V2S-R5 FPGA Engineering Specification

System Design Division

Document number: V2S-R5 EIS
Date of Issue: September 12, 2014
Authors: SDPL

Abstract

This document describes a Cortex-R5 Soft Microcell Module implemented using LogicTile Express 3MG V2F-1XV5 board.

Keywords

R5, FPGA, SMM, Versatile Express Platform

Contents

1 ABOUT THIS DOCUMENT

1.1 Change control

1.1.1 Current status and anticipated changes

1.1.2 Change history

The change history for this document is managed by Domino.Doc.

Brief descriptions of major changes are also described here:

	Status	Remark
1.0	First Release	
2.0	Update for r1p2	
3.0	PISMO removal	Extra remap option to support use without PISMO module.
3.1	Doc fix	Size of lowest area of memory corrected in memory map.

1.2 References

This document refers to the following documents.

Ref	Doc No	Author(s)	Title
[1]	ARM DDI 0416A	ARM Ltd.	Primecell Generic Interrupt Controller (PL390) Revision:r0p0 TRM
[2]	ARM DDI 0424A	ARM Ltd.	Primecell DMA Controller (PL330) Revision:r1p0 TRM
[3]	ARM DDI 0246C	ARM Ltd.	Primecell Level 2 Cache Controller (PL310) Revision:r2p0 TRM
[4]	ARM DDI 0397E	ARM Ltd.	Primecell High-Performance Matrix (PL301) Revision:r1p2 TRM
[5]	ARM DDI 0314H	ARM Ltd.	CoreSight Components TRM
[6]	ARM DDI 0380G	ARM Ltd.	Primecell Static Memory Controller (PL354) Revision:r2p1 TRM
[7]	ARM DDI 0418C	ARM Ltd.	Primecell DDR2 Dynamic Memory Controller (PL341) Revision:r0p1 TRM
[8]	ARM DDI 0293C	ARM Ltd.	Primecell Color LCD Controller (PL111) Revision:r0p2 TRM
[9]	ARM DDI 0397F	ARM Ltd.	AMBA Network Interconnect (NIC-301) Revision:r2p0 TRM
[10]	ARM DDI 0460D	ARM Ltd.	Cortex - R5

1.3 Terms and abbreviations

This document uses the following terms and abbreviations.

Term	Meaning
AMBA	Advanced Microcontroller Bus Architecture
ACP	AXI Coherence Port
APB	Advanced Peripheral Bus
ATB	Advanced Trace Bus
AXI	Advanced eXtensible Interface
CR5	Cortex-R5 processor
DAP	Debug Access Port
DMAC	Direct Memory Access Control
ECC	Error Correction Code
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macro-cell
GIC	Generic Interrupt Controller
HPM	High Performance Matrix
LLPP	Low Latency Peripheral Port
SCU	Snoop Control Unit
SVN	**S**ub**V**ersio**N**, a version control system
TCM	Tightly Coupled Memory
TPIU	Trace Port Interface Unit
VerSoC	**Ver**ification **S**ystem **o**n **C**hip
VIC	Vectored Interrupt Controller
VIL	(R5) VerSoC Integration Layer that encapsulates R5 Integration Layer
SMM	Soft Macro Model
FPGA	Field Programmable Gate Array
DMC	Dynamic Memory Controller
SMC	Static Memory Controller
ZBT RAM	Zero Bus Turnaround Ram

LTE-3MG	LogicTile Express 3MG

2 SCOPE

This document describes features that are unique to the Cortex-R5 Soft Macrocell Model (SMM) implemented on a LogicTile Express 3MG (V2F-1XV5). It will examine the contents of the SMM-R5, system interconnect, the clock structure, and specifics of the programmer's model directly relevant to SMM-R5 operation.

3 INTRODUCTION

The SMM-R5 is a Cortex-R5 derivative processor that implements (in addition to existing CR5 features) the following:

- AXI Coherency Port
- Configurable low-latency peripheral ports
- Split/Lock

An R5 cluster is a sub-system integrating two R5 cores each with L1 memory, and a R5-SCU. The R5 cluster incorporates an example integration level sub-system PL390 (Generic Interrupt Controller), and combinatorial AXI interconnects.

The example integration level also includes the ETM and CTI components.

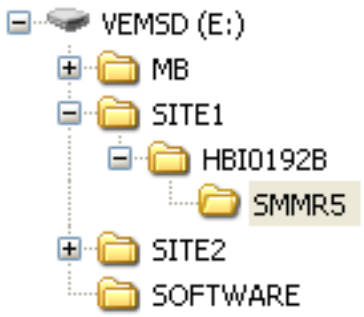
4 GETTING STARTED

1. The SMM R5 only operates on the ARM Versatile Express 3MG (V2F-1XV5) daughterboard placed on the Versatile Express motherboard (V2M-P1).
2. Ensure the Logictile Express 3MG daughterboard is plugged into Site 1 of the Versatile Express Motherboard as described in **Quick Start Guide for the Versatile Express Family - Adding Daughterboards**.
3. Connect USB, UART0 and power cable and power-up the boards as described in **Quick Start Guide for the Versatile Express Family - Powering up the System**.
4. Copy the SMMR5 FPGA images from the PC to the motherboard.
    a. To do this turn the USB controller of the motherboard ON by pressing the black Standby button on the motherboard and type '*usb_on*'

Cmd> usb_on

Enabling debug USB...

5. Access the motherboard from a PC by checking for a mass storage device, and go to SITE1/HBI0192B.



6. From the SMM R5 DVD installation directory, copy the directory of boardfiles/SITE1/HBI0192B/SMMR5 to the motherboards SITE1/HBI0192B/.
7. Change the contents of the motherboards SITE1/HBI0192B/board.txt to refer to the new SMMR5 directory:

    APPNOTE: SMMR5\smmr5.txt

8. Repeat the procedure for SITE1/HBI0192C.
9. Power cycle the motherboard.
10. The motherboard will configure the LTE-3MG daughterboard and the SMM R5 should now be running.

5 OVERVIEW

5.1 General overview

The SMM is based on R5 R1P2. The R5 Integration Level is used with additional logic wrapped around it.

5.1.1 Processor configurations

Processor feature	Configuration	Notes
I-cache	CPU 0 : 64KB CPU 1 : 64KB	-
D-cache	CPU 0 : 64KB CPU 1 : 64KB	-
MPU	CPU 0 : 12 regions CPU 1 : 12 regions	-
TCM	CPU 0 : 64KB x 2 CPU 1 : 64KB x 2	-
FPU	CPU 0 : Yes CPU 1 : Yes	-
DP_Float	CPU 0 : Yes CPU 1 : Yes	Single-precision and double-precision floating-point support
TCM High Init address	0x40000000	-
SCU	Present	-

Table 5.1 Processor Configuration

5.1.2 System level features

System feature	Configuration	Notes
Level 2 cache	Not present	-
AXI slave ports	present	Connected to DMA controller PL330 and external master
AXI LLPP	present	Connect to GIC, internal APB and MB peripherals
AHB LLPP	Not support	Tied off
GIC	Included	Connected to AXI LLPP in R5 Integration level
DMA	Included	Connected to AXI slave ports and ACP

Table 5.2 System level features

5.1.3 Debug features

Debug feature	Configuration	Note
ETM	Present	One for each processor
CTI	Present	One for each processor
CoreSight DK	Present	Include CSTPIU, ITM, SWO, CSETB
ROM table	Two	Primary ROM table for CSSYS Secondary ROM table for processor system

Table 5.3 Debug features

5.2 FPGA Hardware

The FPGA platform is based on an ARM LogicTile Express 3MG (V2F-1XV5) daughterboard with Versatile Express V2M-P1 baseboard.

The processor, memory controllers and the main system logic are implemented in the FPGA. The daughterboard and the motherboard are connected using the Static Memory Bus (SMB) interface. The majority of the slow speed peripherals are implemented on the motherboard (UARTs etc).

A DDR2 PISMO module may be connected on the V2F-1XV5 daughterboard. The module can provide fast local DDR2 memory and previously was supplied with the V2F-1XV5 daughterboard. DDR2 PISMO modules are no-longer available however existing modules are still supported.

5.3 System Level Design

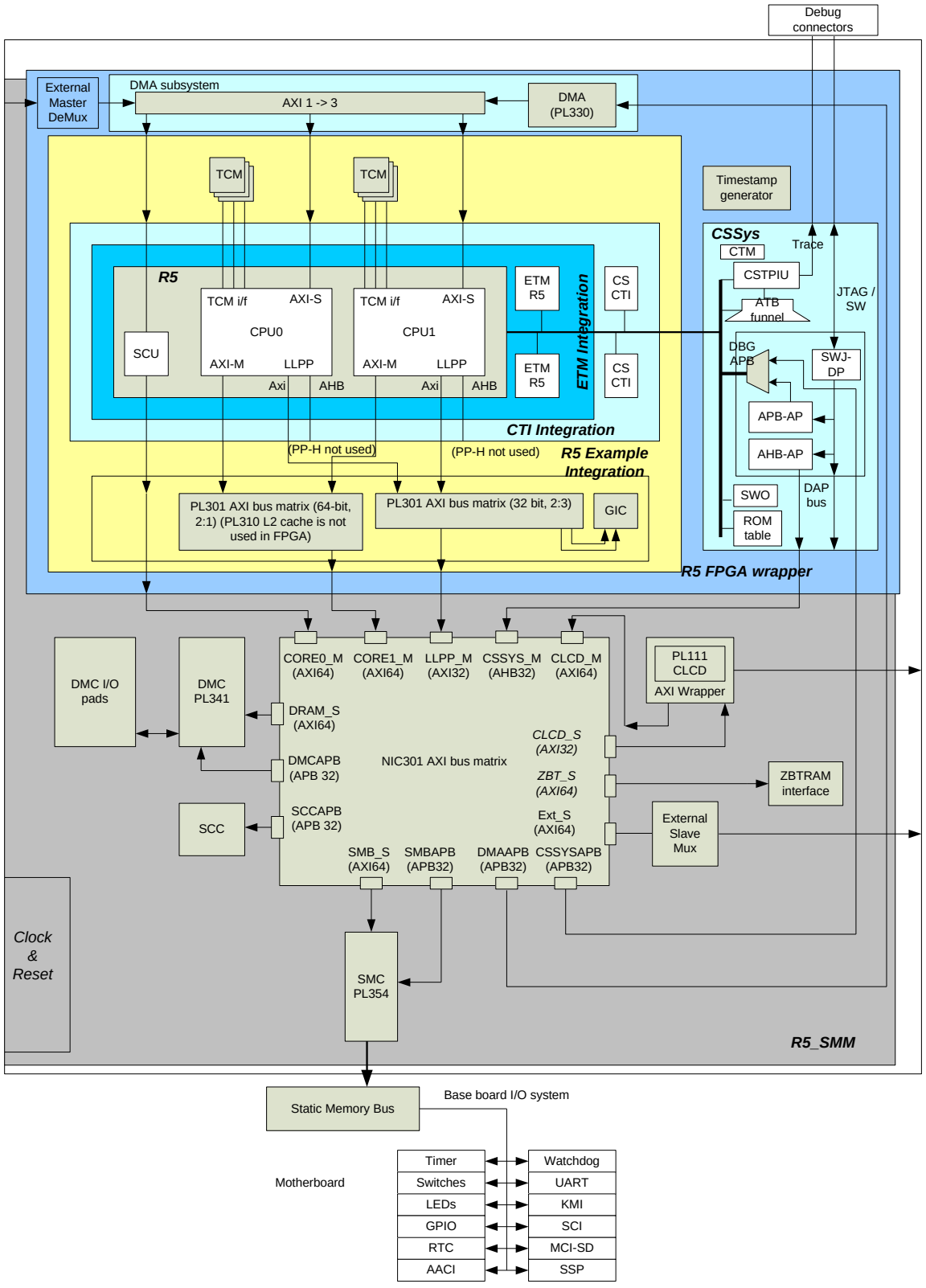


Fig 5.1 System level design

The design is based on the R5 integration level, CSSYS and a number of system components.

5.4 Clock architecture

The clock architecture is carefully designed to minimize the skew (difference) in the clock edge position between different components across the system. The DCMs and clock loops on LogicTile Express 3MG (V2F-1XV5) have been used to achieve this.

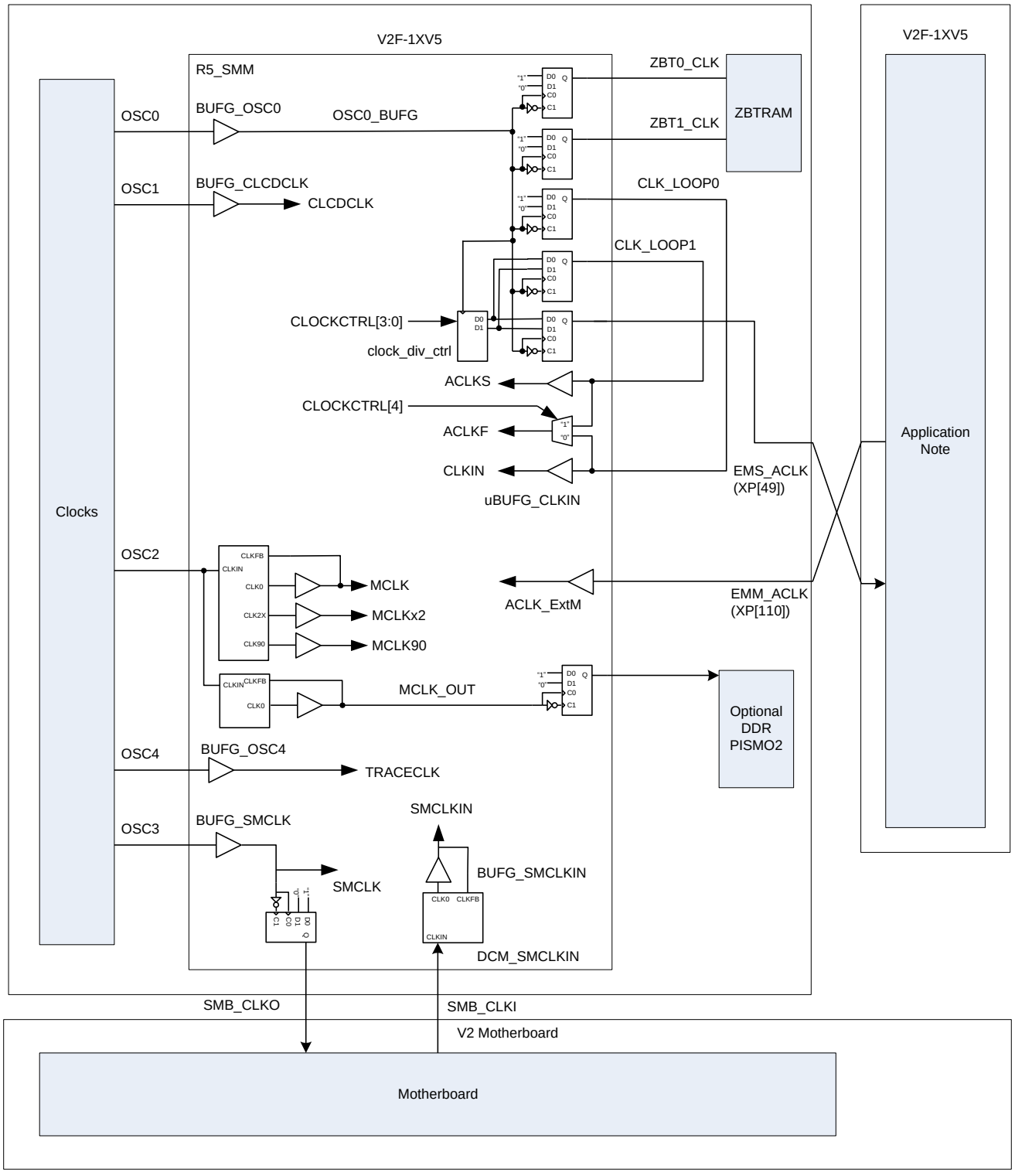


Figure 5.2 Clock architecture

5.4.1 Clock Domains

There are 8 clock domains in this design:

CLKIN

OSC0 is the source for CLKIN which is used to clock CPU and CoreSight components. The clock loop LOOP0 is used to de-skew CLKIN with other clocks generated from OSC0.

ACLKF

This clock domain is synchronous to CLKIN and is driven by OSC0.

ACLKF can be driven by clock loop LOOP0 or LOOP1, this depends on the state of bit 4 of SCC_CLKCTRL SCC register.

If the bit is LOW the ACLKF is the same frequency as CLKIN. If the bit is HIGH the ACLKF is the same frequency as the ACLKS and the ratio between ACLKF and CLKIN can be selected using SCC_CLKCTRL SCC register. The ACLKF is used to clock the AXI matrix, ZBT memory, AXI port of DDR memory controller, CLCD and DMA.

ACLKS

ACLKS is divided version of OSC0. The clock loop LOOP1 is used to de-skew ACLKS to CLKIN and ACLKF. The clock divider is controlled by SCC_CLKCTRL SCC register.

The ACLKS is used to clock external slave AXI interface and APB peripherals.

CLCDCLK

CLCDCLK is directly connected to OSC1. It is reference clock for the CLCD controller. The frequency of this clock must be adjusted to match target screen resolution.

MCLK

OSC2 is a source for two DMC modules. The first DMC module generates MCLK, MCLKX2 and MCLK90 signals used to clock PL341 DMC controller. The second DCM is used to generate MCLK_OUT and provides clock to DDR memory devices in phase with MCLK.

SMCLK

The SMCLK is used to drive Static Memory Controller and Static Memory bus. The SMCLK is directly connected to OSC3. The SMCLKIN feedback clock is used to register data from Static memory bus slave implemented on V2 Motherboard.

TRACECLK

TRACECLK is directly connected to OSC4.

EMM_ACLK

The External AXI Slave Bus is clocked by EMM_ACLK generated by the 2nd daughterboard if fitted.

5.4.2 Resets

There are 2 resets issued by the motherboard in this design:

CB_nRST

The reset used to reset all the AXI domains. This is released after CB_nPOR.

CB_nPOR

This reset is used for resetting the Coresight components and clock components.

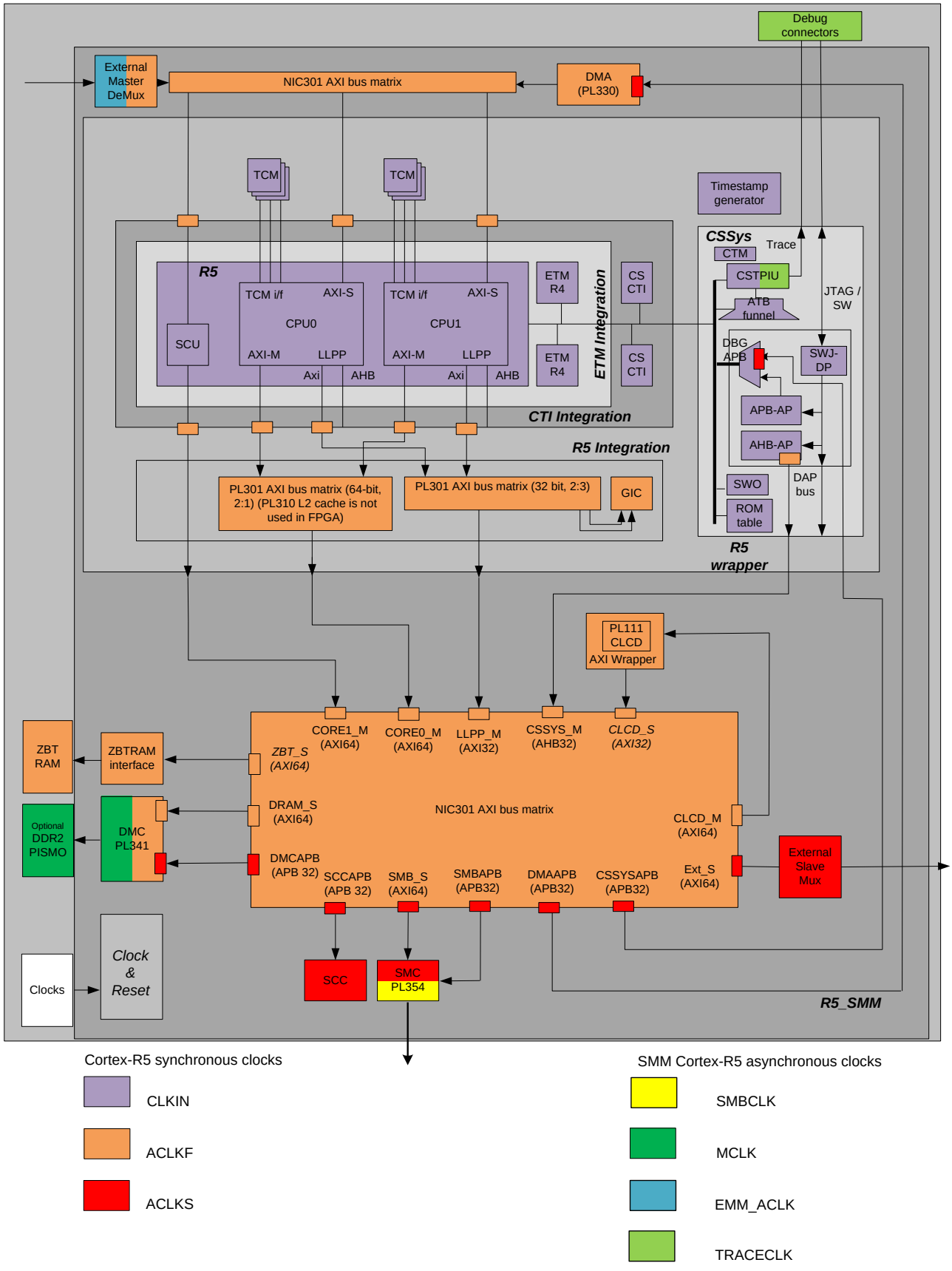


Figure 5.3 Clock domains

5.4.3 Default, minimum and maximum operating frequencies

Clock source	Clock signal	Clock domain	Default Freq	Min Freq	Max Freq
OSC0	CLKIN	CPU, CoreSight	50MHz	2MHz	60MHz
OSC0	ACLKF	AXI Matrix, AXI buses, ZBT RAM, DMA	50MHz	2MHz	60MHz
OSC0	ACLKS	APB, AXI External Slave	25MHz	1MHz	30MHz
OSC1	CLCDCLK	CLCD	28MHz	2MHz	62.5MHz
OSC2	MCLK	PISMO2 DDR,	110MHz	110MHz	110MHz
OSC3	SMCLK/SMCLKIN	Static Memory Bus	50MHz	32MHz	50MHz
OSC4	TRACECLK	Trace	100MHz	2MHz	100MHz
EMM_ACLK	ACLK_ExtM	Ext AXI Master Bus	-	2MHz	50MHz

Table 5.4 Default and maximum operating frequencies

6 MEMORY MAP

The memory map as viewed from the processor is as follows:

The processor cannot access the region between address 0xA6000000 to 0xAA000000. This is for the DMA controller and the external master only. However, the DMA controller and external master cannot access the Low Latency Peripheral Bus (LLPP), therefore they cannot access to the Generic Interrupt Controller (GIC).

The MB peripherals between 0xB0000000 to 0xBF000000 is unique to the implementation on the Versatile Express motherboard V2M-P1.

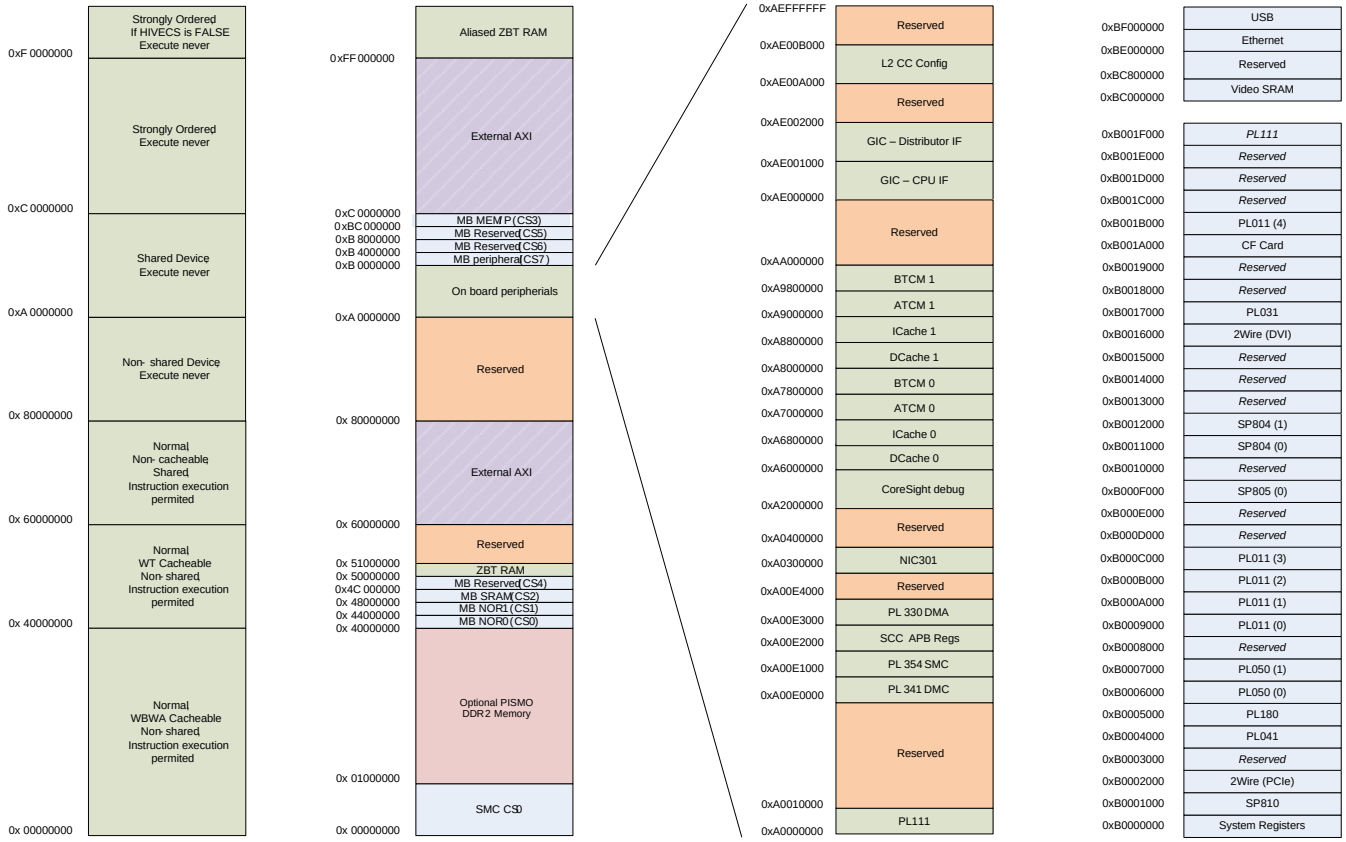


Fig 6.1 Memory Map

6.1 Memory remap

There are 5 remap options for NIC301 slave ports. These can be selected from the NIC-301 remap register found at 0xA0300000.

In addition to the remap options, it is also possible to swap remap memory between SMC CS0 NOR0 and SMC CS1 NOR1 by using the SMB_Remap signal driven from bit [8] of SCC_REMAP register at 0xA00E2004.

The remap options allow the user to:

- Enable different boot scenario (NOR0, NOR1, DDR2, Ext AXI)
- Set up V2 platform with two SMM-CortexR5
- Transfer AXI transaction from external master via ACP port.

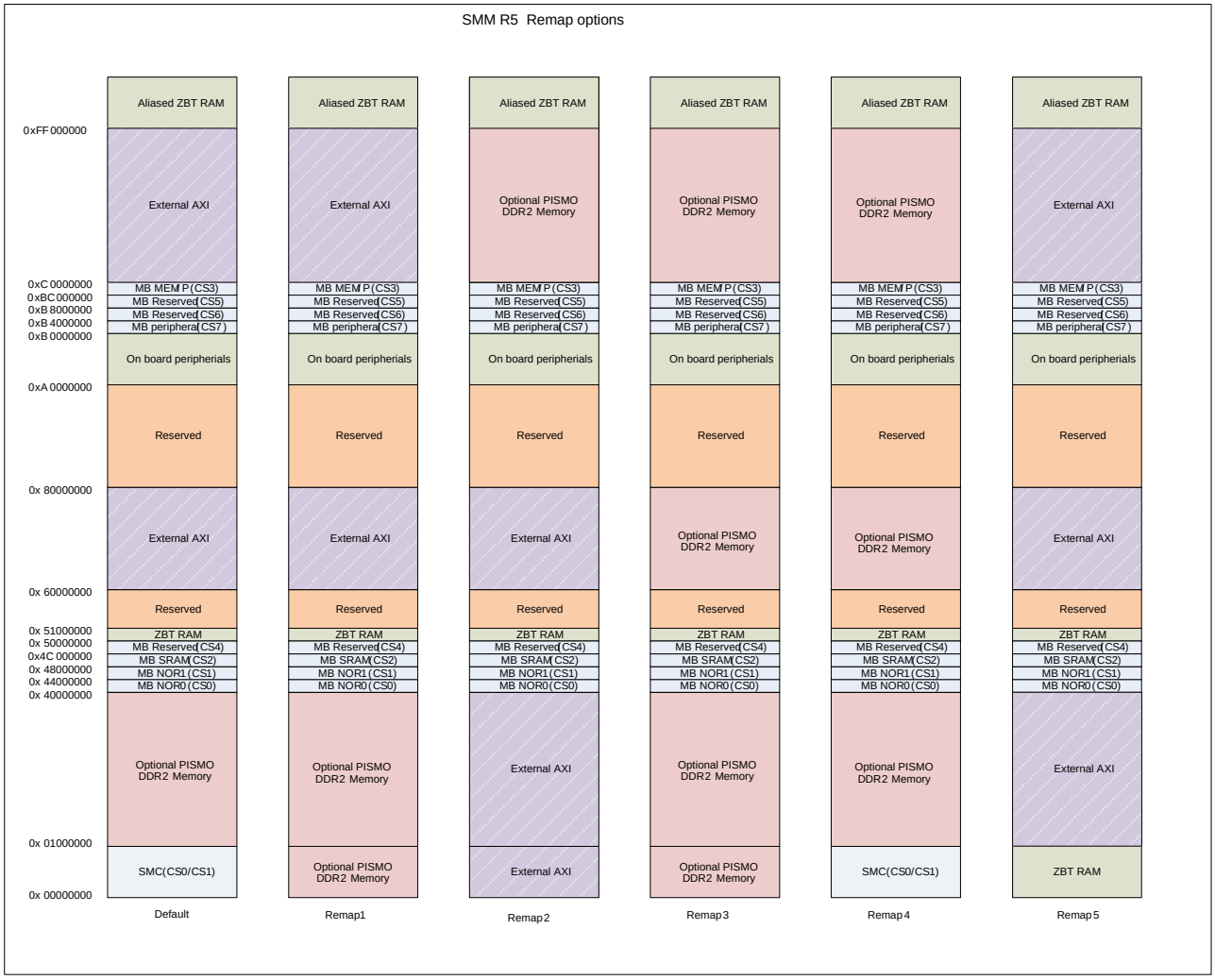


Fig 6.2 Remap options

NIC301 Remap[5:0]	SCC Remap[8]	Remap option	0x00000000-0x00FFFFFF	0x01000000-0x3FFFFFFF	0x60000000-0x7FFFFFFF	0xC0000000-0xFFEFFFFF
7'b00000XX	1'b0	Default	SMC CS0 NOR0	DMC	Ext AXI	Ext AXI
7'b00000XX	1'b1	Default	SMC CS1 NOR1	DMC	Ext AXI	Ext AXI
7'bXXXX1XX	1'bX	Remap1	DMC	DMC	Ext AXI	Ext AXI
7'bXXX10XX	1'bX	Remap2	Ext AXI	Ext AXI	DMC	DMC
7'bXX100XX	1'bX	Remap3	DMC	DMC	DMC	DMC
7'bX1000XX	1'b0	Remap4	SMC CS0 NOR0	DMC	DMC	DMC
7'bX1000XX	1'b1	Remap4	SMC CS1 NOR1	DMC	DMC	DMC
7'b10000XX	1'bX	Remap5	ZBT RAM	Ext AXI	Ext AXI	Ext AXI

Table 6.1 Remap options.

The NIC301 Remap register [1:0] allow limiting CORE0_M, CORE1_M, LLPP_M AXI ports accesses to some memory area.

If the NIC301 Remap[0] is HIGH:

CORE0_M, CORE1_M can only access 0x00000000-0x7fffffff and 0xff000000-0xffffffff memory ranges.

If the NIC301 Remap[1] is HIGH:

LLPP_M can access only 0x80000000-0xffffffff memory range.

6.2 ZBT address range

The ZBTRAM memory at the address 0x50000000 to 0x50FFFFFF supports a read only mode. This is controlled by a control bit in LT_SYSCFG in the APB register block. It is targeted for testing hardware break point.

6.3 Peripherals addresses

Address	Peripherals
0xA0000000	PL111 LCD controller
0xA00E0000	PL341 DMC
0xA00E1000	PL354 SMC
0xA00E2000	SCC APB Registers
0xA00E3000	PL330 DMA Controller
0xA0300000	NIC301
0xA2000000	CoreSight system
0xA6000000	Cache and TCM
0xAE000000	PL390 GIC CPU interface
0xAE001000	PL390 GIC Distributor interface
0xAE00A000	L2CC config

Table 6.2 Peripheral addresses

6.4 SCC/APB Register block

A number of control registers are implemented for system control. The registers can be accessed by APB bus as well as by SCC interface. The SCC interface allows initialization during power up sequence by values from daughter board configuration file.

Address offset	Register	Descriptions
0x0000	SCC_CONFIG0	Bits[31:5] Reserved Bit[4] ACLKF control – if enabled (0x1) the clock for processor slave and master AXI ports, DMA controller, AXI DMC, CLCD, ZBTRAM controller is divided by N. Otherwise the clock is equal to the processor clock. Bit[3:0] N clock divider for “slow AXI clock” use to clock AXI SMBC port and APB buses. It is also use to control ACLKF clock if bit [4] is set HIGH
0x0004	SCC_REMAP	Bit[31:9] Reserved. Bits[8] SMB Remap bit. If the register is not setup by the board configuration file, the reset value is 0x0. Bits[7:6] Reserved Bits[5:0] Software mirror to the NIC301 remap register
0x0008	SCC_CPU0CTRL	Bits[31:2] Reserved Bit[1] Reserved Bit[0] Processor enable. If it is cleared, processor will be in reset state. If the register is not setup by board configuration file, these bits reset as 0x1 (enabled), and cannot be changed unless LT_LOCK is written as 0xA05F.
0x000C	SCC_CPU1CTRL	Bits[31:2] Reserved Bit[1] Reserved Bit[0] Processor enable. If it is cleared, processor will be in reset state. If the register is not setup by board configuration file, these bits reset as 0x0 (disabled), and cannot be changed unless LT_LOCK is written as 0xA05F.
0x0010	SCC_TCM0CTRL	Bits [31:4] Reserved CPU 0 TCM control Bit[3] SLBTCMSB0 Bit[2] LOCZRAMA0 Bit[1] INITRAMB0 Bit[0] INITRAMA0 If the register is not setup by board configuration file, it resets as 0x0, and cannot be changed unless LT_LOCK is written as 0xA05F.
0x0014	SCC_TCM1CTRL	Bits[31:4] Reserved CPU 1 TCM control

		Bit[3] SLBTCMSB1 Bit[2] LOCZRAMA1 Bit[1] INITRAMB1 Bit[0] INITRAMA1 If the register is not setup by board configuration file, it resets as 0x0, and cannot be changed unless LT_LOCK is written as 0xA05F.
0x0018	SCC_SYSCTRL	Bits[31:2] Reserved Bit[17] SCC_FIQ Bit[16] SCC_IRQ Bits[15:4] Reserved Bit[3] scc_coreselect0 Bit[2] scc_coreselectsource Bit[1] TEINIT – Set to 1 for Thumb mode in exception Bit[0] ZBTRAM Read Only – set this bit 1 to disable write to ZBTRAM. If the register is not setup by board configuration file, it resets as 0x0. This register does not require unlock.
0x001C	SCC_DMACTRL	Bits[31:8] Reserved Bits[7:0] Set to 1 to mask write byte strobe signal from DMA controller. These bits are set 0x0 after reset. If the register is not setup by board configuration file, it resets as 0x0, and cannot be changed unless LT_LOCK is written as 0xA05F.
0x0020	SCC_SLSTATUS	Bits[31:10] Reserved Bit[9] STANDBYEXTDWFI. Read only Bit[8] STANDBYEXTDWFE . Read only Bits[7:4] Reserved Bit[3] WFIPIPESTOPPED1. Read only Bit[2] WFIPIPESTOPPED0. Read only Bit[1] WFEPIPESTOPPED1. Read only Bit[0] WFEPIPESTOPPED0. Read only
0x0024	SCC_SLSEQCTRL	Bits[31:4] Reserved Bit[3] SLERRACP. Bit[2] SLERRDBG. Bit[1] Emulate Lock Mode in Split. It should not be asserted while in Lock mode, or when requesting a switch to true lock mode by writing into bit[4] of SL_REQACK. Bit[0] SLSeq (Dynamic SL) Enable. Bit[0] enables dynamic switching. If this bit is zero, static Split/Lock switch (i.e. switch during CPU power-on reset) will be enabled by default. SLSeq will only be responsible for asserting SLCLAMP and SLSPLIT in static switching. SLSeq will assert a wakeup event connected to EVENTI to wakeup the core(s) only in dynamic switching.
0x0028	SCC_SLREQACK	Bits[31:4] Reserved Bit[4] Split<->Lock Switch Req Bits[3:1] Reserved Bit [0] Split->Lock Switch Completed. Read only. This register is for handshaking with Split-Lock Sequencer unit. The program should poll bit[0] for checking that the split/lock mode switch completed. The Req (bit[4]) and Ack (bit[0]) signals form a

		four-phase handshake. The SLSeq may assert a signal that causes EVENTI to wakeup the core (depending on whether SLSeq is enabled), and when this happens is not known to the program.
0x002C	SCC_PCUCTRL	Bit[15] CPU1 Power-Up Ack Bit[13] CPU1 Power-Up Request Bit[12] CPU1 shutdwn(1)/dorm(0) ctl Bit[11] CPU0 Power-Up Ack Bit[9] CPU0 Power-Up Request Bit[8] CPU0 shutdwn(1)/dorm(0) ctl Bit[0] PCU Enable
0x0040	LT_TUBE	Text output register (for retargeting in simulation only).
0x0100	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:
0x0104	SCC_LED	Bits[31:8] Reserved Bits[7:0] These bits control the LTE-3MG LEDs
0x0108	SCC_SW	Bits[31:8] Reserved Bits[7:0] These bits indicate state of the LTE-3MG user switches
0x0120	LT_LOCK	Write : write 0xA05F to this register to unlock access to a number of APB register. Only lowest 16-bit is implemented. Read : Return current value (bit [15:0])and Unlock status (bit [16]). Reset value of this register is 0x00000000. The write is only possible via APB bus.
0x0FF8	SCC_AID	SCC AID register is read only Bits[31:24] FPGA build number Bits[23:11] Reserved Bit[10] if “1” SCC_SW register have been implemented Bit[9] if “1” SCC_LED register have been implemented Bit[8] if “1” DLL lock register have been implemented Bits[7:0] number of SCC config register Currently defined as 0x1500070C
0x0FFC	SCC_ID	SCC ID register is read only Bits[31:24] Implementor ID: 0x41 = ARM Bits[23:20] IP Variant Number Bits[19:16] IP Architecture: 0x5 =AXI Bits[11:4] Primary part number: C15 = CortexR5 Bits[3:0] IP Revision nember Currently defined as 0x4105C150

Table 6.3 SCC register description

6.5 Debug components

The CoreSight debug sub system has the following memory map.

Base Address	Peripherals
0xA2000000	ROM table in CSSYS
0xA2001000	CS ETB
0xA2002000	CS CTI
0xA2003000	CS TPIU
0xA2004000	CS Trace Funnel
0xA2005000	CS ITM
0xA2006000	CS SWO
0xA2100000	ROM table in R5 Integration
0xA2110000	CPU0 Debug
0xA2111000	CPU0 PMU (not present)
0xA2112000	CPU1 Debug
0xA2113000	CPU1 PMU (not present)
0xA2118000	CS CTI for CPU0
0xA2119000	CS CTI for CPU1
0xA211C000	ETM R5 for CPU0
0xA211D000	ETM R5 for CPU1

Table 6.4 Coresight debug components base addresses.

Note : In order to test the debug memory map, the simulation test bench use air wire to set the software access enabling bit in the DAP. Otherwise, CPU access to the CoreSight system is blocked.

7 INTERRUPTS AND EVENTS SIGNALS

7.1 Interrupts

The system contains a GIC (Generic Interrupt Controller – PL390). The controller is generated using the following parameters:

```
SGI = 2
PPI = 16
SPI = 128
```

The interrupt signal assignments on the interrupt input in R5 integration level are:

Signal	Descriptions
SPI	Bit[127:68] : Tied low Bit[67] : COMMTX1 Bit[66] : COMMTX0 Bit[65] : COMMRX1 Bit[64] : COMMRX0 Bit[63] : PMUIRQ[3] (tied low) Bit[62] : PMUIRQ[2] (tied low) Bit[61] : PMUIRQ[1] (inverted of nPMUIRQ[1]) Bit[60] : PMUIRQ[0] (inverted of nPMUIRQ[0]) Bit[59] : Tied low (PCIe not implemented) Bit[58] : Tied low (PCIe not implemented) Bit[57] : Tied low (PCIe not implemented) Bit[56] : Tied low (PCIe not implemented) Bit[55] : DMAC_IRQ[1] Bit[54] : DMAC_IRQ[0] Bit[53] : Tied low. TrustZone not implemented Bit[52] : DMAC_IRQ_ABORT Bit[51] : Tied low. Watchdog not implemented Bit[50] : Tied low. GPIO not implemented Bit[49] : Tied low. Timer1 not implemented Bit[48] : Tied low. Timer0 not implemented Bit[47] : Tied low. Reserved Bit[46] : SMC1_INT Bit[45] : SMC0_INT Bit[44] : CLCD interrupt Bit[43] : L2 combine interrupt Bit[42:0] : SB_IRQ[42:0]
PPI 0	32 bit, all tied to 0. (Only lowest 16 bit are available)
PPI 1	32 bit, all tied to 0. (Only lowest 16 bit are available)
nIRQ0	nIRQ[0] = SB_nCPUIRQ & ~SCC_IRQ & nVALIRQ[0] & nVALIRQ[1]
nIRQ1	nIRQ[1] = SB_nCPUIRQ & ~SCC_IRQ & nVALIRQ[0] & nVALIRQ[1]
nFIQ0	nFIQ[0] = SB_nCPUFIQ & ~SCC_FIQ & nVALFIQ[0] & nVALFIQ[1]
nFIQ1	nFIQ[1] = SB_nCPUFIQ & ~SCC_FIQ & nVALFIQ[0] & nVALFIQ[1]

Table 7.1 Interrupt map.

7.2 Event signals

Event signal connections are as follows. A close coupling between the two processors is required in the FPGA wrapper logic as there is no event cross coupling inside R5.

Connection	Assignment
Event In for CPU0	EVENTI[0]= SB_EVENT | EVENTO[1] | slwakeup
Event In for CPU1	EVENTI[1] = SB_EVENT | EVENTO[0] | slwakeup
Event Output	SB_EVENTO = EVENTO[0] | EVENTO[1]

Table 7.2 Event signals