Application Note AN524

Example CoreLink™ SSE-200 Subsystem for MPS3

Non Confidential

Example CoreLink™ SSE-200 Subsystem for MPS3

Release Information

The following changes have been made to this Application Note.

Change History

Date	Issue	Confidentiality	Change
15 January 2018	A	Non Confidential	First release
19 February 2018	B	Non Confidential	Correct details of PR bitfile generation

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Contents

Example CoreLink™ SSE-200 Subsystem for MPS3

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.

`monospace` denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`
denotes arguments to commands and functions where the argument is to be replaced by a specific value.

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

italic highlights important notes, introduces special terminology, denotes internal cross-references, and citations.

bold highlights interface elements, such as menu names. Denotes signal names. Also used for emphasis 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.
- The number, DAI 0524B.
- 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 Information Center, together with knowledge articles and *Frequently Asked Questions* (FAQs).

Other information

- Arm Information Center, http://infocenter.Arm.com/help/index.jsp
- Arm Technical Support Knowledge Articles, http://infocenter.arm.com/help/topic/com.arm.doc.faqs/index.html
- Arm Support and Maintenance, http://www.arm.com/support/services/support-maintenance.php
- Arm Glossary, http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014g/index.html

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 References

- *Arm® CoreLink™ SSE-200 Subsystem for Embedded Technical Reference Manual (Arm 101104)*
- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual (Arm DDI 0571).*
- *Arm® Cortex®-M System Design Kit Technical Reference Manual (Arm DDI 0479)*
- *MCBQVGA-TS-Display-v12 – Keil MCBSTM32F200 display board schematic.*

2.3 Terms and abbreviations

CMSDK	*Cortex-M* System Design Kit.
DMA	Direct Memory Access.
MCC	Motherboard Configuration Controller.
RAM	Random Access Memory.
FPGA	Field Programmable Gate Array.
SCC	Serial Configuration Controller.
TRM	Technical Reference Manual.
APB	Advanced Peripheral Bus.
AHB	Advanced High-performance Bus.
RTL	Register Transfer Level.
SMM	Soft Macrocell Model.
MSC	Master Security Controller
PPC	Peripheral Protection Controller
EAM	Exclusive Access Controller
MPC	Memory Protection Controller
IDAU	Implementation Defined Attribution Unit

2.4 Subsystem version details

This SMM is generated using various packages. 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.
r3p0	**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	**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

Figure 2-1 : Module versions

2.5 Encryption key

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

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 cores, 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.

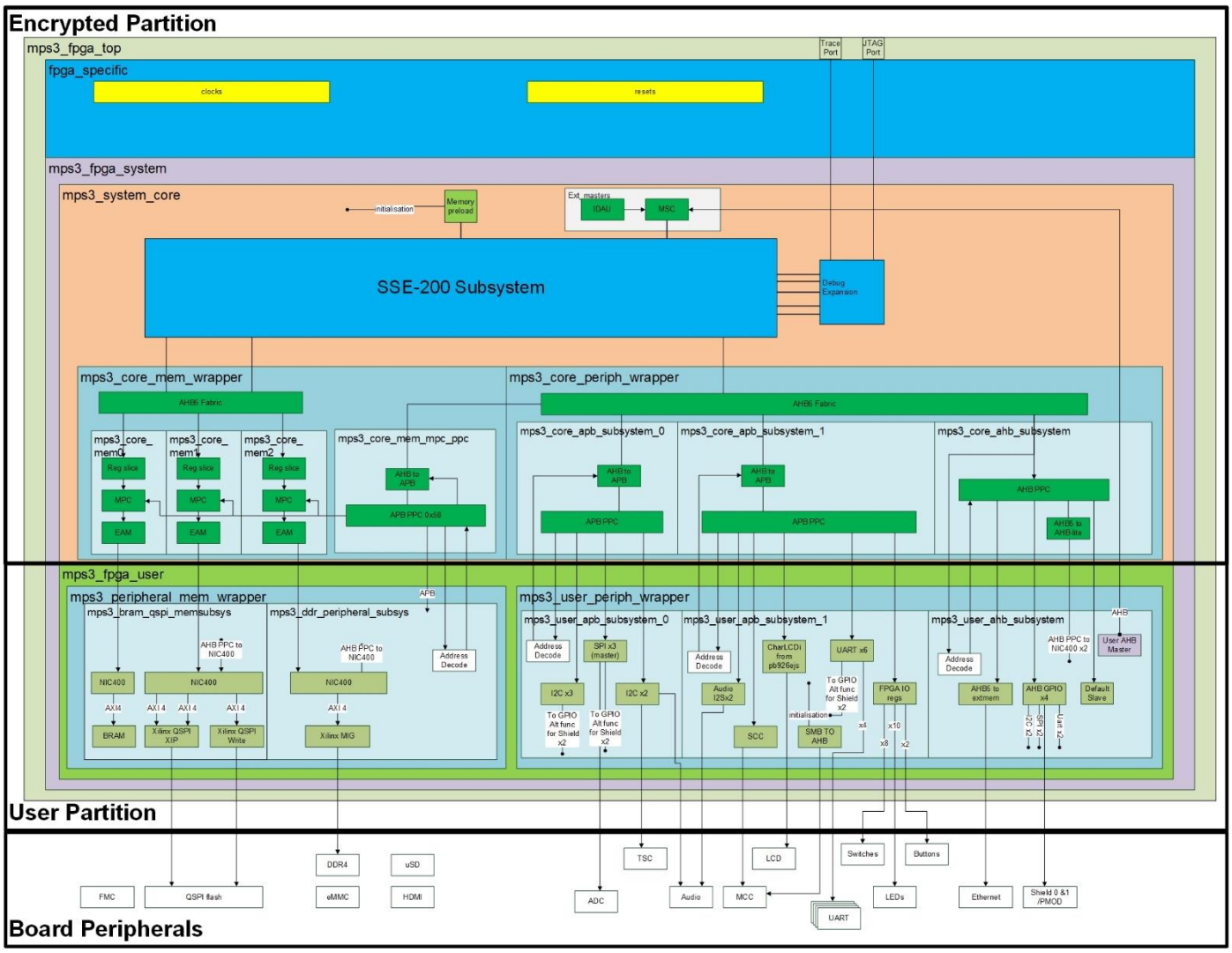

Figure 3-1 : System Overview

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.

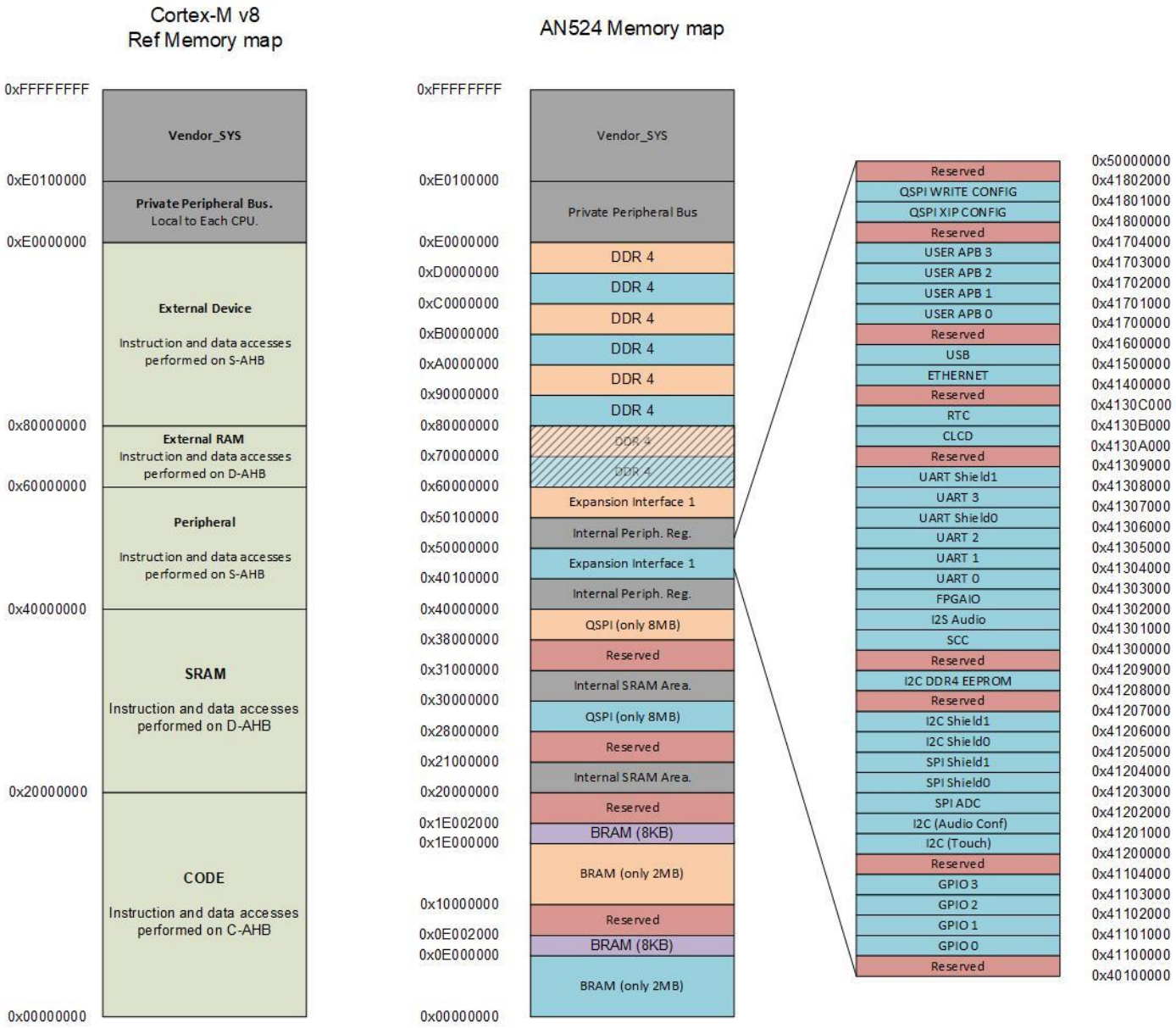

Figure 3-2 : Memory Map

ROW ID	Address From	To	Size	Region Name	Description	Alias With Row ID	IDAU Region Values Security	IDAUID	NSC
1	0x0000_0000	0x0DFF_FFFF	224MB	Code Memory	BRAM (only 2MB)	3	NS	0	0
2	0x0E00_0000	0x0FFF_FFFF	32MB	Reserved	Reserved				
3	0x1000_0000	0x1DFF_FFFF	224MB	Code Memory	Alias to BRAM (only 2MB)	1	S	1	CODE NSC2
4	0x1E00_0000	0x1FFF_FFFF	32MB	Reserved	Reserved				
5	0x2000_0000	0x20FF_FFFF	16MB	Internal SRAM	Internal SRAM Area.	8	NS	2	0
6	0x2100_0000	0x27FF_FFFF	112MB	Reserved	Reserved				
7	0x2800_0000	0x2FFF_FFFF	128MB	Expansion 0	QSPI (only 8MB)	10			
8	0x3000_0000	0x30FF_FFFF	16MB	Internal SRAM	Internal SRAM Area.	5	S	3	RAMNSC
9	0x3100_0000	0x37FF_FFFF	112MB	Reserved	Reserved				
10	0x3800_0000	0x3FFF_FFFF	128MB	Expansion 0	Alias to QSPI (only 8MB)	8			
11	0x4000_0000	0x4000_FFFF	64KB	Base Peripheral	Base Element Peripheral Region.	18	NS	4	0
12	0x4001_0000	0x4001_FFFF	64KB	Private CPU	CPU Element Peripheral Region.	19			
13	0x4002_0000	0x4002_FFFF	64KB	System Control	System Control Element Peripheral region.	20			
14	0x4003_0000	0x4003_FFFF		Reserved	Reserved				
15	0x4004_0000	0x4007_FFFF		Reserved	Reserved				
16	0x4008_0000	0x400F_FFFF	512KB	Base Peripheral	Base Element Peripheral Region.	23			
17	0x4010_0000	0x4FFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	24			
18	0x5000_0000	0x5000_FFFF	64KB	Base Peripheral	Base Element Peripheral Region.	11	S	5	0
19	0x5001_0000	0x5001_FFFF	64KB	Private CPU	CPU Element Peripheral Region.	12			
20	0x5002_0000	0x5002_FFFF	64KB	System Control	System Control Element Peripheral region.	13			
21	0x5003_0000	0x5003_FFFF		Reserved	Reserved				
22	0x5004_0000	0x5007_FFFF		Reserved	Reserved				
23	0x5008_0000	0x500F_FFFF	512KB	Base Peripheral	Base Element Peripheral Region.	16			
24	0x5010_0000	0x5FFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	17			
25	0x6000_0000	0x6FFF_FFFF	256MB	Expansion 0	DDR4	26	NS	6	0

ROW ID	Address From	Address To	Size	Region Name	Description	Alias With Row ID	IDAU Region Values Security	IDAUID	NSC
26	0x7000_0000	0x7FFF_FFFF	256MB	Expansion 0	DDR4	25	S	7	0
27	0x8000_0000	0x8FFF_FFFF	256MB	Expansion 0	DDR4	28	NS	8	0
28	0x9000_0000	0x9FFF_FFFF	256MB	Expansion 0	DDR4	27	S	9	0
29	0xA000_0000	0xAFFF_FFFF	256MB	Expansion 0	DDR4	30	NS	A	0
30	0xB000_0000	0xBFFF_FFFF	256MB	Expansion 0	DDR4	29	S	B	0
31	0xC000_0000	0xCFFF_FFFF	256MB	Expansion 0	DDR4	32	NS	C	0
32	0xD000_0000	0xDFFF_FFFF	256MB	Expansion 0	DDR4	31	S	D	0
33	0xE000_0000	0xE00F_FFFF	1MB	PPB	Private Peripheral Bus. Local to Each CPU.	35	Exempt		
34	0xE010_0000	0xEFFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	36	NS	E	0
35	0xF000_0000	0xF00F_FFFF	1MB	System Debug	System Debug.	33	Exempt		
36	0xF010_0000	0xFFFF_FFFF	255MB	Expansion 1	Maps to AHB5 Master Expansion 1 Interface	34	S	F	0

Table 3-1 : Memory map overview

3.5 REMAP

Remap control via SCC CFGREG0[0]:

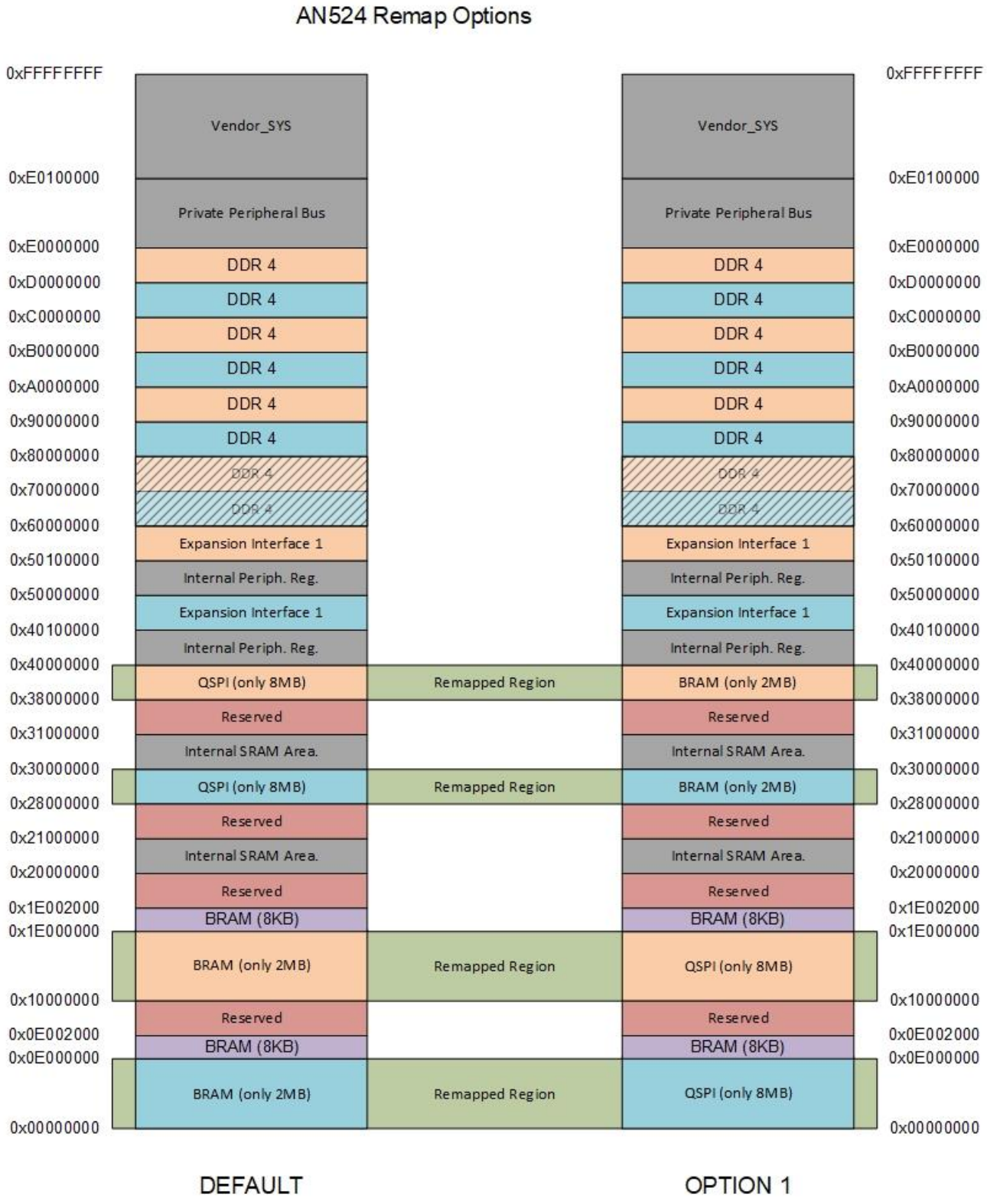

Figure 3-3 : Remap options

3.6 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	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	
	0x4110_4000	0x411F_FFFF		Reserved	
5	0x4120_0000	0x4120_0FFF	4K	FPGA - SBCon I2C (Touch)	**APB0**
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)	
10	0x4120_5000	0x4120_5FFF	4K	SBCon (I2C - Shiled0)	
11	0x4120_6000	0x4120_6FFF	4K	SBCon (I2C - Shiled1)	
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	**APB1**
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	
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	**APB1**
24	0x4130_A000	0x4130_AFFF	4K	CLCD Config Reg	
25	0x4130_B000	0x4130_BFFF	4K	RTC	
	0x4130_C000	0x413F_FFFF		Reserved	

ROW ID	Address From	Address To	Size	Description	Port
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	**APB (Mem)**
29	0x4170_1000	0x4170_1FFF	4K	User APB1	
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	
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	
	0x5110_4000	0x511F_FFFF		Reserved	
5	0x5120_0000	0x5120_0FFF	4K	FPGA - SBCon I2C (Touch)	**APB0**
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)	
10	0x5120_5000	0x5120_5FFF	4K	SBCon (I2C - Shiled0)	
11	0x5120_6000	0x5120_6FFF	4K	SBCon (I2C - Shiled1)	
12	0x5120_7000	0x5120_7FFF	4K	USER APB	
13	0x5120_8000	0x5120_8FFF	4K	DRR4 EEPROM	
	0x5120_9000	0x512F_FFFF		Reserved	
14	0x5130_0000	0x5130_0FFF	4K	FPGA - SCC registers	**APB1**
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	
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	**APB (Mem)**
29	0x5170_1000	0x5170_1FFF	4K	User APB1	
30	0x5170_2000	0x5170_2FFF	4K	User APB2	
31	0x5170_3000	0x5170_3FFF	4K	User APB3	
	0x5170_4000	0x5800_8FFF		Reserved	

ROW ID	Address		Size	Description	Port
	From	To			
32	0x5800_7000	0x5800_7FFF	4K	SSRAM Memory Protection Controller (MPC)	**APB (Mem)**
33	0x5800_8000	0x5800_8FFF	4K	SSRAM Memory Protection Controller (MPC)	
34	0x5800_9000	0x5800_9FFF	4K	SSRAM 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

The primary memory is 2MB of Internal FPGA SRAM, which is the default option for boot memory. This memory is mapped to the address range 0x00000000 - 0x001FFFFF by default and remaps to the address range 0x28000000 - 0x281FFFFF.

4.3 QSPI

The secondary memory is 8MB of external Flash memory which is accessed via a QSPI interface. This memory is mapped to the address-range 0x28000000 - 0x2FFFFFFF by default and remaps to the address range 0x00000000 - 0x001FFFFF

4.4 DDR4

The SMM also includes 4GB of External DDR4 memory which is allocated to the 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 (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 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.

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.

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	Information
0x4130A000	Write Command	A write to this address will cause a write to the LCD commands register. A read from this address will cause a read from the LCD busy register.
0x4130A004	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	Interrupt	Bit 0 indicates Access Complete (write 0 to clear). The bit is set if read data is valid. Bits [31:1] should be ignored.

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.
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.

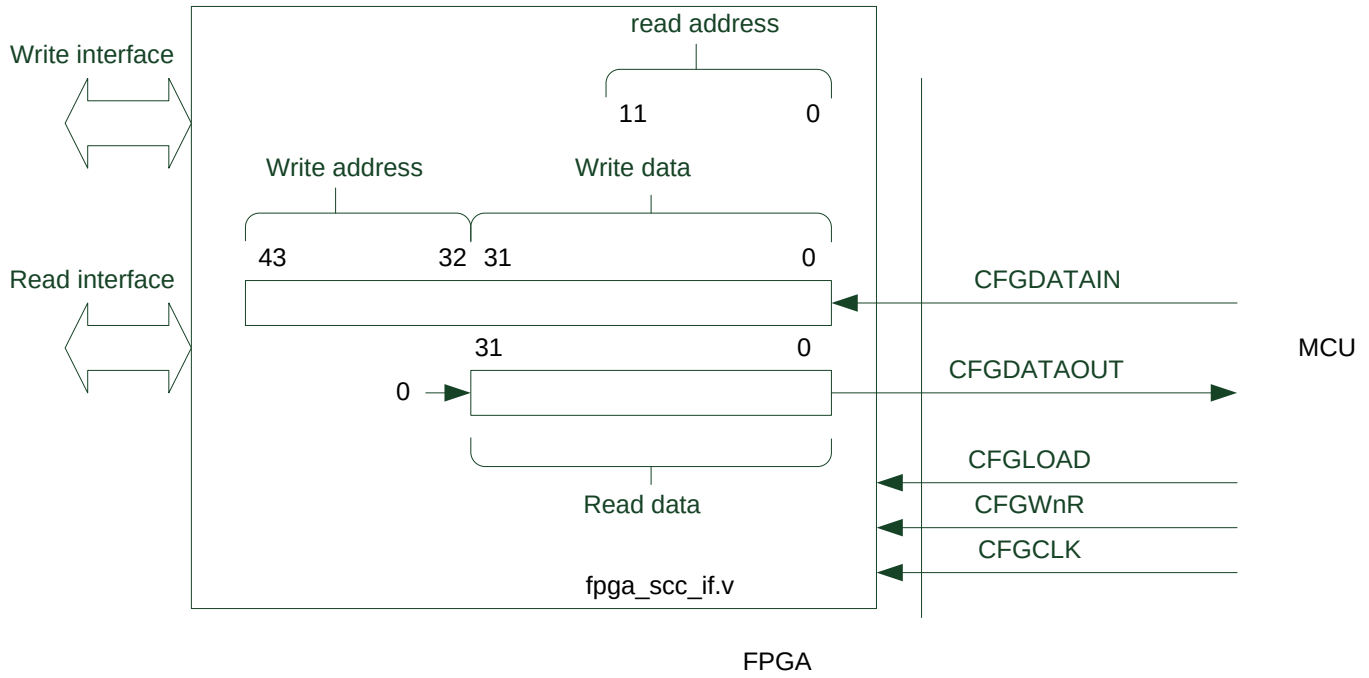

Figure 4-1 : Diagram of the SCC Interface

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
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 – 0x09C	RESERVED	-
0x0A0	SYS_CFGDATA_RTN	32bit DATA [r/w]
0x0A4	SYS_CFGDATA_OUT	32bit DATA [r/w]
0x0A8 – 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

Address	Name	Information
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[11:4] : Primary part number: 524 = AN524
		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 Clocks

5.1.1 Source clocks

The following clocks are inputs to the system.

Clock	Input Pin	Frequency	Note
REFCLK24MHZ	OSCCLK[0]	24MHz	24MHz reference
ACLK	OSCCLK[1]	32MHz	Programmable oscillator
MCLK	OSCCLK[2]	50MHz	Programmable oscillator
GPUCLK	OSCCLK[3]	50MHz	Programmable oscillator
AUDCLK	OSCCLK[4]	24.576MHz	Programmable oscillator
HDLCDCLK	OSCCLK[5]	23.75MHz	Programmable oscillator
DBGCLK	CS_TCK	Set by debugger	JTAG input
CFGCLK	CFG_CLK	Set by MCC	SCC register clock from MCC
DDR4_REF_CLK	c0_sys_clk_p/n	100MHz	Differential input clock to DDR4 controller
SMBM_CLK	SMBM_CLK	Set by MCC (40MHz)	SMB clock from MCC

Table 5-1 : Source clocks

5.1.2 User clocks

The following clocks are generated internally from the source clocks.

Clock	Source	Frequency	Note
MAINCLK	OSCCLK[1]	32MHz	
PERIF_CLK	OSCCLK[3]	50MHz	
AUDMCLK	AUDCLK	12.29MHz	
AUDSCLK	AUDCLK	3.07MHz	
SDMCLK	REFCLK24MHZ	50MHz	
CLK32KHZ	REFCLK24MHZ	32kHz	
CLK100HZ	REFCLK24MHZ	100Hz	
CLK1HZ	REFCLK24MHZ	1Hz	
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 is able to 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 - Shiled0)
6	SBCon (I2C – Shiled1)
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.

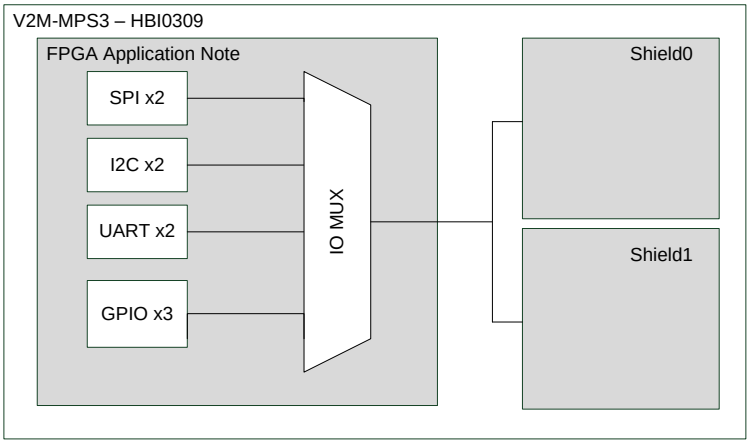

Figure 8-1 : Shield Device Expansion

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

MPS3	Proposed	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	Proposed	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 Modifying and building AN524

9.1 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, changing 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, (SSE200 subsystem), is delivered as a pre-compiled encrypted bitstream and cannot be modified.

In order for the user to be able to compile their modified versions of the mps3_fpga_user subsystem, a Xilinx DCP (Design Checkpoint) file is provided. 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 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.

9.2 Pre-requisites

To build the AN524 FPGA the user must have a licensed copy of Xilinx Vivado HLx Edition, version 2017.4 onwards. The license must also support partial reconfiguration.

The Vivado executable must be in the users path

9.3 Flow overview

The files provided to the user consist 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 is 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 524_uc_X.bit, (clearing), and 524_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. SSE-200 subsystem boots

9.4 Flow detail

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

The steps to build a new version of AN524 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 match 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/smm_toplevel/xilinx/scripts*
- 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 1. The default value of FPGA_BUILD set in the user scripts is 1. 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, *524_u_X.bit*, and *524_uc_X.bit.* These will be written to the *<install_dir>/Boardfiles/MB/HBI0309B/AN524* directory. "X" will equate to the value of *FPGA_BUILD* written into *user_pr_impl.tcl.*
- Copy the new bitfiles *524_u_X.bit*, and *524_uc_X.bit.* to the corresponding directory on the MPS3 board.
- Edit the configuration file *an524_v1.txt* in the same directory to use the new files
  F1FILE: 524_uc_1.bit ;FPGA1 Filename - clear system PR – change this line
  F1MODE: FPGA ;FPGA1 Programming Mode
  F2FILE: 524_u_1.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* files that the new files were successfully programmed.

The MPS3 board is now programmed with the user code.

10 Using AN524 on the MPS3 platform

10.1 Loading a prebuilt image onto the MPS3 platform

To load the pre-built AN524 images, follow these steps

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/HBI0309B/board.txt

Eject the V2M_MPS3 volume from your computer to unmount the drive

Power up the MPS3 platform using the 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 prebuild image is being downloaded onto the platform. Then the color LCD touch screen shows ARM CoreLink SDK-200 splash screen. Simultaneously if you have configured the UART to run a self test program, then the debug UART terminal shows the self-test menu for Application Note AN524.

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

11 Software

11.1 Rebuilding Software

Requirements

- The software directory from the download
- Keil uVision 5.24 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:
    - 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)

11.2 Loading software to the MPS3 Platform

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/an524_XX.axf* to the board *<MPS3_dir>/Software* folder
- Navigate to *<MPS3_dir>MB/HBI0309B/AN524* 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\an524_st.axf ; - selftest uSD

  ;IMAGE0FILE: \SOFTWARE\an524_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 Getting Started Guide accompanying the MPS3 board.