RealView® LT-XC5VLX330

Logic Tile HBI-0172

User Guide

ARM®

ARM DUI 0365C

RealView LT-XC5VLX330
User Guide

Release Information

Change History

Date	Issue	Confidentiality	Change
July 2007	A	Non-confidential	First release
October 2007	B	Non-confidential	Enhancement to text
April 2011	C	Non-confidential	Enhancement to text

Proprietary Notice

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

This device is test equipment and consequently is exempt from part 15 of the FCC Rules under section 15.103 (c).

CE Declaration of Conformity

CE

The system should be powered down when not in use.

The Logic Tile generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures:

- ensure attached cables do not lie across the card
- reorient the receiving antenna
- increase the distance between the equipment and the receiver
- connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- consult the dealer or an experienced radio/TV technician for help

Note

It is recommended that wherever possible shielded interface cables be used.

Contents
RealView LT-XC5VLX330 User Guide

List of Tables
RealView LT-XC5VLX330 User Guide

List of Figures
RealView LT-XC5VLX330 User Guide

Preface

This preface introduces documentation for the RealView® LT-XC5VLX330 Logic Tile. It contains the following sections:

- *About this document* on page xii
- *Feedback* on page xv.

About this document

This document describes how to set up and use the RealView LT-XC5VLX330 Logic Tile.

Intended audience

This document has been written for experienced hardware and software developers as an aid to developing ARM-based products using Logic Tiles either as a standalone development system, using an Interface Module, or an extended development system, using a baseboard.

Organization

This document is organized into the following chapters:

Chapter 1 ***Introduction***

Read this chapter for an introduction to the Logic Tile.

Chapter 2 ***Getting Started***

Read this chapter for a description of how to set up and start using the Logic Tile.

Chapter 3 ***Hardware Description***

Read this chapter for a description of the hardware architecture of the Logic Tile. This includes clocks, resets, and debug features.

Chapter 4 ***Configuring the FPGA***

Read this chapter for a description of how the FPGA is configured at power-up, the configuration options available, and how to download your own FPGA configurations.

Appendix A ***Pinouts and Specifications***

Refer to this appendix for signal descriptions and connector pinouts

Typographical conventions

The following typographical conventions are used in this book:

italic	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.

bold	Highlights interface elements, such as menu names. Denotes ARM processor signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.
`mono`space	Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.
`monospace italic`	Denotes arguments to commands and functions where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.

Further reading

This section lists related publications by ARM Limited and other companies that provide additional information and examples.

ARM publications

The following publications provide information about related ARM products and toolkits:

- *ARM Integrator/IM-LT1 Interface Module User Guide* (ARM DUI 00187)
- *RealView Platform Baseboard for ARM926EJ-S User Guide* (ARM DUI 0224)
- *RealView Emulation Baseboard User Guide* (ARM DUI 0303)
- *RealView LT-XC2V4000+ User Guide* (ARM DUI 0186)
- *RealView LT-XC4VLX100+ User Guide* (ARM DUI0365)
- *IT1 Interface Tile User Guide* (ARM DUI0188)
- *RealView ICE User Guide* (ARM DUI 0155)
- *AMBA® Specification* (ARM IHI 0011)
- *AMBA 3 AXI Protocol Specification* (ARM IHI 0022).

Note

Refer to the application notes and examples on the *Versatile CD* for more information on using the Logic Tile.

Other publications

The following publication provides information about the clock controller chip used on the Logic Tile:

- *ICS Serially Programmable Clock Source Data Sheet* (ICS307-01/02) MicroClock Division of Integrated Circuit Systems Inc, San Jose, CA. (`www.idt.com`).

Feedback

ARM ARM Limited welcomes feedback both on the RealView LT-XC5VLX330 Logic Tile and on the documentation

Feedback on the baseboard

If you have any comments about this document, 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.

Feedback on the RealView LT-XC5VLX330 Logic Tile

If you have any comments or suggestions about this product, please contact your supplier giving:

- the product name
- an explanation of your comments.

Chapter 1
Introduction

This chapter provides an introduction to the RealView LT-XC5VLX330 Logic Tile. It contains the following sections:

1.1 About the LT-XC5VLX330 Logic Tile

The LT-XC5VLX330 Logic Tile is designed as a platform for developing systems based on *Advanced Microcontroller Bus Architecture* (AMBA™) that use the *Advanced eXtensible Interface* (AXI), *Advanced High-performance Bus* (AHB), *Advanced Peripheral Bus* (APB) peripherals, or custom logic for use with ARM cores.

Note

The Logic Tile must be used with an external board that provides power and JTAG connectors. For standalone operation an Integrator® Interface Module (IM-LT1) is required. The Logic Tile can be used with a baseboard to provide additional development resources. Examples are the *RealView Emulation Baseboard* (EB) or the *RealView Platform Baseboard for ARM926EJ-S* (PB926EJ-S).

Some examples of how the Logic Tile can be used are:

- As a standalone system with an interface module such as the Integrator IM-LT1. The IM-LT1 is used to provide the power and JTAG connection. Implement a processor in the Logic Tile FPGA or use a Core Tile.
- For AHB peripheral development or multi-processor development with a RealView PB926EJ-S baseboard. The Logic Tile can be used to hold custom peripherals or an implementation of a synthesizable core. Use the IT1 Interface Tile for access to the peripheral signals on the Logic Tile.

  Note

  ARM recommends that you use *Application Note 170* as a starting point for your AHB design.

- For AXI peripheral development with a RealView EB (or other AXI baseboard). Fit the Logic Tile to tile site 2 of the EB and fit a Core Tile to tile site 1 to provide your processor of choice.

  Caution

  In an AXI bus implementation using a Logic Tile, normally header HDRX implements the master AXI bus and header HDRY implements an AXI slave bus. Due to pin restrictions at the headers, the EB implements a multiplexed AXI bus at the tile sites. A custom Logic Tile image must include suitable logic to interface to the multiplexed AXI bus. See *RealView Emulation Baseboard User Guide* (DUI 0303) for further details of the AXI interface requirements.

  ARM recommends that you use *Application Note 151* as a starting point for your AXI design.

Figure 1-1 on page 1-5 shows the layout of the Logic Tile.

The LT-XC5VLX330 is supplied fitted with the Xilinx Virtex 5 FPGA (XC5VLX330).

The functionality of the Logic Tile is defined by a configuration image loaded into the FPGA at power-up. *Application Notes* are available that describe how to implement both AHB and AXI peripherals in Logic Tiles. Refer to the documentation supplied with the product on the *Versatile CD* and the *Application Notes* listing at `www.arm.com/documentation` for further details.

You can also download your own configurations to flash using the JTAG connector or the USB debug port if this is provided on the baseboard. It is also possible to load an image directly to the FPGA but directly loaded images are lost when power is removed (see *Reconfiguring the FPGA directly* on page 4-10).

Table 1-1 compares the main features of the Virtex-II, Virtex-4, and Virtex-5 Logic Tiles.

Table 1-1 Comparison of Virtex-II, Virtex-4, and Virtex-5 Logic Tiles

Feature	LT-XC2V 6000/8000	LT-XC4VLX 160/200	LT-XC5VLX 330	Notes
FPGA slices	34k / 47k	68/89k	331k	-
LUT architecture	4	4	6	-
Header X/Y/Z I/O pins	914 / 918	918	918	Equivalent to LT-XC2V8000 Logic Tile.
External clocks	26	21	21	The LT-XC2V6000/8000 **CLK_LOOP3** and **CLK_LOOP4** have been removed because they were not normally used.
Tile clocks	4	4	4	**OSC_CLK[2:0]** and **CLK_24MHz**.
ZBT SRAM	4MB	–	2x16MB	2 blocks of 16MB SRAM added, each block is independent
Boot image	-	SelectMAP x8	BPI x16	32MB of flash memory available.
Config flash images	2	2	2	-
User LEDs, switches	4	8	8	-
User push button	1	1	1	-
Config / Normal JTAG	Yes	Yes	Yes	-
Header X/Y/Z Foldover	Upper	Upper, Lower	Upper, Lower	The headers for the LT-XC4VLX160/200 and LT-XC5VLX330 remain in the same structure. X/Y/Z each fold by 36 bits.
Variable I/O voltage	HDRX, HDRY	HDRX	HDRX	-
Resets	2	2	2	-
Over temperature indicator	No	Yes	Yes	Red LED indicator at board edge is activated if FPGA temperature reaches 80°C.
Bit file encryption	Triple DES	256b AES	256b AES	The Virtex 4 and Virtex 5 keys are not programmed in production.

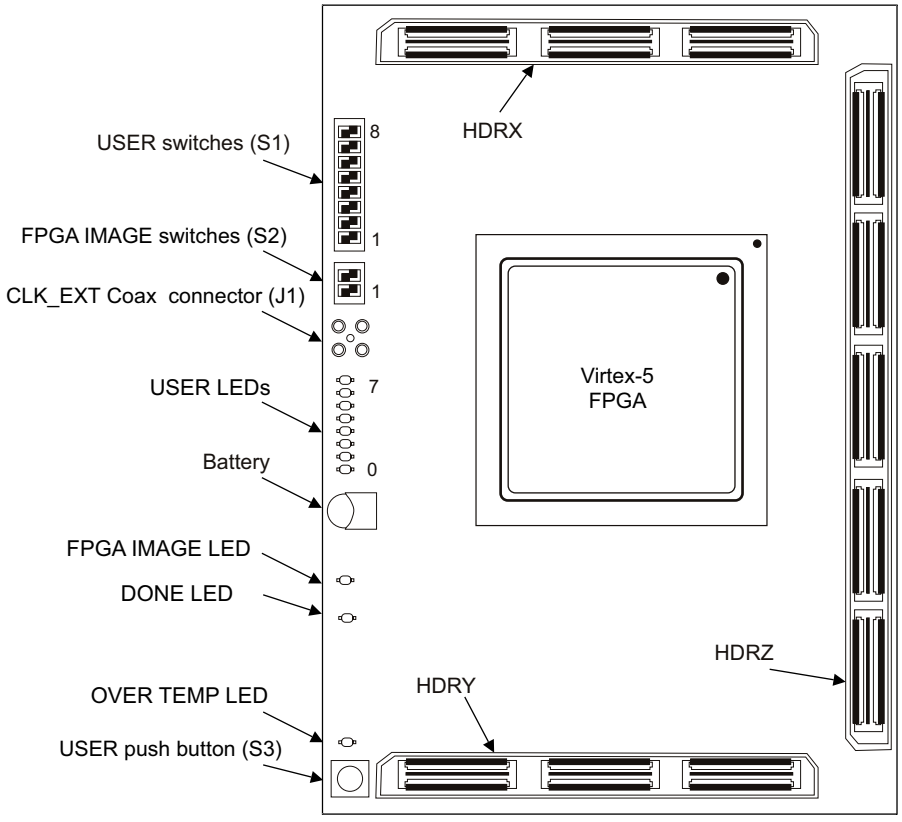

Figure 1-1 LT-XC5VLX330 layout (heatsink not shown)

1.2 Logic Tile architecture

Figure 1-2 shows the architecture of the Logic Tile.

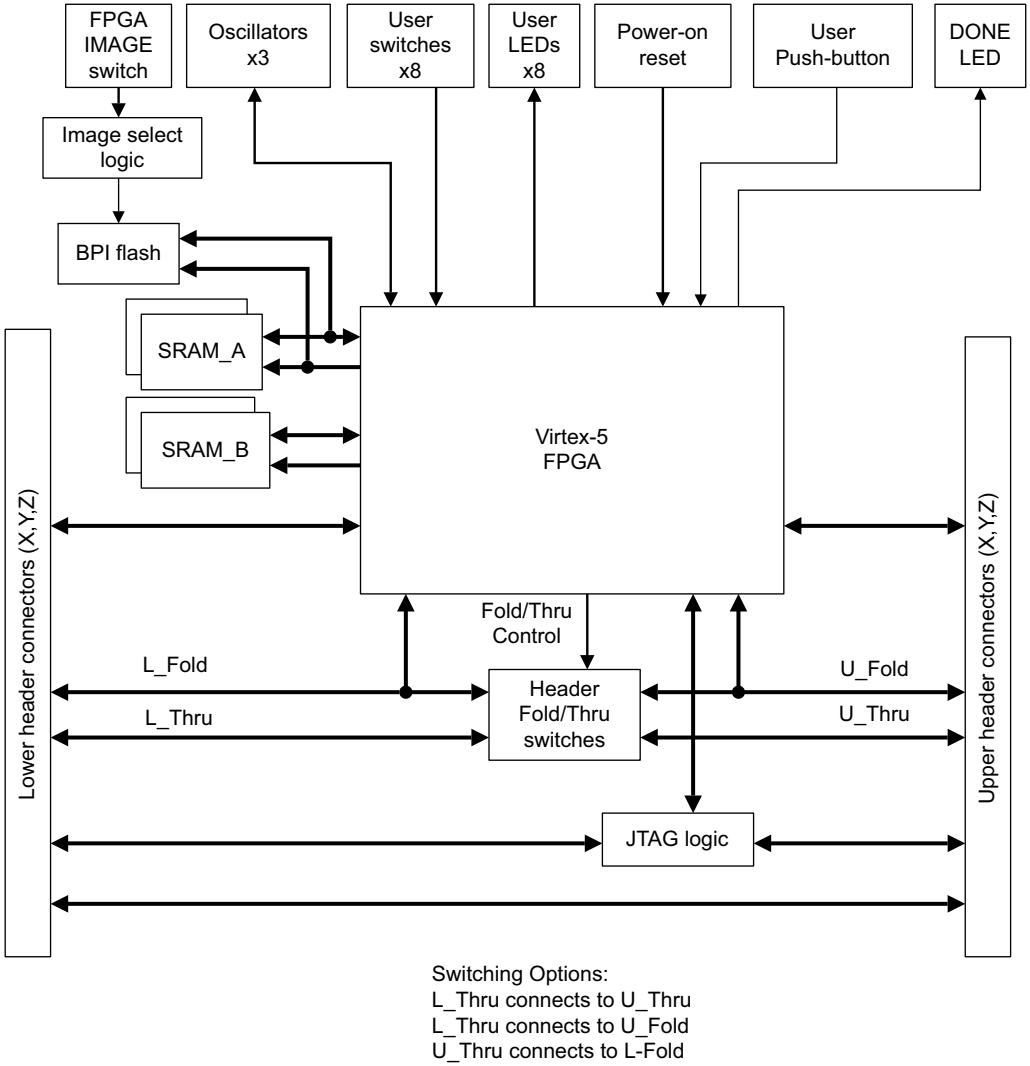

Figure 1-2 System architecture

The Logic Tile comprises the following:

- Xilinx Virtex-5 FPGA
- flash memory for storing FPGA configurations
- clock generators and reset sources
- general purpose switches
- general purpose LEDs
- battery for AES encryption key
- connectors to other tiles
- ZBT memory for user implementations.

These components are discussed in detail in Chapter 3 *Hardware Description*.

1.3 Precautions

This section contains advice about how to prevent damage to your Logic Tile.

1.3.1 Ensuring safety

The Logic Tile is powered from a 3.3V DC and a 5V DC supply. Power is supplied to Logic Tiles through the header connectors. The board that the Logic Tile is mounted on must supply 3.3V DC and 5V DC.

Warning

Do not use the board near equipment that is sensitive to electromagnetic emissions (such as medical equipment).

1.3.2 Preventing damage

The Logic Tile is intended for use within a laboratory or engineering development environment. It is supplied without an enclosure which leaves the board sensitive to electrostatic discharges and allows electromagnetic emissions.

Caution

To avoid damage to the board, observe the following precautions.

- Never subject the board to high electrostatic potentials. Observe *ElectroStatic Discharge* (ESD) precautions when handling any board.
- Always wear a grounding strap when handling the board.
- Only hold the board by the edges.
- Avoid touching the component pins or any other metallic element.
- Ensure that the voltage on the pins of the FPGA and interface circuitry on all connected Logic Tiles is at the correct level. If you are using a PB926EJ-S, or an EB baseboard, some of the Logic Tile FPGA signals are connected directly to the baseboard.
- FPGA pins connected to an external signal source must not be configured as outputs.
- Do not use the board near a transmitter of electromagnetic emissions.

Chapter 2
Getting Started

This chapter describes how to set up and start using the Logic Tile. It contains the following sections:

2.1 Using the Logic Tile with an Interface Module or with a baseboard

The Logic Tile must be used with an external board that provides power and JTAG connectors (for example, an Integrator IM-LT1 Interface Module, or the RealView PB926EJ-S baseboard).

Figure 2-1 shows a standalone system of a Logic Tile mounted onto an IM-LT1 Interface Module. In this system a processor core image (for example an SSM) is loaded directly into the FPGA. See *Integrator IM-LT1 Interface Module User Guide* (DUI 0187) for information on adding Logic Tiles to the IM-LT1. In this configuration, power and JTAG connection is provided by the IM-LT1.

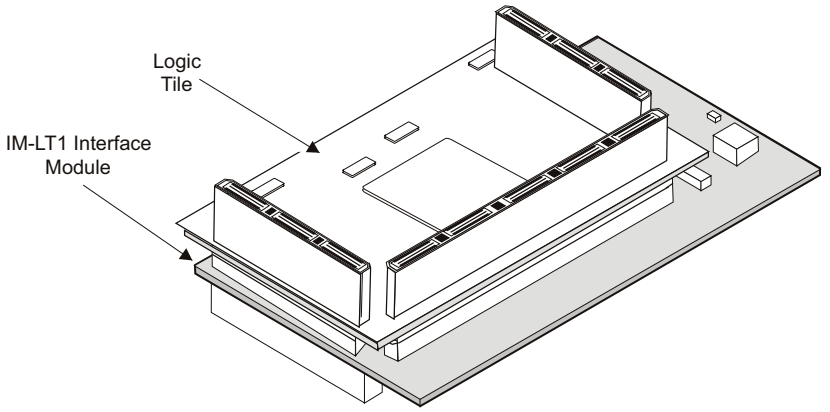

Figure 2-1 Standalone operation with processor in FPGA

Figure 2-2 on page 2-3 shows an example system of a Logic Tile mounted onto a RealView PB926EJ-S baseboard. See *RealView Platform Baseboard for ARM926EJ-S User Guide* (DUI 0224) for information on adding Logic Tiles to the PB926EJ-S. In this configuration, power and JTAG connection is provided by the PB926EJ-S.

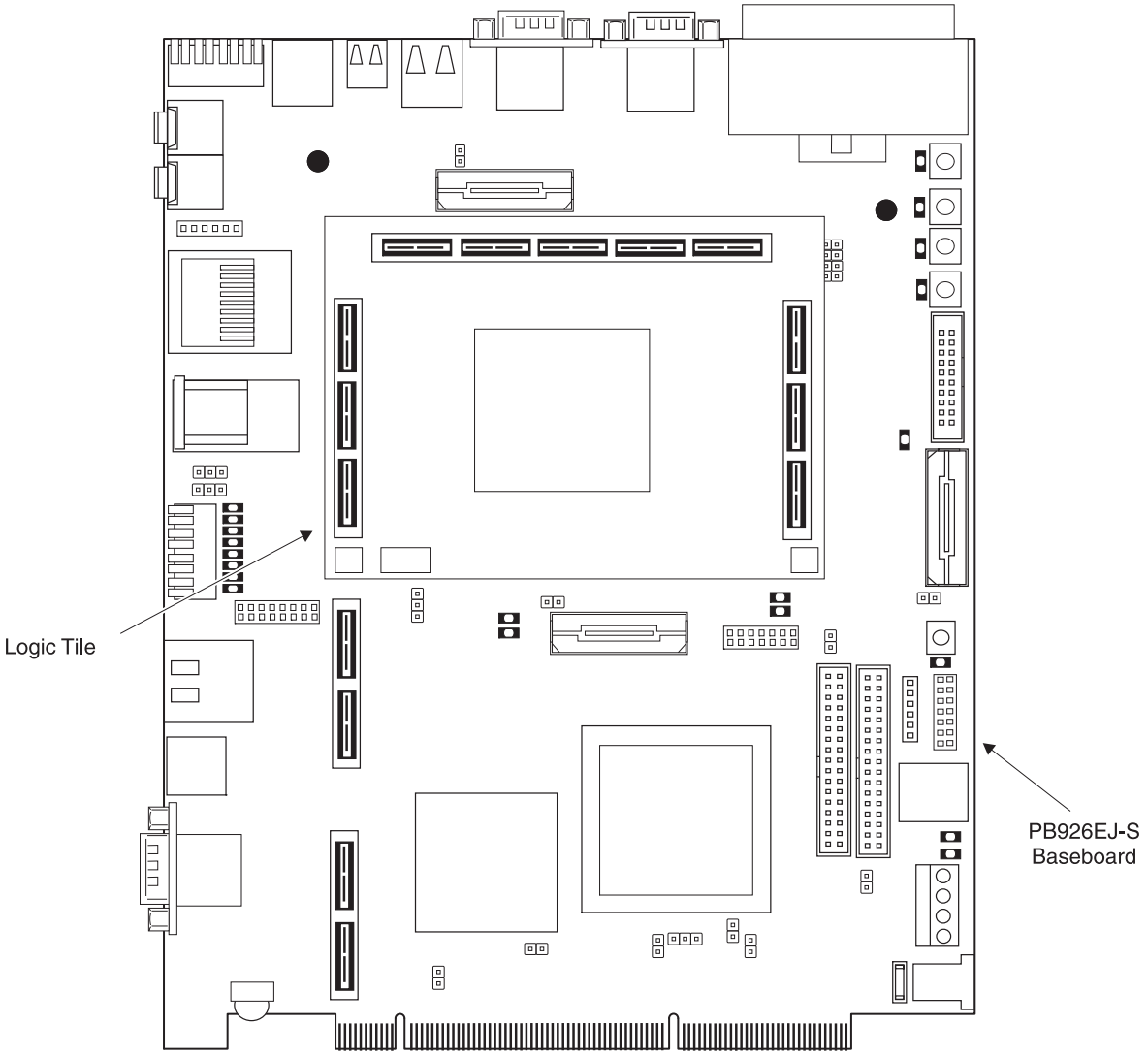

Figure 2-2 Logic Tile mounted on a PB926EJ-S

Figure 2-3 on page 2-4 shows an example system of a Logic Tile mounted onto tile site 2 of a RealView Emulation Baseboard (EB) with a Core Tile mounted onto tile site 1. See the *RealView Emulation Baseboard User Guide* (DUI 0303) for information on adding Logic Tiles and Core Tiles to the EB. In this configuration, power and JTAG connection is provided by the EB.

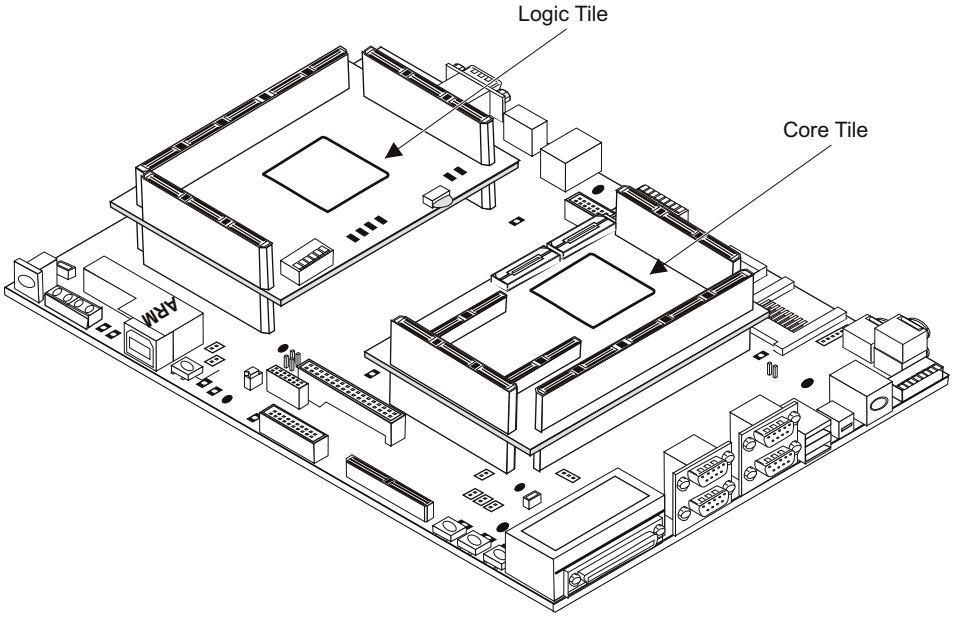

Figure 2-3 Logic Tile and Core Tile mounted on an EB

2.2 Switches and LEDs

This section describes the switches and LEDs on the Logic Tile. The locations of the LEDs and switches are illustrated in Figure 2-4.

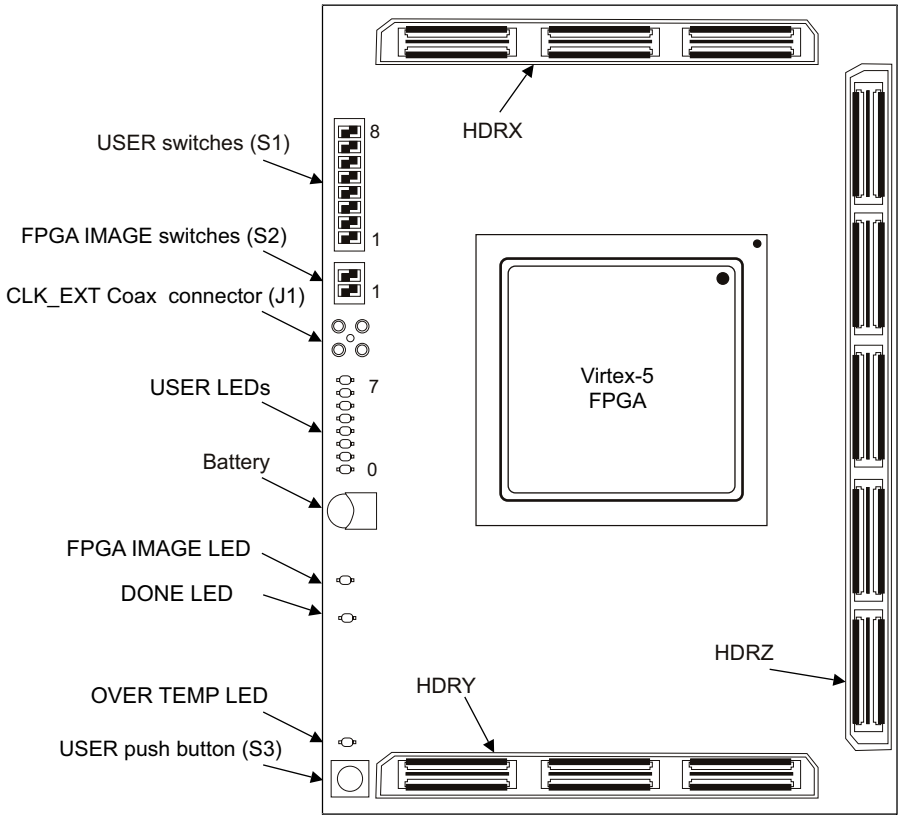

Figure 2-4 Switches and LEDs

2.2.1 Switches

There are two DIP switch banks fitted to the Logic Tile:

Note

The DIP switches used are labelled from one not zero, because that is the numbering convention used in this guide.

S1[8:1] These switches are general purpose switches that are user-defined.

S2[2:1] The functions of these switches are:

SW2[1] Support for encrypted bitfile

SW2[2] Select between unencrypted images.

Note

The HBI-0172 Logic Tile uses a 16-bit FPGA configuration which is not supported by the Virtex 5 FPGA in encrypted mode. To overcome this problem, the top byte of the FLASH memory is programmed with zeros and only the lower byte is programmed with data forcing the FPGA to use an 8-bit configuration mode.

To achieve this, the bit file is padded with zeros. This results in a 20MB bit file image which has to be programmed into both banks of the FLASH. For the FPGA to have access to both areas of FLASH during configuration, the HBI-1072C Logic Tile has been modified to connect to A23 (RAM_A_SD57) of the FPGA.

This means that S2[1] no longer selects the image defined by the HDRZ signal FLASH_IMAGE but now enables the FPGA to access both memory banks.

To select an encrypted bitfile on the HBI-1072C, set:

S2[1] ON

S2[2] OFF

To select a non-encrypted bitfile on the HBI-1072C, set:

S2[1] OFF

S2[2] Selects between image 0 and 1.

For a full description of FPGA configuration image selection, see *Configuring the FPGA from flash* on page 4-8.

S3 This push-button is a general-purpose switch.

The signal is buffered and connected to the FPGA. If the push-button is depressed, the signal to the FPGA is LOW.

2.2.2 LEDs

There are three individual status LEDs and a single bank of eight User LEDs:

FPGA IMAGE This LED indicates the image that is loaded into the FPGA. It is lit if the image from the top half of the flash memory is selected and off if the image from the bottom half of the flash memory is selected.

DONE This LED indicates that FPGA configuration has completed.

OVER TEMP This LED indicates that the FPGA is operating above a safe temperature. The temperature monitor lights the LED once the FPGA temperature exceeds 80°C.

Caution

If the temperature monitor lights, turn off the supply immediately. Prolonged over heating may damage the board or the Virtex-5 device itself. Check for all possible causes such as excessive I/O loading, I/O contention, or the board being over-clocked. Whenever possible, remove the cause of overheating before re-powering the board.

USER[7:0] These LEDs are general purpose indicators and are user-defined.

The FPGA drives the LEDs through butters with 330Ω resistors between the buffer output and the LED. A logic high on the FPGA output lights the LED.

2.3 Using RealView ICE or other JTAG equipment

The JTAG signals for the Logic Tile are routed through the tile headers to the boards above and below the tile. There is no JTAG connector on the Logic Tile. Connect the JTAG equipment to the JTAG 20-way box header on the baseboard or Interface Module being used.

If multiple Logic Tiles are stacked on an Interface Module, the JTAG equipment is always connected to the Interface Module and the signals are routed upwards to the top tile and then back down to the Interface Module. Refer to the JTAG section in the *Integrator IM-LT1 User Guide* for details.

Use RealView ICE to program the configuration flash or directly load the FPGA image.

The *RealView Platform Baseboard* and the *RealView Emulation Baseboard* also support programming of the configuration flash or directly loading the FPGA image using the USB port.

Note

USB debug using RealView ICE is not supported by the *RealView Emulation Baseboard*.

Third-party JTAG tools can be used to configure and debug the FPGA design by connecting them to the 20-way box header on the baseboard or interface board.

Caution

Third-party tools can only be used to program a configuration directly into the FPGA, they cannot be used to program an image into the flash memory. An image that has been directly loaded into the FPGA is lost when power is removed.

Chapter 3
Hardware Description

This chapter describes Logic Tile hardware. It contains the following sections:

3.1 FPGA

The Logic Tile is fitted with a Xilinx Virtex-5 FPGA (XC5VLX330-1FFG1760). The assignment of the input/output banks and JTAG implementation are described in the following sections:

- *FPGA I/O arrangement* on page 3-4
- *JTAG and the FPGA* on page 3-5.

For information about how the FPGA data is loaded, see Chapter 4 *Configuring the FPGA*.

At power-up the FPGA loads configuration data from the on-board flash memory. Parallel data from the flash is streamed into the configuration port of the FPGA, see *FPGA Configuration from flash memory* on page 3-5. It is also possible to load an image directly into the FPGA through the JTAG connector, but the image is lost when power is turned off.

Figure 3-1 on page 3-3 is a simplified view of the tile and illustrates the function of the FPGA and shows how it connects to the other devices in the Logic Tile.

The routing of the L_Fold, L_Thru, U_Fold, and U_Thru signals depends on the Fold/Thru switches, see *Foldover* on page 3-15.

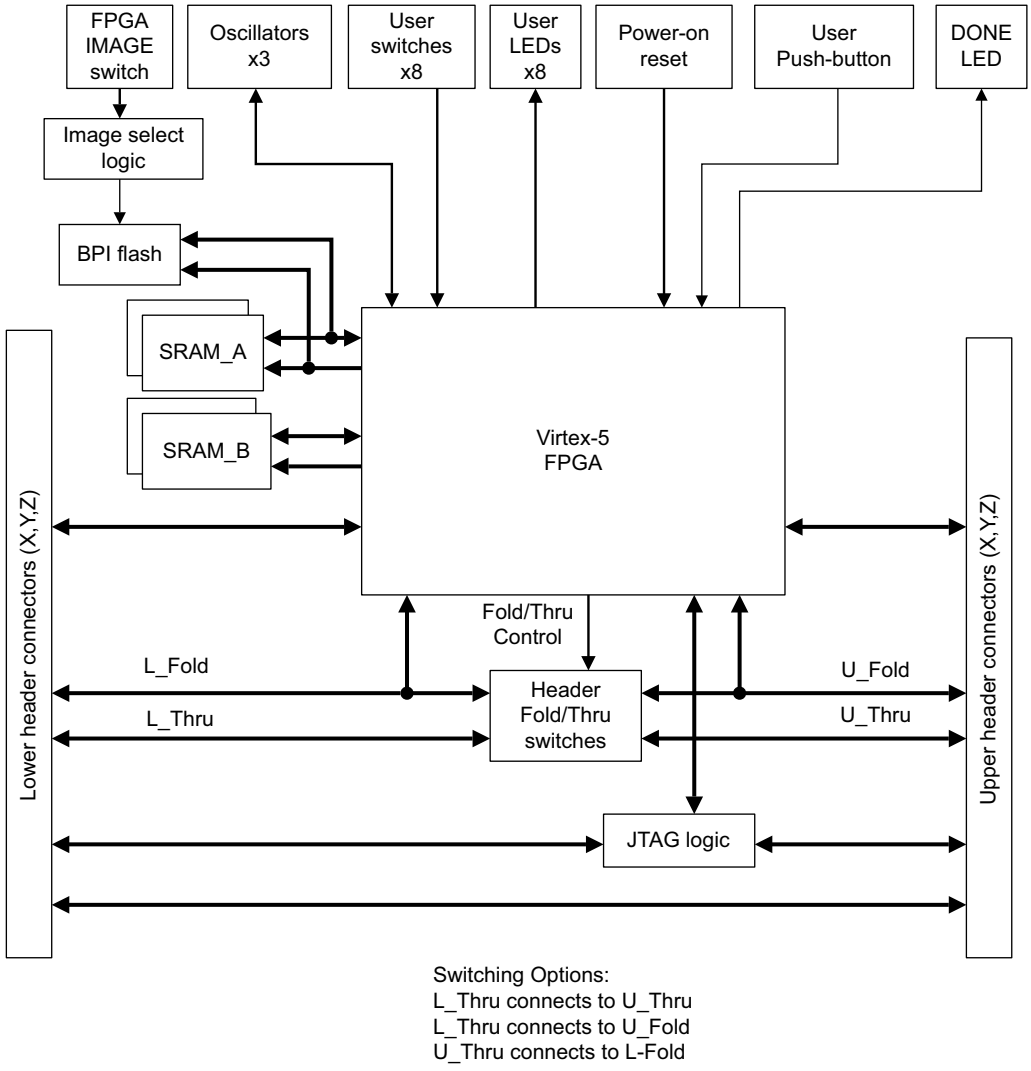

Figure 3-1 LT-XC5VLX330 block diagram

3.1.1 FPGA I/O arrangement

The FPGA input/output pins are organized into 17 banks. Bank 0 is used for JTAG, configuration and miscellaneous functions and Banks 1 to 16 are routed to headers HDRX, HDRY, and HDRZ to support tile interconnection, The upper and lower HDRX, HDRY, and HDRZ header routing is shown in Figure 3-2 and Table 3-5 on page 3-13 lists the bank I/O signal levels. See also *Header signals* on page 3-9 for more details on I/O pins and header connections.

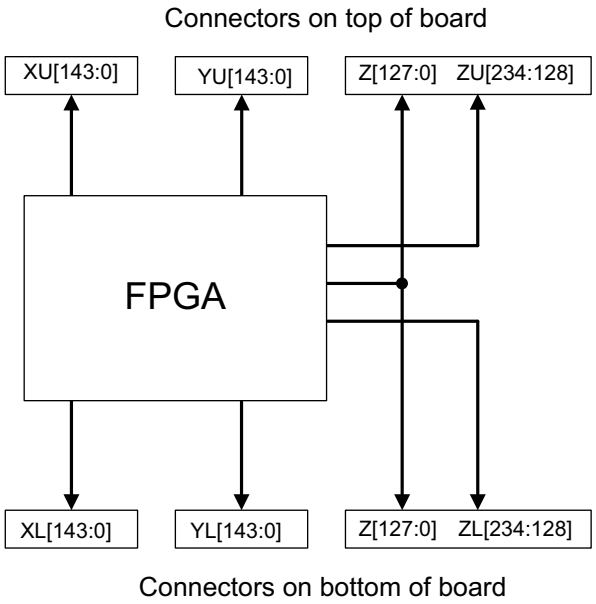

Figure 3-2 LT-XC5VLX330 top and bottom header connections

3.1.2 JTAG and the FPGA

The tile contains configuration and debug JTAG chains.

The two JTAG chains are completely separate:

- The configuration chain connects to the TAP controllers of all programmable devices in the system. You can use the TAP controller on the FPGA for example, to download new FPGA configurations.
- The debug JTAG chain is routed through I/O pins on the FPGA. If you implement a design that does not include a debug TAP controller, ensure that the FPGA design passes the debug JTAG signals through to the next tile in the stack.

  If you have loaded an ARM CPU core image into the FPGA, you can use a JTAG debugger such as RealView ICE connected to the JTAG connector on the baseboard or interface board to debug application software running in the tile.

Both scan chains are available through separate signal groups on the HDRZ tile header. The configuration JTAG signals are prefixed **C_**, and the debug JTAG signals are prefixed **D_**. For example, **C_TDI** is the configuration *JTAG test data in* signal and **D_TDO** is the debug *JTAG test data out* signal. See *HDRZ signals* on page A-12 for details of the available JTAG signals.

3.1.3 FPGA Configuration from flash memory

The configuration FPGA has the following modes of operation:

Configuration mode for flash programming

The FPGA sets the SRAM signals **RAM_A_SnCE[1:0]** and **RAM_B_SnCE[1:0]** to HIGH to disable the SRAM devices and sets **FnCE** and **FnWE** LOW to enable the flash memory. The flash memory address, data, and control signals are passed from the FPGA using the **RAM_A_SA**, **RAM_A_SD**, and **RAM_A_SADVnLD** signals. A simplified diagram is shown in Figure 3-3 on page 3-7.

Note

Use the *Progcards utility* to program the flash using the JTAG port on the baseboard. If provided, the USB debug or Config port on the baseboard can also be used to program the flash. The Progcards utility first loads a flash programmer image into the FPGA, then writes the bit file to the flash memory. You can also use this utility to verify the flash image against the bit file. See *Downloading new FPGA configurations to flash memory* on page 4-9 for details.

Configuration mode for FPGA programming

The JTAG interface is used to directly load a new image to the FPGA.

Note

The FPGA image is lost if the power is removed.

Debug mode for normal operation

The FPGA image must be reloaded after every power-on sequence. The FPGA sets the SRAM signals **RAM_A_SnCE[1:0]** and **RAM_B_SnCE[1:0]** to HIGH to disable the SRAM devices and sets **FnCE** LOW to enable the flash memory. The FPGA sets **FnWE** HIGH because the image is being transferred from the flash memory to the FPGA. The flash memory address, data, and control signals are passed from the FPGA using the **RAM_A_SA**, **RAM_A_SD**, and **RAM_A_SADVnLD** signals. The image to load from flash memory is determined by the image select logic.

After FPGA configuration, the FPGA sets **FnCE** HIGH to disable the flash memory. The **RAM_A_SA** and **RAM_A_SD** signals are then used only for normal SRAM functions. The SRAM signals **RAM_A_SnCE[1:0]** and **RAM_B_SnCE[1:0]** are used to enable the SRAM devices as required.

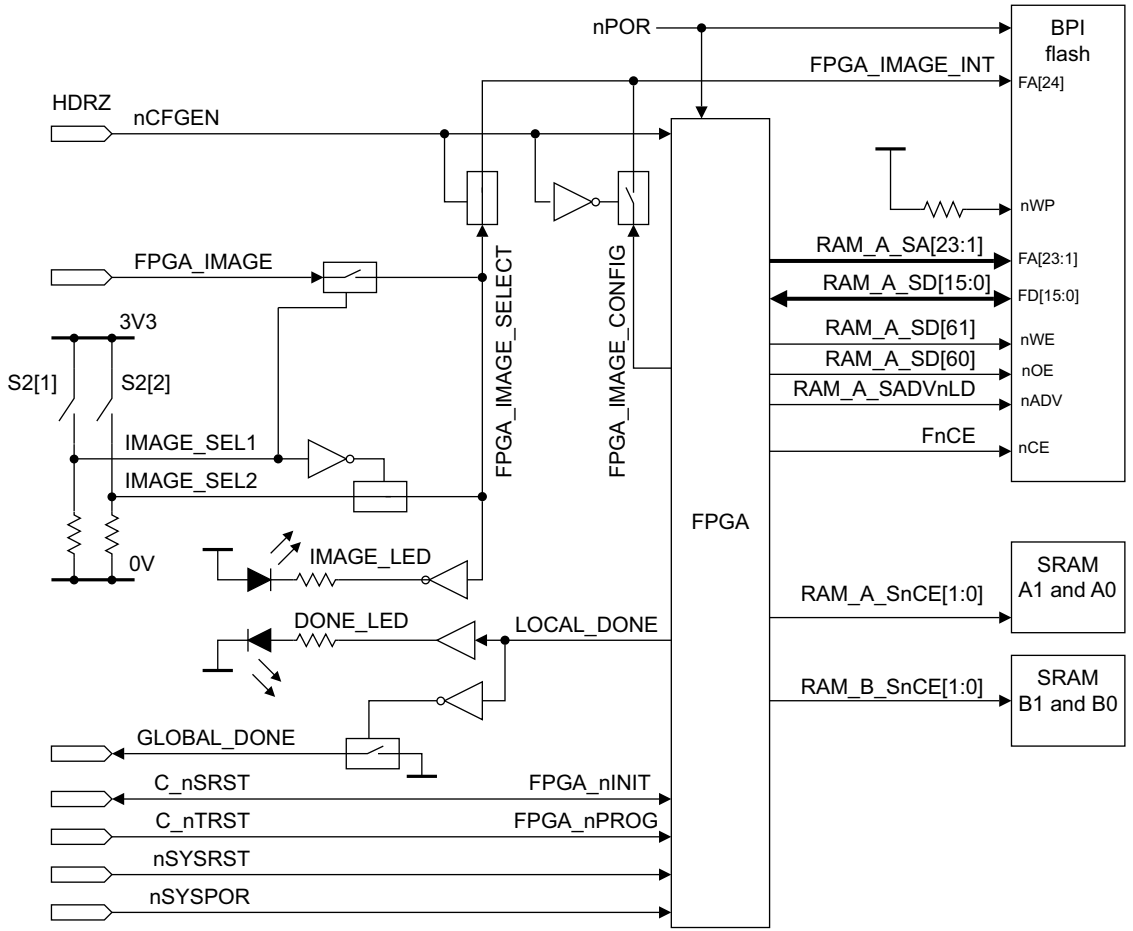

Figure 3-3 Flash memory programming

The timing sequence for loading an image from flash is shown in Figure 3-4.

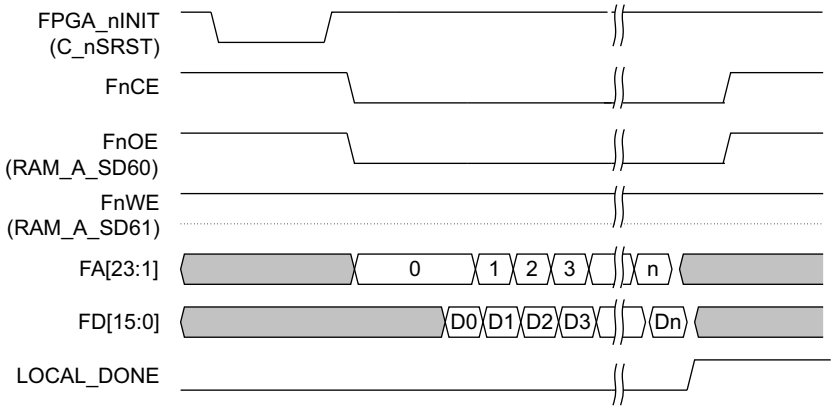

Figure 3-4 Timing sequence for loading the FPGA image from flash

3.2 Header signals

This section gives an overview of the signals present on the header connectors. See *About the LT-XC5VLX330 Logic Tile* on page 1-2 for details of the board layout.

There are three headers on the top and bottom of the tile. The HDRX and HDRY headers are 180-way and the HDRZ connectors are 300-way. Signals that connect only to an upper header are identified by a U (for example, **YU143**), while signals that connect only to a lower header are identified by an L (for example **XL179**). Signals that go through both headers are not identified by a U or L (for example **Z130**). Figure 3-5 on page 3-16 shows a simplified view of the header signal routing (clock, control, and JTAG signals on HDRZ are not shown).

Note

There is no correspondence between the header pin numbers and the generic signal numbers. For example, HDRY has generic signal YU113 on pin 48 of the upper connector.

The LT-XC5VLX330 is fitted with a Virtex-5 XC5VLX330-1FFG1760 and provides a maximum of 1200 I/O pins:

- 918 of the pins are uncommitted and are connected directly to the Logic Tile headers for use in the user design. Table 3-1 lists the distribution of the available I/O pins.

Table 3-1 I/O pin distribution

Header	Upper	Lower
HDRX	144	144
HDRY	144	144
HDRZ	107	107
HDRZ through	128	128

- 186 of the FPGA pins are used for clocks, switches, LEDs, and local control functions.
- 6 FPGA pins are unused and not connected to any device or connector.

For the signals on the upper and lower pins:

- Some upper and lower pins are connected together and pass through signals that are not connected to any devices on the tile (for example the **CLK_UP_THRU** signal on pin 142 of lower HDRZ and pin 138 of upper HDRZ).
- Some upper and lower pins are connected together but are also connected to devices on the tile. For example, the **CLK_GLOBAL** signal on pin 150 of the upper and lower Z header is connected for local use or local generation as required.
- Some upper and lower pins are only connected to the FPGA (for example **YU[143:36]** and **YL[143:0]**). The FPGA can be programmed to pass a modified version of the input signal on one pin to an output signal on a pin on the other side of the board, or the FPGA can treat the signals as completely independent.
- Some pins on the lower connector can be connected to signals that are normally only available on the upper connector (See *Foldover* on page 3-15).
- Some pins on the upper connector can be connected to signals that are normally only available on the lower connector (See *Foldover* on page 3-15).

The header locations and pin numbering are shown in *Header connectors* on page A-2.

Caution

The FPGA can be damaged if pins configured as outputs (on connected Logic Tiles or baseboards) are connected and output different logic levels.

Also, the signal input and output levels for many of the FPGA signals can be determined by an attached tile (see *Variable I/O levels* on page 3-13).

Table 3-2 lists the Virtex-5 FPGA external clocks I/O pins. See *Clock architecture* on page 3-28 for further details on clocks. The programmable clock generators also provide **CLK_24MHZ_FPGA**, **CLK0**, **CLK1**, and **CLK2** signals to the FPGA.

Table 3-2 Virtex-5 FPGA external clocks

Pin name
CLK_POS_DN_OUT
CLK_NEG_DN_OUT
CLK_POS_UP_IN
CLK_NEG_UP_IN
CLK_IN_MINUS2
CLK_IN_MINUS1
CLK_GLOBAL_IN
CLK_GLOBAL_OUT
CLK_POS_DN_IN
CLK_NEG_DN_IN
CLK_POS_UP_OUT
CLK_NEG_UP_OUT
CLK_IN_PLUS2
CLK_IN_PLUS1
CLK_EXTERN
CLK_OUT_TO_BUF
CLK_BUF_LOOP
CLK_LOOP1_OUT
CLK_LOOP0_OUT
CLK_LOOP1_IN
CLK_LOOP0_IN

Table 3-3 lists the Virtex-5 I/O pins related to reset. See *Reset control* on page 3-43 for further details on resets. (The JTAG signals **D_nSRST** and **D_nTRST** can also function as reset signals to the FPGA.)

Table 3-3 Virtex-5 resets

Pin name
nSYSPOR
nSYSRST
nPOR (generated on the Logic Tile)

Table 3-4 lists the Virtex-5 JTAG I/O pins. See *JTAG support* on page 3-48 for further details on JTAG.

Table 3-4 Virtex-5 FPGA JTAG

Pin name
D_nSRST
D_nTRST
FPGA_D_TDO
D_TDO_IN
FPGA_D_TCK
FPGA_D_TMS
FPGA_D_RTCK

3.2.1 Variable I/O levels

All HDRZ and HDRY connector signals are fixed at a 3.3V I/O signalling level.

The XU and XL I/O signalling levels are set to 3.3V by 0Ω links on the tile. You can however, remove the links and allow the VCCO blade of a corresponding connector on a plugged-in board to set the signal level. The signal levels for the FPGA banks are listed in Table 3-5. (All ARM Logic Tiles operate at a 3.3 V I/O signal level, but custom tiles may use a different signal level and supply the voltage to the VCCO blade.)

Table 3-5 FPGA I/O bank signal level

IO bank	Signal Level	Source
Bank 0 - 2, Bank 5 - 7, Bank 9 - 11, Bank 13, Bank 15	**3V3**	See *HDRX, HDRY, and HDRZ (upper) pin numbering* on page A-2 for 3V3 power blade positions.
Bank 14, Bank 20, Bank 26, Bank 30	**VCCO1**	HDRX upper header power blade (P0-P3). Tied to 3V3 if 0Ω link R44 is fitted (default). Tied to 2V5 if 0Ω link R42 is fitted instead of R44 (not fitted at manufacturing by default). Remove both R44 and R42 to supply VCCO from the tile above.
Bank 12, Bank 16, Bank 18, Bank 22	**VCCO2**	HDRX lower header power blade (P0-P3). Tied to 3V3 if 0Ω link R45 is fitted (default). Tied to 2V5 if 0Ω link R43 is fitted instead of R45 (not fitted at manufacturing by default). Remove both R45 and R43 to supply VCCO from the tile below.

Caution

If you provide VCCO from an adjacent tile, you must remove the relevant 0Ω resistors.

If you require any configuration resistor to be removed or changed, it is recommended that the work be carried out by a skilled technician with experience in circuit board soldering. If any board malfunction is proved as a result of any modification to the board or components, this immediately invalidates the warranty and could lead to costs from ARM for repair or replacement.

Do not fit both the 3V3 and 2V5 resistors.

All pins on the lower HDRX connector must operate at 3.3V signal levels if the tile is used with an EB, PB926EJ-S or IM-LT1

3.2.2 Foldover

Because the tile headers have more I/O pins than the FPGA can support, some header pins are not normally connected to the FPGA, however, switches are provided to:

- connect the upper and lower pins together (thru)
- connect some signals on the upper header to unused pins on the lower header (upper foldover)
- connect some signals on the lower header to unused pins on the upper header (lower foldover)

Foldover provides more I/O from the tile to support future platforms. See Figure 3-5 on page 3-16 for a simplified block diagram of the foldover logic and Table 3-6 on page 3-19 through to Table 3-11 on page 3-26 for a detailed list of the foldover and thru connections.

The foldover switches are controlled by outputs from the FPGA. Your FPGA design must instantiate the HDL that implements the serial Foldover control. An example implementation is provided on the CD that accompanies the tile.

Figure 3-6 on page 3-17 shows an example of using the foldover switches to increase the available connections between the FPGAs in a two tile stack. Pins YU[35:0} on the top LT-XC5VLX330 are unused by the stack and the FPGA pins that connect to them are re-routed back down the stack for access by the lower Logic Tile using the upper foldover and lower foldover switches and the spare HDRY header pins Y[179:144].

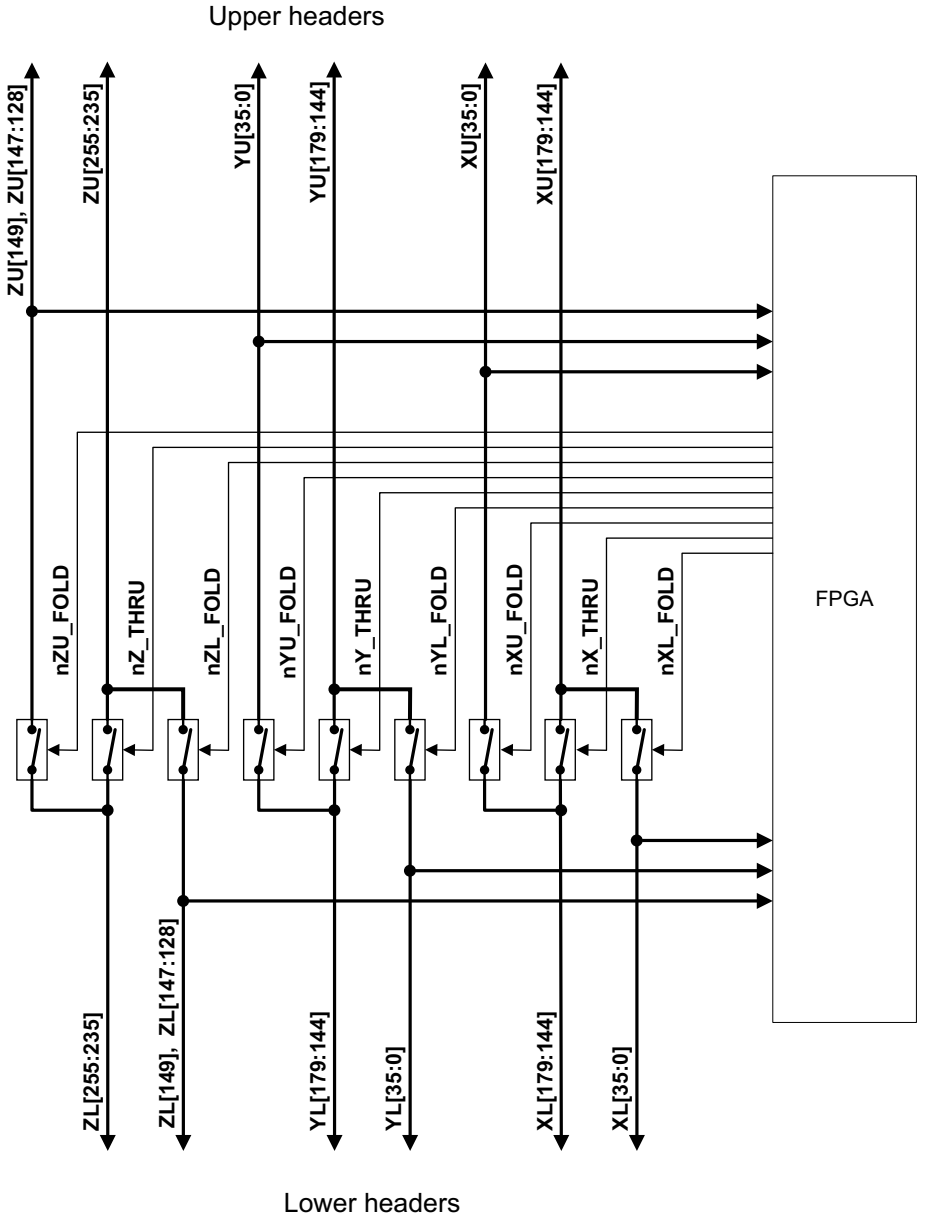

Figure 3-5 Simplified view of foldover and thru signals

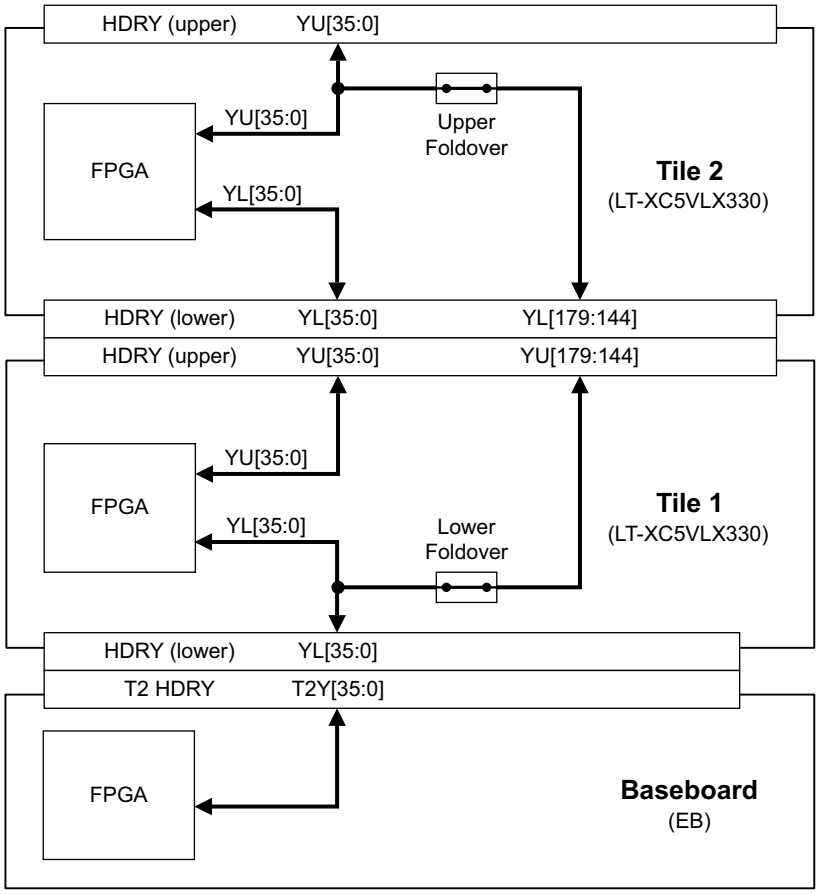

Figure 3-6 Two tile foldover example

Figure 3-7 on page 3-18 shows an example of using the foldover switches to increase the available connections between the top and bottom Logic Tile in a three tile stack. Pins YU[35:0] on the top LT-XC5VLX330 are unused by the stack and so the FPGA pins that connect to them are re-routed back down the stack for access by the bottom Logic Tile using the upper foldover, thru, and lower foldover switches and the spare HDRY header pins Y[179:144].

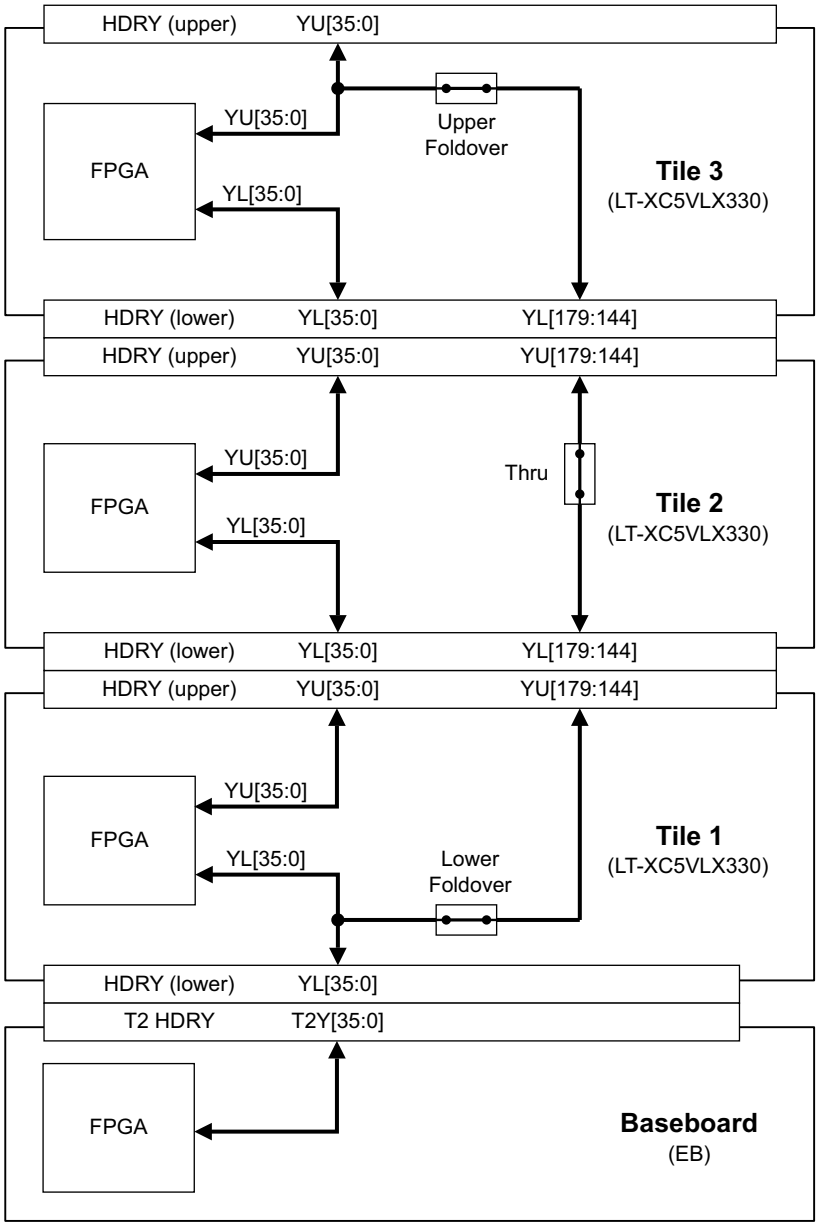

Figure 3-7 Three tile foldover example

The upper foldover connections for HDRX are shown in Table 3-6.

Table 3-6 HDRX upper foldover connection

Lower header	Foldover (nXU_FOLD LOW)	Thru (nX_THRU LOW)
XL178	XU0	XU178
XL179	XU1	XU179
XL176	XU2	XU176
XL177	XU3	XU177
XL174	XU4	XU174
XL175	XU5	XU175
XL172	XU6	XU172
XL173	XU7	XU173
XL170	XU8	XU170
XL171	XU9	XU171
XL168	XU10	XU168
XL169	XU11	XU169
XL166	XU12	XU166
XL167	XU13	XU167
XL164	XU14	XU164
XL165	XU15	XU165
XL162	XU16	XU162
XL163	XU17	XU163
XL160	XU18	XU160
XL161	XU19	XU161
XL158	XU20	XU158
XL159	XU21	XU159
XL156	XU22	XU156

Table 3-6 HDRX upper foldover connection (continued)

Lower header	Foldover (nXU_FOLD LOW)	Thru (nX_THRU LOW)
XL157	XU23	XU157
XL154	XU24	XU154
XL155	XU25	XU155
XL152	XU26	XU152
XL153	XU27	XU153
XL150	XU28	XU150
XL151	XU29	XU151
XL148	XU30	XU148
XL149	XU31	XU149
XL146	XU32	XU146
XL147	XU33	XU147
XL144	XU34	XU144
XL145	XU35	XU145

The lower foldover connections for HDRX are shown in Table 3-7

Table 3-7 HDRX lower foldover connection

Upper header	Foldover (nXL_FOLD LOW)	Thru (nX_THRU LOW)
XU178	XL0	XL178
XU179	XL1	XL179
XU176	XL2	XL176
XU177	XL3	XL177
XU174	XL4	XL174
XU175	XL5	XL175
XU172	XL6	XL172

Table 3-7 HDRX lower foldover connection (continued)

Upper header	Foldover (nXL_FOLD LOW)	Thru (nX_THRU LOW)
XU173	XL7	XL173
XU170	XL8	XL170
XU171	XL9	XL171
XU168	XL10	XL168
XU169	XL11	XL169
XU166	XL12	XL166
XU167	XL13	XL167
XU164	XL14	XL164
XU165	XL15	XL165
XU162	XL16	XL162
XU163	XL17	XL163
XU160	XL18	XL160
XU161	XL19	XL161
XU158	XL20	XL158
XU159	XL21	XL159
XU156	XL22	XL156
XU157	XL23	XL157
XU154	XL24	XL154
XU155	XL25	XL155
XU152	XL26	XL152
XU153	XL27	XL153
XU150	XL28	XL150
XU151	XL29	XL151
XU148	XL30	XL148

Table 3-7 HDRX lower foldover connection (continued)

Upper header	Foldover (nXL_FOLD LOW)	Thru (nX_THRU LOW)
XU149	XL31	XL149
XU146	XL32	XL146
XU147	XL33	XL147
XU144	XL34	XL144
XU145	XL35	XL145

The upper foldover connections for HDRY are shown in Table 3-8.

Table 3-8 HDRY upper foldover connection

Lower header	Foldover (nYU_FOLD LOW)	Thru (nY_THRU LOW)
YL178	YU0	YU178
YL179	YU1	YU179
YL176	YU2	YU176
YL177	YU3	YU177
YL174	YU4	YU174
YL175	YU5	YU175
YL172	YU6	YU172
YL173	YU7	YU173
YL170	YU8	YU170
YL171	YU9	YU171
YL168	YU10	YU168
YL169	YU11	YU169
YL166	YU12	YU166
YL167	YU13	YU167
YL164	YU14	YU164

Table 3-8 HDRY upper foldover connection (continued)

Lower header	Foldover (nYU_FOLD LOW)	Thru (nY_THRU LOW)
YL165	YU15	YU165
YL162	YU16	YU162
YL163	YU17	YU163
YL160	YU18	YU160
YL161	YU19	YU161
YL158	YU20	YU158
YL159	YU21	YU159
YL156	YU22	YU156
YL157	YU23	YU157
YL154	YU24	YU154
YL155	YU25	YU155
YL152	YU26	YU152
YL153	YU27	YU153
YL150	YU28	YU150
YL151	YU29	YU151
YL148	YU30	YU148
YL149	YU31	YU149
YL146	YU32	YU146
YL147	YU33	YU147
YL144	YU34	YU144
YL145	YU35	YU145

The lower foldover connections for HDRY are shown in Table 3-9.

Table 3-9 HDRY lower foldover connection

Upper header	Foldover (nYL_FOLD LOW)	Thru (nY_THRU LOW)
YU178	YL0	YL178
YU179	YL1	YL179
YU176	YL2	YL176
YU177	YL3	YL177
YU174	YL4	YL174
YU175	YL5	YL175
YU172	YL6	YL172
YU173	YL7	YL173
YU170	YL8	YL170
YU171	YL9	YL171
YU168	YL10	YL168
YU169	YL11	YL169
YU166	YL12	YL166
YU167	YL13	YL167
YU164	YL14	YL164
YU165	YL15	YL165
YU162	YL16	YL162
YU163	YL17	YL163
YU160	YL18	YL160
YU161	YL19	YL161
YU158	YL20	YL158
YU159	YL21	YL159
YU156	YL22	YL156

Table 3-9 HDRY lower foldover connection (continued)

Upper header	Foldover (nYL_FOLD LOW)	Thru (nY_THRU LOW)
YU157	YL23	YL157
YU154	YL24	YL154
YU155	YL25	YL155
YU152	YL26	YL152
YU153	YL27	YL153
YU150	YL28	YL150
YU151	YL29	YL151
YU148	YL30	YL148
YU149	YL31	YL149
YU146	YL32	YL146
YU147	YL33	YL147
YU144	YL34	YL144
YU145	YL35	YL145

The upper foldover connections for HDRZ are shown in Table 3-10.

Table 3-10 HDRZ upper foldover connection

Lower header	Foldover (nZU_FOLD LOW)	Thru (nZ_THRU LOW)
ZL254	ZU128	ZU254
ZL255	ZU129	ZU255
ZL252	ZU130	ZU252
ZL253	ZU131	ZU253
ZL250	ZU132	ZU250
ZL251	ZU133	ZU251
ZL248	ZU134	ZU248

Table 3-10 HDRZ upper foldover connection (continued)

Lower header	Foldover (nZU_FOLD LOW)	Thru (nZ_THRU LOW)
ZL249	**ZU135**	**ZU249**
ZL246	**ZU136**	**ZU246**
ZL247	**ZU137**	**ZU247**
ZL244	**ZU138**	**ZU244**
ZL245	**ZU139**	**ZU245**
ZL242	**ZU140**	**ZU242**
ZL243	**ZU141**	**ZU243**
ZL240	**ZU142**	**ZU240**
ZL241	**ZU143**	**ZU241**
ZL238	**ZU144**	**ZU238**
ZL239	**ZU145**	**ZU239**
ZL236	**ZU146**	**ZU236**
ZL237	**ZU147**	**ZU237**
ZL234[a]	-	-
ZL235	**ZU149**	**ZU235**

a. This pin is not controlled by the **nZU_FOLD** and **nZ_THRU** signals

The lower foldover connections for HDRZ are shown in Table 3-11.

Table 3-11 HDRZ lower foldover connection

Upper header	Foldover (nZL_FOLD LOW)	Thru (nZ_THRU LOW)
ZU254	**ZL128**	**ZL254**
ZU255	**ZL129**	**ZL255**
ZU252	**ZL130**	**ZL252**
ZU253	**ZL131**	**ZL253**

Table 3-11 HDRZ lower foldover connection (continued)

Upper header	Foldover (nZL_FOLD LOW)	Thru (nZ_THRU LOW)
ZU250	**ZL132**	**ZL250**
ZU251	**ZL133**	**ZL251**
ZU248	**ZL134**	**ZL248**
ZU249	**ZL135**	**ZL249**
ZU246	**ZL136**	**ZL246**
ZU247	**ZL137**	**ZL247**
ZU244	**ZL138**	**ZL244**
ZU245	**ZL139**	**ZL245**
ZU242	**ZL140**	**ZL242**
ZU243	**ZL141**	**ZL243**
ZU240	**ZL142**	**ZL240**
ZU241	**ZL143**	**ZL241**
ZU238	**ZL144**	**ZL238**
ZU239	**ZL145**	**ZL239**
ZU236	**ZL146**	**ZL236**
ZU237	**ZL147**	**ZL237**
ZU234[a]	-	-
ZU235	**ZL149**	**ZL235**

a. This signal is not controlled by the **nZL_FOLD** and **nZ_THRU** signals.

Note

The numbering for the matched signal pins for HDRX, HDRY, and HDRZ (for example, **ZL238** – **ZU144** and **ZL239** – **ZU145**, **ZL236** – **ZU146** and **ZL237** – **ZU147**) is to keep the polarity of differential signal pairs in the correct order on the header connector. If differential signaling is used, even-numbered pins are negative logic and odd-numbered pins are positive logic.

3.3 Clock architecture

The Logic Tile has three on-board programmable clock generators and a fixed 24MHz clock, **CLK_24MHZ**, that can provide clock sources for the FPGA. The tile can also accept clocks from the system.

Clock signals can be distributed to the tiles above and below and the tile can accept various clock signals from the tiles stacked above and below it. There is also provision for an external clock signal to be input to the tile using an SMB connector.

Caution

The SMB connector is not fitted during manufacture.

ARM will not warrant the product if changes are made to the manufacturing build.

Figure 3-8 on page 3-29 shows the architecture of the clock system. Track lengths are shown for all the distributed clocks.

Note

The upper header connectors for the LT-XC5VLX330 are higher than the connectors used on the LTXC4VLX100+ tiles. The clock paths for signals going to the upper connector have been reduced to compensate for the different connector height.

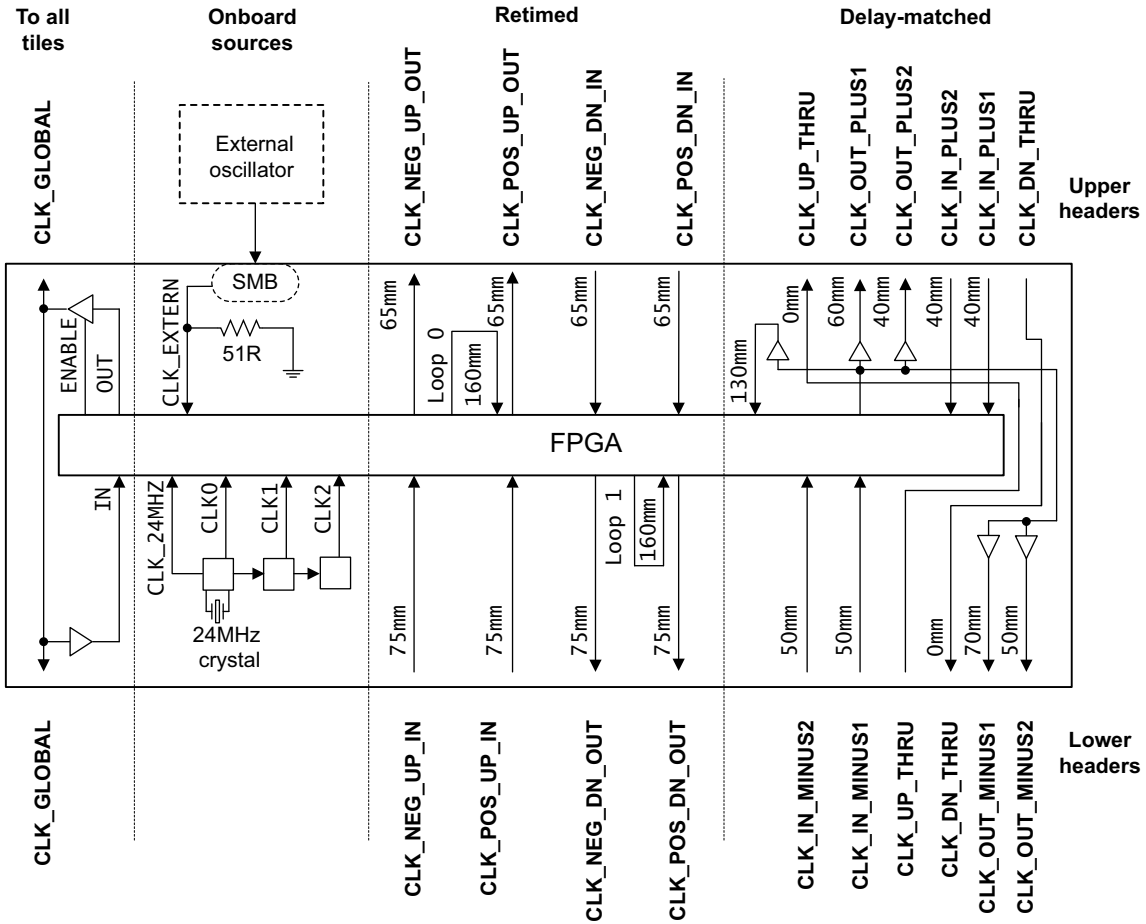

Figure 3-8 Clock signal overview

If multiple tiles are used in a stack, each tile can receive or generate clock signals. There are three basic systems for clocking multiple tiles from one clock source:

Global A single clock line is connected to all tiles. One tile generates the clock and the other tiles accept the clock. The phase of the clock is skewed between the different tiles due to differences in path length. (See *Global clock* on page 3-35.)

Retimed One tile generates a differential (or two single-ended) clock signal. The tiles above and below retime a locally generated clock so that it has the same phase as the clock signal on the generating board. The skew of the incoming clock signal can be removed by a *Delay Locked Loop* (DLL) in the tile FPGA. (See *Retimed clocks on multiple tiles* on page 3-36.)

Delay-matched

A single clock line is generated on one tile and connected to two tiles above and below it in the stack. A delay matched version of the generated clock is input to the FPGA on the tile generating the clock. The trace paths for the five clocks are matched such that all five signals have the same phase when they reach the tile FPGAs. (See *Delay-matched clock distribution (2 up / 2 down)* on page 3-40.)

Details of the various clock signals and the clock generators are given in:

- *Onboard programmable clock generators* on page 3-31
- *Global clock* on page 3-35
- *Retimed clocks on multiple tiles* on page 3-36
- *Delay-matched clock distribution (2 up / 2 down)* on page 3-40.

3.3.1 Onboard programmable clock generators

Three programmable clocks are supplied to the FPGA I/O pins by three serially programmable ICS307M-02 clock sources, as shown in Figure 3-9. These are general purpose clock sources and can be used for your design.

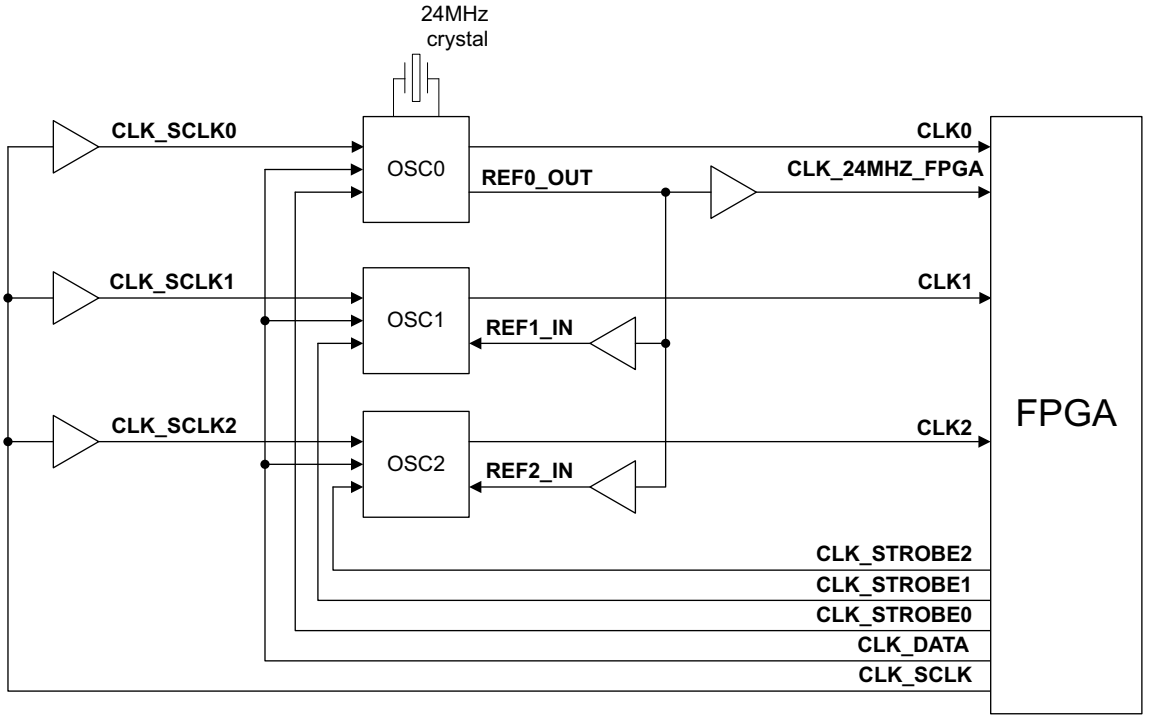

Figure 3-9 Programmable clock generators

Reference clocks

The first oscillator in the chain (OSC0) also provides the fixed-frequency 24MHz clock **CLK_24MHZ_FPGA**.

ICS307M-02 clock generation functional overview

The ICS307M-02 clock sources are supplied with a reference clock by a 24MHz crystal oscillator. The frequency of the outputs from the ICS307M-02 clock sources are controlled by values loaded at the serial data pins. This enables them to produce a wide range of frequencies. Refer to the manufacturer's web site, `www.idt.com` for more information.

Programming the clocks

The frequency of the clock from an ICS307M-02 is set by loading values for the divider and multiplier registers into the serial input port on the clock generator. These control the value of the parameters used to determine the output of the ICS307M-02.

Note

CLK1 and CLK2 in the text below refers to the signals on the ICS307-01/02 data sheet, not to **CLK1** and **CLK2** on the Logic Tile. CLK1 outputs from the ICS307M-02 provide the system clocks and the CLK2 outputs are set to output the 24MHz reference frequency.

You can calculate the frequency of CLK1 using the formula:

$$CLK1 = \frac{48^* (VDW+8)}{(RDW+2)^*OD} \text{ MHz}$$

where:

VDW Is the VCO divider word: 4 – 511 (0, 1, 2, 3 are not permitted).

RDW Is the reference divider word: 1 – 127 (0 is not permitted).

Caution

For a 24MHz reference source the following constraints apply:

$$55\text{MHz} < 48\text{MHz} \times \frac{VDW + 8}{RDW + 2} < 400\text{MHz}$$

and

$$200\text{KHz} < \frac{24\text{MHz}}{RDW + 2}$$

This requires the value of RDW to be within the range 1 – 117.

OD Is the divide ratio: 2 - 10 (selected by S[2:0]).

See Table 3-12 for the available OD (Output Divider) settings.

Table 3-12 ICS307M-02 OD settings

S2	S1	S0	DIVIDE
0	0	0	10
0	0	1	2
0	1	0	8
0	1	1	4
1	0	0	5
1	0	1	7
1	1	0	3
1	1	1	6

The full configuration data stream (**CLK_DATA**) from the FPGAs shown in Figure 3-10 where:

C[1:0] Internal load capacitance for crystal. If an external clock is used, C[1:0] is set to `b00`. See the ICS307-01/02 data sheet for details of capacitance values.

T Duty cycle threshold setting:

The FPGA sets the duty-cycle reference point to VDD/2. T is set to `b1`.

F[1:0] Function of second output:

The FPGA selects the reference signal as source. F is set to `b00`.

S[2:0] Output divider select (OD).

V[8:0] VCO divider word (VDW).

R[6:0] Reference divider word (RDW).

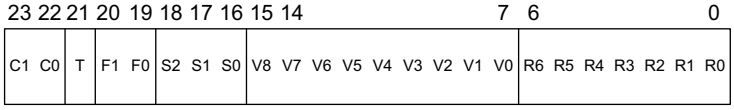

Figure 3-10 VCO configuration data

Note

CLK_DATA is loaded MSB first into the ICS307M-02 shift register. Data is clocked into the register on the rising edge of **SCLK**. The **STROBE** signal is pulsed HIGH after all bits have been shifted into the register.

Serial control of the programmable clocks is implemented in the FPGA design. The C[1:0], T, and F[1:0] fields are fixed and are hard-coded in the FPGA design. The S[2:0], V[8:0] and R[6:0] fields are system dependant and are set in *Debug mode* by the FPGA image. See the `ltxc5vlx330_gtl.v` file supplied on the *Versatile CD* for example values.

For information on the ICS clock generator see the ICS web site at `www.idt.com`.

3.3.2 Global clock

Tiles can receive the global clock or transmit the global clock to all of the boards in the tile stack. Figure 3-11 shows how global clock source selection is controlled.

The **CLK_GLOBAL** signal is present on all Logic Tiles. The signal, after buffering, goes to the **CLK_GLOBAL_IN** input of the FPGAs.

The FPGA on each tile outputs a **CLK_GLOBAL_OUT** signal to a tristate buffer. When the LT-XC5VLX330 is the global clock source, the signal, **CLK_GLBL_nEN** enables the buffer and the local signal, **CLK_GLOBAL_OUT** becomes the global clock, **CLK_GLOBAL** for the system.

Note

The buffers are placed close to the HDRZ connectors, but there will be some skew between tiles. To use in-phase clock signals, use the **CLK_NEG_x**, **CLK_POS_x**, **CLK_IN_x**, or **CLK_OUT_x** signals.

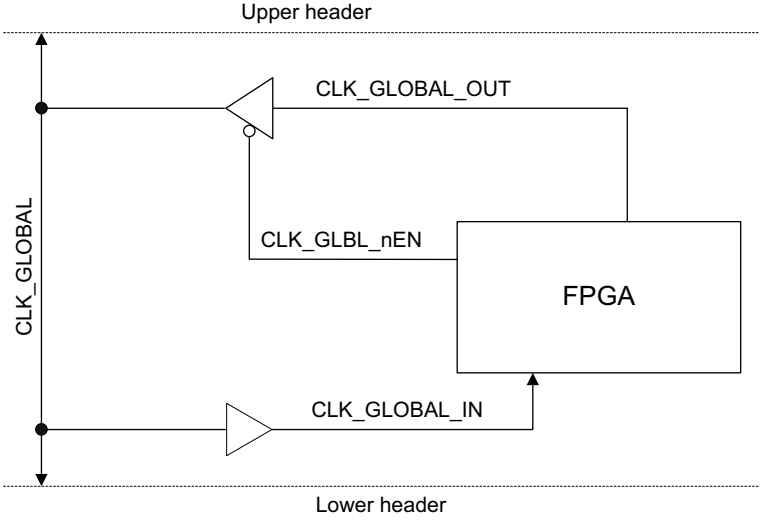

Figure 3-11 Global clock source selection

Caution

Ensure that multiple tiles do not simultaneously drive the **CLK_GLOBAL** signal.

3.3.3 Retimed clocks on multiple tiles

A number of the FPGA **GCLK** inputs and I/O pins enable the FPGA to generate or accept two clocks from the tile above or below as shown in Figure 3-12. (These can be differential clocks.)

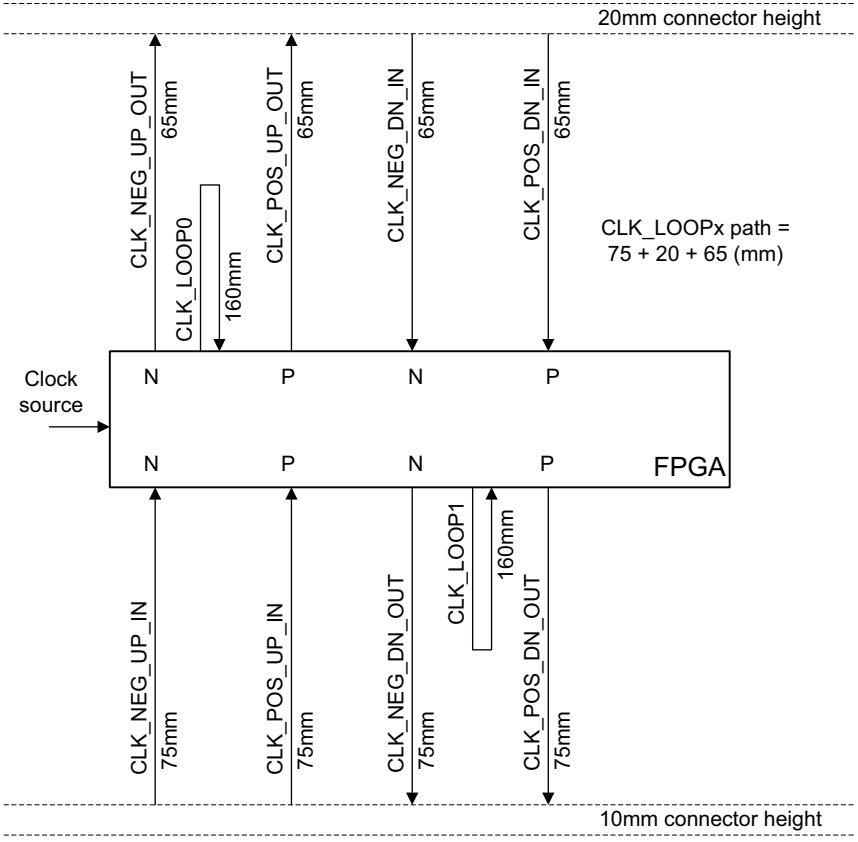

Figure 3-12 Retimed clocking scheme

A differential clock (or two single-ended clocks) can be distributed upwards or downwards to any number of Logic Tiles in the stack. Each tile can use DLLs (in the Virtex-5 FPGA) to phase-align the clock outputs to the source clock. The PCB track lengths for the CLK_LOOPx signal traces are the same length as the total length of the output and input traces and the connector height.

The local clock signals sent along the CLK_LOOPx are inputs to the DLLs and the delay time is used to retime the clocks. The CLK_LOOPx signals are not associated with particular clocks. The retiming is such that the clock edge used in the tile FPGA occurs at the same time as the clock edge on the adjacent tile.

Figure 3-13 shows the DLL connection. Examples of how to route the synchronized clocks between multiple tiles are shown in Figure 3-14 on page 3-38 (routing is shown from the bottom tile) and Figure 3-15 on page 3-39 (routing is shown from the middle tile).

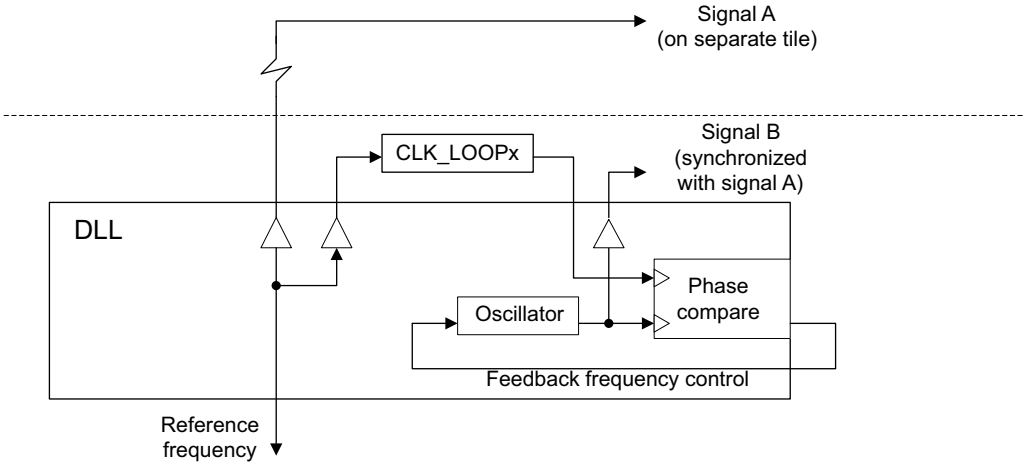

Figure 3-13 Clock retiming using a DLL

Note

There are upper and lower frequency limitations in the retimed clock scheme because of the frequency limits for the DLLs.

There are limitations in the Virtex-5 clock routing that result in **CLK_NEG_DN** and **CLK_NEG_UP** being mapped to local BUFG resources when using single-ended clocks.

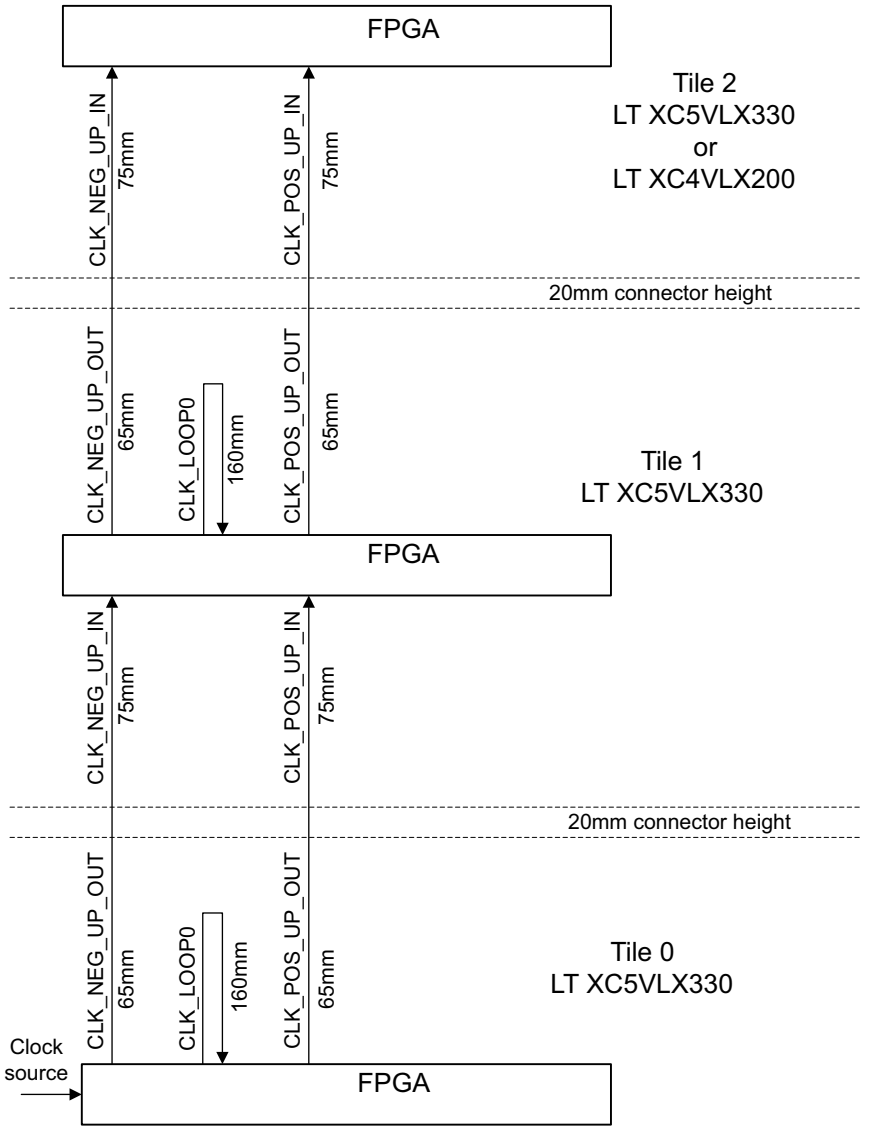

Figure 3-14 Retimed clock routing for three tiles (from bottom tile)

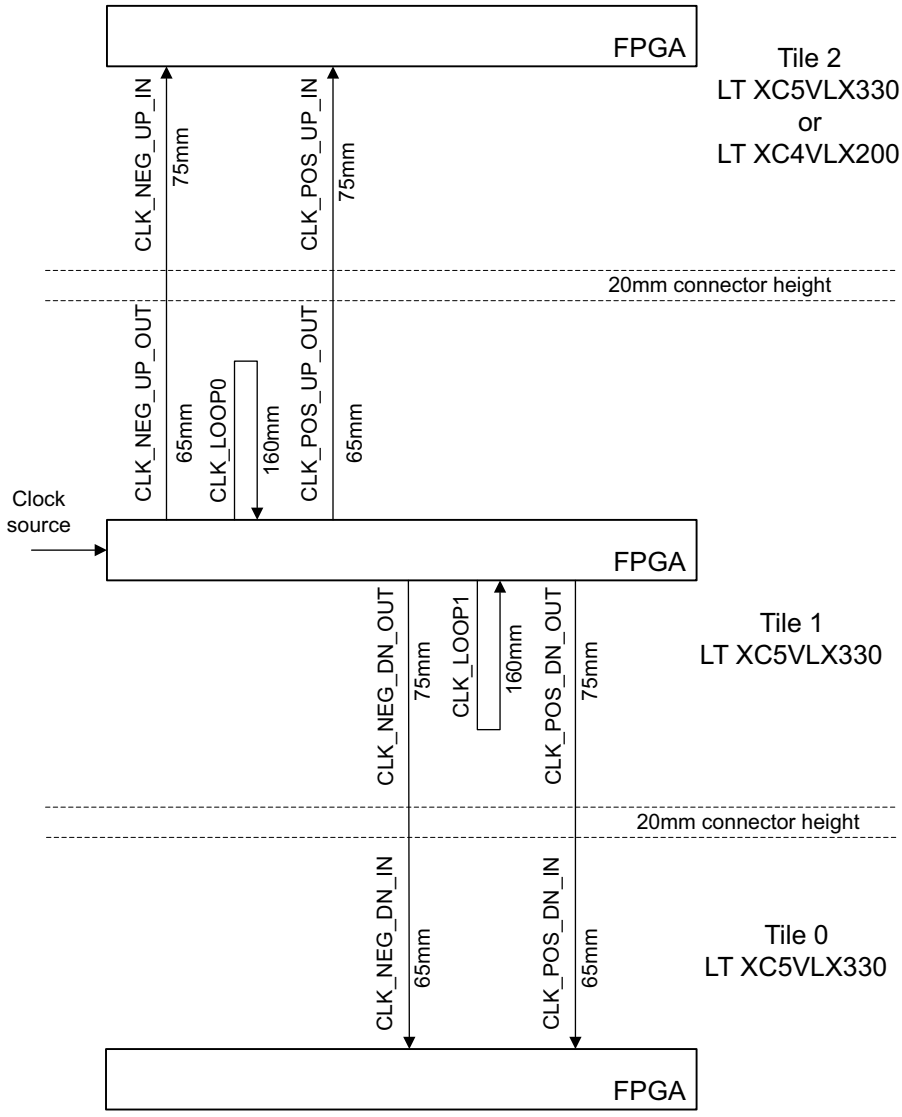

Figure 3-15 Retimed clock routing for three tiles (from middle tile)

3.3.4 Delay-matched clock distribution (2 up / 2 down)

The Logic Tile enables a signal to be distributed to up to five tiles without requiring use of the DLLs to phase realign the signal. (The trace paths are organized such that all source-destination lengths are equal.) The tile generates five versions of a reference clock. These signals are routed to the two tiles above and the two tiles below.

Each tile can also receive four clocks (from the two tiles above and the two tiles below) as shown in Figure 3-17 on page 3-42.

One board in a system can provide a master clock for a five-board system, consisting of the board generating the signal and two boards above and below it. The path lengths for the clock signals are matched so that all clocks are in phase. (The delay loop on the generating tile is equal to the path delays in the signals reaching the other four tiles. The other tiles do not retime the signal, but rely on the path lengths.) See Figure 3-17 on page 3-42.

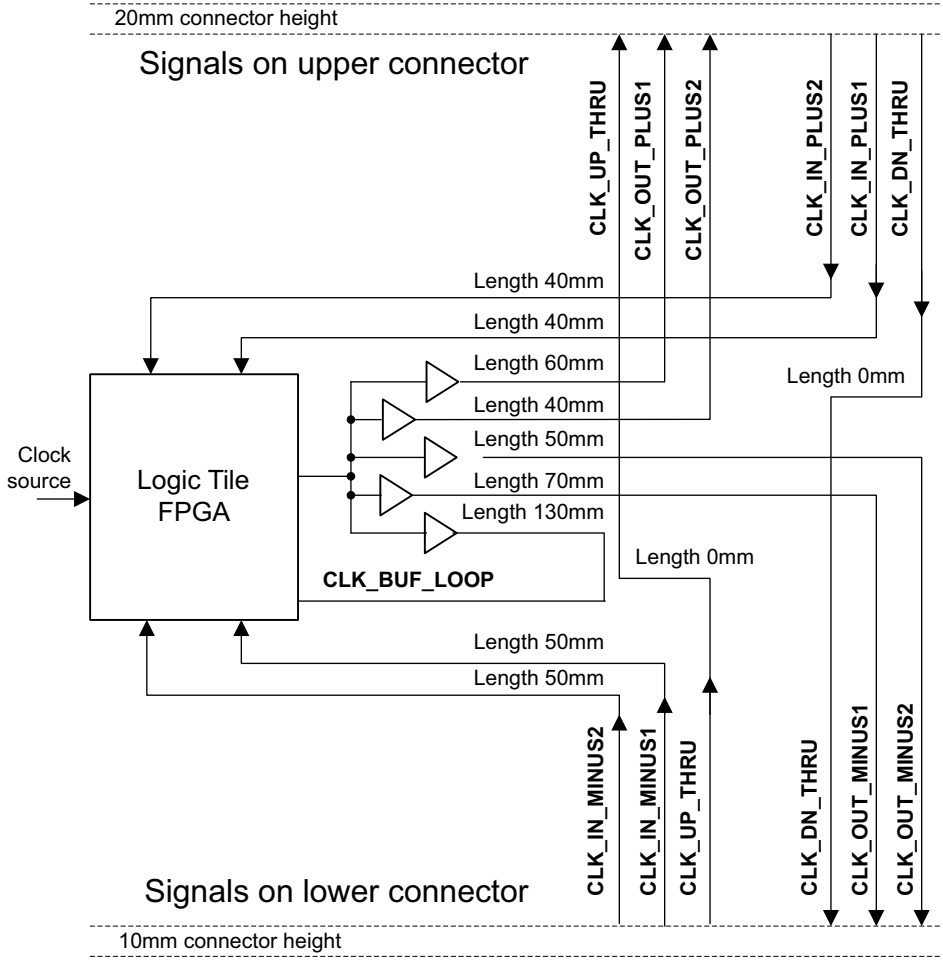

Figure 3-16 Delay-matched clocking scheme

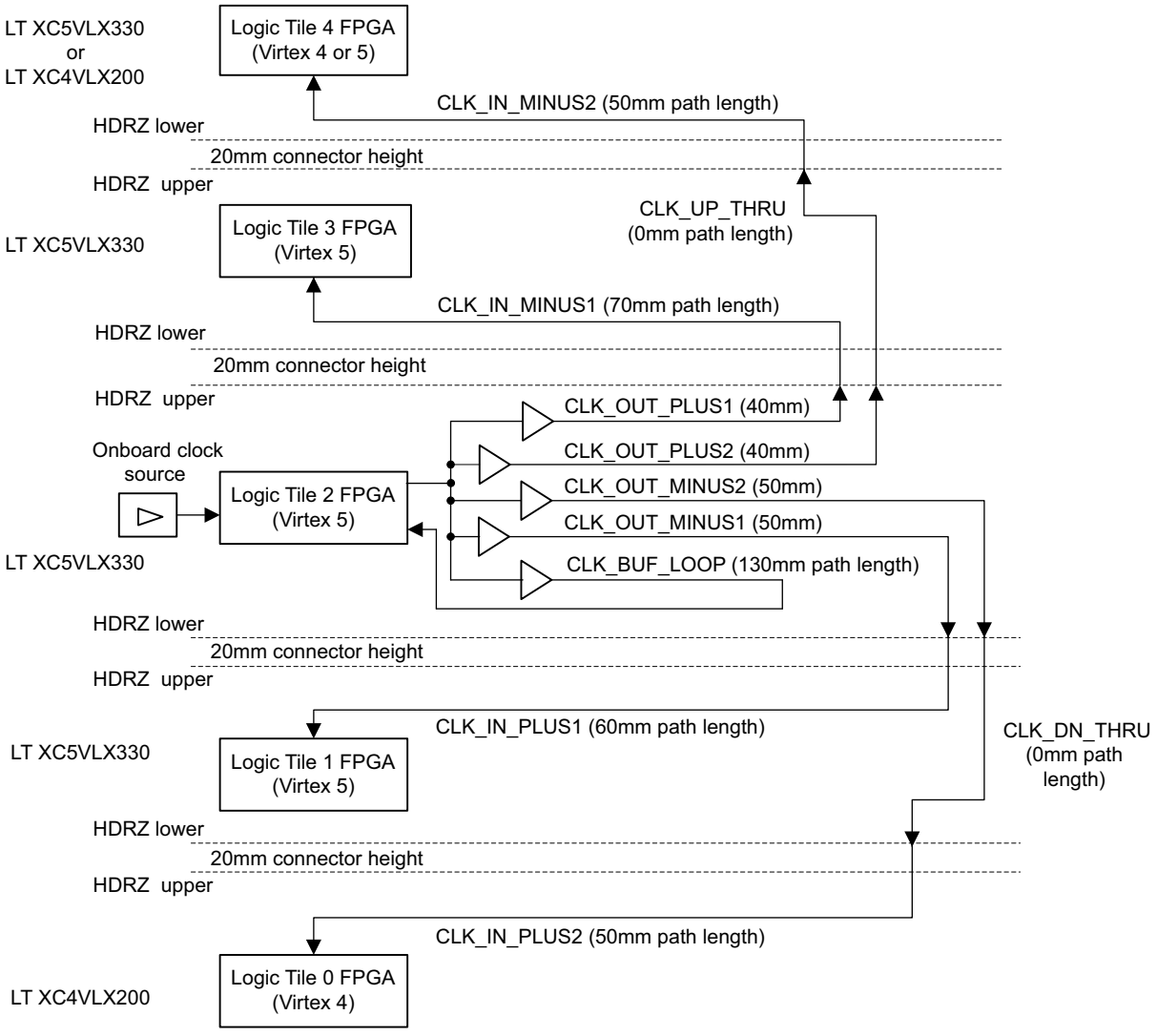

Figure 3-17 Delay-matched clocks for five tiles

3.4 Reset control

The LT-XC5VLX330 has several reset signals. The reset architecture when used with an *Emulation Baseboard* for example, is shown in Figure 3-19 on page 3-44, the full EB reset architecture has been cut down for clarity. See the *RealView Emulation Baseboard User Guide* (DUI 0303) for full details of the EB reset architecture if required.

As configuration of the Logic Tile is linked to the reset process Figure 3-19 on page 3-44 also shows the control of **FPGA_nPROG** by the JTAG and Power-On-Reset (POR) circuits. Figure 3-18 shows the power-on reset sequence, and Table 3-13 on page 3-45 describes the global reset signals.

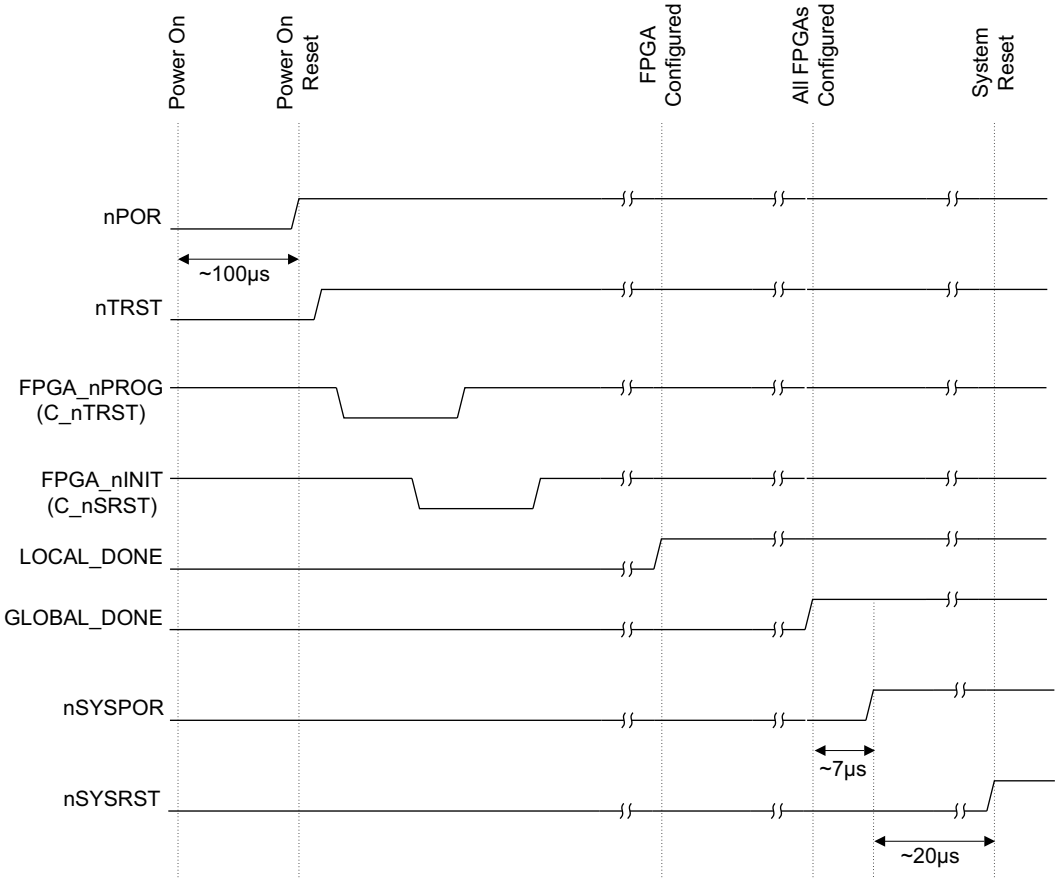

Figure 3-18 Power-on reset sequence

Note

The release of **GLOBAL_DONE** depends on other boards in the system having completed configuration.

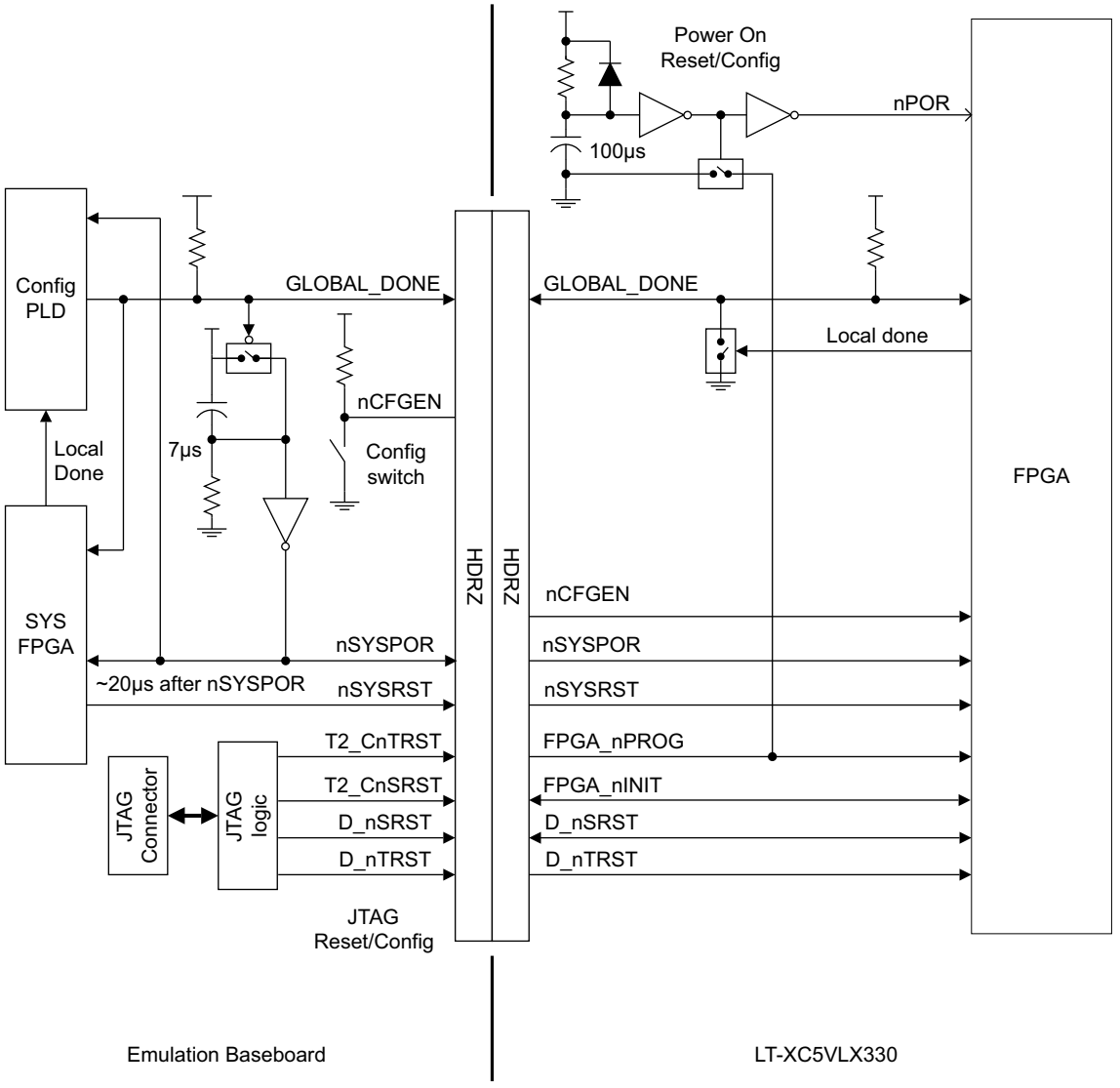

Figure 3-19 Reset and configuration control

Table 3-13 Reset signal descriptions

Name	Description	Function
nPOR	Tile Power-on Reset	This signal is the LT-XC5VLX330 Power-On-Reset. It is also used to generate the **FPGA_nPROG** pulse at power on.
nSRST	System reset	**nSRST** is an active LOW open-collector signal that can be driven by the JTAG equipment to reset the target board. Some JTAG equipment senses this line to determine when you have reset a board. When the signal is driven LOW by the reset controller on the tile, the motherboard resets the whole system by driving **nSYSRST** LOW. This is also used in *Configuration mode* as the initialization signal (**FPGA_nINIT**) to the FPGA. Though not a JTAG signal, **nSRST** is described because it can be controlled by JTAG equipment. **Note** **nSRST** splits into two signals, **D_nSRST** and **C_nSRST**, to provide the debug and configuration signals on HDRZ.
nSYSRST	System reset	A system-wide, master reset signal from the baseboard (for example the EB). This signal is typically used to reset ARM cores, peripherals and user logic. Can be activated from several sources, including **GLOBAL_DONE**=0. (See the *RealView Emulation Baseboard User Guide* for further information on baseboard signals.)
FPGA_nPROG	Configuration reload	The **FPGA_nPROG** signal forces all the FPGAs in the system to reconfigure.
GLOBAL_DONE	Configuration done	Open-collector signal that goes HIGH when all FPGAs have finished configuring. The system is held in reset until this signal goes HIGH.
nSYSPOR	Power-on reset	This is a post-configuration reset signal that is passed to all Logic Tiles in a stack. It is generated by the baseboard. It can be used to reset user logic if required. It remains active for between 1 and 10µS after **GLOBAL_DONE** goes HIGH.
D_nTRST	TAP controller reset	A system-wide, open collector signal that can be driven LOW by JTAG. This is the debug version of the **nTRST** signal. It is connected to an FPGA input/output pin to provide a reset input to the *virtual* TAP controller. There are two possible sources of the **D_nTRST** signal: • JTAG connector • Trace (embedded trace macrocell) connector if supported.

Table 3-13 Reset signal descriptions (continued)

Name	Description	Function
D_nSRST	JTAG system reset	A system-wide, open collector signal that can be driven LOW by JTAG. This is the debug version of the **nSRST** signal.
C_nTRST	TAP controller reset	An open-collector signal that can be driven LOW by JTAG when using a *Progcards utility* to program the FPGA. (This signal is connected to **FPGA_nPROG** and forces the FPGA to reconfigure.) This is the config version of the **nTRST** signal.
C_nSRST	JTAG system reset	A system-wide, open-collector signal. This signal can be driven LOW by JTAG when using a *Progcards utility* to program the FPGA. (This signal is connected to the **INIT** pin of the FPGA.) This is the config version of the **nSRST** signal.

3.5 Memory system

The LT-XC5VLX330 Logic Tile has 32MB of onboard flash memory that holds the FPGA configuration images and two independently controlled 16MB ZBT SRAM modules.

Figure 3-20 shows a block diagram of the memory system.

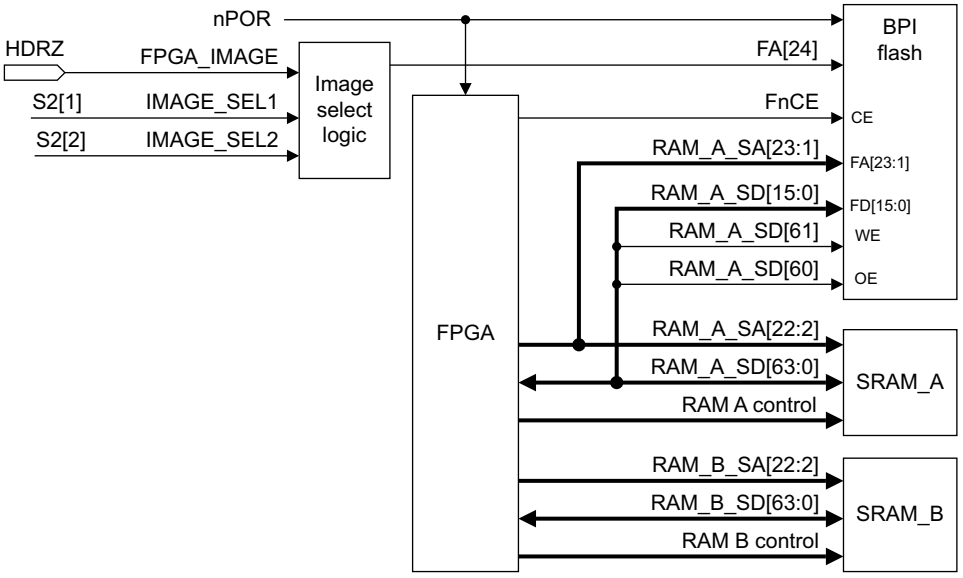

Figure 3-20 Memory system

3.5.1 Flash memory

The Virtex-5 comes equipped with a *Byte Peripheral Interface* (BPI) for programming images into the FPGA. This eliminates the requirement a CPLD to manage programming during power on. The data and address pins are used during the configuration stage, but are released for general I/O purposes after the programming finishes. Two FPGA images (from `.bit` files) can be stored in the flash. See *Configuration system architecture* on page 4-2 for further details.

3.5.2 SRAM Memory

There are two independently controlled ZBT SRAM modules of 16MB size on the Logic Tile. Each module is comprised of two 32 bit *No Bus Latency* (NBL) 8MB SRAM devices that are combined to provide a functionally equivalent 64 bit ZBT SRAM module.

3.6 JTAG support

The Logic Tile does not have a JTAG connector. Use the connector on the baseboard or Interface Module:

- If multiple RealView Logic Tiles are stacked on a baseboard, the JTAG equipment is always connected to the baseboard and the signals are routed upwards to the top tile and then back down through the tiles to the baseboard.
- For use standalone or with an Integrator product, the IM-LT1 Interface Module provides the JTAG connector for accessing the Logic Tile. Refer to the *Integrator/IM-LT1 Interface Module User Guide* for more information on using JTAG and RealView ICE with Integrator products.

There are two separate JTAG paths through the Logic Tile:

- One is used only for configuration of the FPGA. These JTAG signals are identified by the **C_** prefix.
- One is used in *Debug mode* to connect to a virtual TAP controller synthesized into the FPGA. These JTAG signals are identified by the **D_** prefix.

  Note

  If the design in the Logic Tile does not implement a TAP controller then the JTAG signals must be routed through the FPGA. Application Note: AN151 shows an example of this.

The JTAG path chosen depends on whether the system is in *Configuration mode* or *Debug mode*. The CONFIG link on the Interface Module or the CONFIG switch on the EB controls the **nCFGEN** signal that is routed through the Logic Tile connectors.

The **nCFGEN** signal selects between the following modes:

- Install the CONFIG link to use *Configuration mode* for in-system reprogramming of the FPGA in the system.

  The motherboard is also set into *Configuration mode*.
- Remove the CONFIG link on the baseboard or Interface Module for *Debug mode*. If the Logic Tile design contains a TAP controller it is placed in the debug JTAG path.

Figure 3-21 on page 3-49 shows the JTAG data signal routing for an Integrator system. (The CONFIG link is closed, so the figure shows a system in *Configuration mode*.) Figure 3-22 on page 3-50 shows the JTAG clock routing, and Figure 3-23 on page 3-51 shows the JTAG TMS routing. Pull-up resistors are not shown on the drawings.

In *Debug mode*, the JTAG signals from the Interface Module are connected to FPGA input/output pins and enable you to implement a TAP controller in the FPGA design.

Note

If your design (or any other tile in the same stack) does not implement a TAP controller, then you must route **TDI** to **TDO** and **TCK** to **RTCK** in that design.

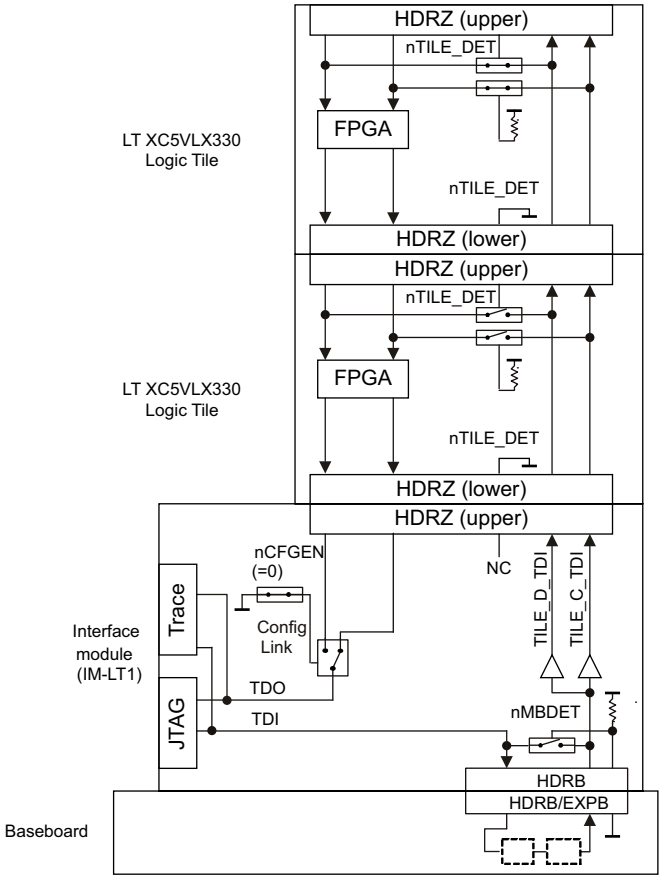

Figure 3-21 JTAG data paths

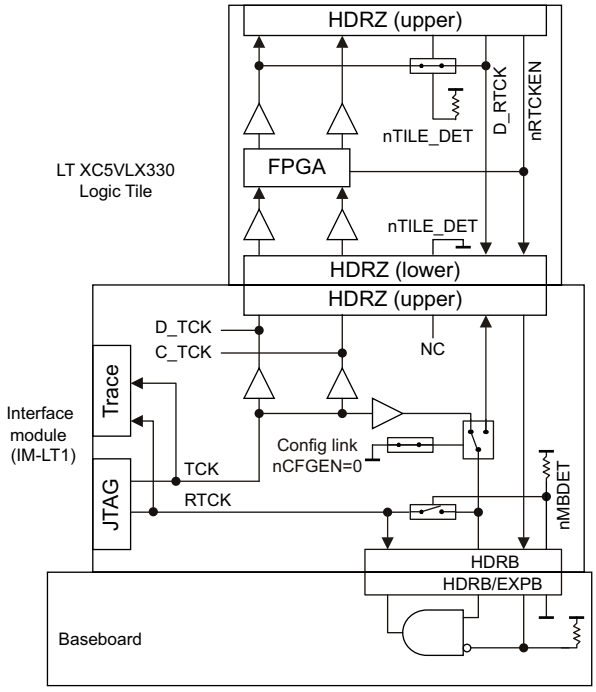

Figure 3-22 JTAG clock paths

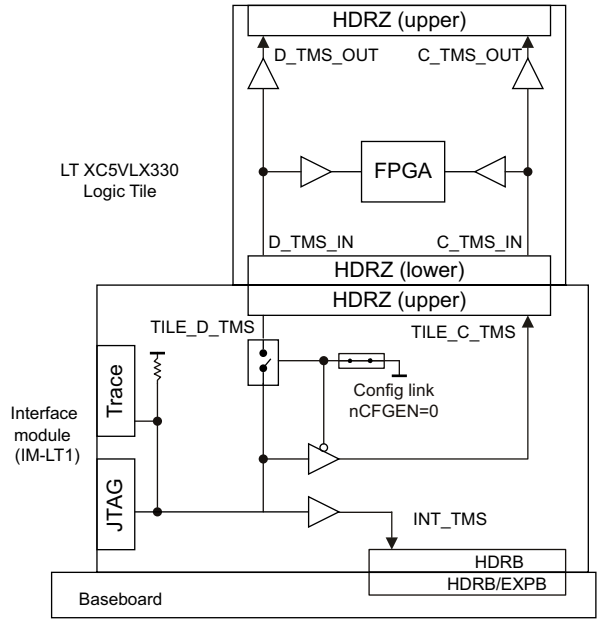

Figure 3-23 JTAG TMS paths

Chapter 4
Configuring the FPGA

This chapter describes how the Xilinx FPGA in the Logic Tile is configured at power-up, the configuration options available, and how to download your own FPGA configurations. It contains the following sections:

4.1 Configuration system architecture

If the CONFIG switch on the baseboard is OFF at power on, the FPGA is configured from the flash memory under control of the self-loading logic contained in the FPGA. If the CONFIG switch on the baseboard is ON (or the baseboard CONFIG link is fitted) you can use RealView ICE to download a configuration directly from the JTAG connector to the FPGA. Figure 4-1 shows the architecture of the FPGA configuration system and Figure 4-2 on page 4-3 shows the reload timing sequence.

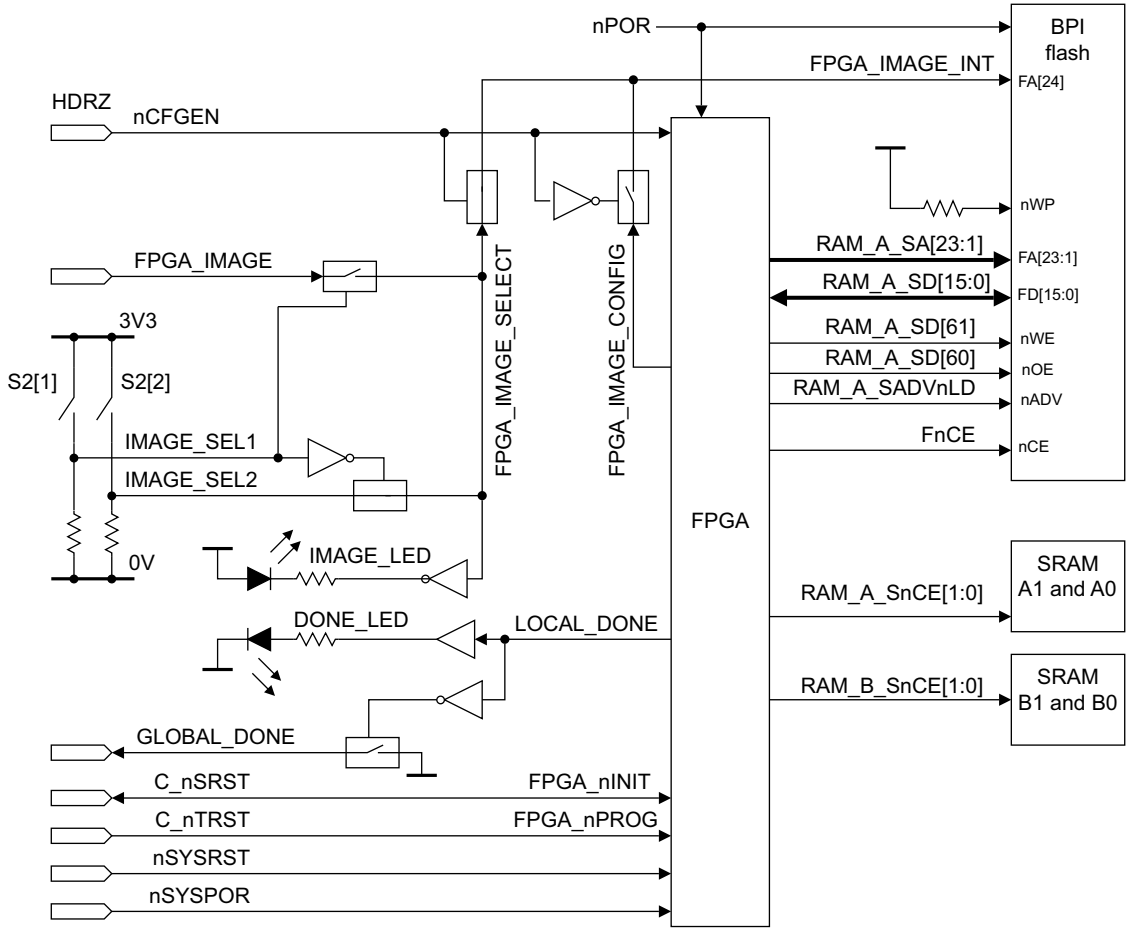

Figure 4-1 Configuration system architecture

Note

The **FPGA_IMAGE** signal is controlled by the baseboard and, by default, selects the low flash image for the Logic Tile if S2[1] is ON.

To manually select the flash image, set S2[1] to OFF and use S2[2] on the Logic Tile to select the image. See *Configuring the FPGA from flash* on page 4-8 for details.

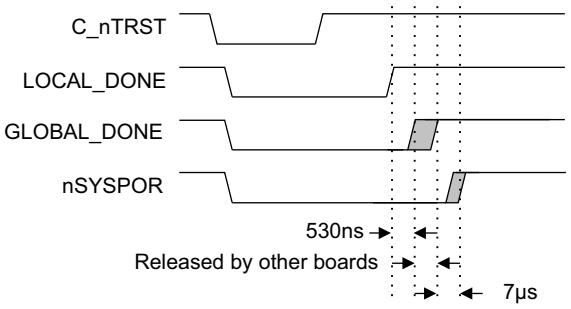

Figure 4-2 Reconfigure sequence

Caution

VBATT is the backup voltage to the FPGA encryption key circuitry within the FPGA. Removing the battery, entering a new encryption key, or shorting across the pads marked R2 erases the encryption key.

If the tile has been supplied with a preloaded encrypted image, the encryption key is required. If the key is erased, you must return the tile to ARM to have the key reloaded.

The battery is expected to last for approximately 10 years from manufacture of the tile. If you replace the battery, ensure that the positive terminal is facing upwards in the holder.

4.1.1 Debug mode

In *Debug mode*, the configuration method used by the FPGA is *BPI mode*. In this mode, the FPGA configuration is loaded from flash memory. The flash must contain valid configuration data and the CONFIG link must not be fitted on the baseboard.

The flash memory can store two FPGA configuration images. The image is selected either by the DIP switch S2 or by the **FPGA_IMAGE** signal from the baseboard (see *Configuring the FPGA from flash* on page 4-8).

The image load sequence consists of:

1. the image select logic decodes **FPGA_IMAGE** signal and the **IMAGE_SEL1** and **IMAGE_SEL2** signals (DIP switch S2[2:1]) and selects the lower or upper flash image accordingly (see *Configuring the FPGA from flash* on page 4-8 for details)
2. setting the **IMAGE_LED** signal to indicate which image is selected for the FPGA
3. setting the flash output enable signal **FnOE** to logic HIGH and automatically loading the flash image to the FPGA. The SRAM signals **RAM_A_SA[61:19]** are shared with the flash address, data and control signals.
4. setting the **DONE_LED** signal LOW to light the FPGA OK LED and indicate that the FPGA is configured with an image.

Figure 4-1 on page 4-2 shows the FPGA configuration mechanism. (For more details see *Configuring the FPGA from flash* on page 4-8.)

4.1.2 Configuration mode

In *Configuration mode*, the configuration method used by the FPGA is *JTAG mode*. In this mode, the FPGA is configured directly from the JTAG connector on the baseboard. You can use JTAG or the USB/Config debug port on the baseboard to download a configuration directly to the FPGA if the CONFIG link is fitted on the baseboard (see *Reconfiguring the FPGA directly* on page 4-10).

4.2 FPGA tool flow

Preparing FPGA configuration files requires two steps:

1. Synthesis.
2. Place and route.

Figure 4-3 illustrates the basic tool flow process.

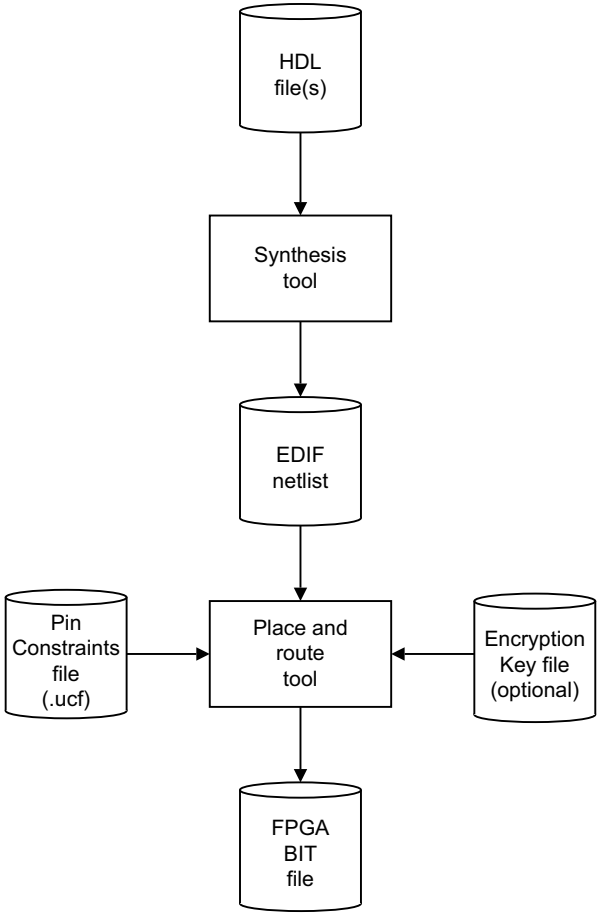

Figure 4-3 Basic tool flow for building FPGA images

4.2.1 Synthesis

The synthesis stage of the tool flow takes the HDL files (either VHDL, Verilog, or a combination) and compiles them into a netlist targeted at a particular technology. In the case of Xilinx Virtex-5, there are several synthesis tools available for both Windows and UNIX platforms.

Synthesis information is supplied either through a GUI front end, or in the form of a command-line script. The information typically includes:

- a list of HDL files
- the target technology
- required optimization, such as area or delay
- timing and frequency requirements
- required pull-ups or pull-downs on the FPGA input/output pads
- output drive strengths.

Refer to the documentation for your particular software tool for further information.

A common netlist file format produced by synthesis is *Electronic Data Interchange Format* (EDIF) (for example, `filename.edf`). This file is used by the next stage of the tool flow, place and route.

4.2.2 Place and route

Place and route for this Logic Tile type is performed using Xilinx-specific software. This produces a `.bit` file that is used to program the FPGA. The `.bit` file is targeted at a particular device, taking into account the device size, package type, and speed grade.

The full part numbers for the Virtex-5 FPGA fitted on the LT-XC5VLX330 is XC5VLX330-1FF1760.

Note

When using the Xilinx tools to generate the programming file (`.bit` file), always specify **CCLK** as the start up clock for your FPGA designs that are to be stored in flash.

In *Configuration mode*, the JTAG start-up clock is used to directly configure the FPGA.

Signal names from the top-level HDL are mapped onto actual device pins by a user constraints file `.ucf`. You can also specify the timing requirements within this file.

Note

The pin constraints file (`.ucf`) for the complete Logic Tile FPGA pin allocation is supplied on the CD. This is intended as a starting point for any design and must be edited before use in the place and route process. The base addresses are set in the progcards board (`.brd`) files.

The current Xilinx tool version that supports the Virtex V FPGA is ISE 9.103.

4.3 Configuring the FPGA from flash

The flash memory has space to store one encrypted or up to two non-encrypted configurations for the Xilinx Virtex-5 FPGA.

Note

S2[1] no longer selects the image defined by the HDRZ signal FLASH_IMAGE but now enables the FPGA to access both memory banks for encrypted bit file support.

To select an encrypted bitfile on the HBI-1072C, set:

S2[1] ON

S2[2] OFF

To select a non-encrypted bitfile on the HBI-1072C, set:

S2[1] OFF

S2[2] Selects between image 0 and 1.

Table 4-1 shows the FPGA image selection options.

Table 4-1 Image selection for XC5V FPGAs

S2[1]	S2[2]	Flash image used	Image LED	Image base address
OFF	OFF	0	Unlit	0x0000000
OFF	ON	1	Lit	0x1000000
ON	x	0 encrypted	Lit/Unlit	0x0000000

Note

The positions of the switches have no effect on the flash programming operation, only image selection on power-up.

See *Application Note AN128* and *Application Note AN170* on the *Versatile CD* for a description of the example lower and upper FPGA images respectively that are stored in flash when the LT-XC5VLX330 is shipped.

4.3.1 Downloading new FPGA configurations to flash memory

The flash memory on the Logic Tile configures the FPGA on power-up when the CONFIG option is not selected on the baseboard. The appropriate *Progcards utility* is used to program the flash. It first loads a flash programmer design into the FPGA, then writes the bit file to the flash memory. You can use the *Progcards utility* to verify the flash image against a bit file.

Note

Progcards utility support for *Multi-ICE* is not provided for this product.
If you are using *RealView ICE* to download the image you must use `progcards_rvi` v2.0 or later for JTAG and `progcards_usb` v2.68 or later for USB.
The latest version of the *Progcards utility* is available from the ARM web site at `www.arm.com/support/downloads/PlatformUtilities.html`.

The steps in writing a bit file to flash are similar to those described in *Reconfiguring the FPGA directly* on page 4-10. The differences are the contents of the `.brd` file (examples are provided on the CD) and the `.bit` file configuration method (**CCLK**).

You must use a board file (`.brd`) to tell the *Progcards utility* about the method of configuring. Examples are provided on the CD supplied with the Logic Tile.
For a full description of this utility, refer to the document file *progcards.pdf* on the supplied CD.

To load the FPGA configuration from flash:

1. Produce a `<filename>.bit` file for your design.
2. Produce a `<filename>.brd` file for your design.
   This is a configuration file for `progcards_rvi` and `progcards_usb`.

   Note

   The `.brd` files are the same regardless of the type of *Progcards utility* being used.
3. Put the Logic Tile in *Configuration mode* by fitting the CONFIG link on the baseboard or Interface Module (later baseboards use a CONFIG slide-switch).
4. Run the appropriate *Progcards utility*. All `.brd` files present in the current directory that match the TAP configuration are offered as options.
   Select one of the options.
5. Wait until the *Progcards utility* finishes and then remove the CONFIG link.
6. Power-cycle the system or press the reconfigure button on the baseboard or Interface Module.

4.4 Reconfiguring the FPGA directly

If you are using JTAG to program the FPGA directly, the configuration is lost when the power supply is removed. Programming an XC5VLX330 takes approximately three minutes to complete on a fast workstation.

4.4.1 Downloading an image directly to the FPGA

You can reprogram the FPGA directly (only on a one-tile system) using the USB debug port. The `progcards_usb` utility (version 2.68 or later) is provided to read `.bit` files and configure the FPGA. You must use a board file (`.brd`) to tell the *Progcards utility* about the method of programming. Examples are provided on the CD supplied with the Logic Tile.

For a full description of this utility, refer to the document file *progcards.pdf* on the supplied CD.

To load a new configuration into the FPGA:

1. Produce a `<filename>.bit` file for your design.
2. Produce a `<filename>.brd` for your design. This is a configuration file for `progcards_rvi` and `progcards_usb`.

   Note

   The `.brd` files are the same regardless of the type of *Progcards utility* being used.
3. Mount the tile on a baseboard such as, for example, the RealView Emulator Baseboard.
4. Put the baseboard in *Configuration mode* by setting the CONFIG switch.
5. Run the *Progcards utility*. All `.brd` files present in the current directory that match the TAP configuration are offered as options. Select one of the options.

4.4.2 Using Xilinx download tools

There is not a dedicated connector for the Xilinx download cable.

Third-party JTAG tools such as the Xilinx parallel cable can be used by connecting them to the 20-way box header on a baseboard. (The 10-way HW-FLYLEAD available from Xilinx enables the Xilinx Parallel Cable IV to connect to the 20-way box header.)

Caution

Although third-party tools can be used to program a configuration directly into the FPGA, they cannot program an image into the flash memory.

Appendix A
Pinouts and Specifications

This appendix describes the interface connectors and signal connections. It contains the following sections:

Caution

There is also provision for an external clock signal to be input to the tile using an SMB connector.

The SMB connector is not fitted during manufacture.

ARM will not warrant the product if changes are made to the manufacturing build.

A.1 Header connectors

Figure A-1 shows the pin numbers and power-blade usage of the HDRX, HDRY, and HDRZ headers on the upper side of the tile.

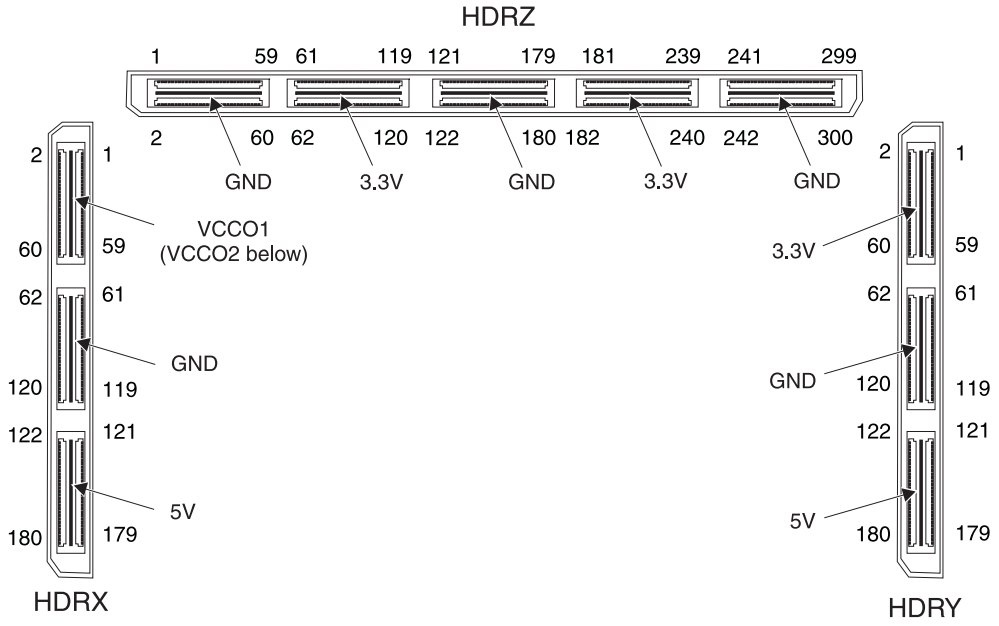

Figure A-1 HDRX, HDRY, and HDRZ (upper) pin numbering

Note

The FPGA I/O bank connections to the **VCCO1** and **VCCO2** power blades are shown in Table 3-5 on page 3-13.

The Foldover logic can reroute some of the signals normally on the upper header connectors back to the lower connectors and some of the signals normally on the lower header connectors back to the upper connectors. See *Foldover* on page 3-15 for details.

A.1.1 HDRX

Table A-1 describes the signals on the HDRX pins. (Replace *x* by `L` for the lower header and by `U` for the upper header.)

Note

Signal levels for the top HDRX connector is set by the tile connected to the top header. Signal levels for the bottom HDRX connector is set by the tile connected to the bottom header. If no tile is connected, the signal I/O level is 3.3 volts.

Table A-1 HDRX signals

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Xx90	1	2	**Xx89**
Xx91	3	4	**Xx88**
Xx92	5	6	**Xx87**
Xx93	7	8	**Xx86**
Xx94	9	10	**Xx85**
Xx95	11	12	**Xx84**
Xx96	13	14	**Xx83**
Xx97	15	16	**Xx82**
Xx98	17	18	**Xx81**
Xx99	19	20	**Xx80**
Xx100	21	22	**Xx79**
Xx101	23	24	**Xx78**
Xx102	25	26	**Xx77**
Xx103	27	28	**Xx76**
Xx104	29	30	**Xx75**
Xx105	31	32	**Xx74**

Table A-1 HDRX signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Xx106	33	34	**Xx73**
Xx107	35	36	**Xx72**
Xx108	37	38	**Xx71**
Xx109	39	40	**Xx70**
Xx110	41	42	**Xx69**
Xx111	43	44	**Xx68**
Xx112	45	46	**Xx67**
Xx113	47	48	**Xx66**
Xx114	49	50	**Xx65**
Xx115	51	52	**Xx64**
Xx116	53	54	**Xx63**
Xx117	55	56	**Xx62**
Xx118	57	58	**Xx61**
Xx119	59	60	**Xx60**
Xx120	61	62	**Xx59**
Xx121	63	64	**Xx58**
Xx122	65	66	**Xx57**
Xx123	67	68	**Xx56**
Xx124	69	70	**Xx55**
Xx125	71	72	**Xx54**
Xx126	73	74	**Xx53**
Xx127	75	76	**Xx52**
Xx128	77	78	**Xx51**
Xx129	79	80	**Xx50**

Table A-1 HDRX signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Xx130	81	82	**Xx49**
Xx131	83	84	**Xx48**
Xx132	85	86	**Xx47**
Xx133	87	88	**Xx46**
Xx134	89	90	**Xx45**
Xx135	91	92	**Xx44**
Xx136	93	94	**Xx43**
Xx137	95	96	**Xx42**
Xx138	97	98	**Xx41**
Xx139	99	100	**Xx40**
Xx140	101	102	**Xx39**
Xx141	103	104	**Xx38**
Xx142	105	106	**Xx37**
Xx143	107	108	**Xx36**
Xx144	109	110	**Xx35**
Xx145	111	112	**Xx34**
Xx146	113	114	**Xx33**
Xx147	115	116	**Xx32**
Xx148	117	118	**Xx31**
Xx149	119	120	**Xx30**
Xx150	121	122	**Xx29**
Xx151	123	124	**Xx28**
Xx152	125	126	**Xx27**
Xx153	127	128	**Xx26**

Table A-1 HDRX signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Xx154	129	130	**Xx25**
Xx155	131	132	**Xx24**
Xx156	133	134	**Xx23**
Xx157	135	136	**Xx22**
Xx158	137	138	**Xx21**
Xx159	139	140	**Xx20**
Xx160	141	142	**Xx19**
Xx161	143	144	**Xx18**
Xx162	145	146	**Xx17**
Xx163	147	148	**Xx16**
Xx164	149	150	**Xx15**
Xx165	151	152	**Xx14**
Xx166	153	154	**Xx13**
Xx167	155	156	**Xx12**
Xx168	157	158	**Xx11**
Xx169	159	160	**Xx10**
Xx170	161	162	**Xx9**
Xx171	163	164	**Xx8**
Xx172	165	166	**Xx7**
Xx173	167	168	**Xx6**
Xx174	169	170	**Xx5**
Xx175	171	172	**Xx4**
Xx176	173	174	**Xx3**

Table A-1 HDRX signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Xx177	175	176	**Xx2**
Xx178	177	178	**Xx1**
Xx179	179	180	**Xx0**

A.1.2 HDRY

Table A-2 describes the signals on the HDRY pins. (Replace *x* by `L` for the lower header and by `U` for the upper header.)

Note

Signal levels for both the top and bottom HDRY connectors are fixed at 3.3 volts.

Table A-2 HDRY signals

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Yx89	1	2	**Yx90**
Yx88	3	4	**Yx91**
Yx87	5	6	**Yx92**
Yx86	7	8	**Yx93**
Yx85	9	10	**Yx94**
Yx84	11	12	**Yx95**
Yx83	13	14	**Yx96**
Yx82	15	16	**Yx97**
Yx81	17	18	**Yx98**
Yx80	19	20	**Yx99**

Table A-2 HDRY signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Yx79	21	22	**Yx100**
Yx78	23	24	**Yx101**
Yx77	25	26	**Yx102**
Yx76	27	28	**Yx103**
Yx75	29	30	**Yx104**
Yx74	31	32	**Yx105**
Yx73	33	34	**Yx106**
Yx72	35	36	**Yx107**
Yx71	37	38	**Yx108**
Yx70	39	40	**Yx109**
Yx69	41	42	**Yx110**
Yx68	43	44	**Yx111**
Yx67	45	46	**Yx112**
Yx66	47	48	**Yx113**
Yx65	49	50	**Yx114**
Yx64	51	52	**Yx115**
Yx63	53	54	**Yx116**
Yx62	55	56	**Yx117**
Yx61	57	58	**Yx118**
Yx60	59	60	**Yx119**
Yx59	61	62	**Yx120**
Yx58	63	64	**Yx121**
Yx57	65	66	**Yx122**
Yx56	67	68	**Yx123**

Table A-2 HDRY signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Yx55	69	70	**Yx124**
Yx54	71	72	**Yx125**
Yx53	73	74	**Yx126**
Yx52	75	76	**Yx127**
Yx51	77	78	**Yx128**
Yx50	79	80	**Yx129**
Yx49	81	82	**Yx130**
Yx48	83	84	**Yx131**
Yx47	85	86	**Yx132**
Yx46	87	88	**Yx133**
Yx45	89	90	**Yx134**
Yx44	91	92	**Yx135**
Yx43	93	94	**Yx136**
Yx42	95	96	**Yx137**
Yx41	97	98	**Yx138**
Yx40	99	100	**Yx139**
Yx39	101	102	**Yx140**
Yx38	103	104	**Yx141**
Yx37	105	106	**Yx142**
Yx36	107	108	**Yx143**
Yx35	109	110	**Yx144**
Yx34	111	112	**Yx145**
Yx33	113	114	**Yx146**
Yx32	115	116	**Yx147**

Table A-2 HDRY signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Yx31	117	118	**Yx148**
Yx30	119	120	**Yx149**
Yx29	121	122	**Yx150**
Yx28	123	124	**Yx151**
Yx27	125	126	**Yx152**
Yx26	127	128	**Yx153**
Yx25	129	130	**Yx154**
Yx24	131	132	**Yx155**
Yx23	133	134	**Yx156**
Yx22	135	136	**Yx157**
Yx21	137	138	**Yx158**
Yx20	139	140	**Yx159**
Yx19	141	142	**Yx160**
Yx18	143	144	**Yx161**
Yx17	145	146	**Yx162**
Yx16	147	148	**Yx163**
Yx15	149	150	**Yx164**
Yx14	151	152	**Yx165**
Yx13	153	154	**Yx166**
Yx12	155	156	**Yx167**
Yx11	157	158	**Yx168**
Yx10	159	160	**Yx169**
Yx9	161	162	**Yx170**
Yx8	163	164	**Yx171**

Table A-2 HDRY signals (continued)

Odd pins		Even pins	
Signal	**Pin**	**Pin**	**Signal**
Yx7	165	166	**Yx172**
Yx6	167	168	**Yx173**
Yx5	169	170	**Yx174**
Yx4	171	172	**Yx175**
Yx3	173	174	**Yx176**
Yx2	175	176	**Yx177**
Yx1	177	178	**Yx178**
Yx0	179	180	**Yx179**

A.1.3 HDRZ

The HDRZ plug and socket have slightly different pinouts. A signal rotation scheme is used to route some of the signals to specific Logic Tiles.

Table A-3 describes the signals on the HDRZ pins.

Note

Signal levels for both the top and bottom HDRZ connectors are fixed at 3.3 volts.

Table A-3 HDRZ signals

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
ZL128	**ZU128**	1	2	**ZU255**	**ZL255**
ZL129	**ZU129**	3	4	**ZU254**	**ZL254**
Z130	**Z130**	5	6	**ZU253**	**ZL253**
Z131	**Z131**	7	8	**ZU252**	**ZL252**
Z132	**Z132**	9	10	**ZU251**	**ZL251**
Z133	**Z133**	11	12	**ZU250**	**ZL250**
Z134	**Z134**	13	14	**ZU249**	**ZL249**
Z135	**Z135**	15	16	**ZU248**	**ZL248**
Z136	**Z136**	17	18	**ZU247**	**ZL247**
Z137	**Z137**	19	20	**ZU246**	**ZL246**
Z138	**Z138**	21	22	**ZU245**	**ZL245**
Z139	**Z139**	23	24	**ZU244**	**ZL244**
Z140	**Z140**	25	26	**ZU243**	**ZL243**
Z141	**Z141**	27	28	**ZU242**	**ZL242**
Z142	**Z142**	29	30	**ZU241**	**ZL241**
Z143	**Z143**	31	32	**ZU240**	**ZL240**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z144	**Z144**	33	34	**ZU239**	**ZL239**
Z145	**Z145**	35	36	**ZU238**	**ZL238**
Z146	**Z146**	37	38	**ZU237**	**ZL237**
Z147	**Z147**	39	40	**ZU236**	**ZL236**
Z148	**Z148**	41	42	**ZU235**	**ZL235**
Z149	**Z149**	43	44	**ZU234**	**ZL234**
Z150	**Z150**	45	46	**ZU233**	**ZL233**
Z151	**Z151**	47	48	**ZU232**	**ZL232**
Z152	**Z152**	49	50	**ZU231**	**ZL231**
Z153	**Z153**	51	52	**ZU230**	**ZL230**
Z154	**Z154**	53	54	**ZU229**	**ZL229**
Z155	**Z155**	55	56	**ZU228**	**ZL228**
Z156	**Z156**	57	58	**ZU227**	**ZL227**
Z157	**Z157**	59	60	**ZU226**	**ZL226**
Z158	**Z158**	61	62	**ZU225**	**ZL225**
Z159	**Z159**	63	64	**ZU224**	**ZL224**
Z160	**Z160**	65	66	**ZU223**	**ZL223**
Z161	**Z161**	67	68	**ZU222**	**ZL222**
Z162	**Z162**	69	70	**ZU221**	**ZL221**
Z163	**Z163**	71	72	**ZU220**	**ZL220**
Z164	**Z164**	73	74	**ZU219**	**ZL219**
Z165	**Z165**	75	76	**ZU218**	**ZL218**
Z166	**Z166**	77	78	**ZU217**	**ZL217**
Z167	**Z167**	79	80	**ZU216**	**ZL216**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z168	**Z168**	81	82	**ZU215**	**ZL215**
Z169	**Z169**	83	84	**ZU214**	**ZL214**
Z170	**Z170**	85	86	**ZU213**	**ZL213**
Z171	**Z171**	87	88	**ZU212**	**ZL212**
Z172	**Z172**	89	90	**ZU211**	**ZL211**
Z173	**Z173**	91	92	**ZU210**	**ZL210**
Z174	**Z174**	93	94	**ZU209**	**ZL209**
Z175	**Z175**	95	96	**ZU208**	**ZL208**
Z176	**Z176**	97	98	**ZU207**	**ZL207**
Z177	**Z177**	99	100	**ZU206**	**ZL206**
Z178	**Z178**	101	102	**ZU205**	**ZL205**
Z179	**Z179**	103	104	**ZU204**	**ZL204**
Z180	**Z180**	105	106	**ZU203**	**ZL203**
Z181	**Z181**	107	108	**ZU202**	**ZL202**
Z182	**Z182**	109	110	**ZU201**	**ZL201**
Z183	**Z183**	111	112	**ZU200**	**ZL200**
Z184	**Z184**	113	114	**ZU199**	**ZL199**
Z185	**Z185**	115	116	**ZU198**	**ZL198**
Z186	**Z186**	117	118	**ZU197**	**ZL197**
Z187	**Z187**	119	120	**ZU196**	**ZL196**
Z188	**Z188**	121	122	**ZU195**	**ZL195**
Z189	**Z189**	123	124	**ZU194**	**ZL194**
ZL190	**ZU190**	125	126	**ZU193**	**ZL193**
ZL191	**ZU191**	127	128	**ZU192**	**ZL192**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
D_nSRST	**D_nSRST**	129	130	**CLK_POS_DN_IN**	**CLK_POS_DN_OUT**
D_nTRST	**D_nTRST**	131	132	**CLK_NEG_DN_IN**	**CLK_NEG_DN_OUT**
D_TDO_OUT	**D_TDO_IN**	133	134	**CLK_POS_UP_OUT**	**CLK_POS_UP_IN**
D_TDI	**D_TDI**	135	136	**CLK_NEG_UP_OUT**	**CLK_NEG_UP_IN**
D_TCK_IN	**D_TCK_OUT**	137	138	**CLK_UP_THRU**	**CLK_IN_MINUS2**
D_TMS_IN	**D_TMS_OUT**	139	140	**CLK_OUT_PLUS1**	**CLK_IN_MINUS1**
D_RTCK	**D_RTCK**	141	142	**CLK_OUT_PLUS2**	**CLK_UP_THRU**
C_nSRST	**C_nSRST**	143	144	**CLK_IN_PLUS2**	**CLK_DN_THRU**
C_nTRST	**C_nTRST**	145	146	**CLK_IN_PLUS1**	**CLK_OUT_MINUS1**
C_TDO_OUT	**C_TDO_IN**	147	148	**CLK_DN_THRU**	**CLK_OUT_MINUS2**
C_TDI	**C_TDI**	149	150	**CLK_GLOBAL**	**CLK_GLOBAL**
C_TCK_IN	**C_TCK_OUT**	151	152	**FPGA_IMAGE**	**FPGA_IMAGE**
C_TMS_IN	**C_TMS_OUT**	153	154	**nSYSPOR**	**nSYSPOR**
GND	**nTILE_DET**	155	156	**nSYSRST**	**nSYSRST**
nCFGEN	**nCFGEN**	157	158	**nRTCKEN**	**nRTCKEN**
GLOBAL_DONE	**GLOBAL_DONE**	159	160	**SPARE12** (reserved)	**SPARE12** (reserved)
SPARE11 (reserved)	**SPARE11** (reserved)	161	162	**SPARE10** (reserved)	**SPARE10** (reserved)
SPARE9 (reserved)	**SPARE9** (reserved)	163	164	**SPARE8** (reserved)	**SPARE8** (reserved)
SPARE7 (reserved)	**SPARE7** (reserved)	165	166	**SPARE6** (reserved)	**SPARE6** (reserved)
SPARE5 (reserved)	**SPARE5** (reserved)	167	168	**SPARE4** (reserved)	**SPARE4** (reserved)
SPARE3 (reserved)	**SPARE3** (reserved)	169	170	**SPARE2** (reserved)	**SPARE2** (reserved)
SPARE1 (reserved)	**SPARE1** (reserved)	171	172	**SPARE0** (reserved)	**SPARE0** (reserved)
Z63	**Z63**	173	174	**Z64**	**Z64**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z62	**Z62**	175	176	**Z65**	**Z65**
Z61	**Z61**	177	178	**Z66**	**Z66**
Z60	**Z60**	179	180	**Z67**	**Z67**
Z59	**Z59**	181	182	**Z68**	**Z68**
Z58	**Z58**	183	184	**Z69**	**Z69**
Z57	**Z57**	185	186	**Z70**	**Z70**
Z56	**Z56**	187	188	**Z71**	**Z71**
Z55	**Z55**	189	190	**Z72**	**Z72**
Z54	**Z54**	191	192	**Z73**	**Z73**
Z53	**Z53**	193	194	**Z74**	**Z74**
Z52	**Z52**	195	196	**Z75**	**Z75**
Z51	**Z51**	197	198	**Z76**	**Z76**
Z50	**Z50**	199	200	**Z77**	**Z77**
Z49	**Z49**	201	202	**Z78**	**Z78**
Z48	**Z48**	203	204	**Z79**	**Z79**
Z47	**Z47**	205	206	**Z80**	**Z80**
Z46	**Z46**	207	208	**Z81**	**Z81**
Z45	**Z45**	209	210	**Z82**	**Z82**
Z44	**Z44**	211	212	**Z83**	**Z83**
Z43	**Z43**	213	214	**Z84**	**Z84**
Z42	**Z42**	215	216	**Z85**	**Z85**
Z41	**Z41**	217	218	**Z86**	**Z86**
Z40	**Z40**	219	220	**Z87**	**Z87**
Z39	**Z39**	221	222	**Z88**	**Z88**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z38	**Z38**	223	224	**Z89**	**Z89**
Z37	**Z37**	225	226	**Z90**	**Z90**
Z36	**Z36**	227	228	**Z91**	**Z91**
Z35	**Z35**	229	230	**Z92**	**Z92**
Z34	**Z34**	231	232	**Z93**	**Z93**
Z33	**Z33**	233	234	**Z94**	**Z94**
Z32	**Z32**	235	236	**Z95**	**Z95**
Z31	**Z31**	237	238	**Z96**	**Z96**
Z30	**Z30**	239	240	**Z97**	**Z97**
Z29	**Z29**	241	242	**Z98**	**Z98**
Z28	**Z28**	243	244	**Z99**	**Z99**
Z27	**Z27**	245	246	**Z100**	**Z100**
Z26	**Z26**	247	248	**Z101**	**Z101**
Z25	**Z25**	249	250	**Z102**	**Z102**
Z24	**Z24**	251	252	**Z103**	**Z103**
Z23	**Z23**	253	254	**Z104**	**Z104**
Z22	**Z22**	255	256	**Z105**	**Z105**
Z21	**Z21**	257	258	**Z106**	**Z106**
Z20	**Z20**	259	260	**Z107**	**Z107**
Z19	**Z19**	261	262	**Z108**	**Z108**
Z18	**Z18**	263	264	**Z109**	**Z109**
Z17	**Z17**	265	266	**Z110**	**Z110**
Z16	**Z16**	267	268	**Z111**	**Z111**
Z15	**Z15**	269	270	**Z112**	**Z112**

Table A-3 HDRZ signals (continued)

Odd pins			Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z14	**Z14**	271	272	**Z113**	**Z113**
Z13	**Z13**	273	274	**Z114**	**Z114**
Z12	**Z12**	275	276	**Z115**	**Z115**
Z11	**Z11**	277	278	**Z116**	**Z116**
Z10	**Z10**	279	280	**Z117**	**Z117**
Z9	**Z9**	281	282	**Z118**	**Z118**
Z8	**Z8**	283	284	**Z119**	**Z119**
Z7	**Z7**	285	286	**Z120**	**Z120**
Z6	**Z6**	287	288	**Z121**	**Z121**
Z5	**Z5**	289	290	**Z122**	**Z122**
Z4	**Z4**	291	292	**Z123**	**Z123**
Z3	**Z3**	293	294	**Z124**	**Z124**
Z2	**Z2**	295	296	**Z125**	**Z125**
Z1	**Z1**	297	298	**Z126**	**Z126**
Z0	**Z0**	299	300	**Z127**	**Z127**

A.2 Electrical specification

This section provides details of the voltage and current characteristics for the tile.

A.2.1 Bus interface characteristics

Table A-4 shows the Logic Tile electrical characteristics.

Table A-4 Electrical characteristics for 3.3V I/O

Symbol	Description	Min	Max	Unit
V_{IH}	High-level input voltage	2.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{OH}	High-level output voltage	2.4	-	V
V_{OL}	Low-level output voltage	-	0.4	V

Caution

The Virtex-5 FPGA supports I/O signalling levels that can require VCCO of 1.5V, 1.8V, 2.5V, or 3.3V. Refer to the electrical characteristics section of the Virtex-5 user guide for details on FPGA I/O. See also *Variable I/O levels* on page 3-13 for information on configurable I/O signal levels.

A.2.2 Current requirements

Table A-5 on page A-20 shows the maximum current requirements for the LT-XC5VLX330 voltage supplies listed by system component when using the example images supplied on the *Versatile CD*.

Note

The actual current depends on the complexity of the design in the Logic Tile and the clock speed used for the system. For a fuller implementation and higher clock rate, the observed value can be several times that shown in the table.

Software to help in calculating the current requirements for your particular application is available on the Xilinx web site at `http://www.xilinx.com`.

Table A-5 Current requirements

Component	1.0V	3.3V	2.5V	1.8V	1.5V
Virtex-5 Core	2.5A	-	-	-	
Virtex-5 I/O	-	2.5A	-	-	
Virtex-5 AUX	-	-	600mA	-	
Virtex-5 AES key	-	-	-	-	0.1mA

The current requirements listed in Table A-5 require a maximum current of 1.2A from the external 5V supply to the LT-XC5VLX330 and 2.5A from the external 3.3V supply.

Caution

The 1.5V supply is provided solely by the internal button-cell battery Removing the battery will result in the loss of the 256b AES encryption key held in the Virtex-5 FPGA making the LT-XC5VLX330 unusable for ARM encrypted files and requiring returning the tile to ARM to have the encryption key reprogrammed.

A.3 Mechanical details

Figure A-2 shows the mechanical outline of the Logic Tile. The position of the two mounting holes and of pin 1 on the Samtec connectors are indicated.

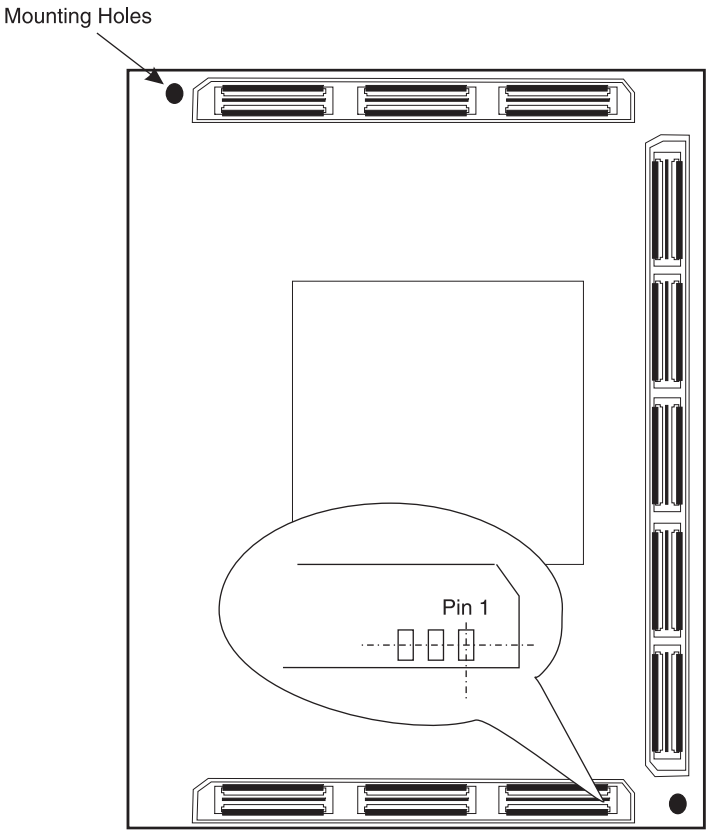

Figure A-2 Board outline

Table A-6 lists the Samtec part numbers and Table A-7 lists the mating heights for each part number extension.

Note

Samtec describes the QTH connector (used on the upper side of the tile) as a header and the QSH connector (used on the lower side of the tile) as a socket.

Table A-6 Samtec part numbers

Header	Part number
HDRXU	QTH-090-05-F-D-A
HDRXL	QSH-090-01-F-D-A-K
HDRYU	QTH-090-05-F-D-A
HDRYL	QSH-090-01-F-D-A-K
HDRZU	QTH-150-05-F-D-A
HDRZL	QSH-150-01-F-D-A-K

Table A-7 Samtec part number and mating heights

Part number (for QTH)	Mating height
-01-	5mm
-02-	8mm
-03-	11mm
-04-	16mm
-05-	19mm
-06-	22mm
-07-	25mm
-08-	30mm

Note

Logic Tiles use the -05- connectors on the upper side of the board and have 19mm board separation to the tile above. The Logic Tiles have a maximum component height of 2.5mm on the bottom and 17mm on the top of the board. This ensures that there are no component interference problems with mated boards.