Application Note 148

Using a CT7TDMI, CT926EJ-S or CT1136JF-S Core Tile with an Emulation Baseboard

Document number: ARM DAI 0148D
Issued: October 2007

ARM

Application Note 148
Using a CT7TDMI, CT926EJ-S or CT1136JF-S Core Tile with an Emulation Baseboard

Release information

The following changes have been made to this Application Note.

Change history

Date	Issue	Change
June 29, 2005	A	First release
January 18, 2006	B	Getting started section added
August 15, 2006	C	Support for CT7TDMI added. Fixed errata.
February 6, 2007	D	CT7TDMI clock frequency updated

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 covers the operation of the RealView® Emulation Baseboard with an AHB based Core Tile or an ARM7TDMI based Core Tile. It describes the contents of the baseboard FPGA, the system interconnect, the clock structure, and specifics of the programmer's model relevant to Core Tile operation.

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

1.2 Emulation Baseboard and Core Tile overview

This application note is designed to work on an Emulation Baseboard with a Core Tile fitted to tile site 1 as shown in Figure 1.

This application note works with CT926EJ-S, CT1136JF-S and CT7TDMI.

The Emulation Baseboard provides the FPGA into which this design is placed.

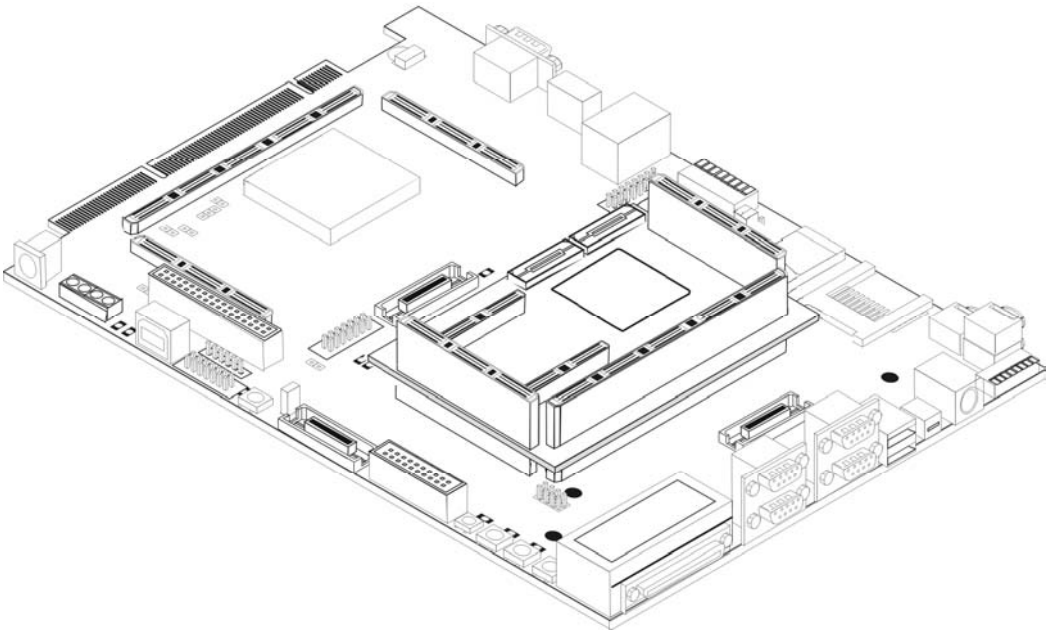


Figure 1: Core Tile and EB System

1.3 Optional Logic Tiles

This application note allows the user to extend the system with further Logic Tiles fitted to tile site 2. These Logic Tiles can contain both additional AHB masters and slaves, which are connected to the AHB infrastructure provided by the baseboard. See application note 146 for information about how to do this.

2 Getting started

Before you can use this application note, you will need to program the Emulation Baseboard with the required FPGA image to enable the Core Tile to function correctly. Follow these steps to program the FPGA image.

1. Plug the Core Tile onto tile site 1 of the Emulation Baseboard.
2. Slide the CONFIG switch (S1) to the ON position.
3. Connect RVI or Multi-ICE to the Emulation Baseboard JTAG ICE connector (J18), or a USB cable to the USB Debug Port (J16).
4. Check that 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 Multi-ICE, run Multi-ICE Server, press Ctrl-L and load the relevant manual configuration file from the \boardfiles\multi-ice directory. Depending on the version of Multi-ICE used it may also be necessary to add new devices to Multi-ICE. Please refer to \boardfiles\irlength_arm.txt for information on how to do this.
7. 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.
8. If using Real View 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).

9. You can now run the relevant 'progcards' utility for the connection you have prepared above.

    - progcards_multiice.exe for Multi ICE connection
    - progcards_usb.exe for USB connection
    - progcards_rvi.exe for RealView ICE connection

10. Select the appropriate option for the Core Tile you are using. The progcards 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 down the boards.
12. Set the configuration switches to load FPGA image 0. (S10 on the Emulation Baseboard set to all OFF).
13. Slide the CONFIG switch to the OFF position, and power-up 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.
14. The system will now be fully configured and ready for use.

3 Architecture

This application note implements an AHB (AMBA 2.0) based system on the Emulation Baseboard FPGA. This Emulation Baseboard image exposes interfaces appropriate to the daughter cards stacked on it:

- On tile site 1: One AHB master and one AHB slave for CT926EJ-S and CT1136JF-S, or one ARM7TDMI slave bus and one AHB master for CT7TDMI
- On tile site 2: One AHB master and one AHB slave going to the Logic Tile.

3.1 Block Diagram

Figure 2 Block Diagram shows a conceptual block diagram of a system consisting of a Core Tile, a Logic Tile and an Emulation Baseboard. Note that the direction of the arrows indicates the direction of control: it points from the Master to the Slave. An AHB bus contains signals going in both directions.

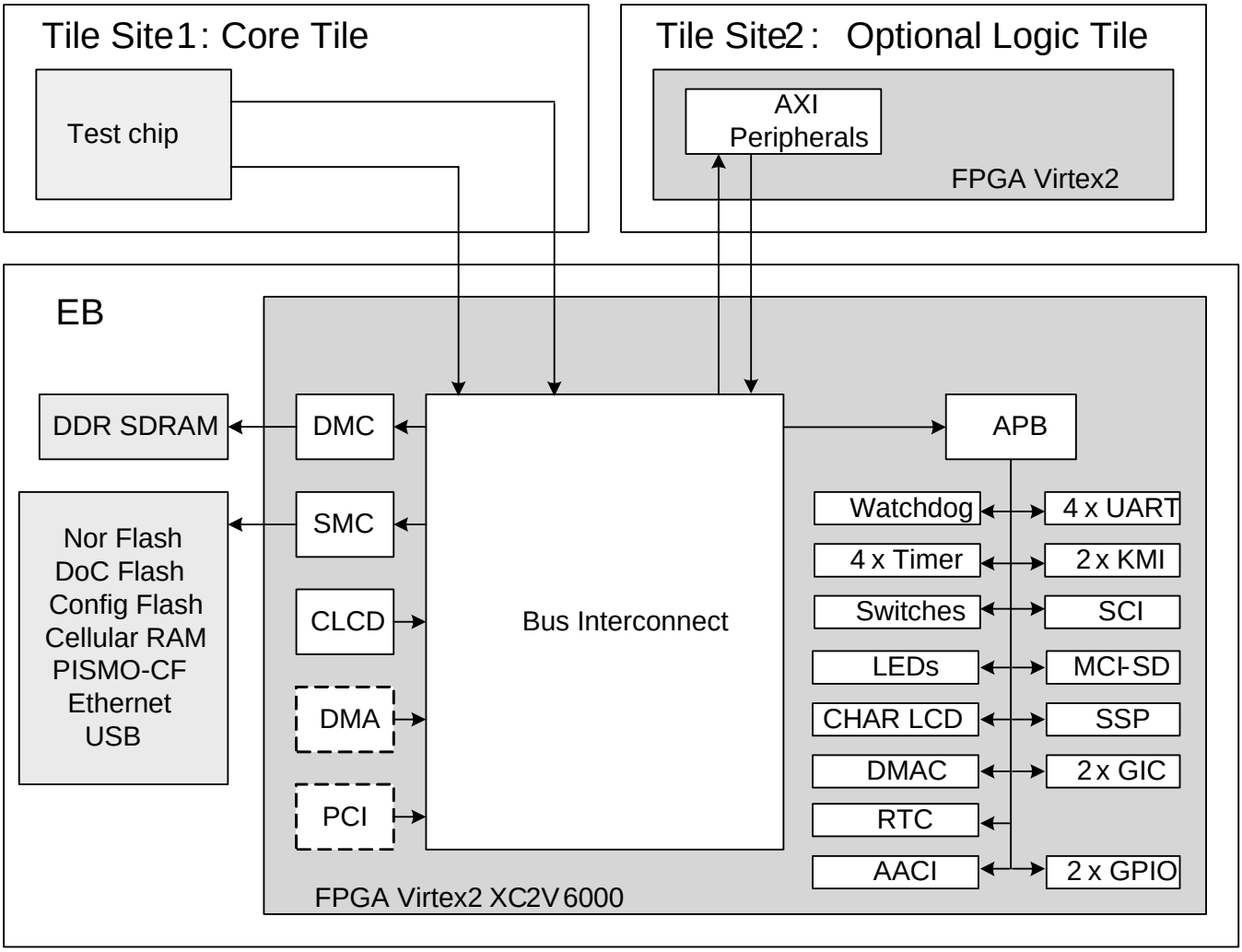


Figure 2 Block Diagram

3.2 Bus architecture

3.2.1 Bus Architecture of CT926EJ-S + EB or CT1136JF-S + EB

In order to support a CT926EJ-S or CT1136JF-S the EB implements an AHB bus infrastructure. An AXI bus matrix is used to give access to the AXI-based dynamic memory controller. An APB bus is used to connect PrimeCell APB peripherals.

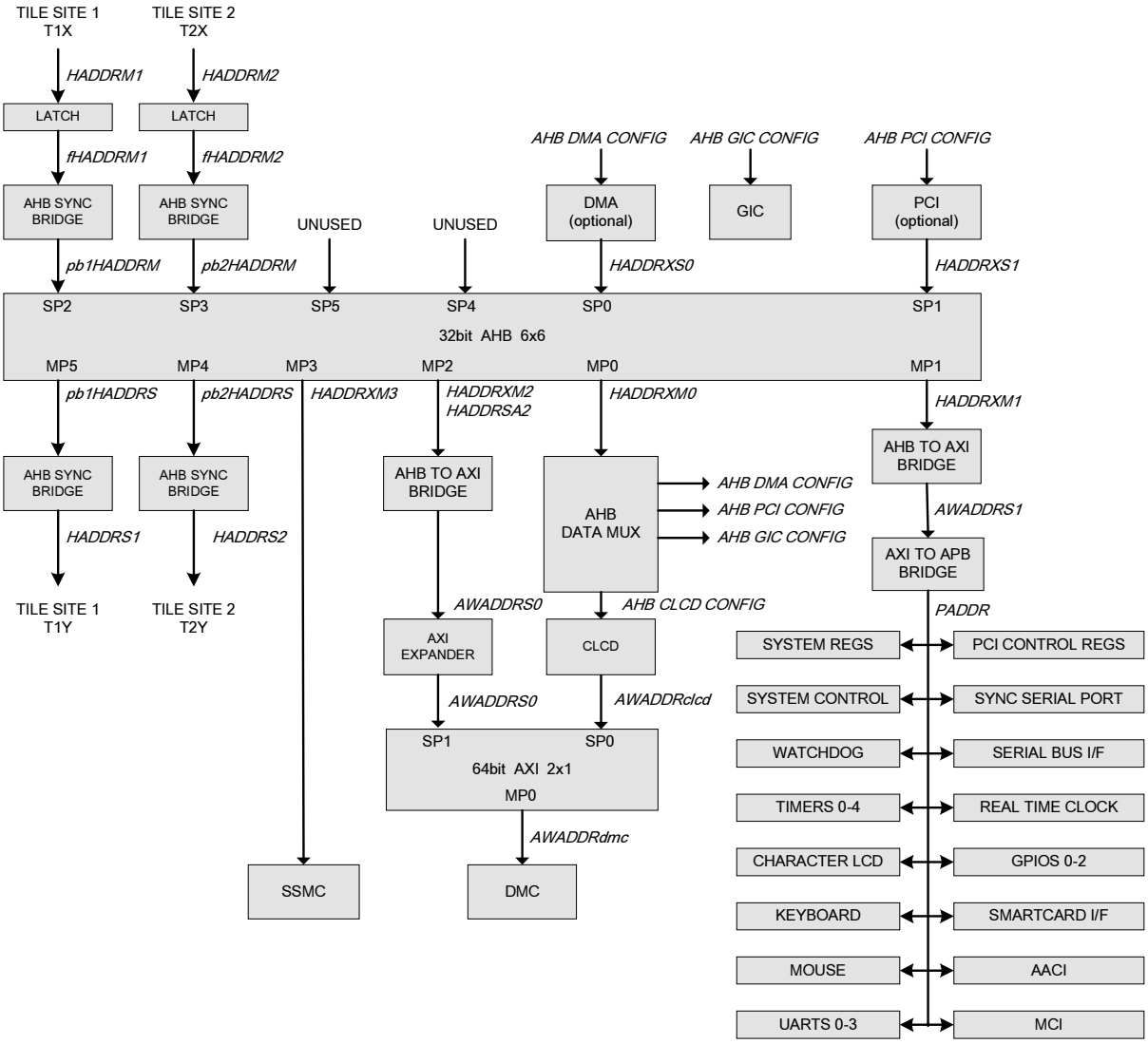


Figure 3: Bus Architecture of CT926EJ-S + EB or CT1136JF-S + EB

3.2.2 Bus Architecture of CT7TDMI + EB

In order to support a CT7TDMI with an AHB system, a wrapper has been inserted between the Core Tile and the AHB infrastructure.

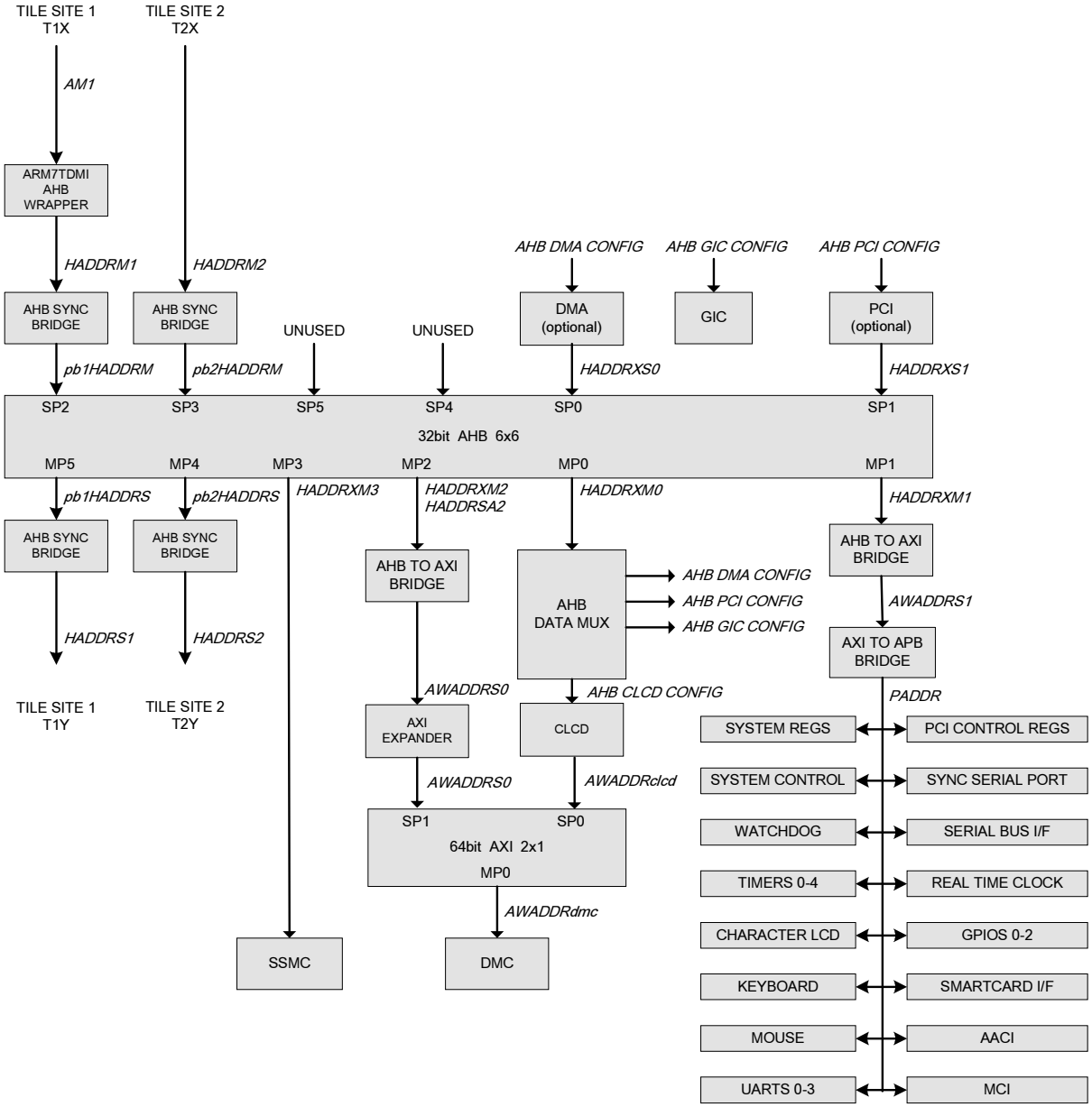


Figure 4: Bus Architecture of CT7TDMI + EB

3.2.3 Module functionality

Bus Infrastructure components

32-bit AHB 6x6 Bus Matrix: provides the bulk of the interconnect structure. It allows any of the 6 master ports to connect to any of the 6 slave 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 port priority of competing masters to a single slave.

64-bit AXI 2x1 Bus Matrix: The PL300 AXI bus matrix allows multiple masters to connect to the DMC. This is done separately from the 6x6 AHB bus matrix to increase the bandwidth from the CLCD controller to memory.

AHB Data Mux: This block allows you to connect one AHB master to up to 4 slaves. If less than 4 slaves are required, the unused parts can be tied off.

AXI Expander: This component is used to expand a 32-bit wide AXI bus to a 64-bit wide

AXI to APB Bridge: The bridge contains the muxing and decoding scheme for the bus, allowing 25 APB peripherals to be connected.

Latch: Optional transparent high latch added to the incoming master ports to ensure that there are no race conditions between the incoming signals and the rising edge of HCLK

Memory

DMC: ARM Primecell PL340 Dynamic Memory Controller

SSMC: ARM Primecell PL093 Synchronous Static Memory Controller

Peripherals

AACI: ARM Primecell PL041 Advanced Audio Codec Interface. The FIFO is increased in size from the standard to better suit the FPGA operating environment (lower bus frequency).

Character LCD: Controller for the Character LCD. It allows the system to communicate with it via memory mapped registers

CLCD: ARM Primecell PL111 Color Liquid Crystal Display Controller. Owing to the bus architecture of the design, the LCD controller can only access DRAM, it does not have access to other areas of the memory map.

DMA: An optional ARM Primecell PL081 single port Direct Memory Access controller can be added to the design

GPIOs 0-2: ARM Primecell PL061 General Purpose Input/Output modules

Keyboard and Mouse: ARM PrimeCell PL050 are used.

MCI: ARM Primecell PL180 Multimedia Card Interface.

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

PCI Control Regs: The PCI control block contains a set of registers to configure the PCI system (if used). See the programmer's reference section for more information on the functionality of these registers.

Real Time Clock: ARM Primecell PL031 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)

Serial Bus I/F: Controls the the serial bus to the PISMO and Time of Year clock.

Smartcard I/F: ARM Primecell PL131

Sync Serial Port: ARM Primecell PL022

System Control: ARM ADK component SP810 contains a generic set of system control registers.

System Regs: Set of registers for hardware control of the Emulation Baseboard. For a complete list of the functionality of these registers refer to the Emulation Baseboard user guide and Programmer Reference section of this application note.

Timers 0-3: ARM ADK component SP804

UARTs 0-3: ARM PrimeCell PL011 Universal Asynchronous Receiver-Transmitter interfaces (RS-232 serial).

Watchdog: ARM ADK component SP805 is the watchdog controller. It allows for the generation of an interrupt or reset after a defined time to prevent against system lockup/failure.

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 design is fully synchronous.

The User Guides for all the boards used in this configuration explain the clock options they support and how they are selected.

The reference clock is always driven by OSCCLK[0] on the baseboard. Refer to the System Registers description in the User Guide for more information on setting the clock oscillators.

3.3.1 System Clock Architecture with CT926EJ-S, CT1136JF-S

Figure 5 shows the clock routing to and from the CT926EJ-S, CT1136JF-S and Logic Tiles. The baseboard generates the reference clock for the Core Tile, which generates from it the CPU clock and the AHB bus clock HCLK. HCLK is used to clock all the peripherals on the EB FPGA. The Core Tile's HCLKIN input is connected to ground.

The CLK_NEG_UP signal is driven from the Core Tile. This makes HCLK available for use in other boards above the Core Tile.

Global Clock is not driven from the Core Tile directly, because there would be a delay between the Global Clock signal on tile site 1 and tile Site 2

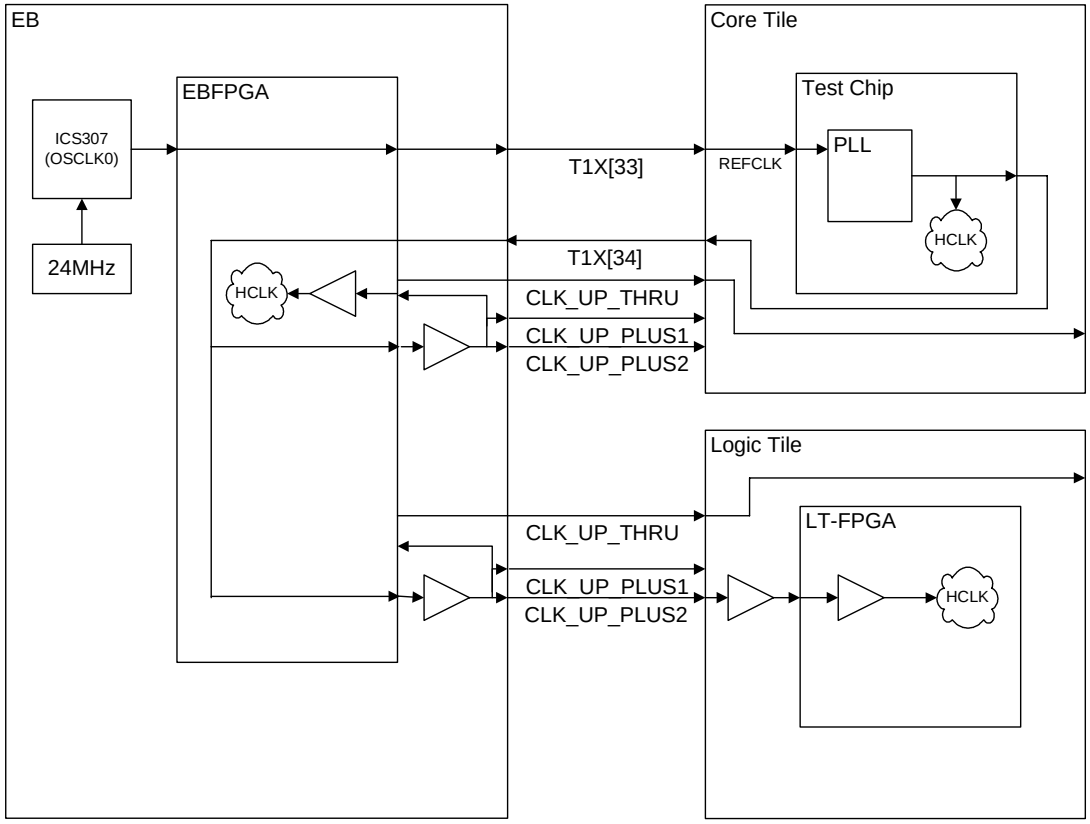


Figure 5: CT926EJ-S and CT1136JF-S Clock Architecture

1. The Core Tile can use either XL[33] or XU[33] as its reference clock. Since the Core Tile will always be connected on top of the Emulation Baseboard, its clock multiplexers are configured so that REFCLK is sourced from XL[33]. The baseboard drives a reference clock on T1X[33], which is generated from OSCCLK0.
2. The test chip generates the HCLK for the system based on its PLL and clock divider configuration settings. REFCLK should not be used to clock AHB peripherals, since it may note be synchronous with HCLK.
3. When the test chip contains a PLL, by default it is either enabled or set into BYPASS mode depending on the selected boot switches (see Section 8 for detailed information).
4. T1X[34] is chosen as the exported HCLK from the test chip, so that test chips with different VDDIO settings can be used. All signals on HDRX are able to support different voltage ranges.
5. In a design with no further expansion capabilities (i.e. no capability to add synchronous designs into additional Logic Tiles) it would be sensible to use T1X[34] to feed into an FPGA BUFG to drive the internal FPGA logic. To see examples of how this has been done in other systems refer to Application Notes 125, 136 or 138.
6. The Emulation Baseboard design feeds two copies of T1X[34] to CLK_OUT_TO_BUF[2:1] so that additional tiles fitted to tile site 1 or tile site 2 can be clocked. Although this signal is now delayed from the HCLK exported by the test chip, it is the best way of feeding the clock to the entire system.
7. A global clock buffer (BUFG) is deliberately not used between T1X[34] and GLOBALCLKOUT[2]. The overall effect on the pad to pad timing is not sufficiently large to warrant its use. The BUFG has a much greater effect being used to drive

other parts of the design. However it should be noted that if the requirement to use BUFGs elsewhere in the design is removed (for example if the user does not have the PCI module in their design), then there might be some advantage to allowing the use of a BUFG here.

8. CLK_BUF_LOOP[1] is used to drive a BUFG and all of the logic inside the baseboard FPGA. This signal is used, as its timing is guaranteed to be very close to the timing of CLK_UP_PLUS1/2 on any other boards on tile site 1.
9. The design is constrained so that the timing of CLK_BUF_LOOP[2] and CLK_BUF_LOOP[1] are as close together as possible. Both pins are very close together on the FPGA, and the routing is correctly constrained. This way the timing of CLK_UP_PLUS1/2 of boards on tile site 2 will be very close to that of tile site 1.
10. CLK_IN_MINUS1 is chosen as the system HCLK for both tile sites, as it is the only clock that allows for a synchronous clock to be provided for a large tile stack. The other clocks are limited to either a single tile, or 2 tiles above the baseboard.

.

3.3.2 System Clock Architecture with CT7TDMI

The clock architecture is slightly modified to allow for the fact that the ARM7TDMI test chip does not generate the AHB clock and the AHB wrapper is not on the same chip as the CPU. The ARM7TDMI clock input is the Core Tile's HCLKIN. This is shown in Figure 6.

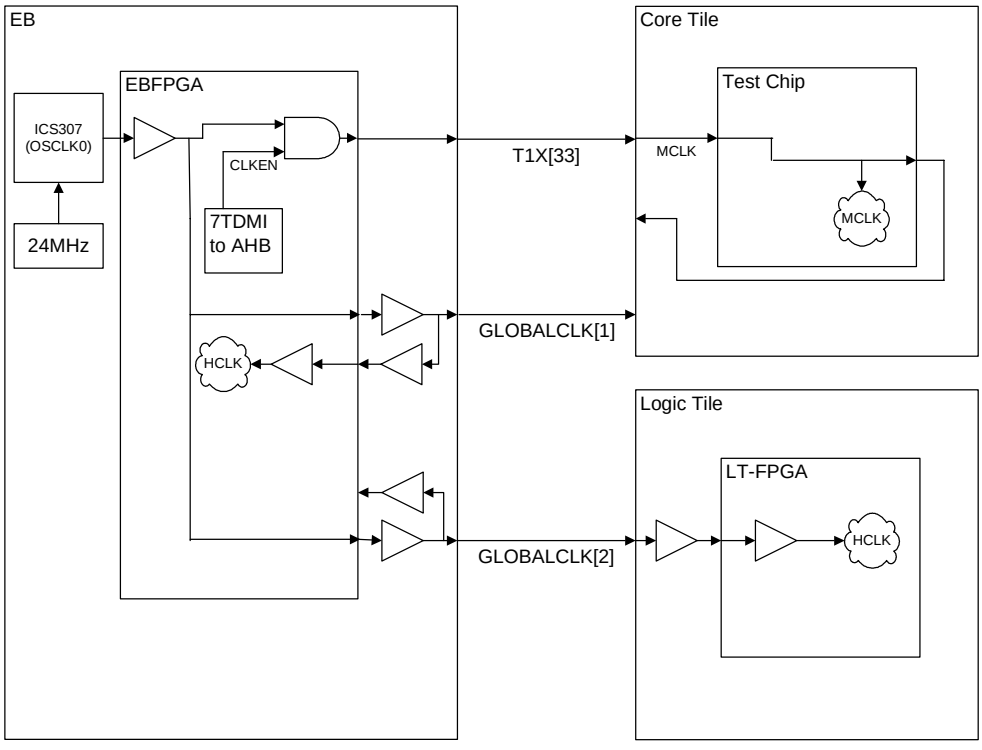


Figure 6: Bus Architecture of CT7TDMI + EB

3.3.3 PL340 dynamic memory clocking

The PL340 Dynamic Memory Controller requires three input clocks to drive the Dynamic Memory Interface: mclk, mclk2x, and fbclk_in. OSCCLK1 is used to generate the Dynamic Memory clock.

For complete descriptions of the functionality of these clocks refer to the PL340 documentation.

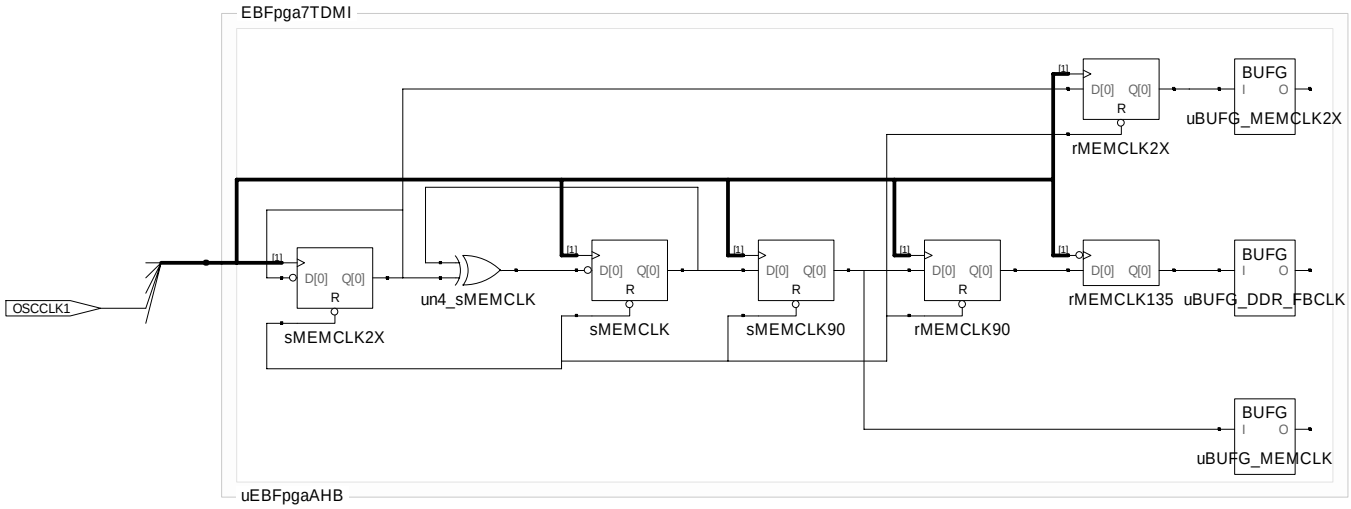


Figure 7: Dynamic Memory clock generation

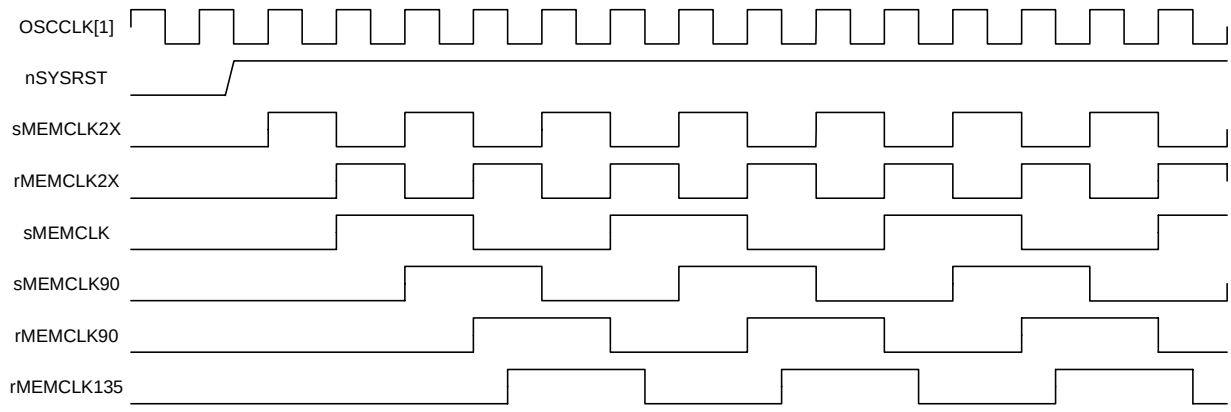


Figure 8: Dynamic Memory clock timing

The above figures show how the Dynamic Memory clocks are generated.

Mclk is fed by sMEMCLK90.

Mclk2X is fed by rMEMCLK2X.

FBCLK_IN is fed by rMEMCLK135.

The system is designed to run at a double rate clock of 40MHz (OSCCLK[1] is set to 80MHz).

3.4 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.

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.

3.5 Debug architecture

The example design provided with this Application Note is based on a simple JTAG debug architecture. It does not contain any CoreSight components.

The debug JTAG chain is routed through the baseboard and the two tile sites. The baseboard does not implement by default a virtual TAP controller. On the Core Tile, the debug JTAG chain is routed through the CPU's TAP.

For the Core Tiles that contain an ETM, the trace port is connected directly from the test chip to a trace MICTOR connector on the Core Tile itself.

4 Hardware description

4.1 Top Level

The top level EB FPGA design is EBFpga.v.

The top level of the design:

- handles all the static tie offs for the Core Tile
- instantiates the necessary pullup and pulldown resistors on FPGA pins to ensure that the system operates in different board configurations
- defines the mapping from the HDRX, HDRY and HDRZ buses from both tile sites to their functional allocations

The top level module instantiates other modules (EBFpgaCT7TDMI.v, EBFpgaCT926.v and EBFpgaCT1136.v), which interconnect the different blocks in the design.

The functionality of these modules is essentially the same for the different Core Tiles. The CT7TDMI image has minor differences:

- Addition of an ARM7TDMI to AHB wrapper
- Modification of the format of the Core Tile PLD serial stream.

4.1.1 CT926EJ-S and CT1136JF-S Core Tile Top Level RTL

Core Tiles can be configured in a number of working modes. This design sets the following mode of operation:

1. HRDATA and HWDATA are multiplexed onto a common HDATA bus routed on X[31:0].

   T1Z[226] indicates the current direction of the data bus.

   In this configuration it is not necessary to use X[179:148]. Therefore T1Z[225:224] are tied to 2'b11. This isolates the data connections to this bus and allows these signals to be used by other boards on the Tile Stack.

```
assign T1Z[224] = 1'b1; // datactl 0 should inactive (high)
assign T1Z[225] = 1'b1; // datactl 1 should inactive (high)
assign T2Z[224] = 1'b1; // datactl 0 should inactive (high)
assign T2Z[225] = 1'b1; // datactl 1 should inactive (high)
```

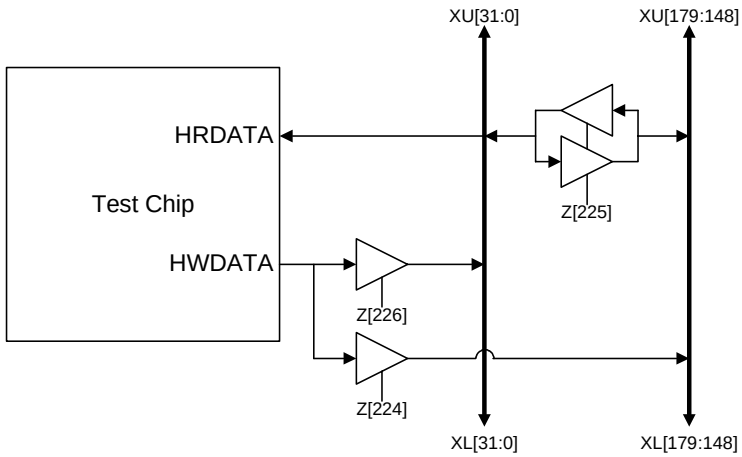


Figure 9: AHB Data Bus Routing

2. This design is for Core Tiles that generate HCLK and ignore HCLKIN. Therefore HCLKIN is tied to 0.

```
assign T1X[32] = 1'b0;//T1_X_HCLKIN_UP
assign T2X[32] = 1'b0;//T2_X_HCLKIN_UP
```

3. This design is for Little Endian operation only. Therefore BIGENDOUT is tied to 0.

```
assign T1X[54] = 1'b0; //little endian only
assign T2X[54] = 1'b0; //little endian only
```

4. This design allows JTAG debugging. This tie off can be changed to emulate a hardware system where debugger access via JTAG is not permitted.

```
assign T1X[60] = 1'b1;// DBGEN
assign T2X[60] = 1'b1;// DBGEN
```

5. External triggering into the ETM is not supported in the default design. In order to use an external trigger, connect it to the ETMEXTIN signal.

```
assign T1X[66] = 1'b0;//ETMEXTIN
assign T2X[66] = 1'b0;//ETMEXTIN
```

6. This system is designed to boot from address 0x0. In order to boot from hi-vectors (0xFFFF0000), change the VINITHI signal and either change the address decoding on the baseboard or add a boot device on a logic tile at this memory address.

```
assign T1X[68] = 1'b0;//VINITHI static signal
assign T2X[68] = 1'b0;//VINITHI static signal
```

7. The PLL in the test chips is enabled or set to bypass mode depending on the contents of the SYS_INIT register. If there is a PLL on the tile site 2, it is disabled by default

   The test chip CPUCLK to HCLK divider is also configured with SYS_INIT. The default contents of this register depend on the state of the boot switches. See Section 8 for more information about default clock setting.

```
assign T1X[70] = CONFIGWORD1[0];
assign T2X[70] = 1'b1;
assign T1X[74:72] = CONFIGWORD1[6:4];
assign T2X[74:72] = 3'b000;
```

8. By default the processor's instruction tightly coupled memory (ITCM) is disabled after reset, so that the processor boots from memory connected to the AHB interface. Since the test chips used on the Core Tiles only implement TCMs with RAM, this setup should not be modified.

```
assign T1X[71] =1'b0;//T1_INITRAM
assign T2X[71] =1'b0;//T1_INITRAM
```

9. This design does not include cross triggering or external debug run control (other than through the JTAG interface). It is possible to change these tie offs to allow a more sophisticated run control mechanism. See application note 125, or Integrator/CP-based designs to see how these signals can be used for hardware cross triggering to facilitate multiple core debugging.

```
assign T1X[75] = 1'b0;//T1_EDBGRQ
assign T2X[75] = 1'b0;//T2_EDBGRQ
```

10. By default, breakpoints can only be set through JTAG, so these signals are tied off.

```
assign T1X[77] = 1'b0;//T1_IEBKPT
assign T1X[76] = 1'b0;//T1_DEWPT
assign T2X[77] = 1'b0;//T2_IEBKPT
assign T2X[76] = 1'b0;//T2_DEWPT
```

11. The test chip user inputs can be configured with SYS_INIT. The default configuration should not be changed, as it is appropriate for the test chips fitted on the Core Tiles.

```
assign T1X[83:78] = CONFIGWORD1[29:24]; //USERIN
assign T2X[83:78] = 6'h00;   //USERIN
```

12. Part of the PLL configuration signals are tied off by the baseboard design to simplify the setting of the CPUCLK frequency. See the Core Tile user guide for a description of these inputs and how to calculate the CPU frequency, as the PLLs vary between test chips. Note that X[137:130] are the PLLFBDIV[7:0] signals into the Core Tile.

```
assign T1X[91:90] = 2'b00; //T1_PLLCTRL;
assign T2X[91:90] = 2'b00; //T2_PLLCTRL;
assign T1X[95:92] = 4'b0000; //T1_PLLREFDIV;
assign T2X[95:92] = 4'b0000; //T2_PLLREFDIV;
assign T1X[134:130] = 5'b00000; //PLLFBDIV
assign T2X[134:130] = 5'b00000; //PLLFBDIV
assign T1X[137:135] = 3'b000;//FBACK signals
assign T2X[137:135] = 3'b000;//FBACK signals
assign T1X[141:138] = 4'b0000;//PLLOUTDIV
assign T2X[141:138] = 4'b0000;//PLLOUTDIV
```

13. This design uses the test chips in their basic default configuration. These tie offs should not be changed

```
assign T1X[128] = 1'b0 ;//TESTSELECT
assign T1X[129] = 1'b0 ;//TICSELECT
assign T2X[128] = 1'b0 ;//TESTSELECT
assign T2X[129] = 1'b0 ;//TICSELECT
```

4.1.2 CT7TDMI Core Tile Top Level RTL

Some CT7TDMI control signals are different from those of CT926EJ-S and CT1136JF-S.

1. The ARM7TDMI bus interface is configured with a 32-bit bidirectional data bus. BUSEN selects a bidirectional bus instead of two unidirectional buses. DBE is tied high and nENIN low (both inactive) to give the CPU full access to the data bus. BL[3:0] is tied high because the memory system is 32-bit wide.

```
assign T1X[42]  = 1’b0;      //BUSEN - Bidirectional
assign T1X[55]  = 1’b1;      //DBE
assign T1X[80]  = 1’b0;      //nENIN
assign T1X[70:67] = 4’b1111; //BL[3:0]
```

2. ALE and APE are tied high to select the timing of the address signals.

```
assign T1X[64]  = 1’b1; //ALE - Address latch enable
assign T1X[65]  = 1’b1; //APE – Address pipeline enable
```

3. This design is for Little Endian operation only. However, with the only exception of the PCI system, the design is fully BIGEND ready. In order to change the design to Bigendian, simply change the top level tie off.

```
assign T1X[54]  = 1'b0; //BIGENDIN
```

4. nWAIT is used to generate wait states, so WAIT2 is unused.

```
assign T1X[48]  = 1'b0; //WAITSEL - nWAIT selected
assign T1X[51]  = 1'b0; //WAIT2
```

5. SEL[1:0] select a 32-bit memory system

```
assign T1X[35:34] = 2'b00; //SEL
```

6. F[1:0] are tied low, which disables the test chip outputs BC[1:0]. The BC bus is used for validation only.

```
assign T1X[46:45] = 2'b00; //F
```

7. This design allows JTAG debugging. This tie off can be changed to emulate a hardware system where debugger access via JTAG is not permitted.

```
assign T1X[60] = 1'b1;//DBGEN
```

8. This design does not include cross triggering or external debug run control (other than through the JTAG interface). It is possible to change these tie offs to allow a more sophisticated run control mechanism. See application note 125, or Integrator/CP-based designs to see how these signals can be used for hardware cross triggering to facilitate multiple core debugging.

```
assign T1X[75] = 1'b0;  //T1_EDBGRQ
assign T1X[66]  = 1'b0; //BREAKPT
```

9. The EXTERN inputs into the CPU EmbeddedICE logic are tied down, but could be used by the user to use signals from the design to configure breakpoints.

```
assign T1X[72:71] = 2'b00; //EXTERN[1:0]
```

10. The ARM7TDMI coprocessor interface is disabled. This setting should not be modified.

```
assign T1X[128] = 1'b1; //CPA
assign T1X[129] = 1'b1; //CPB
```

11. ARM7TDMI interrupts are configured as asynchronous.

```
assign T1X[58]  = 1'b0; // ISYNC
```

4.1.3 Clock Configuration

It is always necessary to tie off all clock outputs in any FPGA design. If this is not done, a floating clock signal can easily cause problems when connecting to other designs. If the user wishes to make use of these signals then they may do so, by changing their assignment here in the top level.

```
assign T1_CLK_POS_UP_OUT = 1'b0;
assign T2_CLK_POS_UP_OUT = 1'b0;
assign T1_CLK_NEG_UP_OUT = 1'b0;
assign T2_CLK_NEG_UP_OUT = 1'b0;
assign T1_CLK_UP_THRU    = 1'b0;
assign T2_CLK_UP_THRU    = 1'b0;
```

4.1.4 JTAG Routing

It is possible to add JTAG components into the debug scan chain inside this FPGA. This design does not add any component, so the JTAG signals are routed through the FPGA.

```
assign FPGA_D_TDO  = FPGA_D_TDI;
assign FPGA_D_RTCK = FPGA_D_TCK;
```

4.2 Core Tile serial stream

Since a Core Tile is stacked directly on top of the Emulation Baseboard, the baseboard FPGA needs to configure the Core Tile PLD through the serial configuration stream. This configuration stream is documented in the Core Tile User Guide. In this design this is done by the serialstream module inside the EBFpgaCTxxx module.

In this design many of the advanced Core Tile features are not used.

- The clocks are configured to match the chosen clock architecture.
- The test chip's multiple power sourcing options are not used. Therefore all power supplies fitted to any core tile are permanently enabled (PWR_nSHDN = 111).
- The Core Tile does not need to act as a Z-bus breaker as it is generally desirable in this system to allow connections from further logic tiles through to the baseboard and the opposite tile site (ZCTL = 0000).
- The MAN_ID and PLD_ID fields are read from the Core Tile into CT_PROCID and CT_PLD_CTRL registers in this design. This allows software to determine which Core Tile is fitted at the present time. This is a recommended operation, as it then allows software to make informed decisions about the hardware system it is operating on.
- PGOOD is ignored by the design. This signal can be used to act as a diagnostic that power is operating correctly on the Core Tile. This is recommended if the user is making use of advanced power management control to adjust the Core Tile regulators towards the edge of their operating range.

- DMEMSIZE and IMEMSIZE allow the user to configure their Core Tile to have a smaller TCM size than they really do. This might be useful for some benchmarking tests. This design makes use of the total amount of TCM on the test chip.
- The DAC inputs are set to their mid-range position (8'b10000000). This is their default position, so this sets all the power supplies up for standard operation. The Core Tiles allow for dynamic changing of their power supplies using these DACs for proving power saving or overclocking techniques. See the 'control' software example how this might be done through software control.
- The ADC allows for power diagnostics and power management performance to be measured. For an example of allowing software to access power information refer to Application Note 125, 136 or 138. This information is simply discarded in this design.

5 Programmer's model

5.1 Memory map

	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 (UART)	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
Reserved	*0x1001A000*	*0x1001FFFF*	*APB*	*28k (4k * 6)*
CLCD configuration	0x10020000	0x1002FFFF	AHB	64k
DMAC configuration	0x10030000	0x1003FFFF	AHB	64k
GIC1 (nFIQ IC) Tile Site 1	0x10040000	0x1004FFFF	AHB	64k
GIC2 (nIRQ IC) Tile Site 1	0x10050000	0x1005FFFF	AHB	64k
GIC3 (nFIQ IC) Tile Site 2	0x10060000	0x1006FFFF	AHB	64k
GIC4 (nIRQ IC) Tile Site 2	0x10070000	0x1007FFFF	AHB	64k
SMC configuration	0x10080000	0x1008FFFF	AHB	64k
Reserved	*0x10090000*	*0x100EFFFF*	*AHB*	*448k (64k * 7)*
DAP ROM table	0x100F0000	0x100FFFFF	APB	64k
Reserved	*0x10100000*	*0x17FFFFFF*	*N/A*	*112M*
Logic Tile Site 1	0x18000000	0x1FFFFFFF	AHB	128M
Reserved	*0x20000000*	*0x3FFFFFFF*	*N/A*	*512M*
Static Memory Controller	0x40000000	0x5FFFFFFF	*AHB*	512M
PCI interface	0x60000000	0x6FFFFFFF	*AHB*	256M
Dynamic Memory (Mirror)	0x70000000	0x7FFFFFFF	AHB/AXI	256M
Logic Tile Site 2	0x80000000	0xFFFFFFFF	AHB	2G

Table 5.1: Memory map

5.2 Core Tile specific registers

A number of registers are specific to the Core Tiles. The contents of these registers are used to access the configuration signals of the Core Tile via directly or via the Core Tile PLD serial stream. These registers are described below:

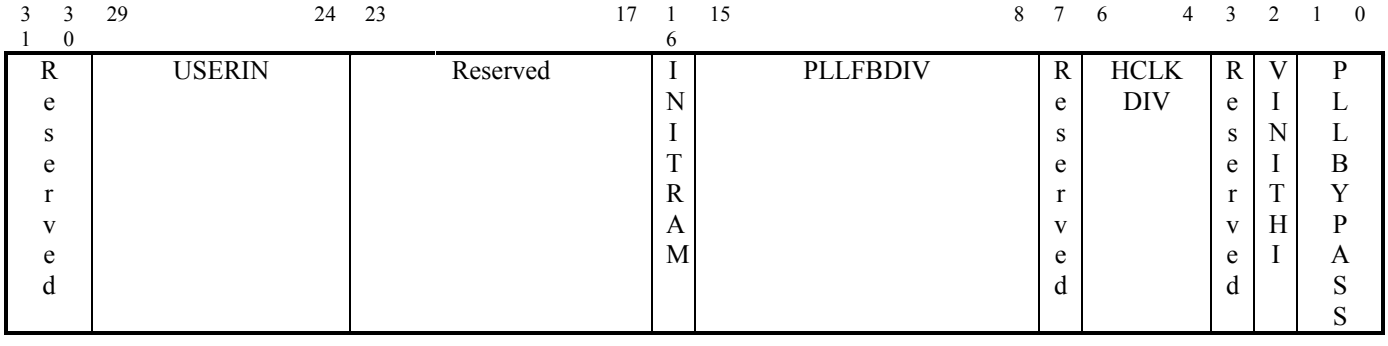
Register	Address	Read/write	Reset by
Core Tile Initialisation			
SYS_INIT	0x10000028	r/w	Power on reset
Core Tile PLD Control			
SYS _PLD_CTRL	0x10000074	r/w	Power on reset
SYS _PLD_CTRL2	0x10000078	r/w	Power on reset
Core Tile Voltage and Power Management			
SYS _VOLTAGE0	0x100000A0	r/w	Power on reset
SYS _VOLTAGE1	0x100000A4	r/w	Power on reset
SYS _VOLTAGE2	0x100000A8	r/w	Power on reset
SYS _VOLTAGE3	0x100000AC	r	Power on reset
SYS _VOLTAGE4	0x100000B0	r	Power on reset
SYS _VOLTAGE5	0x100000B4	r	Power on reset
SYS _VOLTAGE6	0x100000B8	r	Power on reset
SYS _VOLTAGE7	0x100000BC	r	Power on reset

All these registers must be unlocked by writing 0xA05F to SYS_LOCK register (0x10000020) first.

5.2.1 SYS_INIT

This register is used for the initialization of the ARM926EJ-S and ARM1136JF-S test chips fitted to tile site 1. It does not apply to the CT7TDMI.

SYS_INIT controls the configuration signals of the test chip at reset. In order for these values to take effect a soft reset must be applied after changing the contents of the register.

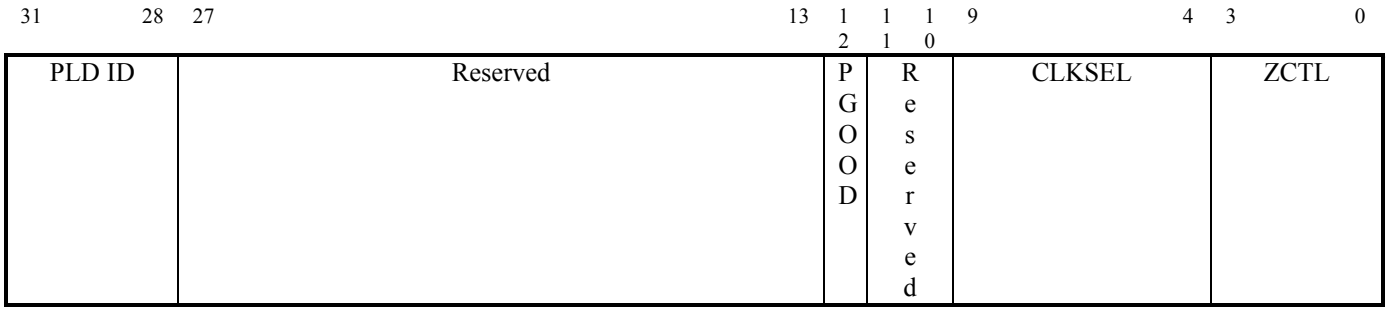


SYS_INIT	Read/write	Test Chip Initialisation
[31:30]	r/w	Reserved
[29:24]	r/w	USERIN[5:0]
[23:17]	r/w	Reserved
[16]	r/w	Internal RAM enable, 0 = disabled (default), 1 = enabled
[15:8]	r/w	PLLFBDIV[7:0]
[7]	r/w	Reserved

[6:4]	r/w	HCLK divider
[3]	r/w	Reserved
[2]	r/w	VINITHI, 0 = vectors at 0x0, 1 = vectors at 0xffff0000
[1]	r/w	PLLBYPASS (value read from pin after reset) – read only
[0]	r/w	PLLBYPASS, 0 = off, 1=on (default), takes effect only after reset

5.2.2 SYS_PLD_CTRL

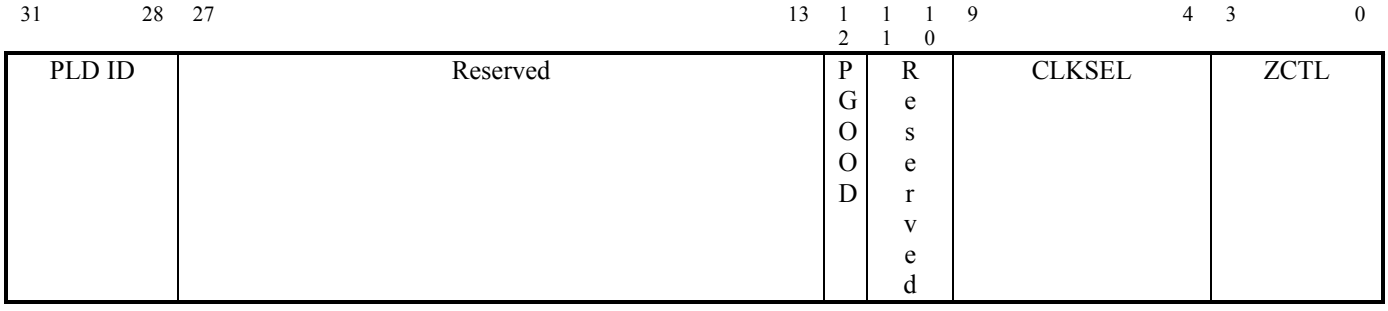
This register allows the user to set the configurable options on the Core Tile fitted to tile site 1. These values are communicated across the Serial Interface to the Core Tile PLD.



SYS_PLD_CTRL	**Read/write**	**Core Tile PLD Control/Status**
[31:28]	r	PLD_ID. Determines the current build/type of the PLD image.
[27:13]	r	Reserved
[12]	r	CT926EJ-S and CT1136JF-S: PGOOD. Active high signal indicating that supply on Core Tile board is working OK. CT7TDMI: Reserved
[9]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL5: NEG_DN_OUT assigned HCLK,NEG_DN_I CT7TDMI: CLKSEL5: Route X_MCLK and X_ECLK across Core Tile
[8]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL4: NEG_UP_OUT assigned HCLK,NEG_UP_I CT7TDMI: CLKSEL4: NEG_DN_OUT assigned HCLK,NEG_DN_IN
[7]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL3: Set low to drive CLK_GLOBAL CT7TDMI: CLKSEL3: NEG_UP_OUT assigned HCLK,NEG_UP_IN
[6:5]	r/w	CLKSEL[2:1]: HCLK gets CLK_NEG, X_HCLK, GND, CLK_GLOBAL
[4]	r/w	CLKSEL0: Set low for tile below/high for tile above
[3]	r/w	ZCTL3: Set high to break Z[127:96] bus
[2]	r/w	ZCTL2: Set high to break Z[95:64] bus
[1]	r/w	ZCTL1: Set high to break Z[63:32] bus
[0]	r/w	ZCTL0: Set high to break Z[31:0] bus

5.2.3 SYS_PLD_CTRL2

This register allows the user to set the configurable options on the Core Tile PLD fitted to tile site 2. These values are communicated across the Serial Interface to the Core Tile PLD.

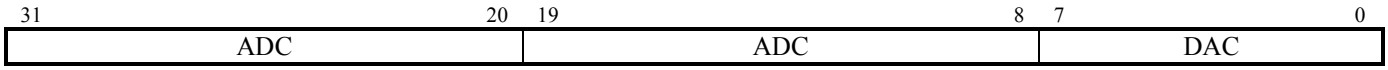


SYS_PLD_CTRL	**Read/write**	**Core Tile PLD Control/Status**
[31:28]	r	PLD_ID. Determines the current build/type of the PLD image.
[27:13]	r	Reserved
[12]	r	CT926EJ-S and CT1136JF-S: PGOOD. Active high signal indicating that supply on Core Tile board is working OK.

		CT7TDMI: Reserved
[9]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL5: NEG_DN_OUT assigned HCLK,NEG_DN_I
		CT7TDMI: CLKSEL5: Route X_MCLK and X_ECLK across Core Tile
[8]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL4: NEG_UP_OUT assigned HCLK,NEG_UP_I
		CT7TDMI: CLKSEL4: NEG_DN_OUT assigned HCLK,NEG_DN_IN
[7]	r/w	CT926EJ-S and CT1136JF-S: CLKSEL3: Set low to drive CLK_GLOBAL
		CT7TDMI: CLKSEL3: NEG_UP_OUT assigned HCLK,NEG_UP_IN
[6:5]	r/w	CLKSEL[2:1]: HCLK gets CLK_NEG, X_HCLK, GND, CLK_GLOBAL
[4]	r/w	CLKSEL0: Set low for tile below/high for tile above
[3]	r/w	ZCTL3: Set high to break Z[127:96] bus
[2]	r/w	ZCTL2: Set high to break Z[95:64] bus
[1]	r/w	ZCTL1: Set high to break Z[63:32] bus
[0]	r/w	ZCTL0: Set high to break Z[31:0] bus

5.2.4 SYS_VOLTAGE

The SYS_VOLTAGE_CTLx registers can be used to access the DACs and ADCs fitted to the Core Tile.



Depending on the Core Tile build not all three Voltage domains may be present, and the scaling of the voltage domains varies between builds. It is recommended that when scaling the voltage, the algorithm to set the new voltage works by incrementing the DAC and using to ADC to determine the effect this is having. See the Core Tile User Guide for a full explanation of this circuitry, and how to make use of these values.

CT7TDMI has fewer voltage domains than CT926EJ-S and CT1136JF-S, so on CT7TDMI only SYS_VOLTAGE_CTL0-3 are used.

SYS_VOLTAGE_CTL0	**Read/write**	**Voltage Control 0**
[31:20]	r	VDDCORE1 DIFF
[19:8]	r	VDDCORE1 voltage ADC value (12 bits)
[7:0]	r/w	Core voltage A DAC value
SYS_VOLTAGE_CTL1	**Read/write**	**Voltage Control 1**
[31:20]	r	VDDCORE2 DIFF
[19:8]	r	VDDCORE2 voltage ADC value (12 bits)
[7:0]	r/w	Core voltage B DAC value
SYS_VOLTAGE_CTL2	**Read/write**	**Voltage Control 2**
[31:20]	r	VDDCORE3 DIFF
[19:8]	r	CT926EJ-S and CT1136JF-S: VDDCORE3 voltage ADC value (12 bits)
		CT7TDMI: PISMO 1.8V supply voltage ADC value (12 bits) – not scaled
[7:0]	r/w	CT926EJ-S and CT1136JF-S: Core voltage C DAC value
		CT7TDMI: Reserved
SYS_VOLTAGE_CTL3	**Read/write**	**Voltage Control 3**
[31:20]	r	CT926EJ-S and CT1136JF-S: VDDCORE4 DIFF
		CT7TDMI: TP1 voltage ADC value (12 bits)
[19:8]	r	CT926EJ-S and CT1136JF-S: VDDCORE4 voltage ADC value (12 bits)
		CT7TDMI: TP2 voltage ADC value (12 bits)
[7:3]	r	Reserved
[2]	r/w	CT926EJ-S and CT1136JF-S: Core voltage C enable (nSHDN) (RESET)
		CT7TDMI: Reserved
[1]	r/w	CT926EJ-S and CT1136JF-S: Core voltage B enable (nSHDN) (RESET)
		CT7TDMI: Reserved
[0]	r/w	CT926EJ-S and CT1136JF-S: Core voltage A enable (nSHDN) (RESET)
		CT7TDMI: Reserved

SYS _VOLTAGE_CTL4	**Read/write**	**Voltage Control 4 (CT926EJ-S and CT1136JF-S only**
[31:20]	r	VDDCORE5 DIFF
[19:8]	r	VDDCORE5 voltage ADC value (12 bits)
[7:0]	r/w	Reserved
SYS _VOLTAGE_CTL5	**Read/write**	**Voltage Control 5 (CT926EJ-S and CT1136JF-S only)**
[31:20]	r	VDDCORE6 DIFF
[19:8]	r	VDDCORE6 voltage ADC value (12 bits)
[7:0]	r/w	Reserved
SYS _VOLTAGE_CTL6	**Read/write**	**Voltage Control 6 (CT926EJ-S and CT1136JF-S only)**
[31:20]	r	VDDPLL2 voltage ADC value (12 bits)
[19:8]	r	VDDPLL1 voltage ADC value (12 bits)
[7:0]	r/w	Reserved
SYS _VOLTAGE_CTL7	**Read/write**	**Voltage Control 7 (CT926EJ-S and CT1136JF-S only)**
[31:20]	r	Test Point voltage ADC value (12 bits)
[19:8]	r	VDDIO voltage ADC value (12 bits)
[7:0]	r/w	Reserved

5.3 Boot operation

In systems built around a Core Tile on top of the Emulation Baseboard the processor should normally boot from the baseboard NOR Flash (all EB S8 configuration switches in the up position). NOR Flash is preprogrammed with the boot monitor. See the Emulation Baseboard user guide and Versatile Firmware documentation for how this operates.

The user can also program custom boot software in NOR Flash, but note that any software configurable devices such as the Dynamic Memory Controller will not work until properly configured.

6 RTL

All of the RTL for this design is provided as Verilog or VHDL or precompiled netlists. Example files are provided to allow building the system with Synplicity Synplify Pro and Xilinx ISE tools. The readme files provided with the application note show the version of the tools used to build the design.

6.1 Directory structure

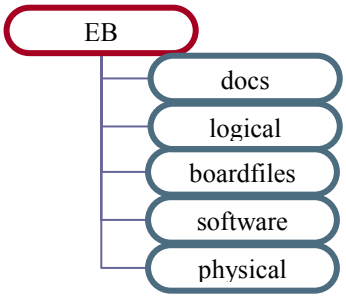


Figure 10: Top Level Directory Structure

The application note has several directories:

- 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 with the utility Progcards
- Software: Example software and utilities specific to the application note
- Physical : Synthesis, P&R scripts, netlists and FPGA builds

6.2 Logical

The logical directory contains all the Verilog or VHDL required to build the system (some blocks are provided as pre built .ngo files in physical\eb_xc2v6000\xxxx\xilinx\netlist). The function of each block is shown earlier in section 3.

Each Primecell or other large IP block has its own directory (e.g. Ahb2AhbSync).

The top level for this Application Note is the EBFpga module in EBFpga.v.

The system registers and their default settings are also in this directory.

6.3 Targets

Each potential target to implement the RTL has a directory. In the case of this application note only one target is available: eb_xc2v6000. Within this directory are all the different builds which exist for this target. In turn each of these has directories for each tool used in the build process.

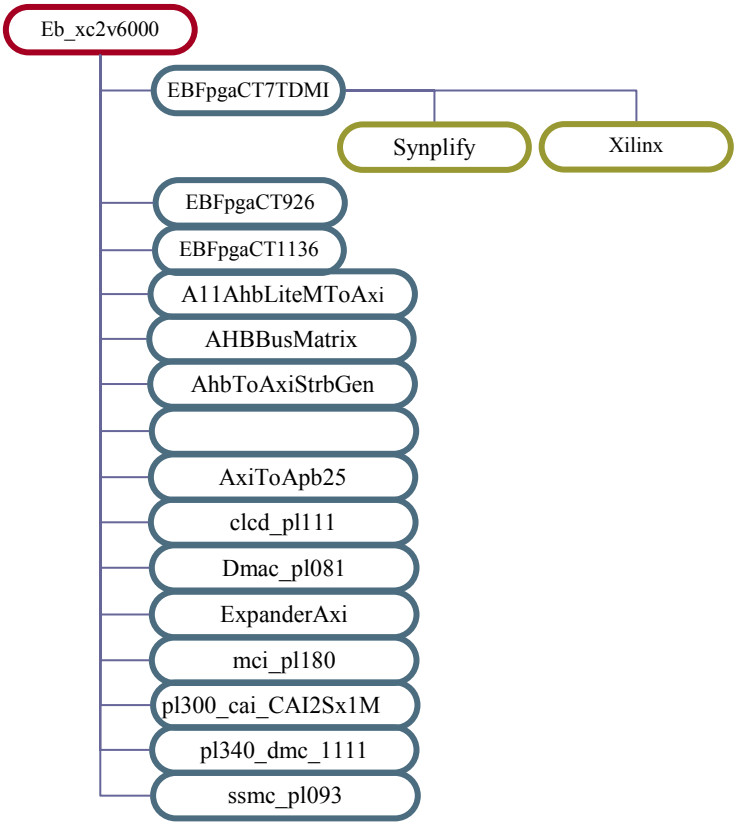


Figure 11: Phisical Directory Structure

6.4 Building the application note

In order to rebuild the Application Note it is necessary to run through both Synthesis and P&R.

There are two supported methods for synthesizing a design. These are PC script, or UNIX script.

1. To run under a PC script, open a command prompt and navigate to the \physical\eb_xc2v6000\EBFpgaCTxxx\ directory. Execute make.bat

   make.bat [option] EBFpgaCTxxx <dma>

   [option] – synth – for Synthesis only

   par – for Place & Route only

   all – for Synthesis and Place & Route

   EBFpgaCTxxx – build name

   <dma> - optional for add dma controller PL081 to design

   Example:

   make.bat all EBFpgaCTxxx dma – synthesize and place and route EBFpgaCTxxx build with dma controller.

2. To run under a UNIX script, open a shell window and navigate to the \physical\eb_xc2v6000\EBFpgaCTxxx\ directory. Execute make.scr

   make.scr [option] EBFpgaCTxxx <dma>

[option] – synth – for Synthesis only

par – for Place & Route only

all – for Synthesis and Place & Route

EBFpgaCTxxx – build name

<dma> - optional for add dma controller PL081 to design

Example:

make.bat synth EBFpgaCTxxx – synthesize EBFpgaCTxxx build without dma controller.

On completion the results can be found in the EBFpgaCTxxx\Xilinx\netlist directory.

7 Functional testing

7.1 Self-test

The selftest code allows the user to confirm the functionality of their baseboard, and provides a starting point for writing end code to make use of the Emulation Baseboard peripherals.

7.1.1 Functionality

Selftest is a piece of diagnostics for testing the following peripherals: AACI, MMCI, SCI, USB, Ethernet, CLCD/VGA, TSCI and keypad, CLCD Bias, SSP, UARTs, character LCD, LEDs, switches, GPIO, keyboard, mouse, SDRAM/SRAM memory, RTC/TOY clocks and system clocks

Selftest is designed to run on a semihosted system through a JTAG debugger. The user can interact with the software operation via the debugger's console window.

The user interface displays a menu and prompts the user on how to operate each test. For more information on exactly how each test is working, refer to the provided code source, and readme files.

7.1.2 Compilation notes

A Codewarrior project file is shipped as part of the selftest suite. This project file can be used to rebuild the code with Codewarrior. A makefile is also provided, so that automated builds can also be run.

8 Clock frequency settings

8.1 Startup frequencies

The startup frequencies should be set to start with a stable system. The Core Tile release notes can be used for information about the maximum frequency of the test chip.

Test Chip	REFCLK	CPUCLK	HCLKI (peripherals on test chip)	HCLKE (peripherals on baseboard)
ARM926EJ-S UMC 0.18um LF712	30 MHz	30 – 270 MHz	-	30 – 270 MHz
ARM1136JF-S TSMC 0.13um	30 MHz	30 – 270 MHz	20 - 280 MHz	30 - 270 MHz
ARM7TDMI Hynix HS255002	30 MHz	30 MHz	-	30 MHz
ARM7TDMI EPSON T0464FA70	30 MHz	30 MHz	-	30 MHz

Table 8.1: Default Clock Frequencies

The OSCCLK[0] setting always matches the system bus clock speed. It is possible to change the CPU frequency by setting the BOOTCSSEL (S8) switches for the CT926EJ-S and CT1136JF-S. However the CPU on CT7TDMI must always run at the same frequency as the system bus.

BOOTCSSEL[7:5]	CPUCLK:HCLK
000	1:1 (PLLBYPASSED)
001	2:1
010	4:1
011	6:1
100	8:1
101	10:1
110	12:1
111	14:1

Table 8.2: Startup CPU to Bus Clock Ratio

The system registers can be used to change the frequency of OSCCLK[0] “on the fly”. When using the PLL (any setting other than ‘000’, it may be necessary to perform a reset operation to allow the PLLs to lock to the new frequency.

9 Example software

The example software has been coded to support CT7TDMI, CT926EJ-S and CT1136JF-S. By reading the processor bus type where applicable the software can operate across different systems. Since all the examples are compiled for the lowest common code architecture (v4 is the lowest expected with this hardware), the code runs on all Core Tiles.

9.1 Voltage Control

The ‘control’ example is intended to be executed directly by the Core Tile. It demonstrates the use of the DAC and ADC parts fitted to the Core Tile. It acts as an easy way to confirm that the system is powered correctly with the correct voltages, and allows the user to implement changes to the programmable supplies.

9.2 SimplePCIscan

The SimplePCIscan code is a simple example application for displaying the current PCI devices connected to the PCI expansion port on the Versatile EB.

10 Signal assignments

This section shows the use of the signals on the tile site 1 and 2 connectors, which carry the signals between the baseboard and the Core Tile or the Logic Tile.

The direction of the signals is shown from the point of view of the baseboard, so an O signal goes from the baseboard to the Core Tile.

10.1 HDRX

HDRX signals are connected directly to the test chip on the Core Tile on tile site 1, or to an AHB master implemented on a Logic Tile on tile site 2

10.1.1 Tile Site 1 HDRX for CT926EJ-S and CT1136JF-S Designs

X[n]	Dir	Signal	Description
0 to 31	I/O	HDATA[0] to HDATA[31]	32-bit AHB from Core Tile to Baseboard – Data bus
32	O	X_HCLKIN	External HCLK source for a Core Tile – Tied to 1
33	O	X_REFCLK	Core Tile reference clock - Sourced from OSCCLK[0]
34	I	X_HCLK	HCLK returned from the Core Tile – clocks AHB system
35	I	HSIZE[2]	32-bit AHB from Core Tile to Baseboard – Access size
36 to 37	I	HTRANS[0] to HTRANS[1]	32-bit AHB from Core Tile to Baseboard – Transaction type
38 to 39	I	HSIZE[0] to HSIZE[2]	32-bit AHB from Core Tile to Baseboard – Access size
40 to 43	I	HPROT[0] to HPROT[3]	32-bit AHB from Core Tile to Baseboard – Sideband information
44 to 46	I	HBURST[0] to HBURST[2]	32-bit AHB from Core Tile to Baseboard – Burst length
47	I	HWRITE	32-bit AHB from Core Tile to Baseboard – Write/read signal
48	I	HBUSREQ	32-bit AHB from Core Tile to Baseboard – Bus request signal
49	I	HLOCK	32-bit AHB from Core Tile to Baseboard – Lock signal
50	O	HREADY	32-bit AHB from Core Tile to Baseboard – Slave is ready for new transaction
51	O	HGRANT	32-bit AHB from Core Tile to Baseboard – Master is granted the bus
52 to 53	O	HRESP[0] to HRESP[1]	32-bit AHB from Core Tile to Baseboard – Response from slave
54	O	BIGENDIN	CPU endianness configuration input – selects endianness at power-up
55	I	BIGENDOUT	CPU endianness status output – shows CPU endianness
56	I	COMMRX	Debug Comms channel interrupts from the CPU, connected to inputs of the interrupt controller.
57	I	COMMTX	
58	O	CONFIGINIT	Test chip configuration signal – Tied to 0
59	I	DBGACK	CPU debug acknowledgement – shows if CPU is in debug state
60	O	DBGEN	CPU debug enable configuration input – Tied to 1
61	O	nFIQ	CPU fast interrupt input
62	O	nIRQ	CPU normal interrupt input
63	O	ARM_nRESET	System reset – HRESETn
64	O	ARM_nPORESET	Power on reset – initializes all the logic on the Core Tile
65	O	nCONFIGRST	Test chip configuration signal – Tied to 1
66	O	ETMEXTIN	Test chip ETM external trigger input – Tied to 0

67	I	ETMEXTOUT	Test chip ETM external trigger output – Unused
68	O	VINITHI	CPU configuration signal – High vector table (0xFFFF0000) – Tied to 0
69	I	PLLLOCK	Indicates that the PLL inside the Core Tile test chip has locked to the reference
70	O	PLLBYPASS	Test chip configuration input – PLL Bypass – Tied to 1
71	O	INITRAM	CPU configuration input – Enables TCM at reset – Tied to 0
72 to 74	O	HCLKDIV[0] to HCLKDIV[2]	Test chip bus divider ratio CPUCLK/HCLKI
75	O	EDBGRQ	CPU debug request input – Tied to 0
76	O	DEWPT	CPU external watchpoint input – Tied to 0
77	O	IEBKPT	CPU external breakpoint input – Tied to 0
78 to 83	O	USERIN[0] to USERIN[5]	Test chip user configuration inputs – Driven with contents from system registers
84 to 89	I	USEROUT[0] to USEROUT[5]	Test chip user configuration outputs – Unused
90 to 91	O	PLLCTRL[0] to PLLCTRL[1]	Test chip PLL control inputs - Exact function varies between test chips – Tied to 0
92 to 95	O	PLLREFDIV[0] to PLLREFDIV[3]	Test chip PLL reference divider configuration input – Exact function varies between test chips – Tied to 0
96 to 127	I	HADDR[0] to HADDR[31]	32-bit AHB from Core Tile to Baseboard – Address bus
128	O	TESTSELECT	Test chip configuration input - Enables scan test mode – Tied to 0
129	O	TICSELECT	Test chip configuration input - Enables TIC testing mode – Tied to 0
130 to 137	O	PLLFBDIV[0] to PLLFBDIV[7]	Test chip PLL feedback divider configuration input – Exact function varies between test chips – Tied to 0
138 to 141	O	PLLOUTDIV[0] to PLLOUTDIV[3]	Test chip PLL output divider configuration input – Exact function varies between test chips – Tied to 0
142	I	EXTTRIG	External trigger condition from Core Tile – Unused
143	N/A	Unused	Unused
144 to 179	N/A	NC	Not connected to EB FPGA

Table 10.1: Tile Site 1 HDRX for CT926EJ-S and CT1136JF-S Designs

10.1.2 Tile Site 1 HDRX for CT7TDMI Design

X[n]	Dir	Signal	Description
0 to 31	I/O	DATA[0] to DATA[31]	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Data bus
32	O	X_MCLK_UP	MCLK source for the Core Tile.
33	I	X_ECLK_DN	ECLK from the test chip – Unused
34 to 35	O	SEL[0] to SEL[1]	Test chip configuration input – Tied to 0 to select a 32-bit memory bus
36	I	nMREQ	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Memory request
37	I	SEQ	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Sequential address
38 to 39	I	MAS[0] to MAS[1]	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Access size
40	I	nOPC	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Opcode fetch
41	I	nTRANS	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Memory translate

42	O	BUSEN	CPU configuration input - Data bus configuration – Tied to 0
43 to 44	I	BC[0] to BC[1]	Test chip output – Reports the cycle type – Drives 0 with configuration used
45 to 46	O	F[0] to F[1]	Test chip configuration signals – Select use of BC[1:0] – Tied to 0
47	I	nRW	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Read/Write
48	O	WAITSEL	Test chip configuration signal – nWAIT (not WAIT2) is selected to generate wait states
49	I	LOCK	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Lock signal
50	O	nWAIT	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Wait signal
51	O	WAIT2	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Alternative wait signal
52	O	ABORT	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Memory abort
53	I	TBIT	CPU configuration output – Reports Thumb status
54	O	BIGENDIN	CPU configuration output – Selects endianness
55	O	DBE	CPU configuration input – Bus control – Tied to 1
56	I	COMMRX	Debug Comms channel interrupts from the CPU, connected to inputs of the interrupt controller.
57	I	COMMTX	
58	O	ISYNC	CPU configuration input – Selects synchronous interrupts – Tied to 0
59	I	DBGACK	CPU debug acknowledgement – shows if CPU is in debug state
60	O	DBGEN	CPU debug enable configuration input – Tied to 1
61	O	nFIQ	CPU fast interrupt input
62	O	nIRQ	CPU normal interrupt input
63	O	nRESET	System reset – HRESETn
64	O	ALE	CPU configuration input – Address latch enable – Tied to 1
65	O	APE	CPU configuration input - Address pipeline enable – Tied to 1
66	O	BREAKPT	CPU input – External breakpoint – Tied to 0
67 to 70	O	BL[0] to BL[3]	CPU configuration input - Byte latch enable – Tied to 1
71 to 72	O	EXTERN[0] to EXTERN[1]	CPU input – External input to EmbeddedICE logic – Tied to 0
73	I	HIGHZ	CPU output – Output from TAP controller – Unused
74	I	DBGRQI	CPU output - Internal Debug request – Unused
75	O	DBGRQ	CPU debug request input – Tied to 0
76 to 79	I	IR[0] to IR[3]	CPU output – Output from TAP controller – Unused
80	O	nENIN	CPU configuration input – Bus not enable - Tied to 0
81	I	nENOUT	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Bus enable
82	I	nENOUTI	CPU output – Bus enable from EmbeddedICE logic – Unused
83	I	nEXEC	CPU output – Instruction executed – Unused
84 to 88	I	nM[0] to nM[4]	CPU output - Processor mode – Unused
89 to 90	I	RANGEOUT[0] to RANGEOUT[1]	CPU output – Range out from Embedded CE logic – Unused
91 to 94	I	SCREG[0] to SCREG[3]	CPU output – Output from TAP controller – Unused
95	I	nTDOEN	CPU output - TDO Enable – Unused
96 to 127	I	ADDR[0] to ADDR[31]	32-bit ARM7tDMI bus interface from Core Tile to Baseboard – Address bus
128	O	CPA	CPU input – Coprocessor interface (coprocessor absent) – Tied to 1
129	O	CPB	CPU input – Coprocessor interface (coprocessor busy) – Tied to 1

130	I	nCPI	CPU output – Coprocessor interface (coprocessor instruction) – Unused
131 to 134	I/O	TAPSM[0] to TAPSM[3]	CPU output – Output from TAP controller – Unused
135	I	TCK1	CPU output – TCK phase 1 – Unused
136	I	TCK2	CPU output – TCK phase 2 – Unused
137 to 139	I	TEST[0] to TEST[2]	Test chip test outputs - Unused
140	O	TEST[3]	Test chip configuration input – Tied to 0
141 to 143	I	NC[0] to NC[2]	Not connected on the ARM7TDMI
144 to 179	N/A	NC	Not connected to EB FPGA

Table 10.2: Tile Site 1 HDRX for CT7TDMI Designs

10.1.3 Tile Site 2 HDRX

The interface to tile site 2 makes it as easy as possible to add a second CT926EJ-S or CT1136JF-S, although minor changes may be required. If a Logic Tile is used (standard system), two AHB interfaces are routed to it for implementation of AHB masters and slaves.

X[n]	Dir	Signal	Description
0 to 31	I/O	HDATA[0] to HDATA[31]	32-bit AHB from Logic Tile to Baseboard – Data bus
32	O	X_HCLKIN	External HCLK source for a Logic Tile – Tied to 0
33	O	X_REFCLK	Logic Tile reference clock - Sourced from OSCCLK[0]
34	I	X_HCLK	Unused input
35	I	HSIZE[2]	32-bit AHB from Logic Tile to Baseboard – Access size
36 to 37	I	HTRANS[0] to HTRANS[1]	32-bit AHB from Logic Tile to Baseboard – Transaction type
38 to 39	I	HSIZE[0] to HSIZE[2]	32-bit AHB from Logic Tile to Baseboard – Access size
40 to 43	I	HPROT[0] to HPROT[3]	32-bit AHB from Logic Tile to Baseboard – Sideband information
44 to 46	I	HBURST[0] to HBURST[2]	32-bit AHB from Logic Tile to Baseboard – Burst length
47	I	HWRITE	32-bit AHB from Logic Tile to Baseboard – Write/read signal
48	I	HBUSREQ	32-bit AHB from Logic Tile to Baseboard – Bus request signal
49	I	HLOCK	32-bit AHB from Logic Tile to Baseboard – Lock signal
50	O	HREADY	32-bit AHB from Logic Tile to Baseboard – Slave is ready for new transaction
51	O	HGRANT	32-bit AHB from Logic Tile to Baseboard – Master is granted the bus
52 to 53	O	HRESP[0] to HRESP[1]	32-bit AHB from Logic Tile to Baseboard – Response from slave
54	O	BIGENDIN	CPU endianness configuration input – selects endianness at power-up
55	I	BIGENDOUT	CPU endianness status output – shows CPU endianness
56	I	COMMRX	Debug Comms channel interrupts from the Logic Tile, connected to inputs of the interrupt controller.
57	I	COMMTX	
58	O	CONFIGINIT	Test chip configuration signal – Tied to 0

59	I	DBGACK	CPU debug acknowledgement – shows if CPU is in debug state
60	O	DBGEN	CPU debug enable configuration input – Tied to 1
61	O	nFIQ	CPU fast interrupt input
62	O	nIRQ	CPU normal interrupt input
63	O	ARM_nRESET	System reset – HRESETn
64	O	ARM_nPORESET	Power on reset – initializes all the logic on the Logic Tile
65	O	nCONFIGRST	Test chip configuration signal – Tied to 1
66	O	ETMEXTIN	Test chip ETM external trigger input – Tied to 0
67	I	ETMEXTOUT	Test chip ETM external trigger output – Unused
68	O	VINITHI	CPU configuration signal – High vector table (0xFFFF0000) – Tied to 0
69	I	PLLLOCK	Indicates that the PLL inside the Core Tile test chip has locked to the reference
70	O	PLLBYPASS	Test chip configuration input – PLL Bypass – Tied to 1
71	O	INITRAM	CPU configuration input – Enables TCM at reset – Tied to 0
72 to 74	O	HCLKDIV[0] to HCLKDIV[2]	Test chip bus divider ratio CPUCLK/HCLKI – Tied to 0
75	O	EDBGRQ	CPU debug request input – Tied to 0
76	O	DEWPT	CPU external watchpoint input – Tied to 0
77	O	IEBKPT	CPU external breakpoint input – Tied to 0
78 to 83	O	USERIN[0] to USERIN[5]	Test chip user configuration inputs – Tied to 0
84 to 89	I	USEROUT[0] to USEROUT[5]	Test chip user configuration outputs – Unused
90 to 91	O	PLLCTRL[0] to PLLCTRL[1]	Test chip PLL control inputs - Exact function varies between test chips – Tied to 0
92 to 95	O	PLLREFDIV[0] to PLLREFDIV[3]	Test chip PLL reference divider configuration input – Exact function varies between test chips – Tied to 0
96 to 127	I	HADDR[0] to HADDR[31]	32-bit AHB from Logic Tile to Baseboard – Address bus
128	O	TESTSELECT	Test chip configuration input - Enables scan test mode – Tied to 0
129	O	TICSELECT	Test chip configuration input - Enables TIC testing mode – Tied to 0
130 to 137	O	PLLFBDIV[0] to PLLFBDIV[7]	Test chip PLL feedback divider configuration input – Exact function varies between test chips – Tied to 0
138 to 141	O	PLLOUTDIV[0] to PLLOUTDIV[3]	Test chip PLL output divider configuration input – Exact function varies between test chips – Tied to 0
142	I	EXTTRIG	External trigger condition from Core Tile – Unused
143	N/A	Unused	Unused
144 to 179	N/A	NC	Not connected to EB FPGA

Table 10.3: Tile Site 2 HDRX

10.2 HDRY

On tile site 1 an AHB slave interface for an optional Logic Tile stacked on top of the Core Tile is implemented on the Y bus, which is routed straight through the Core Tile.

On tile site 2 an AHB slave can be implemented on a Logic Tile directly on top of EB.

Y[n]	Dir	Signal	Description
0 to 31	I/O	HDATA[0] to HDATA[31]	32-bit AHB from Baseboard to Logic Tile – Data bus
32	N/A	Unused	Unused
33	N/A	Unused	Unused
34	O	DUMMY OUT	Spare output – Used as an OR of all unused inputs
35	I	HREADYOUT	External slaves transfer done output. When HIGH indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer.
36 to 37	O	HTRANS[0] to HTRANS[1]	32-bit AHB from Baseboard to Logic Tile – Transaction type
38 to 39	O	HSIZE[0] to HSIZE[1]	32-bit AHB from Baseboard to Logic Tile – Access size
40 to 43	O	HPROT[0] to HPROT[3]	32-bit AHB from Baseboard to Logic Tile – Sideband information
44 to 46	O	HBURST[0] to HBURST[2]	32-bit AHB from Baseboard to Logic Tile – Burst length
47	O	HWRITE	32-bit AHB from Baseboard to Logic Tile – Write/read signal
48	O	HBUSREQ	32-bit AHB from Baseboard to Logic Tile – Bus request signal
49	O	HLOCK	32-bit AHB from Baseboard to Logic Tile – Lock signal
50	O	HREADY	32-bit AHB from Baseboard to Logic Tile –Indicates to external slave that a transfer has finished on the bus.
51	I	HGRANT	32-bit AHB from Baseboard to Logic Tile – Master is granted the bus
52 to 53	I	HRESP[0] to HRESP[1]	32-bit AHB from Baseboard to Logic Tile – Response from slave
54 to 95	N/A	Unused	Unused
96 to 127	O	HADDR[0] to HADDR[31]	32-bit AHB from Baseboard to Logic Tile – Address bus
128 to 143	N/A	Unused	Unused
144 to 179	N/A	NC	Not connected to EB FPGA

Table 10.4: Tile Site 1 and 2 HDRY

10.3 HDRZ

HDRZ routes miscellaneous control signals to and from tile site 1 and 2. An explanation of these signals and how they are normally used can be found in the EB user guide.

Z[n]	Dir	Signal	Description
0 to 199	N/A	NC	Not connected to EB FPGA
200 to 207	I	Tx_INT[0] to Tx_INT[7]	Interrupt source from tile site 1 and 2. Connected to inputs of the GIC.
208	I	DMACSREQ[0]	DMA single request
209	I	DMACBREQ[0]	DMA burst request
210	I	DMACLSREQ[0]	DMA last single request
211	I	DMACLBREQ[0]	DMA last burst request
212	O	DMACCLR[0]	DMA request acknowledge clear
213	O	DMACTC[0]	DMA terminal count
214	I	DMACSREQ[1]	DMA single request
215	I	DMACBREQ[1]	DMA burst request
216	I	DMACLSREQ[1]	DMA last single request
217	I	DMACLBREQ[1]	DMA last burst request
218	O	DMACCLR[1]	DMA request acknowledge clear
219	O	DAMCTC[1]	DMA terminal count
220	I	DMACSREQ[2]	DMA single request
221	I	DMACBREQ[2]	DMA burst request
222	I	DMACLSREQ[2]	DMA last single request
223	I	DMACLBREQ[2]	DMA last burst request
224 to 226	O	DATACTL[0] to DATACTL[2]	DATACTL[1:0]=11. DATACTL[2]=Tri-state enable signal for HDATA bus.
227	O	SER_DIN	Serial Data into the Core Tile
228	I	SER_DOUT	Serial Data out of the Core Tile
229	O	SER_RST	Serial data mode (startup or normal) for Core Tile Serial Stream
230	O	SER_CLK	Serial data clock for Core Tile Serial Stream
231	N/A	Unused	Tied to 0. Reserved for future use.
232 to 255	N/A	NC	Not connected to EB FPGA

Table 10.5: Tile Site 1 and 2 HDRZ