Application Note AN400

ARM® Cortex® - M7 SMM on V2M-MPS2

Non-Confidential

ARM® Cortex® - M7 SMM on V2M-MPS2

Release Information

The following changes have been made to this Application Note.

Change History

Date	Issue	Confidentiality	Change
14 January 2014	A	Non-Confidential	First release
14 November 2014	B	Non-Confidential	Corrected MCC switch register address
17 December 2015	C	Non-Confidential	External shield support added

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Contents

ARM® Cortex® - M7 SMM on V2M-MPS2

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 AN400C.
- 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.aeg0014-/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

This SMM is intended for developers/programmers/users who deploy hardware/software for a purpose.

These topics support the following chapters:

2.1 References

- *ARMv7-M Architecture Reference Manual ARMv7-A and ARMv7-R edition (ARM DDI 0403D) for Cortex-M products*
  http://infocenter.arm.com/help/topic/com.arm.doc.ddi0403c/index.html
- *ARM® Versatile™ Express Cortex®-M Prototyping System (V2M-MPS2)Technical Reference Manual*
- *Cortex®-M System Design Kit*

  http://www.arm.com/products/processors/cortex-m/cortex-m-system-design-kit.php

  The Cortex®-M System Design Kit (CMSDK) is a product to help silicon and FPGA designers to create Cortex-M based systems. It contains ready-to-use example systems for Cortex-M processors and a range of AMBA® bus fabric components for Cortex-M system development.
- *Cortex®-M System Design Kit Technical Reference Manual*
  http://infocenter.arm.com/help/topic/com.arm.doc.ddi0479c/index.html
- *ARM® PrimeCell Synchronous Serial Port (PL022) Technical Reference Manual*
  http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0194g/ I1005344.html
- *Keil® MCBSTM32C Evaluation Board Display Board Schematic*
  http://www.keil.com/mcbstm32c/mcbstm32c-display-board-schematics.pdf
- *ARM V2M-Shield1 Technical Reference Manual*
  http://infocenter.arm.com/insert/link/here.pdf

2.2 Terms and abbreviations

Volatile (storage class qualifier)
: In C and C++ this is the volatile storage class qualifier. In other languages the syntax and semantics might vary slightly if the concept is supported at all. The intent is to cover all storage locations that might be used for inter-processor communication variables that can be used for lock-free programming.

CMSDK
: *Cortex-M* System Design Kit.

MCC
: Motherboard Configuration Controller

2.3 Encryption key

ARM supplies the V2M-MPS2 motherboard with a decryption key programmed into the FPGA. This key is needed to enable loading of the prebuilt images, which are encrypted.

Caution

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

3 Overview

The SMM design is based on the Cortex-M System Design Kit (CMSDK). Extra peripherals are placed in unused memory spaces so that most of the RTL and software in CMSDK can be reused.

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.

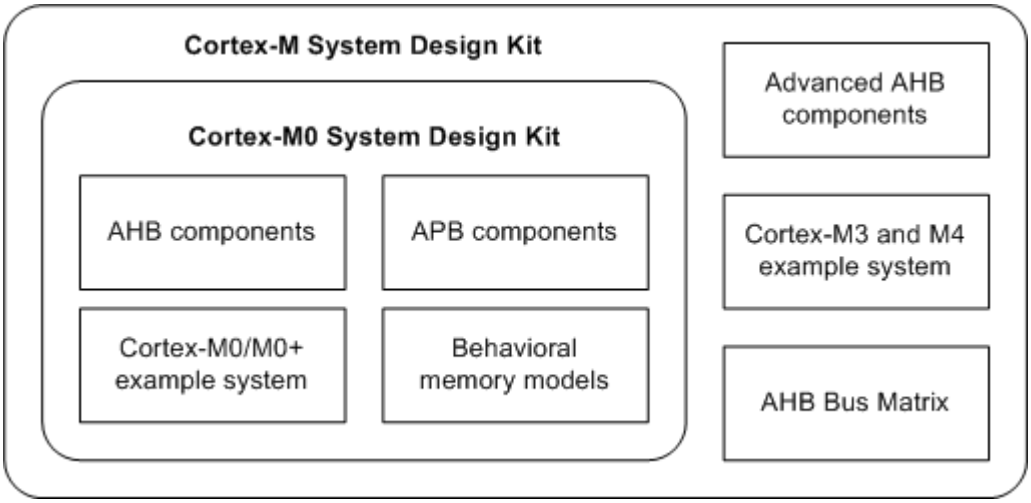

Figure 3-1 : System Design Kit Example Diagram

The documentation of CMSDK can be found in (ARM internal links):

No.	Documents
1	Cortex-M System Design Kit Component Technical Reference Manual http://arminfo.emea.arm.com/help/topic/com.arm.doc.ddi0479c/index.html

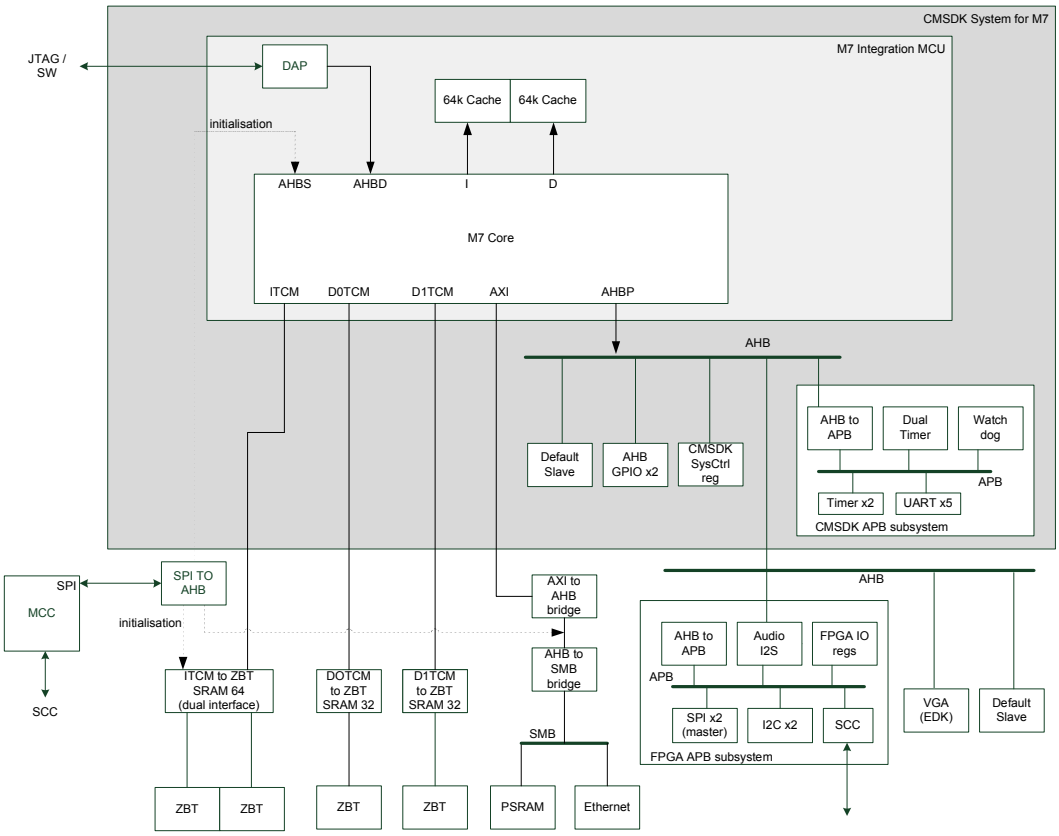

Figure 3-2 System Overview

The following topics describe the aspects of this subject:

- *Memory Map* on page 3-5
- *External ZBT Synchronous SRAM (SSRAM1)*on page 3-6
- *External ZBT Synchronous SRAM (SSRAM2 & SSRAM3)*on page 3-7
- *External PSRAM* on page 3-7
- *CMSDK APB subsystem* on page 3-8
- *AHB GPIO* on page 3.6
- *SPI (Serial Peripheral Interface)* on page 3-8
- *Color LCD parallel interface* on page 3-9
- *Ethernet* on page 3-9
- *VGA* on page 3-9
- *Audio I2S* on page 3-10
- *Audio Configuration* on page 3-11
- *FPGA system control and I/O* on page 3-11

3.1 Memory Map

The following table shows the memory map :

Start Address	End Address	Description	Comment
0xA0000000	0xA000FFFF	Ethernet (via ahb_to_extmem16. Offset 0x0 to 0x0FE for CSRs, 0x100 to 0x1FE for FIFO)	Not available in CMSDK
0x60000000	0x60FFFFFF	PSRAM (16MB)	Not available in CMSDK
0x41100000	0x4110FFFF	VGA Image (512x128) (AHB)	Not available in CMSDK
0x41000000	0x4100FFFF	VGA Console (AHB)	Not available in CMSDK
0x40030000	0x40FFFFFF	RESERVED	-
0x4002F000	0x4002FFFF	SCC register (see SCC section)	Not available in CMSDK
0x4002B000	0x4002EFFF	RESERVED	-
0x4002A000	0x4002AFFF	SBCon (Shield1), APB	Additional to CMSDK
0x40029000	0x40029FFF	SBCon (Shield0), APB	Additional to CMSDK
0x40028000	0x40028FFF	FPGA System Control & I/O, APB	Not available in CMSDK
0x40027000	0x40027FFF	PL022 (Shield1 SPI), APB	Additional to CMSDK
0x40026000	0x40026FFF	PL022 (Shield0 SPI), APB	Additional to CMSDK
0x40025000	0x40025FFF	PL022 (External ADC SPI), APB	Additional to CMSDK
0x40024000	0x40024FFF	Audio I^2S, APB	Not available in CMSDK
0x40023000	0x40023FFF	SBCon (Audio Configuration), APB	Not available in CMSDK
0x40022000	0x40022FFF	SBCon (Touch for LCD module), APB	Not available in CMSDK
0x40021000	0x40021FFF	PL022 (SPI for LCD module), APB	Not available in CMSDK
0x40020000	0x40020FFF	PL022 (SPI), APB	Not available in CMSDK
0x4001F000	0x4001FFFF	CMSDK system controller	CMSDK system controller
0x40014000	0x4001EFFF	Reserved for extra GPIO / other AHB peripherals	Unused
0x40013000	0x40013FFF	CMSDK AHB GPIO #3	Identical to CMSDK
0x40012000	0x40012FFF	CMSDK AHB GPIO #2	Identical to CMSDK

Start Address	End Address	Description	Comment
0x40011000	0x40011FFF	CMSDK AHB GPIO #1	Identical to CMSDK
0x40010000	0x40010FFF	CMSDK AHB GPIO #0	Identical to CMSDK
0x40000000	0x4000FFFF	CMSDK APB subsystem	Identical to CMSDK
0x20800000	0x20FFFFFF	RESERVED	-
0x20000000	0x207FFFFF	ZBTSRAM 2 & 3 (2x 32-bit). Reserved 8MB, 4MB available. The two SRAM blocks are interleaved.	Only 64KB SRAM in CMSDK
0x01010000	0x1FFFFFFF	RESERVED	-
0x00800000	0x00FFFFFF	RESERVED	-
0x00400000	0x007FFFFF	ZBTSRAM 1 (64-bit). Wrapped (only 4MB ZBTSRAM fitted)	Unused in CMSDK.
0x00000000	0x003FFFFF	ZBTSRAM 1 (64-bit)	Flash memory in CMSDK example system is 64KB. This is increased to 4MB in this SMM.

Table 3-1 – Detailed System Memory Map

Note 1 The microcontroller on the V2M-MPS2 board controls the zbt_boot_ctrl signal. The zbt_boot_ctrl signal overrides the boot option to enable the ZBT RAM to be used.

Many parts of the memory map have the same programmer's view as CMSDK.

All components listed below are connected to the AHBP interface unless otherwise stated.

3.2 External ZBT Synchronous SRAM (SSRAM1)

This section describes the Fast Program SRAM in the CODE region.

This is interfaced to two external 32-bit ZBT SSRAM in parallel, forming a 64-bit ZBT SSRAM. 8MB of memory space are allocated, but only 4MB is used (each ZBT SSRAM is 2MB).

This memory space connects through the ITCM interface.

3.3 External ZBT Synchronous SRAM (SSRAM2 & SSRAM3)

The Fast ZBT SSRAM in SRAM region is set up as two external ZBT SSRAMs, connected to two independent ZBT interfaces. In the 8MB memory region, 4MB of ZBT are available.

The address of the ZBT SSRAM is interleaved as shown in the table below.

Upper 32-bit ZBT SSRAM3	Lower 32-bit ZBT SSRAM2
0x207FFFFC (wrap round to 0x203FFFFC)	0x207FFFF8 (wrap round to 0x203FFFF8)
…	…
0x20400004 (wrap round to 0x20000004)	0x20200000 (wrap round to 0x20000000)
0x203FFFFC	0x203FFFF8
…	…
0x2000000C	0x20000008
0x20000004	0x20000000

Table 3-2 – 32 bit ZBT Memory Map

This memory space connects through the DTCM interface.

3.4 External PSRAM

A 16MB 16-bit PSRAM area is available and the memory map allocates the address-range 0x60000000 - 0x60FFFFFF. This enables large test programs to be used, for example *uClinux*, in the External RAM region of the Cortex-M memory space.

Note: Running code from SRAM region is slower than from CODE region because the internal bus structure is not optimized for running programs from this region.

This memory space connects through the AXI interface.

3.5 CMSDK APB subsystem

The SMM uses APB subsystem in CMSDK.

Address	Item	Notes
0x4000F000-0x4000FFFF	APB expansion port 15	Not used. Reserved for micro DMA controller configuration port
0x4000E000-0x4000EFFF	APB expansion port 14	Not used
0x4000D000-0x4000DFFF	APB expansion port 13	Not used
0x4000C000-0x4000CFFF	APB expansion port 12	Not used
0x4000B000-0x4000BFFF	APB test slave	For validation of AHB to APB bridge
0x4000A000-0x4000AFFF	Not used	Ports on APB slave multiplexer disabled
0x40009000-0x40009FFF	UART4	-
0x40008000-0x40008FFF	Watchdog	-
0x40007000-0x40007FFF	UART3	-
0x40006000-0x40006FFF	UART2	-
0x40005000-0x40005FFF	UART1	Not used
0x40004000-0x40004FFF	UART0	-
0x40003000-0x40003FFF	Not used	Port on APB slave multiplexer disabled
0x40002000-0x40002FFF	Dual timer	-
0x40001000-0x40001FFF	Timer1	-
0x40000000-0x40000FFF	Timer0	-

Table 3-3 – APB Memory Map

3.6 AHB GPIO

The SMM uses CMSDK AHB GPIO #0 and #1. See the CMSDK TRM

3.7 SPI (Serial Peripheral Interface)

The SMM implements five PL022 SPI modules:

- One general purpose SPI module that connects to the general-purpose SPI connector, J21.
- Three general purpose SPI modules that connect to the Expansion headers J7 and J8. Intended for use with the V2C-Shield1 where provide an interface with the ADC and SPI on the headers.
- Color LCD module control.

Self-test provided with the MPS2 includes example code for the color LCD module control interface.

Chip Selects are controlled my SCC register ‘fpga_misc’, rather than the PL022 chip select output. Please see *Table 3-6 – System Control and I/O Memory Map* for more details.

3.8 Color LCD parallel interface

The color LCD module has two interfaces:

- SPI for LCD module used for sending image data to the LCD.
- I^2C for touch used to transfer data input via the touch screen.

These interfaces are connected to a STMicroelectronics STMPE811QTR Port Expander with Advanced Touch Screen Controller on the Keil MCBSTM32C display board. (Schematic listed in the reference section).

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

3.9 Ethernet

The SMM design connects SMSC LAN9220 through AXI to external memory block.

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

This memory space connects through the AXI interface.

3.10 VGA

Address	Description
0x41000000 - 0x4100FFFF	Writes to the current location of the cursor.
0x41100000 - 0x4110FFFF	512x128 image area at the top right of the screen. 0x41100000 is the top left of the area and 0x4110FFFF is the bottom right. HADDR[16:2] = YYYYYYYXXXXXXXX where X and Y are the horizontal and vertical pixel offset respectively.

Table 3-4 – VGA Memory Map

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

The pixel in the top left hand corner of the display occupies address 0x41100000 with each successive row using an offset of 0x00000400 from the previous row. For example: the left most pixel (LMP) of the 2nd row is at 0x41100400 and the LMP of the 3rd row is at 0x41100800.

3.11 Audio I^{2}S

A simple FIFO interface generates and receives I^{2}S audio.

Address	Name	Information
0x40024000	CONTROL	Control Register [31:18] : Reserved [17] : Audio CODEC reset control (output pin) [16] : FIFO reset [15] : Reserved [14:12] : RX Buffer IRQ Water Level - Default 2 (IRQ triggers when more less 2 word space available) [11] : Reserved [10: 8] : TX Buffer IRQ Water Level - Default 2 (IRQ triggers when more than 2 word space available) [7: 4] : Reserved [3] : RX Interrupt Enable [2] : RX Enable [1] : TX Interrupt Enable [0] : TX Enable
0x40024004	STATUS	Status register [31:6] : Reserved [5] : RX Buffer Full [4] : RX Buffer Empty [3] : TX Buffer Full [2] : TX Buffer Empty [1] : RX Buffer Alert (Depends on Water level) [0] : TX Buffer Alert (Depends on Water level)
0x40024008	ERROR	Error status register [31:2] : Reserved [1] : RX overrun - write 1 to clear [0] : TX overrun/underrun - write 1 to clear
0x4002400C	DIVIDE	Divide ratio register (for Left/Right clock) [31:10] : Reserved [9: 0] LRDIV (Left/Right) Default = 0x80 12.288MHz / 48KHz / 2 (L+R) = 128
0x40024010	TXBUF	Transmit Buffer FIFO Data Register (WO) [31:16] : Left Channel [15: 0] : Right Channel

Address	Name	Information
0x40024014	RXBUF	Receive Buffer FIFO Data Register (RO) [31:16] Left Channel [15: 0] Right Channel
0x40024018 – 0x400242FC	RESERVED	-
0x40024300	ITCR	Integration Test Control Register [31:1] : Reserved [0] : ITCR
0x40024304	ITIP1	Integration Test Input Register 1 [31:1] : Reserved [0] : SDIN
0x40024308	ITOP1	Integration Test Output Register 1 [31:4] : Reserved [3] : IRQOUT [2] : LRCK [1] : SCLK [0] : SDOUT

Table 3-5 - Audio I^2S Memory Map

3.12 Audio Configuration

The SMM implements a simple SBCon interface based on I^2C.

3.13 FPGA system control and I/O

The SMM implements an FPGA system control block.

Address	Name	Information
0x40028000	FPGAIO->LED0	LED connections [31:2] : Reserved [1:0] : LED
0x40028004	RESERVED	
0x40028008	FPGAIO->BUTTON	Buttons [31:2] : Reserved [1:0] : Buttons
0x4002800C	RESERVED	
0x40028010	FPGAIO->CLK1HZ	1Hz up counter
0x40028014	FPGAIO->CLK100HZ	100Hz up counter

Address	Name	Information
0x40028018	FPGAIO->COUNTER	Cycle Up Counter Increments when 32-bit prescale counter reach zero.
0x4002801C	FPGAIO->PRESCALE	Bit[31:0] – reload value for prescale counter.
0x40028020	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.
0x40028024	RESERVED	
0x4002804C	FPGAIO->MISC	Misc control [31:10] : Reserved [9] : SHIELD1_SPI_nCS [8] : SHIELD0_SPI_nCS [7] : ADC_SPI_nCS [6] : CLCD_BL_CTRL [5] : CLCD_RD [4] : CLCD_RS [3] : CLCD_RESET [2] : RESERVED [1] : SPI_nSS [0] : CLCD_CS

Table 3-6 – System Control and I/O Memory Map

4 Clocks

The following table shows the Source Clocks for the system.

Name	Frequency
OSCCLK[0]	50MHz
OSCCLK[1]	24.576MHz
OSCCLK[2]	25MHz
CFGCLK	0.5MHz
CS_TCK	Determined by debugger
SPICFGCLK	7.5MHz

Table 4-1 : Source Clocks

The following table shows the Derived Clocks for the system.

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

Table 4-2 : Derived Clocks

5 Interrupt assignments

The SMM uses the following Interrupt assignments. This is a change from the default CMSDK assignments:

Number	Interrupt source
NMI	Watchdog
0	UART 0 receive interrupt
1	UART 0 transmit interrupt
2	UART 1 receive interrupt
3	UART 1 transmit interrupt
4	UART 2 receive interrupt
5	UART 2 transmit interrupt
6	GPIO 0 combined interrupt
7	GPIO 1 combined interrupt
8	Timer 0
9	Timer 1
10	Dual Timer
11	SPI #0, SPI #1
12	UART overflow (0, 1 & 2)
13	Ethernet
14	Audio I^2S
15	Touch Screen
16	GPIO 2 combined interrupt
17	GPIO 3 combined interrupt
18	UART 3 receive interrupt
19	UART 3 transmit interrupt
20	UART 4 receive interrupt
21	UART 4 transmit interrupt
22	SPI #2
23	SPI #3, SPI #4
24 – 31	GPIO 0 individual interrupts (0-7)

Table 5-1 : Interrupts

6 Serial Communication Controller (SCC)

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

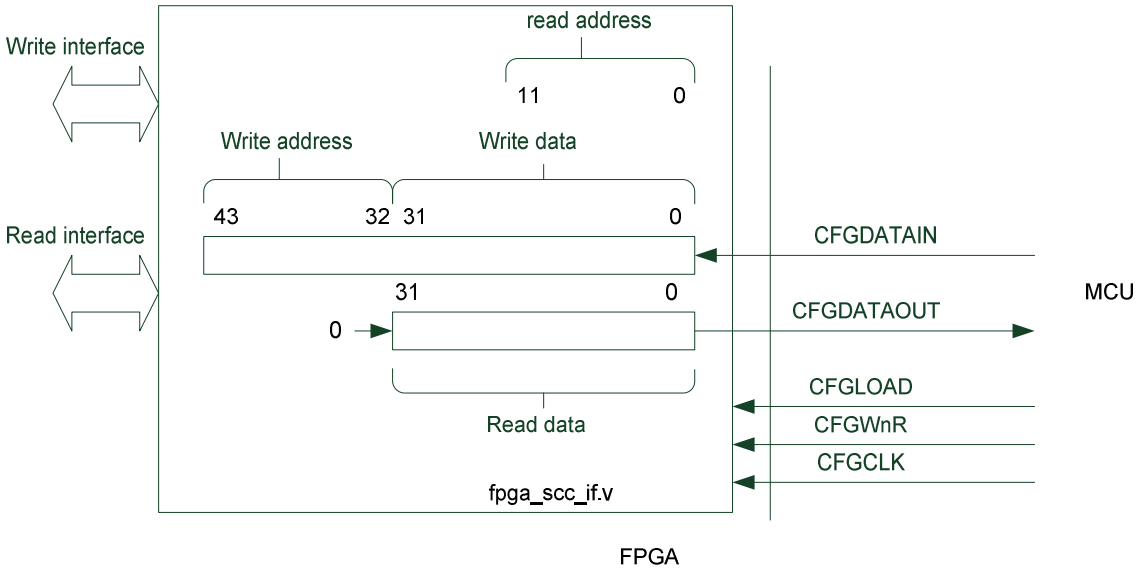

Figure 6-1 : Diagram of the SCC Interface

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[31:0] Reserved
0x004	CFG_REG1	Bits [31:8] Reserved Bits [7:0] MCC LEDs: 0 = OFF 1 = ON
0x008	CFG_REG2	Bits[31:26] Reserved Bits [25:0] CFGSTCALIB Systick calibration Default Bit[25] HIGH Bit[24:0] LOW
0x00C	CFG_REG3	Bits [31:8] Reserved Bits [7:0] MCC switches: 0 = OFF 1 = ON
0x010	CFG_REG4	Bits [31:4] Reserved Bits [3:0] : Board Revision

Address	Name	Information
0x014	CFG_REG5	Control bits for CPU Bits [31:15] Reserved Bit[14] CFGBIGEND D-side endianess config Bits[13:12] INITTCMEN TCM enables out of reset Bit[10:11] INITRMWEN TCM RMW enables out of reset Bits[9:10] INITRETRYEN TCM retry enables out of reset Bit[8] INITAHBPEN AHBP enable out of reset Bit[7] WICENREQ Request to enable WIC Bit[6] WICENACK Acknowledgement WIC enabled Bit[5] DAPEN debug access port enable Bit[4] CFGJTAGnSW Bit[3] Reserved Bit[2] Reserved Bit[1] Reserved Bit[0] : Processor enable. If it is cleared, processor will be in reset state. If this register is not setup by the board configuration file, these bits are reset as 0x1 (enabled), and cannot be changed unless LT_LOCK is written as 0xA05F.
0x018	CFG_REG6	Bits [31:27] Reserved Bit[26] Lockup Bits[25:4] I cache bank information Bits [3:0] I cache detection information
0x01C	CFG_REG7	Bits [31:26] Reserved Bits[25:4] D cache bank information Bits [3:0] D cache detection information
0x020 – 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	Bit 0 : Complete Bit 1 : Error
0x0AD – 0x0FC	RESERVED	-

Address	Name	Information
0x100	SCC_DLL	DLL lock register Bits [31:24] DLL LOCK MASK[7:0] - These bits indicate if the DLL locked is masked. Bits [23:16] DLL LOCK MASK[7:0] - These bits indicate if the DLLs are locked or unlocked. Bits [15:1] : Reserved Bit[0] This bit indicates if all enabled DLLs are locked:
0x104 – 0xFF4	RESERVED	-
0xFF8	SCC_AID	SCC AID register is read only Bits[31:24] : FPGA build number Bits[23:20] : V2M-MPS2 target board revision (A = 0, B = 1, C = 2) Bits[19:8] Reserved Bits[7:0] number of SCC configuration register
0xFFC	SCC_ID	SCC ID register is read only Bits[31:24] : Implementer ID: 0x41 = ARM Bits[23:20] : Application note IP variant number (note 1) Bits[19:16] : IP Architecture: 0x4 =AHB Bits[11:4] : Primary part number: 386 = AN386 Bits[3:0] : Application note IP revision number (note 1)

Table 6-1 – SCC Register memory map

note 1 The variant and revision numbers relate to the rxpy number. For example for r1p0 processors the 1 would be the variant number and the 0 would be the revision number.

7 Shield Support

This SMM can support up to two external shield devices with the addition of the ARM V2C-SHIELD (HBI-0289) expansion board to the V2M-MPS2.

To enable the Shield support, three SPI, three UART and two I2C interfaces are multiplexed with GPIO over the Expansion Headers.

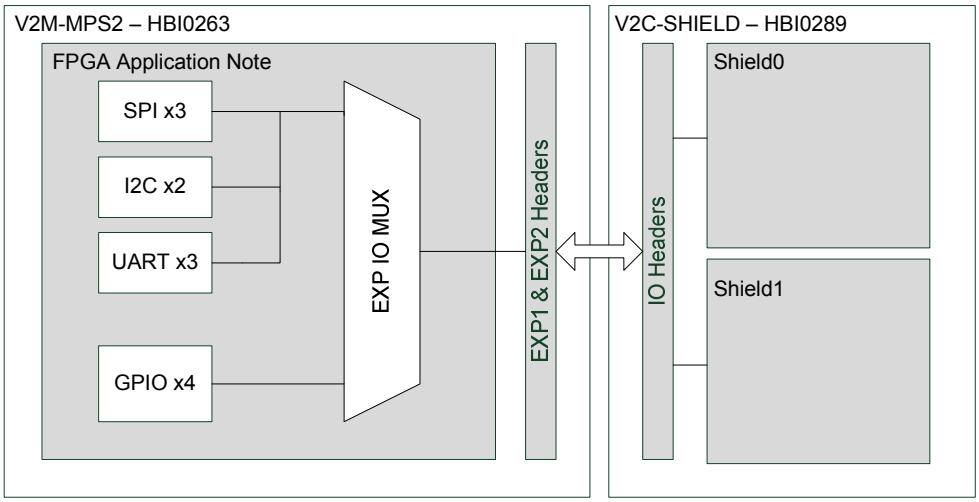

Figure 7-1 : Shield Device Expansion

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

EXP Signal	GPIO Source Port	Alternative Function	Description
EXP[0]	IPO GPIO0 [0]	UART3 RXD	Shield0 UART Receive
EXP[4]	IPO GPIO0 [4]	UART3 TXD	Shield0 UART Transmit
EXP[5]	IPO GPIO0 [5]	SBCON2 SCL	Shield0 I2C Clock
EXP[15]	IPO GPIO0 [15]	SBCON2 SDA	Shield0 I2C Data
EXP[11]	IPO GPIO0 [11]	SPI3 SCK	Shield0 SPI Clock
EXP[12]	IPO GPIO0 [12]	SPI3 SS	Shield0 SPI Chip Select
EXP[13]	IPO GPIO0 [13]	SPI3 MOSI	Shield0 SPI Data Out
EXP[14]	IPO GPIO0 [14]	SPI3 MISO	Shield0 SPI Data In
EXP[26]	IPO GPIO1 [10]	UART4 RXD	Shield1 UART Receive
EXP[30]	IPO GPIO1 [14]	UART4 TXD	Shield1 UART Transmit
EXP[31]	IPO GPIO1 [15]	SBCON3 SCL	Shield1 I2C Clock
EXP[41]	AHB GPIO2 [9]	SBCON3 SDA	Shield1 I2C Data
EXP[38]	AHB GPIO2 [6]	SPI4 SS	Shield1 SPI Chip Select
EXP[39]	AHB GPIO2 7]	SPI4 MOSI	Shield1 SPI Data Out
EXP[40]	AHB GPIO2 [8]	SPI4 MISO	Shield1 SPI Data In
EXP[44]	AHB GPIO2 [12]	SPI4 SCK	Shield1SPI Clock
EXP[16]	IPO GPIO1 [0]	SPI2 SS	ADC SPI Chip Select
EXP[17]	IPO GPIO1 [1]	SPI2 MISO	ADC SPI Date In
EXP[18]	IPO GPIO1 [2]	SPI2 MOSI	ADC SPI Date Out
EXP[19]	IPO GPIO1 [3]	SPI2 SCK	ADC SPI Clock
EXP[23]	IPO GPIO1 [7]	UART1 RXD	XBEE UART Receive
EXP[24]	IPO GPIO1 [8]	UART1 TXD	XBEE UART Transmit

Table 7-1 : Shield Alternative Function Pinout

8 Configurations

This SMM implements a variation of the M7 core. There are two variations available. Each implements a number of the configurable options of the M7 as listed in section 2 of the *ARM® Cortex®-M7 Processor Integration and Implementation Manual* (ARM DII 0239).

1. SMM-M7 (AN399): This variation implements the following features of M7
    - Single and Double precision Floating Point Unit (FPU).
    - 16 region Memory Protection Unit (MPU).
    - Instruction only ETM interface.
    - Debug Access Port (DAP) CM7DAP which is a low gate-count DAP implementation (Further information can be found in the reference section).
2. SMM-M7CS (AN400): This variation implements the following features of M7
    - Single precision Floating Point Unit (FPU)
    - No MPU regions
    - Instruction and Data ETM Interface
    - CoreSight SoC-400. (Further information can be found in the reference section).

Both variants implement the following configurable options

- 32 interrupts
- Debug configuration with 4 DWT and 8 FPB comparators
- ITM and DWT trace functionality