Application Note 224

Example LogicTile Express 3MG design for
a CoreTile Express A9x4.

Document number: ARM DAI 0224

Issued: March 2011

ARM

Application Note 224
Example LogicTile Express 3MG design for a CoreTile Express A9x4

Release information

The following changes have been made to this Application Note.

Change history

Date	Issue	Change
November 28, 2009	A	First release
October 4, 2010	B	Updated default frequency for CLCD
March 14, 2011	C	Corrected SB_INT polarity.

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 3MG (V2F-1XV5). It will examine the contents of the V2F-1XV5, the system interconnect, the clock structure, and specifics of the programmer's model directly relevant to V2F-1XV5 operation.

On reading this Application Note the user should be in a position to debug and analyze the operation of the provided image. He also should be able to make changes to the provided V2F-1XV5 design, connect his own AXI or AHB based masters and AXI, AHB or APB slaves.

1.2 Overview of hardware platform

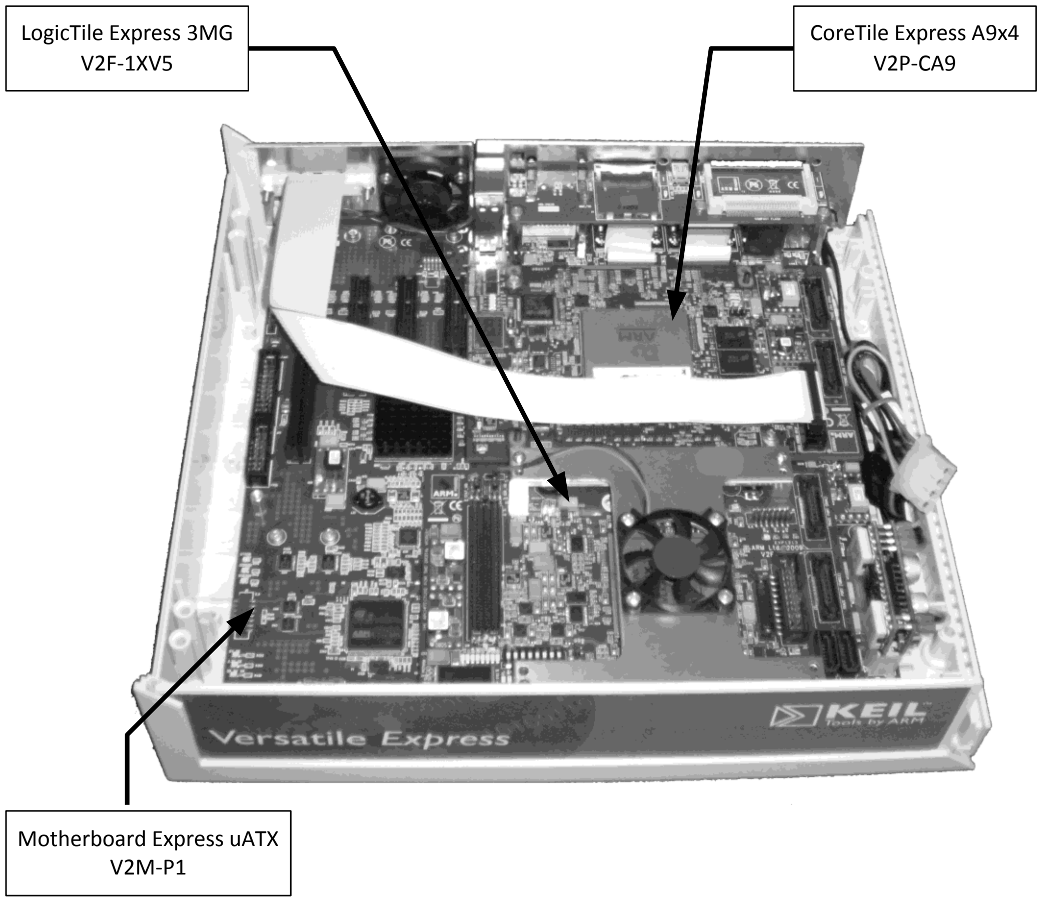


Figure 1-1 V2F-1XV5 and V2P-CA9 daughterboards on a V2M-P1 motherboard

This application note is designed to work on a Motherboard Express uATX (V2M-P1) fitted with CoreTile Express A9x4 (V2P-CA9) in Site 1 and LogicTile Express 3MG V2F-1XV5 fitted in Site 2, as shown in **Figure 1-1 V2F-1XV5 and V2P-CA9 daughterboards on a V2M-P1 motherboard**.

2 Getting Started

The steps below show you how to set up the hardware platform and copy the necessary configuration files to the V2M-P1 USB Flash disk and program the LogicTile Express FPGA image.

1. Plug the V2P-CA9 daughterboard onto site 1 and the V2F-1XV5 daughterboard onto site 2 of the V2 Motherboard as described in **Quick Start Guide for the Versatile Express Family - Adding Daughterboards**.

2. 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**.

3. After the board has been connected via USB and the PC recognizes the motherboard as a USB Flash Drive, copy the application note boardfiles/SITE2/HBI0192x/AN224 directory to the /SITE2/HBI0192x directory on V2M-P1 USB Flash disk. Where x is the V2F-1XV5 daughterboard revision.

4. Edit board.txt text file located at /SITE2/HBI0192x/ on the V2M-P1 USB Flash disk. Modify the APPNOTE field to AN224\a224rxpx.txt where rxpx is the required revision of the application note.

5. Power cycle the boards using the red Power button to upload the application note image to V2F-1XV5 FPGA.

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-1XV5 image exposes one muxed 64 bit AXI master port, External Muxed Slave (EMS) and one muxed 64 bit AXI slave port, External Muxed Master (EMM) to the V2P-CA9 daughterboard. It also makes connections to the Static Memory Bus, Multimedia Bus, System Bus to the V2M-P1 motherboard.

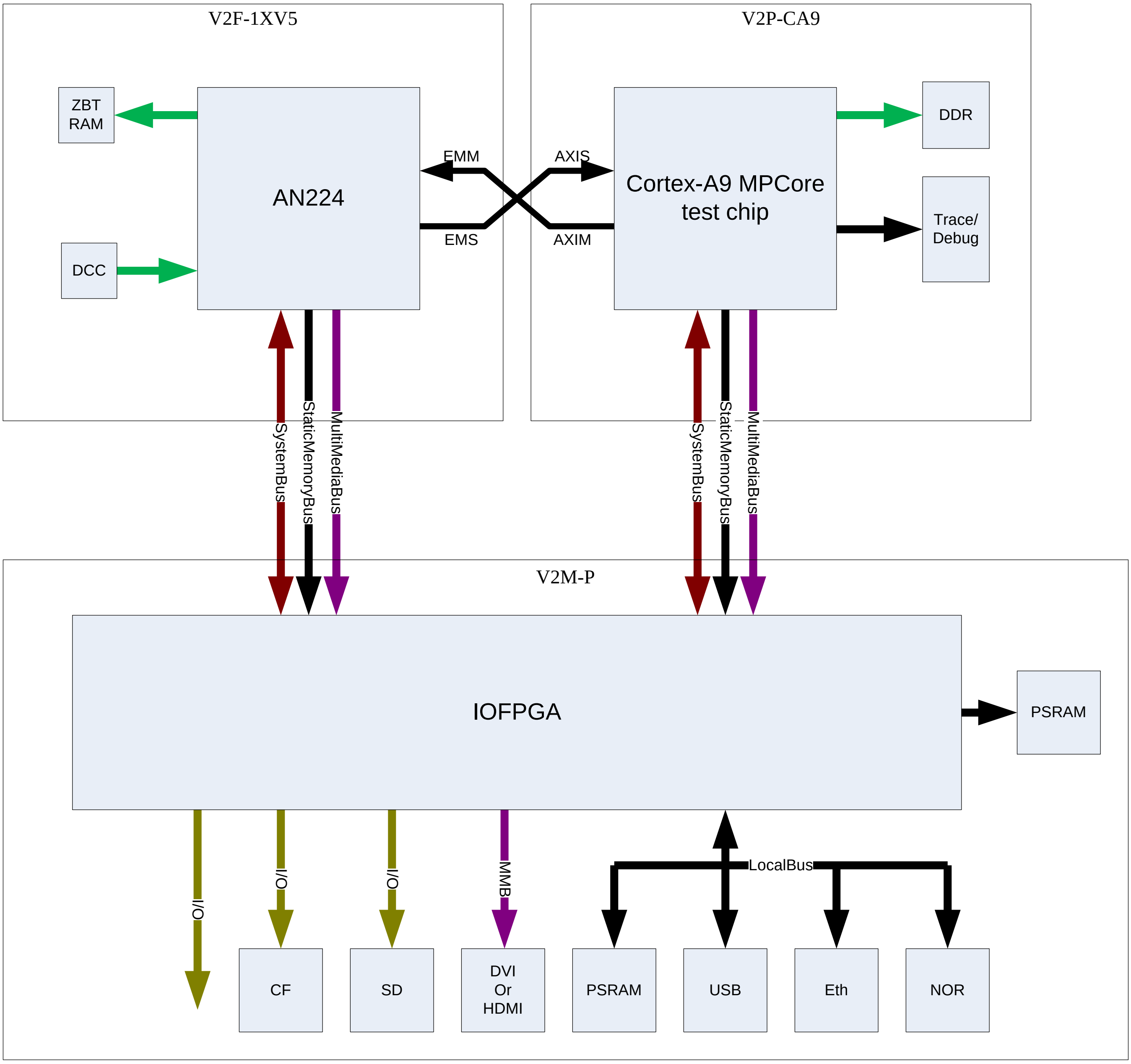


Figure 3-1 Block level architecture

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

3.1 AN224 architecture

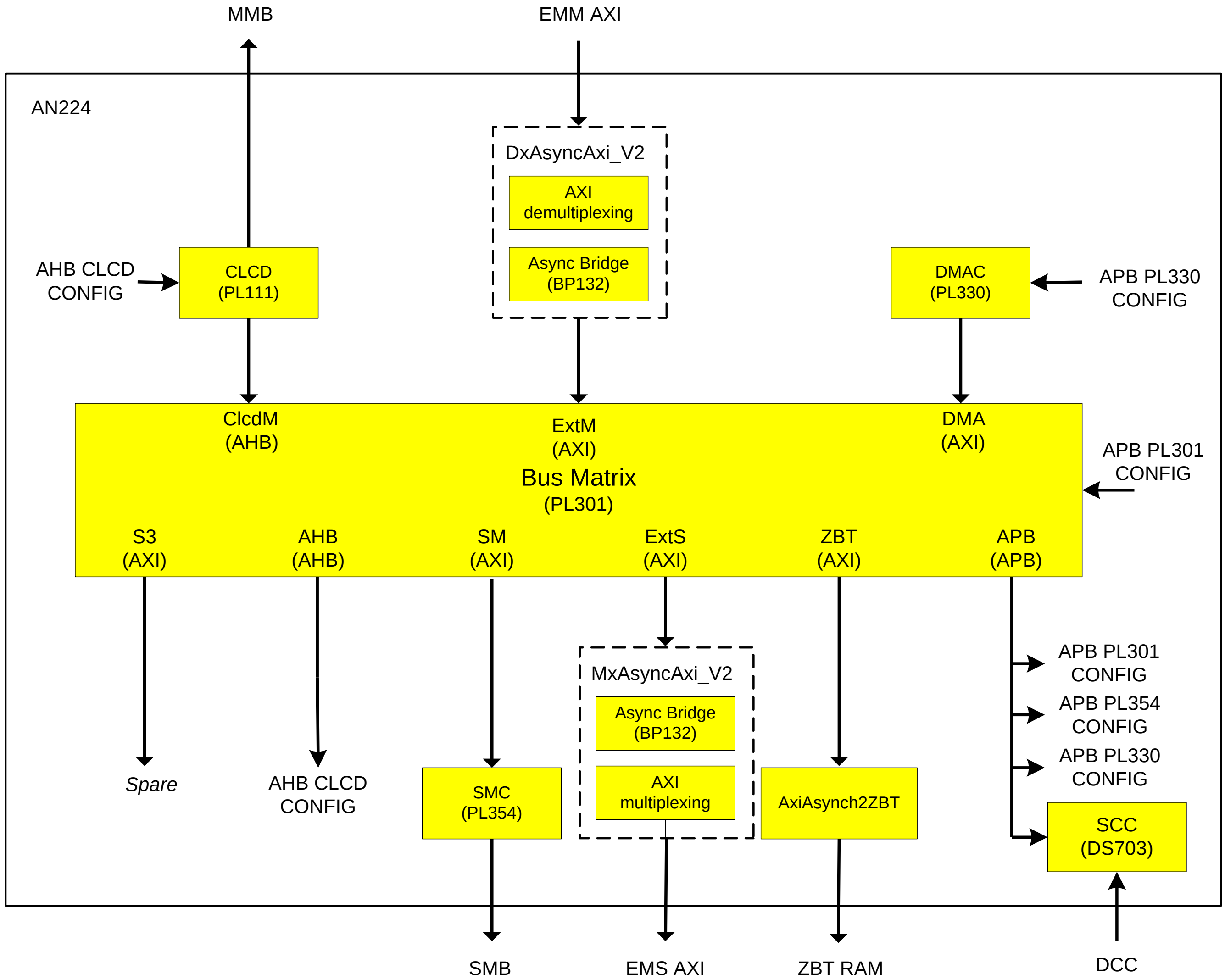


Figure 3-2 AN224 Architecture

3.2 Module functionality

The function of each of these blocks is as follows:

PL301 Bus Matrix

This provides the interconnect structure. It allows any of the 2 AXI and 1 AHB slave ports to connect to one of the 4 AXI, 1 AHB and 1 APB master ports without blocking the other masters (unless they both try to access the same slave).

It also contains the decoder mapping to determine the address map, and a scheme to determine priority of competing masters to a single slave.

The PL301 Bus Matrix blocks all accesses to the EMS AXI master port from the EMM AXI slave port. It prevents AXI transactions loops and reduces the number of EMS AXI master port ID bits.

See the **PrimeCell High-Performance Matrix (PL301) Technical Reference Manual** for more information.

PL111 CLCD

This is a Color Liquid Crystal Display controller. The ARM PrimeCell PL111 is used in this design. See the **PrimeCell Color LCD Controller (PL111) Technical Reference Manual** for more information.

The CLCD signals are connected via MultiMedia Bus (MMB) to a DVI Multiplexor on the V2M-P1 motherboard. The multiplexor selects the source to supply video to the DVI-I connector as either:

• the MMB bus from the V2P-CA9

• the MMB bus from the application note

• the CLCD controller in the V2M-P1 motherboard.

For information how configure the DVI multiplexor see **Motherboard Express µATX V2M-P1 Technical Reference Manual** on page 2-15.

DxAsyncAxi_V2

This block provides an AXI demux block which demultiplexes the 2:1 muxed 64 bit EMM AXI bus and an asynchronous bridge (BP132) to allow the multiplexed bus to operate in a different clock domain. It increases the internal operating frequency and introduces minimum three clock cycles of latency.

PL330 DMAC

This is a Direct Memory Access Controller. The ARM PrimeCell PL330 is used in this design. For more information please refer to the **PrimeCell® DMA Controller (PL330) Technical Reference Manual**.

PL354 SMC

This is a Static Memory Controller. The ARM PrimeCell PL354 is used in this design. See the **PrimeCell Static Memory Controller (PL350 series) Technical Reference Manual** for more information.

MxAsyncAxi_V2

This block provides an asynchronous bridge (BP132) to allow the multiplexed bus to operate in a different clock domain than the internal system domain. It also includes an AXI 2:1 mux block which multiplexes the 64 bit External Slave AXI bus. The asynchronous bridge increases the internal operating frequency and introduces a minimum of three clock cycles of latency.

AxiAsynch2ZBT

This is a 64 bit bridge which converts AXI transfers into ZBT SRAM transfers. The ZBT SRAM operates asynchronous to AXI domain.

SCC

This is example of a Serial Configuration Controller (SCC). The SCC provides a standard serial interface to a V2F-1XV5 Daughterboard Configuration Controller (DCC). The DCC uses this interface by issuing commands to receive/transmit information from/to the SCC registers in the FPGA.

All DCC defined commands have been supported. The SCC enables identification and of systems, passes DCM lock status to V2M-P1 motherboard reset controller as well to light the V2F-1XV5 daughterboard LEDs and update an internal register according to the states of the V2F-1XV5 switches. The SCC also provides two general purpose user registers which value can be uploaded during power up sequence from configuration file.

For more information about DCC interface and commands please refer to **LogicTile Express 3MG TRM**.

The SCC registers are also accessible via APB bus which allows to access SCC registers via AN224 bus matrix. For programming model see **5.2 SCC registers**.

3.3 Modules revision

Module	**Revision**
PL301 Bus Matrix	r1p2
PL111 CLCD	r0p2
DxAsynchAxi_V2	r0p0
PL330 DMAC	r0p0
PL354 SMC	r2p1
MxAsynchAxi_V2	r0p0
SCC	r0p0

Table 3-1 Modules revision in AN224 r0p1

3.4 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 V2F-1XV5 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**.

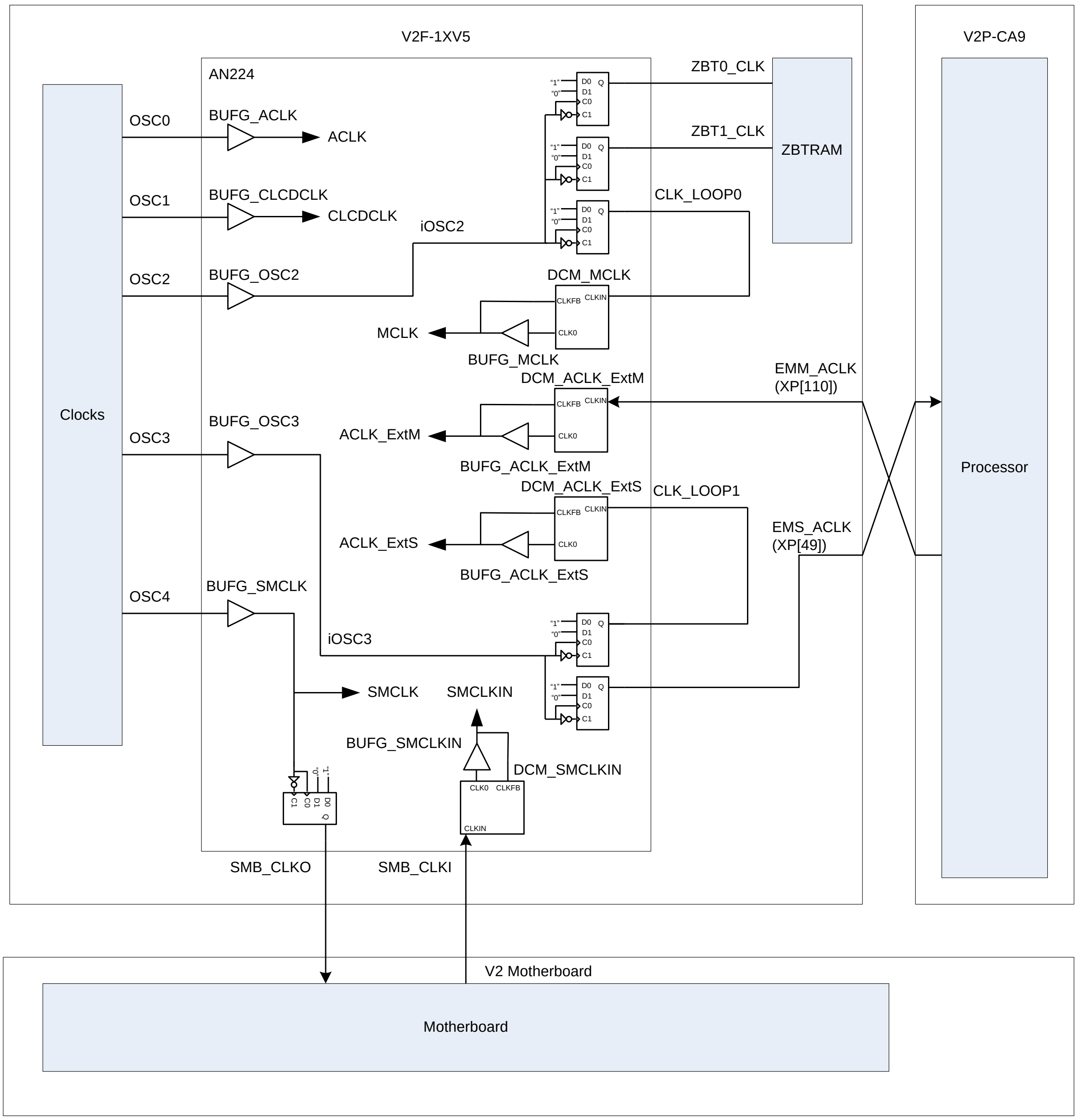


Figure 3-3 Clock architecture for AN224

There are 6 clock domains in this design:

AN224 AXI System Bus

The Example Internal AXI System Bus is clocked by ACLK. The ACLK is generated through a clock buffer from OSC0. The Internal AXI System Bus clock domain includes:

- Bus Matrix
- DMA Controller
- ZBT SRAM controller AXI bus
- CLCD controller AHB buses
- SMC AXI bus
- De-multiplexing AXI master bus, multiplexing AXI slave bus
- APB subsystem

CLCD

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.

ZBT SRAM

OSC2 is the source for MCLK which is used to clock ZBT SRAM memory and the ZBT controller. The clock loop LOOP0 and MCLK_DLL are used to de-skew MCLK and clocks provided to ZBT memory devices.

External AXI Slave Bus

The External AXI Slave Bus is clocked by OSC3. The clock loop LOOP1 and ExtS_DLL are used to de-skew internal ACLK_ExtS and external EMS_ACLK clocks.

Static Memory Bus

The SMCLK is used to drive Static Memory Controller and Static Memory bus. The SMCLK is directly connected to OSC4. The SMCLKIN feedback clock is used to register data from Static memory bus slave implemented on V2 Motherboard.

External AXI Master Bus

The External AXI Slave Bus is clocked by EMM_ACLK provided by V2P-CA9 daughterboard.

3.5 Interrupt architecture

There are 4 interrupts sources generated by AN224 which are connected to V2P-CA9 daughterboard interrupt controller via the V2M-P1 motherboard. The interrupts signals have been inverted in AN224 top level and V2M-P1 motherboard to provide the correct polarity for the interrupt controller in the Cortex-A9 MPCore test chip on V2P-CA9 .

Signal	Source	V2M-P1 int	V2P-CA9 int
Clcdintr	CLCD	SB_INT[0]	INTSOURCE[36]
smc_int	SMC	SB_INT[1]	INTSOURCE[37]
dma_int[0]	DMAC	SB_INT[2]	INTSOURCE[38]
dma_int[1]	DMAC	SB_INT[3]	INTSOURCE[39]

Table 3-2 Interrupts

3.6 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-1XV5 wrapper (V2F_1XV5_wrapper)

This level defines the mapping from the V2F-1XV5 XN, XP and XS busses into their functional allocations. The wrapper contains also clock structure and ties off static pins.

4.2 AN224 top level (AN224_toplevel)

This level connects all the Application Note components together. This includes all the major modules as shown in **Figure 3-2 AN224 Architecture**. The RTL is provided so modules can be added and removed.

Therefore all modules except SCC are or include high value AXI blocks, there are only provided as .NGO netlists.

Two constrain files (ucf) are included, one for the application note and one which shows all FPGA pins used on the V2F-1XV5 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 X 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 AXI timing requirements**.

The multiplexing and demultiplexing blocks are provided as NGO netlist.

4.5 Header HDRX pin allocation

The two AXI buses connect to the HDRX as shown. Multiplexed AXI slave bus EMM connects via HDRX header to AXI External Master and multiplexed AXI master bus EMS connects via header HDRX to AXI External Slave.

4.5.1 Multiplexed AXI master bus

HDRX pin	X bus	Signal (Hi/Lo)	HDRX pin	X bus	Signal (Hi/Lo)
G1	XN6	EMS_WDATA0/32	D3	XP3	EMS_ WID14/WLAST
G2	XP6	EMS_WDATA1/33	E1	XN4	EMS_WSTRB0/4
H2	XN7	EMS_WDATA2/34	E2	XP4	EMS_WSTRB1/5
H3	XP7	EMS_WDATA3/35	F2	XN5	EMS_WSTRB2/6
J1	XN8	EMS_WDATA4/36	F3	XP5	EMS_WSTRB3/7
J2	XP8	EMS_WDATA5/37	G25	XS6	EMS_WVALID
K2	XN9	EMS_WDATA6/38	K18	XP59	EMS_BRESP1/0
K3	XP9	EMS_WDATA7/39	K25	XS9	EMS_BVALID
A4	XN10	EMS_WDATA8/40	H25	XS7	EMS_WREADY
A5	XP10	EMS_WDATA9/41	A10	XN30	EMS_AWADDR0/16
B5	XN11	EMS_WDATA10/42	A11	XP30	EMS_AWADDR1/17
B6	XP11	EMS_WDATA11/43	B11	XN31	EMS_AWADDR2/18
C4	XN12	EMS_WDATA12/44	B12	XP31	EMS_AWADDR3/19
C5	XP12	EMS_WDATA13/45	C10	XN32	EMS_AWADDR4/20
D5	XN13	EMS_WDATA14/46	C11	XP32	EMS_AWADDR5/21
D6	XP13	EMS_WDATA15/47	D11	XN33	EMS_AWADDR6/22
E4	XN14	EMS_WDATA16/48	D12	XP33	EMS_AWADDR7/23
E5	XP14	EMS_WDATA17/49	E10	XN34	EMS_AWADDR8/24
F5	XN15	EMS_WDATA18/50	E11	XP34	EMS_AWADDR9/25
F6	XP15	EMS_WDATA19/51	F11	XN35	EMS_AWADDR10/26
G4	XN16	EMS_WDATA20/52	F12	XP35	EMS_AWADDR11/27
G5	XP16	EMS_WDATA21/53	G10	XN36	EMS_AWADDR12/28
H5	XN17	EMS_WDATA22/54	G11	XP36	EMS_AWADDR13/29
H6	XP17	EMS_WDATA23/55	H11	XN37	EMS_AWADDR14/30
J4	XN18	EMS_WDATA24/56	H12	XP37	EMS_AWADDR15/31
J5	XP18	EMS_WDATA25/57	G7	XN26	EMS_AWID1/0
K5	XN19	EMS_WDATA26/58	G8	XP26	EMS_AWID3/2
K6	XP19	EMS_WDATA27/59	H8	XN27	EMS_AWID5/4
A7	XN20	EMS_WDATA28/60	H9	XP27	EMS_AWID7/6
A8	XP20	EMS_WDATA29/61	J7	XN28	EMS_AWID9/8
B8	XN21	EMS_WDATA30/62	J8	XP28	EMS_AWID11/10
B9	XP21	EMS_WDATA31/63	K8	XN29	EMS_AWID13/12
A1	XN0	EMS_WID1/0	C8	XP22	EMS_AWPROT2/AWID14
A2	XP0	EMS_WID3/2	F8	XN25	EMS_AWLEN0/2
B2	XN1	EMS_WID5/4	F9	XP25	EMS_AWLEN1/3
B3	XP1	EMS_WID7/6	D9	XP23	EMS_AWSIZE0/1
C1	XN2	EMS_WID9/8	C7	XN22	EMS_AWPROT0/1
C2	XP2	EMS_WID11/10	D8	XN23	EMS_AWBURST0/1
D2	XN3	EMS_WID13/12	K9	XP29	EMS_AWLOCK0/1

HDRX pin	X buss	Signal (Hi/Lo)	HDRX pin	X bus	Signal (Hi/Lo)
E7	XN24	EMS_AWCACHE0/2	A19	XN60	EMS_RDATA0/32
E8	XP24	EMS_AWCACHE1/3	A20	XP60	EMS_RDATA1/33
E25	XS4	EMS_AWVALID	B20	XN61	EMS_RDATA2/34
K20	XN69	EMS_BID1/0	B21	XP61	EMS_RDATA3/35
K21	XP69	EMS_BID3/2	C19	XN62	EMS_RDATA4/36
H23	XN77	EMS_BID5/4	C20	XP62	EMS_RDATA5/37
H24	XP77	EMS_BID7/6	D20	XN63	EMS_RDATA6/38
F25	XS5	EMS_AWREADY	D21	XP63	EMS_RDATA7/39
J22	XN78	EMS_BID9/8	E19	XN64	EMS_RDATA8/40
J23	XP78	EMS_BID11/10	E20	XP64	EMS_RDATA9/41
K23	XN79	EMS_BID13/12	F20	XN65	EMS_RDATA10/42
K24	XP79	NC/EMS_BID14	F21	XP65	EMS_RDATA11/43
J25	XS8	EMS_BREADY	G19	XN66	EMS_RDATA12/44
G13	XN46	EMS_ARADDR0/16	G20	XP66	EMS_RDATA13/45
G14	XP46	EMS_ARADDR1/17	H20	XN67	EMS_RDATA14/46
H14	XN47	EMS_ARADDR2/18	H21	XP67	EMS_RDATA15/47
H15	XP47	EMS_ARADDR3/19	J19	XN68	EMS_RDATA16/48
J13	XN48	EMS_ARADDR4/20	J20	XP68	EMS_RDATA17/49
J14	XP48	EMS_ARADDR5/21	A22	XN70	EMS_RDATA18/50
A16	XN50	EMS_ARADDR6/22	A23	XP70	EMS_RDATA19/51
A17	XP50	EMS_ARADDR7/23	B23	XN71	EMS_RDATA20/52
B17	XN51	EMS_ARADDR8/24	B24	XP71	EMS_RDATA21/53
B18	XP51	EMS_ARADDR9/25	C22	XN72	EMS_RDATA22/54
C16	XN52	EMS_ARADDR10/26	C23	XP72	EMS_RDATA23/55
C17	XP52	EMS_ARADDR11/27	D23	XN73	EMS_RDATA24/56
D17	XN53	EMS_ARADDR12/28	D24	XP73	EMS_RDATA25/57
D18	XP53	EMS_ARADDR13/29	E22	XN74	EMS_RDATA26/58
E16	XN54	EMS_ARADDR14/30	E23	XP74	EMS_RDATA27/59
E17	XP54	EMS_ARADDR15/31	F23	XN75	EMS_RDATA28/60
C13	XN42	EMS_ARID1/0	F24	XP75	EMS_RDATA29/61
C14	XP42	EMS_ARID3/2	G22	XN76	EMS_RDATA30/62
D14	XN43	EMS_ARID5/4	G23	XP76	EMS_RDATA31/63
D15	XP43	EMS_ARID7/6	G16	XN56	EMS_RID1/0
E13	XN44	EMS_ARID9/8	G17	XP56	EMS_RID3/2
E14	XP44	EMS_ARID11/10	H17	XN57	EMS_RID5/4
F14	XN45	EMS_ARID13/12	H18	XP57	EMS_RID7/6
J11	XP38	EMS_ARPROT2/ARID14	J16	XN58	EMS_RID9/8
B14	XN41	EMS_ARLEN0/2	J17	XP58	EMS_RID11/10
B15	XP41	EMS_ARLEN1/3	K17	XN59	EMS_RID13/12
K12	XP39	EMS_ARSIZE0/1	F18	XP55	EMS_RLAST/RID14
J10	XN38	EMS_ARPROT0/1	F17	XN55	EMS_RRESP0/1
K11	XN39	EMS_ARBURST0/1	B25	XS1	EMS_RVALID
F15	XP45	EMS_ARLOCK0/1	A25	XS0	EMS_RREADY
A13	XN40	EMS_ARCACHE0/2	K14	XN49	1’b1
A14	XP40	EMS_ARCACHE1/3	K15	XP49	EMS_ACLK
D25	XS3	EMS_ARVALID			
C25	XS2	EMS_ARREADY			

Table 4.1 Header HRDX EMS bus pin allocation

4.5.2 Multiplexed AXI slave bus

HDRX pin	X bus	Signal (Hi/Lo)	X bus	FPGA IO	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

HDRX pin	X bus	Signal (Hi/Lo)	X bus	FPGA IO	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 4.2 Header HRDX EMM bus pin allocation

4.6 Header HDRY pin allocation

4.6.1 Multimedia bus

HDRY pin	MMB bus	CLCD Signal	HDRY 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 4.3 Multimedia bus pin allocation

4.6.2 Static memory bus

HDRY pin	SMB bus	SMC Signal	HDRY 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 4.4 Static Memory bus pin allocation

4.6.3 Interrupts

HDRY 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 4.5 Interrupts outputs pin allocation

5 Programmer's model

The example design for a V2F-3MG provides

- SCC memory mapped registers,
- ZBT SSRAM,
- Static memory bus (SMB) to V2M-P1 motherboard
- Multimedia bus (MMB) to V2M-P1 motherboard
- AXI master bus to V2P-CA9 daughterboard slave port.
- Note that the AXI master from V2P-CA9 can not access the V2P-CA9 slave port via the V2F-1XV5. Only the DMA and CLCD controllers in the V2F-1XV5 can access the V2P-CA9 slave port.

5.1 Example AXI memory map

Memory Start	Memory End	Size	Bus	AN224
0x0000_0000	0x03FF_FFFF	64MB	AXI ZBT	ZBT SRAM
0x0400_0000	0xDFFF_FFFF	3520MB	AXI EMS	AXI External Slave
0xE000_0000	0xE000_0FFF	4KB	APB	SCC registers
0xE000_1000	0xE000_1FFF	4KB	APB	PL301
0xE000_2000	0xE000_2FFF	4KB	APB	PL352
0xE000_3000	0xE000_3FFF	4KB	APB	PL330
0xE000_4000	0xE000_FFFF	48KB	APB	Reserved
0xE001_0000	0xE0FF_FFFF	16320KB	-	Reserved
0xE100_0000	0xE100_0FFF	4KB	AHB	CLCD controller
0xE100_1000	0xE1FF_FFFF	16380KB	AHB	Reserved
0xE200_0000	0xE2FF_FFFF	16MB	AXI S3	Reserved
0xE300_0000	0xE3FF_FFFF	16MB	-	Reserved
0xE400_0000	0xE400_1FFF	8KB	SMB	MB Peripherals
0xE400_2000	0xE4FF_FFFF	16376KB	SMB	Reserved
0xE500_0000	0xE7FF_FFFF	48MB	SMB	Reserved
0xE800_0000	0xEFFF_FFFF	128MB	SMB	MB Video RAM (8MB allias)
0xF000_0000	0xFFFF_FFFF	256MB	AXI ZBT	ZBT RAM (16MB allias)

Table 5.1 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 0xE000_0000.

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	32	DLL locked
0x104	SCC_LED	0x0000000F	R	R/W	8	User LEDs control register
0x108	SCC_SW	0x000000XX	R/W	R	8	User Switches register
0xFF8	SCC_AID	0xXXXX0302	R/W	R	32	Auxiliary ID
0xFFC	SCC_ID	0x41X0224X	R/W	R	32	System ID

Table 5.2 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 AN224 build these registers can be used for any purpose by software.

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

Table 5.3 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]	Read	These bits indicate if the DLL locked is masked.	8'b11111111
[23:16]	DLL LOCK[7:0]	Read	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	Read	This bit indicates if all enabled DLLs are locked: b0 = unlocked	1'bx

			b1 = locked	

Table 5.4 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-1XV5 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]	Read/Write	These bits control LEDs	h0f

Table 5.5 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-1XV5 daughterboard.

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

Table 5.6 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	Read	FPGA build number	hXX
[23:16]	Reserved	Read	Reserved	hXX
[15:11]	Reserved	Read	Reserved	5'b00000
[10]	SWREGP	Read	These bits indicate if SCC_SW register have been implemented	1'b1
[9]	LEDREGP	Read	These bits indicate if SCC_LED register have been implemented	1'b1
[8]	DLLREGP	Read	These bits indicate if DLL lock	1'b1

			register have been implemented	
[7:0]	USERREGN	Read	These bits indicate number of SCC_USERx register	h02

Table 5.7 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	Read	Implementor ID	h41
[23:20]	Variant	Read	Variant Number	hX
[19:16]	Architecture	Read	Architecture. Have to be 0x0 for Application Notes.	h00
[15:4]	AN	Read	Application Note number	h224
[3:0]	Revision	Read	Revision number	hX

Table 5.8 SCC_ID bit pattern

5.3 Reserved and undefined memory

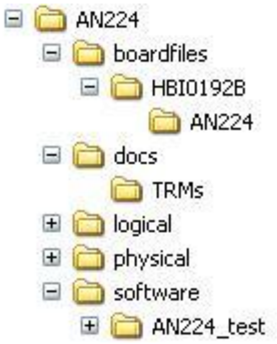
If reserved memory is accessed, it will be caught by the AXI bus matrix and return a decode error ('DECERR') which generates a data abort. The only exception to this is the APB subsystem, which has 13 spare PSEL signals for expansion. If this address range is accessed the AXIToAPB bridge will return an 'OKAY' response.

6 RTL

Only the top level and SCC RTL files are included. AXI components are supplied as netlists. However Amba Designer XML configuration files are provided for that components to allow rebuild them.

Example files are provided to allow building the system with Xilinx ISE tools.

6.1 Directory structure



The application note has directories. These are:

- boardfiles: The files are required to program the design into a V2F-1XV5.
- docs : Related documents including this document.
- 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 AN224 application note.

6.2 Logical

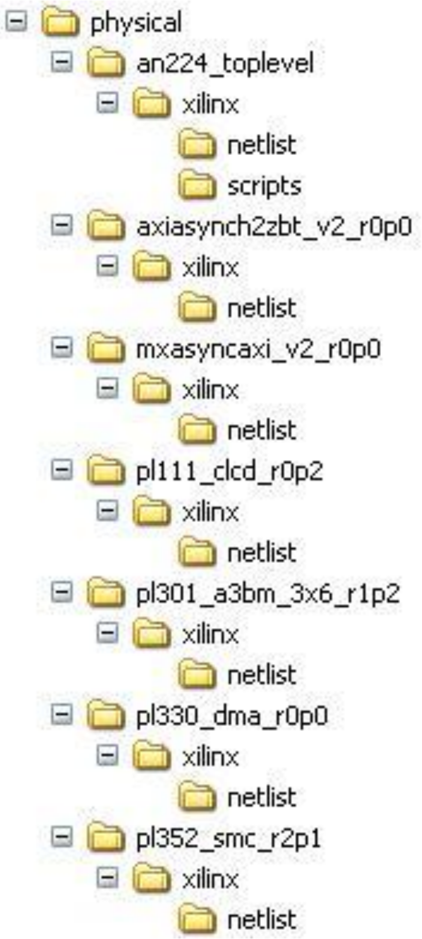
The logical directory contains all the verilog supplied with this application note. It contains also Amba Designer XML configuration files which can be used to regenerate verilog for ARM PrimeCell used in the example design.

The top level for this system is in V2F_1XV5_wrapper.

6.3 Physical

The physical directory contains pre-synthesised components. The function of each block is shown earlier in **3.2 Module functionality**.

Each PrimeCell or other large IP block has its own directory (for example pl111_clcd_r0p2).



The physical directory contains the scripts for the 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 Microsoft Windows

To build the App Note using Microsoft Windows run the make_revx.bat batch file in the following directory: physical/an224_toplevel/xilinx/scripts. Where x is the V2F-1XV5 daughterboard revision.

This synthesizes the design and runs place and route using Xilinx ISE tool on the design pulling in pre synthesized components.

A programmable bit file is generated under physical/an224_toplevel/xilinx/netlist and copied to boardfiles/SITE2/HBI0192x/AN224 as an224cust.bit.

6.5 Building the App Note using Unix

To build the App Note using Unix run the make_revx.scr batch file in the following directory: /physical/an224_toplevel/xilinx/scripts. Where x is the V2F-1XV5 daughterboard revision.

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

A programmable bit file is generated under /physical/an224_toplevel/xilinx/netlist and copied to boardfiles/SITE2/HBI0192x/AN224 as an224cust.bit.

6.6 Using the new bitfile

To use the new an224cust.bit bitfile, the file must be copied to the SITE2/HBI0192x/AN224 on V2M-P1 USB Flash drive directory. Where x is the V2F-1XV5 daughterboard revision.

The board.txt text file located at /SITE2/HBI0192x/ in V2M-P1 USB Flash disk have to be edited and APPNOTE field must be modified to AN224\a224cust.txt.

7 Example software

Example software is provided to verify the example design and the V2F-1XV5 daughterboard hardware.

The source files included written in C and assembler.

After the Versatile Express system is configured you can upload and execute the example software using debugger on V2P-CA9 processor.

The example code communicates with the user via the debugger's console window. It operates as follows:

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.

5. 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.

6. Test the DMA controller, AXI bus to External slave, and interrupt signal from DMA.

8 I/O Timing Requirements

All of these specific timing requirements refer to the r0p1 revision of the AN224. 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	Min Freq	Max Freq
OSC0	ACLK	AN224 AXI System Bus	90MHz	2MHz	90MHz
OSC1	CLCDCLK	CLCD	23.75MHz	2MHz	62.5MHz
OSC2	MCLK	ZBT SRAM	133MHz	33MHz	133MHz
OSC3	ACLK_ExtS/EMS_ACLK	Ext AXI Slave Bus	33MHz	32MHz	33MHz
OSC4	SMCLK/SMCLKIN	Static Memory Bus	50MHz	32MHz	50MHz
EMM_ACLK	ACLK_ExtM	Ext AXI Master Bus	-	32MHz	45MHz

Table 8-1 Default and maximum operating frequencies

8.2 AXI timing requirements

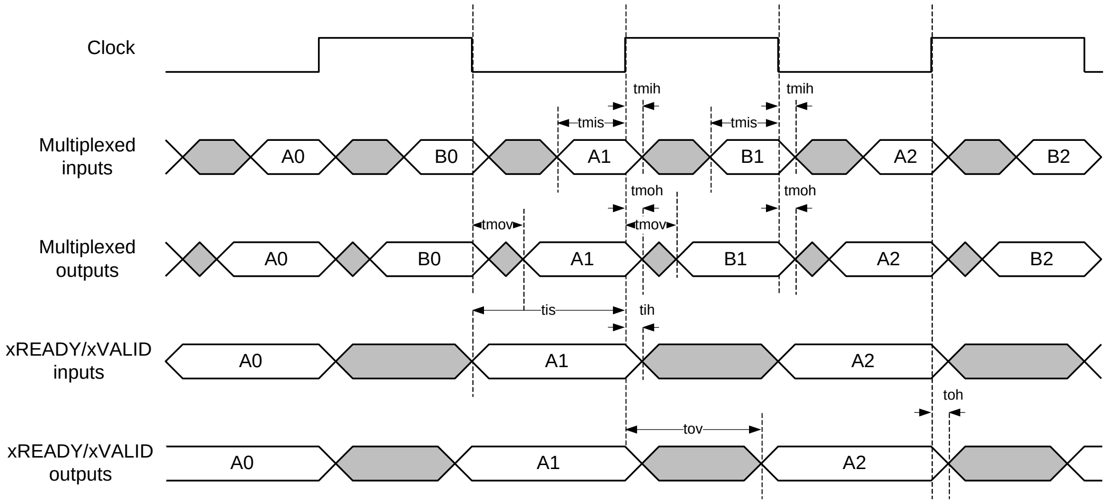


Figure 8-1 AXI timing requirements

Dir	Name	Description	Value
Inputs	tmis [ns]	Max multiplexed inputs setup to clock	1.1
	tmih [ns]	Max multiplexed inputs hold to clock	3.8
	tis [ns]	Max AWREADY, ARREADY, WREADY, RVALID, BVALID inputs setup to clock	12.1
	tih [ns]	Max AWREADY, ARREADY, WREADY, RVALID, BVALID input hold to clock	3.8
Outputs	tmov [ns]	Max clock to multiplexed data valid	6.7
	tmoh [ns]	Min clock to multiplexed data invalid	3.7
	tov [ns]	Max clock to WREADY, BREADY, AWVALID, ARVALID, WVALID outputs valid	17.7
	toh [ns]	Min clock to WREADY, BREADY, AWVALID, ARVALID, WVALID outputs invalid	3.7

Table 8-2 AXI Slave Input/Output timing to EMM_ACLK clock input

Dir	Name	Description	Value
Inputs	tmis [ns]	Max multiplexed inputs setup to clock	-1.1
	tmih [ns]	Max multiplexed inputs hold to clock	4.7
	tis [ns]	Max WREADY, BREADY, AWVALID, ARVALID, WVALID inputs setup to clock	13.9
	tih [ns]	Max WREADY, BREADY, AWVALID, ARVALID, WVALID input hold to clock	4.7
Outputs	tmov [ns]	Max clock to multiplexed data valid	11.5
	tmoh [ns]	Min clock to multiplexed data invalid	5.2
	tov [ns]	Max clock to AWREADY, ARREADY, WREADY, RVALID, BVALID outputs valid	26.5
	toh [ns]	Min clock to AWREADY, ARREADY, WREADY, RVALID, BVALID outputs invalid	5.2

Table 8-3 AXI Master Input/Output timing to EMS_ACLK clock output

8.3 MMB signals timing.

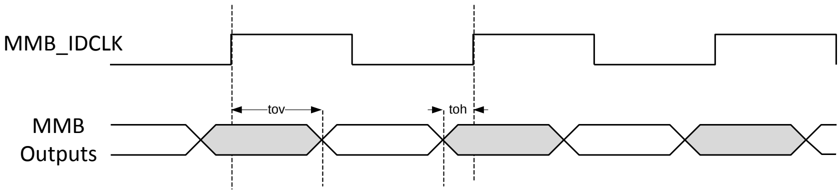


Figure 8-2 MMB Outputs Timing

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 8-4 MMB Output timing

8.4 SMB signals timing.

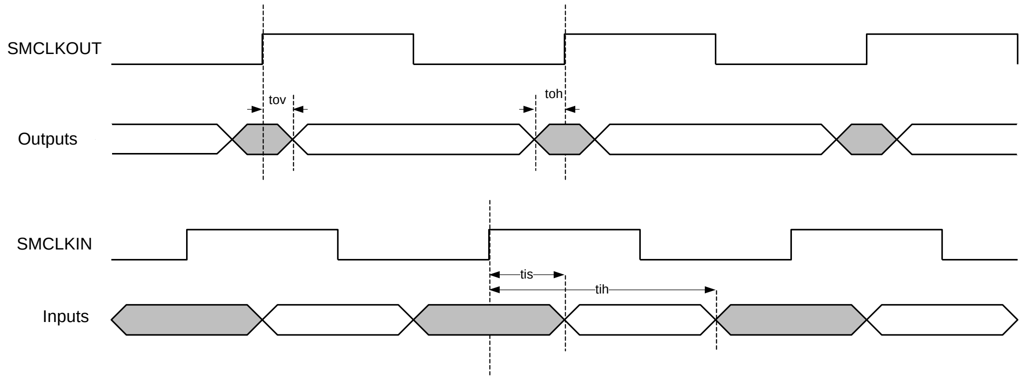


Figure 8-3 SMB Inputs and Outputs Timing

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 8-5 SMB Input/Output timing