Application Note 305

Example LogicTile Express 13MG design for a CoreTile Express A15x2 or a CoreTile Express A15x2_A7x3.

Document number: ARM DAI 0305

Issued: July 2012

ARM

Application Note 305
Example LogicTile Express 13MG design for a CoreTile Express A15x2 or CoreTile Express A15x2_A7x3

Release information

The following changes have been made to this Application Note.

Change history

Date	Issue	Change
January 17, 2012	A	First release

Proprietary notice

Confidentiality status

This document is Open Access. This document has no restriction on distribution.

Feedback on this Application Note

If you have any comments on this Application Note, please send email to errata@arm.com giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- an explanation of your comments.

General suggestions for additions and improvements are also welcome.

ARM web address

http://www.arm.com

Table of Contents

1 Introduction

1.1 Purpose of this application note

This application note discusses the operation of the example design for a LogicTile Express 13MG (V2F-2XV6) with a CoreTile Express A15x2. It will examine the contents of the V2F-2XV6, the system interconnect, the clock structure, and specifics of the programmer's model directly relevant to V2F-2XV6 operation.

On reading this Application Note the user should be in a position to debug and analyze the operation of the provided images. He also should be able to make changes to the provided V2F-2XV6 design, connect their own AXI or AHB based masters and AXI, AHB or APB slaves

1.2 Overview of hardware platform

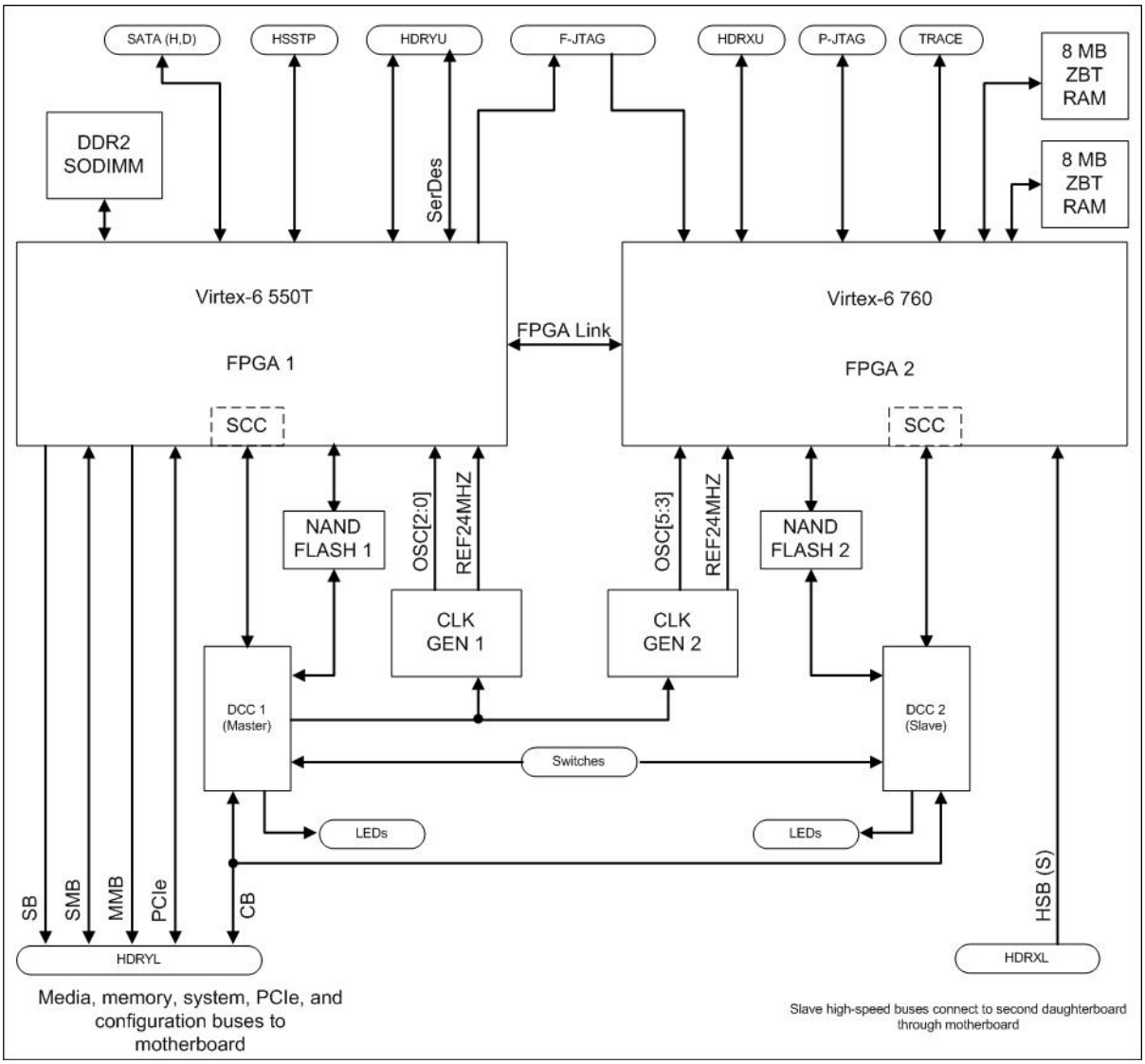

Figure 1-1 LogicTile 13MG (V2F-2XV6) daughter board

This application note is designed to work on a Motherboard Express uATX (V2M-P1) fitted with CoreTile Express A15x2 (V2P-CA15) in Site 1 and LogicTile Express 13MG (V2F-2XV6) fitted in Site 2, as shown in **Figure 1-1 LogicTile 13MG (V2F-2XV6)**.

2 Getting Started

Before using application note for the first time, you need to connect a V2P-CA15 and a V2F-2XV6 daughterboard onto the V2M-P1 motherboard. Copy configuration files to the V2M-P1 motherboard Micro SD Card. Follow these steps to configure and program the FPGA image.

1. Plug the V2P-CA15 daughterboard onto site 1 and the V2F-2XV6 daughterboard onto site 2 of the V2M-P1 Motherboard as described in **Quick Start Guide for the Versatile Express Family - Adding Daughterboards**.

2. Connect the supplied ATX-PCIe power cable to the V2F-2XV6 daughterboard.

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. After the board has been connected via USB and the PC recognizes the motherboard as a USB Flash Drive, copy the application note HBI0217B/AN305 directory to the /SITE2/HBI0217B directory on V2M-P1 USB Flash disk.

5. Power cycle the boards using the red Power button to upload application image to V2F-2XV6 FPGAs.

6. The system will now be fully configured and ready for use.

3 System architecture

This system is an AXI (AMBA 3.0) based system. The example V2F-2XV6 image exposes one muxed 64 bit AXI slave port EMM to the V2P-CA15 daughterboard and Static Memory Bus, Multimedia Bus, System Bus to the V2M-P1 motherboard.

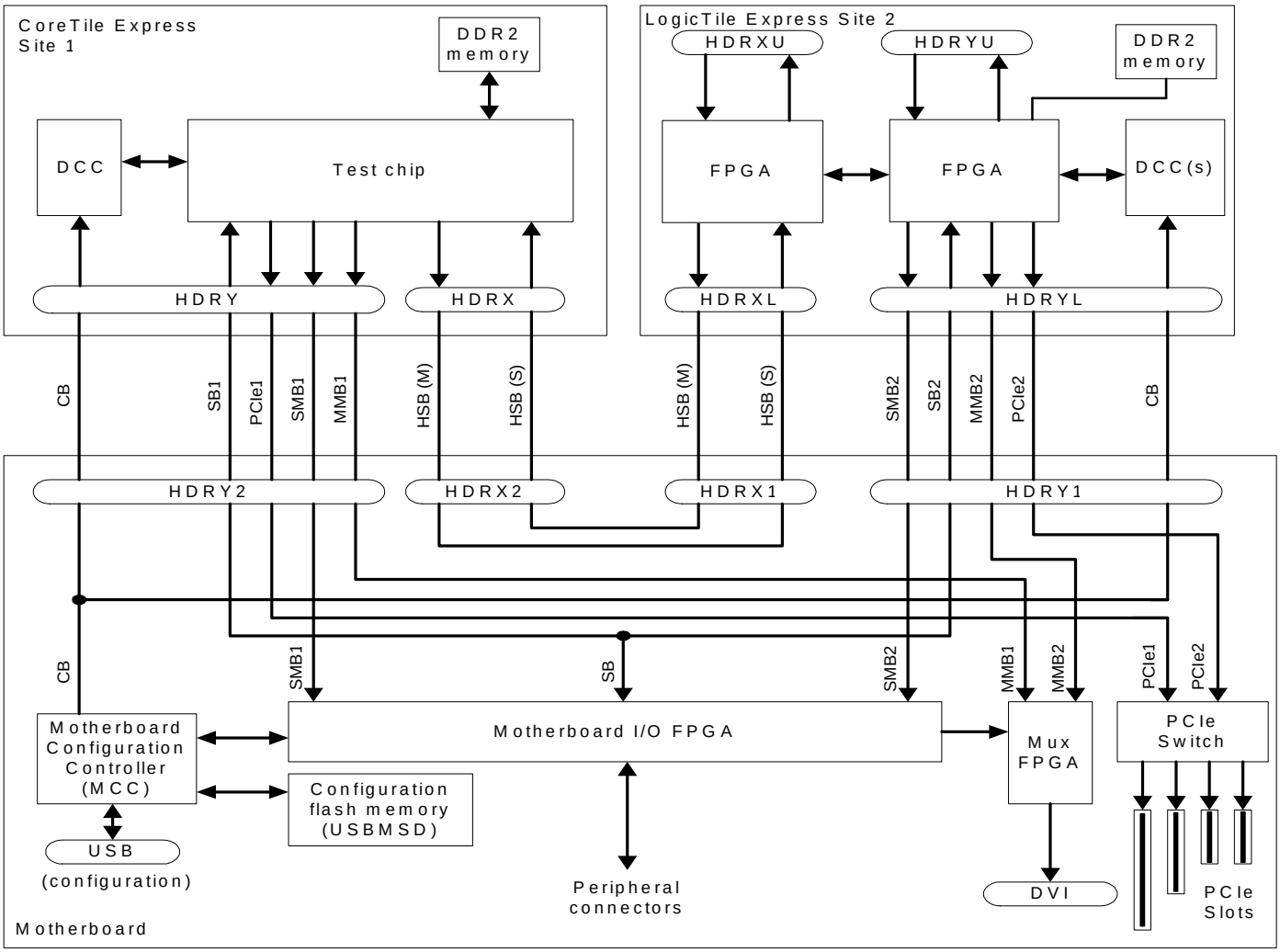

Figure 3-1 Block level architecture

Note that the direction of the arrows indicates the direction of control, i.e. it points from the Master to the Slave. An AXI buses contains signals going in both directions.

3.1 AN305 architecture

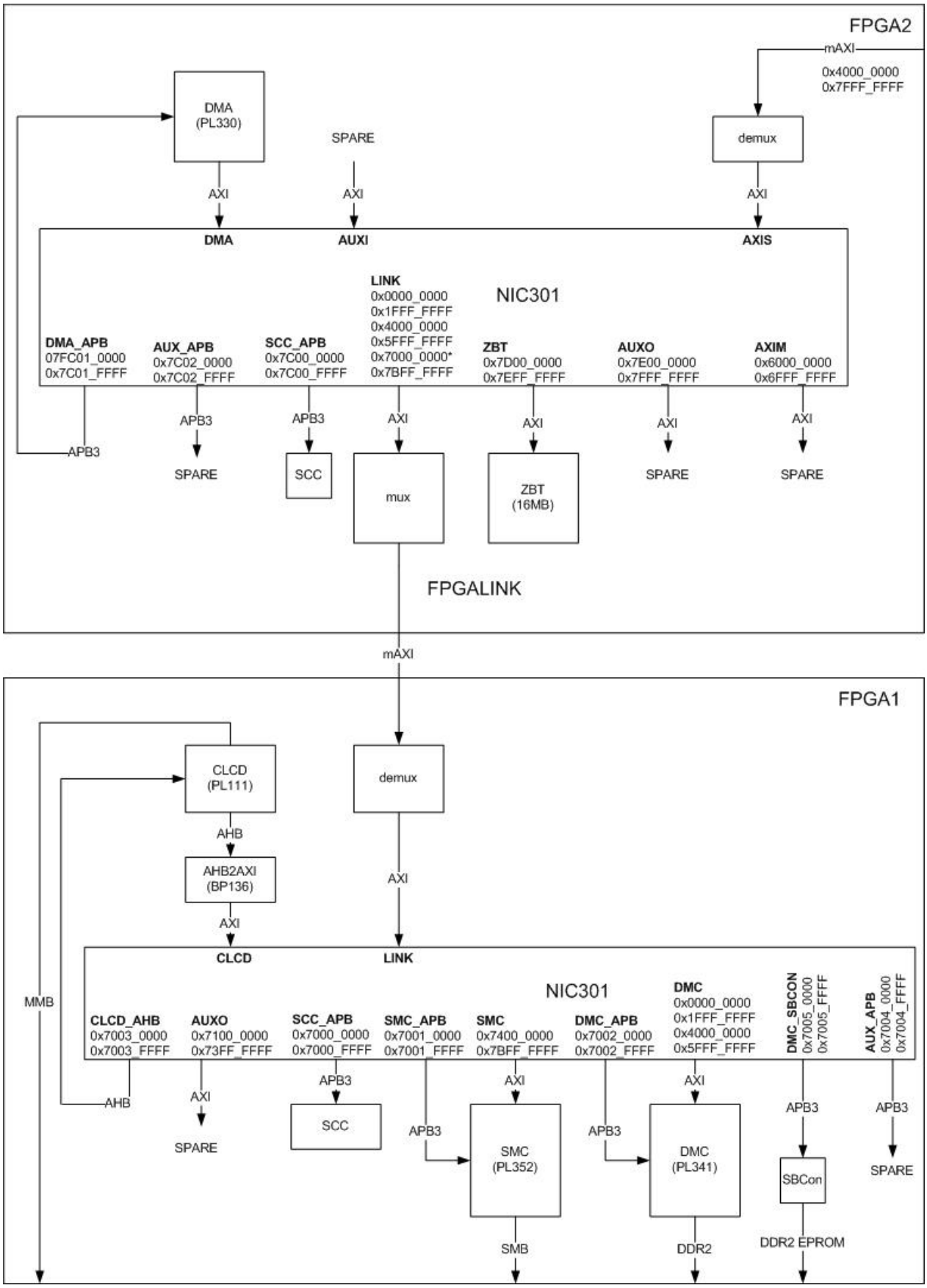

Figure 3-2 AN305 Architecture

3.2 Module functionality

This section only describes custom blocks which are not released as ARM PrimeCells. Technical Reference Manuals (TRM) for other modules are supplied on the Versatile Express DVD.

3.2.1 Demux

The Demux block de-multiplexes the AXI using a 2:1 ratio. This block should only be used with communicating with CoreTile Express A15x2 (V2P_CA15). Information on the multiplexing scheme can be found in section 4.6. These blocks also incorporate an asynchronous bridge.

3.2.2 SCC

The SCC (Serial Configuration Controller) block communicates with the DCC (Daughter board Configuration Controller) connected to that FPGA. It is used to pass system information and control the use of switches and LEDs. Refer to section 5.2

3.2.3 SBCon

The SBCon block is a serial interface which can be used to interrogate the contents of the SODIMM EEPROM. Refer to section 5.3

3.3 Modules revision

Module	Revision
NIC301 Bus Matrix	r2p0
PL111 CLCD	r0p2
Demux (DxAsynchAxi_V2)	r0p0
PL330 DMAC	r0p0
PL341 DMC	r0p1-00rel1
PL352 SMC	r2p1
SCC	r0p0
SBCon	r0p0

Table 1 Modules revision in AN305 r0p0

3.4 Sparse Matrixes

This Application note uses sparse matrixes, meaning that some master's ports **cannot** access all slave ports. The following figures are snapshots from AMBA Designer, which show which masters connect to slave ports.

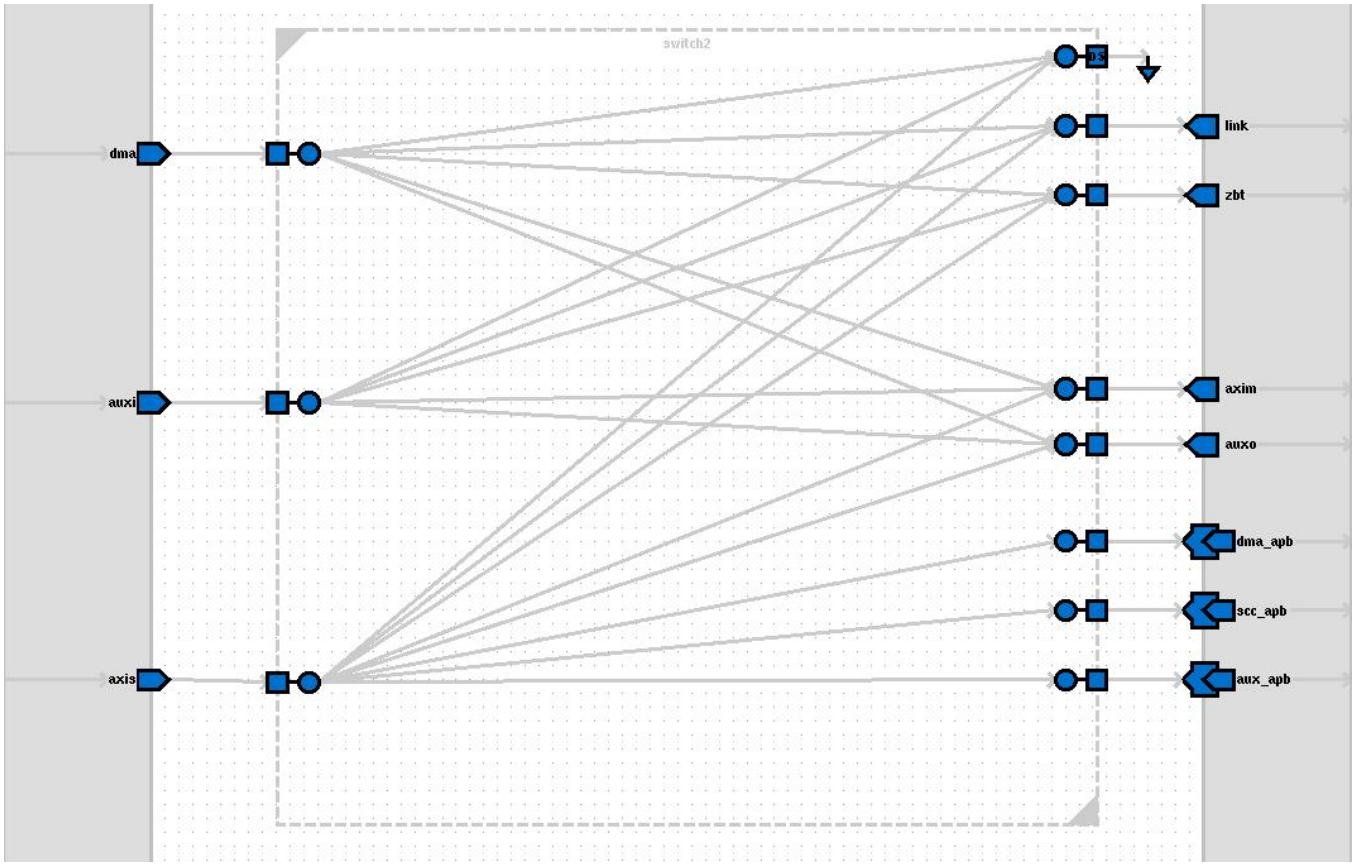

Figure 3 FPGA2 Matrix

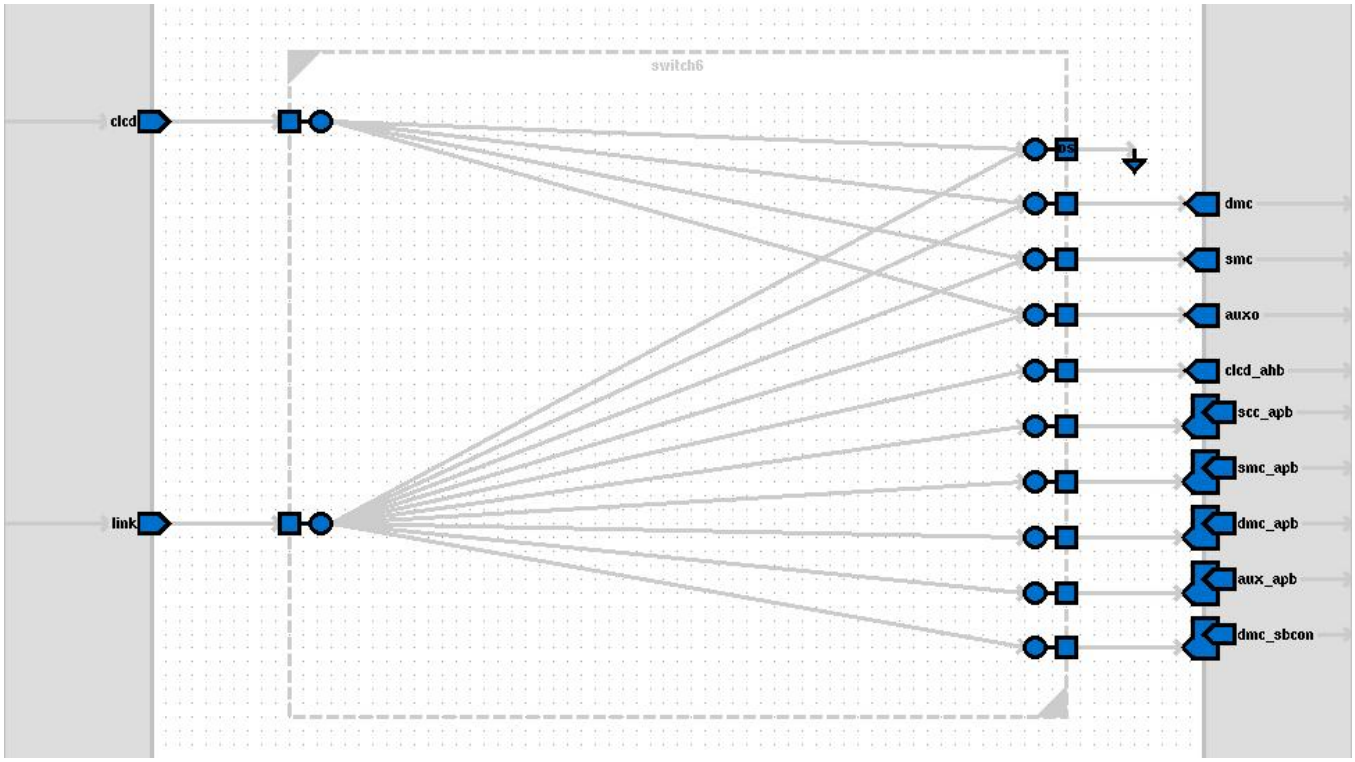

Figure 4 FPGA1 Matrix

3.5 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 V2F-2XV6 have been used to achieve that.

For the maximum and default frequencies of clocks please refer to **8.1 Default, minimum and maximum operating frequencies**.

For information how to set up and change OSC0-OSC5 clocks frequency on please refer to **Motherboard Express uATX TRM**. This application note only discusses the clocks used by this application note and not all possible use cases. All clocks are asynchronous to each other except for a few cases (frequencies in brackets denote frequency changes because of MCMM's).

FPGA2 ACLK is twice the frequency of ACLK_LINK
FPGA1 MCK, MCLK2x and MCLK_90 are synchronous after the MCMM.
SMB_CLKO must be greater than 33MHz for V2M_P1 PLL to lock.

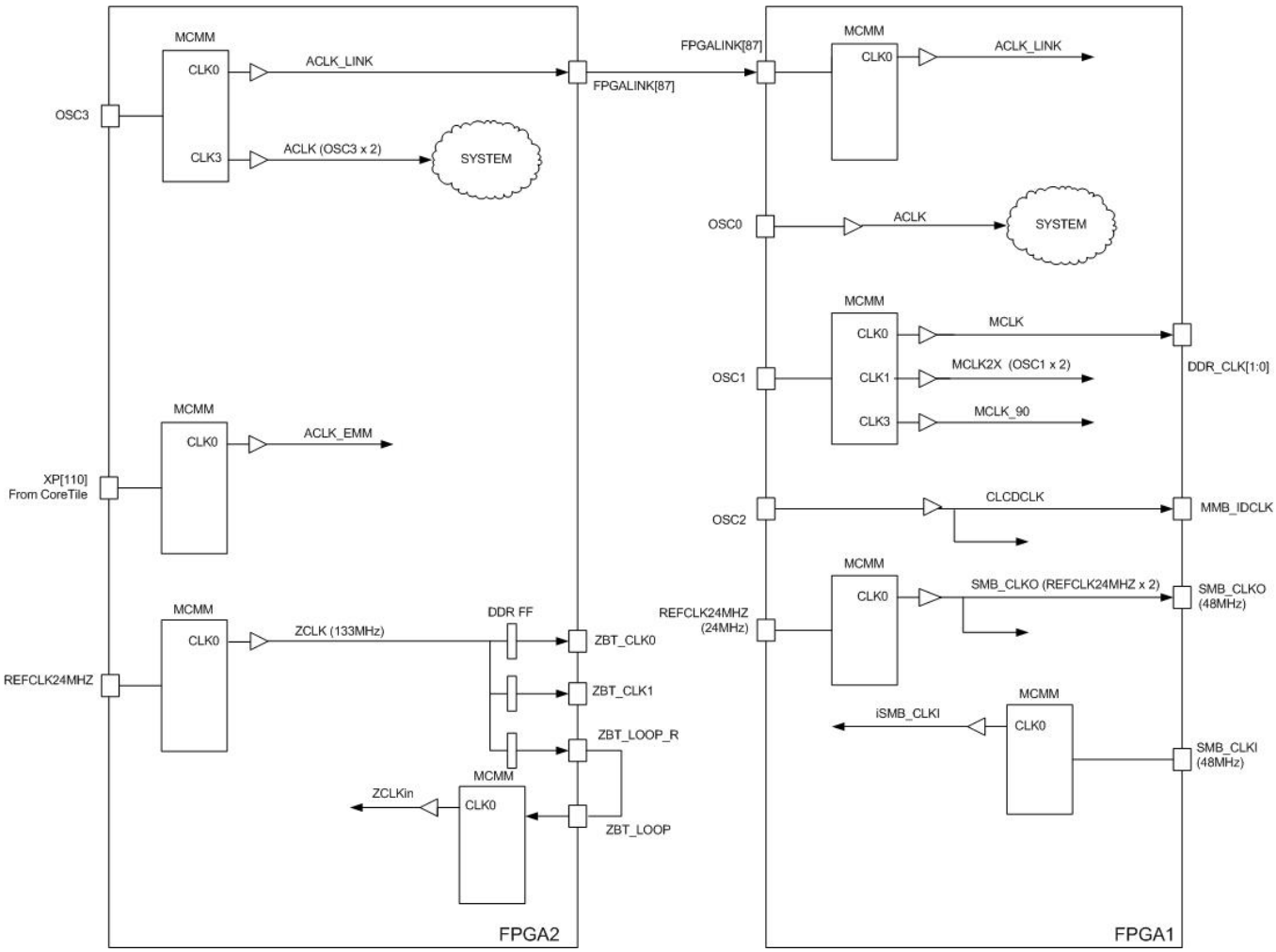

Figure 5 FPGA clock domains

3.6 Interrupt architecture

There are 4 interrupts sources generated by AN305 which are connected to V2P-CA15 daughterboard interrupt controller via the V2M-P1 motherboard. The interrupts signals have been inverted in AN305 top level and V2M-P1 motherboard to provide the correct polarity for the interrupt controller in the Cortex-A15 test chip on V2P-CA15.

Signal	Source	V2M-P1 int	V2P-CA15 int
Clcdintr	CLCD	SB_INT[0]	INTSOURCE[22]
smc_int	SMC	SB_INT[1]	INTSOURCE[23]
dma_int[0]	DMAC	SB_INT[2]	INTSOURCE[24]
dma_int[1]	DMAC	SB_INT[3]	INTSOURCE[25]

Table 2 Interrupts

3.7 Reset architecture

The reset signal CB_nRST from the motherboard is synchronized to the correct clock domains to reset all the peripherals in the V2 FPGA Daughterboard.

The reset signal nPLLRESET from the DCC controller is used to reset DCMs during power-up sequence.

The reset signal CB_nPOR is not used in design.

4 Hardware description

4.1 V2F-2XV6 wrappers

This level has been split into 2, once for each FPGA (v2f_760_wrapper.v and v2f_550t_wrapper.v). They define the mapping from the FPGAs to physical connectors and other devices. The wrapper also contains clock generation.

4.2 AN305 top level files

These files (v2f_760_toplevel.v and v2f_550t_toplevel.v) connect all the Application Note components together. This includes all the major modules as shown in **Figure 3-2 AN305 Architecture**. This level of RTL is provided so modules can be added and removed.

High value blocks are only provided as .NGO netlists.

Constrain files (ucf) are included, 2 for the application note (1 per FPGA) and 2 which shows all the possible FPGA pin connections which could be used on the V2F-2XV6 daughterboard.

4.3 ZBT SRAM memory

The two 8Mb 32 bit ZBT SRAM chips are used by AxiAsynch2ZBT controller as one 16Mb 64 bit wide memory.

4.4 AXI multiplexing scheme

DDR registers have been used to implement 2:1 multiplexing and demultiplexing to reduce the pin count for the two AXI buses for implementation on the HDRXL header.

The registers are part of the AXI asynchronous bridge logic therefore multiplexing does not introduce any extra latency to the AXI bus.

The xVALID and xREADY signals on AXI are not multiplexed in this way due to their timing requirements and are passed directly between devices.

For multiplexed AXI timing requirements please refer to **8.2 Demux AXI** timing requirements.

The multiplexing and demultiplexing blocks are provided as NGO netlist.

4.5 DDR2 memory

This application note requires a 4GB SODIMM (MICRON MT16HTS51265HY-667A1) to be present in the SODIMM connector. Only a subset of the full memory capacity of this SODIMM is used in this application note. Please refer to section 5.1 for the application note memory map.

4.6 Header HDRXL pin allocation

One AXI bus connects to HDRXL as shown. Multiplexed AXI slave bus EMM connects via HDRXL header to AXI External Master

4.6.1 Multiplexed AXI slave bus

HDRXL pin	X bus	Signal (Hi/Lo)	HDRX pin	X bus	Signal (Hi/Lo)
D50	XN153	EMM_WDATA0/32	E48	XP154	EMM_WSTRB3/7
D49	XP153	EMM_WDATA1/33	D26	XS13	EMM_WVALID
C49	XN152	EMM_WDATA2/34	C26	XS12	EMM_WREADY
C48	XP152	EMM_WDATA3/35	K41	XN129	EMM_AWADDR0/16
B50	XN151	EMM_WDATA4/36	K40	XP129	EMM_AWADDR1/17
B49	XP151	EMM_WDATA5/37	J40	XN128	EMM_AWADDR2/18
A49	XN150	EMM_WDATA6/38	J39	XP128	EMM_AWADDR3/19
A48	XP150	EMM_WDATA7/39	H41	XN127	EMM_AWADDR4/20
K47	XN149	EMM_WDATA8/40	H40	XP127	EMM_AWADDR5/21
K46	XP149	EMM_WDATA9/41	G40	XN126	EMM_AWADDR6/22
J46	XN148	EMM_WDATA10/42	G39	XP126	EMM_AWADDR7/23
J45	XP148	EMM_WDATA11/43	F41	XN125	EMM_AWADDR8/24
H47	XN147	EMM_WDATA12/44	F40	XP125	EMM_AWADDR9/25
H46	XP147	EMM_WDATA13/45	E40	XN124	EMM_AWADDR10/26
G46	XN146	EMM_WDATA14/46	E39	XP124	EMM_AWADDR11/27
G45	XP146	EMM_WDATA15/47	D41	XN123	EMM_AWADDR12/28
F47	XN145	EMM_WDATA16/48	D40	XP123	EMM_AWADDR13/29
F46	XP145	EMM_WDATA17/49	C40	XN122	EMM_AWADDR14/30
E46	XN144	EMM_WDATA18/50	C39	XP122	EMM_AWADDR15/31
E45	XP144	EMM_WDATA19/51	D44	XN133	EMM_AWID1/0
D47	XN143	EMM_WDATA20/52	D43	XP133	EMM_AWID3/2
D46	XP143	EMM_WDATA21/53	C43	XN132	EMM_AWID5/4
C46	XN142	EMM_WDATA22/54	C42	XP132	EMM_AWID7/6
C45	XP142	EMM_WDATA23/55	B44	XN131	EMM_AWID9/8
B47	XN141	EMM_WDATA24/56	B43	XP131	EMM_AWID11/10
B46	XP141	EMM_WDATA25/57	A43	XN130	EMM_AWID13/12
A46	XN140	EMM_WDATA26/58	H43	XP137	EMM_AWPROT2/AWID14
A45	XP140	EMM_WDATA27/59	E43	XN134	EMM_AWLEN0/2
K44	XN139	EMM_WDATA28/60	E42	XP134	EMM_AWLEN1/3
K43	XP139	EMM_WDATA29/61	G42	XP136	EMM_AWSIZE0/1
J43	XN138	EMM_WDATA30/62	H44	XN137	EMM_AWPROT0/1
J42	XP138	EMM_WDATA31/63	G43	XN136	EMM_AWBURST0/1
K50	XN159	EMM_WID1/0	A42	XP130	EMM_AWLOCK0/1
K49	XP159	EMM_WID3/2	F44	XN135	EMM_AWCACHE0/2
J49	XN158	EMM_WID5/4	F43	XP135	EMM_AWCACHE1/3
J48	XP158	EMM_WID7/6	F26	XS15	EMM_AWVALID
H50	XN157	EMM_WID9/8	E26	XS14	EMM_AWREADY
H49	XP157	EMM_WID11/10	A31	XN90	EMM_BID0/1
G49	XN156	EMM_WID13/12	A30	XP90	EMM_BID2/3
G48	XP156	EMM_WLAST/WID14	C28	XN82	EMM_BID4/5
F50	XN155	EMM_WSTRB0/4	C27	XP82	EMM_BID6/7
F49	XP155	EMM_WSTRB1/5	B29	XN81	EMM_BID8/9
E49	XN154	EMM_WSTRB2/6	B28	XP81	EMM_BID10/11

HDRXL pin	X bus	Signal (Hi/Lo)	HDRXL pin	X bus	Signal (Hi/Lo)
A28	XN80	EMM_BID12/13	F32	XN95	EMM_RDATA8/40
A27	XP80	EMM_BID14/1'b0	F31	XP95	EMM_RDATA9/41
A33	XP100	EMM_BRESP0/1	E31	XN94	EMM_RDATA10/42
A26	XS10	EMM_BVALID	E30	XP94	EMM_RDATA11/43
B26	XS11	EMM_BREADY	D32	XN93	EMM_RDATA12/44
D38	XN113	EMM_ARADDR0/16	D31	XP93	EMM_RDATA13/45
D37	XP113	EMM_ARADDR1/17	C31	XN92	EMM_RDATA14/46
C37	XN112	EMM_ARADDR2/18	C30	XP92	EMM_RDATA15/47
C36	XP112	EMM_ARADDR3/19	B32	XN91	EMM_RDATA16/48
B38	XN111	EMM_ARADDR4/20	B31	XP91	EMM_RDATA17/49
B37	XP111	EMM_ARADDR5/21	K29	XN89	EMM_RDATA18/50
K35	XN109	EMM_ARADDR6/22	K28	XP89	EMM_RDATA19/51
K34	XP109	EMM_ARADDR7/23	J28	XN88	EMM_RDATA20/52
J34	XN108	EMM_ARADDR8/24	J27	XP88	EMM_RDATA21/53
J33	XP108	EMM_ARADDR9/25	H29	XN87	EMM_RDATA22/54
H35	XN107	EMM_ARADDR10/26	H28	XP87	EMM_RDATA23/55
H34	XP107	EMM_ARADDR11/27	G28	XN86	EMM_RDATA24/56
G34	XN106	EMM_ARADDR12/28	G27	XP86	EMM_RDATA25/57
G33	XP106	EMM_ARADDR13/29	F29	XN85	EMM_RDATA26/58
F35	XN105	EMM_ARADDR14/30	F28	XP85	EMM_RDATA27/59
F34	XP105	EMM_ARADDR15/31	E28	XN84	EMM_RDATA28/60
H38	XN117	EMM_ARID1/0	E27	XP84	EMM_RDATA29/61
H37	XP117	EMM_ARID3/2	D29	XN83	EMM_RDATA30/62
G37	XN116	EMM_ARID5/4	D28	XP83	EMM_RDATA31/63
G36	XP116	EMM_ARID7/6	D35	XN103	EMM_RID0/1
F38	XN115	EMM_ARID9/8	D34	XP103	EMM_RID2/3
F37	XP115	EMM_ARID11/10	C34	XN102	EMM_RID4/5
E37	XN114	EMM_ARID13/12	C33	XP102	EMM_RID6/7
B40	XP121	EMM_ARPROT2/ARID14	B35	XN101	EMM_RID8/9
J37	XN118	EMM_ARLEN0/2	B34	XP101	EMM_RID10/11
J36	XP118	EMM_ARLEN1/3	A34	XN100	EMM_RID12/13
A39	XP120	EMM_ARSIZE0/1	E33	XP104	EMM_RID14/RLAST
B41	XN121	EMM_ARPROT0/1	E34	XN104	EMM_RRESP1/0
A40	XN120	EMM_ARBURST0/1	J26	XS18	EMM_RVALID
E36	XP114	EMM_ARLOCK0/1	K26	XS19	EMM_RREADY
K38	XN119	EMM_ARCACHE0/2	A37	XN110	NC
K37	XP119	EMM_ARCACHE1/3	A36	XP110	EMM_ACLK
G26	XS16	EMM_ARVALID			
H26	XS17	EMM_ARREADY			
K32	XN99	EMM_RDATA0/32			
K31	XP99	EMM_RDATA1/33			
J31	XN98	EMM_RDATA2/34			
J30	XP98	EMM_RDATA3/35			
H32	XN97	EMM_RDATA4/36			
H31	XP97	EMM_RDATA5/37			
G31	XN96	EMM_RDATA6/38			
G30	XP96	EMM_RDATA7/39			

Table 3 Header HRDXL EMM bus pin allocation

4.7 Header HDRYL pin allocation

4.7.1 Multimedia bus

HDRYL pin	MMB bus	CLCD Signal	HDRYL pin	MMB bus	CLCD Signal
A46	MMB_DATA0	CLD0	J45	MMB_DATA18	CLD18
B46	MMB_DATA1	CLD1	K45	MMB_DATA19	CLD19
C46	MMB_DATA2	CLD2	A43	MMB_DATA20	CLD20
D46	MMB_DATA3	CLD3	B43	MMB_DATA21	CLD21
E46	MMB_DATA4	CLD4	C43	MMB_DATA22	CLD22
F46	MMB_DATA5	CLD5	D43	MMB_DATA23	CLD23
G46	MMB_DATA6	CLD6	G43	MMB_DE	CLAC
H46	MMB_DATA7	CLD7	E43	MMB_HS	CLLP
J46	MMB_DATA8	CLD8	K43	MMB_IDCLK	CLCP
K46	MMB_DATA9	CLD9	F43	MMB_VS	CLFP
A45	MMB_DATA10	CLD10			
B45	MMB_DATA11	CLD11			
C45	MMB_DATA12	CLD12			
D45	MMB_DATA13	CLD13			
E45	MMB_DATA14	CLD14			
F45	MMB_DATA15	CLD15			
G45	MMB_DATA16	CLD16			
H45	MMB_DATA17	CLD17			

Table 4 Multimedia bus pin allocation

4.7.2 Static memory bus

HDRYL pin	SMB bus	SMC Signal	HDRYL pin	SMB bus	SMC Signal
C26	SMB_ADDR0	add_0	B29	SMB_DATA11	data_11
D26	SMB_ADDR1	add_1	C29	SMB_DATA12	data_12
E26	SMB_ADDR2	add_2	D29	SMB_DATA13	data_13
F26	SMB_ADDR3	add_3	E29	SMB_DATA14	data_14
G26	SMB_ADDR4	add_4	F29	SMB_DATA15	data_15
H26	SMB_ADDR5	add_5	G29	SMB_DATA16	data_16
J26	SMB_ADDR6	add_6	H29	SMB_DATA17	data_17
K26	SMB_ADDR7	add_7	J29	SMB_DATA18	data_18
A24	SMB_ADDR8	add_8	K29	SMB_DATA19	data_19
B24	SMB_ADDR9	add_9	A27	SMB_DATA20	data_20
C24	SMB_ADDR10	add_10	B27	SMB_DATA21	data_21
D24	SMB_ADDR11	add_11	C27	SMB_DATA22	data_22
E24	SMB_ADDR12	add_12	D27	SMB_DATA23	data_23
F24	SMB_ADDR13	add_13	E27	SMB_DATA24	data_24
G24	SMB_ADDR14	add_14	F27	SMB_DATA25	data_25
H24	SMB_ADDR15	add_15	G27	SMB_DATA26	data_26
J24	SMB_ADDR16	add_16	H27	SMB_DATA27	data_27
K24	SMB_ADDR17	add_17	J27	SMB_DATA28	data_28
A23	SMB_ADDR18	add_18	K27	SMB_DATA29	data_29

SMB bus	SMB bus	SMB bus	SMB bus	SMB bus	SMB bus
B23	SMB_ADDR19	add_19	A26	SMB_DATA30	data_30
C23	SMB_ADDR20	add_20	B26	SMB_DATA31	data_31
D23	SMB_ADDR21	add_21	F20	SMB_nADV	adv_n
E23	SMB_ADDR22	add_22	G20	SMB_nBAA	baa_n
F23	SMB_ADDR23	add_23	K23	SMB_nBLS0	bls_n_0
G23	SMB_ADDR24	add_24	A21	SMB_nBLS1	bls_n_1
H23	SMB_ADDR25	add_25	B21	SMB_nBLS2	bls_n_2
J23	SMB_ADDR26	add_26	C21	SMB_nBLS3	bls_n_3
A19	SMB_ALEN	1'b0	C19	SMB_nCEN	1'b1
B19	SMB_CLEN	1'b0	E21	SMB_nCS0	1'b1
K21	SMB_CLKI	fbclk_in_0	F21	SMB_nCS1	1'b1
K19	SMB_CLKO	SMCLK	G21	SMB_nCS2	1'b1
H20	SMB_CRE	cre	H21	SMB_nCS3	cs_n_1
A30	SMB_DATA0	data_0	J21	SMB_nCS4	1'b1
B30	SMB_DATA1	data_1	A20	SMB_nCS5	1'b1
C30	SMB_DATA2	data_2	B20	SMB_nCS6	1'b1
D30	SMB_DATA3	data_3	C20	SMB_nCS7	cs_n_0
E30	SMB_DATA4	data_4	D20	SMB_nOE	oe_n
F30	SMB_DATA5	data_5	D19	SMB_nREN	1'b1
G30	SMB_DATA6	data_6	F19	SMB_nREQ	1'b0
H30	SMB_DATA7	data_7	E20	SMB_nWAIT	Wait
J30	SMB_DATA8	data_8	D21	SMB_nWE	we_n
K30	SMB_DATA9	data_9	E19	SMB_nWEN	1'b1
A29	SMB_DATA10	data_10			

Table 5 Static Memory bus pin allocation

4.7.3 Interrupts

HDRYL pin	SB bus	Signal
J11	SB_INT0	clcdint
K11	SB_INT1	smc_int
A10	SB_INT2	dma_int_0
B10	SB_INT3	dma_int_1

Table 6 Interrupts outputs pin allocation

4.8 FPGALINK signal assignment

The FPGALINK bus is used to communicate between the 2 FPGAs on the board, the following table shows the usage of this bus for this application note.

FPGALINK BUS	Signal (Hi/Lo)	FPGALINK BUS	Signal (Hi/Lo)
FPGALINK_P87	reserved	FPGALINK_N87	reserved
FPGALINK_P86	reserved	FPGALINK_N86	reserved
FPGALINK_P85	WDATA0/32	FPGALINK_N85	BID9/8
FPGALINK_P84	WDATA1/33	FPGALINK_N84	BID11/10
FPGALINK_P83	WDATA2/34	FPGALINK_N83	BID13/12
FPGALINK_P82	WDATA3/35	FPGALINK_N82	b0/BID14
FPGALINK_P81	WDATA4/36	FPGALINK_N81	BRESP0/1
FPGALINK_P80	WDATA5/37	FPGALINK_N80	BVALID
FPGALINK_P79	WDATA6/38	FPGALINK_N79	BREADY
FPGALINK_P78	WDATA7/39	FPGALINK_N78	ARADDR0/16
FPGALINK_P77	WDATA8/40	FPGALINK_N77	ARADDR1/17
FPGALINK_P76	WDATA9/41	FPGALINK_N76	ARADDR2/18
FPGALINK_P75	WDATA10/42	FPGALINK_N75	ARADDR3/19
FPGALINK_P74	WDATA11/43	FPGALINK_N74	ARADDR4/20
FPGALINK_P73	WDATA12/44	FPGALINK_N73	ARADDR5/21
FPGALINK_P72	WDATA13/45	FPGALINK_N72	ARADDR6/22
FPGALINK_P71	WDATA14/46	FPGALINK_N71	ARADDR7/23
FPGALINK_P70	WDATA15/47	FPGALINK_N70	ARADDR8/24
FPGALINK_P69	WDATA16/48	FPGALINK_N69	ARADDR9/25
FPGALINK_P68	WDATA17/49	FPGALINK_N68	ARADDR10/26
FPGALINK_P67	WDATA18/50	FPGALINK_N67	ARADDR11/27
FPGALINK_P66	WDATA19/51	FPGALINK_N66	ARADDR12/28
FPGALINK_P65	WDATA20/52	FPGALINK_N65	ARADDR13/29
FPGALINK_P64	WDATA21/53	FPGALINK_N64	ARADDR14/30
FPGALINK_P63	WDATA22/54	FPGALINK_N63	ARADDR15/31
FPGALINK_P62	WDATA23/55	FPGALINK_N62	ARID1/0
FPGALINK_P61	WDATA24/56	FPGALINK_N61	ARID3/2
FPGALINK_P60	WDATA25/57	FPGALINK_N60	ARID5/4
FPGALINK_P59	WDATA26/58	FPGALINK_N59	ARID7/6
FPGALINK_P58	WDATA27/59	FPGALINK_N58	ARID9/8
FPGALINK_P57	WDATA28/60	FPGALINK_N57	ARID11/10
FPGALINK_P56	WDATA29/61	FPGALINK_N56	ARID13/12
FPGALINK_P55	WDATA30/62	FPGALINK_N55	ARPROT2/ARID14
FPGALINK_P54	WDATA31/63	FPGALINK_N54	ARLEN0/2
FPGALINK_P53	WID1/0	FPGALINK_N53	ARLEN1/3
FPGALINK_P52	WID3/2	FPGALINK_N52	ARSIZE0/1
FPGALINK_P51	WID5/4	FPGALINK_N51	ARPROT0/1
FPGALINK_P50	WID7/6	FPGALINK_N50	ARBURST0/1

FPGALINK BUS	Signal (Hi/Lo)	FPGALINK BUS	Signal (Hi/Lo)
FPGALINK_P49	WID9/8	FPGALINK_N49	ARLOCK0/1
FPGALINK_P48	WID11/10	FPGALINK_N48	ARCACHE0/2
FPGALINK_P47	WID13/12	FPGALINK_N47	ARCACHE1/3
FPGALINK_P46	WLAST/WID14	FPGALINK_N46	ARVALID
FPGALINK_P45	WSTRB0/4	FPGALINK_N45	ARREADY
FPGALINK_P44	WSTRB1/5	FPGALINK_N44	RDATA0/32
FPGALINK_P43	WSTRB2/6	FPGALINK_N43	RDATA1/33
FPGALINK_P42	WSTRB3/7	FPGALINK_N42	RDATA2/34
FPGALINK_P41	WVALID	FPGALINK_N41	RDATA3/35
FPGALINK_P40	WREADY	FPGALINK_N40	RDATA4/36
FPGALINK_P39	AWADDR0/16	FPGALINK_N39	RDATA5/37
FPGALINK_P38	AWADDR1/17	FPGALINK_N38	RDATA6/38
FPGALINK_P37	AWADDR2/18	FPGALINK_N37	RDATA7/39
FPGALINK_P36	AWADDR3/19	FPGALINK_N36	RDATA8/40
FPGALINK_P35	AWADDR4/20	FPGALINK_N35	RDATA9/41
FPGALINK_P34	AWADDR5/21	FPGALINK_N34	RDATA10/42
FPGALINK_P33	AWADDR6/22	FPGALINK_N33	RDATA11/43
FPGALINK_P32	AWADDR7/23	FPGALINK_N32	RDATA12/44
FPGALINK_P31	AWADDR8/24	FPGALINK_N31	RDATA13/45
FPGALINK_P30	AWADDR9/25	FPGALINK_N30	RDATA14/46
FPGALINK_P29	AWADDR10/26	FPGALINK_N29	RDATA15/47
FPGALINK_P28	AWADDR11/27	FPGALINK_N28	RDATA16/48
FPGALINK_P27	AWADDR12/28	FPGALINK_N27	RDATA17/49
FPGALINK_P26	AWADDR13/29	FPGALINK_N26	RDATA18/50
FPGALINK_P25	AWADDR14/30	FPGALINK_N25	RDATA19/51
FPGALINK_P24	AWADDR15/31	FPGALINK_N24	RDATA20/52
FPGALINK_P23	AWID1/0	FPGALINK_N23	RDATA21/53
FPGALINK_P22	AWID3/2	FPGALINK_N22	RDATA22/54
FPGALINK_P21	AWID5/4	FPGALINK_N21	RDATA23/55
FPGALINK_P20	AWID7/6	FPGALINK_N20	RDATA24/56
FPGALINK_P19	AWID9/8	FPGALINK_N19	RDATA25/57
FPGALINK_P18	AWID11/10	FPGALINK_N18	RDATA26/58
FPGALINK_P17	AWID13/12	FPGALINK_N17	RDATA27/59
FPGALINK_P16	AWPROT2/AWID14	FPGALINK_N16	RDATA28/60
FPGALINK_P15	AWLEN0/2	FPGALINK_N15	RDATA29/61
FPGALINK_P14	AWLEN1/3	FPGALINK_N14	RDATA30/62
FPGALINK_P13	AWSIZE0/1	FPGALINK_N13	RDATA31/63
FPGALINK_P12	AWPROT0/1	FPGALINK_N12	RID1/0
FPGALINK_P11	AWBURST0/1	FPGALINK_N11	RID3/2
FPGALINK_P10	AWLOCK0/1	FPGALINK_N10	RID5/4
FPGALINK_P9	AWCACHE0/2	FPGALINK_N9	RID7/6
FPGALINK_P8	AWCACHE1/3	FPGALINK_N8	RID9/8
FPGALINK_P7	AWVALID	FPGALINK_N7	RID11/10

FPGALINK BUS	Signal (Hi/Lo)	FPGALINK BUS	Signal (Hi/Lo)
FPGALINK_P6	AWREADY	FPGALINK_N6	RID13/12
FPGALINK_P5	BID1/0	FPGALINK_N5	RLAST/RID14
FPGALINK_P4	BID3/2	FPGALINK_N4	RRESP0/1
FPGALINK_P3	BID5/4	FPGALINK_N3	RVALID
FPGALINK_P2	BID7/6	FPGALINK_N2	RREADY
FPGALINK_P1	reserved	FPGALINK_N1	DMA_INT[1]
FPGALINK_P0	reserved	FPGALINK_N0	DMA_INT[0]

Table 7 FPGALINK bus assignments

5 Programmer's model

The example design for a V2F-13MG provides

- SCC memory mapped registers,
- ZBT SSRAM interface.
- Dynamic memory interface to DDR2 SODIMM
- Static memory bus (SMB) to V2M-P1 motherboard
- Multimedia bus (MMB) to V2M-P1 motherboard

5.1 Example AXI memory map

Memory Start	Memory End	Size	Bus	AN305
0x0000_0000	0x1FFF_FFFF	512MB	AXI	External SODIMM
0x2000_0000	0x3FFF_FFFF	512MB	N/C	Internal to A15 TestChip
0x4000_0000	0x5FFF_FFFF	512MB	AXI	External SODIMM
0x6000_0000	0x6FFF_FFFF	256MB	AXI	AXIM (spare)
0x7000_0000	0x7000_FFFF	64KB	APB	FPGA 1 SCC
0x7001_0000	0x7001_FFFF	64KB	APB	PL352 config
0x7002_0000	0x7002_FFFF	64KB	APB	PL341 config
0x7003_0000	0x7003_FFFF	64KB	APB	PL111 config
0x7004_0000	0x7004_FFFF	64KB	APB	APB Aux (spare)
0x7005_0000	0x7005_FFFF	64KB	APB	SBCon
0x7006_0000	0x70FF_FFFF	~16MB	-	Reserved
0x7100_0000	0x73FF_FFFF	48MB	AXI	AXI Aux (spare)
0x7400_0000	0x77FF_FFFF	64MB	SMB	MB Peripherals (nCS3)
0x7800_0000	0x7BFF_FFFF	64KB	SMB	MB Peripherals (nCS7)
0x7C00_0000	0x7C00_FFFF	64KB	APB	FPGA 2 SCC
0x7C01_0000	0x7C01_FFFF	64KB	APB	PL330 config
0x7C02_0000	0x7C02_FFFF	56KB	APB	APB Aux (spare)
0x7C03_0000	0x7CFF_FFFF	~16MB	-	Reserved
0x7D00_2000	0x7DFF_FFFF	16MB	AXI	ZBT
0x7E00_2000	0x7FFF_FFFF	32MB	AXI	AXI Aux (spare)
0x8000_0000	0xFFFF_FFFF	2GB	N/C	Internal to A15 TestChip

Table 8 Memory map

5.2 SCC registers

Table 4-3.1 shows the location of the SCC registers in the example design. The addresses shown are on APB bus offsets from the SCC base address 0x7000_0000 for FPGA1 and 0x7C00_0000 for FPGA2.

Offset address	Name	Reset value	SIF Type	APB Type	Size	Function
0x000	SCC_USER0	0xXXXXXXXX	R/W	R/W	32	R/W register
0x004	SCC_USER1	0xXXXXXXXX	R/W	R/W	32	R/W register
0x100	SCC_DLLLOCK	0xFFXX000X	R/W	R/W	32	DLL locked
0x104	SCC_LED	0x0000000F	R/W	R/W	8	User LEDs control register
0x108	SCC_SW	0x000000XX	R/W	R/W	8	User Switches register
0xFF8	SCC_AID	0xXXXX0302	R/W	R/W	32	Auxiliary ID
0xFFC	SCC_ID	0x41X0305X	R/W	R/W	32	System ID

Table 9 Serial Configuration Control registers

5.2.1 SCC_USERx registers

The registers SCC_USER0 and SCC_USER1 (at offset 0x000-0x004) are general purpose user registers initialized during power up sequence by values from daughter board configuration file.

In existing AN305 build these registers can be used for any purpose by software.

Bits	Name	Access	Function	Default
[31:0]	SCC_USERx[31:0]	R/W	General purpose registers configured during power up from configuration file	hXXXXXXXX

Table 10 SCCCTRL_CFGx bit pattern

5.2.2 DLL lock register

The lock register SCC_DLLLOCK (at offset 0x100) indicated if all DLLs in system have been locked.

Bits	Name	Access	Function	Default
[31:24]	DLL LOCK MASK[7:0]	R/W	These bits indicate if the DLL locked is masked.	8'b11111111
[23:16]	DLL LOCK[7:0]	R/W	These bits indicate if the DLLs are locked or unlocked: b0 = unlocked b1 = locked	8'b111xxxxx
[15:1]	Reserved	Reserved	Reserved	15'b0
[0]	LOCKED	R/W	This bit indicates if all enabled DLLs are locked: b0 = unlocked b1 = locked	1'bx

Table 11 SCC_DLLLOCK bit pattern

5.2.3 User LEDs control register

The SCC_LED register (at offset 0x104) controls the 8 of user LEDs on the V2F-2XV6 daughterboard.

Writing the value b11111111 will light all 8 LED's. LED's can be lit individually for example writing b00000011 will light only the LED0 and LED1.

Bits	Name	Access	Function	Default
[31:8]	Reserved	Read	Reserved	hXXXXXX
[7:0]	LED[7:0]	R/W	These bits control LEDs	h0f

Table 12 SCC_LED bit pattern

5.2.4 User switches register

The SCC_SW register (at offset 0x108) indicates state of the 8 of user switches on the V2F-2XV6 daughterboard.

Bits	Name	Access	Function	Default
[31:8]	Reserved	Read	Reserved	hXXXXXX
[7:0]	SW[7:0]	R/W	These bits indicate state of user switches	hXX

Table 13 SCC_SW bit pattern

5.2.5 SCC_AID register

The SCC_AID register (at offset 0xff8) includes the 16-bit SCC registers description.

Bits	Name	Access	Function	Default
[31:24]	Build	R/W	FPGA build number	hXX
[23:16]	Reserved	R/W	Reserved	hXX
[15:11]	Reserved	R/W	Reserved	5'b00000
[10]	SWREGP	R/W	These bits indicate if SCC_SW register have been implemented	1'b1
[9]	LEDREGP	R/W	These bits indicate if SCC_LED register have been implemented	1'b1
[8]	DLLREGP	R/W	These bits indicate if DLL lock register have been implemented	1'b1
[7:0]	USERREGN	R/W	These bits indicate number of SCC_USERx register	h02

Table 14 SCC_AID bit pattern

5.2.6 SCC_ID registers

The SCC_ID register (at offset 0xffc) includes the 32-bit AN identification.

Bits	Name	Access	Function	Default
[31:24]	Implementor	R/W	Implementor ID	h41
[23:20]	Variant	R/W	Variant Number	hX
[19:16]	Architecture	R/W	Architecture. Have to be 0x0 for Application Notes.	h00
[15:4]	AN	R/W	Application Note number	H305
[3:0]	Revision	R/W	Revision number	hX

Table 15 SCC_ID bit pattern

5.3 SBCon

This block controls writes and reads to the SO-DIMM EEPROM

Offset address	Name	Reset value	SIF Type	APB Type	Size	Function
0x000	SB_CONTROL	0x00000000	R	R	32	Read serial control bits
0x000	SB_CONTROLS	0x00000000	W	W	32	Write serial control bits
0x004	SB_CONTROLC	0x00000000	W	W	32	Clear serial control bits

Table 16 SBCon registers

5.3.1 SB_CONTROL

Bits	Name	Access	Function	Default
[31:2]	Reserved	Read	Reserved	h0
[1]	SDA	Read	Serial data	b0
[0]	SCL	Read	Serial clock	b0

Table 17 SBCon CONTROL register

5.3.2 SB_CONTROLS

Bits	Name	Access	Function	Default
[31:2]	Reserved	Write	Reserved	h0
[1]	SDA	Write	Serial data	b0
[0]	SCL	Write	Serial clock	b0

Table 18 SBCon CONTROLS register

5.3.3 SB_CONTROLC

Bits	Name	Access	Function	Default
[31:2]	Reserved	Write	Reserved	h0
[1]	SDA	Write	Clear serial data	b0
[0]	SCL	Write	Clear serial clock	b0

Table 19 SBCon CONTROLC register

Note:
Software must manipulate the **SCL** and **SDA** bits directly to access the data in the devices. **SDA** is an open-collector signal that is used for sending and receiving data. Set the output (sending) value HIGH before reading the current value.

5.4 Reserved and undefined memory

If reserved memory is accessed, it will be caught by the AXI bus matrix and return a decode error ('DECERR') which generates a data abort. The only exception to this is the spare ports on the matrixes which will return and "OKAY" response and read as 0x0.

6 RTL

Only the top level and low level RTL files are included. AXI components are supplied as netlists. Example scripts are provided to allow building the system with Xilinx ISE tool chain.

6.1 Directory structure

The application note has directories. These include:

- docs : Related documents including this document.
- boardfiles: The files are required to program the design into a V2F-2XV6.
- logical : All the verilog RTL for this design.
- physical : Synthesis and place and route (P&R) scripts and builds for target board.
- software : ARM code to run on the AN305 application note.

6.2 Logical

The logical directory contains all the verilog supplied with this application note. The top level files for this design are v2f_760_wrapper.v and v2f_550t_wrapper.v

6.3 Physical

The physical directory contains pre-synthesised components. The function of custom blocks are shown earlier in **3.2 Module functionality**. The functions of PrimeCells are documented in their respective TRM or User guide.

The physical directory contains the scripts for the ISE tools used in the build process. For tool revision used to build App Note please refer to **/doc/readme.txt** file.

6.4 Building the App Note using Unix

To build the App Note using Unix run the make_revx.scr batch file in the following directories:

/physical/v2f_760/xilinx/scripts.

/physical/v2f_550t/xilinx/scripts.

Where x is the V2F-2XV6 daughterboard revision.

This synthesizes the design and runs place and route on the design pulling in pre synthesized components.

The programmable bit files are generated under

/physical/v2f_760/xilinx/netlist

/physical/v2f_550t/xilinx/netlist

6.5 Using the new bitfile

To use the new bitfiles, the files must be copied to the HBI0217B/AN305 on V2M-P1 USB Flash drive directory. The board.txt text file located at /SITE2/HBI0217B/ in V2M-P1 USB Flash disk have to be edited and APPNOTE field must be modified to point to new bitfiles.

7 Example software

Example software is provided to verify the example design and the V2F-2XV6 daughterboard hardware.

The source files included written in C and assembler. Eclipse project files are included for rebuilding, refer to the readme.txt for tool versions used.

After the Versatile Express system is configured you can upload and execute the example software using debugger on V2P-CA15 processor.

The example code communicates with the user via the debugger's console window. It performs the following tests:

1. Reads the identification register to ensure that the software is executed on the correct system.

5. Tests the ZBT SSRAM for word, half-word and byte accesses.

4. Tests the Static Memory bus by writing reading to V2 Motherboard VRAM.

6. Tests the DDR2 SODIMM.

7. Tests the CLCD and Multimedia Bus.

   Note that for that test VGA, DVI-D or HDMI monitor that supports VGA resolution have to be connected to the V2M-P1 DVI-I connector.

8. Test the DMA controller, and interrupt signal from DMA.

8 I/O Timing Requirements

All of these specific timing requirements refer to the r0p0 revision of the AN305. All units are in nano-seconds “ns” and have been rounded to a worst case value.

Signals with setup, hold and clock to data values are bidirectional signals or have been grouped by function in the table.

8.1 Default, minimum and maximum operating frequencies

Clock source	Clock signal	Clock domain	Default Freq (MHz)	Min Freq (MHZ)	Max Freq (MHz)
F2_REFCLK24MHz	REFCLK24MHz	FPGA2 REFCLK24MHz	24	24	24
OSC0	ACLK	FPGA2 System clock	80	2	85
OSC1	MCLK	FPGA2 DDR clock	120	85	120
OSC2	CLCDCLK	CLCD clock	23.75	2	70
OSC3	ACLK_LINK ACLK (2x input frequency)	FPGA1 System clock and link clock	35	33	40
REFCLK24MHZ	ZCLK (fixed multiple of REFCLK24MHz)	ZBT clock	133	133	133
ACLK_EMM	ACLK_EMM	AXI clock in	25	25	25

Table 20 Default and maximum operating frequencies

8.2 Demux AXI timing requirements

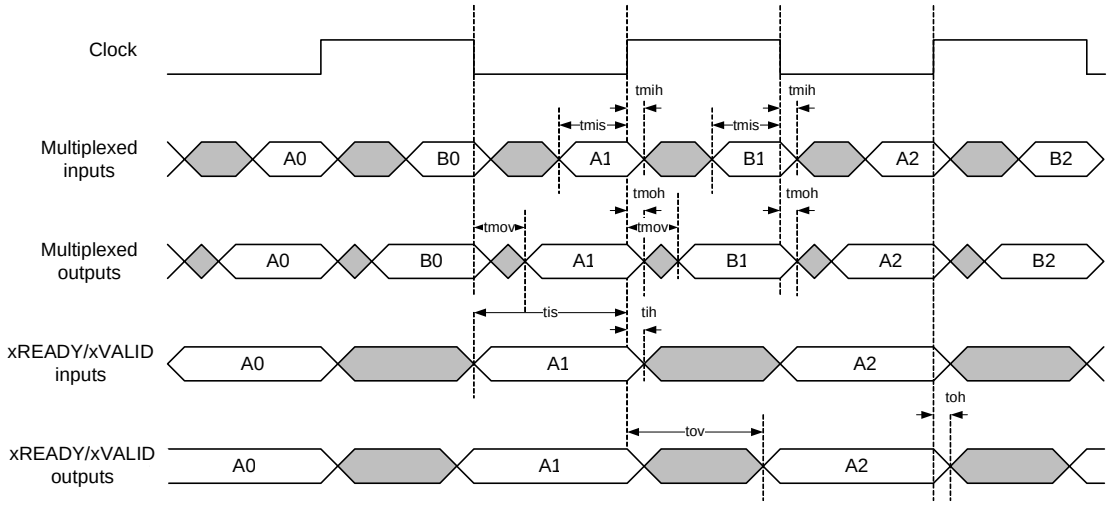

Figure 8-1 A9 AXI timing requirements

Dir	Name	Description	Value
Inputs	tmis [ns]	Max multiplexed inputs setup to clock	6.7
	tmih [ns]	Max multiplexed inputs hold to clock	2.45
	tis [ns]	Max non multiplexed inputs setup to clock	19.7
	tih [ns]	Max non multiplexed input hold to clock	2.45
Outputs	tmov [ns]	Max clock to multiplexed data valid	7.2
	tmoh [ns]	Min clock to multiplexed data invalid	1.65
	tov [ns]	Max non multiplexed outputs valid	20.3
	toh [ns]	Min non multiplexed outputs invalid	1.65

Table 21 AXI Slave Input/Output timing to ACLK_EMM clock input

8.3 FPGALINK Mux/Demux AXI timing requirements

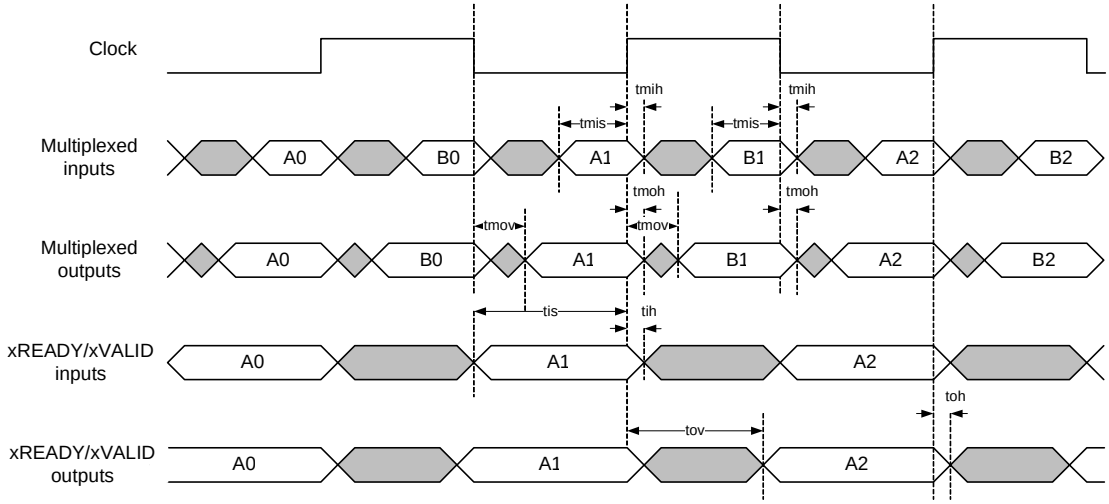

Figure 8-2 AXI timing requirements

Dir	Name	Description	Value
Inputs	tmis [ns]	Max multiplexed inputs setup to clock	6
	tmih [ns]	Max multiplexed inputs hold to clock	3
	tis [ns]	Max non multiplexed inputs setup to clock	12
	tih [ns]	Max non multiplexed input hold to clock	3
Outputs	tmov [ns]	Max clock to multiplexed data valid	6
	tmoh [ns]	Min clock to multiplexed data invalid	3
	tov [ns]	Max clock to non multiplexed outputs valid	12
	toh [ns]	Min clock to non multiplexed outputs invalid	3

Table 22 AXI Master Input/Output timing to ACLK_LINK clock output

8.4 MMB signals timing.

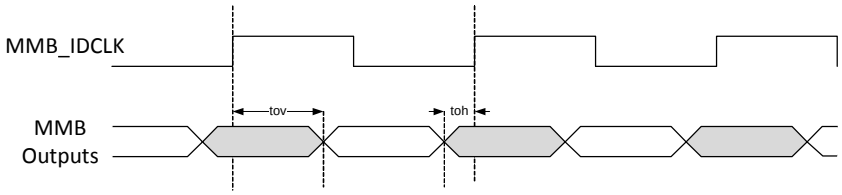

Figure 8-3 MMB Outputs Timing

Dir	Name	Description	Value
Outputs	tov [ns]	Max rising edge of MMB_IDCLK to data valid	6

	toh [ns]	Min rising edge of MMB_IDCLK to data invalid	-2

Table 23 MMB Output timing

8.5 SMB signals timing.

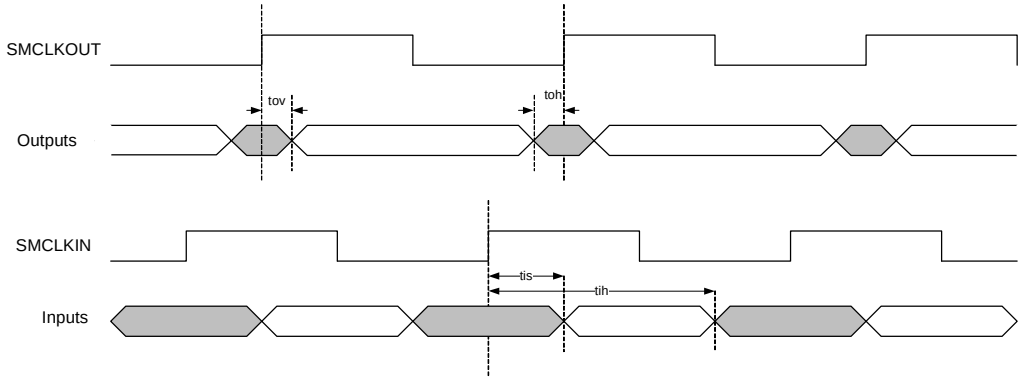

Figure 8-4 SMB Inputs and Outputs Timing

Dir	Name	Description	Value
Inputs	tis [ns]	Max inputs setup to rising edge of SMCLKIN	7
	tih [ns]	Max inputs hold to rising edge of SMCLKIN	17
Outputs	tov [ns]	Max rising edge of SMCLKOUT to data valid	4
	toh [ns]	Min rising edge of SMCLKOUT to data invalid	2

Table 24 SMB Input/Output timing