Application Note 177

Using a CT1176JZF-S with the RealView™ Emulation Board

Document number: ARM DAI 0177A

Issued: April 2007

Application Note 177
Using a CT1176JZF-S with EB

Release information

The following changes have been made to this Application Note.

Change history

Date	Issue	Change
4th April 2007	A.05	First release

Proprietary notice

ARM, the ARM Powered logo, Thumb and StrongARM are registered trademarks of ARM Limited.
The ARM logo, AMBA, Angel, ARMulator, EmbeddedICE, ModelGen, Multi-ICE, CT1176JZF-S, TDMI and STRONG are trademarks of ARM Limited.
All other products, or services, mentioned herein may be trademarks of their respective owners.

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 covers the operation of the Emulation Baseboard (EB) with a CT1176JZF-S. It examines the contents of the baseboard FPGA, the system interconnect, the clock structure, and specifics of the programmer's model directly relevant to Core Tile operation.

After reading this Application Note the user should be in a position to make changes to the provided baseboard FPGA design, add their own AHB or AXI based peripherals to it, or debug and analyze the operation of the provided images.

1.2 EB and CT1176JZF-S Tile overview

This application note is designed for a CT1176JZF-S Tile fitted to Tile Site 1 on the EB as shown in **Figure 1-1 Core Tile and EB system**. This application note only works with CT1176JZF-S. The FPGA image for the baseboard is provided as part of the EB CD installation.

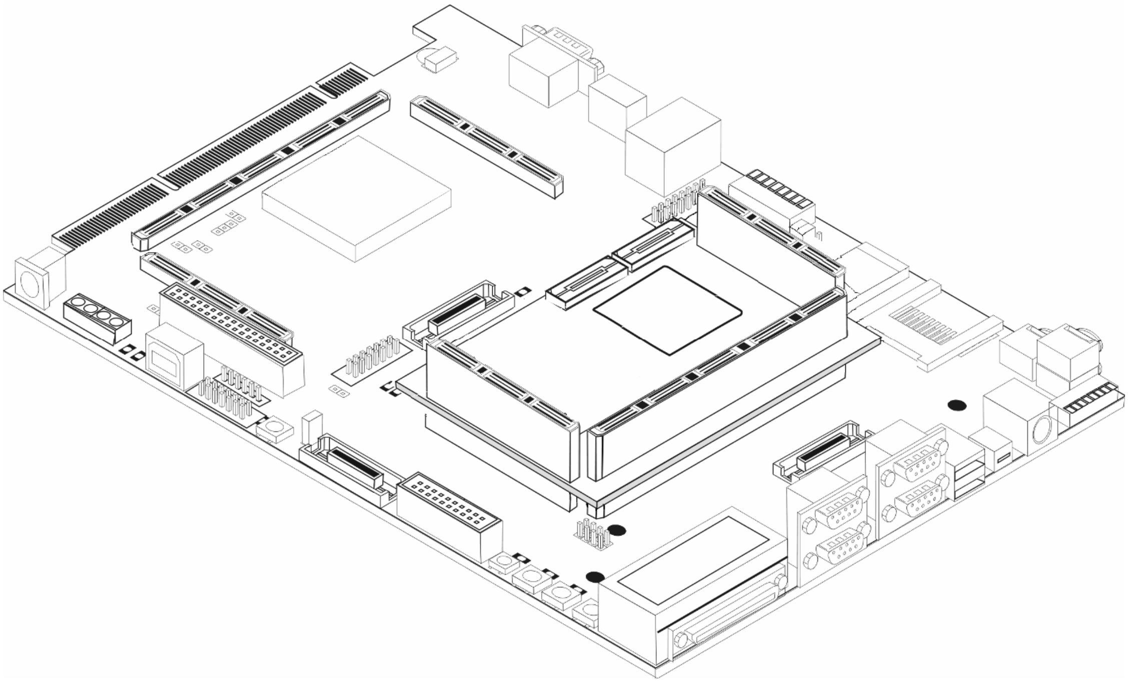


Figure 1-1 Core Tile and EB system

1.3 Optional Logic Tile

One or more logic tiles can be fitted to Tile Site 2. These Logic Tiles can contain both additional masters and slaves. See application note AN151 for an example and information about how to do this.

2 Getting started

Before you can use this application note, you will need to program the EB with the required FPGA image to enable the CT1176JZF-S to function correctly. Follow these steps to program the FPGA image.

1. Plug the CT1176JZF-S Tile onto TILE SITE 1 of the Emulation Baseboard.
2. Slide the CONFIG switch (S1) to the ON position.
3. Connect RVI to the Emulation Baseboard JTAG ICE connector (J18), or a USB cable to the USB Debug Port (J16).
4. Check the external supply voltage is +12V (positive on center pin, +/-10%, 35W), and connect it to the power connector (J28).
5. Power-up the boards. The '3V3 OK' LED and '5V OK' on the Emulation Baseboard should both be lit.
6. If using the USB connection, ensure that your PC has correctly identified an ARM® RealView™ ICE Micro Edition device is connected to the USB port. If the Windows operating system requires a USB driver to be installed please refer to the EB or PB926 \boardfiles\USB_Debug_driver\readme.txt.
7. If using RealView ICE (RVI), you must ensure that the RVI unit is powered and has completed its start-up sequence (check the LEDs on the front panel have stopped flashing).
8. You can now run the relevant 'progcards' utility for the connection you have prepared above.

- progcards_usb.exe for your USB connection
- progcards_rvi.exe for your RealView ICE connection

When using RVI select the target RVI box you are using.

9. Select the build of AN177 you would like to program, it may take several minutes to download. A successful configuration download will be terminated with a request to select an image for the CT1176JZF-S.
10. Select the option for CT1176JZF-S. The utility will report its progress, it may take several minutes to download. A successful configuration download will be terminated with the message "Programming Successful".
11. Power off the boards.
12. Set the configuration switches to load FPGA image 0. (S10 on the Emulation Baseboard set to all OFF).
13. If debug access to secure code is required, ensure user switches 7 and 8 are off.
14. Slide the CONFIG switch to the OFF position, and power on the boards. Ensure GLOBAL_DONE (D35) and the 'POWER' (D1) LEDs are lit. The Character LCD should show the Firmware and Hardware versions indicating that the Boot monitor firmware is running.
15. The system will now be fully configured and ready for use.

3 Architecture

This application note implements an AXI (AMBA 3.0) based system on the EB FPGA. The EB image exposes one master port and two slave ports (all 64 bit muxed AXI). There is one slave port routed to tile site1 and one slave and one master port routed to tile site 2.

Note that the direction of the arrows indicates the direction of control: it points from the Master to the Slave. An AXI bus contains signals going in both directions.

The PCI core is not included in the build as it is third party IP, it is an optional component.

3.1 System overview

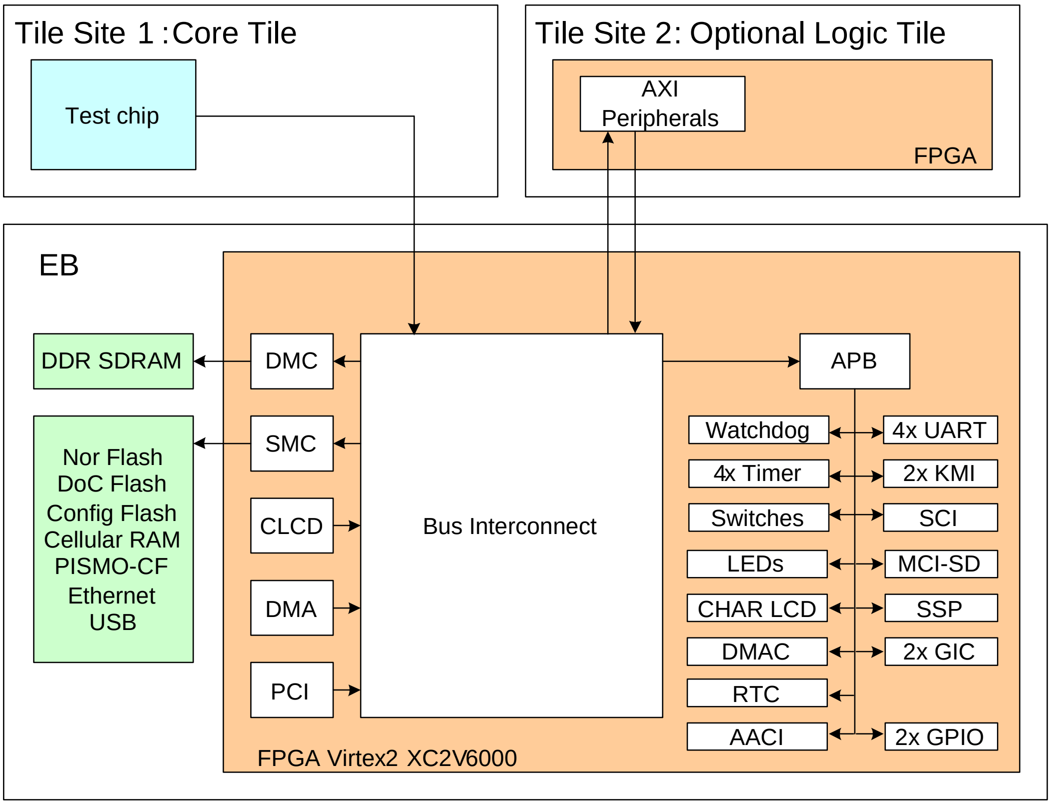


Figure 3-1 System interconnect

3.2 System architecture

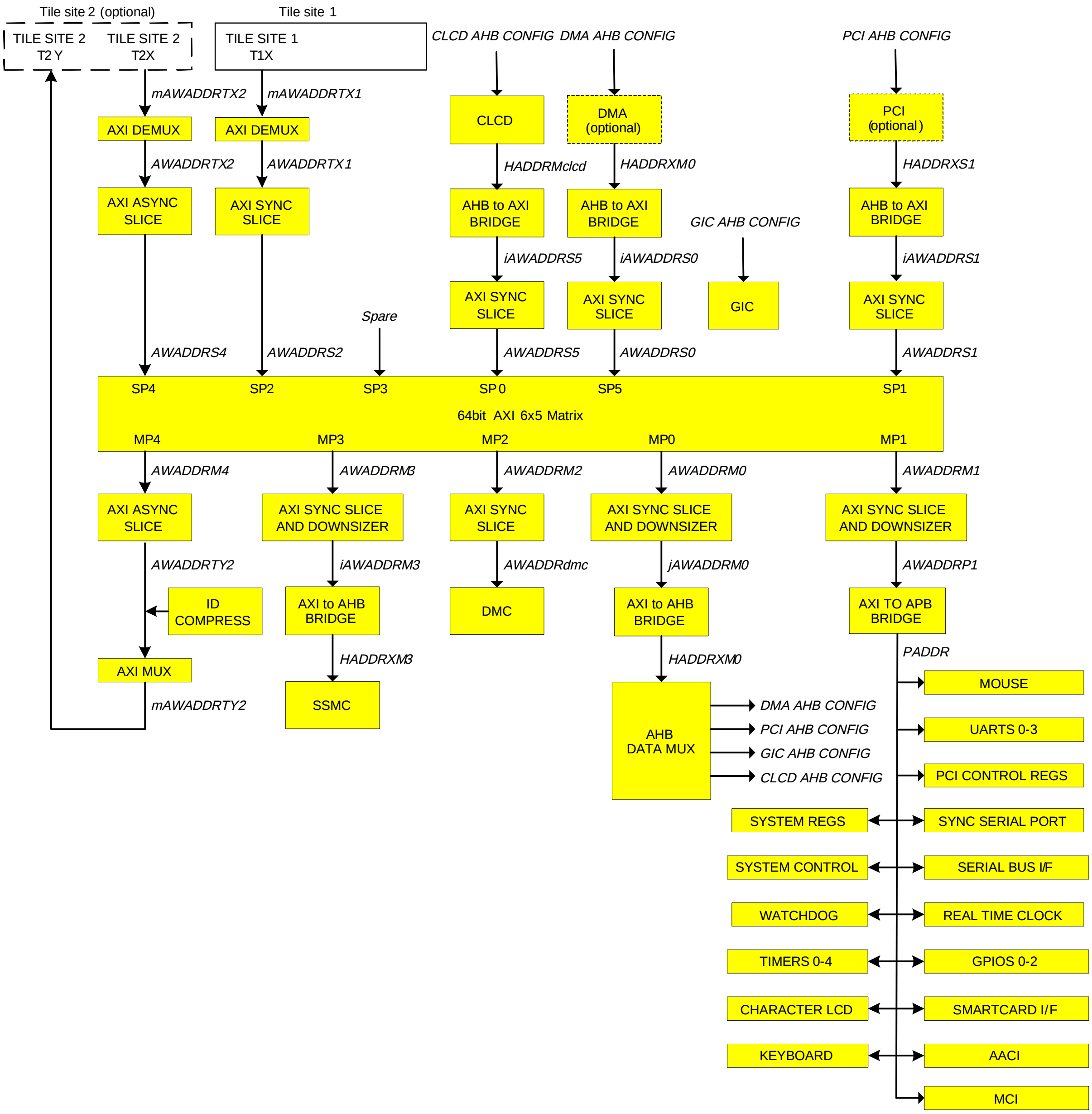


Figure 3-2 AN177 example design block diagram

3.2.1 EB Module functionality

The function of each of these blocks is as follows:

AXI SYNC SLICE

This is a synchronous register slice that is added to the incoming master ports S0, S1, S2, S3 and S5 on the bus matrix to ensure that data is available for a complete cycle prior to AXI matrix address decoding. It also helps to increase the operating frequency.

AXI ASYNC SLICE

This is an asynchronous bridge that is added to the incoming master port from tile site 2 to ensure that data is available for a complete cycle prior to AXI matrix address decoding and allows that site to operate in a different clock domain. It increases the operating frequency and introduces one clock cycle of latency.

AXI DEMUX

This block demultiplexes the multiplexed signals from the master ports on tile site 1 and 2. The muxing scheme is described in a section 4.3.

ID COMPRESS

This block merges the ID fields on the AXI bus to reduce the width of each ID channel. The ID is compressed using a simple algorithm to save pins.

AXI MUX

This block multiplexes the signals from the master port Mp4. The muxing scheme is described in a section **Error! Reference source not found.**.

64-bit AXI 6x5 Bus Matrix

This provides the bulk of the interconnect structure. It allows any of the 6 slave ports to connect to any of the 5 master ports without blocking the other masters. It also contains the decoder mapping to determine the address map, and a scheme to determine priority of competing masters to a single slave.

SSMC

This is a Synchronous Static Memory Controller. ARM PrimeCell PL093 is used in this design. For more information please refer to the PrimeCell documentation.

AHBtoAXI BRIDGE

This is a bridge component to change from an AHB bus to an AXI bus. Where nn is the input width and mm is the output width in decimal.

DMC

This is a Dynamic Memory Controller. ARM PrimeCell PL340 is used in this design. For more information please refer to the PrimeCell documentation. The default frequency for the DMC is 30MHz (OSCCLK1 divided by 4)

Since the Dynamic Memory Controller connects to high speed double data rate clocked devices it is necessary to make use of special pad and signal types to and from the Dynamic Memory. This block instantiated within the DMC

AXItoAPB BRIDGE

This is a bridge component to change from an AXI bus to an APB3 bus. This component contains decoding scheme for the bus, allowing 25 APB peripherals to be connected.

DMA

This is a direct memory access component. This design allows for the PrimeCell PL081 controllers to be added. It also allows for no DMA. For more information on the DMA blocks refer to the PrimeCell documentation.

PCI

Xilinx 32-bit PCI core operating at 33MHz. The RTL for the PCI core cannot be provided, so it is not available in FPGA images rebuilt by the user

CLCD

This is a color liquid crystal display controller. ARM PrimeCell PL111 is used in this design. For more information please refer to the PrimeCell documentation.

SYSTEM REGS

This contains a set of APB registers for hardware control of the EB. For a complete list of the functionality of these registers refer to the EB user guide.

SYSTEM CONTROL

This contains a generic set of system control registers. ARM ADK component SP810 is used in this design.

SERIAL BUS iI/F

This is a controller for the serial bus to the PISMO and Time of Year clock.

AACI

This is an advanced audio codec interface. ARM PrimeCell PL041 is used in this design. The FIFO is increased in size from the standard to better suit the FPGA operating environment (lower bus frequency). For more information please refer to the PrimeCell documentation. This block instantiates Xilinx block rams.

MCI

This is the multimedia card interface. ARM PrimeCell PL180 is used in this design. For more information please refer to the PrimeCell documentation.

MOUSE

These are keyboard and mouse interfaces. ARM PrimeCell PL050 is used in this design. For more information please refer to the PrimeCell documentation.

CHARACTER LCD

This is a character LCD display interface. It allows the system to communicate with the character LCD display fitted to the baseboard.

UART0-3

These are universal asynchronous receiver-transmitter interfaces (RS-232 serial). ARM PrimeCell PL010 is used in this design. For more information please refer to the PrimeCell documentation.

SSP

This is a synchronous serial port. ARM PrimeCell PL022 is used in this design. For more information please refer to the PrimeCell documentation. This block instantiates Xilinx block rams.

SCI

This is the smart card interface. ARM PrimeCell PL131 is used in this design. For more information please refer to the PrimeCell documentation.

WATCHDOG

This is a watchdog controller. It allows for the generation of an interrupt or reset after a defined time, to prevent against system lockup/failure. ARM ADK component SP805 is used in this design.

TIMERS 0-3

These are timer modules. ARM ADK component SP804 is used in this design.

GPIO 0-2

These are general purpose input/output modules. ARM PrimeCell PL061 is used in this design. For more information please refer to the PrimeCell documentation.

REAL TIME CLOCK

This is a real time clock module. Real time refers to total time from an event, and not actual real world time (measured using the Time Of Year Clock). ARM PrimeCell PL031 is used in this design. For more information please refer to the PrimeCell documentation.

PCI CONTROL REGS

This is the PCI control block. This contains a set of registers to allow configuration of the PCI system (if used). See the programmer's reference section for more information on the functionality of these registers.

3.3 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 User Guides for all the boards used in this configuration explain the clock options.

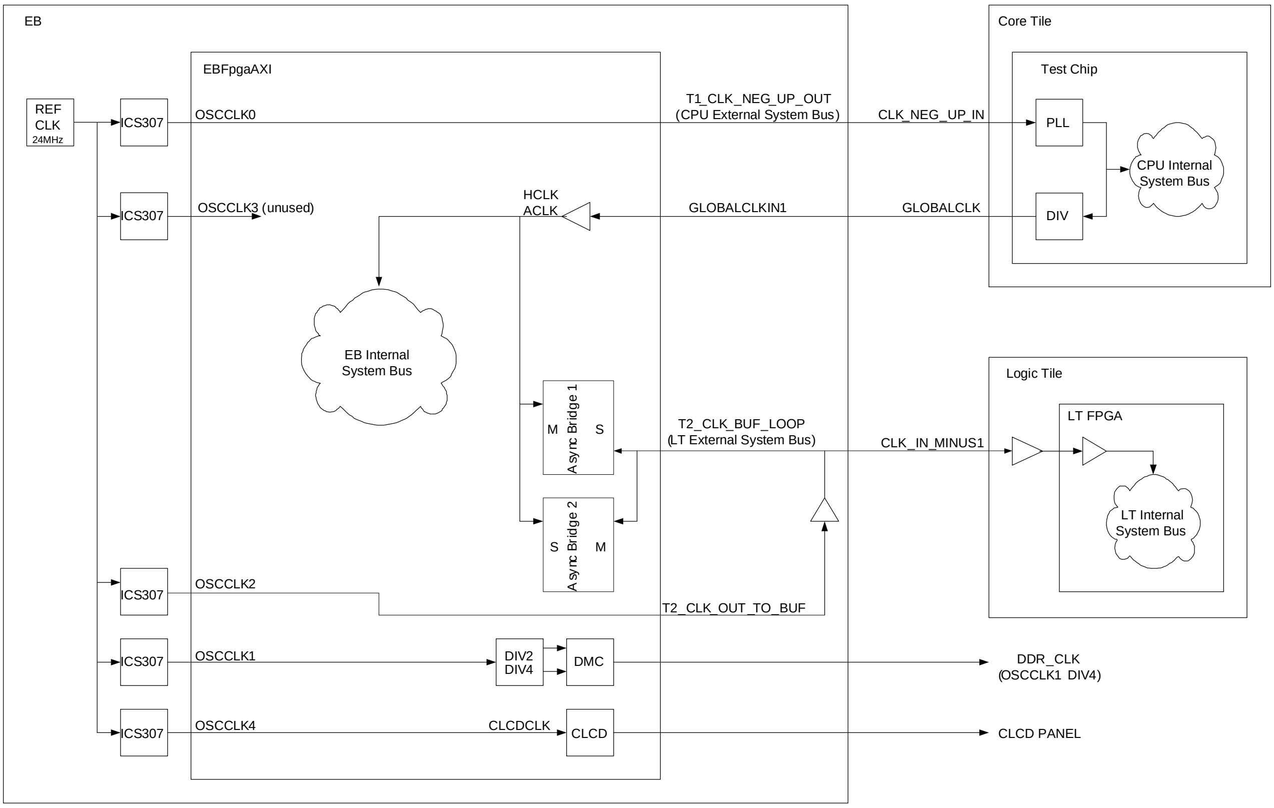


Figure 3-3 Clock architecture

There are 6 clock domains in this design:

3.3.1 EB Internal System Bus

The EB Internal System Bus is clocked by HCLK. HCLK is generated from GLOBALCLKIN1 through a clock buffer, HCLK is used to drive both AHB and AXI peripherals, ACLK = HCLK.

3.3.2 CPU External System Bus

By default, the CPU External System Bus is clocked at the same speed as the Core Tile input clock (T1_CLK_NEG_UP_OUT), which is directly connected to OSCCLK0. The Core Tile multiplies this frequency by 7 to drive the CPU Internal System Bus.

GLOBALCLKIN1 is generated from the Core Tile by dividing down the CPU Internal System Bus frequency by 7. This is used to clock data out of the Test Chip.

3.3.3 CPU Internal System Bus

The CPU Internal System Bus clock domain includes the ARM1176JZF-S CPU, the L220 Cache Controller and on-chip peripherals. The default behavior is for the PLL in the test chip to multiply the Core Tile input clock by 7 to drive this clock domain.

3.3.4 LT System Bus

The LT External System Bus is clocked by OSCCLK2.

The LT Internal System Bus in this diagram is the same frequency as the LT External System Bus. (LT Internal System Bus = LT External).

3.3.5 DDR_CLK

DDR_CLK is one quarter the frequency of OSCCLK1. OSCCLK1 is the source for all the DMC clocks, it is divided by 2 and 4 as required by the DMC.

3.3.6 CLCDCLK

CLCDCLK is directly connected to OSCCLK4.

3.3.7 Default operating frequencies

Clock	Use	Default Frequency
OSCCLK0	Tile site 1 CT1176JZF-S REFCLK	30MHz
OSCCLK1	DMC (DDR clock)	120MHz
OSCCLK2	Tile site 2 (Logic Tile)	20MHz
OSCCLK3	Unused	30MHz
OSCCLK4	CLCD	25MHz

Table 3-1 Default operating frequencies

This clock architecture has been chosen for this system for a number of reasons.

1. The test chip then generates the ACLK for the system based on its PLL configuration settings. The default setting are shown in section 3.13.
2. Tile site 2 can support a logic tile, the clock supplied to tile site 2 can be synchronous to tile site 1 by setting T2CLKSEL from the register block to b0 and select GLOBALCLKIN1 rather than OSCCLK2.

3.3.8 EB Clock Configuration Switches

Name	switch	Note
Clock selection	8[6:5]	See below for encoding

Table 3-2 Clock selection on EB

Boot switches SW8.6 and SW8.5 are used to select the bus clock to make it easy to set the clock for most customers. CPU internal, external and core clocks are generated from HCLK so are also affected.

SW8.6 – sets the default HCLK frequency

SW8.6	Clock Setting
OFF	EB Internal system bus (HCLK & ACLK) is 30MHz. CPU External system bus is 30MHz CPU Internal system bus is 120MHz CPU Core Clock is 360MHz
ON	EB Internal system bus (HCLK & ACLK) is 25MHz. CPU External system is 25MHz CPU Internal system bus is 100MHz CPU Core Clock is 300MHz

Table 3-3 Clock selection on EB

SW8.5 – sets the DMC clock frequency.

SW8.5	Clock Setting
OFF	DMC (DDR clock) 120MHz OSCCLK3 30MHz
ON	DMC (DDR clock) 100MHz OSCCLK3 25MHz

Table 3-4 Clock selection on EB

These switches are only sampled at the initial power up, and it is still possible to set any clock frequency in software using the EB SYS_OSCRESETx registers and the EB SYS_PLD_INIT and SYS_PLD_CTRL1 registers.

3.3.9 PL340 Dynamic memory clocking

The PL340 Dynamic Memory Controller requires three input clocks to drive the Dynamic Memory Interface. These are mclk, mclk2x, and fbclk_in. For complete descriptions of the functionality of these clocks refer to the PL340 PrimeCell documentation (ARM DUI 0267).

3.4 Trustzone

The ARM1176JFZ-S test chip is TrustZone enabled. AN177 exposes the configuration options required to allow debug of the ARM1176JFZ-S test chip. The signals required to implement more advanced configurations are available to designers.

The AXI bus connected through header X contains the ARPROT, AWPROT signals from the processor. These allow secure peripherals to be added to a base board if the correct modules are implemented.

The TRUSTZONENSA and TRUSTZONESD registers on the ARM1176JZF-S enable you to configure the security level of each of the on-chip peripherals. Please see **section 4.4 TrustZone security control - Core Tile for ARM1176JZF-S HBI-0154 User Guide**. TRUSTZONENSA-init and TRUSTZONESD-init controls are exposed through the serial read register and allow initial values to be set for the corresponding registers.

SPNIDEN is secure privileged non-invasive debug enable and is controlled by pin USERINX3. SPIDEN is the invasive form of secure debug enable and is controlled by pin USERINX2. Both these pins are passed up the Header Z bus and the through the PLD unchanged.

SPIDEN and SPNIDEN can be controlled by user switches 7 and 8.

SW6.7	SPIDEN - Secure privileged invasive debug enable
OFF	Asserted
ON	Not asserted

SW6.8	SPNIDEN - Secure privileged non-invasive debug enable
OFF	Asserted
ON	Not asserted

Two user LED’s have been connected to show TrustZone reads and writes on the tile site 1 AXI bus. LED 6 shows TrustZone write events and LED 7 shows TrustZone read events. The LED’s show the status of the last event seen indefinitely.

LED 6	TrustZone write events on tile site 1 X header
OFF	Last write access was not secure
ON	Last write access was secure

LED 7	TrustZone read events on tile site 1 X header
OFF	Last read access was not secure
ON	Last read access was secure

3.5 CT1176JZF-S reset structure

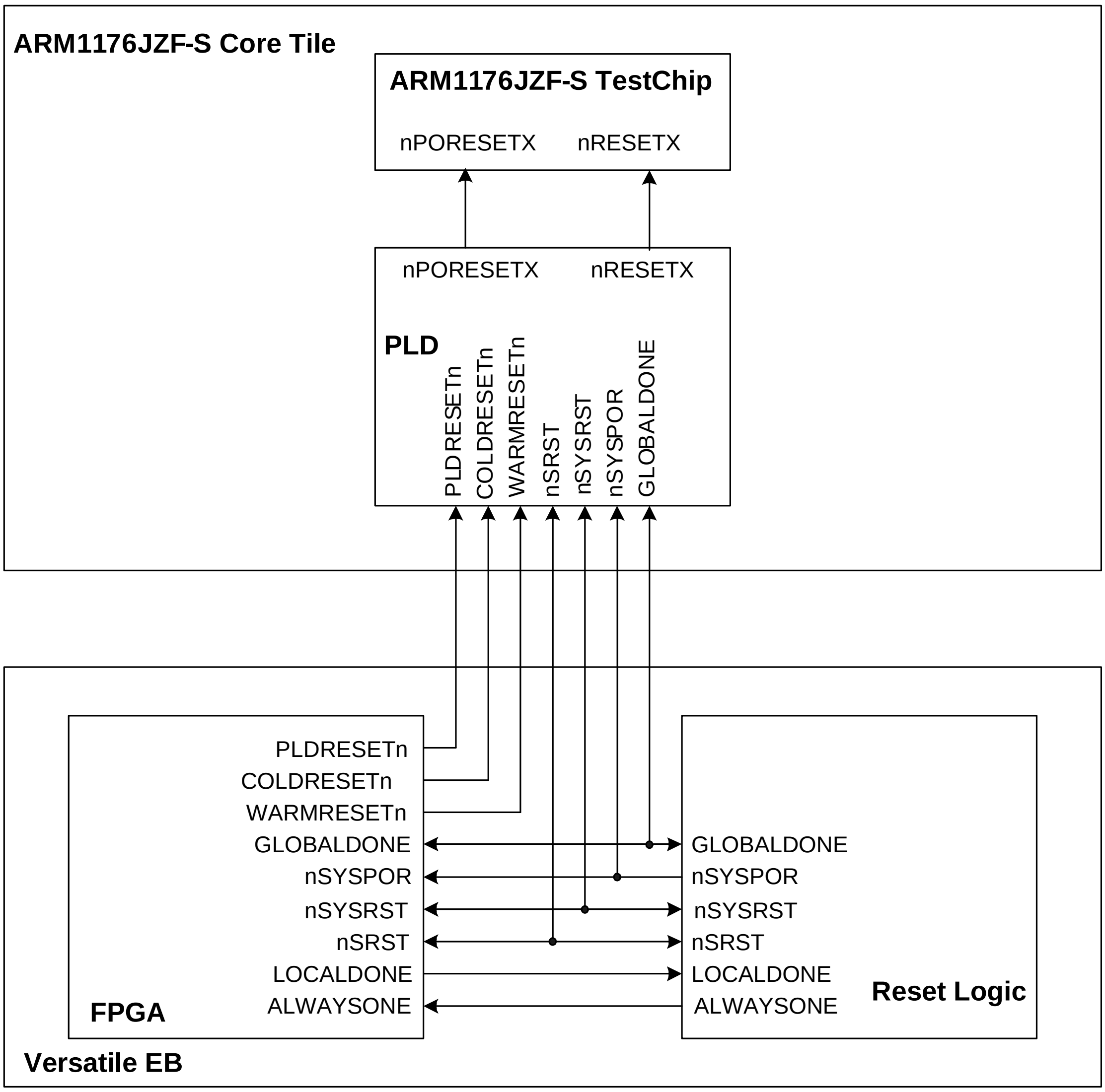


Figure 3-4 CT1176JZF-S reset routing

Name	Width (bits)	Direction to/from Reset Logic	Note
LOCALDONE	1	Output	Goes high when the EB FPGA has configured
GLOBAL_DONE	1	Input open drain	Goes high when all FPGAs have configured and the serial PLD interfaces have configured the TestChip PLL and Tile Mux/Switch
nSYSPOR	1	Output	Goes high approximately 7µs after GLOBAL_DONE is high
nSYSRST	1	Output	Goes high approximately 20µs after nSYSPOR goes high or goes low after nSRST goes low and returns high approximately 20us after nSRST goes high
PLDRESETn	1	Output	Resets and starts the PLD serial data transfer
nSRST	1	Input open drain	Generates an nSYSRST request
WARMRESETn	1	Output	Connected to nSYSRST in EB FPGA
COLDRESETn	1	Output	Connected to nSYSPOR in EB FPGA

Table 3-5 CT1176JZF-S Reset Signals

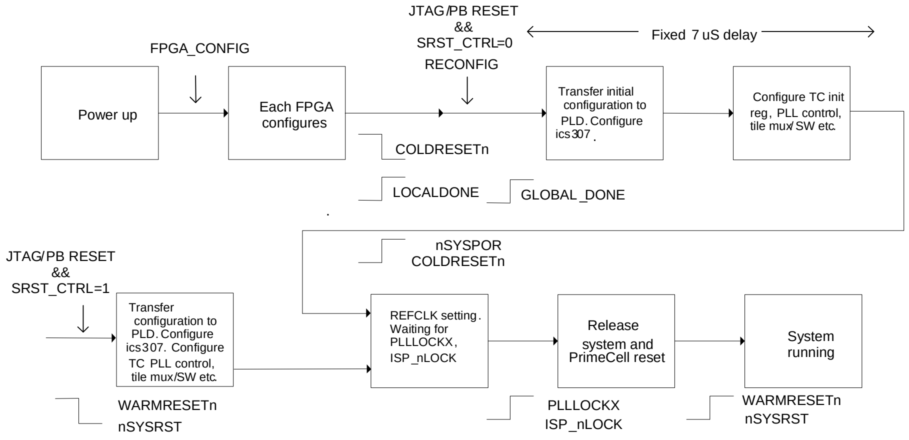

JTAG/PB RESET
&&
SRST_CTRL=0
Fixed 7 uS delay
FPGA_CONFIG
RECONFIG
Power up
Each FPGA configures
Transfer initial configuration to PLD. Configure ics307 .
Configure TC init reg, PLL control, tile mux/SW etc.
COLDRESETn
LOCALDONE
GLOBAL_DONE
JTAG/PB RESET
&&
SRST_CTRL=1
nSYSPOR
COLDRESETn
Transfer configuration to PLD. Configure ics307. Configure TC PLL control, tile mux/SW etc.
REFCLK setting. Waiting for PLLLOCKX, ISP_nLOCK
Release system and PrimeCell reset
System running
WARMRESETn
nSYSRST
PLLLOCKX
ISP_nLOCK
WARMRESETn
nSYSRST

Figure 3-5 Reset sequence

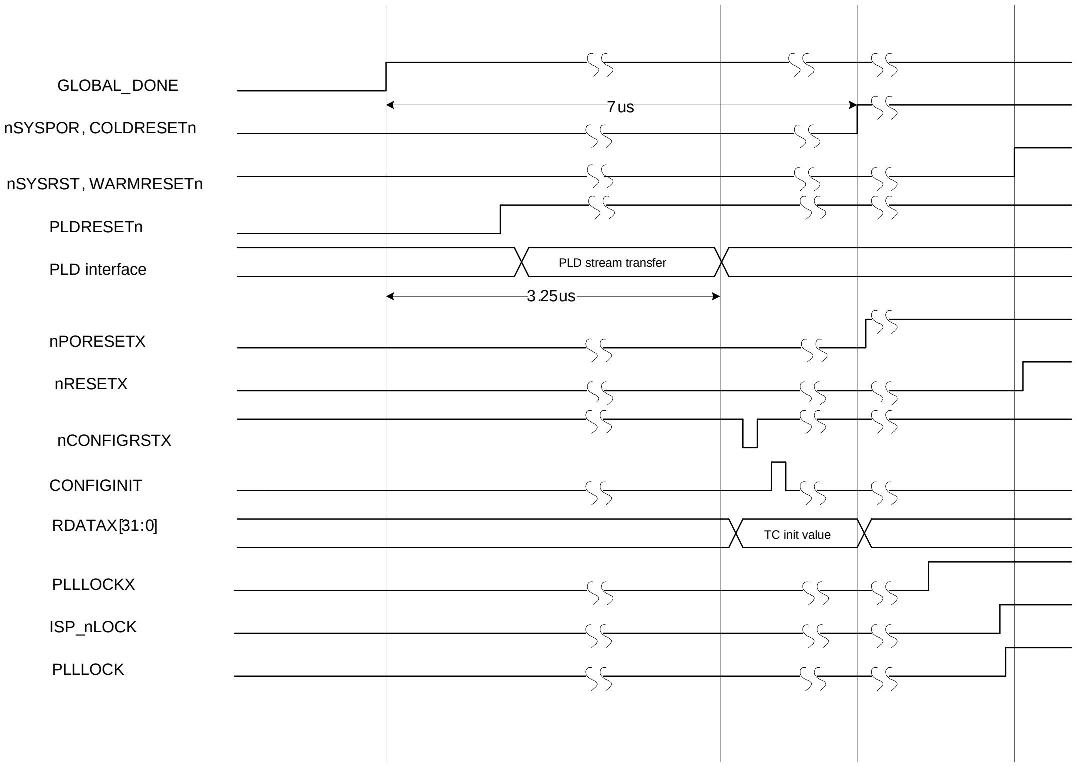


Figure 3-6 Power-on reset timing

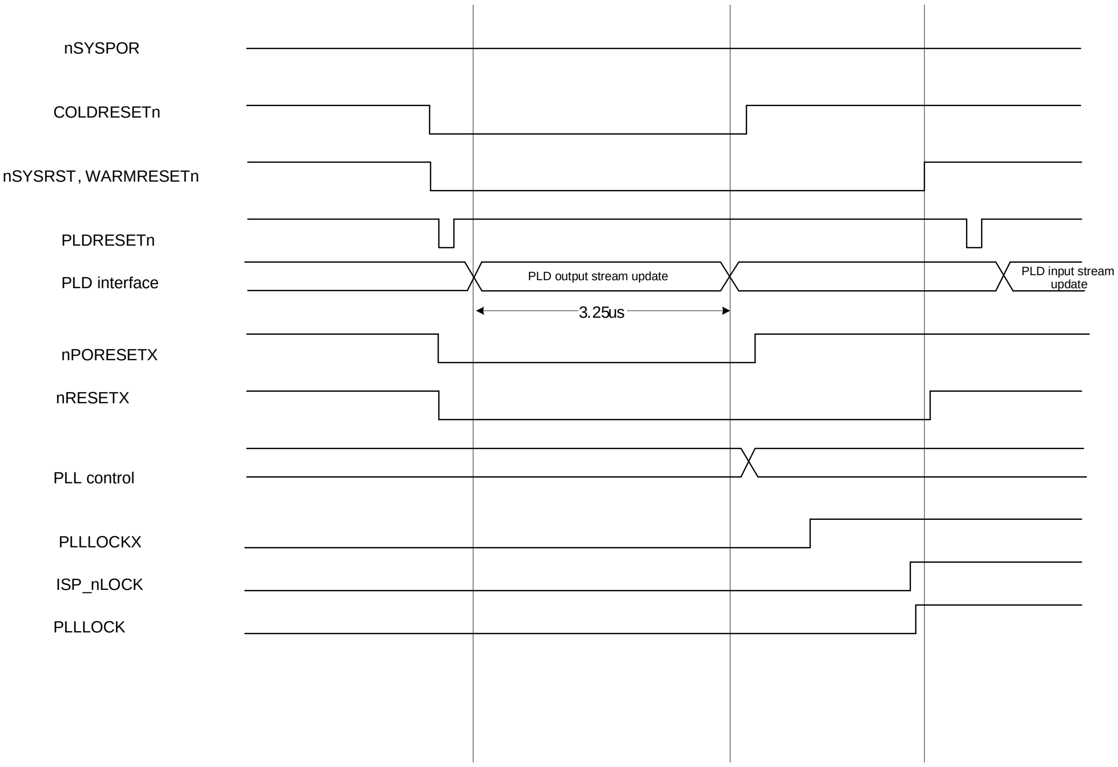


Figure 3-7 Reconfig timing (SRST_CTRL low)

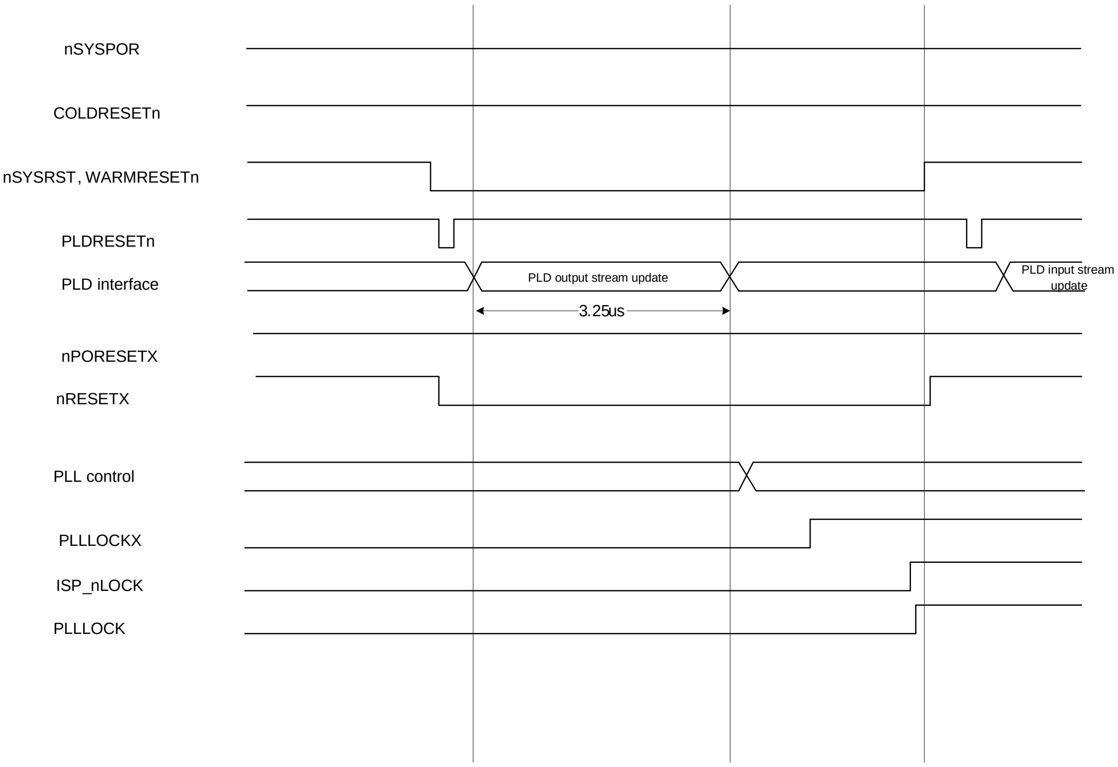


Figure 3-8 Reset timing (SRST_CTRL high)

3.6 Interrupt architecture

The interrupt scheme makes use of multiple Interrupt Controllers to facilitate the connection of an IRQ and FIQ to both tile sites, which would allow the connection of a second Core Tile on the EB tile site 2. Both tile sites can generate sources of interrupts and receive interrupts generated by the interrupt controllers.

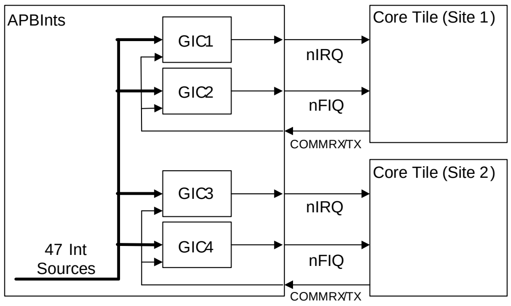


Figure 3-9 Interrupt architecture

A GIC (Generic Interrupt Controller) is chosen for this design, as it accepts a large number of interrupt sources without cascading interrupt controllers.

The majority of interrupts are common to all four GICs. The only exceptions are the COMMRX and COMMTX signals, which only connect to the interrupt controllers connected to the source of the COMMRX/TX signals.

For a mapping of the 47 interrupt sources onto the 64-bit GIC interrupt input refer to the Emulation Baseboard User Guide. The memory mapping of the GICs is also shown in the User Guide.

4 Hardware Description

4.1 EB Top Level (EBFpga.v)

The top level of the design is of particular importance for a number of reasons. This level defines the mapping from the HDRX, HDRY and HDRZ busses from the tile site into their functional allocations.

4.2 EB AXI Subsystem (EBFpgaCT1176.v)

This level connects all the components together and ties off static pins. This includes all the major blocks as shown in **Figure 3-1 System interconnect**.

4.3 EB AXI Multiplexing Scheme

By using a 2:1 multiplexer and latch scheme as shown below on **Figure 4-1 AXI multiplexing scheme** it is possible to reduce the pin count for the AXI bus into a realistic size for implementation on the only Tile X header.

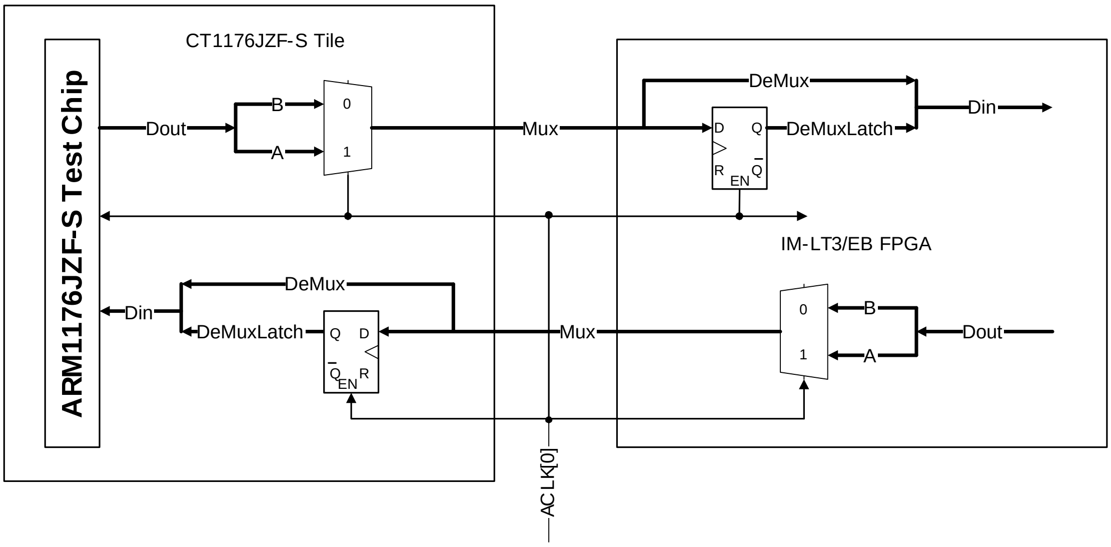


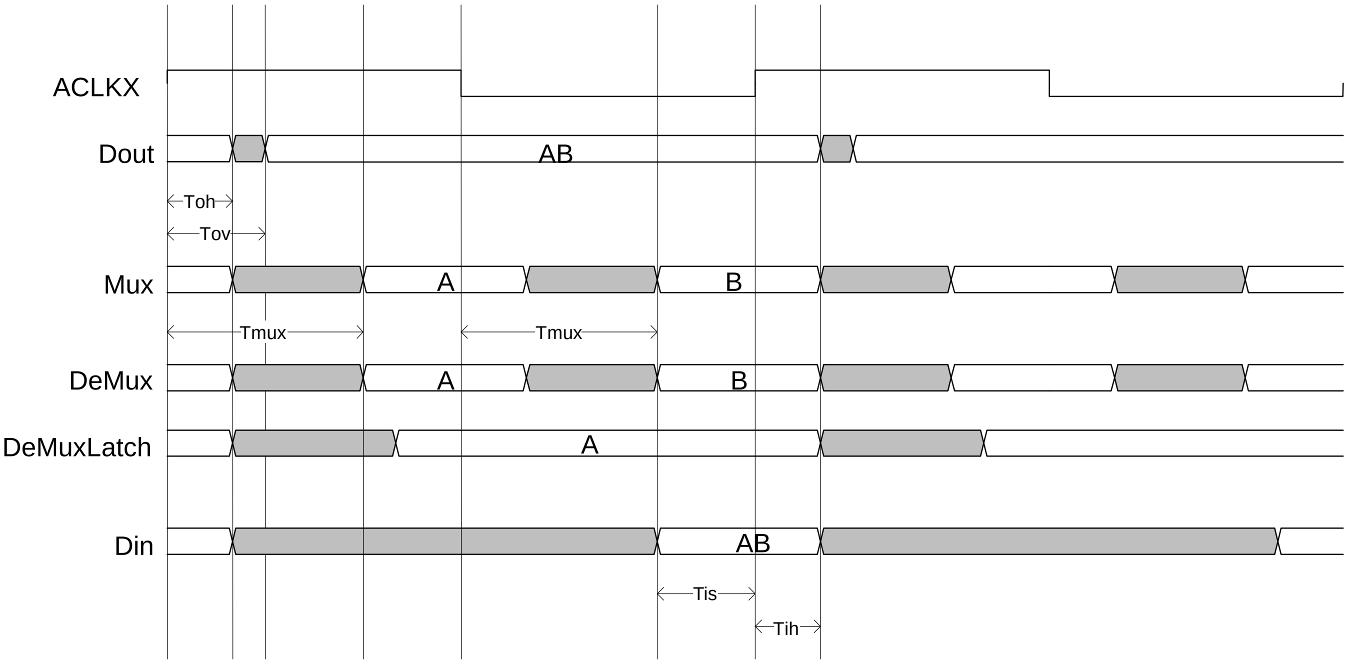
Figure 4-1 AXI multiplexing scheme

The output data is multiplexed on the level of CLK, to generate the multiplexed bus (X). The de-multiplexing is performed by latching the data (A) generated on the high level of CLK when CLK goes low (DeMuxLatch). The data (B), generated on the low side of CLK is passed straight through (DeMux). This design assumes data is always generated and captured on the rising edge of CLK.

The Valid and Ready signals on AXI can not be multiplexed in this way due to their timing requirements and must be passed directly between devices.

CT1176JZF-S implements the multiplexing and de-multiplexing with discrete logic on the board. The EB baseboard implements the multiplexing and de-multiplexing logic within the FPGA.

The following timing diagram (see **Figure 4-2 AXI timing requirements**) shows the data flow through the design with expected delays from multiplexing and demultiplexing standard components. The diagram does not include TC and FPGA timing.



Timing requirements ;

Toh	min	= 0ns	(output hold)
Tov	max	= 2ns	(output valid)
Tis	max	= 2ns	(Input setup)
Tih	max	= 0ns	(input hold)
Tmux	max	= 6 ns	(multiplexer and board delay)

The ACLKX is the clock driven from the CT1176JZF-S TC. All I/O timing must be with respect to this clock .

Figure 4-2 AXI timing requirements

4.4 CT1176JZF-S AXI multiplexing logic

Key to the performance of the bus multiplexing scheme is the switching logic showed on **Figure 4-3 CT1176JZF-S AXI multiplexing logic** which multiplexes and de-multiplexes the AXI bus on the X header. The X header logic consists of a 2:1 switch on the multiplexing side and a transparent latch on the de-multiplexing side. In addition care must be taken in the clock control for these devices to ensure maximum bandwidth.

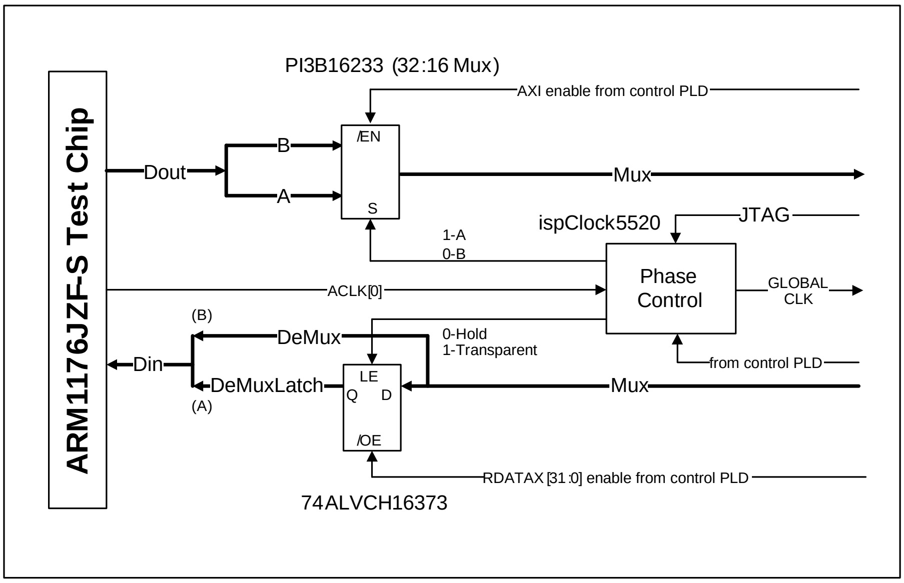


Figure 4-3 CT1176JZF-S AXI multiplexing logic

The AXI bus clock from the ARM1176JZF-S test chip is distributed to the PI2B16233 multiplexer, the 74ALVCH16373 latch and the FGPA on the board below (via GLOBAL_CLK or CLK_NEG_DN_OUT and CLKNEG_UP_OUT). Phase control of each of these signals is achieved with an ispClock5620 which minimises clock skew issues in the multiplexing logic. The ispClock5620 non-volatile configuration settings are JTAG configurable and allow four profiles to be stored. Profile selection and PLLBYPASSX is configured from the system control PLD on the CT1176JZF-S.

4.5 CT1176JZF-S Custom Header Z Signals

A number of signals from the ARM1176JZF-S test chip differ from the generic Tile header pin allocation. These signals will be connected to the Z headers along with serial PLD control signals as show on **Table 4-1 Z bus signals**.

Z Bus	HDRZ	CT1176JZF-S	Dir	Function
200	112	SPARE0	I/O	Not used. Connected to PLD
201	110	SPARE1	I/O	Not used. Connected to PLD
202	108	SPARE2	I/O	Not used. Connected to PLD
203	106	SPARE3	I/O	Not used. Connected to PLD
204	104	SPARE4	I/O	Not used. Connected to PLD
205	102	nFIQX	I	Testchip External fast interrupt – asynchronous. Taking the nFIQX input LOW generates The Fast Interrupt Request
206	100	nIRQX	I	Testchip External interrupt – asynchronous. LOW level on the nIRQX input causes the IRQ exception
207	98	DBGACKX	O	Testchip debug acknowledge signal. The processor asserts this output signal to indicate the system has entered Debug state.
208	96	STANDBYWFIX	O	Testchip signal indicating that processor is in Standby mode
209	94	BIGENDOUTX	O	Indicates big endian status
210	92	COMMRXX	O	Communications receive buffer not empty. This signal is only supported for functional (low-speed) validation.
211	90	COMMTXX	O	Communications transmit buffer not empty. This signal is only supported for functional (low-speed) validation.
212	88	EXTTRIG	O	Signal from the Trace Port connector.
213	86	DEWPTX	I	Testchip external watchpoint – synchronous. ARM1176JZF-S TC uses this input for test function only (otherwise drive to 0). Passed through PLD.
214	84	IEBKPTX	I	Testchip external breakpoint – synchronous. ARM1176JZF-S testchip uses this input for test function only (otherwise drive to 0). Passed through PLD.
215	82	EDBGRQX	I	Testchip external debug request – synchronous. When this signal is HIGH it causes the processor to enter debug state when execution of the current instruction has completed. Passed through PLD.
216	80	nCOLDRESET	I	Power-on reset to clock and reset control logic – asynchronous
217	78	USERINX.0	I	Additional testchip login input. Passed through PLD.
218	76	USERINX.1	I	Additional testchip login input. Passed through PLD.

Z Bus	HDRZ	CT1176JZF-S	Dir	Function
219	74	USERINX.2	I	SPIDEN. Passed through PLD.
220	72	USERINX.3	I	SPNIDEN. Passed through PLD.
221	70	ETMEXTINX	I	ETM test control input – synchronous. ETMEXTINX can be used for external trace triggering. Passed through PLD.
222	68	USEROUTX.0	O	Additional testchip login output. Passed through PLD.
223	66	USEROUTX.1	O	Additional testchip login output. Passed through PLD.
224	64	USEROUTX.2	O	Additional testchip login output. Passed through PLD.
225	62	USEROUTX.3	O	Additional testchip login output. Passed through PLD.
226	60	ETMEXTOUTX	O	Testchip ETM test control output – synchronous. ETMEXTOUTX can be used for external trace triggering. Passed through PLD.
227	58	PLDD1	I	Serial PLD interface data input
228	56	PLDD0	O	Serial PLD interface data output
229	54	PLDRESETn	I	Serial PLD interface reset
230	52	PLDCLK	I	Serial PLD interface clock
231	50	nWARMRESET	I	Asynchronous RESET input. Active LOW.

Table 4-1 Z bus signals

4.6 Header HDRX and HDRY AXI pin allocation

The three AXI buses (T1X, T2X, T2Y) connect to the HDRX and HDRY Tile headers. The pin connections are the same for all four buses. T1X is shown as an example on **Table 4-2 Header HDRX and HDRY AXI pin allocation**.

X/Y Bus	HDRX	HDRY	signal	X/Y Bus	HDRX	HDRY	signal
0	180	179	WDATA0/32	72	36	35	BID1/3
1	178	177	WDATA1/33	73	34	33	BID4/BID5
2	176	175	WDATA2/34	74	32	31	BRESP0/1
3	174	173	WDATA3/35	75	30	29	BVALID
4	172	171	WDATA4/36	76	28	27	BREADY
5	170	169	WDATA5/37	77	26	25	ARADDR0/16
6	168	167	WDATA6/38	78	24	23	ARADDR1/17
7	166	165	WDATA7/39	79	22	21	ARADDR2/18
8	164	163	WDATA8/40	80	20	19	ARADDR3/19
9	162	161	WDATA9/41	81	18	17	ARADDR4/20
10	160	159	WDATA10/42	82	16	15	ARADDR5/21
11	158	157	WDATA11/43	83	14	13	ARADDR6/22
12	156	155	WDATA12/44	84	12	11	ARADDR7/23
13	154	153	WDATA13/45	85	10	9	ARADDR8/24
14	152	151	WDATA14/46	86	8	7	ARADDR9/25
15	150	149	WDATA15/47	87	6	5	ARADDR10/26
16	148	147	WDATA16/48	88	4	3	ARADDR11/27
17	146	145	WDATA17/49	89	2	1	ARADDR12/28
18	144	143	WDATA18/50	90	1	2	ARADDR13/29
19	142	141	WDATA19/51	91	3	4	ARADDR14/30
20	140	139	WDATA20/52	92	5	6	ARADDR15/31
21	138	137	WDATA21/53	93	7	8	ARID0/2
22	136	135	WDATA22/54	94	9	10	ARID1/3
23	134	133	WDATA23/55	95	11	12	ARLEN0/2
24	132	131	WDATA24/56	96	13	14	ARLEN1/3
25	130	129	WDATA25/57	97	15	16	ARSIZE0/1
26	128	127	WDATA26/58	98	17	18	ARID4/ARPROT2
27	126	125	WDATA27/59	99	19	20	ARPROT0/1
28	124	123	WDATA28/60	100	21	22	ARBURST0/1
29	122	121	WDATA29/61	101	23	24	ARLOCK0/1
30	120	119	WDATA30/62	102	25	26	ARCACHE0/2
31	118	117	WDATA31/63	103	27	28	ARCACHE1/3
32	116	115	WID0/2	104	29	30	ARVALID/ARID5
33	114	113	WID1/3	105	31	32	ARREADY
34	112	111	WSTRB0/4	106	33	34	RDATA0/32
35	110	109	WSTRB1/5	107	35	36	RDATA1/33
36	108	107	WSTRB2/6	108	37	38	RDATA2/34
37	106	105	WSTRB3/7	109	39	40	RDATA3/35
38	104	103	WLAST/WID4	110	41	42	RDATA4/36
39	102	101	WVALID/WID5	111	43	44	RDATA5/37
40	100	99	WREADY	112	45	46	RDATA6/38
41	98	97	AWADDR0/16	113	47	48	RDATA7/39
42	96	95	AWADDR1/17	114	49	50	RDATA8/40
43	94	93	AWADDR2/18	115	51	52	RDATA9/41
44	92	91	AWADDR3/19	116	53	54	RDATA10/42
45	90	89	AWADDR4/20	117	55	56	RDATA11/43

46	88	87	AWADDR5/21	118	57	58	RDATA12/44
47	86	85	AWADDR6/22	119	59	60	RDATA13/45
48	84	83	AWADDR7/23	120	61	62	RDATA14/46
49	82	81	AWADDR8/24	121	63	64	RDATA15/47
50	80	79	AWADDR9/25	122	65	66	RDATA16/48
51	78	77	AWADDR10/26	123	67	68	RDATA17/49
52	76	75	AWADDR11/27	124	69	70	RDATA18/50
53	74	73	AWADDR12/28	125	71	72	RDATA19/51
54	72	71	AWADDR13/29	126	73	74	RDATA20/52
55	70	69	AWADDR14/30	127	75	76	RDATA21/53
56	68	67	AWADDR15/31	128	77	78	RDATA22/54
57	66	65	AWID0/2	129	79	80	RDATA23/55
58	64	63	AWID1/3	130	81	82	RDATA24/56
59	62	61	AWLEN0/2	131	83	84	RDATA25/57
60	60	59	AWLEN1/3	132	85	86	RDATA26/58
61	58	57	AWSIZE0/1	133	87	88	RDATA27/59
62	56	55	AWID4/AWPROT2	134	89	90	RDATA28/60
63	54	53	ARM_nRESET	135	91	92	RDATA29/61
64	52	51	AWPROT0/1	136	93	94	RDATA30/62
65	50	49	AWBURST0/1	137	95	96	RDATA31/63
66	48	47	AWLOCK0/1	138	97	98	RID0/2
67	46	45	AWCACHE0/2	139	99	100	RID1/3
68	44	43	AWCACHE1/3	140	101	102	RRESP0/1
69	42	41	AWVALID/AWID5	141	103	104	RLAST/RID4
70	40	39	AWREADY	142	105	106	RVALID/RID5
71	38	37	BID0/2	143	107	108	RREADY

Table 4-2 Header HDRX and HDRY AXI pin allocation

Some pins going to HDRX on tile site 1 have a different signal description.

T1X Pin No.	HDRX	Signal
38	104	WLASTX/1’B0
39	102	WVALIDX/1’B0
62	56	1’B0/AWPROT2
63	54	NC
69	42	AWVALID/1’B0
73	34	NC/NC
98	17	1’B0/ARPROT2
104	29	ARVALID/1’B0
141	103	RLAST/NC
142	105	RVALID/NC

Table 4-3 - Differences to signal names for HRDX on tile site 1

CT1176JZF-S configuration PLD

The CT1176JZF-S Tile contains a configuration PLD (see **Figure 4-4 CT1176JZF-S configuration PLD**) which defines:

- ARM1176JZF-S testchip initialization signals
- ARM1176JZF-S testchip resets
- 2:1 AXI multiplexing and demultiplexing logic enable
- status of the phase control logic
- DAC and ADC control.

A serial data stream from the FPGA on the board below is used to configure the PLD settings after power up.

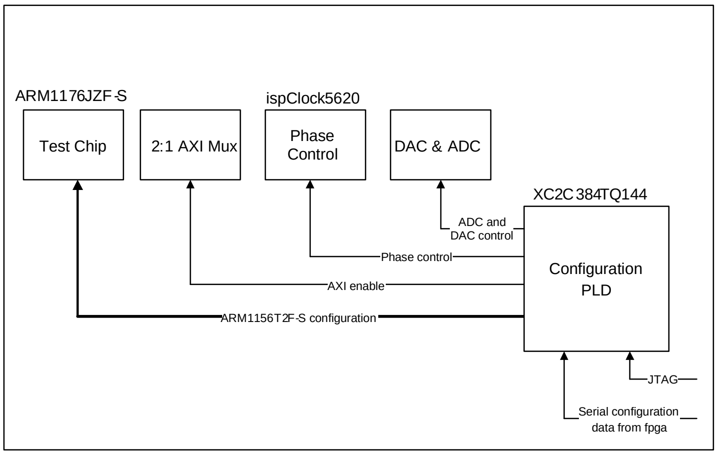


Figure 4-4 CT1176JZF-S configuration PLD

4.7 CT1176JZF-S configuration PLD serial interface

Serial control of the configuration PLD is achieved through a three wire interface (see **Figure 4-5 CT1176JZF-S configuration PLD serial interface**). Reset signal PLDRESETn is also used to ensure the PLD logic is in the defined state before the data stream is loaded.

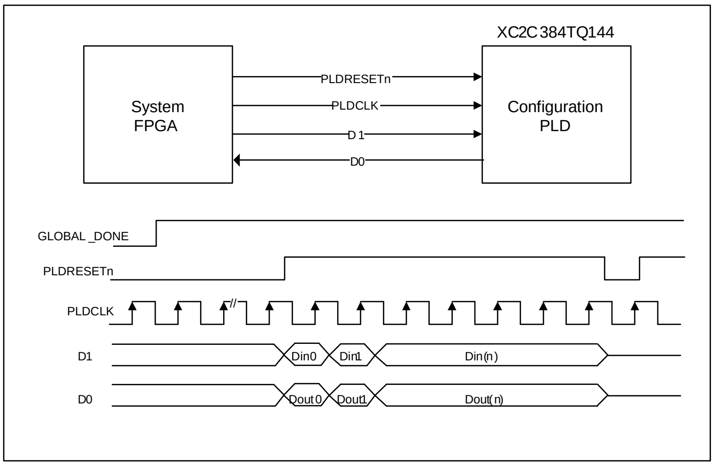


Figure 4-5 CT1176JZF-S configuration PLD serial interface

All data bits are clocked into the PLD on the rising edge of PLDCLK. The serial stream is constantly updating ensuring both FPGA and PLD registers are identical.

4.8 Serial write data register

The system FPGA writes a 73 bit value to the data register in the configuration PLD. The write data register is mapped on to the external pins of the configuration PLD with the signals as shown in **Table 4-4 Serial write data register** below. ACLKDIV[0] is the first piece of data transmitted, the others follow in the order shown.

Serial Bits	Pin name	Default value	Definition
3	ACLKDIV	b000	Ratio of AXI clock to main clock
4	PLLREFDIV	0x3	TC PLL reference divider. Does not operate as name would suggest. See CT1176EJZ-S user guide for more details.
12	PLLFBDIV	0x090	TC PLL feedback divider. Does not operate as name would suggest. See CT1176EJZ-S user guide for more details.
1	PLLBYPASS	b0	TC PLL bypass select
4	PLLOUTDIV	0x1	TC PLL output divider. Does not operate as name would suggest. See CT1176EJZ-S user guide for more details.
2	PLLCTRL	b00	TC PLL control – Not used
1	REMAP-init	b0	Clock generation control initial value
7	BLKDISABL -init[7:1]	b0000000	Clock generation control initial value
6	DivCore-init[5:0]	b000000	Clock generation control initial value
6	DivInt-init[5:0]	b000010	Clock generation control initial value
6	DivExt-init[5:0]	b000011	Clock generation control initial value
1	SYNCMODEREQI /RW-init	b0	SYNCMODEREQI/RW initial value
1	SYNCMODEREQ D/P-init	b0	nSYNCMODEREQP initial value
1	AMBAClkSource -init	b0	AMBAClkSource initial value
1	TRUSTZONESD -init	b0	TRUSTZONESD initial value
1	TRUSTZONENSA -init	b1	TRUSTZONENSA initial value
1	AXIrandom-init	b0	Ports wait enable
1	UBITINIT	b0	Initial value of ARM1176 UBIT

Serial Bits	Pin name	Default value	Definition
1	VINITHI	b0	Location of exception vectors at reset
1	INITRAM	b0	Internal ram enabled at reset
1	BIGENDIN	b0	Selects big endian
1	DBGEN	b1	Debug enable – synchronous. When this signal is LOW the processor behaves as if in debug disabled mode.
8	DACDAT[7:0]	0x80	DAC data
1	DACSEL	b0	DAC select
1	CLKSEL	b0	Clock select
73	Total		

Table 4-4 Serial write data register

4.9 Serial read data register

The system FPGA reads a 30 bit value from the data register in the configuration PLD. The read data register is mapped from the external pins of the configuration PLD with the following signals, as shown in **Table 4-5 PLD Read Register** below.

Serial Bits	Pin name	Definition
1	PLLLOCK	TC and ispClock5620 In-lock indication from PLL
1	ARM_PLLLOCKX	TC In-lock indication from PLL
1	ISPnLOCK	ispClock5620 In-lock indication from PLL
12	ADCDAT0[11:0]	Power sensing 12 bit ADC value (0 to 7 selected by ADCSEL)
3	ADCSEL[2:0]	ADC channels currently being converted (0-7)
4	PLDVER[3:0]	PLD build version
8	BUILDID	Build variant
30	Total	

Table 4-5 PLD Read Register

4.10 EB CT1176JZF-S specific registers

The CT register base is 0x10000000. All registers must be unlocked by writing 0xA05F to base+0x20 first.

The SYS_PLD_INIT register at base+0x7C on EB board loads the ARM1176JZF-S init register ONLY after reset.

SYS_PLD_INIT	AXIrandom:TRUSTZONENSA:TRUSTZONESD:AMBASource:SYNCMODEREQ D/P:SYNCMODEREQI/RW:DivExt:DivInt:DivCore:BLKDISABLE:REMAP
Direction	W:W:W:W:W:W:WWWWWW:WWWWWW:WWWWWW:WWWWWWW:W
Default	0:1:0:0:0:0:000011:000010:000000:0000000:0

Table 4-6 SYS_PLD_INIT Register

The SYS_SET_VOLTAGE0 register at base+0xA0 on EB board sets the VDDCORE (TC, ETM, ARM, PLL10) voltage and reads the VDDCORE voltage.

SYS_SET_VOLTAGE0	000000000000:ADC_DATA0[11:0] (VDDCORE voltage):DAC_DATAA[7:0]
Direction	RRRRRRRRRRRR:RRRRRRRRRRRR:WWWWWWWW
Default	000000000000:XXXXXXXXXXXX:10000000

Table 4-7 SYS_SET_VOLTAGE0 register

The SYS_SET_VOLTAGE1 register at base+0xA4 on EB board sets the PLDVDD25 voltage and reads the PLDVDD25 voltage.

SYS_SET_VOLTAGE1	000000000000:ADC_DATA1[11:0](PLDVDD25 voltage):DAC_DATAB[7:0]
Direction	RRRRRRRRRRRR:RRRRRRRRRRRR:WWWWWWWW
Default	000000000000:XXXXXXXXXXXX:10000000

Table 4-8 SYS_SET_VOLTAGE1 register

The SYS_READ_CURRENT0 register at base+0xA8 on EB board reads the VDDCORE_TC and VDDCORE_ETM current.

SYS_READ_CURRENT0	00000000:ADC_DATA3[11:0](VDDCORE_ETM current): ADC_DATA2[11:0](VDDCORE_TC current)
Direction	RRRRRRRR:RRRRRRRRRRRR:RRRRRRRRRRRR
Default	00000000:XXXXXXXXXXXX:XXXXXXXXXXXX

Table 4-9 SYS_READ_CURRENT0 register

The SYS_READ_CURRENT1 register at base+0xAC on EB board reads the VDDCORE_ARM current.

SYS_READ_CURRENT1	00000000:ADC_DATA5[11:0](ARM_VDDIO voltage): ADC_DATA4[11:0](VDDCORE_ARM current)
Direction	RRRRRRRR:RRRRRRRRRRRR:RRRRRRRRRRRR
Default	00000000:XXXXXXXXXXXX:XXXXXXXXXXXX

Table 4-10 SYS_READ_CURRENT1register

The SYS_READ_CURRENT2 register at base+0xB0 on EB board reads PLLVDD25 the PLLVDD10 currents.

SYS_READ_CURRENT2	00000000:ADC_DATA7[11:0](PLLVDD10 current): ADC_DATA6[11:0](PLLVDD25 current)
Direction	RRRRRRRR:RRRRRRRRRRRR:RRRRRRRRRRRR
Default	00000000:XXXXXXXXXXXX:XXXXXXXXXXXX

Table 4-11 SYS_READ_CURRENT2 register

The SYS_PLD_CTRL1 register at base+0x74 on EB board connects to the following serial register.

SYS_PLD_CTRL1	BIGENDIN:INITRAM:VINITHI:UBITINIT:CLKSEL:PLLCTRL:PLLOUTDIV:PLLBYPASS: PLLFBDIV:PLLREFDIV:0:ACLKDIV
Direction	W:W:W:W:W:WW:WWWW:W:WWWWWWWWWWWW:WWWW:R:WWW
Default	0:0:0:0:0:00:0001:0:000010010000:0011:0:000

Table 4-12 SYS_PLD_CTRL1 register

The SYS_PLD_CTRL2 register at base+0x78 on EB board connects to the following input signals.

SYS_PLD_CTRL2	0000000:SRST_CTRL:0000:nUSERSW[7:6]:USERIN[1:0]:0000: USEROUT:COMMTX:COMMRX:BIGENDOUTX:STANDBYWFI:PLDVER
Direction	RRRRRRR:W:RRRR:RR:WW:RRRR:RRRR:R:R:R:R:RRRR
Default	0000000:1:0000:00:00:0000:XXXX:X:X:X:X:XXXX

Table 4-13 SYS_PLD_CTRL2 register

5 Programmer's Model

5.1 CT1176JZF-S boot up operation overview

Please refer to the ARM1176JZF-S Processor TRM for more information on the use of CP15 registers and exact operation.

5.2 ARM1176JZF-S TC Memory Map

Please refer to the CT1176JZF-S User Guide for more ARM1176JZF-S Processor memory map information.

5.3 EB Memory Map

The CT1176JZF-S on EB example design provides a software interface with the majority of features required for a system. Refer to the EB user guide for more information

	Memory range		**Bus type**	**Memory region size**
Peripheral	**Lower limit**	**Upper limit**		
Dynamic Memory	0x00000000	0x0FFFFFFF	AHB/AXI	256M
System Registers	0x10000000	0x10000FFF	APB	4K
System Controller (SP810)	0x10001000	0x10001FFF	APB	4K
I2C control	0x10002000	0x10002FFF	APB	4K
Reserved	0x10003000	0x10003FFF	APB	4K
AACI	0x10004000	0x10004FFF	APB	4K
MCI0	0x10005000	0x10005FFF	APB	4K
KMI0	0x10006000	0x10006FFF	APB	4K
KMI1	0x10007000	0x10007FFF	APB	4K
Character LCD	0x10008000	0x10008FFF	APB	4K
UART0	0x10009000	0x10009FFF	APB	4K
UART1	0x1000A000	0x1000AFFF	APB	4K
UART2	0x1000B000	0x1000BFFF	APB	4K
UART3	0x1000C000	0x1000CFFF	APB	4K
SSP0	0x1000D000	0x1000DFFF	APB	4K
SCI0	0x1000E000	0x1000EFFF	APB	4K
Reserved	*0x1000F000*	*0x1000FFFF*	*APB*	*4K*
Watchdog	0x10010000	0x10010FFF	APB	4K
Timer 0&1	0x10011000	0x10011FFF	APB	4K
Timer 2&3	0x10012000	0x10012FFF	APB	4K
GPIO 0	0x10013000	0x10013FFF	APB	4K
GPIO 1	0x10014000	0x10014FFF	APB	4K
GPIO 2 (Misc onboard I/O)	0x10015000	0x10015FFF	APB	4K
Reserved	*0x10016000*	*0x10016FFF*	*APB*	*4K*
RTC	0x10017000	0x10017FFF	APB	4K
DMC configuration	0x10018000	0x10018FFF	APB	4K
PCI configuration	0x10019000	0x10019FFF	AHB	4K

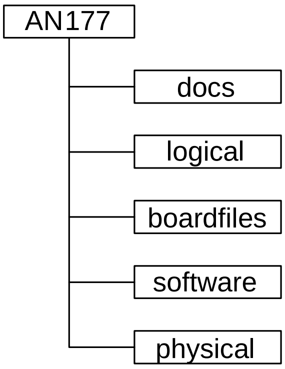
Peripheral	Lower limit	Upper limit	Bus type	Memory region size
Reserved for future use	0x1001A000	0x1001FFFF	APB	28K (4K * 6)
CLCD configuration	0x10020000	0x1002FFFF	AHB	64K (Note CLCD can only access Dynamic memory)
DMAC configuration	0x10030000	0x1003FFFF	AHB	64K
GIC1 (nIRQ IC) Tile Site 1	0x10040000	0x1004FFFF	AHB	64K
GIC2 (nFIQ IC) Tile Site 1	0x10050000	0x1005FFFF	AHB	64K
GIC3 (nIRQ IC) Tile Site 2	0x10060000	0x1006FFFF	AHB	64K
GIC4 (nFIQ IC) Tile Site 2	0x10070000	0x1007FFFF	AHB	64K
SMC configuration	0x10080000	0x1008FFFF	AHB	64K
Reserved for future use	0x10090000	0x100EFFFF	AHB	448K (64K * 7)
Reserved	0x100F0000	0x100FFFFF	APB	64K
Reserved	0x10100000	0x17FFFFFF	N/A	112M
Reserved	0x18000000	0x1FFFFFFF	N/A	128M
Reserved	0x20000000	0x3EFFFFFF	N/A	492M
ARM1176 AMBA peripherals	0x3F000000	0x3FFFFFFF	N/A	16M
Static Memory Controller	0x40000000	0x5FFFFFFF	AHB/AXI	512M
PCI interface	0x60000000	0x6FFFFFFF	AHB/AXI	256M
Dynamic Memory (alias)	0x70000000	0x7FFFFFFF	AHB/AXI	256M
Logic Tile Site 2	0x80000000	0xCF1FFFFF	AHB	2G
Reserved	0xCF200000	0xCF4FFFFF	N/A	3M
Logic Tile Site 2	0xCF500000	0xFFFFFFFF	AHB	2G

Table 5-1 Memory map

6 RTL

All of the APB RTL for this design is provided as verilog except for MMCI and GIC. AXI components are supplied as netlists. Example files are provided to allow building the system with Synplicity Synplify Pro and Xilinx ISE tools.

6.1 Directory structure



The application note has directories. These are:

docs	Related documents including this document.
logical	All the verilog RTL required for the design.
boardfiles	The files required to program the design into ARM development boards.
physical	Synthesis and place and route (P&R) scripts and builds for target board. Netlists for peripherals not provided as RTL (excluding PCI block).
software	ARM code to run on the AN177 system

6.2 logical

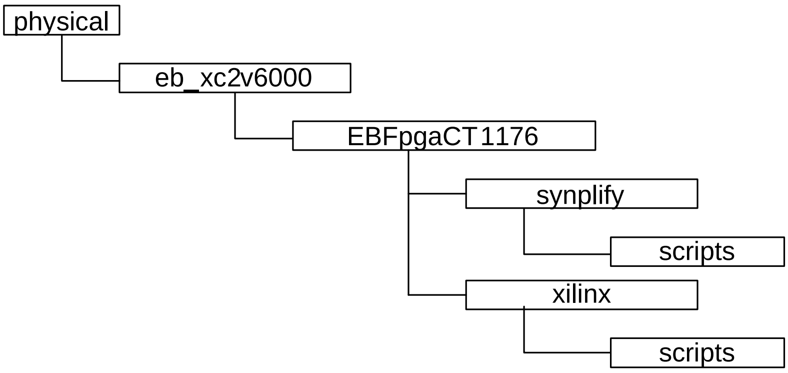
The logical directory contains all the verilog required to build the system. The physical directory contains pre-synthesised components. The function of each block is shown earlier in **3.2.1 EB Module functionality**.

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

The top level for this system is in EBFpgaCT1176.

6.3 physical

The physical directory contains the scripts for the tools used in the build process.



6.4 Building the App Note using Microsoft Windows or Unix

To rebuild the FPGA image for the EB with CT1176JZF-S you need to change into the EBFpgaCT1176 directory and run the make.scr (for UNIX) or make.bat (for windows) script. This will synthesis and Place & Route the design, linking in the required .NGO files at P&R. Read the readme.txt file in the directory for further build options.

make.bat	(default Synthesis and Place & Route)
make.bat all	(Synthesis and Place & Route)
make.bat synth	(Synthesis only)
make.bat par	(build -> map -> par -> bitgen)
make.bat all EBFpgaCTxxx dma	(build with DMA)
make.bat all EBFpgaCTxxx	(build without DMA)

To build the EBFpgaCT1176 image with DMA

```
cd EBFpgaCT1176
make.bat all EBFpgaCT1176 dma
```

To build the EBFpgaCT1176 image without DMA

```
cd EBFpgaCT1176
make.bat all EBFpgaCT1176 std
```

To synthesis the EBFpgaCT1176 with DMA

```
cd EBFpgaCT1176 DMA
make.bat synth EBFpgaCT1176
```

Note:
You need to ensure that the Synplify and Xilinx tools executable directories are in the path environment variable for DOS and UNIX.
The Synplify tools do not automatically add the path and the user is required to enter it manually. The Xilinx tools give you the option at installation time.

Any attempts to build the PCI image will fail without the appropriate Xilinx PCI netlist.

6.5 Board file selection

To use the pre-built bit file, use one of the following board files.

an177_eb_140c_xc2v6000_ct1176_le….brd

an177_eb_140c_xc2v6000_ct1176_pci_le….brd

To use a customer version, use one of the following board files.

an177_eb_140c_xc2v6000_ct1176_customer_rebuild.brd

an177_eb_140c_xc2v6000_ct1176_dma_customer_rebuild.brd