Versatile/LT-XC2V4000+

Logic Tile

User Guide

ARM DUI 0186E

Versatile/LT-XC2V4000+
User Guide

Release Information

Date	Issue	Confidentiality	Change
November 2002	A	Non-Confidential	New document.
April 2004	B	Non-Confidential	Examples moved from book to CD. Updated information on fold switch.
August 2006	C	Non-Confidential	Fixed reported documentation errors.
May 2007	D	Non-Confidential	Fixed reported documentation errors.
October 2007	E	Non-Confidential	Fixed reported documentation errors.

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
Versatile/LT-XC2V4000+ User Guide

List of Tables
Versatile/LT-XC2V4000+ User Guide

List of Figures
Versatile/LT-XC2V4000+ User Guide

Preface

This preface introduces documentation for the ARM Versatile/XC2V4000+ Logic Tile. It contains the following sections:

About this document

This document describes how to set up and use the ARM Versatile/LT-XC2V4000+ 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 as a standalone development system or with a Versatile/PB926EJ-S baseboard or an Integrator/AP motherboard.

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 and PLD***

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

Appendix A ***Pinouts and Specifications***

See 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)
- *ARM Integrator/CP User Guide* (ARM DUI 0159)
- *ARM Multi-ICE User Guide* (ARM DUI 0048)
- *AMBA Specification* (ARM IHI 0011)
- *ARM Architecture Reference Manual* (ARM DDI 0100)
- *ARM Firmware Suite Reference Guide* (ARM DUI 0102)
- *Multi-ICE™ User Guide* (ARM DUI 0048)
- *RealView™ ICE User Guide* (ARM DUI 0155)
- *Trace Debug Tools User Guide* (ARM DUI 0118)
- *ARM MultiTrace® User Guide* (ARM DUI 0150)
- *ARM RealView Logic Tile LT-XC2V4000+ User Guide* (ARM DUI 0186)
- *RealView Debugger User Guid*e (ARM DUI 0153)

- *RealView Compilation Tools Compilers and Libraries Guide* (ARM DUI 0205)
- *RealView Compilation Tools Developer Guide* (ARM DUI 0203)
- *RealView Compilation Tools Linker and Utilities Guid*e (ARM DUI 0206)
- *Versatile/Core Tile User Guide* (ARM DUI0273)
- *Versatile/IT1 Interface Tile User Guide* (ARM DUI0188)
- *Versatile/Platform Baseboard for ARM926EJ-S User Guide* (ARM DUI0224).

Note

See the application notes and examples on the 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), MicroClock Division of ICS, San Jose, CA.

Feedback

ARM Limited welcomes feedback both on the ARM Versatile/LT-XC2V4000+ Logic Tile and on the documentation.

Feedback on this document

If you have any comments about this document, send an 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 ARM Versatile/LT-XC2V4000+ Logic Tile

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

- the product name
- an explanation of your comments.

Chapter 1
Introduction

This chapter provides an introduction to the LT-XC2V4000+ Logic Tile. It contains the following sections:

1.1 About the LT-XC2V4000+ Logic Tile

The LT-XC2V4000+ Logic Tile is designed as a platform for developing *Advanced Microcontroller Bus Architecture* (AMBA™) *Advanced System Bus* (ASB), *Advanced High-performance Bus* (AHB), *Advanced Peripheral Bus* (APB), *Advanced eXtensible Interface* (AXI) peripherals, or custom logic for use with ARM cores.

Note

The Logic Tile must be used with an external board that provides power and Multi-ICE® connectors (for example, the Versatile/PB926EJ-S or the Integrator/IM-LT1 Interface Module).

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

- For peripheral development or multi-processor development with a Versatile/PB926EJ-S baseboard. The Logic Tile can be used to hold custom peripherals or an implementation of a synthesizable core. Use the Versatile/IT1 Interface Tile for access to the peripheral signals in the Logic Tile.
- As a standalone system (together with an interface module such as the Integrator/IM-LT1 provides the power and JTAG connection). Implement a processor in the Logic Tile FPGA or use a Core Tile.
- For peripheral development with a motherboard (such as the Integrator/AP or Integrator/CP), an Integrator/IM-LT1 Interface Module, and a core module. (If a synthesized processor core is implemented in the Logic Tile FPGA, the core module is not required.)

Figure 1-1 on page 1-3 shows the layout of the Logic Tile.

The LT-XC2V4000+ can be supplied fitted with different Xilinx FPGAs:

XC2V4000 Contains a Xilinx XC2V4000 FPGA.

XC2V6000 Contains a Xilinx XC2V6000 FPGA.

XC2V8000 Contains a Xilinx XC2V8000 FPGA.

The functionality of the Logic Tile is defined by a configuration image loaded into the FPGA at power-up. Two FPGA configuration examples are preloaded into flash (one standalone design and one AMBA AHB slave design).

You can also download your own configurations to flash using Multi-ICE. It is also possible to load an image directly to the FPGA (with either Multi-ICE or JTAG tools supported by the FPGA manufacturer) but directly loaded images are lost when power is removed (see *Reconfiguring the FPGA directly* on page 4-9).

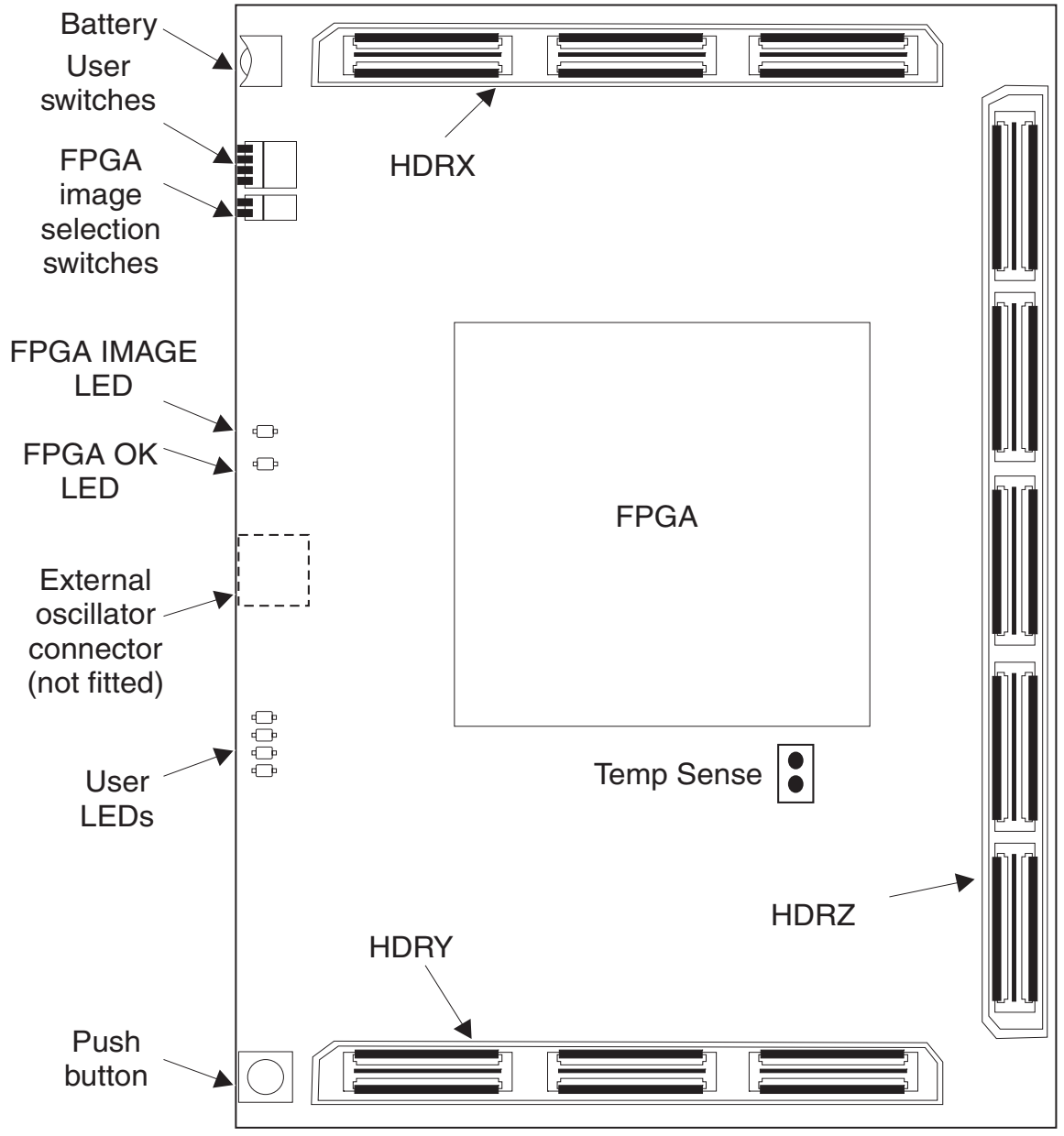

Figure 1-1 LT-XC2V4000+ layout

1.2 Logic tile architecture

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

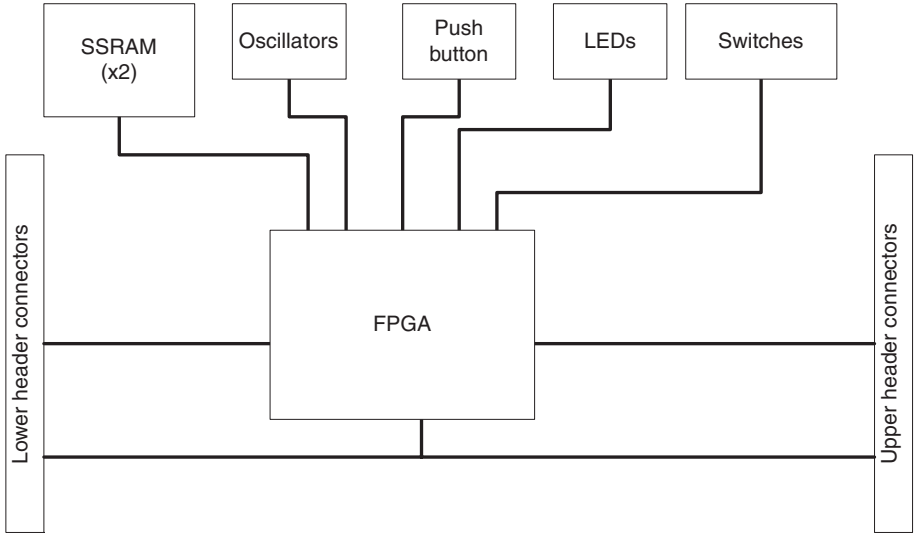

Figure 1-2 System architecture

The Logic Tile comprises the following:

- Xilinx Virtex II FPGA
- configuration *Programmable Logic Device* (PLD) and flash memory for storing FPGA configurations
- Two 2MB ZBT SSRAM chips (these can be used, for example, to model IRAM and DRAM for an ARM core)
- clock generators and reset sources
- switches
- LEDs
- battery for DES encryption keys
- connectors to other tiles.

These components are described 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 permits 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 Versatile/PB926EJ-S, or an Integrator motherboard and a IM-LT1, some of the Logic Tile FPGA signals are connected directly to the motherboard.
- 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 standalone or with a motherboard

The Logic Tile must be used with an external board that provides power and Multi-ICE® connectors (for example, the Versatile/PB926EJ-S or the Integrator/IM-LT1 Interface Module).

The assembly of the Logic Tile and Interface Module can be used standalone or plugged into a motherboard (for example, a Versatile/PB926EJ-S or an Integrator/CP Compact Platform).

Figure 2-1 shows an example system of an Interface Module and a single Logic Tile. To power the Logic Tile as a standalone system, connect the tile to an Interface Module and connect a power supply to the connector on the Interface Module.

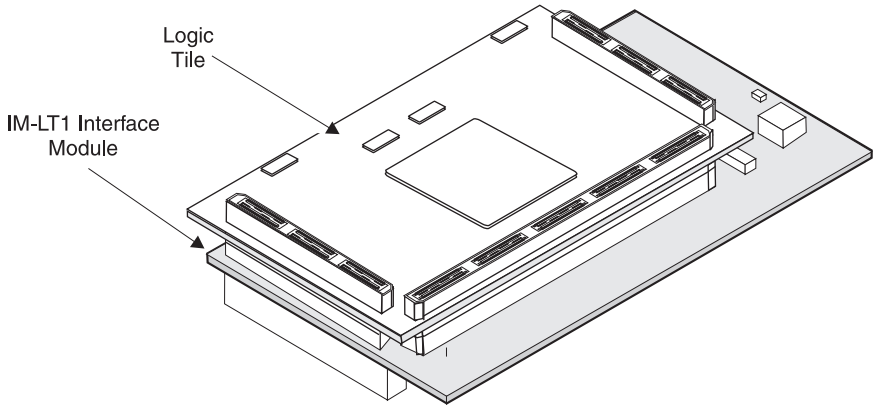

Figure 2-1 Standalone operation with processor in FPGA

Figure 2-2 on page 2-3 shows a Logic Tile mounted onto a Versatile/PB926EJ-S.

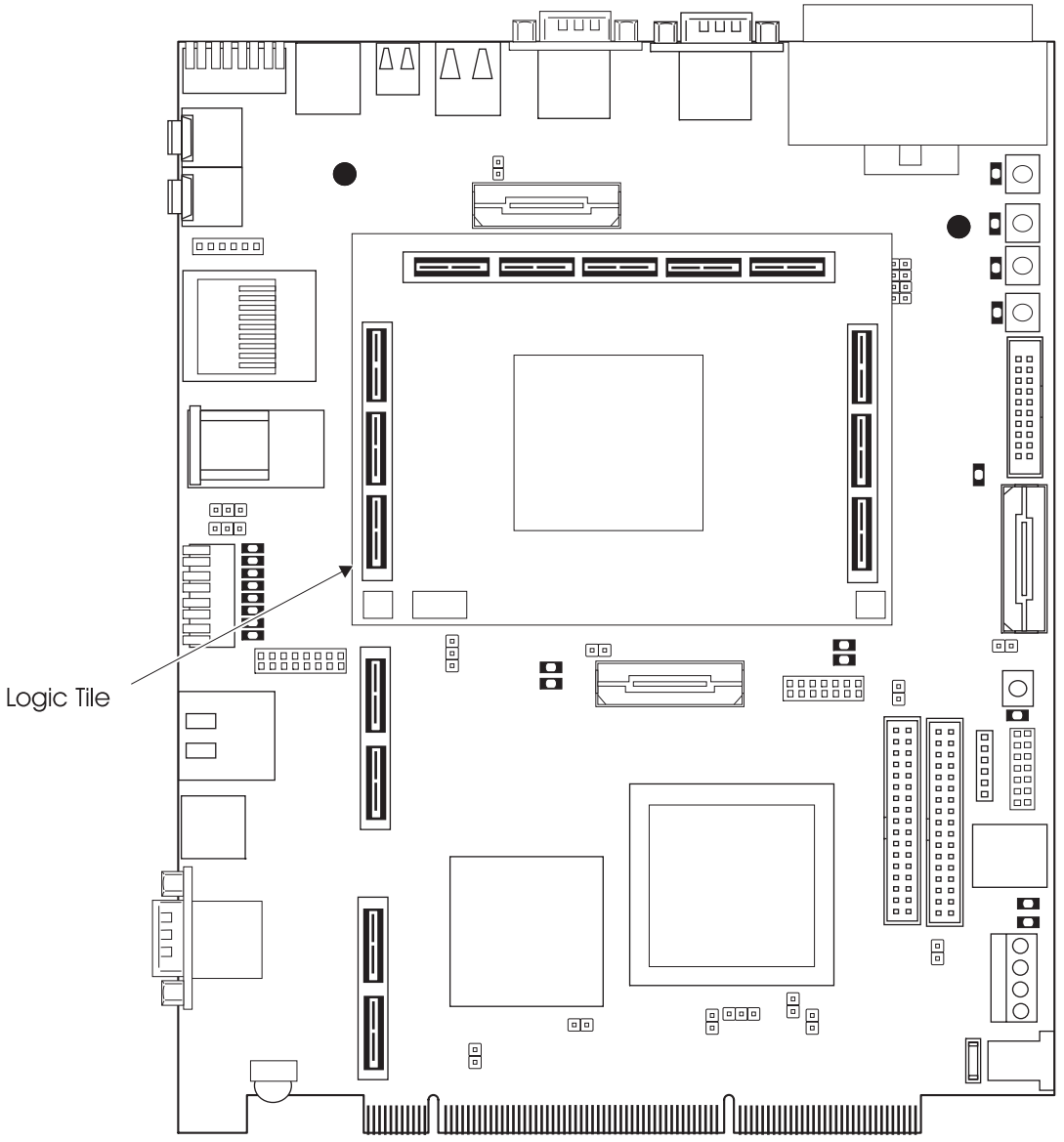

Figure 2-2 Logic tile mounted on a Versatile/PB926EJ-S

Use the Interface Module to mount Logic Tiles onto an Integrator/AP or Integrator/CP motherboard.

Figure 2-3 shows an example system of an Interface Module and Logic Tiles attached to an Integrator/CP (see the *Integrator/CP User Guide* for information on adding expansion modules to the Integrator/CP platform). In this configuration, power is supplied by the Integrator/CP motherboard and the Multi-ICE connection is provided on the Interface Module.

Note

For details on the signals on the Interface Module connectors, see the *Integrator/IM-LT1 Interface Module User Guide*.

See the *Versatile Platform Baseboard for the ARM926EJ-S User Guide* for details on installaing a Logic tile on a Versatile/PB926EJ-S. See the *Integrator/IM-LT1 User Guide* for details on connecting the Logic Tile to the Interface Module.

See the Versatile CD for examples of FPGA images and test software for different Logic Tile configurations.

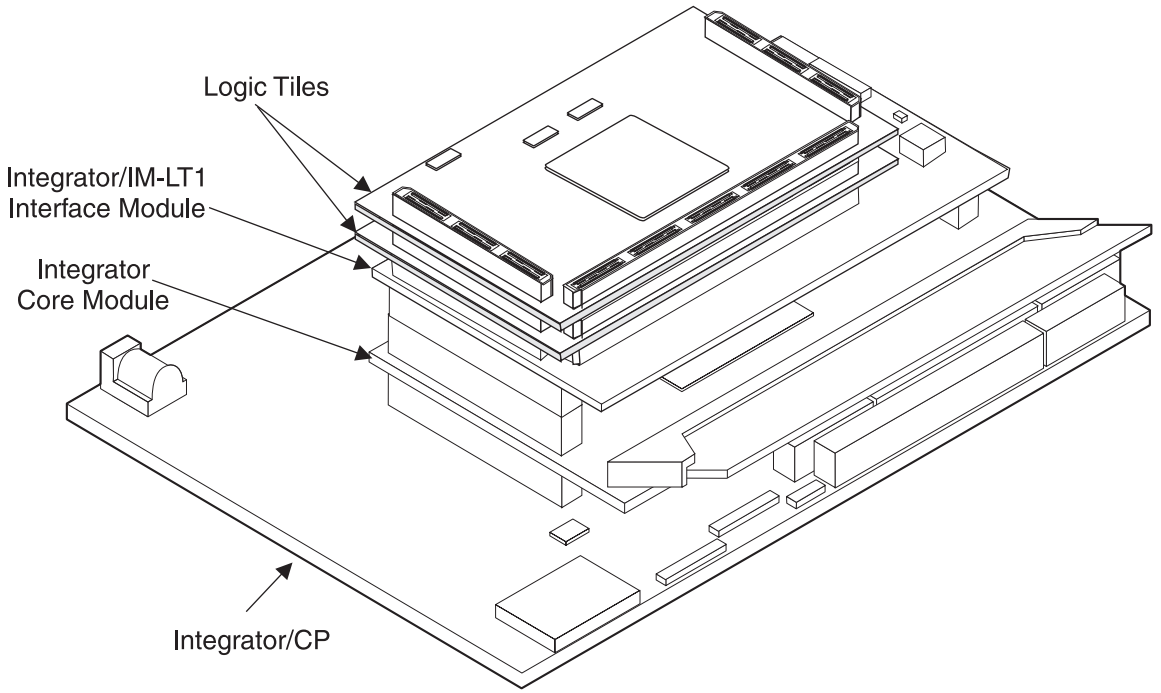

Figure 2-3 Logic tiles mounted on an Integrator/CP

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.

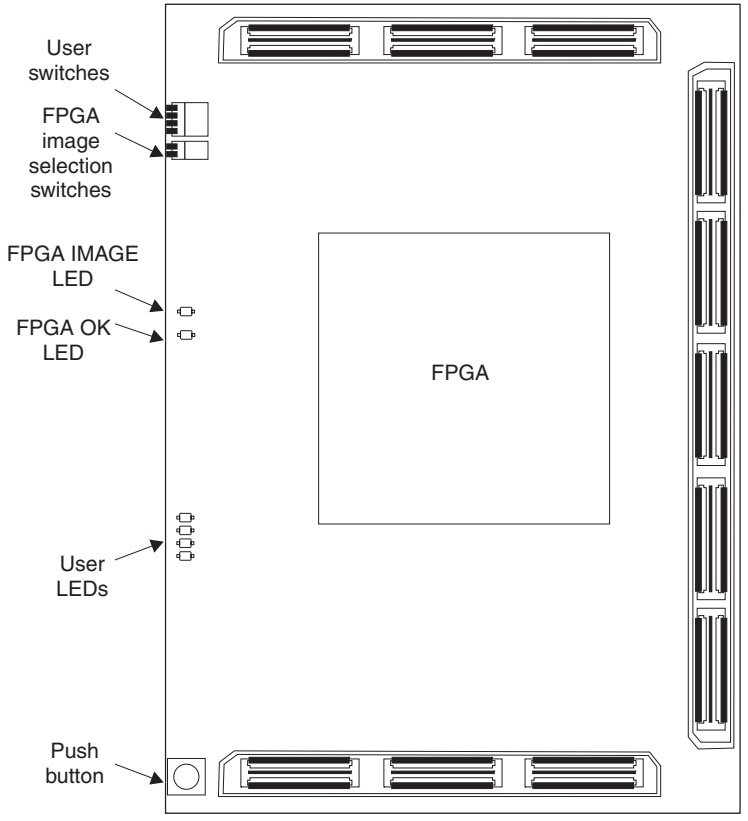

Figure 2-4 Switches and LEDs

2.2.1 Switches

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

S2[1:0] These switches select the image to load into the FPGA at power on.

If the CONFIG link is not fitted to the Interface Module, the Logic Tile powers up in debug mode and the FPGA loads configuration data from flash memory. The flash memory is preloaded with two example configuration images. The control signal to select the image is

determined by the setting of DIP switch S2[2:1] and the state of the global signal **FPGA_IMAGE**. For a full description of FPGA configuration image selection, see *Configuring the FPGA from flash* on page 4-7.

S1[3:0] These switches are connected to the FPGA and are user-defined (after configuration). If the switch is ON, then the signal to the FPGA is HIGH.

The push button is a general-purpose switch. The signal from the push button goes LOW when the push button is depressed. The signal is buffered and connected to the FPGA.

2.2.2 LEDs

There are two individual status LEDs and a single bank of four user LEDs:

FPGA OK This LED indicates that FPGA configuration has completed.

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 is selected.

LED[3:0] These LEDs are connected to the FPGA and are user defined. The FPGA sources current to the leds through 330Ω resistors. A logic high on the FPGA output lights the LED.

2.3 Using Multi-ICE or other JTAG equipment

The JTAG signals for the FPGA on the Logic Tile are routed through the tile headers to the boards above and below the tile. There is not a JTAG connector on the Logic Tile. Use the Multi-ICE 20-way box header on the Versatile/PB926EJ-S platform baseboard or on an interface board (such as an Integrator/IM-LT1 Interface Module).

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. See the Multi-ICE section in the *Integrator IM-LT1 User Guide* for details.

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

Third-party JTAG tools such as the Xilinx parallel cable can be used by connecting them to the 20-way box header.

Note

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

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 II FPGA. The assignment of the input/output banks and JTAG implementation are described in the following sections:

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

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

For information about the configurations supplied with your Logic Tile, see the CD.

At power-up the FPGA loads its configuration data from a flash memory device. Parallel data from the flash is streamed by the PLD into the configuration port of the FPGA (see *PLD* on page 3-13). 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 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. (Configuration devices are not shown.)

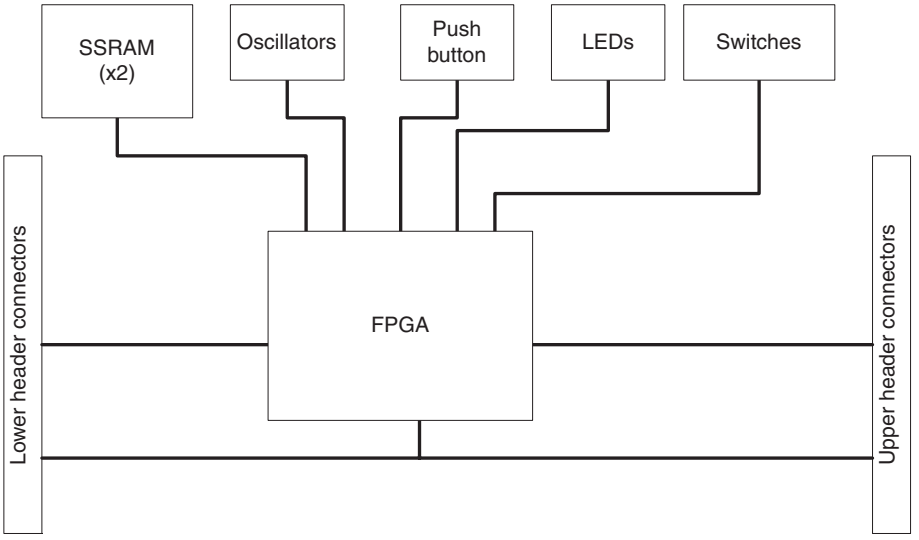

Figure 3-1 LT-XC2V4000+ block diagram

3.1.1 FPGA I/O arrangement

The FPGA input/output pins are organized into eight banks. The majority of the input/output pins are routed to headers to support tile interconnection. Some of the I/O pins have variable I/O signal levels and others are fixed at 3.3V signal levels. See *Header signals* on page 3-4 for more details on I/O pins and 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. The image loaded into the tile FPGA implements a *virtual* TAP controller connection. You can use the virtual TAP controller connection to access devices that are synthesized in the tile FPGA. If you implement a design that does not have a virtual TAP controller, ensure that the FPGA design passes the signals through to the next tile in the stack.

  If you have loaded an ARM CPU core image into the FPGA, you can use the JTAG connector on the Interface Module and Multi-ICE to debug application software running in the tile.

Both scan chains are available through separate signals on the tile headers.

3.2 Header signals

This section gives an overview of the signals present on the header connectors. See *About the LT-XC2V4000+ 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 on the upper headers are generally identified by a U (for example, YU143), while signals on the lower headers are generally identified by an L (for example XL179). Signals that go through both headers, however, do not use the U and L identification. Figure 3-2 on page 3-5 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 signal numbers. For example, HDRY has signal YU113 on pin 48 of the upper 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 connects to a buffer).
- 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 3-7).

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

Caution

The FPGA can be damaged if several pins configured as outputs (on connected Logic Tiles or motherboards) are connected and output a 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 3-6).

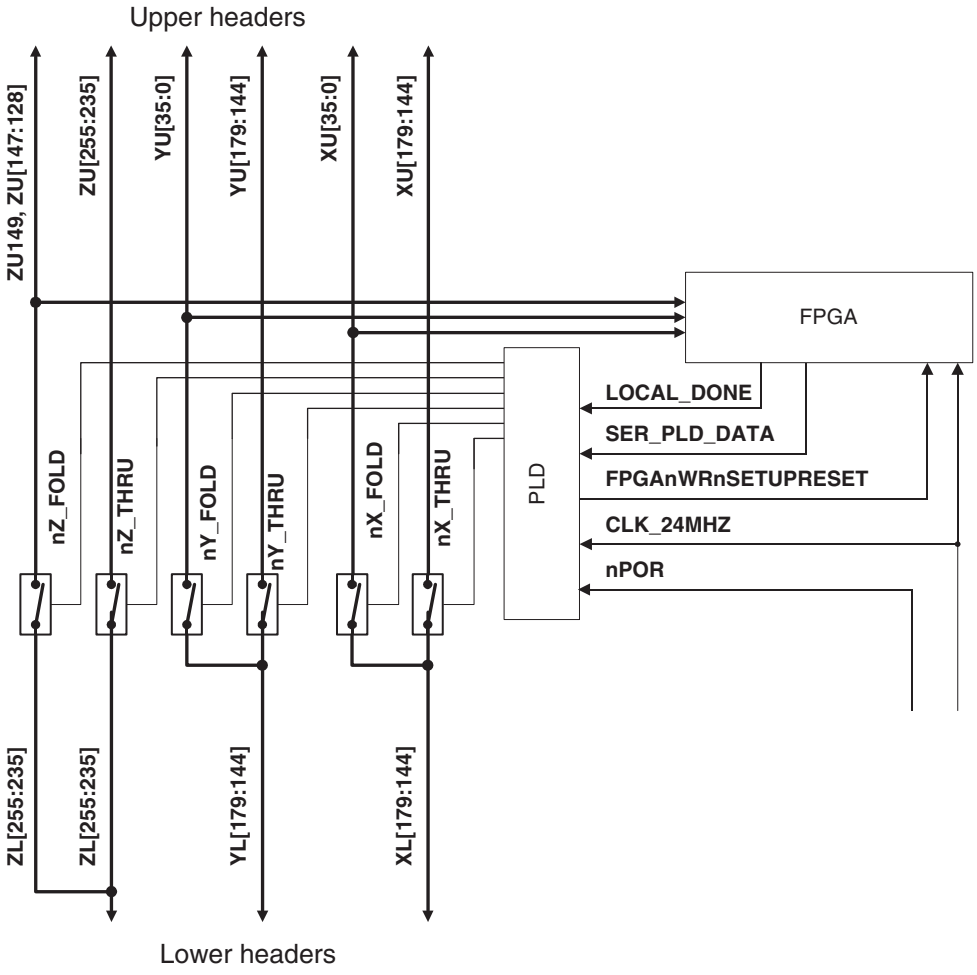

Figure 3-2 Simplified view of fold and through signals

3.2.1 Variable I/O levels

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

The XU, XL, YU, and YL I/O signalling levels are set to 3.3V by 0Ω links on the tile. You can however, remove the links and enable 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 3-1. (All ARM Logic Tiles operate at 3.3 volt signal level, but custom tiles might use a different signal level and supply the voltage to the VCCO blade.)

Table 3-1 I/O signalling levels

Bank	VCCO Voltage	Destination	Description
0	3.3V	HDRZ and onboard signals	Fixed
1	3.3V	HDRZ and onboard signals	Fixed
2	3.3V default	XU signals	Variable, remove R13 to power from tile above
3	3.3V default	XL signals	Variable, remove R14 to power from tile below
4	3.3V	HDRZ and onboard signals	Fixed
5	3.3V	HDRZ and onboard signals	Fixed
6	3.3V default	YL signals	Variable, remove R12 to power from tile below
7	3.3V default	YU signals	Variable, remove R11 to power from tile above

Caution

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

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.

All pins on the lower HDRX and HDRY connectors must operate at 3.3V signal levels if the tile is used with a VPB/926EJ-S or an Integrator motherboard.

3.2.2 Available header pins

Different Xilinx FPGAs have different amounts of I/O available. Depending on the FPGA installed, some header pins are not used as shown in Table 3-2.

Table 3-2 I/O pins available on different Xilinx FPGAs

FPGA	XU/XL[179:0]	YU/YL[179:0]	ZU/ZL[255:128]	Z[127:0]
XC2V4000	120	120	52 on U, 54 on L	128
XC2V6000	144	144	104 on U, 106 on L	128
XC2V8000	144	144	107 on U, 107 on L	128

Note

For the LT-XCV6000, signals **ZL234**, **ZU234**, **ZL233**, and **ZU233** are not available on the XC2V6000 FPGA.

For more information on Xilinx FPGAs, see the Xilinx web site at `www.xilinx.com`.

3.2.3 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, the unused pins have a switch to either connect the upper and lower pins together (pass through) or connect some signals on the upper header to unused pins on the lower connector (foldover). This provides more downwards I/O from the tile to support future platforms. See Figure 3-2 on page 3-5 for a simplified block diagram of the foldover logic. See *HDRX foldover connection* on page 3-8, *HDRY foldover connection* on page 3-10, and *HDRZ foldover connection* on page 3-11 for a detailed list of the fold and through connections.

The foldover switches are controlled by outputs from the PLD. The FPGA uses **SER_PLD_DATA** to serially output configuration information to the PLD after **nSETUPRESET** is released by the PLD. The configuration information determines the I/O signals that control the state of the pass-through and foldover switches (see Figure 3-2 on page 3-5). The serial interface reduces the number of FPGA I/O pins required for static configuration of the Logic Tile. 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.

Note

FPGAnWR_nSETUPRESET is a dual-purpose signal. It is referred to as FPGAnWR before the FPGA is configured and is used to control the transfer of data to the FPGA. (For more details, see the Xilinx documentation.) After configuration, the signal is referred to as **nSETUPRESET** and is used to configure the foldover switches. (For details, see *Reset control* on page 3-25.)

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

Table 3-3 HDRX foldover connection

Lower header	Foldover (nX_FOLD LOW)	Through (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

Table 3-3 HDRX foldover connection (continued)

Lower header	Foldover (nX_FOLD LOW)	Through (nX_THRU LOW)
XL160	XU18	XU160
XL161	XU19	XU161
XL158	XU20	XU158
XL159	XU21	XU159
XL156	XU22	XU156
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 foldover connections for HDRY are shown in Table 3-4.

Table 3-4 HDRY foldover connection

Lower header	Foldover (nY_FOLD LOW)	Through (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
YL165	YU15	YU165
YL162	YU16	YU162
YL163	YU17	YU163
YL160	YU18	YU160
YL161	YU19	YU161
YL158	YU20	YU158
YL159	YU21	YU159
YL156	YU22	YU156

Table 3-4 HDRY foldover connection (continued)

Lower header	Foldover (nY_FOLD LOW)	Through (nY_THRU LOW)
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 foldover connections for HDRZ are shown in Table 3-5.

Table 3-5 HDRZ foldover connection

Lower header	Foldover (nZ_FOLD LOW)	Through (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 3-5 HDRZ foldover connection (continued)

Lower header	Foldover (nZ_FOLD LOW)	Through (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 not controlled by the **nZFOLD** and **nZTHRU** signals

Note

The numbering for the matched 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 you use differential signaling, even-numbered pins are negative logic and odd-numbered pins are positive logic.

3.3 PLD

The PLD implements a bytestreamer design that loads an image from configuration flash into the FPGA at power-up (when in debug mode). The PLD itself is preloaded with a nonvolatile image.

The **GLOBAL_DONE** signal indicates that every FPGA in the system has finished configuring. The system is held in reset until this signal goes HIGH. The PLD holds the **GLOBAL_DONE** signal low for 64 clock cycles after the **LOCAL_DONE** signal indicates that the FPGA has been configured. The PLD also provides the **nRTCKEN** signal to the motherboard and the signals to control the pass-through and foldover switches. The values for these signals are loaded serially from the FPGA after configuration has finished. If you are not using the example design, you must include the example HDL design to perform the serial transfer at startup.

Caution

Do not load the PLD with any image other than that supplied on the CD that accompanies the tile. Loading an incorrect image might render the board unusable. If the image in the PLD has been accidently erased, reload the image into the PLD by inserting the CONFIG link on the Interface Module and using the Progcards utility.

3.4 Clock architecture

The Logic Tile has three on-board programmable clock generators 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 SMA, SMB, or SMC connector (not fitted). Figure 3-3 shows the architecture of the clock system.

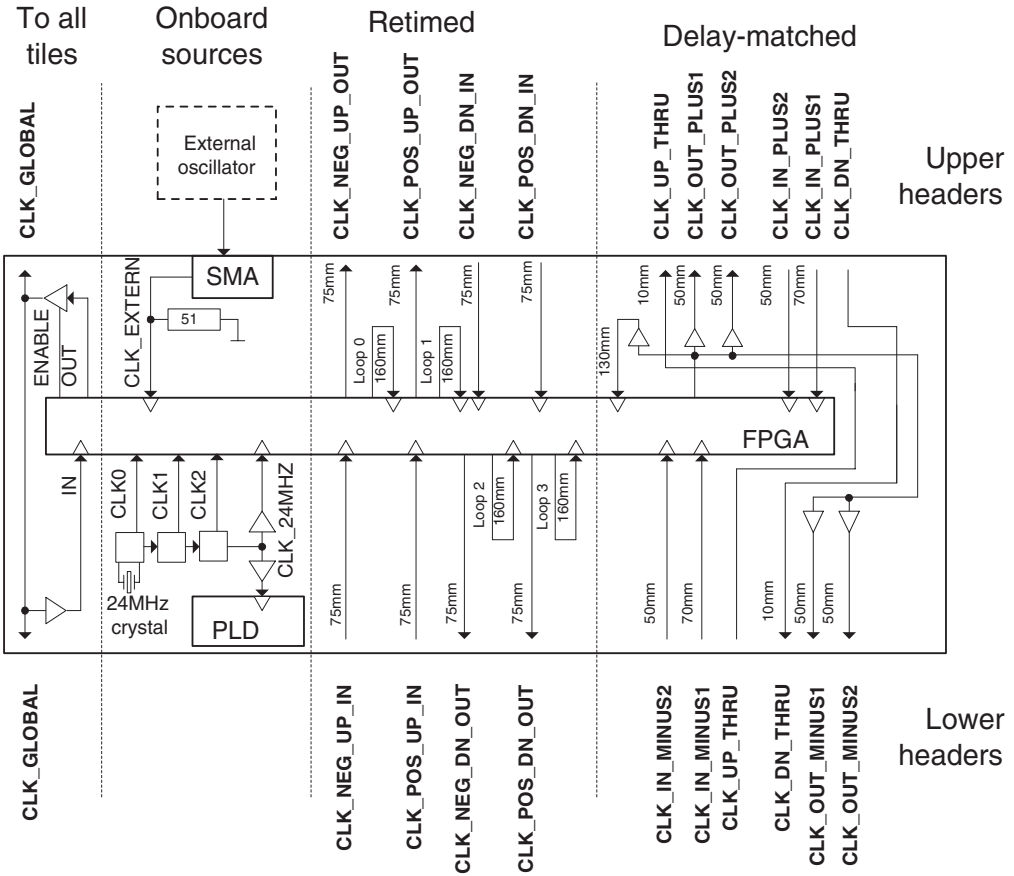

Figure 3-3 Clock signal overview

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 connects 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 because of differences in path length. (See *Global clock* on page 3-19.)

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 3-19.)

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 so 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 3-22.)

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

- *Onboard programmable clock generators* on page 3-16
- *Global clock* on page 3-19
- *Retimed clocks on multiple tiles* on page 3-19
- *Delay-matched clock distribution (2 up/2 down)* on page 3-22.

3.4.1 Onboard programmable clock generators

Three programmable (6 – 200 MHz) clocks are supplied to the I/O pins by three serially programmable MicroClock ICS307 clock generators, as shown in Figure 3-4. These are general purpose clock sources and can be used for your design.

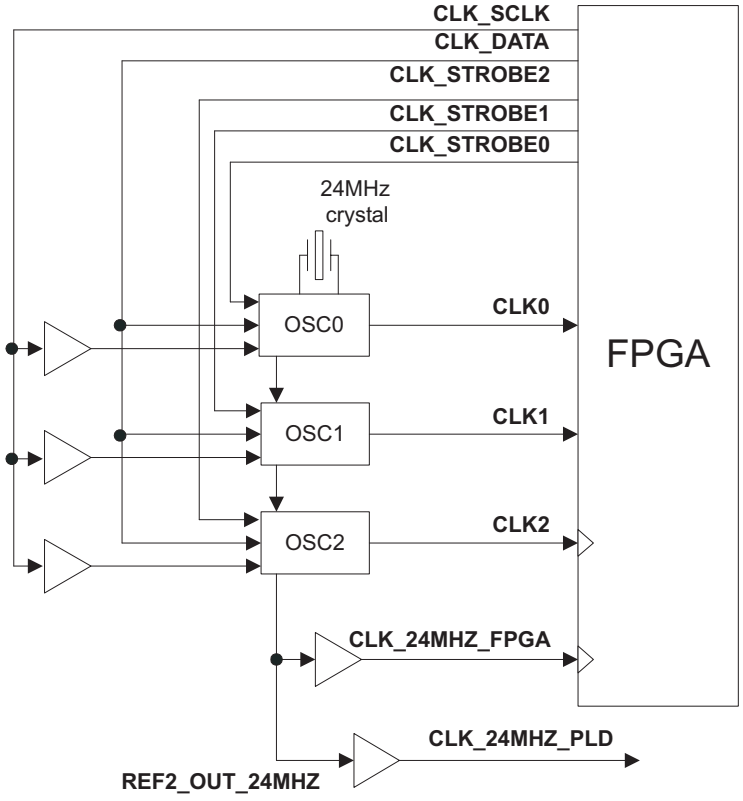

Figure 3-4 Programmable clock generators

Reference clocks

The oscillators also provide the fixed-frequency 24MHz clocks **CLK_24MHZ_FPGA** and **CLK_24MHZ_PLD**.

ICS307 clock generation functional overview

The ICS307s are supplied with a reference clock by a 24MHz crystal oscillator. The frequency of the outputs from the ICS307s are controlled by values loaded into the serial data pins. This enables them to produce a wide range of frequencies. See the manufacturer's web site for more information.

Programming the clocks

The frequency of the clock from an ICS307 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 ICS307.

Note

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

You can calculate the frequency using the formula:

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

where:

VDW Is the VCO divider word (4 – 511).

RDW Is the reference divider word (1 – 127).

DIVIDE Is the divide ratio (2 to 10) selected by the OD bits.

The configuration data stream from the FPGA is shown in Figure 3-5 where:

C[1:0] Internal load capacitance for crystal. If you use an external clock, set C[1:0] to `10`. See the ICS data sheet for details of capacitance values.

T Duty cycle threshold setting:

- 0 selects 1.4V as duty-cycle reference point
- 1 selects VDD/2 as duty-cycle reference point.

F[1:0] Function of CLK2 output:

- 00 selects reference signal

  Note

  This must be set for OSC0 and OSC1 and is recommended for OSC2.

- 01 selects reference signal divided by two
- 10 disables output for CLK2
- 11 selects CLK1 signal divided by two.

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

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

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

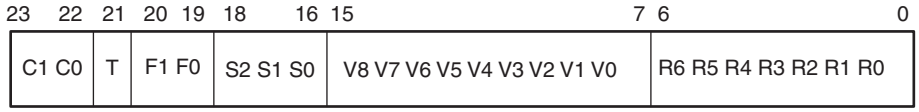

Figure 3-5 VCO configuration data

Note

Bit 23 is loaded into the shift register first and bit 0 is loaded last. 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.

For more information on the ICS clock generator and a frequency calculator, see the ICS web site at `www.icst.com`.

Serial control of the programmable clocks must be implemented in the FPGA design. Example HDL (APBClocks and APBClockArbiter) is provided on the CD. (See the examples provided on the CD.)

3.4.2 Global clock

Tiles can receive the global clock or transmit the global clock to all of the boards in the tile stack.

The **CLK_GLOBAL** signal is present on all Logic Tiles. The signal 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. The signal **CLK_GLOBAL_OUT_nEN** enables the buffer and the local signal **CLK_GLOBAL_OUT** becomes the global clock for the system.

Note

The buffers are placed close to the HDRZ connectors, but there is 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.

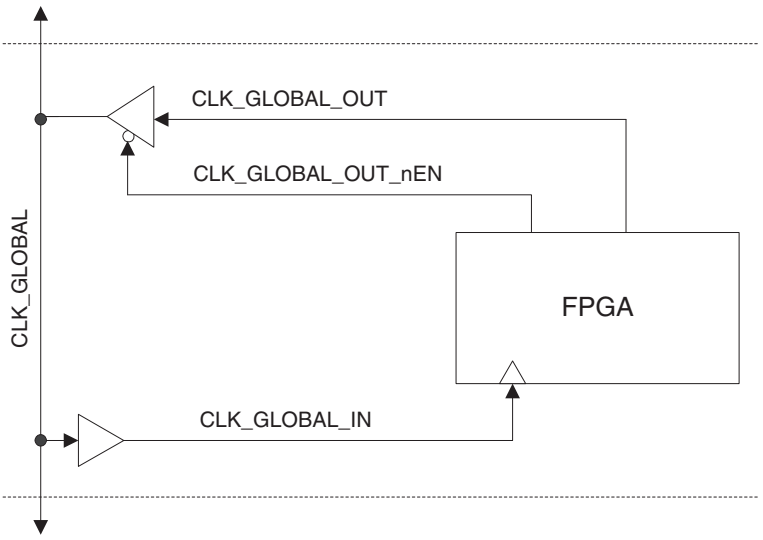

Figure 3-6 Global clock selection

3.4.3 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 3-7 on page 3-20. (These can be differential clocks.)

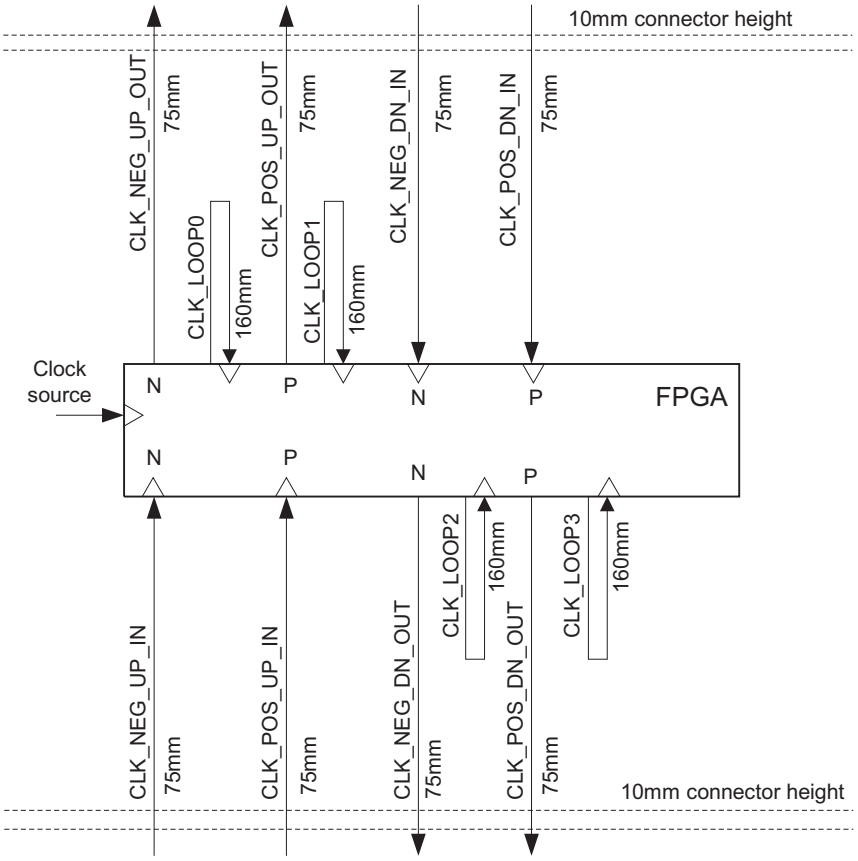

Figure 3-7 Retimed clocking scheme

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 uses DLLs (in the Virtex II 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 enables the retiming of the clocks. The DLL_LOOPx signals are not associated with particular clocks. The retiming is so that the clock edge used in the tile FPGA occurs at the same time as the clock edge on the adjacent tile. See Figure 3-8 on page 3-21.

Note

The clock frequency for the retimed clock scheme must be between 24 and 180MHz because of the frequency limits for the DLLs.

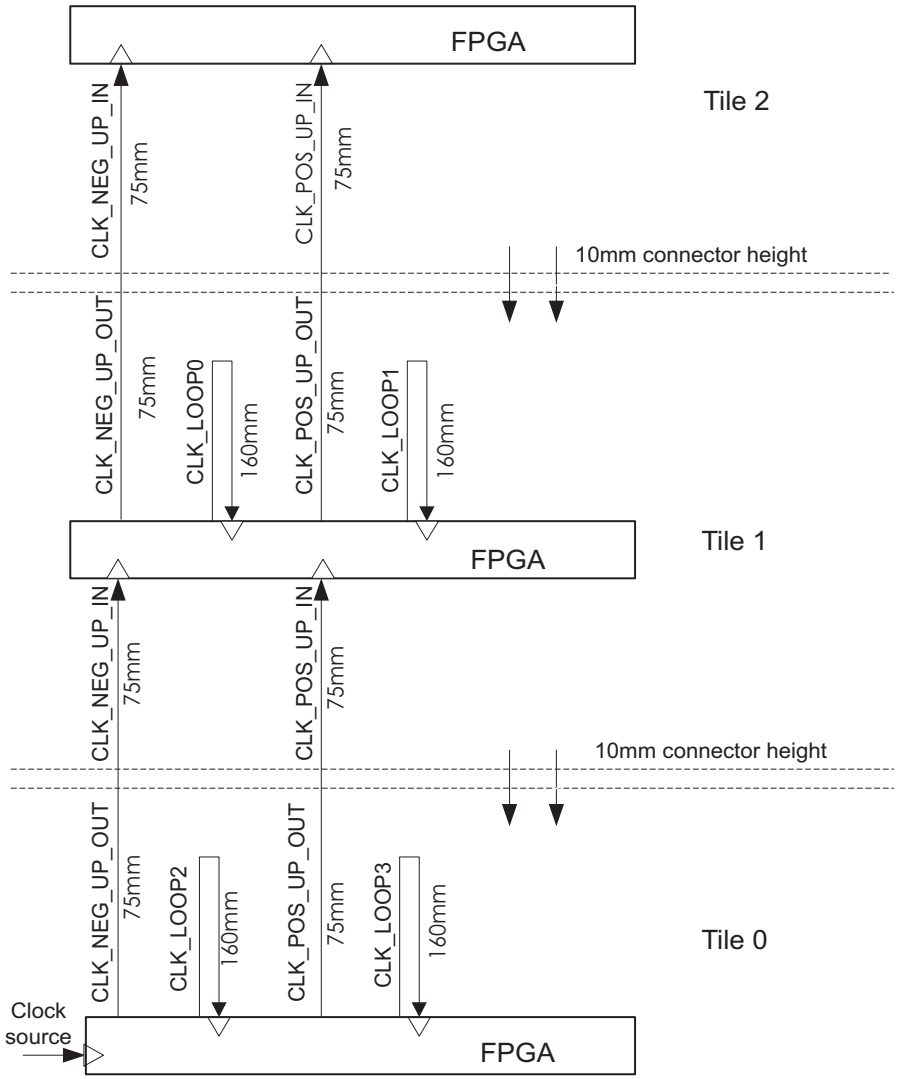

Figure 3-8 Retimed clock routing for three tiles (from bottom tile)

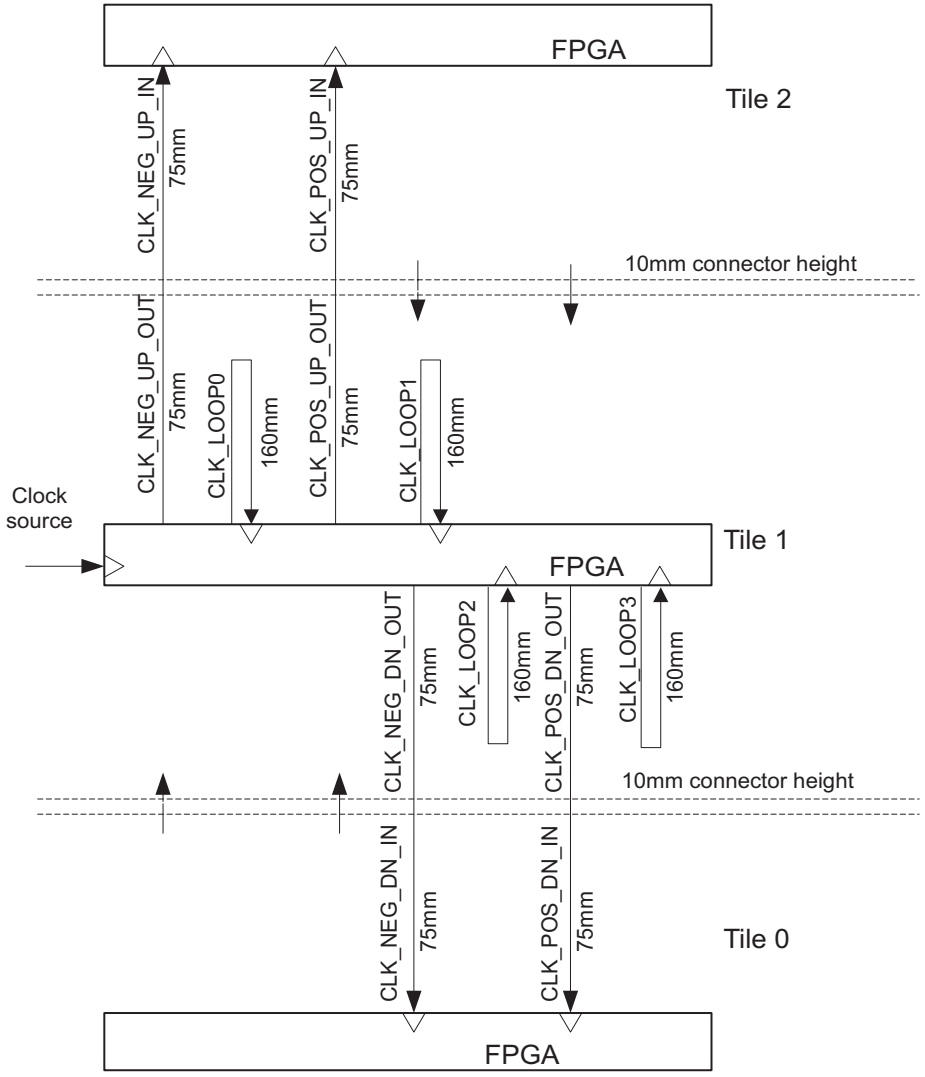

Figure 3-9 Retimed clock routing for three tiles (from middle tile)

3.4.4 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 so 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 3-11 on page 3-24.

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 3-11 on page 3-24.

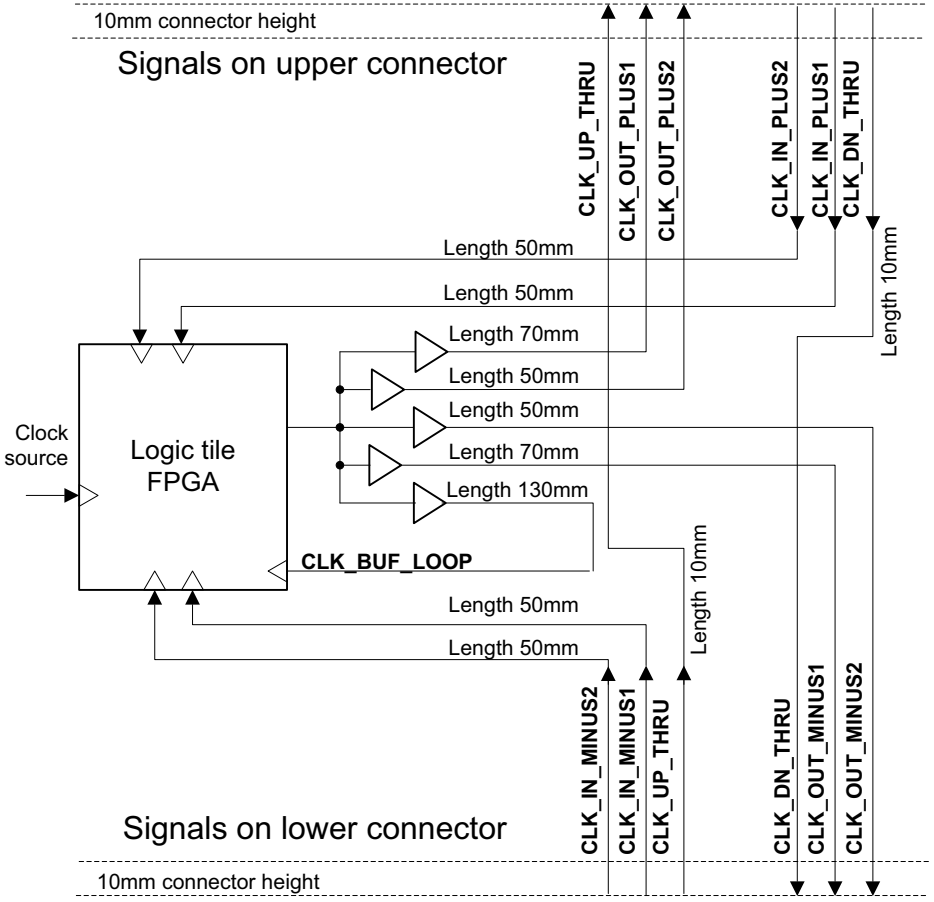

Figure 3-10 Delay-matched clocking scheme

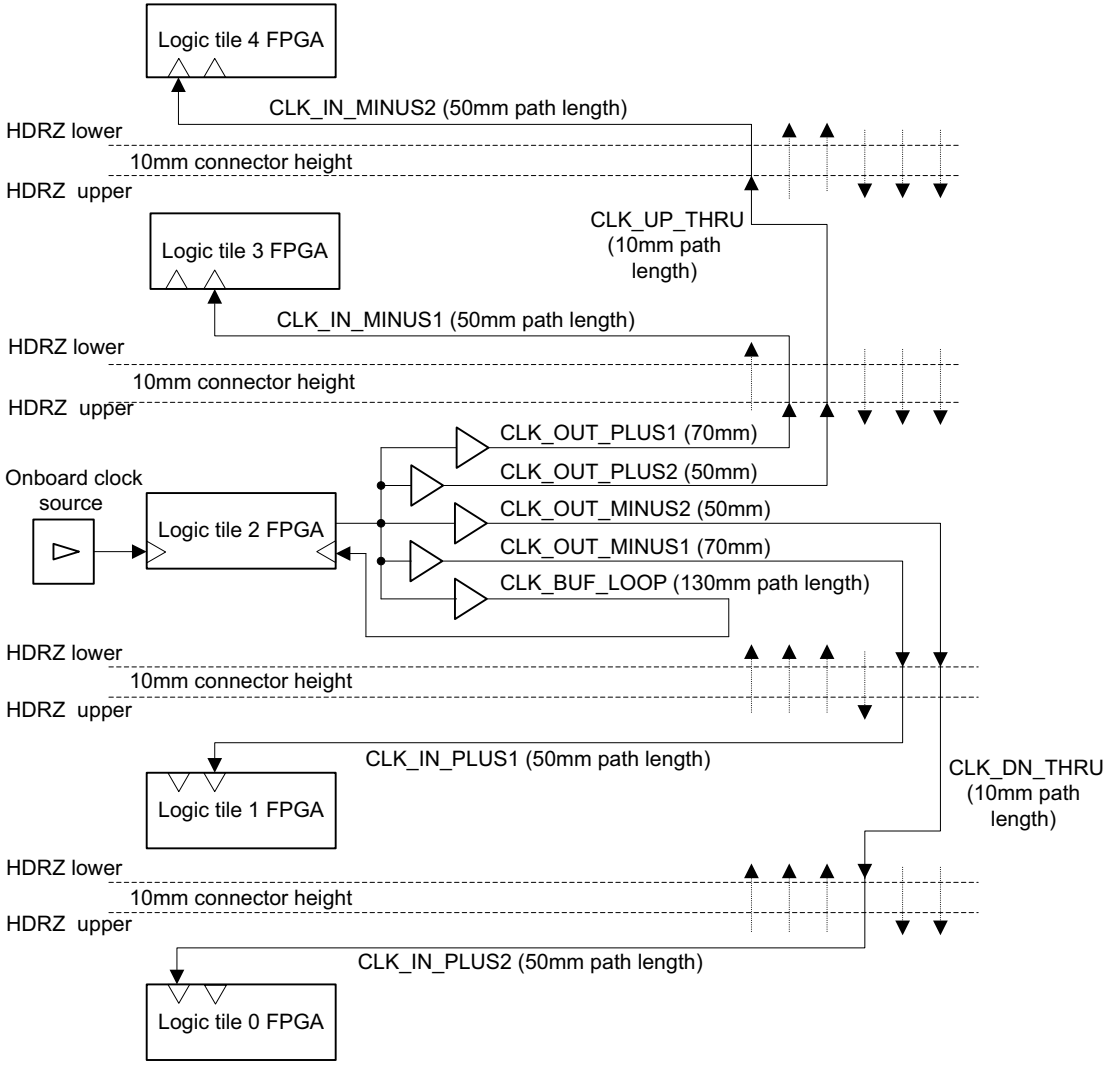

Figure 3-11 Delay-matched clocks for five tiles

3.5 Reset control

The Logic Tile has several reset signals (most are inputs from the Interface Module or baseboard). Figure 3-13 on page 3-26 shows the power-on reset sequence. Figure 3-12 shows the architecture of the reset system. Table 3-6 on page 3-27 describes the global reset signals.

Note

You can use the reconfigure pushbutton on the Interface Module or VPB/926EJ-S to reload the FPGA image without resetting the entire system. See the *Integrator/IM-LT1 User Guide* or the *Versatile Platform Baseboard for ARM926EJ-S User Guide* for details.

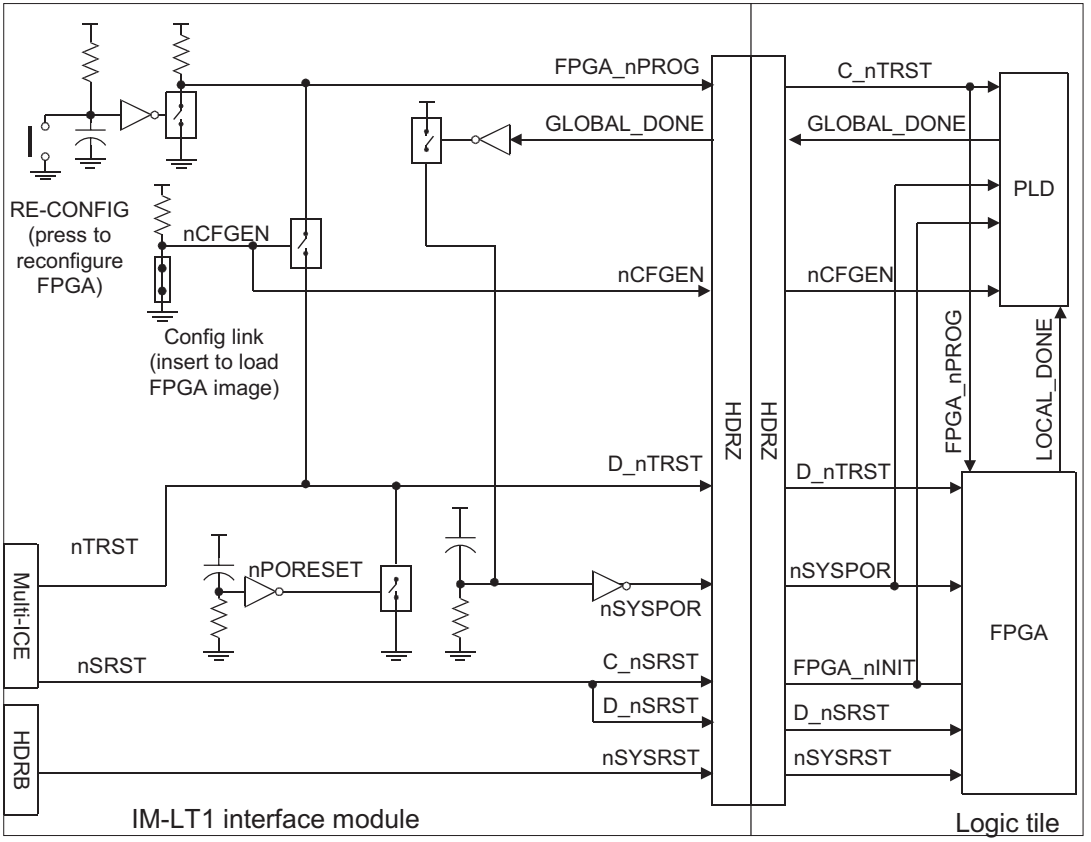

Figure 3-12 Reset and image loading control

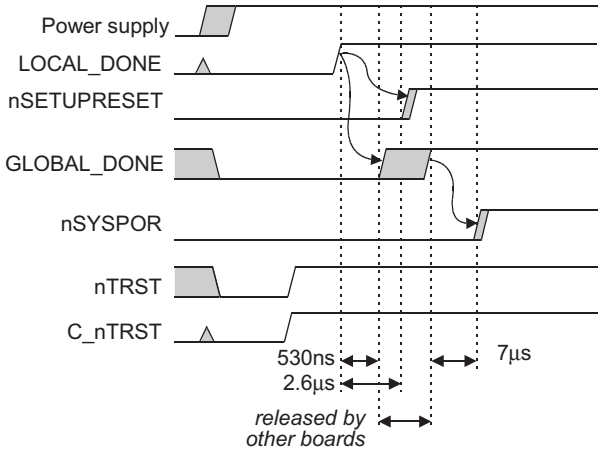

Figure 3-13 Power-on reset timing

Note

The release time for GLOBAL_DONE depends on other boards in the system. It might be held low longer if other boards take longer to configure.

Table 3-6 Reset signal descriptions

Name	Description	Function
nPORESET	Power-on reset	This signal generates the **D_nTRST** 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 to control the initialization pin (**nINIT**) on the FPGAs. 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 platform board (for example the Integrator/AP and IM-LT1). 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 *Integrator/AP Motherboard User Guide* or the *Integrator/IM-LT1 Interface Module User Guide* for more information on motherboard signals.)
FPGA_nPROG	Configuration reload	The **FPGA_nPROG** signal forces all 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 analogue circuitry on the IM-LT1/2 Interface Module. It can be used to reset user logic if required. It remains active for between 1 and 10µS after **GLOBAL_DONE** goes HIGH.

Table 3-6 Reset signal descriptions (continued)

Name	Description	Function
D_nTRST	TAP controller reset	A system-wide, open collector signal that can be driven LOW by Multi-ICE. This is the debug version of the **nTRST** signal. It connects 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: • Multi-ICE connector • Trace (embedded trace macrocell) connector.
D_nSRST	Multi-ICE system reset	A system-wide, open collector signal that can be driven LOW by Multi-ICE. This is the debug version of the **nSRST** signal.
C_nTRST	TAP controller reset	An open-collector signal that can be driven LOW by Multi-ICE when using the Progcards utility to program the FPGA. (This signal connects to the **PROG** pin of the FPGA.) This is the config version of the **nTRST** signal.
C_nSRST	Multi-ICE system reset	A system-wide, open-collector signal. The IM-LT1/2 shorts this signal to **D_nSRST**. This signal can be driven LOW by Multi-ICE when using the Progcards utility to program the FPGA. (This signal connects to the **INIT** pin of the FPGA.) This is the config version of the **nSRST** signal. **Note** On the Integrator/AP, the expansion connector (EXPB) **nSRST** signal is completely separate from the core module (HDRB) **nSRST** signal (see the *Integrator/AP User Guide* for more details).
nSETUPRESET	Configuration control	This signal is an output from the PLD indicating to the FPGA that serial foldover configuration transfers can begin.

3.6 Memory

The Logic Tile provides two 2MB ZBT SSRAM devices and one 8MB flash memory device. Figure 3-14 shows the block diagram of the memory system.

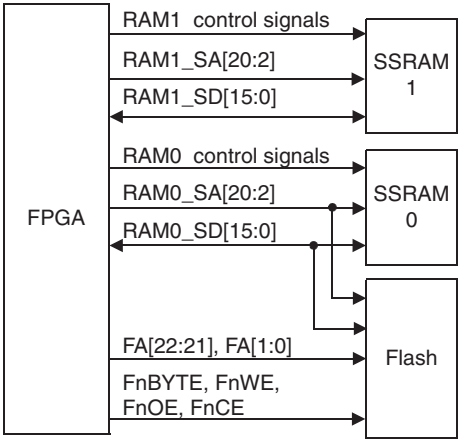

Figure 3-14 Memory block diagram

3.6.1 SSRAM

Two 512K x 32-bit ZBT SSRAMs are provided with separate address, data, and control signals routed to the FPGA.

Note

The flash and SSRAM share some address and data signals. If spare flash memory is not being used by a user design, the flash signals **FnOE** and **FnWE** must be driven high to inhibit accidental flash reads or writes.

3.6.2 Flash memory

You use flash memory for FPGA configuration. Configuration is managed by the configuration PLD based on the state of **SEL1**, **SEL2**, and **FPGA_IMAGE** (see *PLD bytestreamer operation* on page 4-4).

Two FPGA images (from `.bit` files) can be stored in the flash. Depending on the size of the FPGA fitted, there are some unused areas of the flash device (see Figure 3-15 on page 3-30). The unused area is available for general purpose use, but care must be taken not to corrupt the FPGA configuration image. Use the Progcards utility to program binary files (`.bin`) into the unused areas of the flash.

You can use flash memory in either 8 or 16-bit mode. (FPGA configuration, however, is always done in 8-bit mode.)

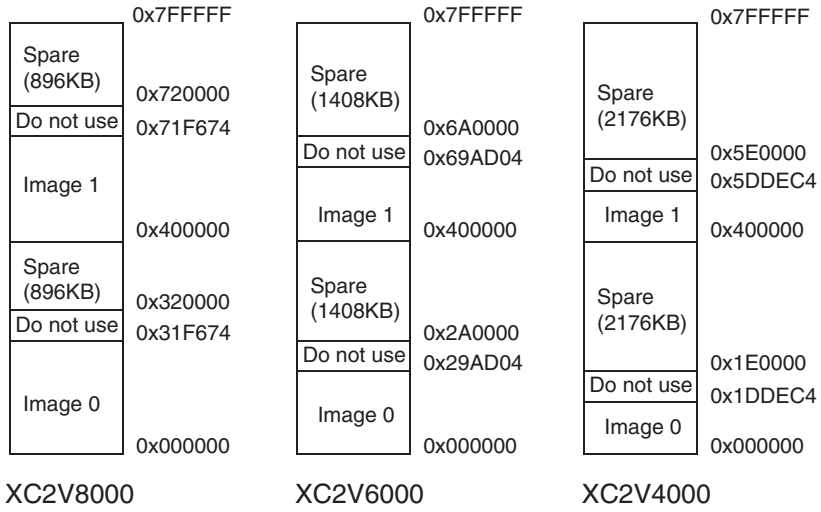

Figure 3-15 Spare locations in flash memory

Note

Figure 3-15 shows values for uncompressed bit files. If you are using bit-file compression, the location and size of the *Do not use* area might be changed. The *Do not use* area is because of the 128KB block size of the flash.

Caution

Performing a block erase in the regions marked *Do not use* corrupts the FPGA image.

3.7 JTAG support

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

- For the VPB/926EJ-S, the JTAG signals for the Logic Tile are routed through the headers to the tile at the top of the stack and from there back down through the tile. Use the JTAG or USB debug connector on the VPB/926EJ-S.

  If multiple RealView Logic Tiles are stacked on a Versatile/PB926EJ-S, the JTAG equipment is always connected to the Versatile/PB926EJ-S and the signals are routed upwards to the top tile and then back down to the Versatile/PB926EJ-S.

- For use standalone or with an Integrator product, the Integrator/IM-LT1 Interface Module provides the JTAG connector for accessing the Logic Tile. See the *Integrator/IM-LT1 Interface Module User Guide* for more information on using JTAG and Multi-ICE.

There are two separate JTAG paths through the Logic Tile:

- One path is for the configuration of programmable devices (FPGA, PLD, and flash memory). These JTAG signals are identified by the **C_** prefix.

- One path used in debug mode, connects to a TAP controller if one has been synthesized into the FPGA. These JTAG signals are identified by the **D_** prefix.

The JTAG path chosen depends on whether the system is in configuration mode or debug mode. The CONFIG link on the baseboard or Interface Module controls the **nCFGEN** signal that is routed through the Interface Module and 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 FPGAs, flash memory, or PLDs in the system.

  If an Interface Module connects to a motherboard, such as the Integrator/AP, the motherboard is also set into config mode.

  Note

  A manual configuration file (included on the CD supplied with the tile) must be used to configure Multi-ICE. The configuration file sets the JTAG **TCK** speed to 1MHz to ensure reliable operation (see also the *Multi-ICE User Guide*).

- Remove the CONFIG link on the baseboard or Interface Module for debug mode. If the Logic Tile contains a virtual TAP controller it is placed in the debug JTAG path.

Note

If the Logic Tile and Interface Module are mounted on an Integrator motherboard, the JTAG signals are routed from the Interface Module down to the motherboard and back up to the Interface Module and then up to the stacked Logic Tiles.

If the Logic Tile is mounted on a VPB/926EJ-S, the JTAG signals are routed from the baseboard up to the stacked Logic Tiles.

Figure 3-16 on page 3-33 shows the JTAG data signal routing for an Integrator system. (The config link is closed, so the figure shows a system in config mode.) Figure 3-17 on page 3-34 shows the JTAG clock routing, and Figure 3-18 on page 3-35 shows the JTAG TMS routing. Pullup 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 *virtual* TAP controller. This facility is provided for FPGA designs that require a TAP controller, for example, designs that include a synthesized processor.

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**.

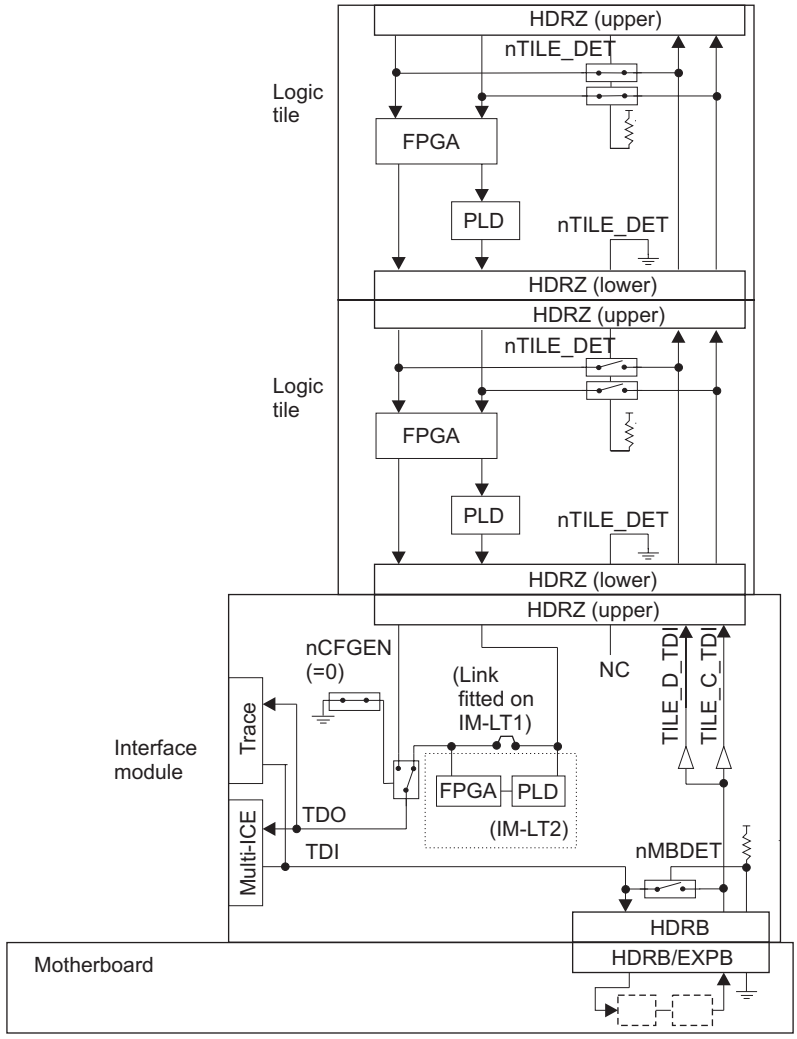

Figure 3-16 JTAG data paths

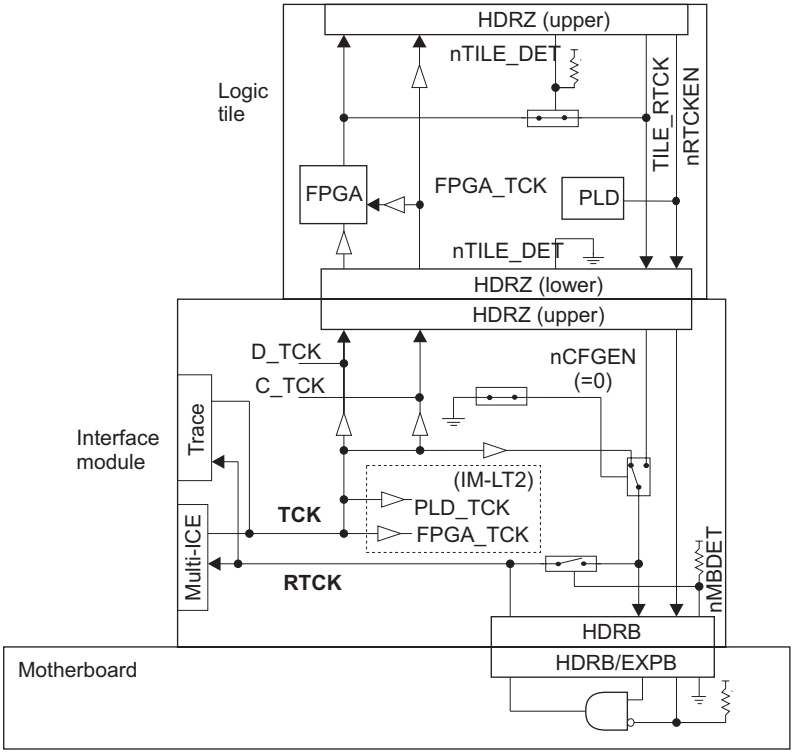

Figure 3-17 JTAG clock paths

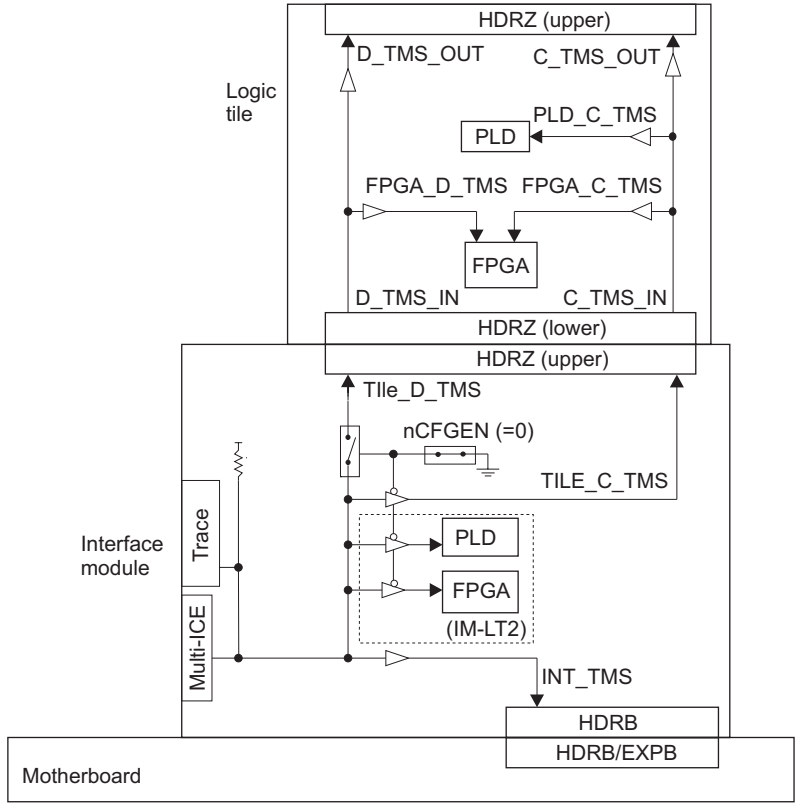

Figure 3-18 JTAG TMS paths

3.8 Test points

The test points listed in Table 3-7 enable you to measure clock frequencies and FPGA core temperature.

Table 3-7 Test points

Test Point	Signal	Description
TP1	**CLK_GLOBAL_IN**	Buffered version of global clock signal. This is fed to a **GCLK** input of the FPGA.
TP2	**CLK_GLOBAL_OUT**	Buffered version of global clock signal from the FPGA. (This signal is driven even if **CLK_GLOBAL_OUT** is disabled.)
TP3	**CLK_24MHZ**	Buffered replica of the fixed-frequency reference clock signal that feeds **CLK_24MHZ_FPGA** and **CLK_24MHZ_PLD**.
TP8	**CLK_OUT**	Buffered version of clock signal **CLK_OUT_TO_BUF** that drives the buffers for the dual/differential clocking scheme: • CLK_OUT_PLUS1 • CLK_OUT_PLUS2 • CLK_OUT_MINUS1 • CLK_OUT_MINUS2 • CLK_BUF_LOOP.
TP9	**CLK_SCLK**	Buffered version of serial data input clock to programmable oscillators
J7	**DXP** and **DXN**	Temperature sensing diode output. (There is not a socket fitted to the board, but you can fit a socket or measure the voltage directly from the socket pads.)

Chapter 4
Configuring the FPGA and PLD

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

The FPGA image can be loaded from the flash memory (debug mode) or from the JTAG interface (configuration mode). Figure 4-1 shows the architecture of the FPGA configuration system and Figure 4-2 on page 4-3 shows the reload timing sequence.

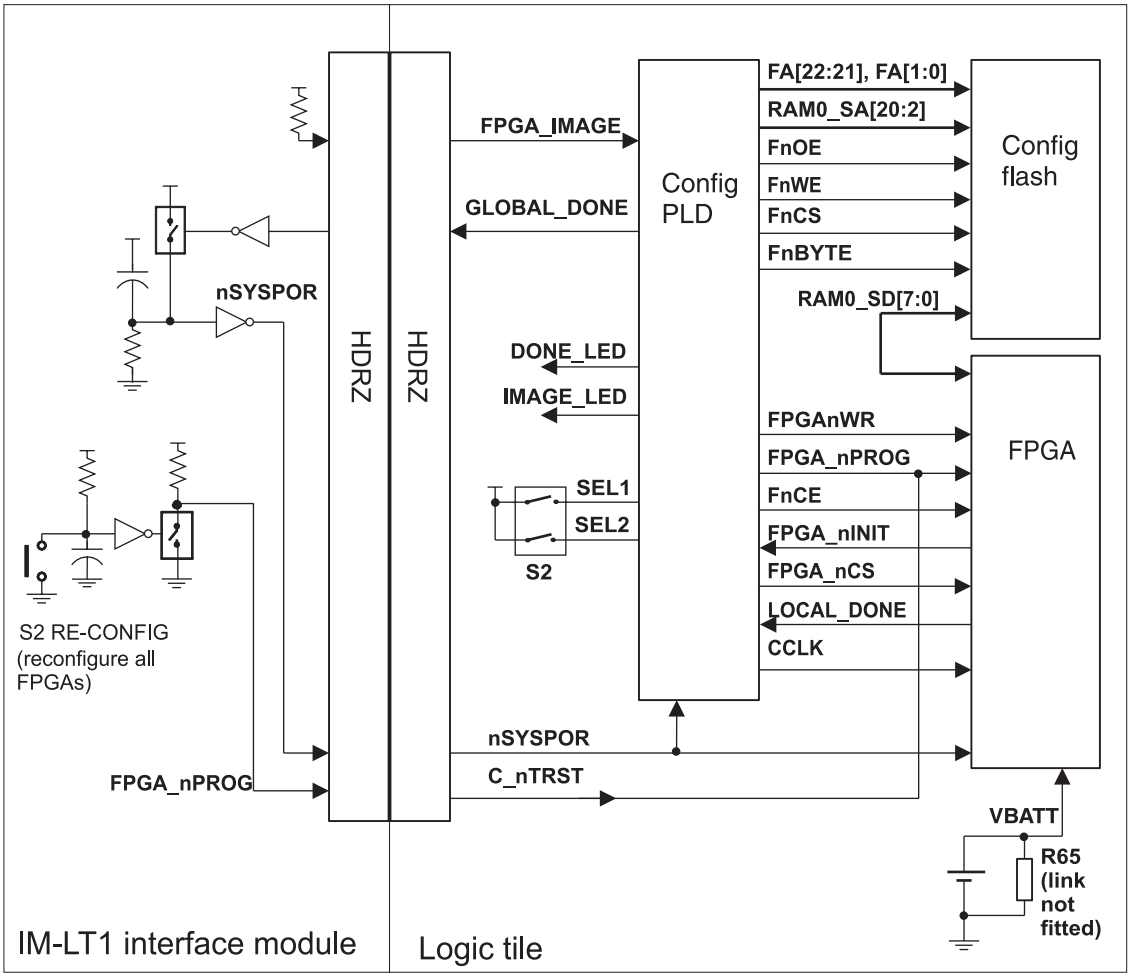

Figure 4-1 FPGA configuration architecture

Note

The FPGA_IMAGE signal is pulled high on the IM-LT1. This selects the high flash image for the FPGA if you are using this signal to select the image. To select the low flash image, use the image selection switches on the Logic Tile.

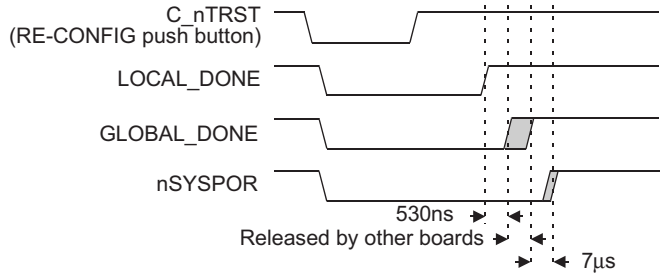

Figure 4-2 Reconfigure sequence

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 R65 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 PLD bytestreamer operation

The configuration flash contains two images that enable the FPGA to be configured at power-on reset. The configuration switches and the global configuration signal controls the image selection.

The image load sequence consists of:

- the PLD reading the **FPGA_IMAGE** signal and the **SEL1** and **SEL2** signals (DIP switch S2[2:1]) and selecting the lower or upper flash image accordingly (see *Configuring the FPGA from flash* on page 4-7 for details)
- setting the **IMAGE_LED** signal to indicate which image is selected for the FPGA
- bytestreaming the flash image into the FPGA
- setting the **DONELED** signal LOW after 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 4-7.)

4.1.2 Debug mode

In debug mode, the FPGA uses the *slave select map mode* configuration method. In this mode, **CCLK** is an input to the FPGA and is driven by the configuration PLD. The FPGA configuration is loaded from flash memory and the process is managed by the configuration PLD. The flash must contain valid configuration data and the CONFIG link must not be fitted.

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 IM-LT1 Interface Module or a motherboard (see *Configuring the FPGA from flash* on page 4-7).

4.1.3 Configuration mode

In configuration mode, the *FPGA uses the JTAG mode* configuration method. In this mode, the FPGA is configured directly form the JTAG connector on the Interface Module. You can use Multi-ICE to download a configuration directly to the FPGA if the CONFIG link is fitted on the Interface Module (see *Reconfiguring the FPGA directly* on page 4-9).

4.2 FPGA tool flow

Preparing FPGA configuration files entails two steps:

1. Synthesis.
2. Place and route.

Figure 4-3 illustrates the basic tool flow process.

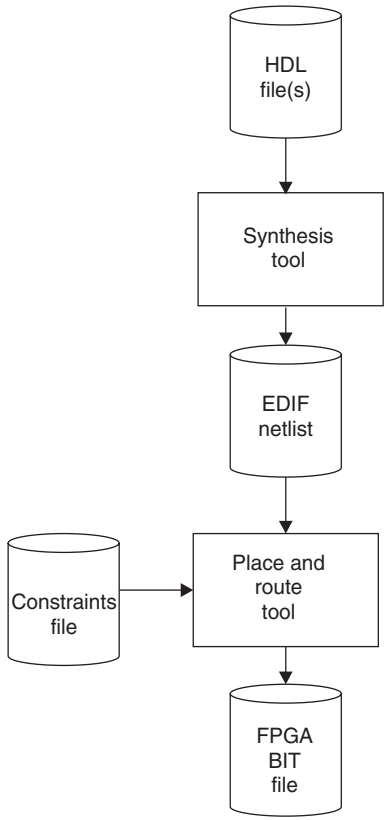

Figure 4-3 Basic tool flow

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 net list targeted at a particular technology. In the case of Xilinx Virtex II, there are several synthesis tools available for both Windows and UNIX platforms, that provide support for a variety of programmable logic vendors.

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.

See the documentation for your particular software tool for more information.

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

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 programs the FPGA. The `.bit` file is aimed at a particular device, taking into account the device size, package type, and speed grade.

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 directly configures 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 files (`.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.

Current Xilinx tools (ISE 5.1) support bit-file compression. This reduces the size of the bit file and reduces download time.

4.3 Configuring the FPGA from flash

The flash memory has space to store up to two configurations for Xilinx Virtex II FPGAs. The configuration image is selected when the Logic Tile is powered according to the setting of S2[2:1] and the **FPGA_IMAGE** signal from the motherboard:

- if S2[1] is OFF, then S2[2] selects the image
- if S2[1] is ON, the signal **FPGA_IMAGE** signal selects the image.

Table 4-1 shows the FPGA image selection options.

Table 4-1 Image selection for XC2V FPGAs

S2[1]	S2[2]	FPGA_IMAGE signal	Flash image used	Image LED	Image base address
OFF	OFF	x	0	Unlit	`0x000000`
OFF	ON	x	1	Lit	`0x400000`
ON	x	0	0	Unlit	`0x000000`
ON	x	1	1	Lit	`0x400000`

Note

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

See the CD for a description of the example images stored in flash when the Logic Tile is shipped.

4.3.1 Downloading new FPGA configurations into FLASH

The flash memory on the Logic Tile configures the FPGA on power-up when the CONFIG link is not fitted. The `Progcards` utility programs 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.

The steps in writing a bit file to flash are similar to those described in *Reconfiguring the FPGA directly* on page 4-9. 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.

Note

The Progcards utility requires Multi-ICE release 1.4 or higher. Progcards version 2.40 or higher must be used.

For a full description of this utility, see 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` for your design. This is a configuration file for `progcards.exe`.
3. Put the Logic Tile in configuration mode by fitting the CONFIG link on the baseboard or Interface Module.
4. Configure the Multi-ICE server using a configuration file. For example, `ltxc2v6000_with_imlt1.cfg`.
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.
6. Wait until `Progcards` utility finishes and then remove the CONFIG link.
7. Power-cycle the system or press the reconfigure button on the baseboard or Interface Module.

Note

You must manually configure the Multi-ICE server to program devices on the tile. Manual configuration files are provided on the CD supplied with the Logic Tile. You must also ensure that the `irlength.arm` file in the Multi-ICE directory contains one of the following lines, as appropriate for the FPGA fitted to your Logic Tile:

```
XC2V4000=6        ;for 4000 size FPGA

XC2V6000=6        ;for 6000 size FPGA

XC2V8000=6        ;for 8000 size FPGA
```

See the *Multi-ICE User Guide* for more information about using Multi-ICE.

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 XC2V6000 takes approximately 100 seconds to complete using Multi-ICE on a fast workstation.

4.4.1 Using Multi-ICE

You can reprogram the FPGA directly (only on a one-tile system) using Multi-ICE. A Multi-ICE client application called Progcards is provided to read `.bit` files and configure the FPGA using the Multi-ICE hardware.

You must use a board file (`.brd`) to tell the `progcards` utility about the method of programming (JTAG). Examples are provided on the CD supplied with the Logic Tile.

Note

The Progcards utility requires Multi-ICE release 1.4 or higher. Progcards version 2.40 or higher must be used.

For a full description of this utility, see 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.exe`.
3. Put the Logic Tile in configuration mode by fitting the CONFIG link.
4. Configure the Multi-ICE server using a configuration file. For example, `ltxc2v6000_with_imlt1.cfg`.
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.

Note

You must manually configure the Multi-ICE server to program devices on the tile. Manual configuration files are provided on the CD supplied with the Logic Tile. You must also ensure that the `irlength.arm` file in the Multi-ICE directory contains one of the following lines, as appropriate for the FPGA fitted to your Logic Tile:

```
XC2V4000=6        ;for 4000 size FPGA

XC2V6000=6        ;for 6000 size FPGA

XC2V8000=6        ;for 8000 size FPGA
```

See the *Multi-ICE User Guide* for more information about using Multi-ICE.

4.4.2 Using Xilinx download tools

There is not a dedicated connector for a 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. (The 10-way HW-FLYLEAD available from Xilinx enables the Xilinx Parallel Cable IV to connect to the 20-way box header.)

Note

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

4.5 Reprogramming the PLD

The Logic Tile is supplied with the PLD already programmed. The information in this section is provided, however, in case of accidental erasure of the PLD.

Caution

You are advised not to reprogram this device with any images other than those provided by ARM Limited.

Program the PLD as follows:

1. Put the Logic Tile into configuration mode by fitting the CONFIG link on the baseboard or Interface Module and power-up.
2. Start the Multi-ICE server and the load the configuration file (`.cfg`) for your Logic Tile.
3. Run the `Progcards` utility from the *`install_directory`*`\lt-xc2v4000\boardfiles\` directory.
4. Choose the required PLD image.

Appendix A
Pinouts and Specifications

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

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.

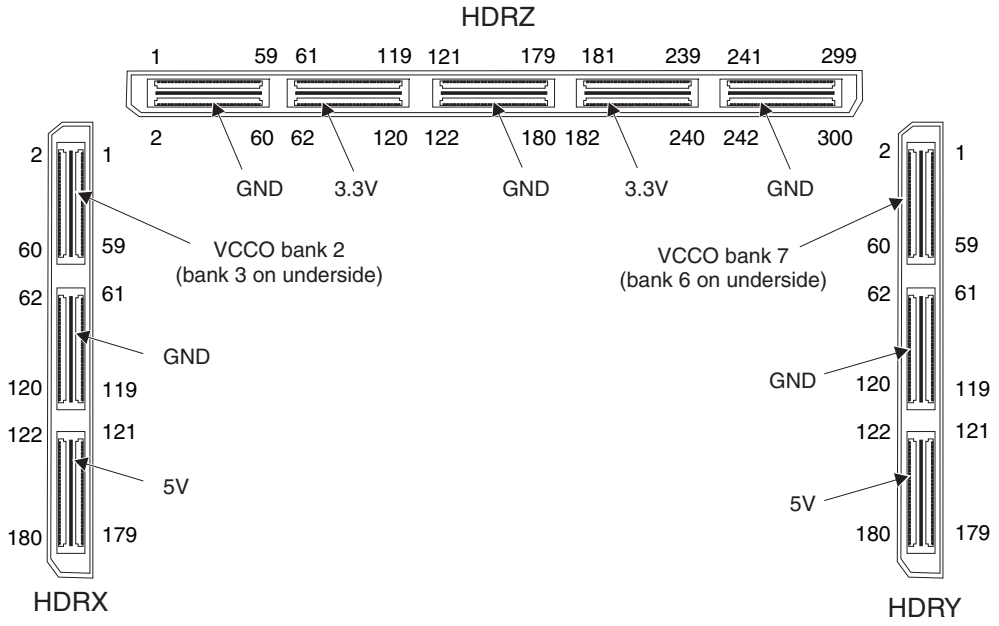

Figure A-1 HDRX, HDRY, and HDRZ (upper) pin numbering

Note

The foldover logic can reroute some of the signals normally on the upper header connectors back to the lower connectors. See *Header signals* on page 3-4 for details.

A.1.1 HDRX and HDRY

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.)

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**
Xx106	33	34	**Xx73**
Xx107	35	36	**Xx72**
Xx108	37	38	**Xx71**
Xx109	39	40	**Xx70**
Xx110	41	42	**Xx69**

Table A-1 HDRX signals (continued)

	Odd pins	Even pins	
Signal	**Pin**	**Pin**	**Signal**
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**
Xx130	81	82	**Xx49**
Xx131	83	84	**Xx48**
Xx132	85	86	**Xx47**
Xx133	87	88	**Xx46**
Xx134	89	90	**Xx45**

Table A-1 HDRX signals (continued)

	Odd pins	Even pins	
Signal	**Pin**	**Pin**	**Signal**
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**
Xx154	129	130	**Xx25**
Xx155	131	132	**Xx24**
Xx156	133	134	**Xx23**
Xx157	135	136	**Xx22**
Xx158	137	138	**Xx21**

Table A-1 HDRX signals (continued)

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

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.)

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**
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**

Table A-2 HDRY signals (continued)

	Odd pins	Even pins	
Signal	**Pin**	**Pin**	**Signal**
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**
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**

Table A-2 HDRY signals (continued)

	Odd pins	Even pins	
Signal	**Pin**	**Pin**	**Signal**
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**
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**

Table A-2 HDRY signals (continued)

	Odd pins	Even pins	
Signal	**Pin**	**Pin**	**Signal**
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**
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.1.2 HDRZ

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

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

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**
Z144	**Z144**	33	34	**ZU239**	**ZL239**
Z145	**Z145**	35	36	**ZU238**	**ZL238**
Z146	**Z146**	37	38	**ZU237**	**ZL237**

Table A-3 HDRZ signals (continued)

		Odd pins	Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
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**
Z168	**Z168**	81	82	**ZU215**	**ZL215**
Z169	**Z169**	83	84	**ZU214**	**ZL214**

Table A-3 HDRZ signals (continued)

		Odd pins	Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
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**
D_nSRST	**D_nSRST**	129	130	**CLK_POS_DN_IN**	**CLK_POS_DN_OUT**

Table A-3 HDRZ signals (continued)

		Odd pins	Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
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**
Z62	**Z62**	175	176	**Z65**	**Z65**

Table A-3 HDRZ signals (continued)

		Odd pins	Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
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**

Table A-3 HDRZ signals (continued)

		Odd pins	Even pins		
Signal (lower)	**Signal (upper)**	**Pin**	**Pin**	**Signal (upper)**	**Signal (lower)**
Z15	**Z15**	269	270	**Z112**	**Z112**
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 II FPGA supports I/O signalling levels that can require VCCO of 1.5V, 1.8V, 2.5V, or 3.3V. See the electrical characteristics section of the Virtex II user guide for details on FPGA I/O. See also *Variable I/O levels* on page 3-6 for information on configurable I/O signal levels.

A.2.2 Current requirements

Table A-5 shows the current requirements at room temperature and nominal voltage for a Logic Tile implementing an ARM966 core at 6MHz.

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.

Table A-5 Current requirements

At 3.3V	At 5V
1A	100mA

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.

Note

The PCB Gerber files are available from the ARM website at:
`www.arm.com/support/downloads/versatile.html`

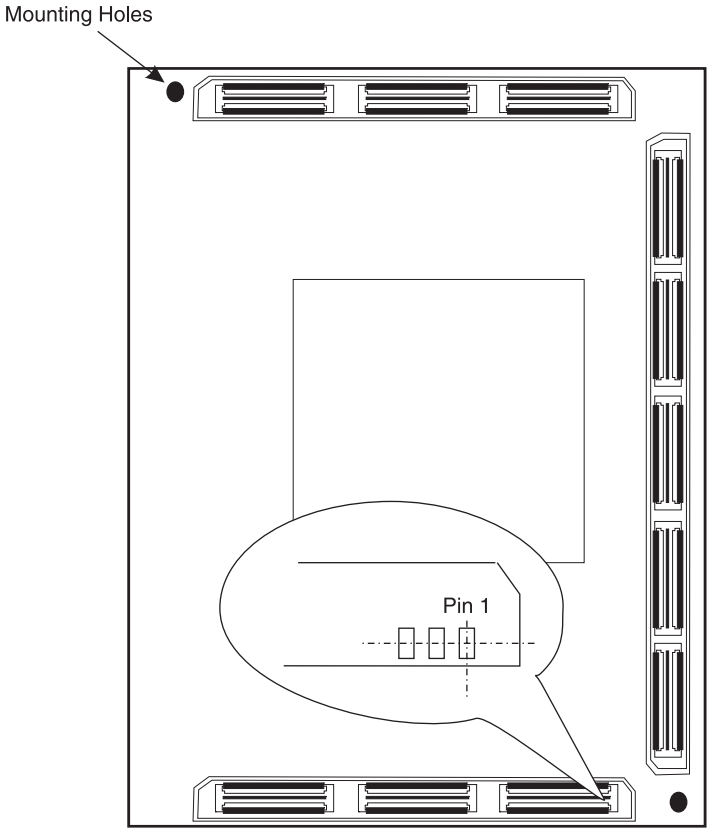

Figure A-2 Board outline

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-090-02-F-D-A-K
HDRXL	QSH-090-01-F-D-A-K
HDRYU	QTH-090-02-F-D-A-K
HDRYL	QSH-090-01-F-D-A-K
HDRZU	QTH-150-02-F-D-A-K
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 -02- connectors and have 8mm board separation. The Logic Tiles have a maximum component height of 2.5mm on the bottom and 5mm on the top of the board. This ensures that there are no component interference problems with mated boards.

Index

The items in this index are listed in alphabetical order, with symbols and numerics appearing at the end. The references given are to page numbers.

A

B

C

D

E

F