Example SSE-200 Subsystem for MPS3

Application Note AN524

Non-Confidential

Issue F

DAI 0524

Example SSE-200 Subsystem for MPS3

Application Note AN524

Release information

Document history

Issue	Date	Confidentiality	Change
A	15 January 2018	Non Confidential	First release.
B	19 February 2018	Non Confidential	Correct details of PR bitfile generation.
C	11 September 2018	Non Confidential	Update to memory map. Added Subsystem Configuration Section. Update to CLCD and SCC register descriptions.
D	23 May 2019	Non Confidential	Removed the word CoreLink from the title.
E	10 December 2019	Non Confidential	Versions of SSE-200 and SIE-200 updated to REL. Conversion of BRAM into synchronous block.
F	30 June 2021	Non Confidential	Added MCC memory map overview. Updated the section on Debug.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

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

Product Status

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

Web Address

http://www.arm.com

Progressive terminology commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used terms that can be offensive. Arm strives to lead the industry and create change.
This document includes terms that can be offensive. We will replace these terms in a future issue of this document. If you find offensive terms in this document, please contact 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, DEFINES THE LICENCE GRANTS.

DELIVERABLES

Part A

Hardware Binaries:
Encrypted FPGA bitstream file containing the SSE-200 Subsystem and other Arm technology.

Hardware Source Code:

Hardware netlists of Arm® CoreLink™ NIC-400, PrimeCell Infrastructure AMBA™ 2 AHB™ to AMBA 3 AXI™ Bridges (BP136).
RTL of Arm® PrimeCell Synchronous Serial Port (PL022) and Real Time Clock (PL031)
apb_i2s_top, CharLCDI, SBCon.
RTL of components in the Arm® 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 Arm®Keil® USB and SD card drivers, and Analog Devices FMC EEPROM reader.
selftest binary.

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 selftest

Part D

None

Contents

1 Introduction

1.1 Intended audience

This application note document is written for experienced hardware , System-on-Chip (SoC) and software engineers who might or might not have experience with Arm products. Such engineers typically have experience in writing Verilog and of performing synthesis, but might have limited experience of integrating and implementing Arm products.

1.2 Conventions

The following subsections describe conventions used in Arm documents.

1.2.1 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.

1.2.2 Typographical conventions

Convention	Use
italic	Introduces special terminology, denotes cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`Monospace` **`bold`**	Denotes language keywords when used outside example code.
`monospace italic`	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`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: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
	Caution
	Warning
	Note

1.3 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 Y/N
Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual	101104	No
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	No
Arm® MPS3 FPGA Prototyping Board Technical Reference Manual	100765	No
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	No
Arm® MPS3 FPGA Prototyping Board Getting Started Guide	-	No
MCBQVGA-TS-Display-v12 – Keil MCBSTM32F200 display board schematic	-	No

Table 1-2 Other publications

Document name	Document ID
Xilinx Vivado Design Suite User Guide	UG909

1.4 Feedback

Arm welcomes feedback on this product and its documentation.

1.4.1 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.

1.4.2 Feedback on content

If you have comments on content, send an e-mail to errata@arm.com and give:

- The title *Example SSE-200 Subsystem for MPS3- Application Note AN524*.
- The number DAI 0524, Issue F.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

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.

1.4.3 Other information

- 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

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.

2 Preface

2.1 Purpose of this application note

This document describes the features and functionality of application note AN524. AN524 is an FPGA implementation of the SSE-200 Subsystem that uses SIE-200 components together with CMSDK peripherals to provide an example design.

2.2 Terms and Abbreviations

AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
BRAM	Block Random Access Memory
CMSDK	Cortex® -M System Design Kit
DCP	Design Checkpoint
DMA	Direct Memory Access
DS	Arm® Development Studio
EAM	Exclusive Access Controller
FPGA	Field Programmable Gate Array
I2S	Inter-IC Sound
IDAU	Implementation Defined Attribution Unit
KB	Kilo Byte
MB	Mega Byte
MCC	Motherboard Configuration Controller
MPC	Memory Protection Controller
MSC	Manager Security Controller
PPC	Peripheral Protection Controller
PR	Partial Reconfiguration
QSPI	Quad Serial Peripheral Interface
RAM	Random Access Memory
RTC	Real Time Clock
RTL	Register Transfer Level
SSE	Subsystem for Embedded
SCC	Serial Configuration Controller
SMB	Static Memory Bus
SMM	Soft Macrocell Model
SPI	Serial Peripheral Interface
TRM	Technical Reference Manual
UART	Universal Asynchronous Receiver/Transmitter
XIP	Execute in place

2.3 Subsystem version details

Version	Descriptions
BP210	Cortex-M System Design Kit Full version of the design kit supporting Cortex-M0 processor, Cortex-M0 DesignStart™, Cortex-M0+, Cortex-M3 and Cortex-M4 processors. 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 Arm® TrustZone® for v8-M ecosystem. All SIE-200 components have AHB5 interfaces to support Armv8-M processors.
r2p0	SSE-200 The SSE-200 is a collection of a pre-assembled elements to use as the basis of an IoT SoC.
r1p3-00rel1	PL022 Arm® PrimeCell Synchronous Serial Port

Table 2-1 : Module versions

2.4 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.

3 Overview

This SMM is based on the SSE-200 Subsystem which contains two Cortex-M33 cores, interconnect, peripherals and other systems.

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

3.1 System block diagram

The following shows a high-level view of the full MPS3 SSE-200 FPGA System with the default user partition:

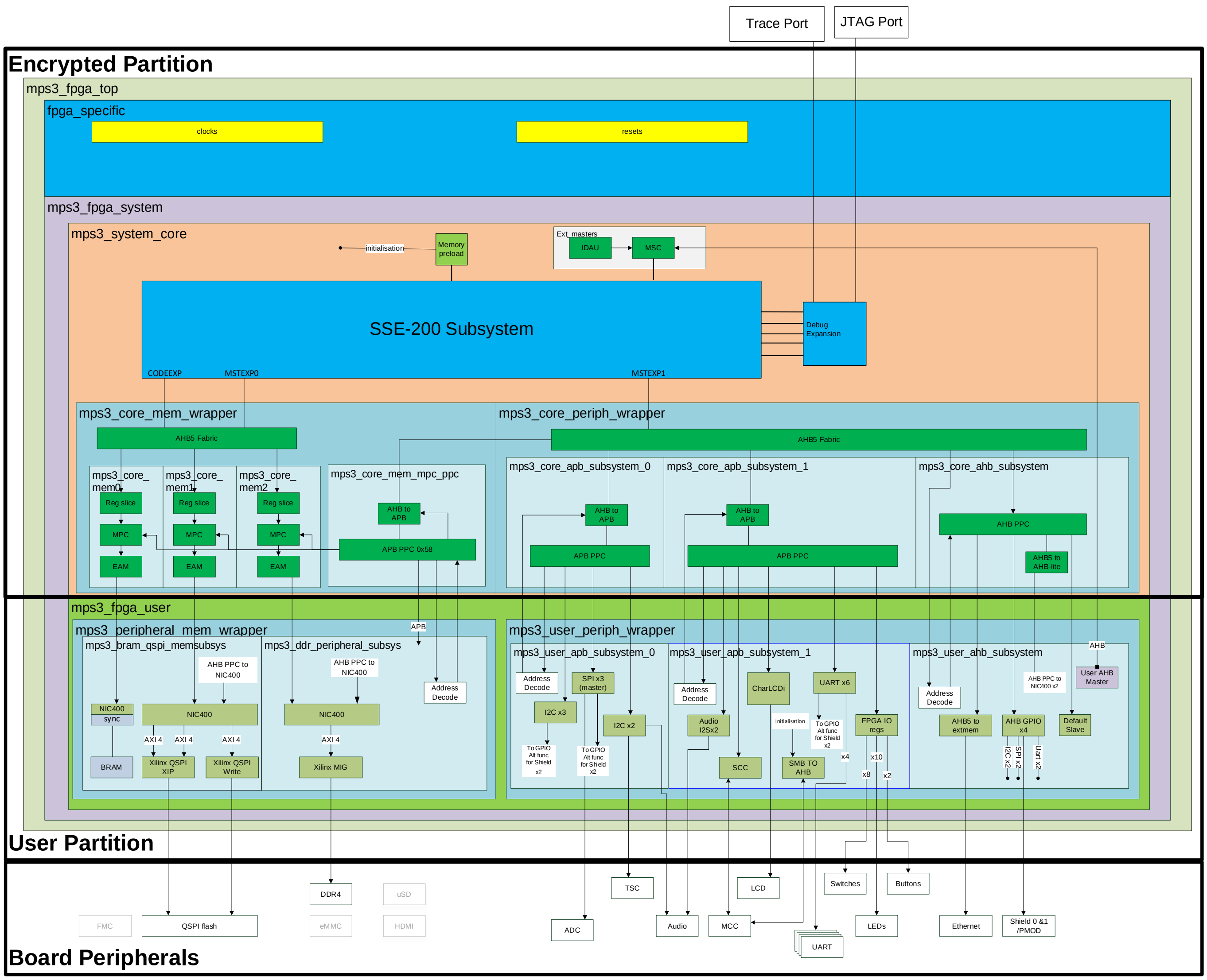


Figure 3-1 : System overview

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

3.2 SIE-200 components

The system uses the following SIE-200 components:

- TrustZone AHB5 peripheral protection controller
- TrustZone AHB5 Manager 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 subordinate

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. See the *Arm® SIE-200 System IP Technical Reference Manual* (Arm DDI0571) for more information.

3.4 Memory map overview

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

See the SIE-200 components documentation for more information.

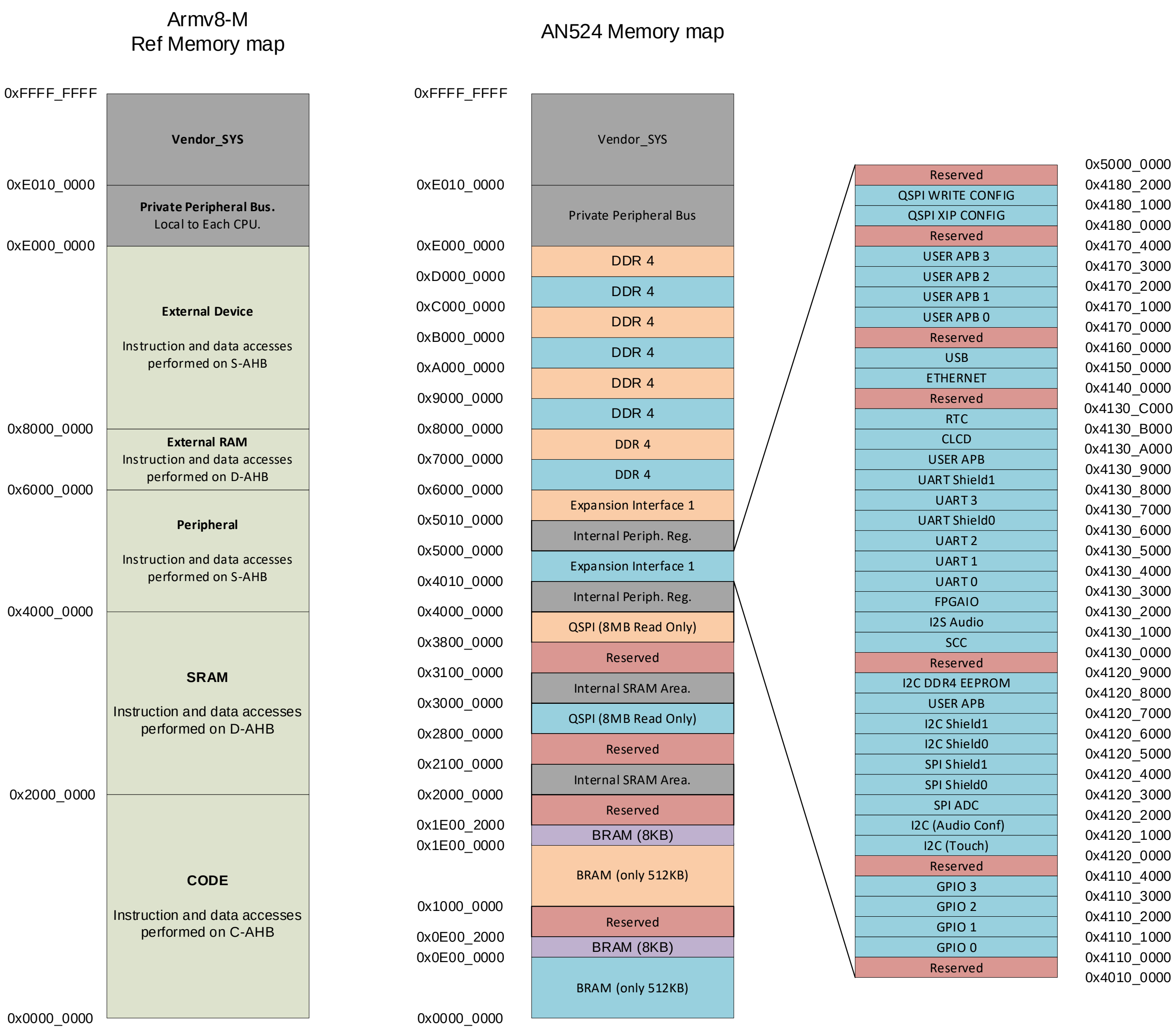


Figure 3-1 : Memory map

The following table shows the memory map overview:

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	NS	0	0
2	0x0008_0000	0x0DFF_FFFF	223MB	Reserved	Reserved				
3	0x0E00_0000	0x0E00_1FFF	8KB	NVM Code	Arm® 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	S	1	CODE NSC2
6	0x1008_0000	0x1DFF_FFFF	223MB	Reserved	Reserved				
7	0x1E00_0000	0x1E00_1FFF	8KB	NVM Code	Arm® 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	NS	2	0
10	0x2000_8000	0x2000_FFFF	32KB	Internal SRAM	Internal SRAM 1 Area				
11	0x2001_0000	0x2001_7FFF	32KB	Internal SRAM	Internal SRAM 2 Area				
12	0x2001_8000	0x2001_FFFF	32KB	Internal SRAM	Internal SRAM 3 Area				
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	S	3	RAMN SC
17	0x3000_8000	0x3000_FFFF	32KB	Internal SRAM	Internal SRAM 1 Area				
18	0x3001_0000	0x3001_7FFF	32KB	Internal SRAM	Internal SRAM 2 Area				
19	0x3001_8000	0x3001_FFFF	32KB	Internal SRAM	Internal SRAM 3 Area				
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	NS	4	0
24	0x4001_0000	0x4001_FFFF	64KB	Private CPU	CPU Element Peripheral Region.	31			
25	0x4002_0000	0x4002_FFFF	64KB	System Control	System Control Element Peripheral region.	32			
26	0x4003_0000	0x4003_FFFF		Reserved	Reserved				
27	0x4004_0000	0x4007_FFFF		Reserved	Reserved				

ROW ID	Address		Size	Region Name	Description	Alias With Row ID	IDAU Region Values		
	From	To					Security	IDAUID	NSC
28	0x4008_0000	0x400F_FFFF	512KB	Base Peripheral	Base Element Peripheral Region.	35			
29	0x4010_0000	0x4FFF_FFFF	255MB	Expansion 1	Maps to AHB5 Manager 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 Manager 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 Manager 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 Manager Expansion 1 Interface	46	S	F	0

Table 3-1 : Memory map overview

3.5 Remap

The memory remap function is controlled via SCC CFGREG0[0] register. It can be setup in the FPGA_REMAP section of `an524_v3.txt` file.

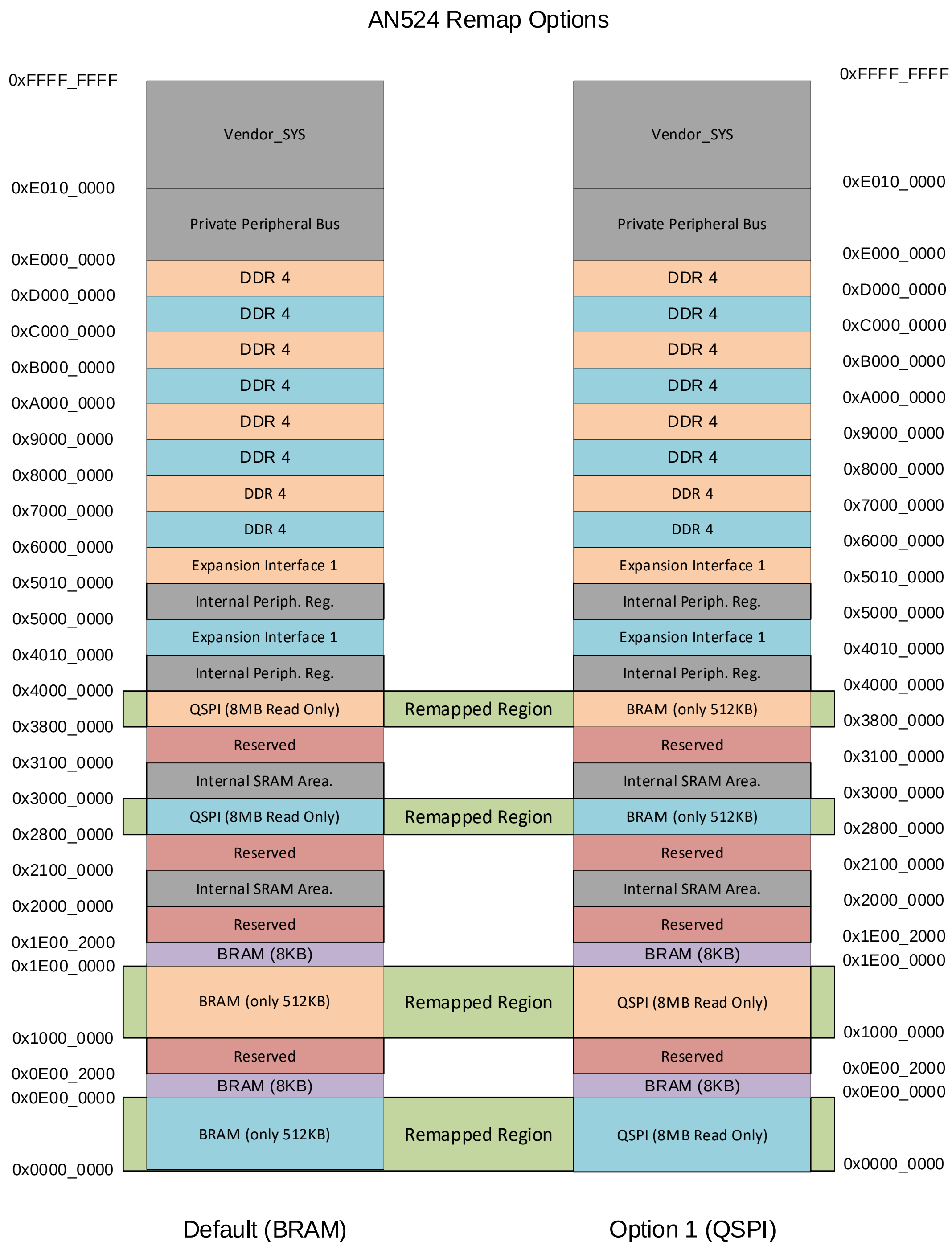


Figure 3-2 : Remap options

3.6 MCC memory map for AN524

MCC has some visibility into the memory for initiating boot memory areas and configuring peripherals if needed. MCC has limited access , 4x64MB, to the design memory map. So, it is unable to cover the whole map and only those regions which are necessary for the design functionality are visible.
The memory map as viewed from the MCC is below:

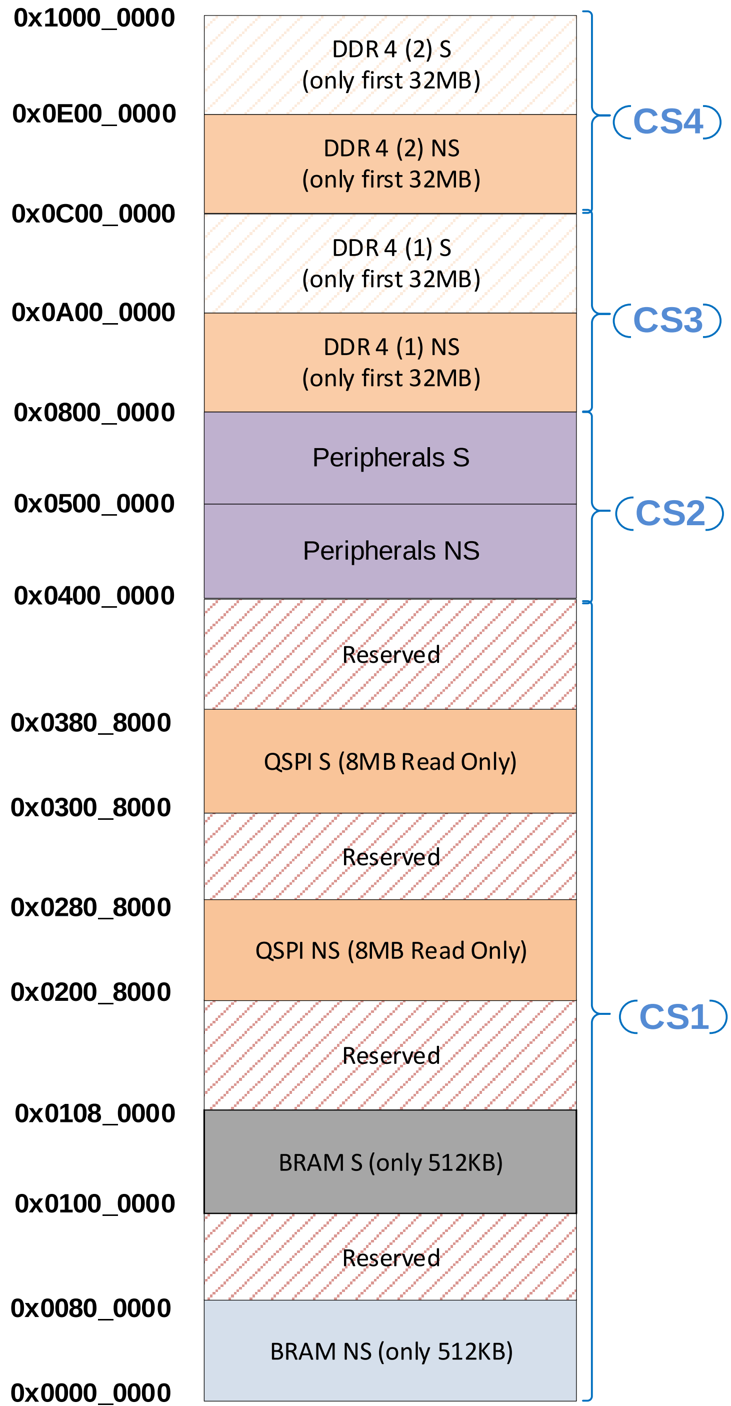


Figure 3-3 : MCC memory map for AN524

The following table shows the MCC Memory map:

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

Table 3-2 : MCC memory map table

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

To support TrustZone-Armv8M, several peripherals are mapped to both secure or non-secure address space.

APB PPC's and AHB PPC's gate access to the peripherals. FPGA Secure and Non-Secure Privilege Control blocks define the peripheral security access settings by controlling the PPCs. For expansion AHB Subordinates within the system, there is a Manager Security Controller (MSC) added to each subordinate with an associated IDAU. The user has access to one of these interfaces via AHB from the user peripheral area of the design.

The following tables show the FPGA expansion peripheral maps of Non-Secure and Secure regions:

ROW ID	Address From	Address To	Size	Description	Port
				Non-Secure Region	
1	0x4110_0000	0x4110_0FFF	4K	GPIO 0	AHB
2	0x4110_1000	0x4110_1FFF	4K	GPIO 1	
3	0x4110_2000	0x4110_2FFF	4K	GPIO 2	
4	0x4110_3000	0x4110_3FFF	4K	GPIO 3	
5	0x4110_4000	0x411F_FFFF		Reserved	
6	0x4120_0000	0x4120_0FFF	4K	FPGA - SBCon I2C (Touch)	APB0
7	0x4120_1000	0x4120_1FFF	4K	FPGA - SBCon I2C (Audio Conf)	
8	0x4120_2000	0x4120_2FFF	4K	FPGA - PL022 (SPI ADC)	
9	0x4120_3000	0x4120_3FFF	4K	FPGA - PL022 (SPI Shield0)	
10	0x4120_4000	0x4120_4FFF	4K	FPGA - PL022 (SPI Shield1)	
11	0x4120_5000	0x4120_5FFF	4K	SBCon (I2C - Shield0)	
12	0x4120_6000	0x4120_6FFF	4K	SBCon (I2C - Shield1)	
13	0x4120_7000	0x4120_7FFF	4K	USER APB	
14	0x4120_8000	0x4120_8FFF	4K	FPGA - SBCon I2C (DDR4 EEPROM)	
15	0x4120_9000	0x412F_FFFF		Reserved	
16	0x4130_0000	0x4130_0FFF	4K	FPGA - SCC registers	APB1
17	0x4130_1000	0x4130_1FFF	4K	FPGA - I2S (Audio)	
18	0x4130_2000	0x4130_2FFF	4K	FPGA - IO (System Ctrl + I/O)	
19	0x4130_3000	0x4130_3FFF	4K	UART0 - UART_F[0]	
20	0x4130_4000	0x4130_4FFF	4K	UART1 - UART_F[1]	
21	0x4130_5000	0x4130_5FFF	4K	UART2 - UART_F[2]	
22	0x4130_6000	0x4130_6FFF	4K	UART3 - UART Shield 0	
23	0x4130_7000	0x4130_7FFF	4K	UART4 - UART Shield 1	
24	0x4130_8000	0x4130_8FFF	4K	UART5 - UART_F[3]	
25	0x4130_9000	0x4130_9FFF	4K	USER APB	APB1
26	0x4130_A000	0x4130_AFFF	4K	CLCD Config Reg	
27	0x4130_B000	0x4130_BFFF	4K	RTC	
28	0x4130_C000	0x413F_FFFF		Reserved	
29	0x4140_0000	0x414F_FFFF	1M	Ethernet	EAM
30	0x4150_0000	0x415F_FFFF	1M	USB	
31	0x4160_2000	0x416F_FFFF		Reserved	
32	0x4170_0000	0x4170_0FFF	4K	User APB0	APB (Mem)
33	0x4170_1000	0x4170_1FFF	4K	User APB1	
34	0x4170_2000	0x4170_2FFF	4K	User APB2	
35	0x4170_3000	0x4170_3FFF	4K	User APB3	
36	0x4170_4000	0x4800_6FFF		Reserved	
37	0x4800_7000	0x4800_7FFF	4K	FPGA Non-Secure Privilege Control	
38	0x4800_8000	0x4FFF_FFFF		Reserved	

Table 3-3 : FPGA expansion peripheral map of Non-secure region

ROW ID	Address From	Address To	Size	Description	Port
				Secure Region	
1	0x5110_0000	0x5110_0FFF	4K	GPIO 0	AHB
2	0x5110_1000	0x5110_1FFF	4K	GPIO 1	
3	0x5110_2000	0x5110_2FFF	4K	GPIO 2	
4	0x5110_3000	0x5110_3FFF	4K	GPIO 3	
5	0x5110_4000	0x511F_FFFF		Reserved	
6	0x5120_0000	0x5120_0FFF	4K	FPGA - SBCon I2C (Touch)	APB0
7	0x5120_1000	0x5120_1FFF	4K	FPGA - SBCon I2C (Audio Conf)	
8	0x5120_2000	0x5120_2FFF	4K	FPGA - PL022 (SPI ADC)	
9	0x5120_3000	0x5120_3FFF	4K	FPGA - PL022 (SPI Shield0)	
10	0x5120_4000	0x5120_4FFF	4K	FPGA - PL022 (SPI Shield1)	
11	0x5120_5000	0x5120_5FFF	4K	SBCon (I2C - Shield0)	
12	0x5120_6000	0x5120_6FFF	4K	SBCon (I2C - Shield1)	
13	0x5120_7000	0x5120_7FFF	4K	USER APB	
14	0x5120_8000	0x5120_8FFF	4K	FPGA - SBCon I2C DDR4 EEPROM	
15	0x5120_9000	0x512F_FFFF		Reserved	
16	0x5130_0000	0x5130_0FFF	4K	FPGA - SCC registers	APB1
17	0x5130_1000	0x5130_1FFF	4K	FPGA - I2S (Audio)	
18	0x5130_2000	0x5130_2FFF	4K	FPGA - IO (System Ctrl + I/O)	
19	0x5130_3000	0x5130_3FFF	4K	UART0 - UART_F[0]	
20	0x5130_4000	0x5130_4FFF	4K	UART1 - UART_F[1]	
21	0x5130_5000	0x5130_5FFF	4K	UART2 - UART_F[2]	
22	0x5130_6000	0x5130_6FFF	4K	UART3 - UART Shield 0	
23	0x5130_7000	0x5130_7FFF	4K	UART4 - UART Shield 1	
24	0x5130_8000	0x5130_8FFF	4K	UART5 - UART_F[3]	
25	0x5130_9000	0x5130_9FFF	4K	USER APB	
26	0x5130_A000	0x5130_AFFF	4K	CLCD Config Reg	
27	0x5130_B000	0x5130_BFFF	4K	RTC	
28	0x5130_C000	0x513F_FFFF		Reserved	
29	0x5140_0000	0x514F_FFFF	1M	Ethernet	EAM
30	0x5150_0000	0x515F_FFFF	1M	USB	
31	0x5160_0000	0x516F_FFFF		Reserved	
32	0x5170_0000	0x5170_0FFF	4K	User APB0	APB (Mem)
33	0x5170_1000	0x5170_1FFF	4K	User APB1	
34	0x5170_2000	0x5170_2FFF	4K	User APB2	
35	0x5170_3000	0x5170_3FFF	4K	User APB3	
36	0x5170_4000	0x5800_8FFF		Reserved	
37	0x5800_7000	0x5800_7FFF	4K	BRAM Memory Protection Controller (MPC)	APB (Mem)
38	0x5800_8000	0x5800_8FFF	4K	QSPI Memory Protection Controller (MPC)	
39	0x5800_9000	0x5800_9FFF	4K	DDR4 Memory Protection Controller (MPC)	
40	0x5800_8000	0x5FFFF_FFFF		Reserved	

Table 3-4 : FPGA expansion peripheral map of Secure region

Note: Reserved regions should not be accessed.

3.8 FPGA Utilization

This application note is designed for MPS3 board. The board will use a a Xilinx Kintex Ultrascale XCKU115 FPGA.
The FPGA features up to 8MB BRAM (2160 BlocRAM tiles) and up to 663360 LUTs.
Full part number: XCKU115-FLVB1760-1-C.

3.8.1 Total design utilization

The following table shows the total number of LUTs and BRAMs currently used in the provided image.

Site Type	Used	Util%
LUTs	177383	26
BlockRAM Tile	193	8

Table 3-5 AN524 utilization summary

Note

These numbers relate to the complete image, not individual IP blocks. The numbers must not be used to infer IP size, or the relative sizes of different IP blocks, because the implementation and system design can significantly differ.

3.8.2 User partition

User-reserved area is about 26% of total FPGA. The provided example design is using 31% of User-reserved area.

Site Type	Total Available	Used	Util%
LUTs	174240	53497	31
BlockRAM Tile	576	159	28

Table 3-6 Size and utilization of the User Partition

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

Internal FPGA SRAM is the primary, default boot memory of Size 512KB

Size:	512KB FPGA BRAM
Address Range:	0x0000_0000 - 0x0007_FFFF
Alias Range:	0x1000_0000 - 0x1007_FFFF

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:	0x2800_0000 - 0x287F_FFFF
Alias Range:	0x3800_0000 - 0x387F_FFFF

4.4 DDR4

The SMM also includes 2GB of External DDR4 memory

Size:	2GB DDR4
Address Range:	0x6000_0000 - 0xDFFF_FFFF

4.5 AHB GPIO

The SMM uses four CMSDK AHB GPIO blocks, each providing 16 bits of IO. These IO are connected to the two Arduino compatible shield headers 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 peripherals or GPIOs are available on each pin. See section 8 - Shield Support for mappings.

4.6 SPI

The SMM implements three PL022 PrimeCell Synchronous Serial Port 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.
- 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 (I^2C)

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 s - Shield Support for mappings.
- One SBCon module reads EEPROM from DDR4 SODIMM.

The selftest program that is provided with the MPS3 includes example code for the color LCD module control and Audio interfaces.
The following table shows the register map for the two-wire SBCon:

Address	Name	Access	Description
0x000	SB_CONTROL	Read	Read serial control bits: Bit [0] is SCL Bit [1] is SDA
0x000	SB_CONTROLS	Write	Set serial control bits: Bit [0] is SCL Bit [1] is SDA
0x004	SB_CONTROLC	Write	Clear serial control bits: Bit [0] is SCL Bit [1] is SDA

Table 4-2 : SBCon register map

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^2C to transfer data input from the touch screen

The color LCD Module is a custom peripheral that provides an interface to an 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.

selftest that is provided with the MPS3 includes drivers and example code for both these interfaces.

The following table shows the control and data registers for the CLCD interface :

Address	Name	Type	Information
0x4130_A000	CHAR_COM	Write command, read busy status	A write to this address causes a write to the LCD command register. A read from this address causes a read from the LCD busy register.
0x4130_A004	CHAR_DAT	Write data RAM, Read data RAM	A write to this address causes a write to the LCD data register. A read from this address causes a read from the LCD data register.
0x4130_A008	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.
0x4130_A00C	CHAR_RAW	Write to reset access complete flag, Read to determine if data in CHAR_RD is valid	Bit [0] indicates Access Complete (write 0b0 to clear). The bit is set if read data is valid. Bits [31:1] are reserved.
0x4130_A010	CHAR_MASK	Write interrupt mask	Set Bit [0] to 0b1 to enable Access Complete to generate an interrupt.
0x4130_A014	CHAR_STAT	Read status	Bit[0] is the state of Access Complete ANDed with the CHAR_MASK.
0x4130_A04C	CHAR_MISC	Miscellaneous Control	Bit field description : Bits [31:7] : Reserved. Bit[6]: CLCD_BL. Bit[5]: CLCD_RD. Bit[4]: CLCD_RS. Bit[3]: CLCD_RESET. Bit[2]: Reserved. Bit[1]: CLCD_WR. Bit[0]: CLCD_CS.

Table 4-3 : CLCD interface register map

4.10 Ethernet

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

The selftest program includes example code for a simple loopback operation.

4.11 USB

The SMM design connects to a Hi-Speed USB 2.0 OTG controller (ISP1763) device through a static memory interface.

The selftest program includes example code for a simple loopback operation.

4.12 RTC

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

4.13 Audio I^{2}S

The I^{2}S interface supports transfer of digital audio to and from the audio CODEC.

The following table shows the register memory map for I^{2}S audio registers:

Address	Name	Description
0x000	CONTROL	Control Register Bits[31:18] : Reserved. Bit[17]: Audio codec reset control (output pin). Bit[16] : FIFO reset. Bit[15] : Reserved. Bits [14:12] : Rx Buffer IRQ Water Level - Default 2. (IRQ triggers when less than two-word space is available). Bit [11] : Reserved. Bits [10:8] : TX Buffer IRQ Water Level - Default 2. (IRQ triggers when more than two-word space is available). Bits [7:4] : Reserved. Bit [3] : Rx Interrupt Enable. Bit [2] : Rx Enable. Bit [1] : Tx Interrupt Enable. Bit [0] : Tx Enable.

Address	Name	Description
0x004	STATUS	Status Register Bits[31:6] : Reserved. Bit[5]: Rx Buffer Full. Bit[4]: Rx Buffer Empty. Bit[3]: Tx Buffer Full. Bit[2]: Tx Buffer Empty. Bit[1]: Rx Buffer Alert (Depends on Water level). Bit[0]: Tx Buffer Alert (Depends on Water level).
0x008	ERROR	Error Status Register Bits[31:2] : Reserved. Bit [1] : Rx overrun. Set this bit to clear. Bit [0] : Tx overrun or underrun. Set this bit to clear.
0x00C	DIVIDE	Clock Divide Ratio Register (for left or right clock) Bits[31:10] : Reserved. Bits[9:0] : LRDIV (Left/Right). The default value is 0x80. 12.288MHz / 48kHz / 2*(L+R) = 128.
0x010	TXBUF	Transmit Buffer FIFO Data Register. This is a write-only register. Bits[31:16] : Left channel. Bits[15:0] : Right channel.
0x014	RXBUF	Receive Buffer FIFO Data Register. This is a read-only register. Bits[31:16] : Left channel. Bits[15:0] : Right channel.
0x015- 0x2FC	RESERVED	-
0x300	ITCR	Integration Test Control Register. Bits[31:1] : Reserved. Bit[0] : ITCR.
0x304	ITIP1	Integration Test Input Register 1. Bits[31:1] : Reserved Bit[0] : SDIN
0x308	ITOP1	Integration Test Output Register 1. Bits[31:4] : Reserved Bit[3] : IRQOUT Bit[2] : LRCK Bit[1] : SCLK Bit[0] : SDOUT

Table 4-4 : I²S register memory map

4.14 Audio configuration

The SMM implements a simple SBCon interface based on I²C. It configures 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 following table shows the FPGA system control block implemented by the SMM:

Address	Security	Name	Information
0x4130_2000	Non - Secure	FPGAIO->LED0	LED connections Bits [31:10] : Reserved. Bits[9:0] : LED.
0x5130_2000	Secure		
0x4130_2004	Non - Secure	RESERVED	-
0x5130_2004	Secure		
0x4130_2008	Non - Secure	FPGAIO->BUTTON	Buttons Bits[31:2] : Reserved. Bits[1:0] : Buttons (Read – Only).
0x5130_2008	Secure		
0x4130_200C	Non - Secure	RESERVED	-
0x5130_200C	Secure		
0x4130_2010	Non - Secure	FPGAIO->CLK1HZ	32-bit 1Hz up counter.
0x5130_2010	Secure		
0x4130_2014	Non - Secure	FPGAIO->CLK100HZ	32-bit 100Hz up counter.
0x5130_2014	Secure		
0x4130_2018	Non - Secure	FPGAIO->COUNTER	32-bit Cycle Up Counter. Increments when 32-bit prescale counter reaches zero and automatically reloads.
0x5130_2018	Secure		
0x4130_201C	Non - Secure	FPGAIO->PRESCALE	Bits[31:0] – reload value *for* prescale counter.
0x5130_201C	Secure		
0x4130_2020	Non - Secure	FPGAIO->PSCNTR	32-bit Prescale counter – current value of the pre-scaler counter. The Cycle Up Counter increments when the prescale down counter reaches 0. The pre-scaler counter is reloaded with PRESCALE after reaching 0b0.
0x5130_2020	Secure		
0x4130_2024	Non - Secure	RESERVED	-
0x5130_2024	Secure		
0x4130_2028	Non - Secure	FPGAIO->SWITCH	Switches Bits[31:8] : Reserved. Bits[7:0] : Switches (0 – Off, 1 – On) .
0x5130_2028	Secure		
0x4130_204C	Non - Secure	FPGAIO->MISC	Misc control

Address	Security	Name	Information
0x5130_204C	Secure		Bits[31:3] : Reserved. Bits[2] : SHIELD1_SPI_nCS (R/W). Bits[1] : SHIELD0_SPI_nCS (R/W). Bits[0] : ADC_SPI_nCS (R/W).

Table 4-5 : System control and I/O memory map

4.16 SCC

The SMM implements communication between the MCC and the FPGA system through an SCC interface.

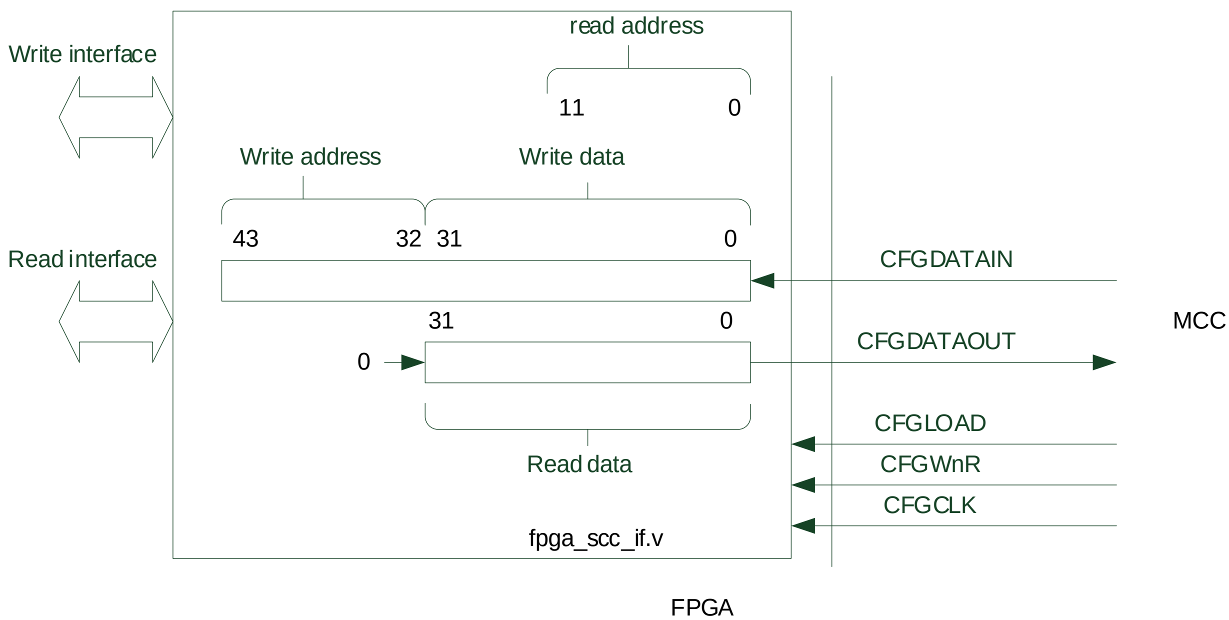


Figure 4-1 : SCC interface

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

The following table shows the SCC Register memory map in address offset order from the base memory address.:

Offset 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, (0 - XIP read only, 1 - Write enable)
0x00C	CFG_REG3	Bits [31:0] : Reserved
0x010	CFG_REG4	Bits [31:4] : Reserved Bits [3:0] : Board Revision [r]

Offset Address	Name	Information
0x014	CFG_REG5	Bits [31:0] : ACLK Frequency in Hz,(read only)
0x018	CFG_REG6	Bits [3:0] : Clock divider for BRAM (1, 2, 4, 8). Value of the divider can be 2, 4 or 8. Any other value in this register will result in 1:1 ratio.
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)
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. This Release has a value 3. User can rebuild the user space and replace it to differ from the released build. Bits[23:20] : V2M-MPS3 target board revision (A = 0, B = 1, C = 2) Bits[19:0] : Reserved
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: 524 = AN524 Bits[3:0] : Reserved

Table 4-6 : 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 FPGA:

Input Pin	Board File Name	Frequency	Note
OSCCLK[0]	-	24MHz	Constant 24MHz reference, used for RTC and timers.
OSCCLK[1]	OSC1	32MHz	ACLK, main clock used to clock SSE-200 subsystem. Frequency can be changed in the board file an524_v3.txt
OSCCLK[2]	OSC2	50MHz	MCLK, Reserved
OSCCLK[3]	OSC3	50MHz	GPUCLK, aux clock used to generate PERIPH_CLK for user space. Frequency can be changed in the board file an524_v3.txt
OSCCLK[4]	OSC4	24.576MHz	AUDCLK, clock used to clock I2S audio module. Frequency can be changed in the board file an524_v3.txt
OSCCLK[5]	OSC5	23.75MHz	HDLCDCLK, clock can be used to clock video module. MCC overrides this value. Frequency can be changed in the board file an524_v3.txt
c0_sys_clk_p/n	OSC6 (GTX Clock)	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.It can be selected by writing into SCC register address 0x18 (See Table 4-6 : SCC Register memory map) .
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 provides expansion security control signals to control the various security gating units within the subsystem. The following table lists the connectivity of system security extension signals.

Component name	Components signal	Security expansion signal
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.

Each APB <n> interface is controlled by APB_NS_PPCEXP0[n] and APB_P_PPCEXP0[n].

The following table lists the peripherals that are controlled by APB PPC EXP 0 :

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 : Peripheral mapping of APB PPC EXP 0

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

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

APB PPC EXP 2 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 : Peripheral mapping of APB PPC EXP 2

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

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 : Peripheral mapping of AHB PPC EXP 0

7 Interrupt Map

The following table shows how Interrupts in the FPGA subsystem extend the SSE-200 Interrupt map by adding to the expansion area:

7.1 FPGA interrupt map

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). Overflow interrupts are ORed together.
IRQ[48]	Ethernet. Interrupt from LAN Chip is inverted and synced to HCLK
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]	Reserved

Table 7-1 : FPGA expansion interrupt map.

7.2 UART interrupts

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

- TXINT
- RXINT
- TXOVRINT
- EXOVRINT
- UARTINT

The TXINT, RXINT and UARTINT interrupt signal of each UART drive a separate interrupt inputs of the Cortex-M33 CPU.

In addition, the TXOVRINT and RXOVRINT interrupt signals of all six UARTs, twelve signals in total, 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 Shield headers.

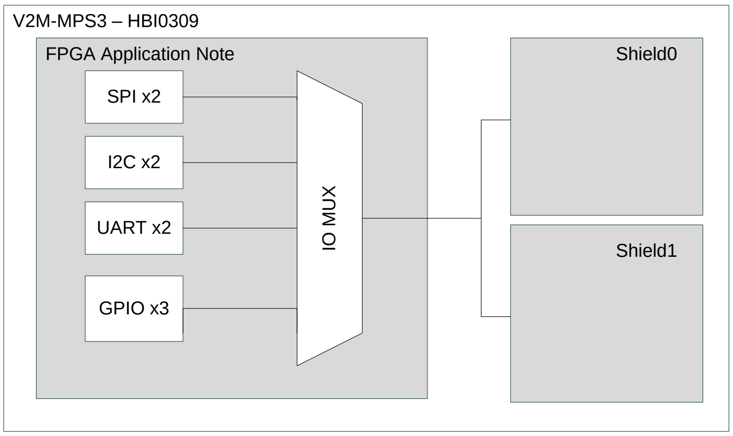


Figure 8-1 : Shield device expansion

Multiplexing is controlled by the alternative function output from the associated GPIO Register. The following table shows the Shield alternative function pinout:

MPS3	AN524	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	AN524	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 Configuration options

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. The following table shows some of the most important configuration settings and also where this application note uses non-default values:

Parameter	Implemented Values	Default Values	Description
HAS_CRYPTO	No {0}	Yes {1}	Include CryptoCell 312
CPU0_FPU	No {0}	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 97+32 = 129 interrupts.
CPU1_EXP_NUMIRQ	97	64	Specifies the number of expansion interrupt. This means that the M33 NVIC has 97+32 = 129 interrupts.
CPU0_EXP_IRQDIS	0xAAAA	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	0x0	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.
INITSVTOR0_RST	0x100000		Sets bits [31:7] of the reset value of the Secure vector table offset address in the Cortex-M33 processor, in CPU element 0.
INITSVTOR0_RST	0x100000		Sets bits [31:7] of the reset value of the Secure vector table offset address in the Cortex-M33 processor, in CPU element 1.

Table 9-1 : SSE-200 configuration options

9.2 Cortex-M33

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

10 ZIP Bundle Description

10.1 Overall Structure

The accompanying `.zip` bundle contains:

- This Application Note Document.
- An example Keil® MDK Version 5.27 software project, that can be run on the MPS3 board peripherals and interfaces.
- `Boardfiles/` directory containing 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.

10.2 Bundle Directory Tree/Structure

The directory tree structure of the bundle is shown below.

```
. .
|-- Boardfiles
|    |-- MB
|    |    |-- BRD_LOG.TXT
|    |    |-- HBI0309B
|    |    `-- HBI0309C
|    |-- SOFTWARE
|    |    |-- an524_dm.axf
|    |    `-- an524_st.axf
|    `-- config.txt
|-- Docs
|    `-- DAI0524F_example_sse200_subsystem_for_mps3.pdf
|-- Licence.pdf
|-- Luna
|    |-- FPGA
|    |    `-- AN524
|    `-- Logical
|         |-- AN524_SMM_SSE200
|         `-- Resources
|-- Software
|    |-- demo
|    |    |-- Build_Keil
|    |    |-- apmain
|    |    |-- aptsc
|    |    |-- cmsis
|    |    |-- demo
|    |    |-- icons
|    |    `-- v2m_mps3
|    `-- selftest
|         |-- Build_Keil
|         |-- apaaci
|         |-- apclcd
|         |-- apgpio
|         |-- aplan
|         |-- apleds
|         |-- apmain
|         |-- apmem
|         |-- apqspi
|         |-- aprtc
|         |-- apssp
|         |-- aptimer
|         |-- aptsc
|         |-- apuart
|         |-- apusb
|         |-- cmsis
|         `-- v2m_mps3
|-- readme.txt
`-- revision_history.txt
```

10.3 Documentation

This Application Note Document, AN524, is in the `Docs/` folder of the bundle.

11 Board Revision And Support

11.1 Identifying the MPS3 board revision

The bundle supports MPS3 board revisions A, 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 :

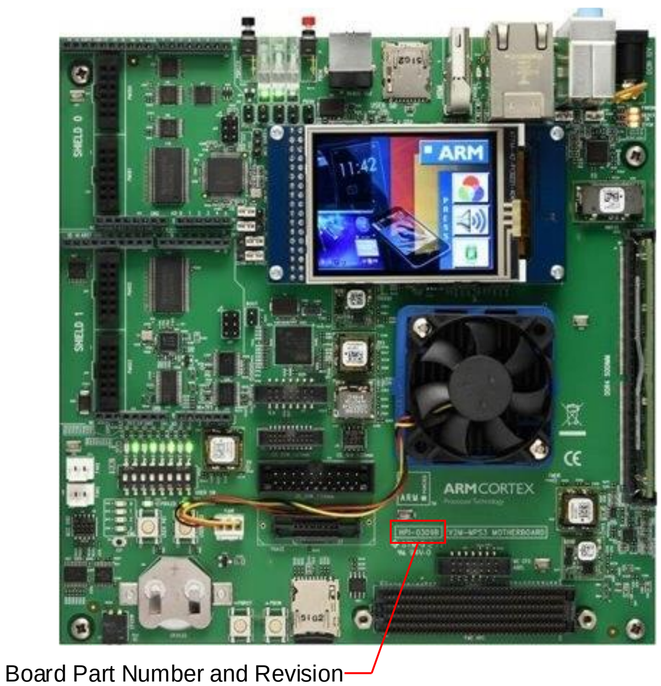


Figure 11-1 : MPS3 board revision identifier

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.

11.2 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 11.1 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 are used by the MCC. Care must be taken to ensure that the correct directory contents are modified if required.

12 Modifying and building AN524

12.1 Partial reconfiguration

AN524 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 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 AN524 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, (`524_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: Since the top level is preplaced and routed, only the mps3_fpga_user partition is placed and routed.
4. PR bitfile is produced for the mps3_fpga_user PR partition.

The following two files are produced for any PR partition:

- `524_uc_X.bit` : The clearing bitstream to clear the appropriate part of FPGA configuration memory.
- `524_u_X.bit` : The programming bitstream.

5. Top level static encrypted bitfile is downloaded to MPS3 board.
6. Two user PR partition bitfiles are downloaded to MPS3 board.
7. SSE-200 subsystem boots.

12.4 Flow detail

The user partition code is located in `<install_dir>/Luna/Logical/Resources/mps3_user_peripheral/AN524`. The top-level file, `mps3_fpga_user.v` is further located in the `user_wrapper` directory.

The following procedure describes how to build a new version of AN524:

1. 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`
2. Navigate to `<install_dir>/Luna/FPGA/AN524/smm_toplevel/xilinx/scripts`
3. If different version numbers are required for the planned bitfiles, then edit `user_pr_impl.tcl` and set the variable *FPGA_BUILD* to the desired single digit number
   Note : The version number of the supplied files is 3. The default value of *FPGA_BUILD* set in the user scripts is 3. Therefore, in order to avoid any new bitfiles overwriting the pre-compiled files it is suggested that the value of *FPGA_BUILD* is modified.
4. For a Linux system, execute:

   ```
   $ ./user_pr_flow.scr
   ```

   For a Windows system, execute:

   ```
   > user_pr_flow.bat
   ```

   from the Vivado HLS Command Prompt.
5. When the flow has completed it produces two bit files, `524_u_X.bit` and `524_uc_X.bit`. These will be written to the `<install_dir>Luna/FPGA/AN524/smm_toplevel/Xilinx/netlist_user` directory. *The* "X" will equate to the value of *FPGA_BUILD* written into `user_pr_impl.tcl`.
6. Copy the new bitfiles `524_u_X.bit`, and `524_uc_X.bit` to the directory `<MPS3_dir>MB/HBI0309C/AN524/` on the MPS3 board.
7. Edit the configuration file `an524_v3.txt` in the same directory to use the new files

   ```
   F1FILE: 524_uc_3.bit    ;FPGA1 Filename - clear system PR - change this line
   F1MODE: FPGA            ;FPGA1 Programming Mode
   F2FILE: 524_u_3.bit     ;FPGA2 Filename - write system PR- change this line
   F2MODE: FPGA            ;FPGA2 Programming Mode
   ```

   Here 524_uc_3.bit and 524_u_3.bit are the files provided with the AN524 zip bundle.

8. 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 AN524 on the MPS3 board

13.1 Loading a prebuilt FPGA image onto the MPS3 board

The following procedure describes how to load the pre-built AN524 image:

1. Power up the MPS3 board using the PBON push button and wait for the V2M_MPS3 drive to appear.
2. 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
3. Note : You can 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/AN524/images.txt`
4. Eject the V2M_MPS3 volume from your computer to unmount the drive.
5. Power cycle the MPS3 board using the PBRST push button and then launch firmware update and FPGA configuration by pressing PBON push button. The LEDs flash rapidly to indicate that new firmware is being downloaded (this only occurs the first time when the firmware is being 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, 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, the debug UART0 terminal simultaneously shows the selftest menu for Application Note AN524.
6. If the MPS3 board does not boot correctly, 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

Four serial ports are supported on this implementation and are accessible through the MPS3 board Debug USB port:

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

Note

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.

13.3 UART Serial Port Terminal Emulator Settings

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

Baud Rate:	115200 bps
New-Line:	CR
Data:	8 bits
Parity:	none
Stop:	1 bit
Flow control:	none

See 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.

13.4 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/developer/tools-software/oss-platforms/w/docs/589/accessing-mps3-serial-ports-in-windows-10

14 Software

In the Arm® Keil® µVision®, 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.

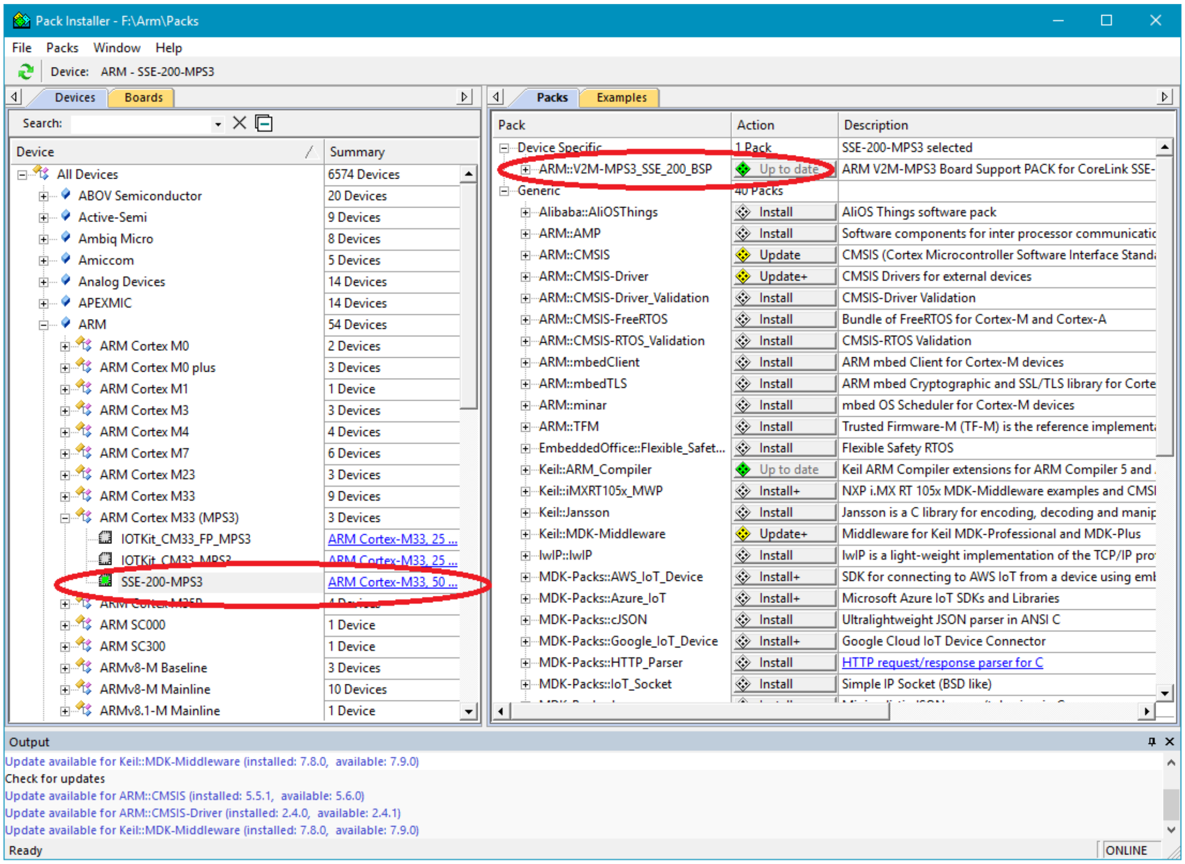


Figure 14-1 : Keil MDK pack installation

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:

1. Navigate to `<install_dir>/Software/YYY/Build_keil/`
2. Load `YYY.uvprojx` (where YYY will be selftest or demo , depending on which project is chosen) in Keil uVision
3. Once loaded, the project can be rebuilt by selecting either:
   - Project - > Build Target
   - Project - > Rebuild all target files

- The output can then be found in `<install_dir>/Software/selftest/Build_keil/an524_XX.axf` (where XX will be st or dm depending on which project is being built)

14.2 Loading software on 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:

1. Cop the software `<install_dir>/Software/selftest/Build_keil/an524_XX.axf to the board <MPS3_dir>/Software` folder
2. Navigate to `<MPS3_dir>MB/HBI0309C/AN524` and open the `images.txt` file in a text editor
3. Uncomment the test you wish to run and make sure the others are commented out, for example.

```
IMAGE0FILE: \SOFTWARE\an524_st.axf ; - selftest uSD

;IMAGE0FILE: \SOFTWARE\an524_dm.axf ; - demo uSD
```

(selftest image is uncommented, which is therefore selected and mem test is commented out)

The MPS3 can now be booted according to the instructions in the *Arm® MPS3 FPGA Prototyping Board Getting Started Guide* accompanying the MPS3 board.

15 Debug

15.1 Debug Connectivity

The following table shows the supported connectivity between the supported MPS3 Board debug connectors (See Figure 15-2 : MPS3 Board Debug Connector Locations for locating the connectors on board) and supported debug in the FPGA implementation:

Debug Connector Type	P-JTAG Debug	SWD	4-bit Trace	16-bit Trace
20 pin Cortex debug and ETM	Yes	Yes	Yes	Yes
20 pin IDC	Yes	Yes	Yes	No
Mictor 38	Yes	Yes	Yes	Yes

Table 15-1 : Debug Connectivity and Support

Debug has been tested using Keil uVision 5.27. To support warm reset over debug tool using Arm® Keil® ULINK™ Pro Armv8-M Debugger or CMSIS-DAP Armv8-M Debugger.

Apply the following debug settings :

Reset: HW RESET

Connect: without Stop

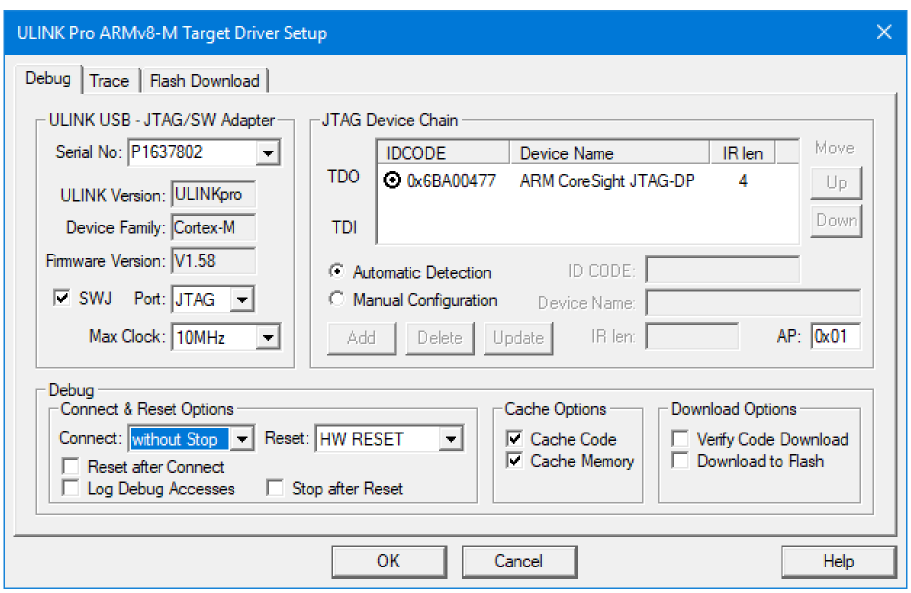


Figure 15-1 :Keil MDK debug configuration

15.2 Trace support for Keil MDK

The Keil Board support pack for SSE-200 on MPS3 ARM::V2M-MPS3_SSE_200_BSP version 1.0.0 does not support trace output in Arm® Keil® µVision®. Until it is updated in the future versions of the pack, the instructions on the community page can be followed to get the trace working :

https://community.arm.com/developer/tools-software/oss-platforms/w/docs/616/how-to-view-trace-for-application-note-an524-on-mps3-board

15.3 Debug and Trace support for Arm Development Studio

In Arm Development studio 2020.1 or above debug configurations can be found under Run>Debug Configurations.

15.3.1 Establishing a Debug Session

Following steps needs to be carried out to establish a debug connection :

1. Ensure the Arm DSTREAM debug probe is
    a. Powered, and connected to the host running the Development Studio software.
    b. Connected to the MPS3 using the 20-pin Cortex / 20-pin IDC / Mictor 38 port on the MPS3 as shown below:

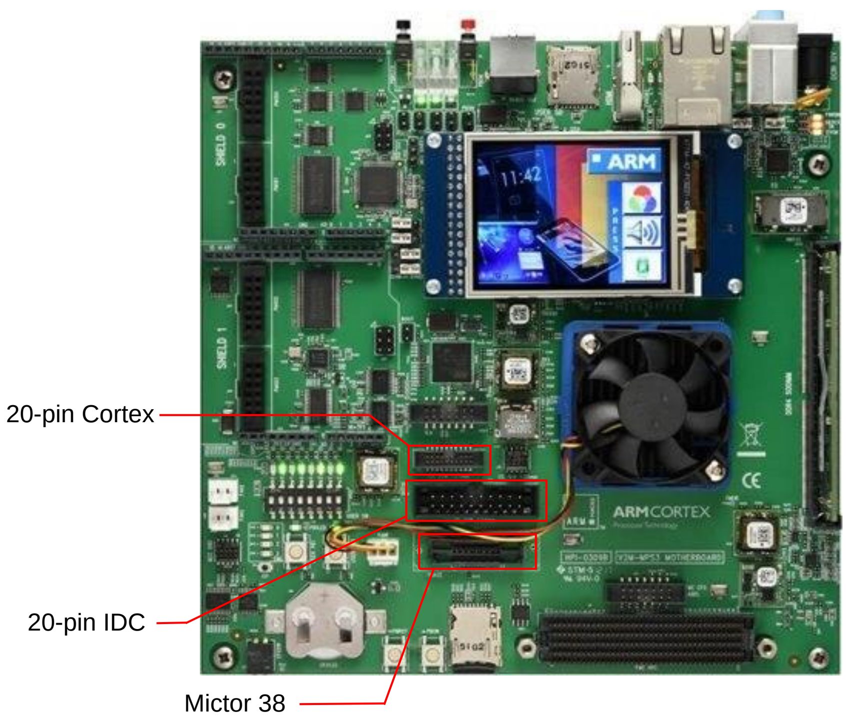
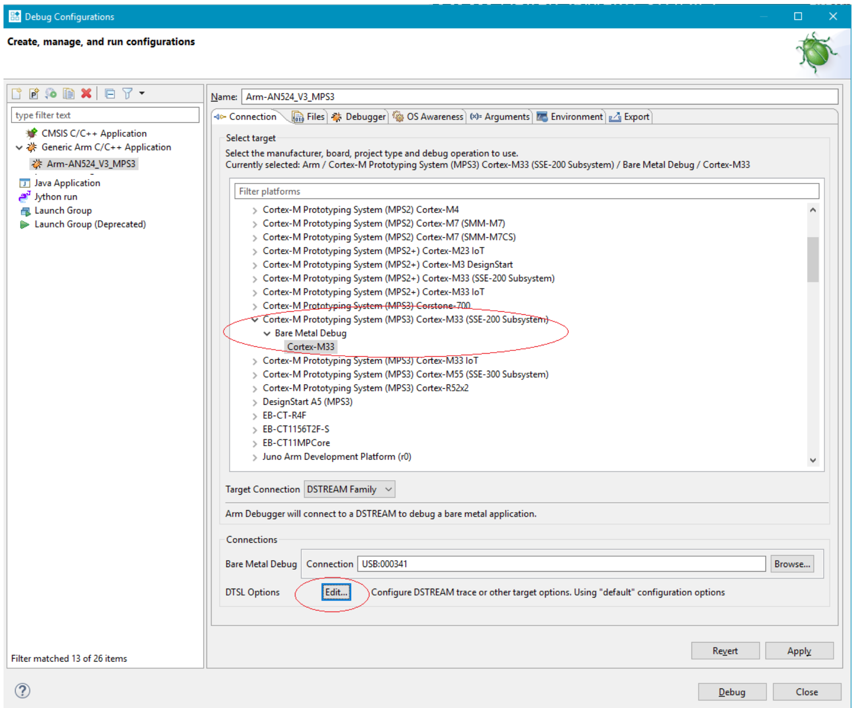
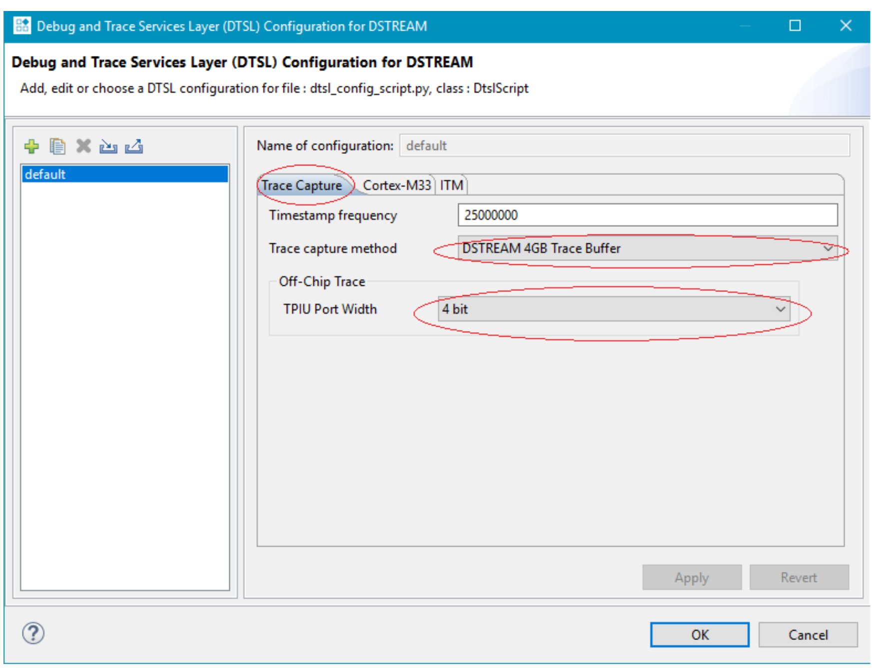


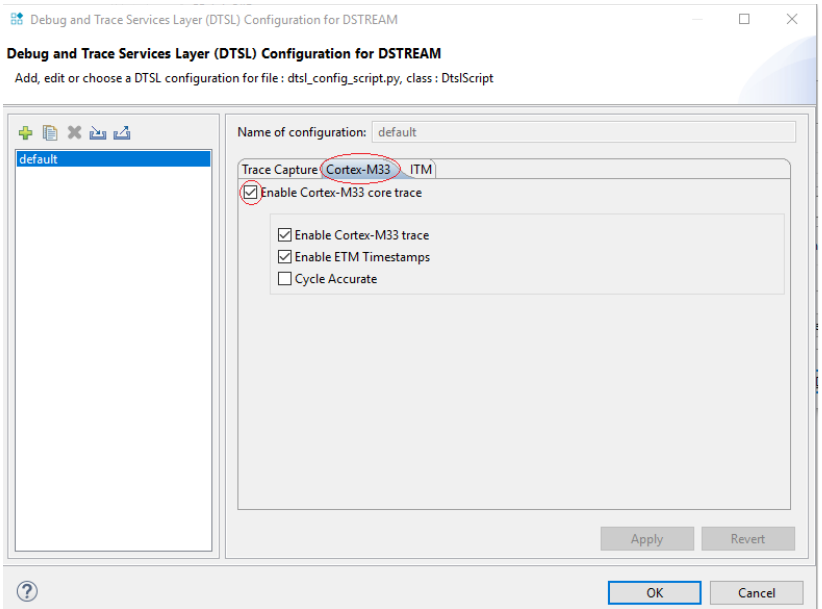
Figure 15-2 : MPS3 Board Debug Connector Locations

2. Select Arm Cortex-M Prototyping System (MPS3) Cortex-M33 (SSE-200 Subsystem)>Bare Metal Debug > Cortex – M33.



3. Enable trace capture by editing the DTSL options as below:





4. Click Ok, Apply and then Click on Debug. Trace should now be visible in the Trace window.