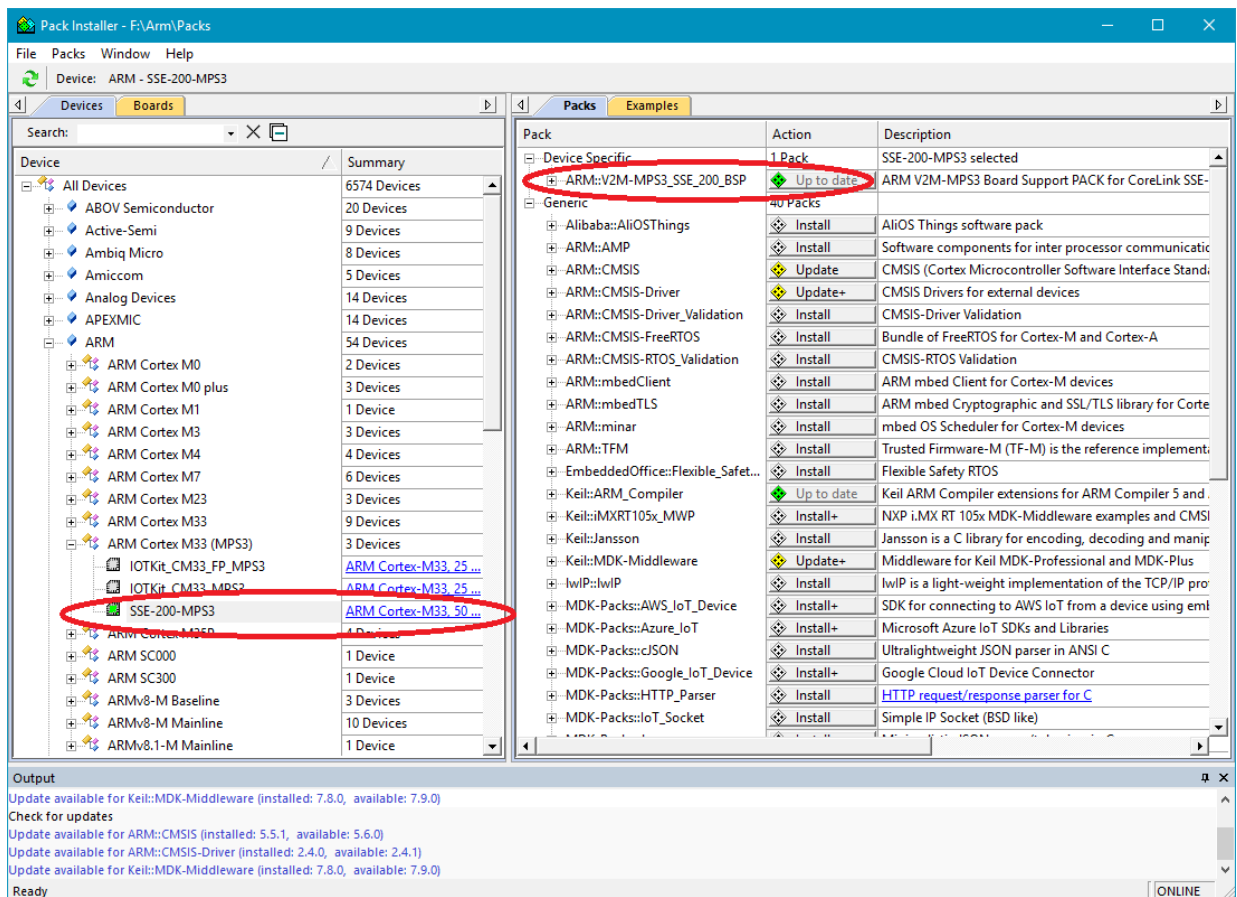
This pack can be also download form Keil website: http://www.keil.com/dd2/pack/

14.1 Rebuilding Software

Requirements

- The software directory from the download

- Keil uVision 5.27 or later

The following instructions apply to all software packages provided:

- Navigate to *<install_dir>/Software/YYY/Build_keil/*

- Load *YYY.uvprojx* (where YYY will be selftest or mem_test dependent on which project is chosen) in Keil uVision

- Once loaded, the project can be rebuilt by selecting either:

  o Project - > Build Target

  o Project - > Rebuild all target files

- The output can then be found in *<install_dir>/Software/selftest/Build_keil/an545_XX.axf* (where XX will be st or dm depending on which project is being built)

14.2 Loading software to the MPS3 board

Requirements

- MPS3 board powered and USB cable connected

- MPS3 USB mass storage open in a file explorer

The following instructions apply to all versions of software:

- Copying the software *<install_dir>/Software/selftest/Build_keil/an545_XX.axf* to the board *<MPS3_dir>/Software* folder

- Navigate to *<MPS3_dir>MB/HBI0309C/AN545* and open the *images.txt* file in a text editor

- Uncomment the test you wish to run and make sure the others are commented out e.g.

  ```
  IMAGE0FILE: \SOFTWARE\an545_st.axf ; - selftest uSD
  ;IMAGE0FILE: \SOFTWARE\an545_mt.axf ; - mem_test uSD
  ```

  (Selftest test is uncommented and therefore selected and mem test is commented out)

The MPS3 can now be booted up as per the instructions in the *Arm® MPS3 FPGA Prototyping Board Getting Started Guide* accompanying the MPS3 board.
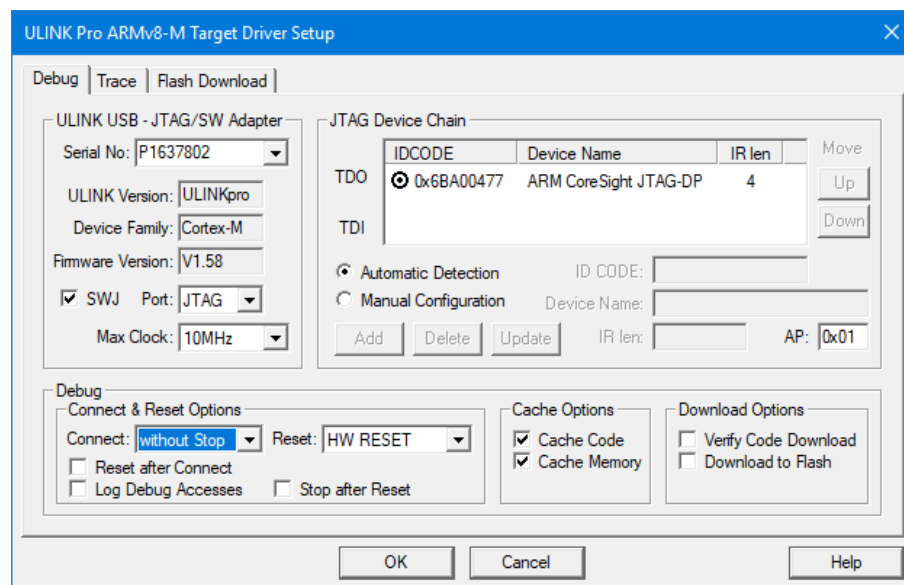
15 Debug

Debug has been tested using Keil uVision 5.27. To support warm reset over debug tool using ULINK Pro ARMv8-M Debugger or CMSIS-DAP ARMv8-M Debugger.

Apply the following debug settings:

Reset: HW RESET

Connect: without Stop

15.1 Trace support for Keil MDK

It is planned to include trace support for debug configuration of Keil tool in the future versions. Please follow the announcements of tool and pack updates related to the platform.

15.2 Debug and Trace support for Arm Development Studio

Trace support on armDS available on versions starting from 2020.

Please choose "Cortex-M Prototyping System (MPS3) Cortex-M33 (SSE-200 Subsystem)" as a debug target.