Core Tile

HBI-0131(CT926EJ-S and CT1136JF-S) HBI-0141 (CT7TDMI and CT7TDMI-S)

User Guide

ARM®

ARM DUI 0273G (ID060911)

Core Tile
User Guide

Release Information

Description	Issue	Confidentiality	Change
October 2004	A	Non-Confidential	First release (ARM926EJ-S and ARM1136JF-S versions)
February 2005	B	Non-Confidential	Second release (includes ARM7TDMI and ARM7TDMI-S)
August 2006	C	Non-Confidential	Third release updated to fix various documentation defects
May 2007	D	Non-Confidential	Fourth release updated to fix various documentation defects
October 2007	E	Non-Confidential	Fifth release updated to fix a documentation defect
January 2009	F	Non-Confidential	Sixth release updated to fix a documentation defect
June 2011	G	Non-Confidential	Seventh release updated to fix a documentation defect

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.

Unrestricted Access is an ARM internal classification.

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 Core 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
Core Tile User Guide

Part B Printed Circuit Board HBI-0131 (CT926EJ-S and CT1136JF-S)

Chapter 3 HBI-0131 Hardware Description

Chapter 4 HBI-0131 Features specific to the CT926EJ-S

Chapter 5 HBI-0131 Features specific to the CT1136JF-S

Chapter 6 HBI-0131 Signal Descriptions

Part C Printed Circuit Board HBI-0141 (CT7TDMI and CT7TDMI-S)

Chapter 7 HBI-0141 Hardware Description

List of Tables
Core Tile User Guide

List of Figures
Core Tile User Guide

Preface

This preface introduces the Core Tiles and their reference documentation. It contains the following sections:

About this document

This document describes how to set up and use Core Tiles.

Intended audience

This document has been written for experienced hardware and software developers to aid the development of ARM-based products using the Core Tiles as part of a development system.

Organization

This document is organized as follows:

Chapter 1 ***Introduction***

Read this chapter for an introduction to Core Tiles.

Chapter 2 ***Getting Started***

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

Chapter 3 ***HBI-0131 Hardware Description***

Read this chapter for a description of the hardware architecture of the CT926EJ-S and CT1136JF-S Core Tiles that use the HBI-0131 printed circuit board.

Chapter 4 ***HBI-0131 Features specific to the CT926EJ-S***

Refer to this chapter for features that are specific to the CT926EJ-S Core Tiles that use the HBI-0131 printed circuit board.

Chapter 5 ***HBI-0131 Features specific to the CT1136JF-S***

Refer to this chapter for features that are specific to the CT1136JF-S Core Tiles that use the HBI-0131 printed circuit board.

Chapter 6 ***HBI-0131 Signal Descriptions***

Refer to this chapter for connector pinouts for the CT926EJ-S and CT1136JF-S Core Tiles that use the HBI-0131 printed circuit board.

Chapter 7 ***HBI-0141 Hardware Description***

Read this chapter for a description of the hardware architecture of the CT7TDMI and CT7TDMI-S Core Tiles that use the HBI-0141 printed circuit board.

Chapter 8 ***HBI-0141 Signal Descriptions***

Refer to this chapter for connector pinouts for the CT7TDMI and CT7TDMI-S Core Tiles that use the HBI-0141 printed circuit board.

Appendix A ***Static Memory Expansion Board***

Refer to this appendix for details of the optional memory expansion board.

Appendix B ***Specifications***

Refer to this appendix for electrical and mechanical specifications that apply to all Core Tiles.

Typographical conventions

The following typographical conventions are used in this document:

bold	Highlights ARM signal names within text, and interface elements such as menu names. This style is also used for emphasis in descriptive lists where appropriate.
italic	Highlights special terminology, cross-references and citations.
`monospace`	Denotes text that can be entered at the keyboard, such as commands, file names 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 or 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 following publications by ARM Limited provide additional information on using the Core Tile and related products:

- *ARM926EJ-S Technical Reference Manual* (ARM DDI 0198)
- *ARM946E-S Technical Reference Manual* (ARM DDI 0155)
- *ARM966E-S Technical Reference Manual* (ARM DDI 0164)
- *ARM7TDMI Technical Reference Manual* (ARM DDI 0210)
- *ARM7TDMI-S Technical Reference Manual* (ARM DDI 0084)
- *ARM1136JF-S and ARM1136J-S Technical Reference Manual* (ARM DDI 0211)
- *ETM7 Technical Reference Manual* (ARM DDI 0158)
- *ETM9 Technical Reference Manual* (ARM DDI 0157)
- *ETM10 Technical Reference Manual* (ARM DDI 0206)
- *ETB Technical Reference Manual* (ARM DDI 0242)
- *RealView Platform Baseboard for ARM926EJ-S User Guide* (ARM DUI 0224)
- *ARM RealView Logic Tile LT-XC2V4000+ User Guide* (ARM DUI 0186)
- *ARM RealView AT1 Analyzer Tile User Guide* (ARM DUI 0189)
- *ARM RealView IT1 Interface Tile User Guide* (ARM DUI 0188)
- *ARM Integrator IM-LT1 Interface Module User Guide* (ARM DUI 0187)
- *ARM Integrator IM-LT3 Interface Module User Guide* (ARM DUI 0216)
- *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)
- *ADS Tools Guide* (ARM DUI 0067)
- *ADS Debuggers Guide* (ARM DUI 0066)
- *ADS Debug Target Guide* (ARM DUI 0058)
- *ADS Developer Guide* (ARM DUI 0056)
- *ADS CodeWarrior IDE Guid*e (ARM DUI 0065)
- *RealView Debugger User Guid*e (ARM DUI 0153)
- *RealView Compilers and Libraries Guid*e (ARM DUI 0205)
- *RealView Linker and Utilities Guid*e (ARM DUI 0206)
- *RealView ICE User Guide* (ARM DUI 0155).

Feedback

ARM Limited welcomes feedback both on the ARM Core Tiles and on the documentation.

Feedback on this document

If you have any comments about this document, 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 ARM Core Tiles

If you have any comments or suggestions about these products, contact your supplier giving:

- the product name
- an explanation of your comments.

Part A
Introduction

Chapter 1
Introduction

This chapter introduces the Core Tiles. It contains the following sections:

Note

This guide covers both the HBI-0131 and HBI-0141 printed-circuit boards.

These boards support a variety of test chips, for example, the ARM926EJ-S and the ARM7TDMI. The availability of individual Core Tiles however, depends on the availability of test chips. Contact your sales representatives for details on currently available Core Tiles.

1.1 About the Core Tiles

Core Tiles are a compact development platform that enable you to develop products based on ARM processors. Examples of using the Core Tile with compatible Versatile or Integrator family boards are listed in Table 1-1.

Table 1-1 Product combinations with Core Tile

Configuration	Interface Tile	Logic Tile	Interface module	Baseboard
Minimal standalone Core Tile system. The FPGA in the IM-LT3 provides the system controller functions of a motherboard and drives clock selection and other control signals on the Core Tile.	Not used	Not used	IM-LT3	None
Standalone Core Tile system. The FPGA in the Logic Tile provides the system controller functions of a motherboard and drives clock selection and other control signals on the Core Tile. The IM-LT1 provides power and JTAG connections.	Optional	Required	IM-LT1	None
Integrator/CP system. The FPGA in the IM-LT3 provides the interface between the buses on the Core Tile and the Compact Platform and drives clock selection and other control signals on the Core Tile. **Note** Integrator/CP system can also use the IM-LT1 and a Logic Tile. The FPGA in the Logic Tile provides the interface between the bus on the Core Tile and the Compact Platform and drives clock selection and other control signals on the Core Tile. An FPGA image is not provided for this configuration.	Optional	Optional	IM-LT3	Integrator Compact Platform
Versatile/PB926EJ-S system with additional processors. The Logic Tile provides the interface between the bus on the Versatile/PB926EJ-S and the Core Tile and drives clock selection and other control signals on the Core Tile.	Optional	Required	None	Versatile/ PB926EJ-S
Third-party custom development system.	Optional	Optional	Third-party	Third party

For all configurations, a Versatile/IT1 Interface Tile or Versatile/AT1 Analyzer Tile can be used to provide access to signals on the tile header connectors.

The Core Tile must be used in conjunction with a Logic Tile (or IM-LT3) that implements the system and memory controllers in an FPGA. Through-board connectors on tile products allow stacking of multiple systems. Multiple combinations of Core Tile and Logic Tile can be used to create a multiprocessor system.

The Core Tile and Logic Tile do not have power or JTAG connectors. The tiles must be stacked on an Interface Module or baseboard that provides power and JTAG connections. The Core Tiles require a reference clock (or clocks) supplied by an attached Logic Tile or baseboard.

1.2 Overview of Core Tiles

The major components on the Core Tile are as follows:

- microprocessor test chip
- Versatile-style tile headers on the top and bottom of the board
- connectors for memory expansion
- connectors for Trace Port Analyzer (not all Core Tiles support trace)
- bus interface logic
- clock selection logic
- controllable power supply for the test chip core.

Figure 1-1 shows the layout of a Versatile/CT926EJ-S.

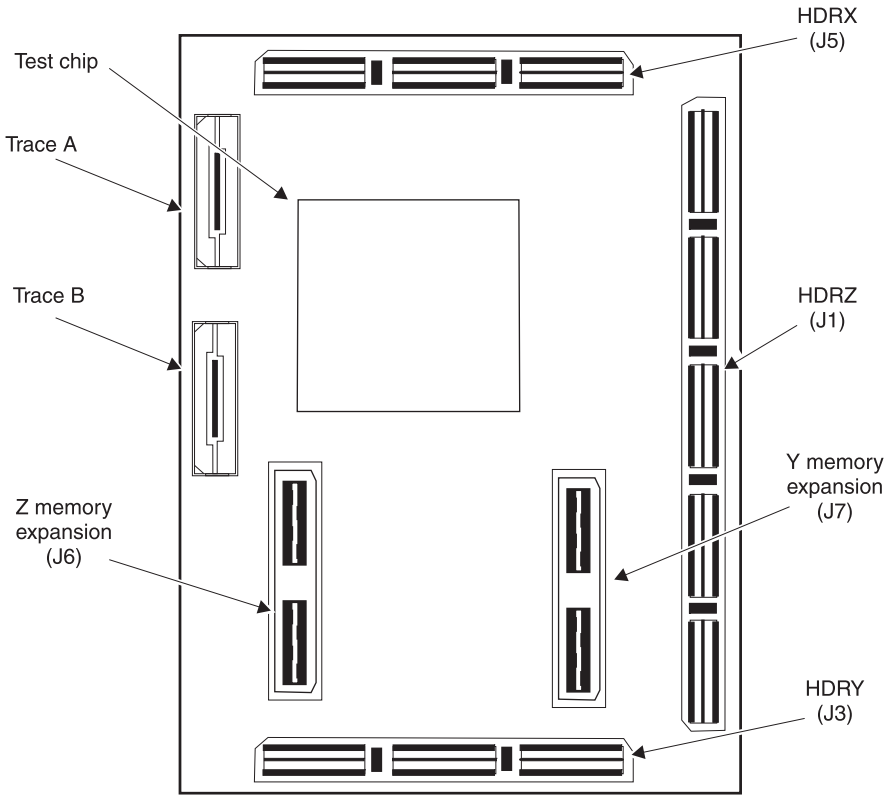

Figure 1-1 Core Tile layout

Figure 1-2 on page 1-5 shows a Core Tile mounted on an Integrator/IM-LT3 Interface Module.

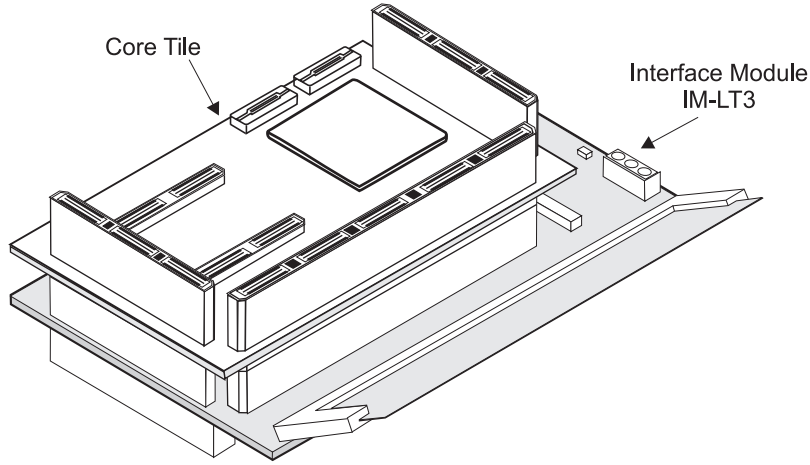

Figure 1-2 Core Tile and Integrator/IM-LT3

1.2.1 System architecture

Figure 1-3 shows the architecture of a minimal Core Tile system consisting of a CT926EJ-S and an IM-LT3 Interface Module.

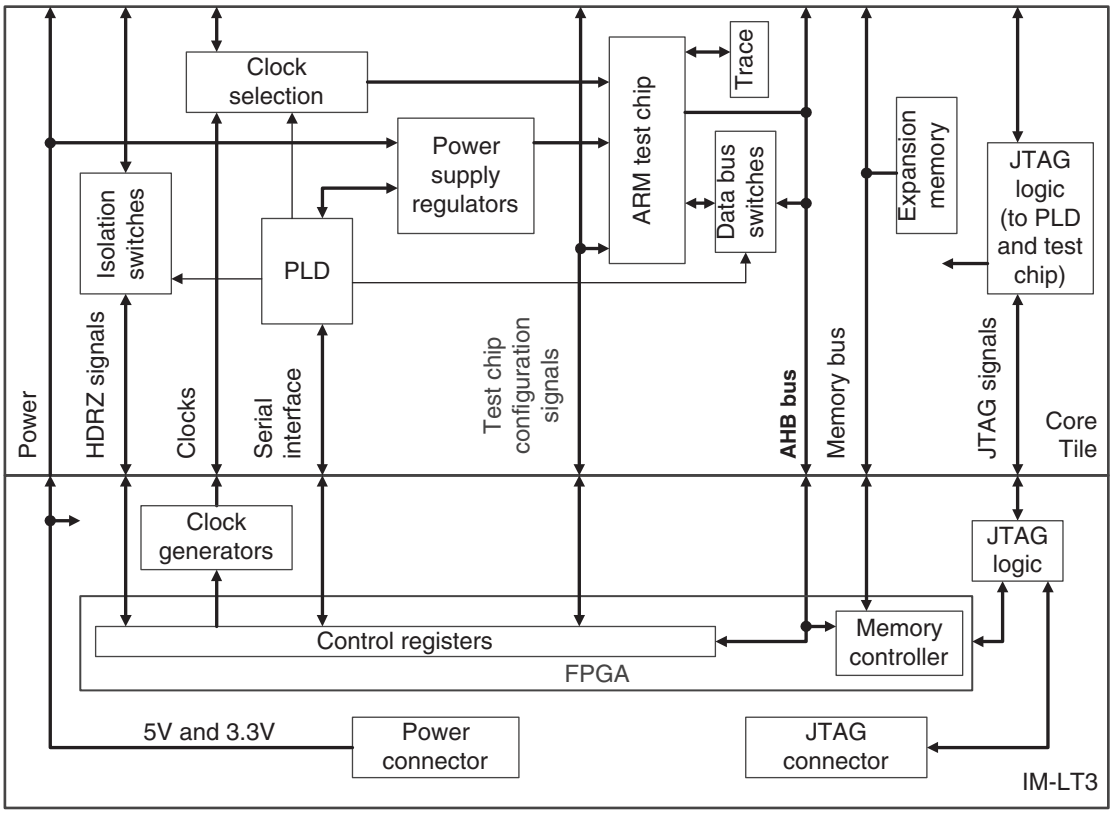

Figure 1-3 Typical system block diagram

1.2.2 External logic

The Core Tile cannot operate standalone. It must have an external board (or boards) that provides:

- power and JTAG connectors
- reference clock
- initialization signals to the clock divider inputs on the test chip

- memory controller for the expansion memory modules
- peripheral devices (for example, interrupt controller, DMA controller, system and reset controller, serial I/O)
- a serial interface to the PLD that loads the desired configuration for HDRZ isolation, clock multiplexors, AHB data bus interconnection, and programmable power supply values.

The external connectors and control logic are typically provided by an Interface Module or Logic Tile. The FPGA on the Logic Tile or Interface Module provides system control functions for the Core Tile. See also Chapter 3 *HBI-0131 Hardware Description*, Chapter 7 *HBI-0141 Hardware Description*, and the documentation for your Versatile products.

Note

The FPGAs on the external baseboard, IM-LT3, or Logic Tile must be loaded with an appropriate image. For more information on FPGA images, see the application note covering the combination of products that you are using.

1.2.3 ARM processor test chip

The name of the Core Tile reflects the test chip fitted, for example the CT926EJ-S has an ARM926EJ-S test chip.

Note

For more details on processors and test chips, see the *Technical Reference Manual* for the processor family. Details on variations between test chips are described in documents in the *product_name*\docs\test chips directory of the CD.

You can configure the processor using several input signals to the test chip (for example, **HCLKDIV**, **PLLCTRL**, and **PLLFBDIV**). In a production ASIC, these core signals are permanently tied HIGH or LOW. However, you can reprogram these inputs in order to experiment with different processor configurations.

1.2.4 PLD

The PLD on the Core Tile receives configuration signals from a serial link (controlled by an attached Logic Tile or IM-LT3 Interface Module) and outputs signals that control:

- the DACs that control the power supplies for the test chip core and I/O voltages
- the clock multiplexors that control the clock sources to the test chip
- the AHB data bus interface

- isolation for some of the signals on HDRZ.

The PLD transmits the following over the serial interface:

- a manufacturing identifier (MANID) that indicates the board version
- a PLL identifier (PLLID) that can be used to identify the version of the PLL image
- the current value of the VDDCORE, VDDPLL, and VDDIO voltages
- the current drawn from the VDDCORE supplies.

Note

The current-sense resistors fitted to the board at manufacture are zero Ohm. If you require monitoring of the test chip currents, you must replace the existing resistors on the board, see *Reading the voltages and currents* on page 3-23.

- Different Core Tiles might also transmit signals that are specific to that tile. For example, the CT926EJ-S transmits a power good signal from the voltage regulators.

1.2.5 Processor bus

The processor bus depends on the type of test chip fitted. For example, the test chip on the CT926EJ-S Core Tile has an AHB bus interface. See *Overview of Core Tile configuration* on page 3-29 for a description of configuration options for the AHB data bus and *Overview of Core Tile configuration* on page 7-31 for a description of configuration options for the ARM7TDMI data bus.

1.2.6 Memory

There are two memory expansion sockets on the Core Tile. These accept static memory boards. There is not a direct connection between the memory modules and the test chip bus. The memory control and data signals are provided by external logic, typically an FPGA on a Logic Tile.

There is typically *Tightly Coupled Memory* (TCM) present in the test chip. The memory present depends on the test chip.

1.2.7 Clock generation

The primary reference clock for the Core Tile is provided by an attached Logic Tile or baseboard. Depending on the version of the Core Tile, the tile has options to generate additional secondary reference clocks. The functionality of the clocking system depends on the particular Core Tile.

1.2.8 JTAG and Trace

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

If the test chip contains an ETM, the Core Tile has one or two trace connectors fitted to allow tracing program flow of the test chip processor. The size and configuration of the Trace Port depends on the Core Tile.

1.2.9 Power supply control

Interface logic on the Core Tile enables you to read different supply voltages. The current drawn by the test can also be read, however this requires the sense resistors on the circuit board to be replaced.

On some tiles, the interface logic also enables you to control the core voltages or shut down the voltage regulators from software. The voltage control range is set when the board is manufactured. Core voltages can be switched off to test power saving modes.

By default, the memory modules and the test chip I/O voltage level is 3.3V. Links on the Core Tile allow a different I/O voltage to be selected.

1.2.10 Links and indicators

The Core Tile has factory-installed links for tile identification code, the size of test chip memory, clock-routing, core power-supply range, and I/O voltage source. These links do not normally require changing. You must change the links if you are using non-standard I/O or memory voltages. For more details, see the section on links in the section that covers your test chip.

The Core Tile has an LED indicator for the power supply. For details on indicators present on the Logic Tiles, Interface Modules, and baseboard products, see the documentation supplied with the product.

1.3 Precautions

This section contains safety information and advice on how to avoid damage to the Core Tile.

1.3.1 Ensuring safety

The Core Tile is powered from 3.3V and 5V DC supplies.

Warning

To avoid a safety hazard, only *Safety Extra Low Voltage* (SELV) equipment must be connected to the tile.

1.3.2 Preventing damage

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

Caution

To avoid damage to the Core Tile you must observe the following precautions.

- Never subject the tile to high electrostatic potentials.
- Always wear a grounding strap when handling the tile.
- Only hold the tile by the edges.
- Avoid touching the component pins or any other metallic element.

Do not use the board near equipment that could be:

- sensitive to electromagnetic emissions, such as medical equipment
- a transmitter of electromagnetic emissions.

Chapter 2
Getting Started

This chapter describes how to set up and prepare a Core Tile for use. It contains the following sections:

Note

The Core Tile must be used with an external board that provides the configuration control and additional peripherals. The FPGA on the external board must contain an image that is explicitly designed to interface with the Core Tile.

The information in this chapter provides only a general overview. Details on using a Core Tile with different ARM products are covered in application notes. Ensure that you use the correct application note and FPGA image for your configuration.

2.1 Using the Core Tile with an Interface Module

To set up a Core Tile and IM-LT3 Interface Module as a standalone development system:

1. Mount the Core Tile on an IM-LT3 Interface Module.

   Note

   You can use the IM-LT1 Interface Module instead of the IM-LT3. The IM-LT1, however, does not contain an FPGA so you must mount a Logic Tile between the Core Tile and the IM-LT1.

2. You can fit one or two memory expansion boards to the Core Tile if you require additional memory.
3. If you are constructing a multiprocessor system, add additional pairs of Core Tile and Logic Tile.
4. Connect a JTAG debugger to the Interface Module (see *Connecting a JTAG device to an Interface Module* on page 2-13).
5. If required, you can connect an Analyzer Tile between the Logic Tile and the Core Tile to enable monitoring of signals between the tiles.
6. You can also place a Logic Tile and an Interface Tile on the top of the tile stack. The Logic Tile must be loaded with an appropriate image that contains your peripherals.

   The connectors on the Interface Tile provide access to the peripherals instantiated in the Logic Tile.
7. Connect the CONFIG link on the Interface Module.
8. Supply power to the Interface Module, see *Supplying power to an Interface Module* on page 2-9.
9. Load the appropriate FPGA image into the IM-LT3 Interface Module, or Logic Tile if using the IM-LT1. See the application note for details on the image to use.
10. Remove the CONFIG link and use a JTAG debugger to load your application program.

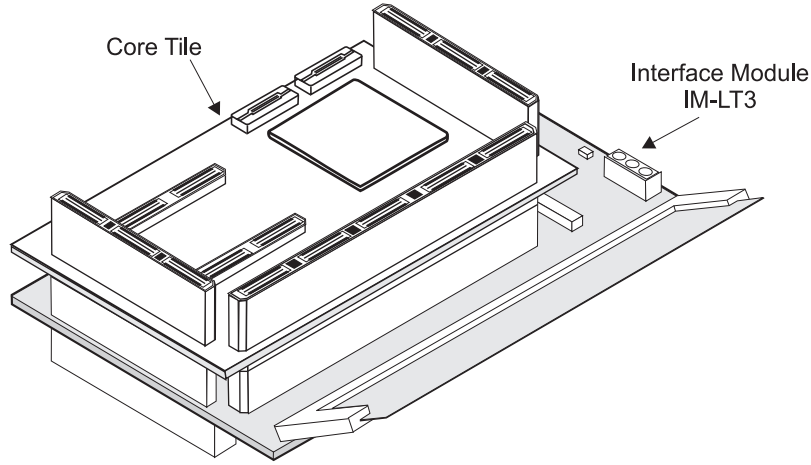

Figure 2-1 Core Tile and an Integrator/IM-LT3

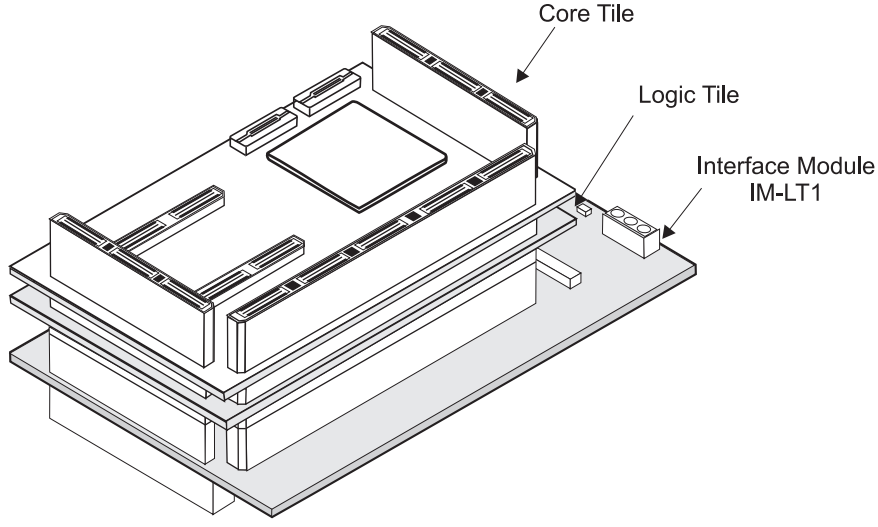

Figure 2-2 Core Tile, Logic Tile, and Integrator/IM-LT1

2.2 Using the Core Tile with an Integrator Compact Platform baseboard

To add a Core Tile to an Integrator/CP development system:

1. Mount the Core Tile on an IM-LT3 Interface Module.

   Note

   You can use the IM-LT1 Interface Module instead of the IM-LT3. The IM-LT1, however, does not contain an FPGA so you must mount a Logic Tile between the Core Tile and the IM-LT1. ARM Ltd. does not provide an application note or FPGA image for this configuration and you must therefore create a custom FPGA image for the IM-LT1 system.

2. You can fit one or two memory expansion boards to the Core Tile if you require additional memory.
3. If you are constructing a multiprocessor system, add additional pairs of Core Tile and Logic Tile.

   Note

   The Integrator/CP does not support multiple masters on the system bus.

4. If required, you can connect an Analyzer Tile between the Logic Tile and the Core Tile to enable monitoring of signals between the tiles.
5. You can also place a Logic Tile and an Interface Tile on the top of the tile stack. The Logic Tile must be loaded with an appropriate image that contains your peripherals.

   The connectors on the Interface Tile provide access to the peripherals instantiated in the Logic Tile.
6. Connect the Interface Module and its tiles to the Integrator/CP.
7. Connect a JTAG device to the Interface Module.
8. Connect the CONFIG link on the Interface Module.
9. Supply power to the Compact Platform baseboard, see *Supplying power to an Integrator/CP* on page 2-11.
10. Load the appropriate FPGA images into the Interface Module (and Logic Tiles if used) and baseboard. See the application note for details on the image to use.
11. Remove the CONFIG link and load your application program.

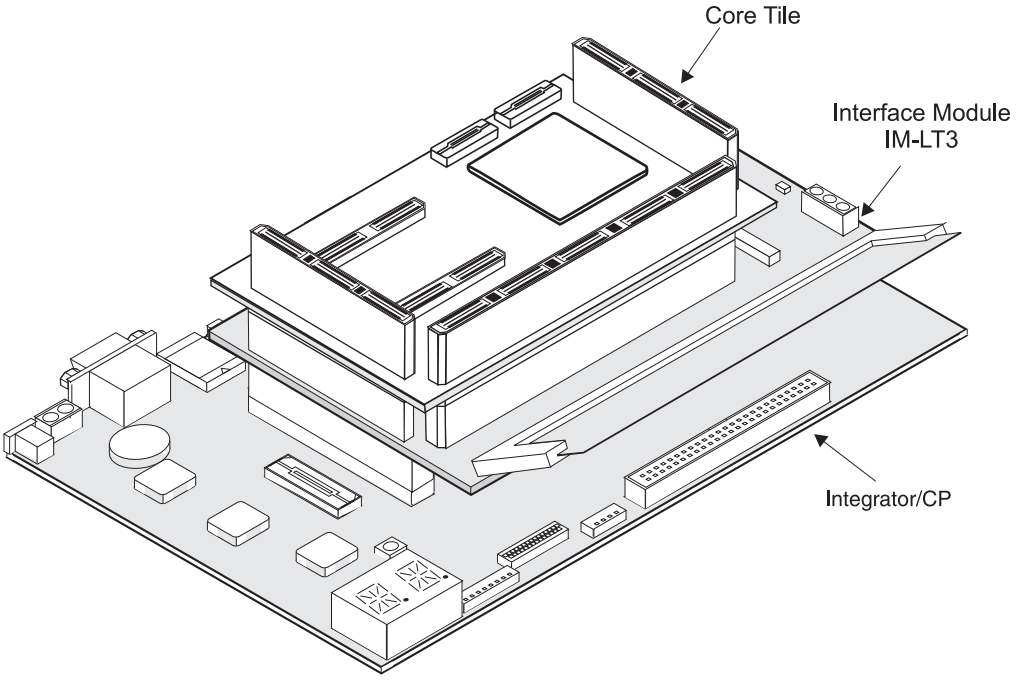

Figure 2-3 Core Tile, Integrator/IM-LT3, and Integrator/CP

Note

For details on how to load and run applications on an Integrator/CP system, see the *ARM Integrator/CP User Guide* and the *ARM Firmware Suite Reference Manual*.

2.3 Using the Core Tile with a Versatile Baseboard

To set up one or more Core Tiles on a Versatile platform baseboard (such as the Versatile/PB926EJ-S shown in Figure 2-4 on page 2-7) as a multi-processor development system:

1. You can fit memory expansion boards to the Core Tile or baseboard if you require additional memory.
2. Fit one or more sets of Logic Tile and Core Tile to the tile expansion headers on the baseboard.

   The Logic Tile provides the interface and arbitration between the baseboard signals and the processor bus on the Core Tile.

   Note

   For normal operation, a Logic Tile between the Core Tile and the Versatile/PB926EJ-S baseboard provides the interface control. For programming the PLD on the Core Tile however, the Logic Tile must be mounted on top of the Core Tile for the system to configure correctly. This is because the **RTCK** signal must be floating in configuration mode.
3. If required, you can connect an Analyzer Tile between the Logic Tile and the Core Tile to enable monitoring of signals between the tiles.
4. You can also place a Logic Tile and an Interface Tile on the top of the tile stack. The Logic Tile must be loaded with an appropriate image that contains your peripherals.

   The connectors on the Interface Tile provide access to the peripherals instantiated in the Logic Tile.
5. Connect a JTAG debugger to the baseboard (see *Connecting a JTAG device to a baseboard* on page 2-14).
6. Connect the CONFIG link on the baseboard.
7. Supply power to the Versatile/PB926EJ-S (see *Supplying power to a Versatile/PB926EJ-S* on page 2-10).
8. Load the appropriate FPGA images into the Logic Tiles. See the application note for details on the image to use. See the *Versatile/PB926EJ-S User Guide* and *Versatile/LT-XC2V4000+ User Guide* for details on programming procedures.
9. Remove the CONFIG link and load your application program.

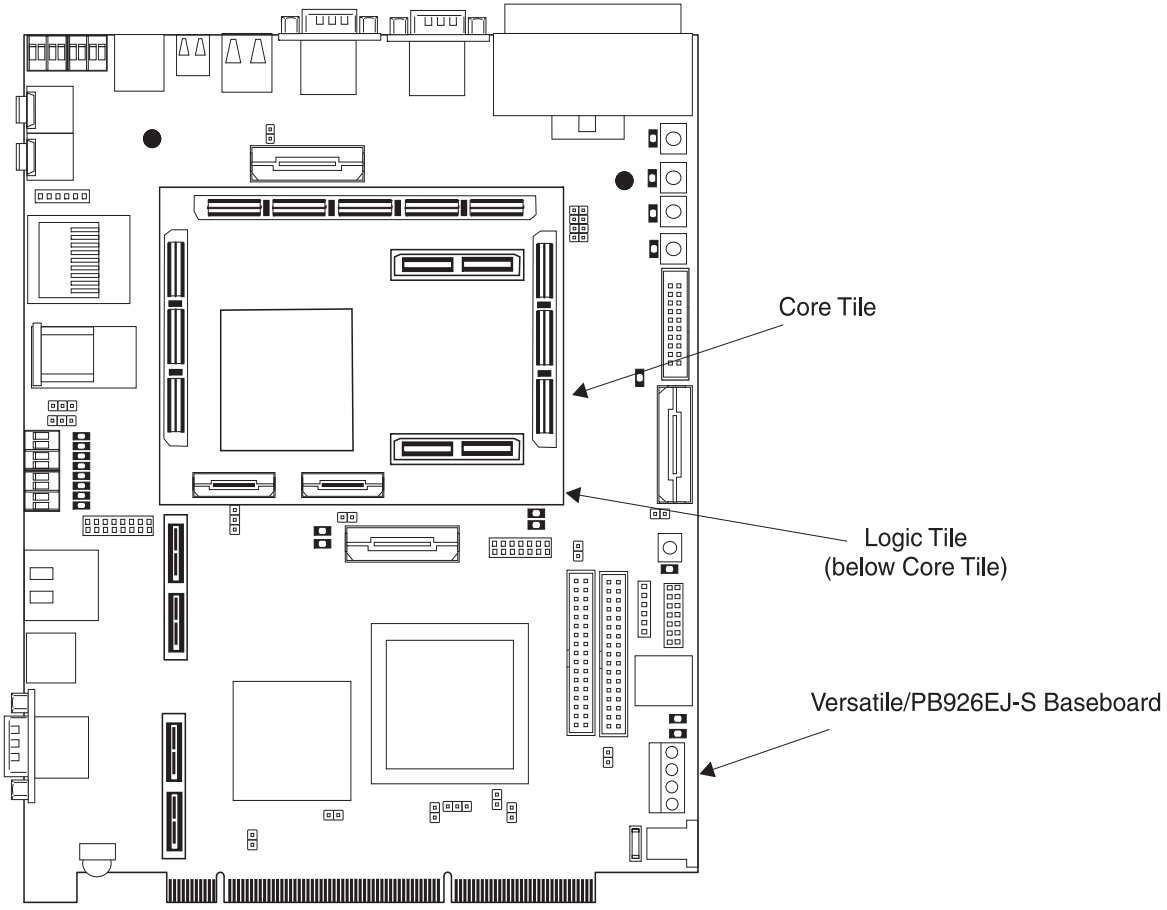

Figure 2-4 Core Tile and a Versatile baseboard

Note

For details on how to load and run applications on an Versatile/PB926EJ-S system, see the *Versatile Platform Baseboard for ARM926EJ-S User Guide*.

2.4 Using the Core Tile with a custom baseboard

If you are designing a custom baseboard to accept a Core Tile, you must ensure that your board meets the following requirements:

Mechanical layout

The mechanical layout is described in Appendix B *Specifications*.

Power supplies

The Core Tile uses a 3.3V and 5V supplies to generate local voltages. The test chip core and I/O voltage are controlled by logic in the PLD on the Core Tile. There are resistor links that control the source of the VDDIO voltage (present on HDRX) and the VCCOY voltage (present on HDRY). See the hardware chapters that cover specific Core Tiles for details on the voltage control logic.

Clock control

The primary reference clock must be supplied by an attached Logic Tile or baseboard. The clocking system is described in the hardware chapters that cover specific Core Tiles. There might also be test-chip specific clocking requirements. See the chapters in this manual covering your test chip and the *Technical Reference Manual* for your core.

JTAG control

The Core Tile does not have a JTAG connector. The baseboard must provide a JTAG connector and route the JTAG signals to the header signals on the Core Tile. JTAG routing is described in *Overview of Core Tile configuration* on page 3-29 and *JTAG support* on page 7-40.

Bus configuration signals

The processor bus and some of the signals on header HDRZ are controlled through bus switches. Bus configuration is managed by the PLD on the Core Tile, see *Overview of Core Tile configuration* on page 3-29 or *Overview of Core Tile configuration* on page 7-31.

Memory controller

The Core Tile has two PISMO memory connectors. The memory boards are controlled from external logic (typically from an FPGA on a Logic Tile). If you are using additional memory on the Core Tile, you must implement your own memory controller on your baseboard. The memory interface signals and memory specification is described in *Memory expansion connector pinout* on page 6-20 and Appendix A *Static Memory Expansion Board*.

2.5 Connecting power

Core Tile power is supplied through the header connectors. You must connect a power source to either an external Interface Module or baseboard.

Caution

You can use only one power source for the system. For example, do not connect power supplies to both the IM-LT3 and the Integrator/CP.

For baseboards that have two different power connectors, use only one of the connectors for the power supply.

2.5.1 Supplying power to an Interface Module

When using the Core Tile with an IM-LT3 Interface Module (or with an IM-LT1 Interface Module and a Logic Tile), you must connect a bench power supply with 3.3V and 5V outputs to the power connector on the Interface Module, as illustrated in Figure 2-5.

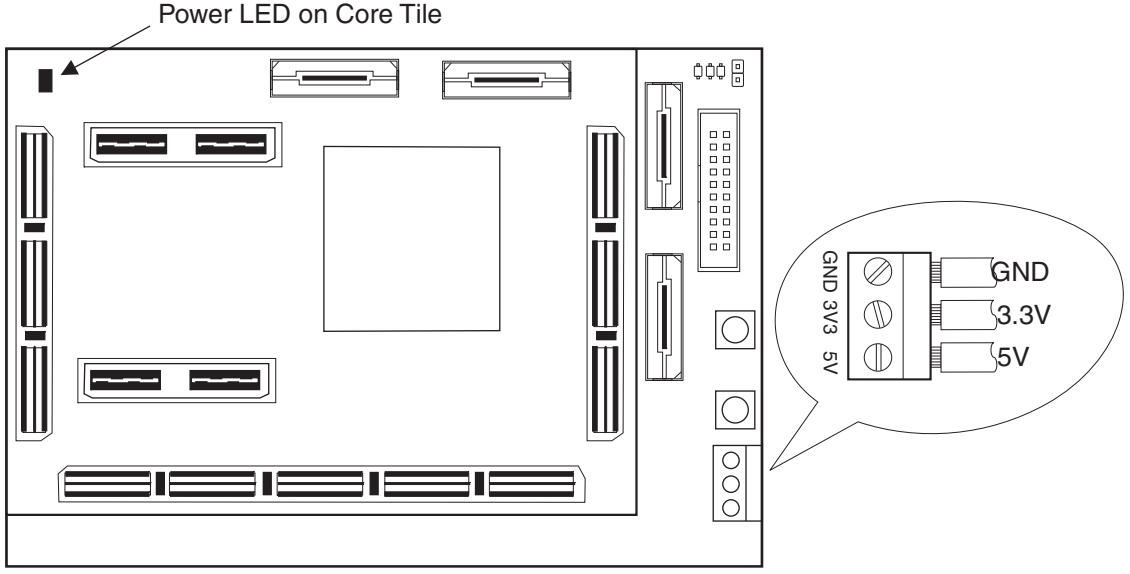

Figure 2-5 Power connector on interface Module

2.5.2 Supplying power to a baseboard

If you are using the Core Tile on a baseboard development system, connect power to the baseboard. For more details, see the documentation supplied with your baseboard.

Supplying power to a Versatile/PB926EJ-S

Connect either the supplied brick power supply to power socket J35 or an external bench power supply to the screw-terminal connector as shown in Figure 2-6.

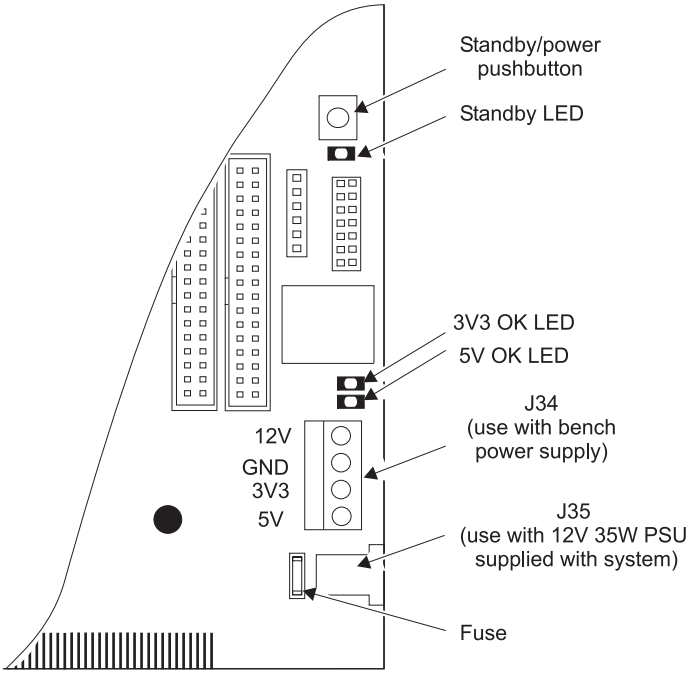

Figure 2-6 Power connectors on Versatile/PB926EJ-S

Note

If you are using the supplied brick power supply connected to J35 on the Versatile/PB926EJ-S, the Standby/power pushbutton toggles the power on and off. If you are using a different power source, the Standby/power switch is not used.

Caution

You can use only one power source for the system.

Supplying power to an Integrator/CP

Connect either the supplied brick power supply to power socket J35 or an external bench power supply to the screw-terminal connector as shown in Figure 2-7.

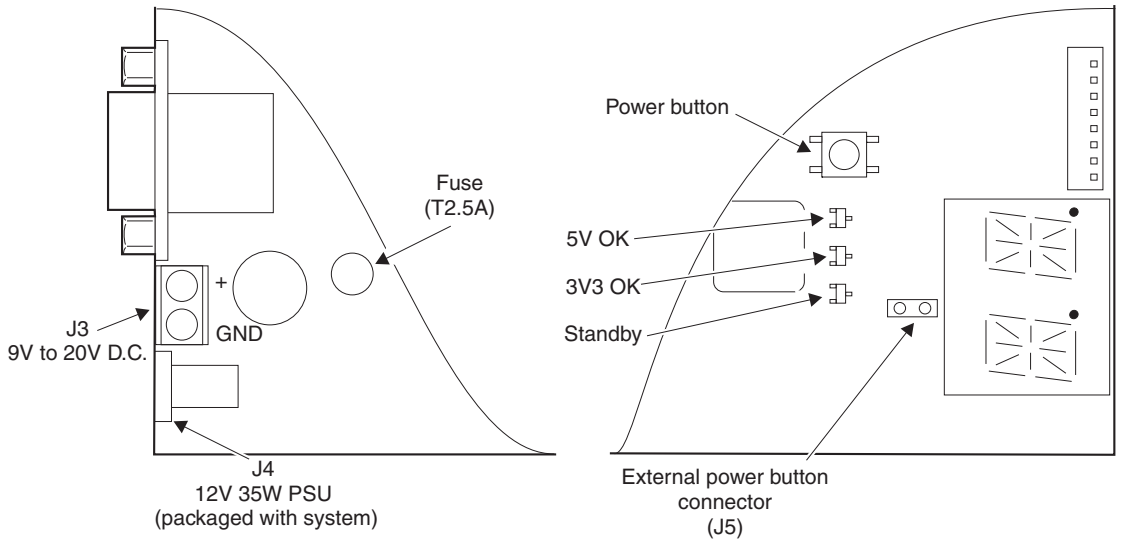

Figure 2-7 Powering the assembled Integrator/CP system

Onboard voltage regulators provide 3.3V and 5V.

When you apply power to J3 or J4, the standby LED illuminates. To power the system up, press the power button.

Caution

You can use only one power source for the system.

2.6 Connecting RealView ICE or Multi-ICE

The Core Tile does not have a JTAG connector, but there are JTAG signals present on the header connectors. The Core Tile must be used with an Interface Module or baseboard that contains a JTAG connector. External JTAG equipment can be used to:

- download and debug programs.

  You can connect RealView ICE or Multi-ICE (or other JTAG debuggers) to the external JTAG connector and download and debug programs. (For a Core Tile attached to a Versatile/PB926EJ-S, you can also use the USB debug port on the Versatile/PB926EJ-S.)

- download new images to the PLD on the Core Tile or to FPGAs or PLDs present on attached boards, for example, Logic Tiles.

  Note

  For a Core Tile attached to an Integrator/CP, currently only Multi-ICE is supported for downloading images to FPGAs on attached boards.

  For a Core Tile attached to a Versatile/PB926EJ-S, either Multi-ICE or the USB debug port on the Versatile/PB926EJ-S can be used for downloading FPGA images.

Selection between debugging programs and downloading new images to the FPGA is controlled by the CONFIG link that is present on the Interface Module or baseboard. See the documentation supplied with your Interface Module or baseboard for more details on connecting JTAG and connecting the CONFIG link.

Caution

Because the Core Tile does not provide nonvolatile memory, programs are lost when the power is removed. Use flash memory for nonvolatile storage. The flash memory can be:

- Any unused space in the Logic Tile flash. The Logic Tile flash is primarily used for configuration and must contain a valid configuration image for the FPGA.
- The flash memory on the baseboard.
- Nonvolatile memory in one of the two PISMO memory expansion slots present on the Core Tile, see Appendix A *Static Memory Expansion Board*. A memory controller must be implemented in an external Logic Tile to access the Core Tile PISMO expansion memory.

The JTAG connector provides a set of JTAG signals that allow JTAG debugging equipment to be used. If you are debugging a development system with multiple tiles, connect the JTAG debugging equipment to the Interface Module or baseboard and the JTAG signals will be routed through any connected tiles.

Note

Do not use the JTAG connector on the Trace Port Adaptor board. The adapter board must only be used for Trace. Use the JTAG connector on the baseboard or interface module.

The JTAG paths are described in *JTAG support* on page 3-39 and *JTAG support* on page 7-40.

2.6.1 Connecting a JTAG device to an Interface Module

The JTAG setup for an Interface Module is shown in Figure 2-8.

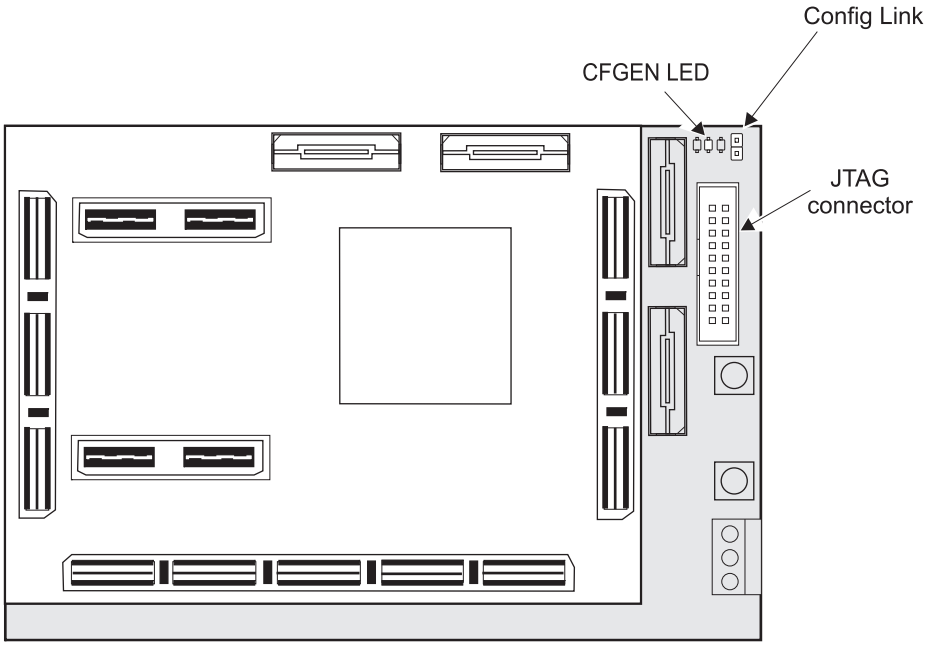

Figure 2-8 JTAG connector on the Interface Module

2.6.2 Connecting a JTAG device to a baseboard

The JTAG setup for the Versatile/PB926EJ-S and Multi-ICE or RealView ICE is shown in Figure 2-9. (Refer to the documentation supplied with your debugger for information on connecting other JTAG interface products.)

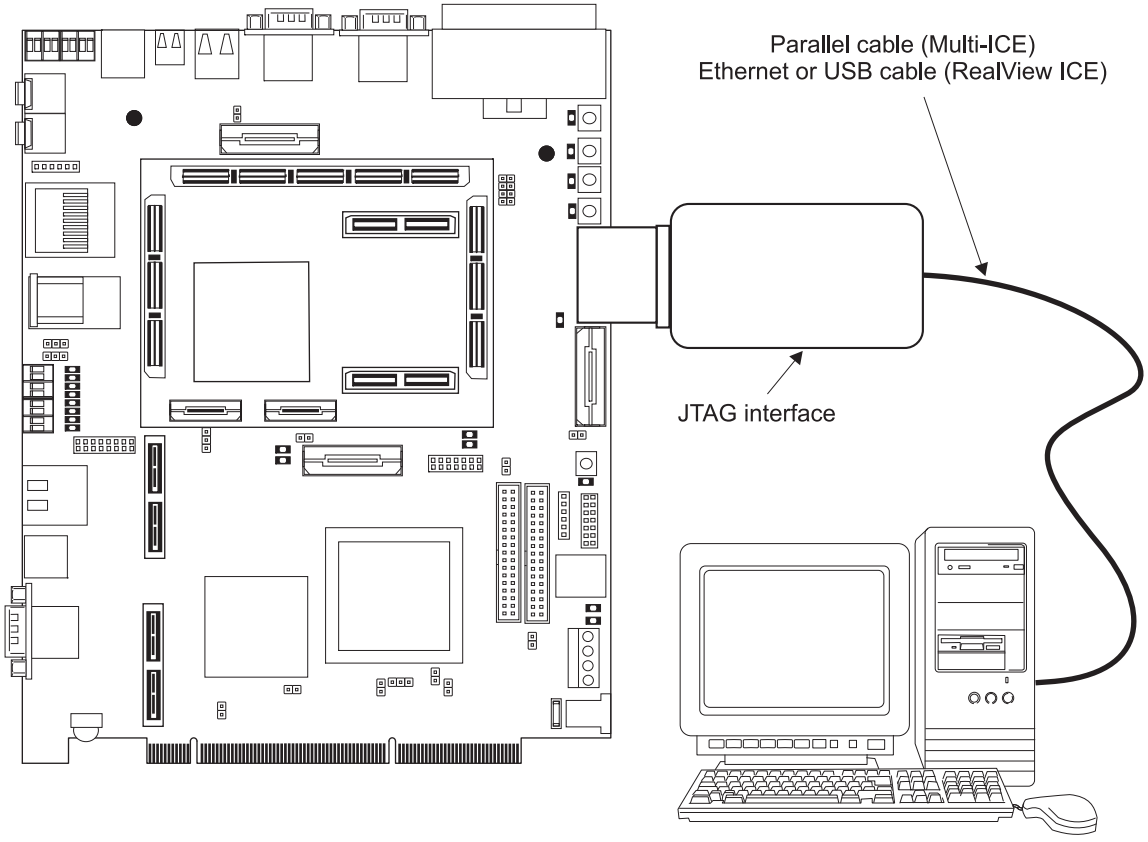

Figure 2-9 JTAG connection to a Versatile/PB926EJ-S

Note

The Versatile/PB926EJ-S has a USB debug port that can be used in place of the JTAG interface and JTAG connector.

2.7 Connecting Trace

If the Core Tile test chip contains an *Embedded Trace Macrocell* (ETM), a Trace connector is provided on the Core Tile. Use the JTAG connector on the Interface Module or baseboard to provide the JTAG signals that are required for controlling the ETM.

Note

Trace tools using multiplexed trace packets (such as RealView Trace and Multi-Trace) require only one Mictor connector (Trace Port A). The Core Tile supports both multiplexed (one trace connector) and demultiplexed mode (two trace connectors). The second connector is present for support trace tools that use demultiplexed trace packets.

Figure 2-10 illustrates a trace debugging setup with RealView Trace, RealView ICE, and the IM-LT3. Figure 2-11 on page 2-16 illustrates a trace debugging setup with Multi-Trace, Multi-ICE, and the IM-LT3.

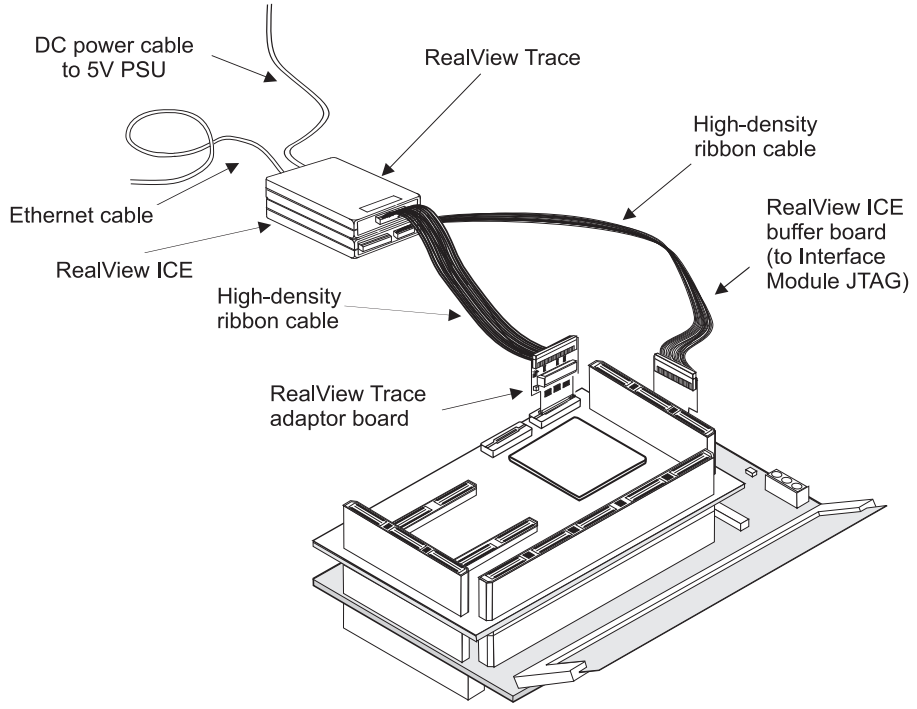

Figure 2-10 Trace connection with RealView Trace

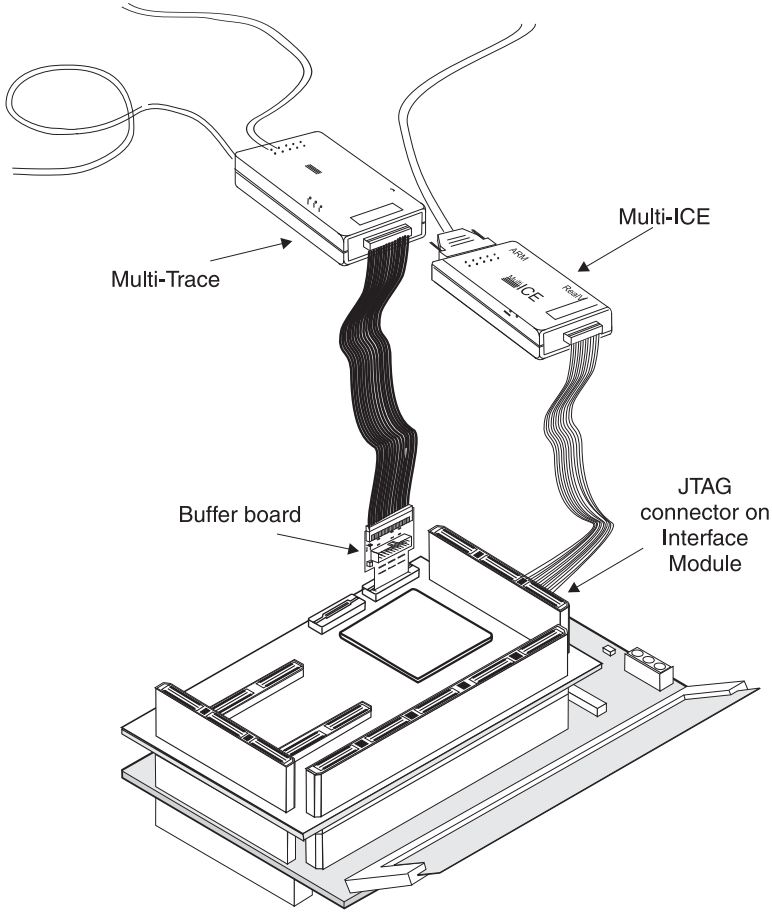

Figure 2-11 Trace connection with Multi-Trace

Note

For more details on using Trace, see the documentation supplied with your Trace hardware and with your Interface Module or baseboard product.

The routing of the JTAG scan chain is described in *JTAG signals* on page 3-39 and *JTAG signals* on page 7-40.

Part B

Printed Circuit Board HBI-0131 (CT926EJ-S and CT1136JF-S)

Chapter 3
HBI-0131 Hardware Description

This chapter describes the on-board hardware in the CT926EJ-S and CT1136JF-S Core Tiles that use the HBI-0131 printed circuit board (companion Logic Tile or Interface Module are also described where relevant). It contains the following sections:

Note

The HBI-0131 board supports both the CT926EJ-S and CT1136JF-S processors, but the availability of boards depends on the availability of test chips. Contact your sales representatives for details on currently available Core Tiles.

Note

This chapter describes the generic hardware and is independent of the FPGA image used in an external Logic Tile, Interface Module, or baseboard.

The diagrams in this chapter (such as the block diagram shown in *Core Tile block diagram* on page 3-3 for example) are typical of Core Tiles, but your Core Tile architecture might be different depending on the test chip used.

The specific clock interconnections, voltage setting resistors and link, and bus interfaces might be different for different test chips. Different test chips might also use the configuration signals in different ways.

3.1 Core Tile architecture

The Core Tile supports test chips packaged according to the *Generic Test Chip* (GTC) specification. Configuration and interface logic on the Core Tile is connected to the upper and lower header connectors as shown in Figure 3-1.

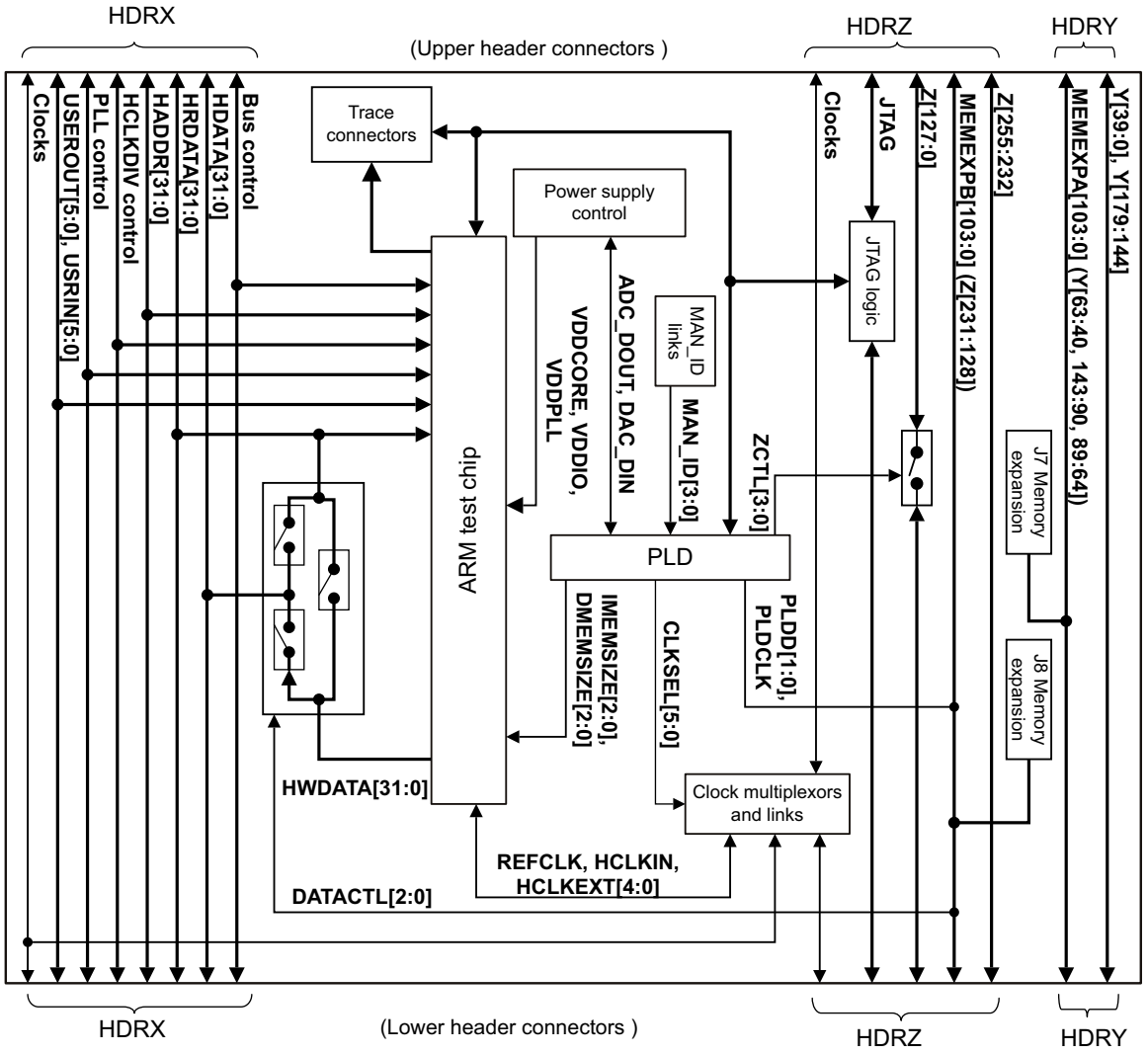

Figure 3-1 Core Tile block diagram

3.2 ARM microprocessor test chip

Table 3-1 provides a brief overview of the signals present on test chips that follow the *Generic Test Chip* (GTC) specification.

Note

Test chip configuration is covered in:

- *Memory located inside test chip* on page 3-8
- *Clocks* on page 3-10
- *Power supply control* on page 3-17
- *Data read/write control for AHB data* on page 3-26.

Table 3-1 Test chip signals

Group	Signal	Description	Direction
AHB	**HADDR[31:0]**	AHB address bus	Output
AHB	**HBURST[2:0]**	AHB burst length indication	Output
AHB	**HBUSREQ**	AHB bus request	Output
AHB	**HGRANT**	AHB bus grant	Input
AHB	**HLOCK**	AHB locked transfer indication	Output
AHB	**HPROT[3:0]**	AHB protection indication	Output
AHB	**HREADY**	Transfer ready indication	Input
AHB	**HRESP[1:0]**	AHB transfer response	Input
AHB	**HSIZE[2:0]**	AHB transfer length indication	Output
AHB	**HTRANS[1:0]**	AHB transfer type	Output
AHB	**HWRITE**	AHB data direction	Output
AHB/Test	**HRDATA[31:0]**	AHB data bus in	Input
AHB/Test	**HWDATA[31:0]**	AHB data bus out	Output
Clock	**CLK**	Clock from PLL bypass multiplexer (this signal is internal to the test chip and not brought out to the pads)	-

Table 3-1 Test chip signals (continued)

Group	Signal	Description	Direction
Clock	**HCLK**	Local memory clock (this signal is internal to the test chip but some test chips output buffered versions of **HCLK** as **HCLKEXT[4:0]**)	-
Clock	**HCLKDIV[2:0]**	Ratio of the AHB clock to **CLK**.**HCLKDIV[1:0]** is used as SnA and FASTBUS respectively in cores requiring these signals	Input
Clock	**HCLKEXT[4:0]**	Some test chips output buffered versions of HCLK as **HCLKEXT[4:0]**	Output
Clock	**HCLKIN**	Input clock from multiplexors. Some test chips use this clock as the main reference clock for the bus interface clock **HCLK**.	Input
Clock	**PLLCTRL[1:0]**	PLL Control	Input
Clock	**PLLLOCK**	In-lock indication from PLL	Output
Clock/Test	**PLLFBDIV[7:0]**	PLL control	Input
Clock/Test	**PLLOUTDIV[3:0]**	PLL control	Input
Clock	**REFCLK**	Input clock from multiplexors. Some test chips use this clock as the main reference clock and use the divider settings to produce the internal CPU clock **CLK**.	Input
Debug	**COMMRX**	Communications receive buffer not empty	Output
Debug	**COMMTX**	Communications transmit buffer empty	Output
Debug	**DBGACK**	Debug acknowledge	Output
Debug	**DBGEN**	Debug enable - synchronous	Input
Debug	**DEWPT**	External watchpoint - synchronous	Input
Debug	**EDBGRQ**	External debug request - synchronous	Input
Debug	**IEBKPT**	External breakpoint - synchronous	Input
Interrupt	**nFIQ**	External fast interrupt - asynchronous	Input
Interrupt	**nIRQ**	External interrupt - asynchronous	Input
JTAG	**nTDOEN**	Enable TDO tristate buffer	Output
JTAG	**nTRST**	Test reset to core, TAP and ICE	Input

Table 3-1 Test chip signals (continued)

Group	Signal	Description	Direction
JTAG	**RTCK**	Indicates sync of Multi-ICE TCK	Output
JTAG	**TCK**	Test clock	Input
JTAG	**TDI**	Boundary scan Input	Input
JTAG	**TDO**	Boundary scan Output	Output
JTAG	**TMS**	Test mode select	Input
Misc	**BIGENDIN**	Selects big endian	Input
Misc	**BIGENDOUT**	Indicates big-endian	Output
Misc	**DMEMSIZE[2:0]**	Size of data memory/cache	Input
Misc	**IMEMSIZE[2:0]**	Size of instruction memory/cache	Input
Misc	**INITRAM**	Internal RAM enabled at reset	Input
Misc	**USERIN[5:0]**	User inputs	Input
Misc	**USEROUT[5:0]**	User Outputs	Output
Misc	**VINITHI**	Location of exception vectors at reset	Input
Mux	**CONFIGINIT**	Control of test chip configuration	Input
Mux	**nCONFIGRST**	Reset of test chip configuration	Input
Power	**ARM_VDDIO**	Voltage level used for input and output signals	Input
Power	**ARM_VDDCORE[6:1]**	Voltage level used for processor core	Input
Power	**ARM_VDDPLL[2:1]**	Voltage level used for Phase-locked loop in the clock generation circuit	Input
Reset	**nPORESET**	Power-on reset to clock and reset control logic - asynchronous	Input
Reset	**nRESET**	Reset input - asynchronous	Input
Test	**TESTSELECT**	Test mode select	Input
Test	**TICSELECT**	Test mode select	Input
Trace	**ETMEXTIN**	ETM test control input - synchronous	Input
Trace	**ETMEXTOUT**	ETM test control output	Output

Table 3-1 Test chip signals (continued)

Group	Signal	Description	Direction
Trace	**PIPESTATA[2:0]**	Pipeline status A	Output
Trace	**PIPESTATB[2:0]**	Pipeline status B	Output
Trace	**TRACECLK**	ETM clock output	Output
Trace	**TRACEPKTA[15:0]**	ETM output to analyzer A	Output
Trace	**TRACEPKTB[15:0]**	ETM output to analyzer B	Output
Trace	**TRACESYNCA**	ETM clock and main clock in sync A	Output
Trace	**TRACESYNCB**	ETM clock and main clock in sync B	Output

3.3 Core Tile memory

This section describes the memory and memory expansion connectors present on the Core Tile.

3.3.1 Memory located inside test chip

Many ARM test chips contain internal *Tightly Coupled Memory* (TCM) or configuration registers. Refer to the processor-specific chapters in this book for details on memory internal to specific test chips. If the size of the test chip internal data and instruction memory are configurable, they can be set by the **DMEMSIZE[2:0]** and **IMEMSIZE[2:0]** signals from the PLD, see Table 3-2. The PLD configures the memory size to maximum at power-on.

Table 3-2 Memory size configuration

IMEMSIZE[2:0] and DMEMSIZE[2:0]			
2	**1**	**0**	**Memory size**
LOW	LOW	LOW	0KB
LOW	LOW	HIGH	4KB
LOW	HIGH	LOW	8KB
LOW	HIGH	HIGH	16KB
HIGH	LOW	LOW	32KB
HIGH	LOW	HIGH	64KB
HIGH	HIGH	LOW	Reserved
HIGH	HIGH	HIGH	Reserved

Note

The printed circuit board has links (LK1 to LK6) in parallel with the **DMEMSIZE[2:0]** and **IMEMSIZE[2:0]** signals from the PLD. These links are present for systems that do not use serial configuration of the PLD. The use of these links is not recommended and memory size configuration should be done with the PLD.

See *Core Tile PLD signals* on page 3-30 for details of the PLD interface. The PLD is configured from an FPGA in an attached Logic Tile or Interface Module. The FPGA image in the FPGA must match the configuration of boards in the stack.

3.3.2 Memory expansion boards

The Core Tile contains two memory connectors for expansion memory boards:

J7 Y memory expansion connector (Y MEMEXP).

J8 Z memory expansion connector (Z MEMEXP).

Note

Typically the expansion boards contain static RAM. The memory connectors conform to the PISMO standard. Refer to the *PISMO Memory Interface Connector Specification* and Appendix A *Static Memory Expansion Board* for details on signals on the connectors and signals. Different memory boards might require special interface software or configuration settings.

The memory controller for expansion memory must be implemented in an external Logic Tile (or IM-LT3 Interface Module). The memory expansion signals are connected directly to HDRY and HDRZ. The Core Tile test chip pins are not connected to the memory expansion sockets.

To use the SSTL standard for a PISMO board on the Y MEMEXP connector, fit resistor links R26, R27, and R28.

3.3.3 Memory map

All bus accesses that do not access the RAM or memory-mapped peripherals inside the test chip are presented on the external 32-bit wide AHB interface consisting of **HADDR[31:0]**, **HRDATA[31:0]**, **HWDATA[31:0]** and the bus control signals.

The location of boot memory is determined by the state of the **VINITHI** signal at reset.

The byte order of memory can be selected by CP15 or by the **BIGENDIN** signal. The byte order in use is indicated by the **BIGENDOUT** signal.

3.4 Clocks

The Core Tile uses reference clocks provided by an attached Logic Tile (or Interface Module) as shown in Figure 3-2.

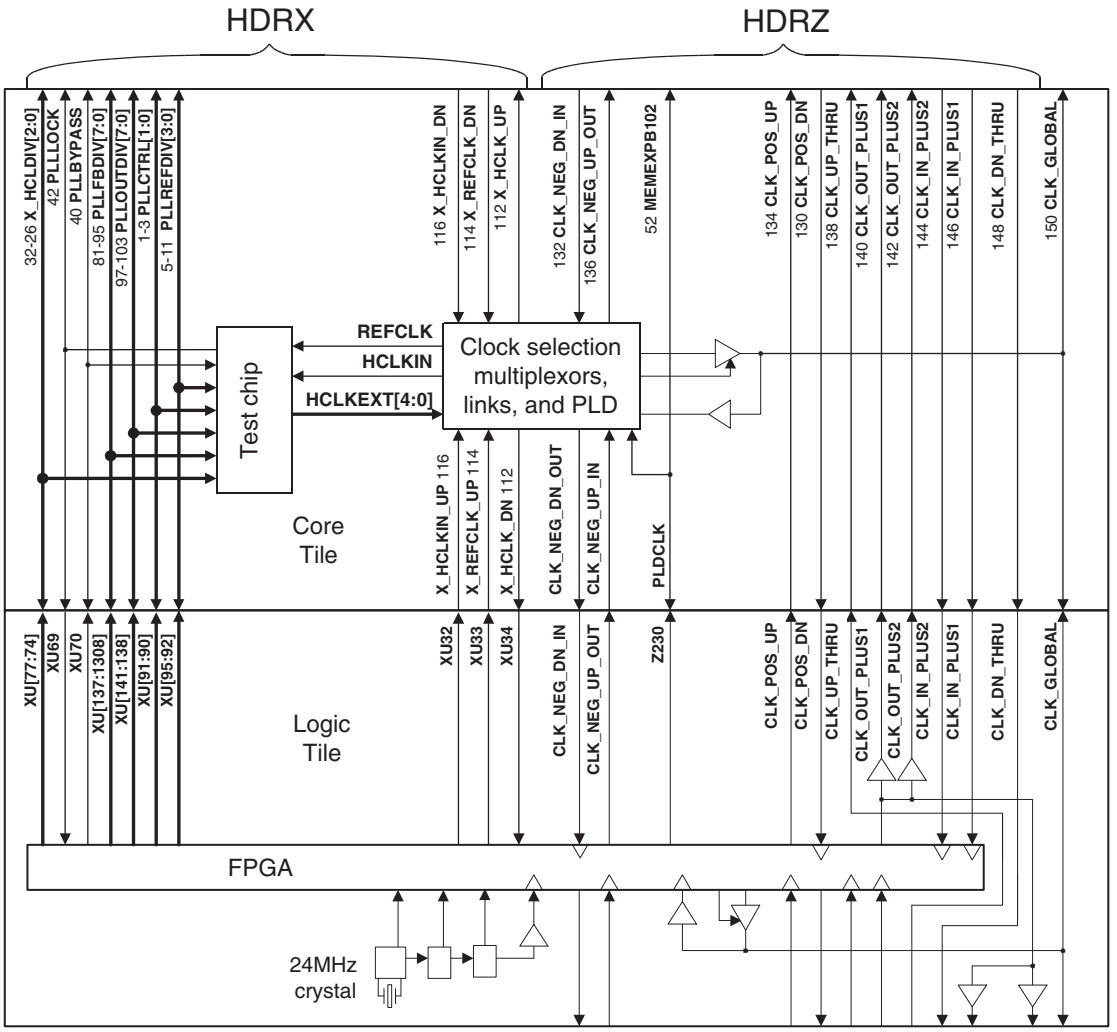

Figure 3-2 Core Tile clock signals

Selection of clocks for the Core Tile is done by setting multiplexors as shown in Figure 3-3 on page 3-12. Figure 3-3 on page 3-12 shows a Core Tile with an ARM926EJ-S test chip. The test chip clock logic and clock control might be different for other test chips.

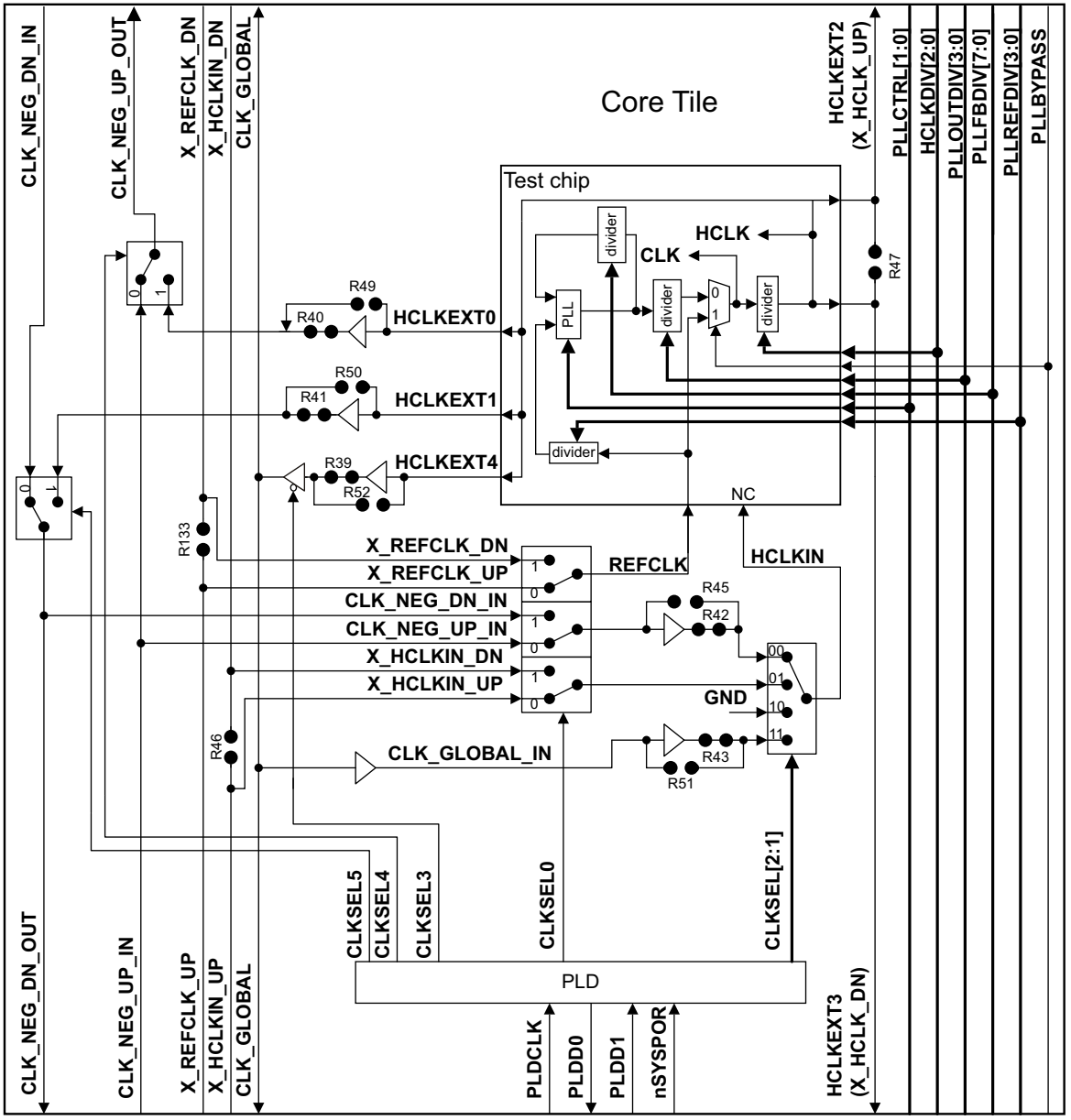

Figure 3-3 Test chip clock selection (CT926EJ-S)

3.4.1 Clock multiplexors

There is not a clock generator present on the Core Tile. The internal **HCLK** (if present) and **CLK** signals in the test chip are derived from an external reference clock to the test chip (**REFCLK**), an internal PLL, and programmable dividers. See *Configuration signals on HDRX* on page 3-37 for details on signals that control the internal PLL and dividers.

Clock selection signals **CLKSEL[5:0]** control the multiplexors on the Core Tile:

CLKSEL[0] This selects between clocks on the upper and lower header connectors.

For **REFCLK**, if **CLKSEL[0]** is HIGH **X_REFCLK_DN** is the source and **X_REFCLK_UP** is the source if **CLKSEL[0]** is LOW.

For **HCLKIN**, if **CLKSEL[0]** is HIGH, **CLK_NEG_DN_IN** and **X_HCLKIN_DN** from the upper header are passed to the next multiplexor. If **CLKSEL[0]** is LOW, **CLK_NEG_UP_IN** and **X_HCLKIN_UP** from the lower header are passed to the multiplexor.

CLKSEL[2:1]

This selects the source for **HCLKIN** from **CLK_GLOBAL**, ground (no clock), or one of **CLK_NEG_DN_IN**, **CLK_NEG_UP_IN**, **X_HCLKIN_UP**, or **X_HCLKIN_DN**.

The CT926EJ-S Core Tile does not use the **HCLKIN** signal.

CLKSEL[3] If LOW, the Core Tile drives the local **HCLKEXT4** signal from the test chip onto the **CLK_GLOBAL** line. Each board in the tile stack can accept or generate **CLK_GLOBAL**.

CLKSEL[4] If HIGH, the **CLK_NEG_UP_OUT** on the top header is driven from the **HCLKEXT0** signal from the test chip.

If LOW, **CLK_NEG_UP_OUT** is driven by the **CLK_NEG_UP_IN** signal from the lower header on the Core Tile.

CLKSEL[5] If HIGH, the **CLK_NEG_DN_OUT** on the lower header is driven from the **HCLKEXT1** signal from the test chip.

If LOW, **CLK_NEG_DN_OUT** is driven by the **CLK_NEG_DN_IN** signal from the upper header on the Core Tile.

Note

Many of the existing Logic Tile clocks are not used by the Core Tile, for example **CLK_POS_DN**. Some Logic Tile general-purpose signals, however, are used as clock signals on the Core Tile, for example, **X_HCLK_DN** and **PLDCLK**.

The clock sources and clock outputs are different for different test chip variants. See the test chip information on the CD for details particular to the test chip used on the Core Tile.

For more information on clock generation and control for your test chip, see the chapter in this manual that covers your Core Tile, the *Technical Reference Manual* for the processor, and the Application Notes for using your Core Tile with other Versatile products.

The resistor links shown in Figure 3-3 on page 3-12 are fitted at manufacture depending on the Core Tile and do not require changing. See *Clock selection links* on page 6-33.

See *Core Tile PLD signals* on page 3-30 for details on controlling the **CLKSEL[5:0]** signals from the PLD. The **REFCLK** and **HCLKIN** sources for values of **CLKSEL[5:0]** are listed in Table 3-3 and Table 3-4.

Table 3-3 CLKSEL[5:0] signals and HCLKIN source

CLKSEL[5:0]	HCLKIN
bxxx000	**CLK_NEG_UP_IN**
b0xx001	**CLK_NEG_DN_IN**
b1xx001	**HCLKEXT1** (from test chip)
bxxx010	**X_HCLKIN_UP**
bxxx011	**X_HCLKIN_DN**
bxxx10x	**GND** (no clock input)
bxx011x	**CLK_GLOBAL** (driven from test chip **HCLKEXT4**)
bxx111x	**CLK_GLOBAL** (driven by an attached tile)

Table 3-4 CLKSEL[5:0] signals and REFCLK source

CLKSEL[5:0]	REFCLK
bxxxxx0	**X_REFCLK_UP**
bxxxxx1	**X_REFCLK_DN**

3.4.2 Overview of clock signals

The clock related signals are summarized in Table 3-5 on page 3-15.

Table 3-5 Clock-related signals on Core Tile

Signal	Direction	Description
CLK	-	Internal test chip clock from the PLL bypass multiplexer
CLK_GLOBAL	Input/output/thru	A global clock shared with all tiles in the stack. Each tile can accept or drive the signal. **HCLKEXT4** can drive this signal.
CLK_NEG_DN	Input/output/thru	A clock signal routed from the upper header (**CLK_NEG_DN_IN**) to the lower header (**CLK_NEG_DN_OUT**). The Core Tile can replace **CLK_NEG_DN_OUT** with **HCLKEXT1**.
CLK_NEG_UP	Input/output/thru	A clock signal routed from the lower header (**CLK_NEG_UP_IN**) to the upper header (**CLK_NEG_UP_OUT**). The Core Tile can replace **CLK_NEG_UP_OUT** with **HCLKEXT0**.
CLK_POS_UP, **CLK_POS_DN**, **CLK_UP_THRU**, **CLK_OUT_PLUS1**, **CLK_OUT_PLUS2**, **CLK_IN_PLUS1**, **CLK_IN_PLUS1**, and **CLK_DN_THRU**	Through	These Logic Tile clock signals are not used by the Core Tile. They are passed through the Core Tile unmodified for use by other tiles.
CLKSEL[5:0]	Local outputs from PLD	These signals (from the PLD) control the clock-selection multiplexors.
HCLK	-	Internal test chip local memory clock output (not present on all test chips).
HCLKDIV[2:0]	Input/through	Ratio of AHB clock to **CLK**.
HCLKEXT[4:0]	Output	Some test chips output buffered versions of **HCLK** as **HCLKEXT[4:0]**.
HCLKIN	Input/through	For some test chips, the internal AHB clock is supplied directly from this pin and is not divided down from the internal **CLK** signal.
PLDCLK	Input	Clocks configuration data into (and status data out of) the PLD on the Core Tile.
PLLCTRL[1:0]	Input/through	PLL control (typically power down and enable signals)
PLLFBDIV[7:0]	Input/through	PLL feedback divisor
PLLLOCK		In-lock indication from PLL
PLLOUTDIV[3:0]	Input/through	PLL output divider. The output of this divider is the internal **CLK** signal.

Table 3-5 Clock-related signals on Core Tile (continued)

Signal	Direction	Description
REFCLK	-	A reference clock to the test chip clock-dividers, driven from either the **XREFCLK_DN** or **XREFCLK_UP** inputs
X_CLK_UP, **X_CLK_DN**	Output	The test chip **HCLKEXT2** to the upper header as **X_CLK_UP** and **HCLKEXT3** to the lower header as **X_CLK_DN**.
X_HCLKIN_UP, **X_HCLKIN_DN**	Input	**X_HCLKIN_UP** from the lower header or **X_HCLKIN_DN** from the upper header can be selected as the source of **HCLKIN** to the test chip.
X_REFCLK_UP, **X_REFCLK_DN**	Input	**X_REFCLK_UP** from the lower header or **X_REFCLK_DN** from the upper header can be selected as the source of **REFCLK** to the test chip.

3.5 Power supply control

The power supply on the Core Tile is controlled by the PLD and by resistor links as shown in Figure 3-4.

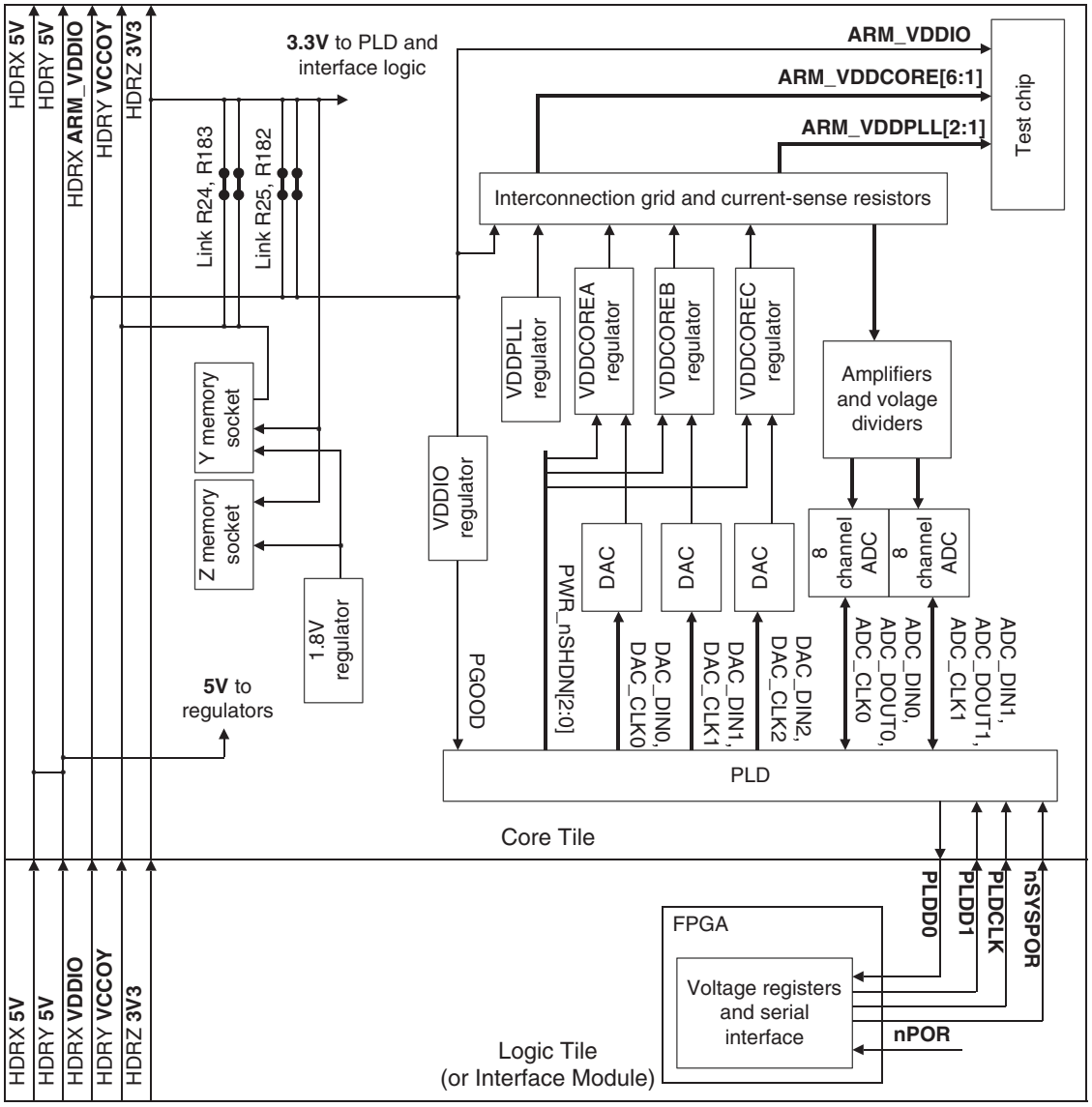

Figure 3-4 Voltage control and monitoring

Caution

Changing the resistors that control the core power-supply voltage might result in damage to the test chip, see *Setting the VDDCORE voltage* on page 3-21. The resistors that are fitted at manufacture enable you to modify the core voltage within a safe region.

Modifying the core voltage is not required for normal operation.

The resistor values and links described in this section are typical values, but the values on your Core Tile might be different. Refer to the BOM and schematic for your Core Tile for details of build options that are used for your board.

The Logic Tile FPGA implements registers and a PLD interface (as shown in Figure 3-4 on page 3-17) that control the PLD on the Core Tile to enable you to:

- Individually shutdown the programmable regulators that generate the test chip core and PLL voltages. The PLL outputs **PWR_nSHDN[2:0]** control the regulators that generate **VDDCOREC**, **VDDCOREB**, and **VDDCOREA** voltages.
- Change the voltage supplied to the core by writing values to the serially-programmed *digital to analog converters* (DACs).
- Read onboard voltages from two 8-channel 12-bit *analog to digital converter* (ADC):
  - **ARM_VDDCORE[6:1]**, **ARM_VDDIO**, and **ARM_VDDPLL[2:1]**. These voltages are divided by two before being fed to the ADC to prevent the ADC input range being exceeded.
  - **VDDCORE_DIFF[6:1]**. This is the voltage generated through current-sensing resistors in series with the **ARM_VDDCORE[6:1]** power connections to the test chip. The value of the voltage is proportional to the current consumed by the core.
  - **TP_SENSE**. Test point 22 is connected directly to the ADC.

  Note

  The ADC controller continuously reads the values from the ADC and stores them in the registers, see *PLD function after power on* on page 3-34. The serial interface between the PLD on the Core Tile and the FPGA on an attached Logic Tile regularly updates the registers in the PLD and FPGA.
- Detect that the regulators for the **ARM_VDDIO**, **VDDCOREC**, **VDDCOREB**, and **VDDCOREA** voltages are outputting the selected voltage. The **PGOOD** signals from the regulators are tied together and input to the PLD for transmitting on the serial interface.

3.5.1 Resistor links for power supply

The connections between the programmable regulators and the test chip voltage pins can be configured by resistor links on the printed circuit board as shown in Figure 3-5 on page 3-20. The Core Tile voltage sources are:

VDDIO The output of this regulator is input to the test chip and is connected to the **ARM_VDDIO** power blade on HDRX.

Resistor links R182 and R25 connect the 3.3V supply from HDRZ with the **VDDIO** voltage blades on HDRX. Do not use this link if you require custom I/O voltages on the HDRX blade.

The voltage level can be measured from ADC and a power good signal (**PGOOD**) is output from the regulator to the PLD.

VCCOY Some memory modules have the output of an onboard regulator connected to the **VCCOY** signals. **VCCOY** is present on the power blade of HDRY.

Resistor links R183 and R24 connect the 3.3V supply from HDRZ with the **VCCOY** voltage. Do not use this link if you require custom I/O voltages on the HDRY blade.

VDDPLL The output of this regulator is connected to the **VDDPLL1** input on the test chip. The voltage level can be measured from ADC.

VDDCOREA and VDDCOREB

The output of these two programmable regulators can be selected as the source for **VDDPLL2** and one or more of **VDDCORE[6:1]**.

VDDCOREC

The output of this programmable regulator can be selected as the source for one or more of **VDDCORE[6:1]**.

VDDCORE[6:1] interconnection

There are six groups of pads on the test chip dedicated to core voltage supply. The interconnection links to connect two or more of the groups together. These links are after the sense resistors (if fitted) therefore all of the connected voltage groups are at the same voltage. If the interconnection is made before the sense resistors, the actual voltages on the groups is different if the current loading is different. The voltage and current levels for **VDDCORE[6:1]** can be measured from ADC.

VDDPLL2 The **VDDPLL2** pads on the test chip do can be connected to **VDDPLL1**, **VDDIO**, **VDDCOREA**, or **VDDCOREB** regulator outputs. The voltage level can be measured from ADC.

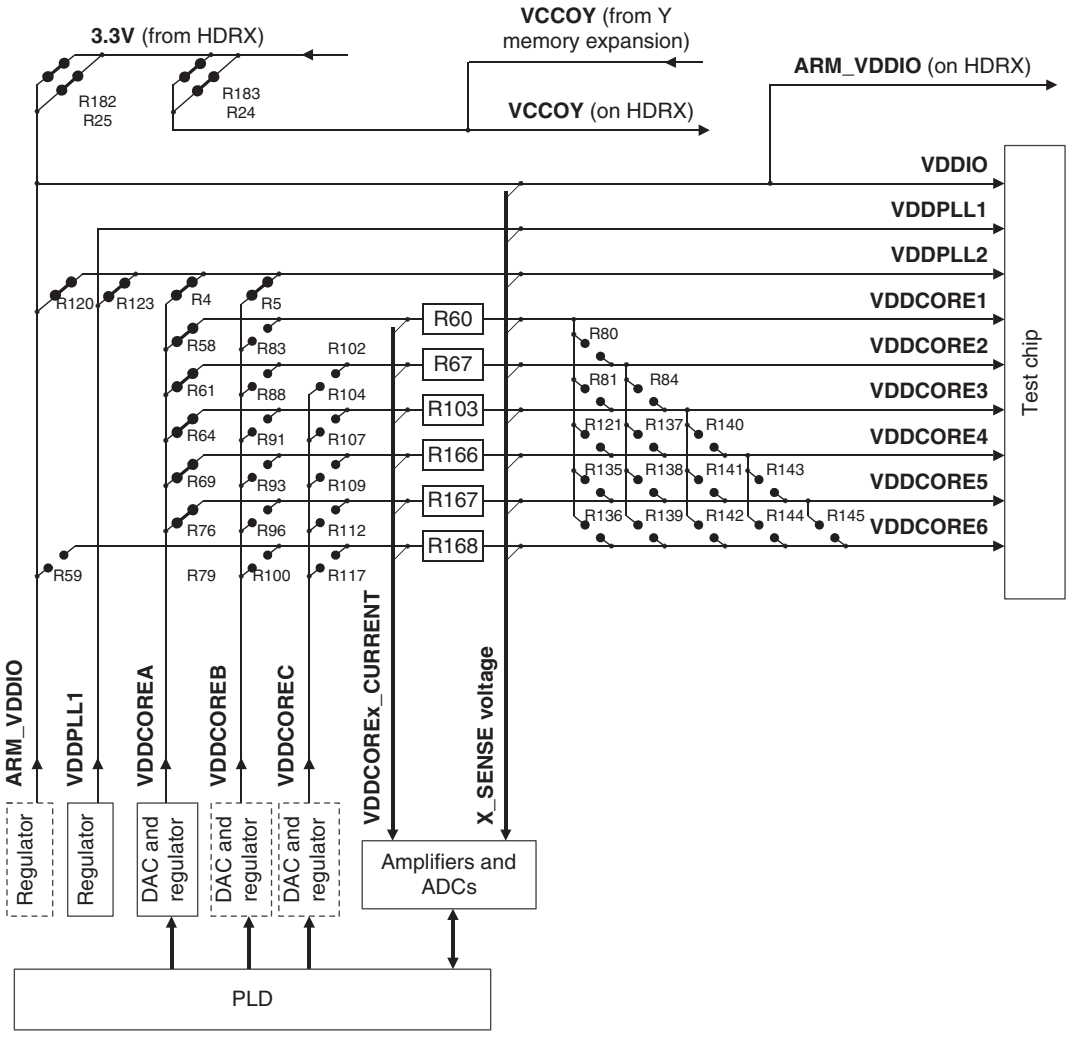

Figure 3-5 Voltage interconnection links

Note

Only a few of the resistor links shown in Figure 3-5 are fitted at manufacture. Also the ARM_VDDIO, VDDCOREA, and VDDCOREC regulators might not be fitted for some boards. **VDDIO** is typically powered from the 3.3V supply from the HDRX power blades.

3.5.2 Setting the VDDCORE voltage

The core voltages depend on:

- The feedback resistors for the regulators. Resistors R63, R74, and R176 are the feedback resistors for **VDDCOREA** supply as shown in Figure 3-6 on page 3-22.

  These define the core voltage at power-on and the adjustment range for **VDDCOREA**, **VDDCOREB**, and **VDDCOREC**. These resistors are fitted at manufacture to give the correct core voltage for the test chip fitted to the Core Tile.

- The values loaded into the PLD for the DAC settings. The values provide a positive or negative offset to the default power-on voltages.

- The interconnection links between the regulators and the **VDDCORE[6:1]** power connections, see Figure 3-5 on page 3-20.

The output voltages are given by:

VDDCOREA = 0.8V * (1 + R63/R74 + R63/R176) - CT_VOLTAGEx[7:0]*R63*(I_{DAC}/255)

VDDCOREB = 0.8V * (1 + R89/R94 + R89/R177) - CT_VOLTAGEx[7:0]*R89*(I_{DAC}/255)

VDDCOREC = 0.8V * (1 + R106/R113 +R106/R178) - CT_VOLTAGEx[7:0]*R106*(I_{DAC}/255)

where:

I_{DAC} The full-range DAC output current (50…A)

CT_VOLTAGEx[7:0]

The eight-bit data value loaded into the PLD from the Logic Tile FPGA. There are two DACs that drive the programmable regulators for **VDDCOREA** and **VDDCOREB**.

Resistor values

Refer to the BOM for the resistor values fitted to the Core Tile. The resistor values depend on the test chip fitted and the build variant. The resistor values are typically chosen to give a 0.5V adjustment range for the core voltage. The default value loaded into the DAC is `0x80`. A value of `0xFF` gives maximum negative offset from the default (-0.25V) and a value of `0x0` gives maximum positive offset from the default (+0.25V).

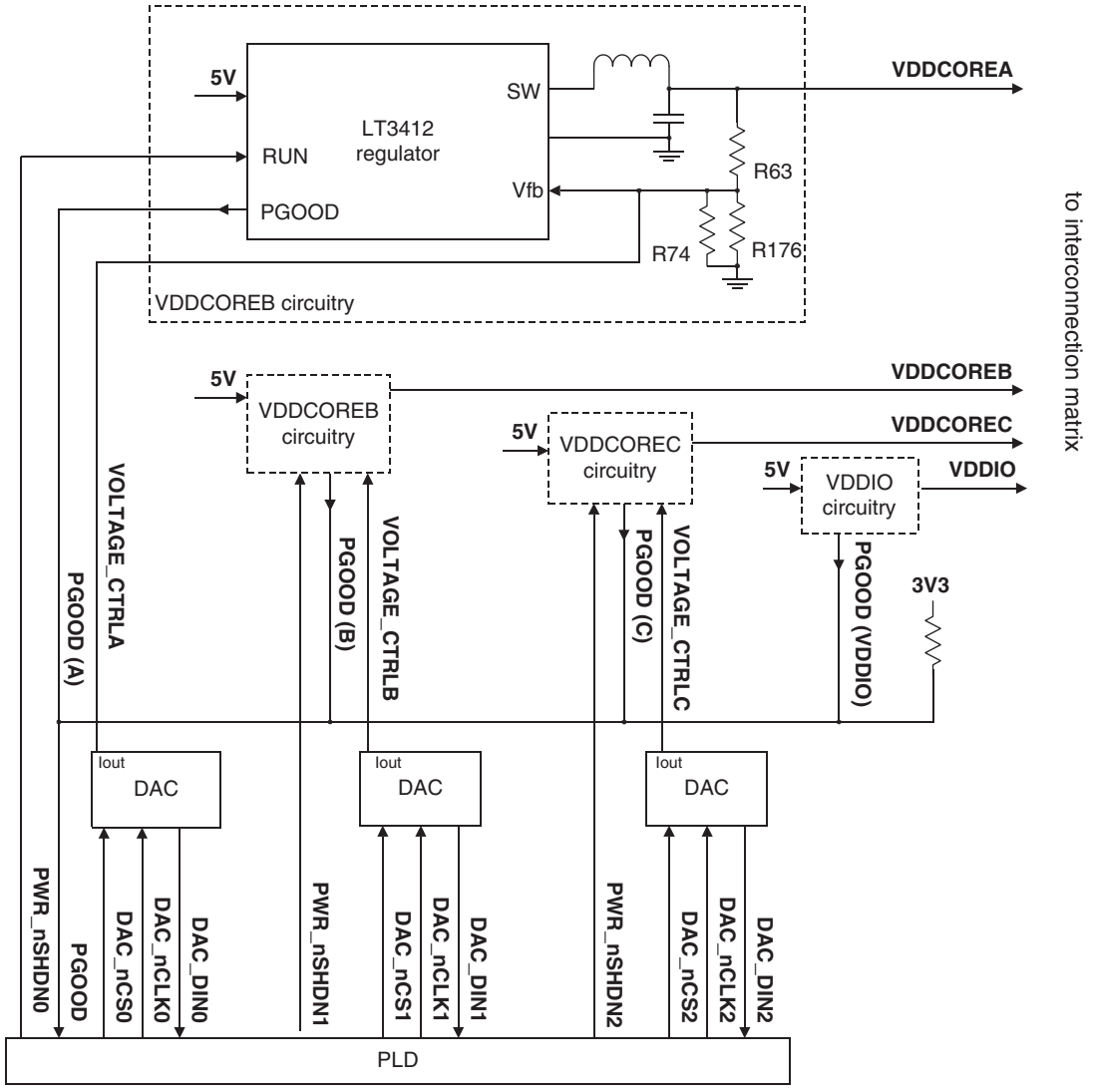

Figure 3-6 Programmable regulators

Shutdown of the VDDCORE regulators

The programmable regulators have can be individually shutdown by the **PWR_nSHDN[2:0]** signals from the PLD:

- if **PWR_nSHDN[0]** is LOW, the VDDCOREA regulator is shut off.
- if **PWR_nSHDN[1]** is LOW, the VDDCOREB regulator is shut off.
- if **PWR_nSHDN[2]** is LOW, the VDDCOREC regulator is shut off.

See *Core Tile PLD signals* on page 3-30 for a description of the PLD interface.

3.5.3 Reading the voltages and currents

The ADCs on the Core Tile continuously read the voltages and currents and update registers in the PLD, see *PLD function after power on* on page 3-34. The PLD transmits these values over the serial interface to the FPGA in the attached Logic Tile.

The LSB of the ADC reading corresponds to 1.221 mV. (Two times the 2.5V V_{REF} of the ADC divided by 4095.) The formula for the **VDDCORE**, **VDDPLL**, **VDDIO**, and voltages is:

$V_{VDD} = 1.221mV * VOLTAGE[11:0]$

Caution

All voltages except **VDDCOREx_DIFF** and **TP_SENSE** are divided by two before being fed to the ADC. The formula for the test point voltage is therefore:

$V_{TP} = 2.442mV * VOLTAGE[11:0]$

The **TP_SENSE** voltage should not exceed 2.5V for accurate readings and must not exceed 3.3V to prevent damage to the converters.

Voltages proportional to the **ARM_VDDCORE[6:1]** currents are developed across the sense resistors (see Figure 3-7 on page 3-24). Each of the sense resistors has a voltage amplifier because the sense voltage is too low to measure directly with the ADC. To calculate the current through an **ARM_VDDCOREx** line:

$I_{ARM\_VDDCOREx} = V_{ref} * VOLTAGE[11:0] / (R_{SENSE} * GAIN * (2^{12}-1))$

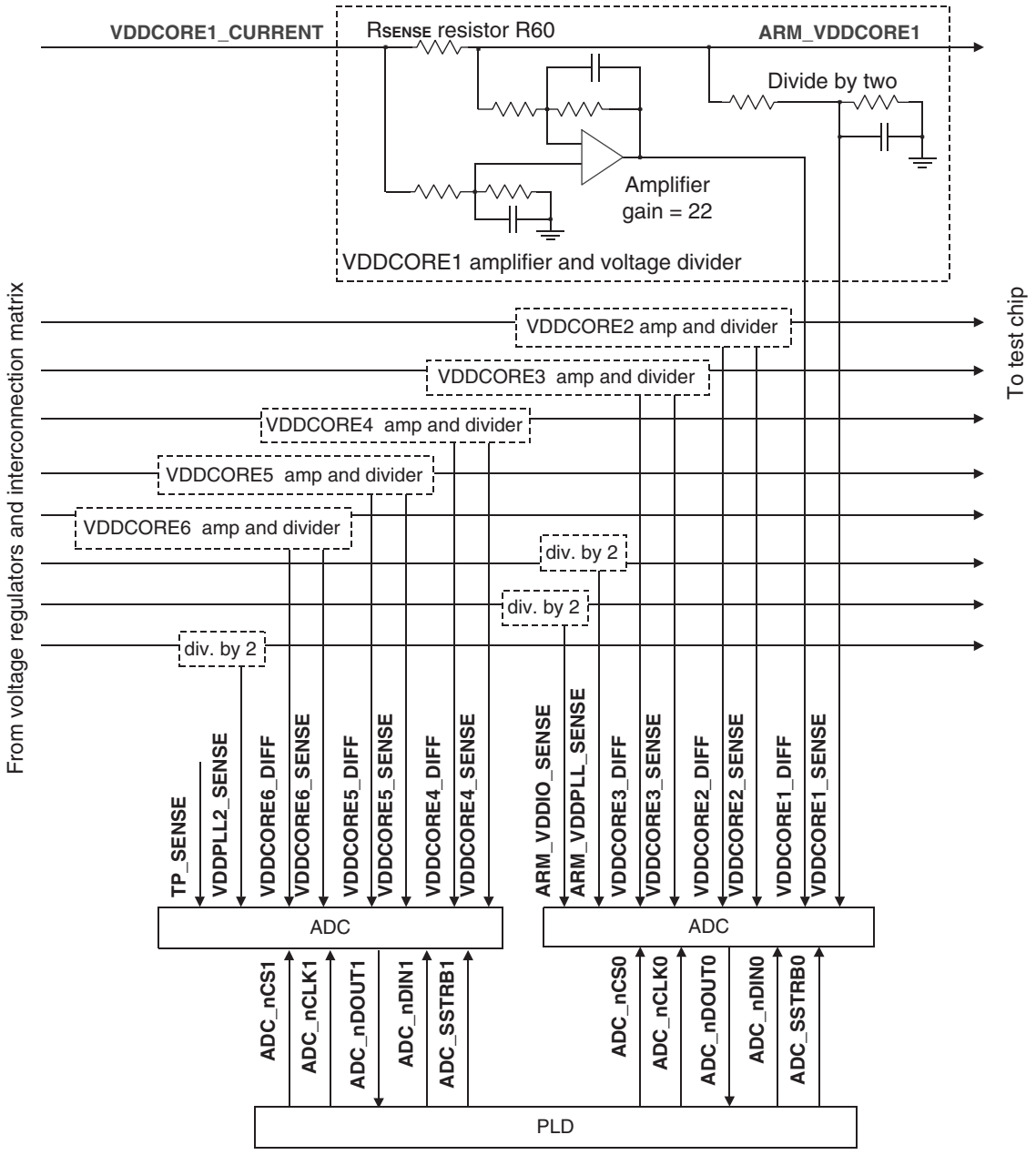

Figure 3-7 Voltage divider and current sense circuit

Note

By default, all of the R_{SENSE} resistors are 0Ω . It is not therefore possible to measure the current. You can replace the existing sense resistors with new ones of, for example, 1Ω.

If you change the sense resistors, adjust the core voltage to compensate for the voltage drop, for example, a 1Ω sense resistor drops 0.1V at 100mA. The default `GAIN` is 22, so a 1Ω sense resistor with 100mA current results in an ADC voltage input of 2.2V.

PGOOD signal

The power-good signal (**PGOOD**) to the PLD is pulled LOW if one of the **VDDCOREA**, **VDDCOREA**, **VDDCOREA**, or **VDDIO** output voltages is not within 7.5% of its selected value.

The **VDDCOREA**, **VDDCOREA** and **VDDCOREA** voltages can be measured by the ADC if the voltage outputs are connected to one of the **VDDCOREx_CURRENT** lines. Setting the VDDCORE voltages is described in *Setting the VDDCORE voltage* on page 3-21.

The **ARM_VDDIO** voltage is measured by the **ARM_VDDIO_SENSE** signal to the ADC. The **ARM_VDDIO** voltage is fixed at manufacture by the values of resistors R122, R129, and R179 (for the CT926EJ-S, these are 30kΩ, 11kΩ, and 15kΩ) and is given by:

```
ARM_VDDIO = 0.8V * ( 1 + R122/R129 + R122/R179) = 4.58V
```

Note

If link R123 is fitted, the **ARM_VDDIO** supply is connected directly to the 3.3V.

3.6 Control of AHB data bus and HDRZ signals

The Core Tile uses electronic switches to control signals to and from some of the header connectors:

- **HDATA[31:0]** on HDRX can be connected to **HRDATA[31:0]** or **HWDATA[31:0]**.

  The switches for the AHB data bus are required because not all test chips use a two-layer data bus with data out on **HWDATA[31:0]** and data in on **HRDATA[31:0]**.

- Signals **ZL[127:0]** can be disconnected from signals **ZU[127:0]**.

  For Core Tiles, **ZL[127:0]** signals are normally connected to the **ZU[127:0]** signals. The HDRZ signals can be split if, for example, there is a Logic Tile mounted on both the top and bottom of the Core Tile and the Logic Tiles have different functions for the **Zx[127:0]** signals.

3.6.1 Data read/write control for AHB data

Interconnection between **HDATA[31:0]**, **HRDATA[31:0]**, and **HWDATA[31:0]** is selected by control signals **DATACTL[2:0]** from the Core Tile HDRZ. Figure 3-8 on page 3-27 shows an example of a Logic Tile controlling the data bus switches. The typical data bus modes are listed in Table 3-6.

See *Core Tile PLD signals* on page 3-30 for a description of the PLD interface.

Table 3-6 Data bus switch function

DATACTL[2:0]	Connection	Description
b111	None	Multiplexed read and write data bus in test chip (on **HRDATA**) and from Logic Tile. The **HDATA** bus is not used on either the Core Tile or Logic Tile.
b110	**HDATA** to **HWDATA**	Separate read (**HRDATA**) and write (**HWDATA**) data buses in test chip and from Logic Tile
bx00	**HDATA** to **HWDATA**, **HRDATA** to **HWDATA**	Separate read and write data buses from Logic Tile are combined to connect to single multiplexed data bus (either **HRDATA** or **HWDATA**) on test chip
b011	**HRDATA** to **HWDATA**	Separate read and write data buses in test chip and multiplexed **HRDATA** bus from Logic Tile. The **HDATA** bus is not used on either the Core Tile or Logic Tile.

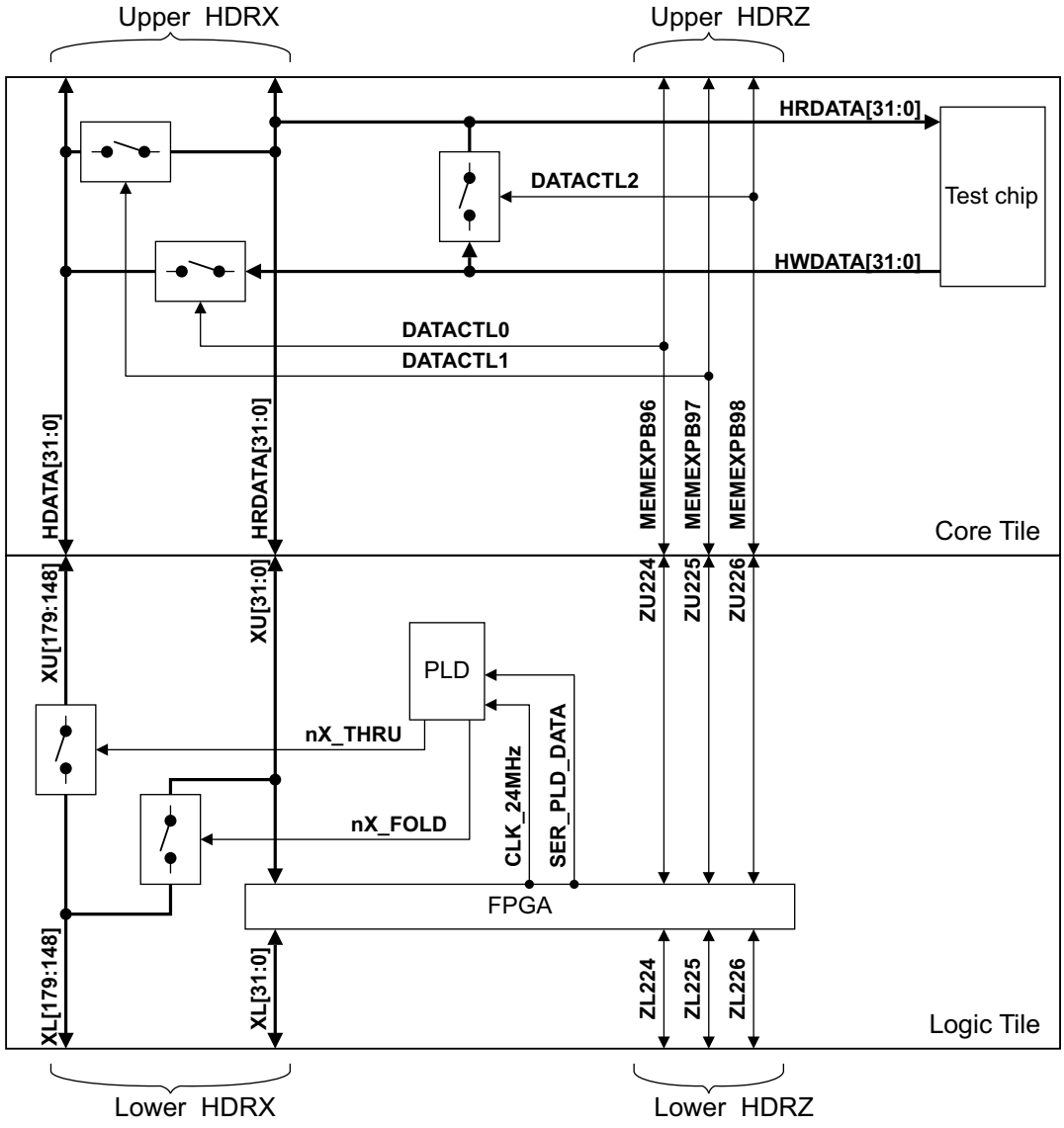

Figure 3-8 Read/write control

If the **DATACTLx** signal is LOW, the corresponding switches are closed.

3.6.2 Through/Break control for HDRZ

Interconnection between **ZL[127:0]** and **ZU[127:0]** is selected by control signals **ZCTL[3:0]** from the PLD as shown in Figure 3-9. If the **ZCTLx** signal is LOW, the corresponding switches are closed.

See *Core Tile PLD signals* on page 3-30 for a description of the PLD interface.

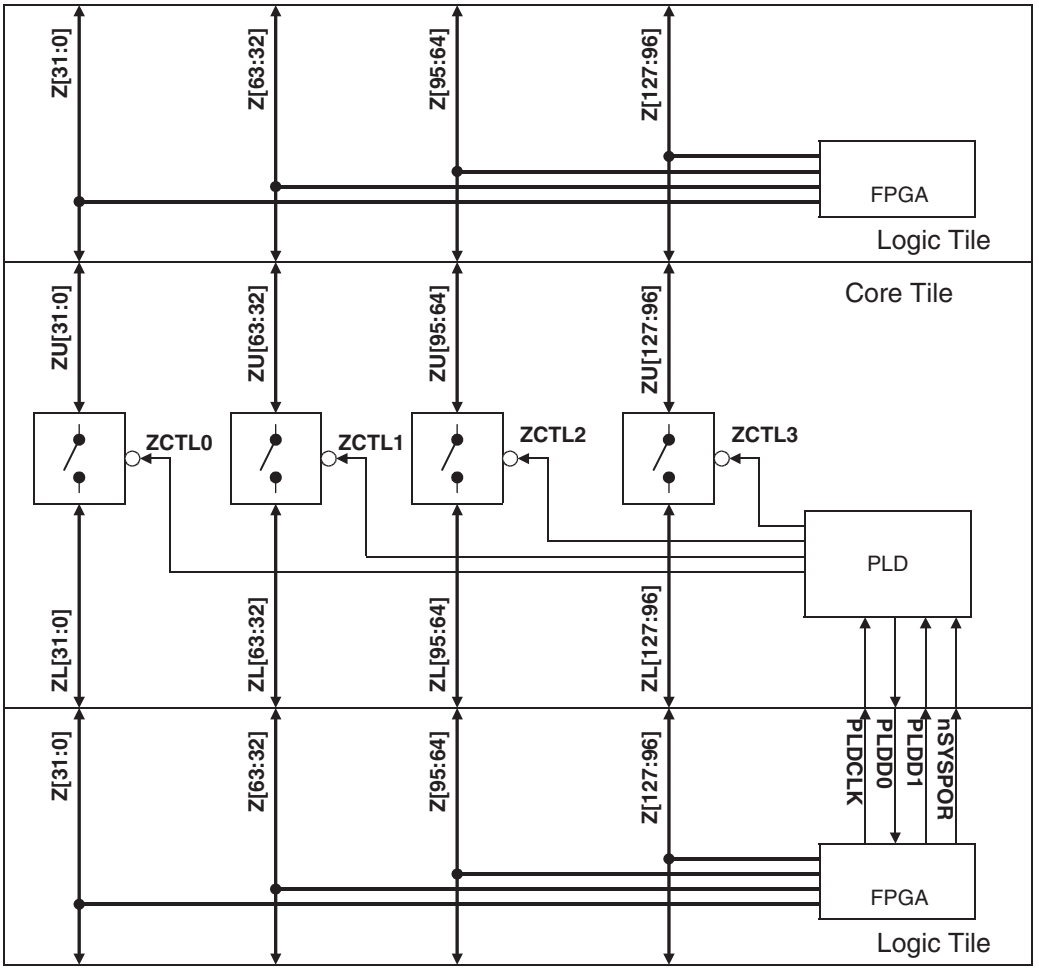

Figure 3-9 Through/break control

3.7 Overview of Core Tile configuration

The test chip, clock source, and voltages levels on the Core Tile are configurable from input signals on the header connectors and signals output from the on-board PLD. In a final product, core configuration is static and the core configuration signals are tied HIGH or LOW and the voltage and clocks are fixed. However, the Core Tile allows you to program these signals for experimentation.

There are several ways that Core Tile configuration occurs:

- Signals present on the Core Tile connector HDRX:
  - control signals for the PLL in the test chip (**PLLCTRL**, **PLLOUTDIV**, **PLLFBDIV**, **PLLREFDIV**, and **PLLBYPASS**)
  - **HCLKDIV** divider value for the **HCLKEXT** outputs
  - internal memory enable (**INITRAM**), memory order (**BIGENDIN** and **BIGENDOUT**) and vector location memory (**VINITHI**)
  - configuration control signals **CONFIGINIT** and **nCONFIGRST** are used by some test chips to load configuration data from the AHB data bus.

  HDRX signals are described in *Configuration signals on HDRX* on page 3-37.
- Configuration by the serial interface to the PLD on the Core Tile. (The PLD is normally controlled by a serial interface implemented in an FPGA present on an external Logic Tile.) The PLD controls the following signals:
  - the DAC inputs that select the voltages for the programmable regulators
  - the clock multiplexors
  - the HDRZ isolation switches,

  PLD configuration signals are described in *Core Tile PLD signals* on page 3-30.
- Control registers in the test chip:
  - Registers in coprocessor 15 are typically used to control endian settings and memory management units.
  - Some test chips (the CT1136JF-S, for example) have internal registers for controlling the test chip clocks.

  See the *Technical Reference Manual* for the core used in your test chip for details of register usage. See Chapter 5 *HBI-0131 Features specific to the CT1136JF-S*.

  You can use an external JTAG unit such as Multi-ICE or RealView ICE to modify the contents of the test chip registers or internal memory. The contents can of course also be modified by an application program.

- Links present on the printed circuit board. The links, usually 0Ω resistors, are normally set at manufacture to select voltage and clock options. These links are normally not modified after manufacture. Some of the functions performed by links are:
  - voltage range for the programmable regulators
  - clock selected for connection to multiplexors
  - manufacturing identification number
  - voltage source for header and memory connectors.

  Caution

  The resistor links are set at manufacture and do not normally require modification. Changing the voltage regulator resistors can result in excessive voltages being applied to the test chip.

To change the configuration of the processor, program the appropriate values in the control registers implemented in the FPGA on the Logic Tile (or IM-LT3). Depending on the test chip present in the Core Tile, the contents of registers in the test chip might require modification. See the application note for your product configuration for details on the control registers.

3.7.1 Core Tile PLD signals

The PLD on the Core Tile performs the following functions:

- loading data to the DACs that control the programmable power supplies, see *Power supply control* on page 3-17
- selectively shutting down the regulators on the Core Tile, see *Power supply control* on page 3-17
- reading data from the ADCs that monitor the test chip voltages, see *Reading the voltages and currents* on page 3-23
- controlling the clock selection multiplexors, see *Clocks* on page 3-10
- setting the size of the data and instruction TCM and cache, see *Memory located inside test chip* on page 3-8
- isolating HDRZ signals, see *Through/Break control for HDRZ* on page 3-28.

The PLD is controlled by the serial interface signals listed in Table 3-7 on page 3-31. These signals typically connect to an attached Logic Tile or IM-LT3 Interface Module. The FPGA in the external tile contains registers that hold the values to send to the PLD and received values from the PLD. The FPGA also provides the serialization and deserialization logic required for the PLD interface.

Note

The images for the Logic Tile FPGA and Core Tile PLD depend on the combination of boards that are in the stack. The application notes include FPGA and PLD images.

Table 3-7 PLD control signals

Signal	Description
PLDCLK	Clocks data into or out of the PLD
PLDD1	Serial data input to PLD
PLDD0	Serial data output from PLD
PLDRESETn	Reset selects mode for PLD (LOW is startup, HIGH is runtime configuration)

The Core Tile PLD manages the Core Tile configuration and status signals listed in Table 3-8

Table 3-8 PLD configuration signals

Signal	Direction	Function
ZCTL[3:0]	PLD output	Z Through control, see *Through/Break control for HDRZ* on page 3-28.
CLKSEL[4:0]	PLD output	Clock selection, see *Clocks* on page 3-10.
DACnCS[2:0]	PLD output	Chip select to DAC.
DAC_DIN[2:0]	PLD output	Data to DAC, see *Setting the VDDCORE voltage* on page 3-21.
ADC_nCS[1:0]	PLD output	Chip select to ADC.
ADC_CLK[1:0]	PLD input	Clock for data from ADC.
ADC_SSTRB[1:0]	PLD input	Strobe for ADC. Indicates that a conversion has finished.
ADC_DOUT[1:0]	PLD input	Data from ADC, see *Reading the voltages and currents* on page 3-23.
PGOOD	PLD input	Power good indication from power supply regulators, see *PGOOD signal* on page 3-25.

Table 3-8 PLD configuration signals (continued)

Signal	Direction	Function
DMEMSIZE[2:0]	PLD output	Test chip data TCM configuration, see *Memory located inside test chip* on page 3-8.
IMEMSIZE[2:0]	PLD output	Test chip instruction TCM configuration, see *Memory located inside test chip* on page 3-8.
MAN_ID[3:0]	PLD input	Board identification from resistor links. This is set at manufacture and identifies the board build.
PWR_nSHDN[2:0]	PLD output	Shutdown to the Vdd core power supplies, see *Power supply control* on page 3-17.

PLD function at power on

At power-on-reset, a controller in an external tile sends a configuration sequence to the PLD on the Core Tile as shown in Figure 3-10. The PLD control signals are described in Table 3-7 on page 3-31. **nSYSRST** is an external signal from an attached motherboard.

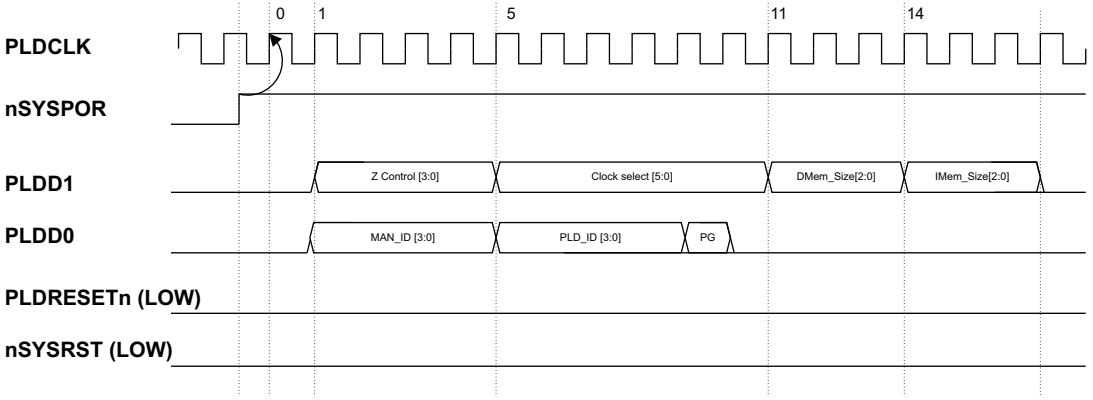

Figure 3-10 Power on signals to the Core Tile PLD

Table 3-9 lists the configuration and status signals that are received by or transmitted from the PLD at power on (after **nSYSPOR** goes HIGH).

Table 3-9 Power-on configuration signals by clock cycle

Clock	PLDD1	PLDD0	Description
-	LOW	LOW	No activity while **nSYSPOR** is LOW
0	LOW	LOW	First rising edge of clock after **nSYSPOR** goes HIGH
1	**ZCTL[3]**	**MANID[3]**	ZCTL data is clocked into the PLD and the state of the MANID links is clocked out of the PLD.
2	**ZCTL[2]**	**MANID[2]**	
3	**ZCTL[1]**	**MANID[1]**	
4	**ZCTL[0]**	**MANID[0]**	
5	**CLKSEL[5]**	**PLDID[3]**	CLKSEL multiplexor data is clocked into the PLD and the internal PLD ID is clocked out.
6	**CLKSEL[4]**	**PLDID[2]**	
7	**CLKSEL[3]**	**PLDID[1]**	
8	**CLKSEL[2]**	**PLDID[0]**	
9	**CLKSEL[1]**	**PGOOD**	
10	**CLKSEL[0]**	-	
11	**DMEMSIZE[2]**	-	The requested size for the TCM/cache data memory is clocked into the PLD.
12	**DMEMSIZE[1]**	-	
13	**DMEMSIZE[0]**	-	
14	**IMEMSIZE[2]**	-	The requested size for the TCM/cache instruction memory is clocked into the PLD.
15	**IMEMSIZE[1]**	-	
16	**IMEMSIZE[0]**	-	
17	LOW	LOW	No further activity until **nSYSRST** goes HIGH

PLD function after power on

After the power-on configuration finishes, the PLD is inactive until the **nSYSRST** signal goes HIGH. If **nSYSRST** is HIGH, The PLD continuously receives DAC packets from the serial interface and transmits ADC packets to the serial interface as shown in Figure 3-11.

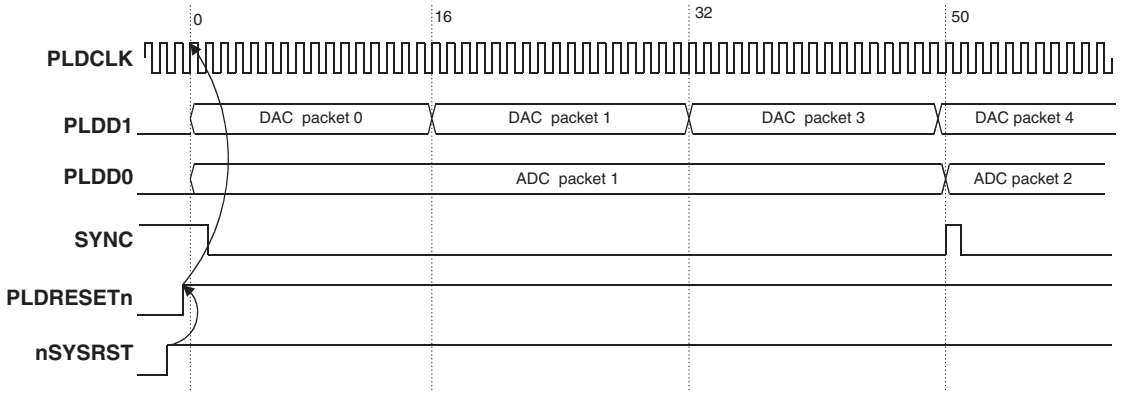

Figure 3-11 ADC and DAC data stream

Note

After **nSYSRST** goes HIGH, the ADC and DAC data packets repeat continuously.

The format of the ADC packet is shown in Figure 3-12.

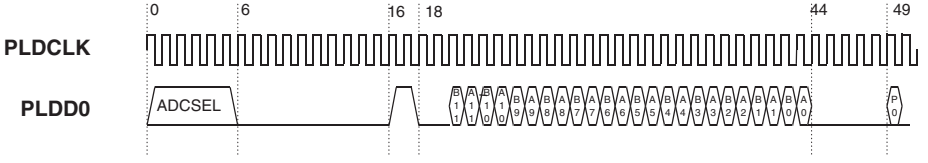

Figure 3-12 ADC packet format

The format of the DAC packet is shown in Figure 3-13 on page 3-35.

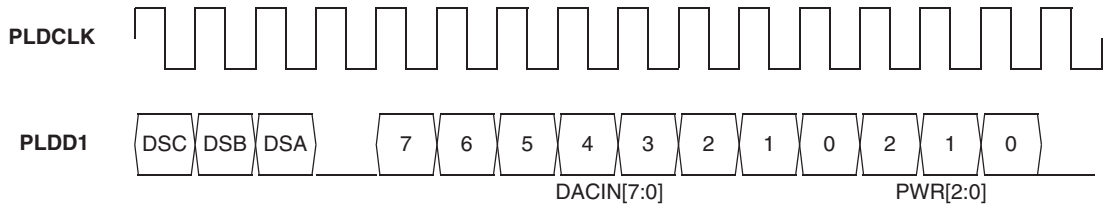

Figure 3-13 DAC packet format

Table 3-10 lists the PLD input and output signals after **nSYSPOR** goes HIGH.

Table 3-10 Run configuration signals by clock cycle

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
-	LOW	LOW	No activity while nSYSPOR is LOW
0	**DACSELA[2]**	**ADCSELB[2]**	First cycle after **nSYSPOR** goes HIGH. The **DACSEL** signals select which DAC value is transmitted in the current packet. The value of the ADCSEL signals indicate the ADC value that is transmitted in the current packet.
1	**DACSELA[1]**	**ADCSELA[2]**	
2	**DACSELA[0]**	**ADCSELB[1]**	
3	LOW	**ADCSELA[1]**	
4	**DACDIN[7]**	**ADCSELB[0]**	Start of DAC data.
5	**DACDIN[6]**	**ADCSELA[0]**	-
6	**DACDIN[5]**	LOW	-
7	**DACDIN[4]**	LOW	-
8	**DACDIN[3]**	LOW	-
9	**DACDIN[2]**	LOW	-
10	**DACDIN[1]**	LOW	-
11	**DACDIN[0]**	LOW	-
12	**PWR_nSHDN[2]**	LOW	Shutdown settings for power supply are clocked into the PLD.
13	**PWR_nSHDN[1]**	LOW	-
14	**PWR_nSHDN[0]**	LOW	-

Table 3-10 Run configuration signals by clock cycle (continued)

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
15	LOW	LOW	-
16	**DACSELA[2]**	HIGH	Second DAC packet starts with DAC selection bits.
17	**DACSELA[1]**	HIGH	
18	**DACSELA[0]**	LOW	-
19	LOW	LOW	-
20	**DACDIN[7]**	**ADC_DOUTB[11]**	ADC data starts. DAC data starts.
21	**DACDIN[6]**	**ADC_DOUTA[11]**	-
22	**DACDIN[5]**	**ADC_DOUTB[10]**	-
23	**DACDIN[4]**	**ADC_DOUTA[10]**	-
24	**DACDIN[3]**	**ADC_DOUTB[9]**	-
25	**DACDIN[2]**	**ADC_DOUTA[9]**	-
26	**DACDIN[1]**	**ADC_DOUTB[8]**	-
27	**DACDIN[0]**	**ADC_DOUTA[8]**	-
28	**PWR_nSHDN[2]**	**ADC_DOUTB[7]**	Shutdown settings for power supply are clocked into the PLD.
29	**PWR_nSHDN[1]**	**ADC_DOUTA[7]**	
30	**PWR_nSHDN[0]**	**ADC_DOUTB[6]**	
31	LOW	**ADC_DOUTA[6]**	
32	**DACSELA[2]**	**ADC_DOUTB[5]**	Third DAC packet starts.
33	**DACSELA[1]**	**ADC_DOUTA[5]**	-
34	**DACSELA[0]**	**ADC_DOUTB[4]**	-
35	LOW	**ADC_DOUTA[4]**	-
36	**DACDIN[7]**	**ADC_DOUTB[3]**	Start of DAC data
37	**DACDIN[6]**	**ADC_DOUTA[3]**	-
38	**DACDIN[5]**	**ADC_DOUTB[2]**	-

Table 3-10 Run configuration signals by clock cycle (continued)

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
39	**DACDIN[4]**	**ADC_DOUTA[2]**	-
40	**DACDIN[3]**	**ADC_DOUTB[1]**	-
41	**DACDIN[2]**	**ADC_DOUTA[1]**	-
42	**DACDIN[1]**	**ADC_DOUTB[0]**	-
43	**DACDIN[0]**	**ADC_DOUTA[0]**	End of ADC data. End of DAC packet.
44	**PWR_nSHDN[2]**	LOW	Shutdown settings for power supply are clocked into the PLD.
45	**PWR_nSHDN[1]**	LOW	
46	**PWR_nSHDN[0]**	LOW	
47	LOW	LOW	-
48	**DACSELA[2]**	LOW	Fourth DAC packet starts.
49	**DACSELA[1]**	**PGOOD**	Power good signal is clocked out of PLD. End of ADC packet.
50	**DACSELA[0]**	LOW	-

3.7.2 Configuration signals on HDRX

The signals listed in Table 3-11 are present on HDRX and are also used for configuring the Core Tile. Unless otherwise indicated, signal changes immediately affect the configuration. These signals typically connect to a Logic Tile (or IM-LT3 Interface Module) connected to the Core Tile.

Table 3-11 Configuration signals on header connectors

Signal	Direction	Description
PLLFBDIV[7:0]	Input	Feedback divisor for PLL logic. Clock dividers should be set at reset and not changed. Changing the clock dividers might cause an out-of-range clock or a glitch in the clock signal.
PLLREFDIV[3:0]	Input	Reference divisor for PLL logic.
PLLOUTDIV[3:0]	Input	Output divisor for PLL logic.
HCLKDIV[2:0]	Input	Divisor for PLL output frequency to HCLK frequency.

Table 3-11 Configuration signals on header connectors (continued)

Signal	Direction	Description
INITRAM	Input	Use this signal to enable (HIGH) or disable (LOW) the internal (tightly-coupled) SRAM. The signal is tested at reset.
VINITHI	Input	Use this signal to initialize the vector base address to `0xFFFF0000` (HIGH) or `0x00000000` (LOW). If the high vector address is used, ensure that there is physical memory at this location. The signal is tested at reset.
PLLBYPASS	Input	Use this signal to control clock usage by processors that contain an internal PLL. When **PLLBYPASS** is HIGH, the processor uses the clock signal supplied on its **PLLCLKIN** signal pin. When **PLLBYPASS** is LOW, the processor uses the clock signal supplied by its internal PLL (derived from **PLLCLKIN**).
BIGENDIN	Input	Use this signal to control whether the big-endian or little-endian memory organization is used. This signal is typically modified by setting a bit in CP15. When **BIGENDIN** is HIGH, the processor uses big-endian memory organization. The signal is tested at reset.
BIGENDOUT	Output	Indicates whether the big-endian or little-endian memory organization is used. When **BIGENDOUT** is HIGH, the processor uses big-endian memory organization.
HRDATA[31:0]	Input/output	Some test chips (the ARM1136JF-S for example) load configuration data at reset from the data bus.
CONFIGINIT	Input	Some test chips load test chip configuration from the HDATA bus when this signal goes HIGH.
nCONFIGRST	Input	Some test chips reset the test chip configuration to its default values when this signal is LOW.

3.7.3 Core configuration from ARM CP15 and on-chip registers

Some processor configuration settings (for example, cache enable/disable, MMU enable/disable, TCM enable/disable, and translation registers) are controlled by setting bits in coprocessor 15 (CP15) registers.

For example, set bit 7 of CP15 register r1 to select big-endian operation. For more information on system control via the CP15 registers, see the *Technical Reference Manual* for the processor.

3.8 JTAG support

JTAG signals are present on both the upper and lower HDRZ connectors. An external board provides the JTAG connector and the routing of the JTAG signals from the connector to HDRZ (see *JTAG signals*). The Core Tile routes the JTAG scan path through devices on the board. The logic devices that are placed in the Core Tile scan chain depend on the JTAG mode:

Debug mode Debug mode is selected by default. When a jumper is *not* fitted at the CONFIG link on the external Interface Module or baseboard. It is the mode used for general system development and debug, including using trace, see *Embedded Trace support* on page 3-47. In this mode, the JTAG signals flow through the Debug Scan Chain, this typically just connects to the system microprocessor.

Configuration mode

In configuration mode, all FPGAs and PLDs in the system are placed into the scan chain. This mode allows the programmable logic devices in the system to be reprogrammed.

To select configuration mode, fit a jumper to the CONFIG link on the Interface Module or baseboard. This pulls the **nCFGEN** signal LOW on the system and reroutes the JTAG scan path.

3.8.1 JTAG signals

There are two separate JTAG paths through the Core Tile:

- One is used for configuration of programmable devices (the PLD for example). If the test chip has a boundary scan interface, this is included in the scan chain. These JTAG signals are identified by the **C_** prefix.
- One is used in debug mode to connect to the debug TAP controller in the test chip. These JTAG signals are identified by the **D_** prefix.

Table 3-12 on page 3-40 provides a description of the JTAG signals.

Note

In the description in Table 3-12 on page 3-40, the term JTAG equipment refers to any hardware that can drive the JTAG signals to devices in the scan chain. Typically, Multi-ICE or RealView ICE is used, although you can also use hardware from third-party suppliers to debug ARM processors.

Some signals are split in to configuration and debug versions. For example, **D_TDO** is the data signal for the debug mode chain and **C_TDO** is the data signal for the configuration mode chain.

Table 3-12 JTAG signal description

Name	Description	Function
nBSTAPEN	Boundary scan TAP enable	In configuration mode, the boundary scan TAP logic (if present) in a test chip is enabled.
DBGRQ	Debug request (from JTAG equipment)	**DBGRQ** is a request for the processor core to enter the debug state.
DBGACK	Debug acknowledge (to JTAG equipment)	**DBGACK** indicates to the debugger that the processor core has entered debug mode.
DONE	FPGA configured	**DONE** is an open-collector signal that indicates when FPGA configuration is complete. Although this signal is not a JTAG signal, it does affect **nSRST**. The **DONE** signal is routed between all FPGAs in the system. The master reset controller on the motherboard senses this signal and holds all the boards in reset (by driving **nSRST** LOW) until all FPGAs are configured.
nCFGEN	Configuration enable (from jumper on the board at the bottom of the stack)	**nCFGEN** is an active LOW signal used to put the boards into configuration mode. In configuration mode all FPGAs and PLDs are connected to the scan chain so that they can be configured by the JTAG equipment.
nRTCKEN	Return **TCK** enable (from Core Tile to motherboard)	**nRTCKEN** is an active LOW signal driven by any Core Tile that requires **RTCK** to be routed back to the JTAG equipment. If **nRTCKEN** is HIGH, the motherboard drives **RTCK** LOW. If **nRTCKEN** is LOW, the motherboard drives the **TCK** signal back up the stack to the JTAG equipment.

Table 3-12 JTAG signal description (continued)

Name	Description	Function
nSRST	System reset (bidirectional)	**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 a board has been reset by the user. The open collector **nSRST** reset signal can be driven LOW by the reset controller on the Core Tile to cause the motherboard to reset the whole system by driving the motherboard signal **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.
nTRST	Test reset (from JTAG equipment)	This active LOW open-collector signal is used to reset the JTAG port and the associated debug circuitry on the processor. It is asserted at power-up by each module, and can be driven by the JTAG equipment. This signal is also used in configuration mode to control the programming pin (**nPROG**) on FPGAs.
RTCK, **D_RTCK**	Return TCK (to JTAG equipment)	Some devices sample **TCK** (for example a synthesizable core with only one clock), and this has the effect of delaying the time at which a component actually captures data. **RTCK** is a mechanism for returning the sampled clock to the JTAG equipment, so that the clock is not advanced until the synchronizing device captured the data. In *adaptive clocking mode*, Multi-ICE is required to wait until it detects an edge on **RTCK** before changing **TCK**. In a multiple device JTAG chain, the **D_RTCK** output from a component connects to the **TCK** input of the down-stream device. The **RTCK** signal on the module connectors HDRB returns **TCK** to the JTAG equipment. **Note** If an Integrator motherboard is present (**nMBDET** LOW), the **RTCK** signal to the motherboard is gated with **RTCKEN**. **nRTCKEN** is HIGH if there are no synchronizing components in the scan chain the **RTCK** signal returned from the motherboard is disabled. **D_RTCK** is the **RTCK** signal in the debug scan chain.

Table 3-12 JTAG signal description (continued)

Name	Description	Function
TCK, **D_TCK**, **C_TCK**	Test clock (from JTAG equipment)	**TCK** synchronizes all JTAG transactions. **TCK** connects to all JTAG components in the scan chain. Series termination resistors are used to reduce reflections and maintain good signal integrity. **TCK** flows up the stack of modules and connects to each JTAG component. However, if there is a device in the scan chain that synchronizes **TCK** to some other clock, then all down-stream devices are connected to the **RTCK** signal on that component (see **RTCK**). **D_TCK** is the clock for the debug mode chain and **C_TCK** is the clock for the configuration mode chain.
TDI, **D_TDI**, **C_TDI**	Test data in (from JTAG equipment)	**TDI** goes up the stack of tiles from the baseboard (or Interface Module) and then back down the stack (as **TDO**) connecting to each component in the scan chain. **D_TDI** is the data signal for the debug mode chain and **C_TDI** is the data signal for the configuration mode chain.
TDO, **D_TDO**, **C_TDO**	Test data out (to JTAG equipment)	**TDO** is the return path of the data input signal **TDI**. For a stack of Versatile products, **TDI** goes up to the top of the stack and returns down as **TDO**. The JTAG components are connected in the return path so that the length of track driven by the last component in the chain is kept as short as possible. **D_TDO** is the data signal for the debug mode chain and **C_TDO** is the data signal for the configuration mode chain.
TMS	Test mode select (from JTAG equipment)	**TMS** controls transitions in the tap controller state machine. **TMS** connects to all JTAG components in the scan chain.

JTAG data and clock logic

Figure 3-14 on page 3-43 shows the JTAG clock and data signals.

Note

Single-pole switches are closed if a HIGH signal is present at the control input. Multi-pole switch positions are marked with a 1 for the condition where a HIGH signal is present at the control input.

For Figure 3-14 on page 3-43, **nCFGEN** and **nTILEDET** are both HIGH indicating normal debug mode and no tile connected above the Core Tile.

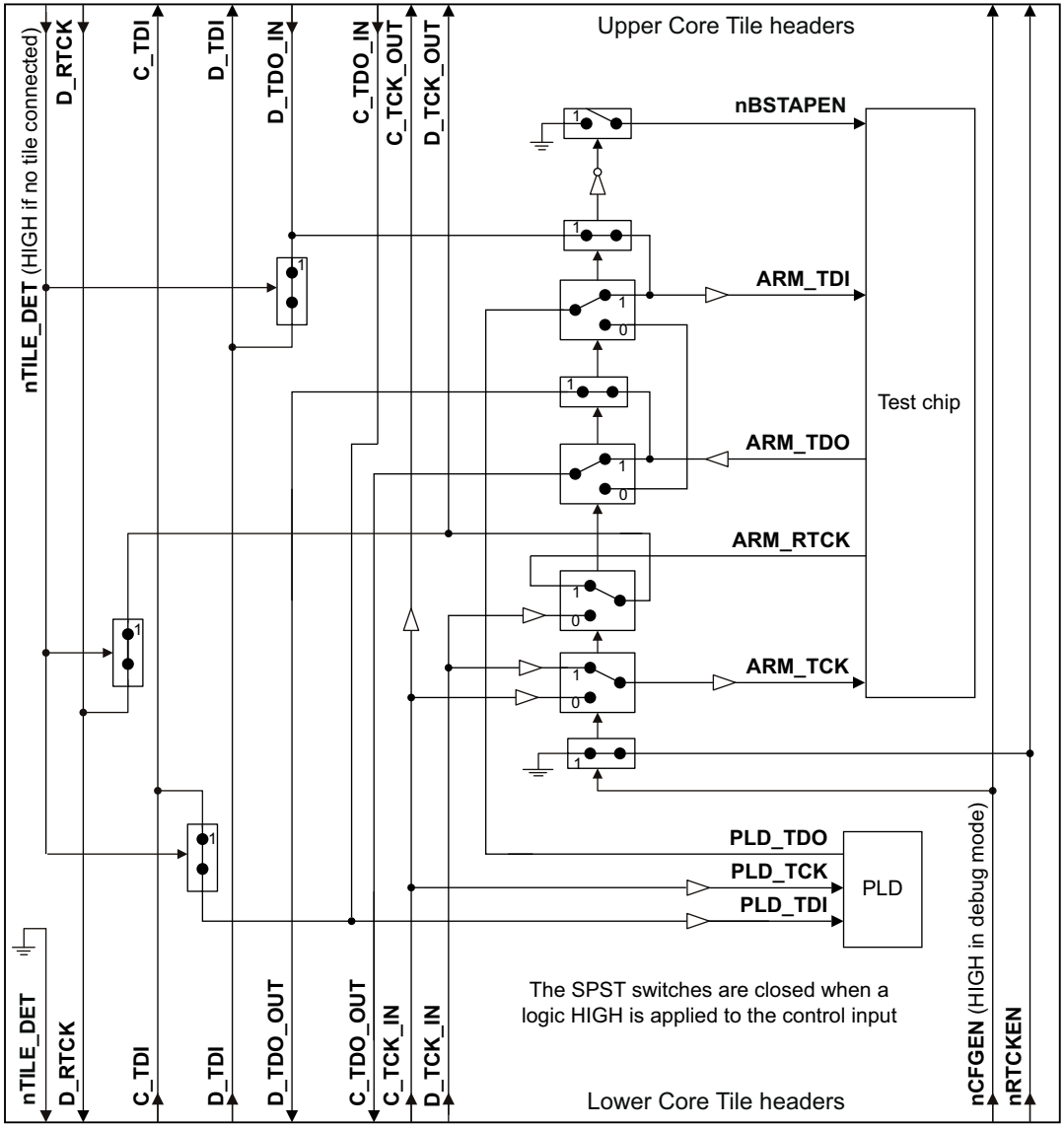

Figure 3-14 JTAG clock and data signals

The clock and data flow in debug and config mode are different. Figure 3-15 on page 3-44 and Figure 3-16 on page 3-44 show the equivalent flow for each mode. The switches have been replaced by direct connections and the diagram simplified. There is not a tile present above the Core Tile (**nTILEDET** is pulled HIGH).

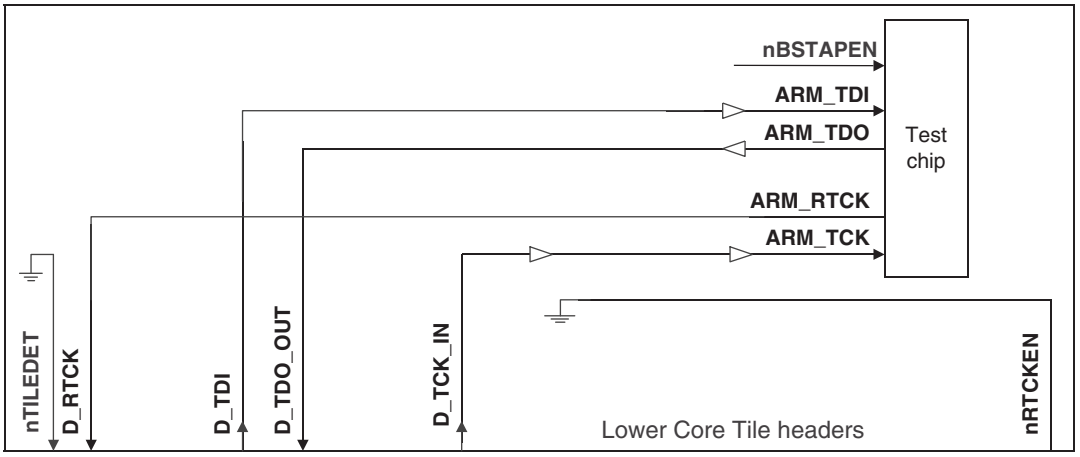

Figure 3-15 JTAG data flow in debug mode

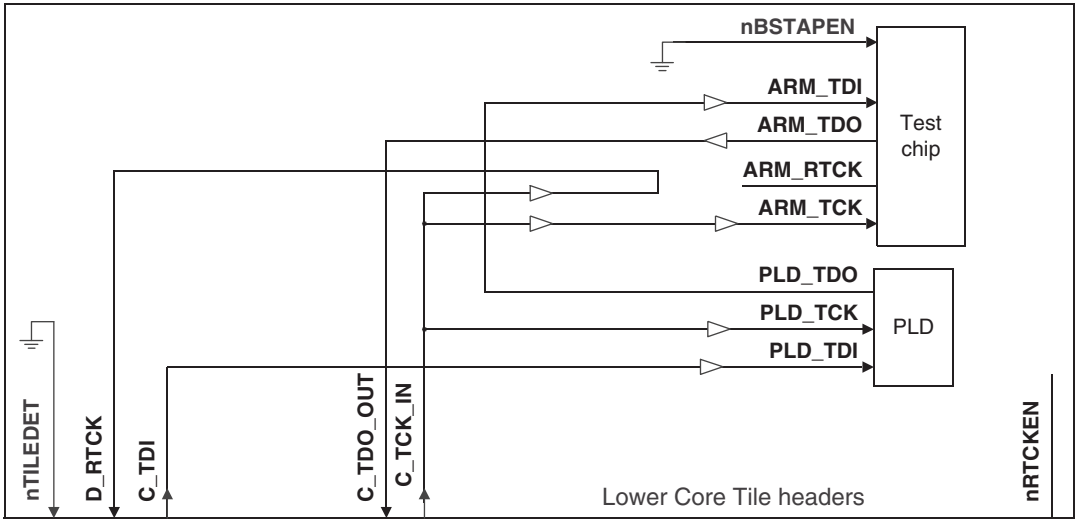

Figure 3-16 JTAG data flow in config mode

JTAG reset and configure logic

Figure 3-17 shows the reset and TMS signals.

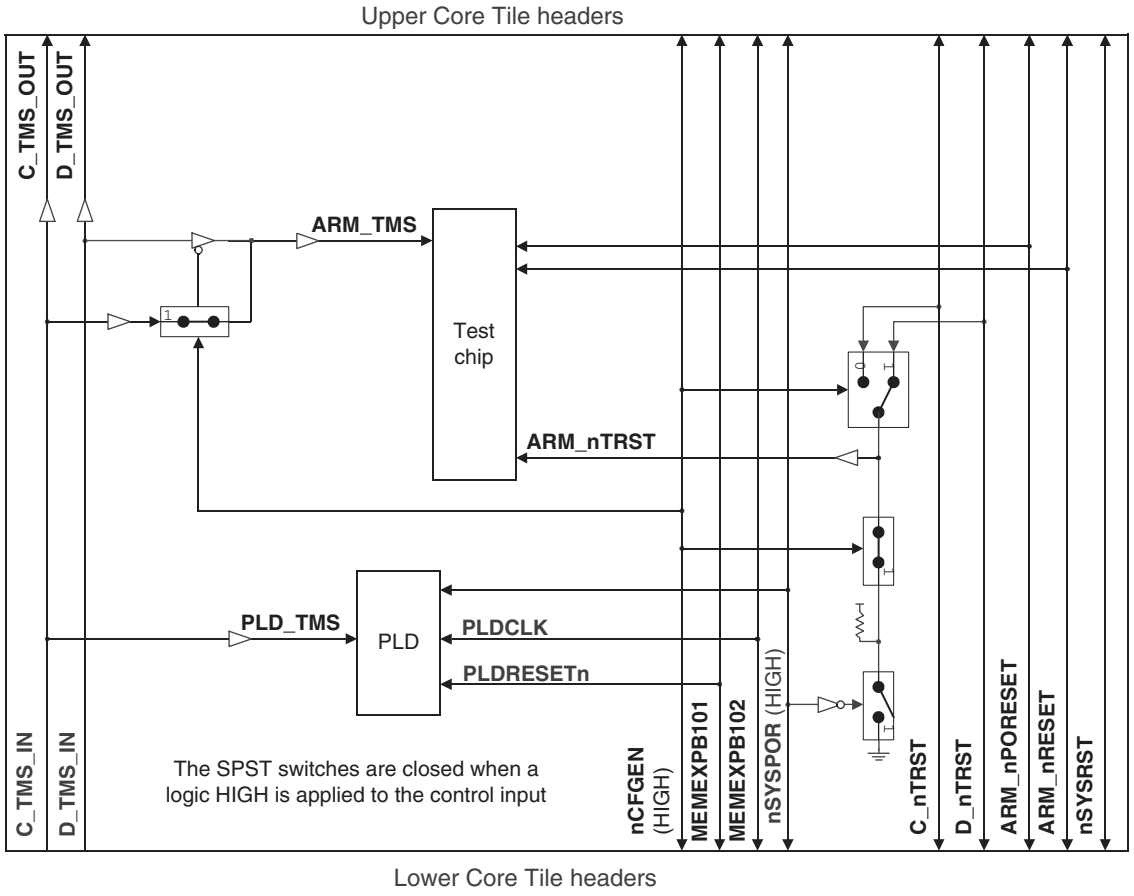

Figure 3-17 JTAG reset signals

3.8.2 Debug communications interrupts

The processor core incorporates EmbeddedICE logic that contains a communications channel used for passing information between the core and the JTAG equipment. The debug communications channel is implemented as coprocessor 14.

The processor accesses the debug communications channel registers using `MCR` and `MRC` instructions. The JTAG equipment reads and writes the register using the scan chain. For a description of the debug communications channel, see the Technical Reference Manual for your test chip.

The debug interrupt signals from the test chip are connected to HDRX pin 66 (**COMMTX**) and pin 68 (**COMMRX**).

3.9 Embedded Trace support

Many ARM test chips incorporate an *Embedded Trace Macrocell* (ETM).

The trace macrocell enables you to carry out real-time debugging by connecting external trace equipment to the Core Tile. To trace program flow, the ETM broadcasts branch addresses, data accesses, and status information through the trace port. Later in the debug process, the complete instruction flow can be reconstructed by debug software running on a host machine that is connected to the RealView Trace or Multi-Trace interface.

The logic analyzer connection is a high-density AMP Mictor connector. The pinout for this connector is provided in *Trace Port connectors* on page 6-27.

Note

The presence (and size) of the ETM packets depends on the test chip fitted to the board. ETM functionality is not guaranteed for all test chip versions.

Chapter 4
HBI-0131 Features specific to the CT926EJ-S

This chapter describes features specific to the CT926EJ-S Core Tiles that use the HBI-0131 printed circuit board. It contains the following sections:

Note

Additional information on the CT926EJ-S might be available on the release notes supplied with the product.

4.1 ARM926EJ-S test chip characteristics

The CT926EJ-S Core Tile test chip conforms to the generic test chip specification.

ARM926EJ-S processor macrocell in the test chip is based on the ARM9EJ-S™ RISC processor core. The macrocell is user-code compatible with code written for earlier versions of the ARM architecture (for example, ARM7TDMI™ and ARM9TDMI™). They also support the instructions specific to the ARMv5TE architecture. The additional instructions improve DSP performance.

The ARM926EJ-S combines the ARM9EJ-S core with:

- instruction and data caches
- tightly coupled instruction and data SRAM
- write buffer
- memory management unit
- ETM9 Embedded Trace Macrocell.

See the *ARM926EJ-S Technical Reference Manual* for more information on the ARM926EJ-S processor.

4.2 Clocks

Clock related signals for the CT926EJ-S are shown in Figure 4-1.

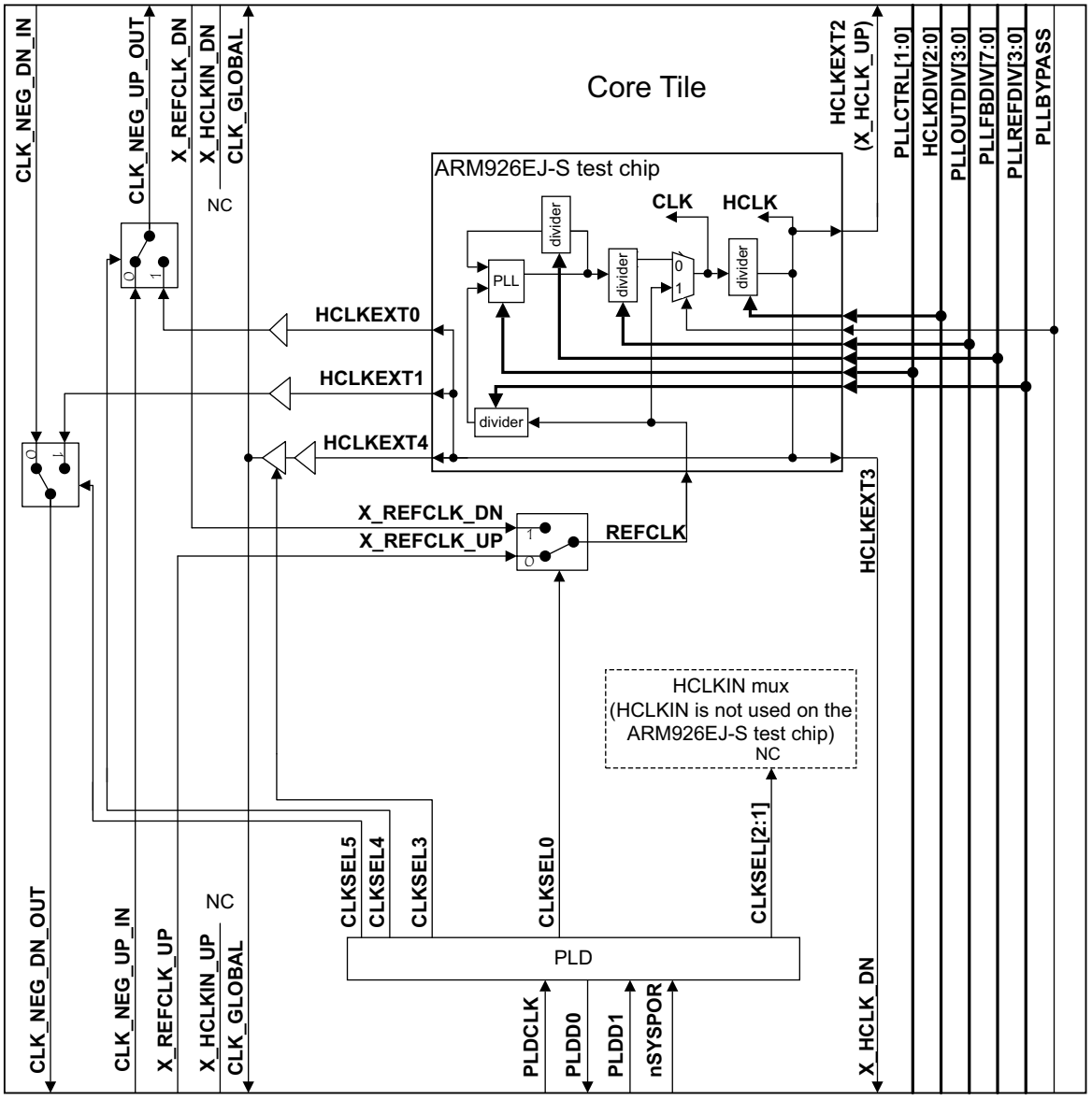

Figure 4-1 Clock signals

The ARM926EJ-S test chip on the Core Tile accepts or generates the following clocks:

REFCLK A reference clock to the test chip clock dividers. This is the source for the PLL incorporated into the test chip. The PLL generates the core clock **CLK** that is used internally in the test chip.

CLK the core clock (this is internal to the test chip). The frequency of **CLK** is controlled by a PLL within the test chip.

The internal PLL can be bypassed by setting **PLLBYPASS** HIGH. This sets **CLK** to the same frequency as the reference clock and allows you to model implementations where the core does not have a PLL.

HCLK The test chip produces the **HCLK** signal by dividing **CLK** by a value of between 1 and 8 as determined by **HCLKDIV[2:0]**. The divider can be effectively bypassed by setting a divide by value of 1. The **HCLK** signal is distributed by the test chip on five identical signals:

- **HCLKEXT0** can be selected for **CLK_NEG_UP_OUT**
- **HCLKEXT1** can be selected for **CLK_NEG_DN_OUT**
- **HCLKEXT2** is output on the upper header as **X_HCLK_UP**
- **HCLKEXT3** is output on the lower header as **X_HCLK_DN**
- **HCLKEXT4** can be selected for **CLK_GLOBAL**.

See *Clocks* on page 3-10 and the *ARM926EJ-S Technical Reference Manual* for details of the Core Tile clock signals.

The core clock frequency for the CT926EJ-S is calculated from:

```
CLK = (REFCLK/(PLLREFDIV[3:0]+1) * (PLLFBDIV[7:0]+1)) / (PLLOUTDIV[2:0]+1)
```

That is:

```
137kHz < REFCLK/PLLRFDIV[3:0] + 1 < 275MHz

55MHz < CLK < 275MHz
```

Note

The ARM926EJ-S test chip does not use the **HCLKIN** signal. The test chip **HCLK** is always generated internally based on the **REFCLK** input and the dividers.

The ARM926EJ-S Core Tile has a manufacturing test block that, if enabled, changes the clock division ratios and disables the interrupt request signals. The test block can be disabled by writing `0x0` to the USERIN bits of the SYS_CONFIGDATA1 register on the Emulation Baseboard. The block can also be disabled by using a FPGA image in the baseboard that drives the **USERIN[3]** signal LOW.

If the PLLBYPASS signal is HIGH, the core uses the external clock instead of the internal PLL. (This signal is typically LOW.)

4.3 Memory map

The behavior of memory accesses from the core depend on the operating mode and the specific test chip. In general, bus accesses in the range `0x3FF00000`–`0x3FFFFFFF` are reserved for on-chip registers and other accesses are presented on the test chip external 32-bit wide AHB interface.

4.4 Voltage control

The CT926EJ-S Core Tile voltage connections are listed in Table 4-1.

Table 4-1 Supply voltage connections

Voltage	Source	Description
ARM_VDDCORE1, **ARM_VDDCORE2**, **ARM_VDDCORE3**, **ARM_VDDCORE4**, **ARM_VDDCORE5**, and **ARM_VDDCORE6**	VDDCOREA regulator	All core voltages are supplied from the VDDCOREA regulator.
VDDIO	3.3V	The I/O voltage is supplied from the HDRX 3.3V power blade (See *Power supply control* on page 3-17).
VDDPLL1	VDDPLL regulator	Supplied from the dedicated PLL regulator.
VDDPLL2	VDDPLL regulator	Supplied from the dedicated PLL regulator.

Chapter 5
HBI-0131 Features specific to the CT1136JF-S

This chapter describes features specific to the CT1136JF-S Core Tiles that use the HBI-0131 printed circuit board. It contains the following sections:

Note

Additional information on the CT1136JF-S might be available on the release notes supplied with the product.

5.1 ARM1136JF-S test chip characteristics

The CT1136JF-S Core Tile uses test chips that conform to the generic test chip specification. The test chip in the CT1136JF-S contains:

- a Clock Generator Control Block that configures the clocking scheme for the core
- an AHB matrix
- a vectored interrupt controller (VIC)
- Tightly-coupled memory (TCM)
- Data and instruction cache
- The test chip is contains an ETM11RV embedded trace macrocell.

See the *ARM1136JF-S Technical Reference Manual* for further details of the TCMs and cache memory. See the *ETM11RV Technical Reference Manual* for more details on the ETM.

5.2 Clocks

The ARM1136JF-S can be operated with synchronous or asynchronous core and internal bus clocks.

If the ARM1136JF-S is operating synchronously, the core frequency is divided to clock the internal bus and the Core Tile local memory bus. The dividers are programmable and set the bus frequencies.

5.2.1 Test chip clock control

During a power-on reset the value on the **HRDATA[31:0]** pins is loaded into the test chip PLL configuration register. The value comes from the attached Logic Tile FPGA image and typically sets:

- the clock circuit to synchronous operation with a ratio of 1:5:5
- **REFCLK** is an input to the test chip
- the PLL control is set to BYPASS by default
- the CPU core runs from the core clock **CLK**.

The core clock frequency can be set from the clock control logic on the external Logic Tile. The ARM1136JF-S test chip does not use the HCLK DIV functions implemented on other Core Tiles. For details of the clock selection and routing logic that is inside the test chip, see the *ARM1136JF-S Technical Reference Manual*.

Note

If the PLLBYPASS signal is HIGH, the core uses the external clock instead of the internal PLL. (This signal is typically LOW.)

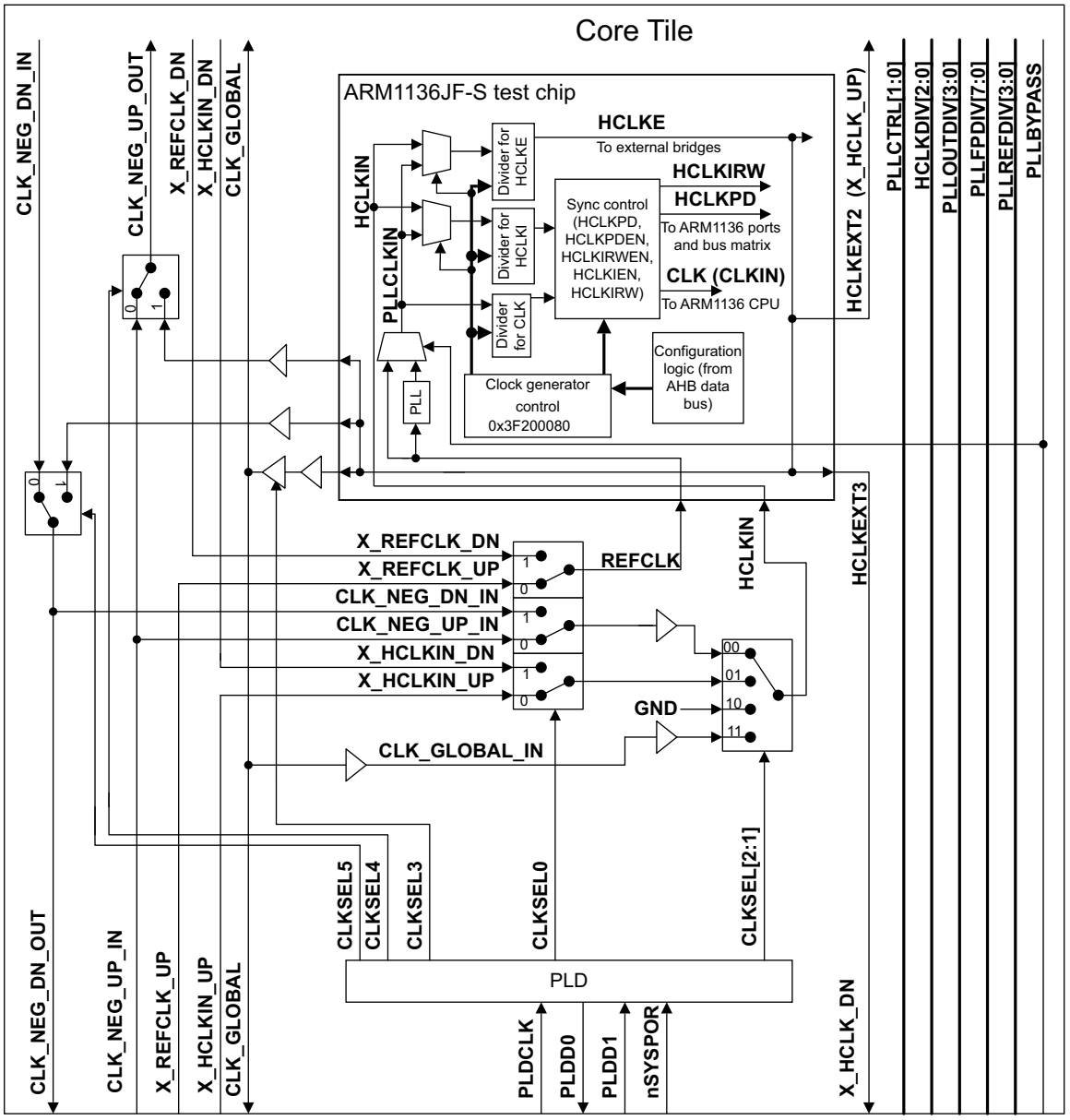

Figure 5-1 Clock selection

The clocks generated by the clock and reset generator are controlled by the Clock Generator Control Register at `0x3F200080`. Table 5-1 lists the format of the Clock Generator Control Register.

Table 5-1 Clock Generator Control Register

Bit	Description
[31]	Reserved, should be zero
[30]	Value of **HSYNCENIRW** and **SYNCENIRW** when the port is configured as asynchronous: 0 = **HSYNCENIRW** and **SYNCENIRW** signals are driven to 0 1 = **HSYNCENIRW** and **SYNCENIRW** signals are driven to 1.
[29]	Value of **HSYNCENPD** and **SYNCENPD** when the port is configured as synchronous: 0 = **HSYNCENPD** and **SYNCENPD** signals are driven to 0 1 = **HSYNCENPD** and **SYNCENPD** signals are driven to 1.
[28]	Selects the clock source for **HCLKI** and **HCLKE**: 0 = **HCLKI** and **HCLKE** derived from **PLLCLK** 1 = **HCLKI** and **HCLKE** derived from **HCLKIN**.
[27]	Configures Instruction, Data Read, and Data Write ports of the ARM1136JF-S test chip as synchronous or asynchronous: If Bit 27 is set to 0 the ports are configured as synchronous: **HCLKIRW** is driven from **CLK** **HCLKIRWEN** is driven from **HCLKIEN** **HSYNCENIRW** and **SYNCENIRW** are driven to 1. If Bit 27 is set to 1 the ports are configured as asynchronous: **HCLKIRW** is driven from **HCLKI** **HCLKIRWEN** is driven to 1 **HCLKIRW** is driven from **HCLKI** **HSYNCENIRW** and **SYNCENIRW** are driven from bit 30 of this register.

Table 5-1 Clock Generator Control Register (continued)

Bit	Description
[26]	Configures whether Peripheral and DMA ports of the ARM1136JF-S processor are configured as synchronous or asynchronous: If Bit 26 is set to 0 the ports are configured as synchronous: **HCLKPD** is driven from CLK **HCLKPDEN** is driven from **HCLKIEN** **HSYNCENPD** and **SYNCENPD** are driven to 1 If Bit 26 is set to 1 the ports are configured as asynchronous: **HCLKPD** is driven from **HCLKI** **HCLKPDEN** is driven to 1 **HSYNCENPD** and **SYNCENPD** are driven from bit 30 of this register.
[25:20]	Sets the clock ratio for the **HCLKE** domain.
[19:14]	Sets the clock ratio for the **HCLKI** domain.
[13:8]	Sets the clock ratio for the **CLK** domain.
[7:0]	Reserved, should be zero

The programmed clock ratios determine the divisor between the source clock and the output clock. The source clock for **CLK** is always **PLLCLK** (the signal **PLLBYPASS** must be LOW). The divisor selected is given by the following equation:

```
Divisor = Programmed ratio +1
```

The maximum divisor that you can program is 63.

Not all values of the Clock Generator Control Register are legal. The following restrictions apply:

1. **HCLKE** ratio must be an integer multiple of the **HCLKI**.
2. If you program the ports to be synchronous, that is if bit 27 is 0 and bit 26 is 0, you must program the **CLK** ratio to be 1:1.
3. If you select the clock source for **HCLKI** and **HCLKE** from **HCLKIN** (bit 28 is set to 1), then you must program the ports to be asynchronous (set bits 27 and 26 to 1). You must also set the **HCLKI** ratio must be 1:1.

Caution

Changes to bits [28:26] of this register only take effect during reset (**ARM_nRESET** is LOW). The changes to the clock ratios can take place at any point. You must exercise care to ensure that the programming of bits [28:26] have propagated to the clock

generator before selecting an asynchronous ratio. This means that the sequence for programming an asynchronous clock ratio involves setting bits [27:26], forcing a simple reset, and then programming the asynchronous clock ratio.

The clocks are disabled if **ARM_nPORESET** is LOW.

Changing any of the clock ratio values only takes effect during a reset. This is because clock ratio values are propagated from the register to the clock dividers during a reset only. This reset might be the same reset used to propagate changes in the asynchronous control bits, bits[28:26].

The minimum duration of an externally applied reset on the signal **ARM_nRESET** is one clock cycle of the slowest clock.

The Clock Divisor Register is accessed using the AHBPCAPT block.

The reset values of the Clock Generator Control Register are defined from the Configuration Register, see *Configuration Register* on page 5-21.

5.2.2 Synchronous/asynchronous clock control

Under normal operation, the clock ratios can be changed dynamically by accessing the Clock Generator Control Register at address `0x3F200080`. However, the synchronous/asynchronous mode of operation cannot be changed dynamically and requires the test chip to be reset.

During reset, predefined bits in the same format as the Test Chip Clock Generator are presented on **HRDATA**. This value is sampled on the rising edge of **CONFIGINIT**, when **nCONFIGRST** is HIGH and **ARM_nPORESET** is LOW, and written into the Test Chip Clock Generator Control Register. This programs the default clock ratios for the CP image from a power on reset

In asynchronous mode, **HCLKIN** is driven directly from **BUSCLK**. This enables you to drive core clock **REFCLK** independently from **HCLKIN**.

5.3 AHB memory map

This section describes the AHB interfaces present on the test chip.

5.3.1 AHB matrix and memories block

The AHB matrix and memories block provides a routing network for all memory accesses generated by the ARM1136JF-S ports. Memory accesses can be routed to:

- RAM implemented on the test chip
- memory mapped peripherals contained on the test chip
- the external AHB interface of the test chip.

The implementation of this block is constrained by the pinout of the test chip, and requires the external AHB interface to have a 32-bit data bus. The ARM1136JF-S processor has four 64-bit AHB-Lite master ports, and one 32-bit AHB-Lite master port. Table 5-2 describes the five AHB-Lite interfaces.

Table 5-2 AHB interfaces

Port	Type of port	Direction	Data bus width
ARM1136 Instruction Port	AHB-Lite	Input	64
ARM1136 Data Read Port	AHB-Lite	Input	64
ARM1136 Data Write Port	AHB-Lite	Output	64
ARM1136 DMA Port	AHB-Lite	I/O	64
ARM1136 Peripheral Port	AHB-Lite	I/O	32

The RAM implemented on the test chip is split into two parts:

- a large area of 64-bit wide RAM (the primary test chip RAM)
- a 8KB block of 32-bit wide RAM (the secondary test chip RAM).

These blocks, combined with the external AHB interface, enable accesses from any four ports to occur concurrently. Figure 5-2 on page 5-9 shows the AHB matrix and memories

Note

The **HADDRx** signal names in Figure 5-2 on page 5-9 indicate the different AHB buses.

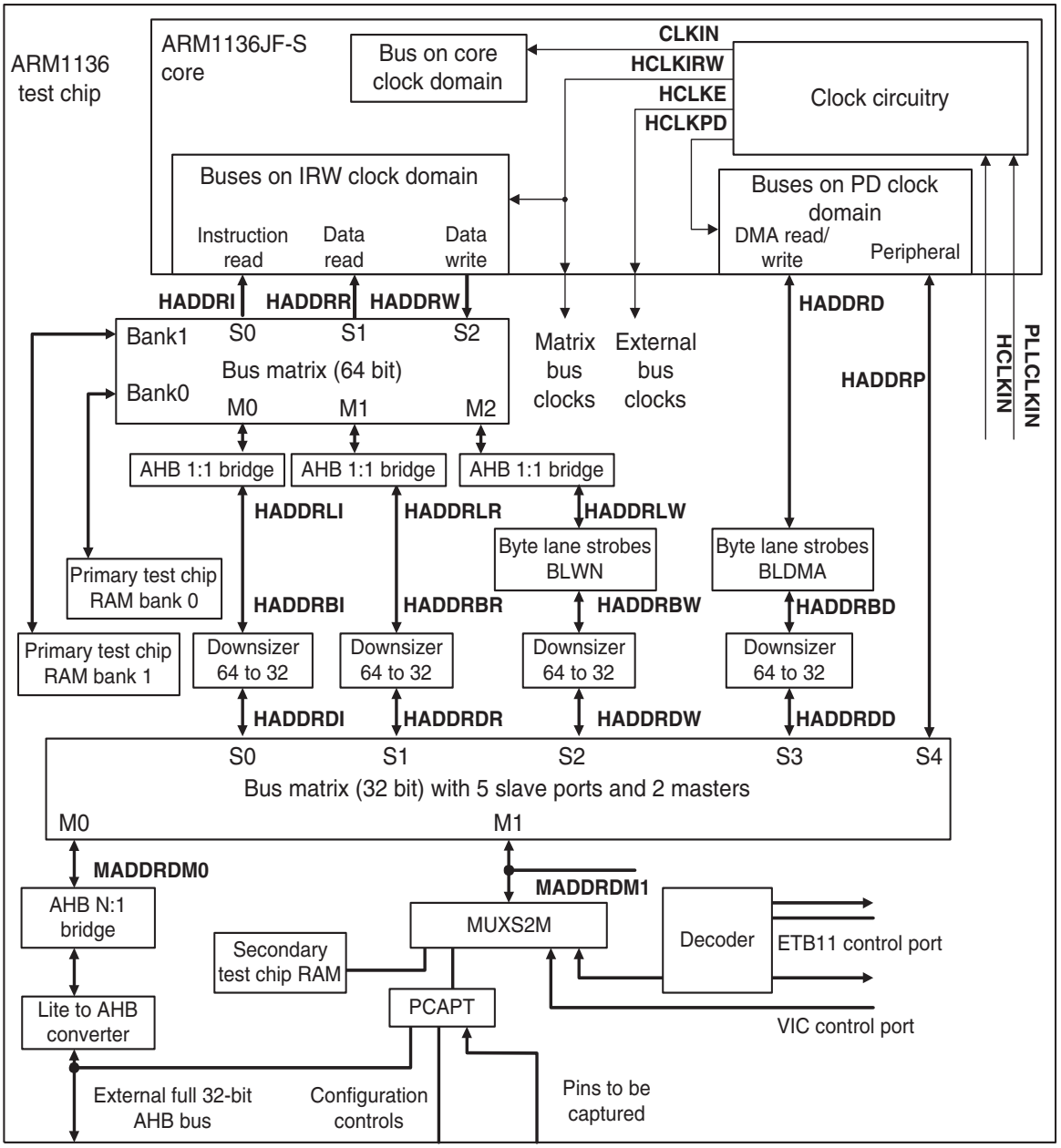

Figure 5-2 AHB matrix and bus organization

For details of components, see the *AMBA Design Kit Technical Reference Manual* and *AMBA Extension Infrastructure Blocks Technical Reference Manual*.

5.3.2 External AHB interface

All accesses that do not access the primary test chip RAM, the secondary test chip RAM, or the on-chip memory-mapped peripherals are presented on the test chip external 32-bit wide AHB interface. This is a full AHB interface provides the interface to the majority of the memory map of the test chip. The rest of the test chip memory infrastructure is implemented using AHB-Lite and multi-layer AHB. For more details on the memory map see the documentation for the motherboard or the application note covering the combination of Core Tile and motherboard you are using.

5.3.3 AHB address map

Table 5-3 lists the base address of the various ARM1136JF-S test chip AHB peripherals. These peripherals are multiply mapped throughout their allocated address ranges. The table also indicates the ARM1136JF-S AHB-Lite ports that are able to access each peripheral.

Table 5-3 Test chip address map

Module name	Address	Accessible from	Size	Description
XRAM2Kx32	`0x3F000000-0x3F0FFFFF`	All ports	1MB	Internal 32-bit secondary test chip RAM. 8KB RAM is typically implemented within this space.
AHBVIC	`0x3F100000-0x3F1FFFFF`	All ports	1MB	Control port for VIC.
AHBPCAPT	`0x3F200000-0x3F2FFFFF`	All ports	1MB	Pin capture and ETM11RV validation control.
Etb11TraceSRAM	`0x3F380000-0x3F3FFFFF`	All ports	512KB	ETB11 memory.
ETBREG	`0x3F300000-0x3F37FFFF`	All ports	512KB	ETB11 registers.

Table 5-3 Test chip address map (continued)

Module name	Address	Accessible from	Size	Description
SPARE	0x3F400000-0x3F7FFFFF	All ports	4MB	SPARE (access is presented on the test chip AHB interface).
TestChipRAM (BANK1)	0x3F800000-0x3FFFFFFF	Instruction read, Data Read, Data Write	4MB	Internal 64-bit primary test chip RAM bank. Interleaved on 1KB boundaries with BANK0. Actual size of this RAM depends on the space available within the test chip, but typically it is 256KB.
TestChipRAM (BANK0)	0x3F800000-0x3FFFFFFF	Instruction read, Data Read, Data Write	4MB	Internal 64-bit primary test chip RAM bank. Interleaved on 1KB boundaries with BANK1. Actual size of this RAM depends on the space available within the test chip, but typically it is 256KB.

For ARM test chips, accesses in the range 0x3FF00000–0x3FFFFFFF are normally reserved for on-chip registers and other accesses are presented on the test chip external 32-bit wide AHB interface

5.3.4 Block disables

Table 5-4 lists block disables controlled from the Configuration Register and defined for the memory block. The programming of these registers is described in *Configuration Register* on page 5-21.

Table 5-4 External control inputs

Signal	Description
BLKDISABL[5]	Disables the AHBRAM0 and AHBRAM1 primary test chip RAM blocks. Accesses are presented on the test chip AHB interface if this bit is configured HIGH.
BLKDISABL[4]	Disables the AHBETB Mem address range. Accesses are presented on the test chip AHB interface if this bit is configured HIGH.
BLKDISABL[3]	Disables the AHBETB Reg address range. The accesses are presented on the test chip AHB interface if this bit is configured HIGH.

Table 5-4 External control inputs (continued)

Signal	Description
BLKDISABL[2]	Disables the AHBPCAPT block. Accesses are presented on the test chip AHB interface if this bit is configured HIGH.
BLKDISABL[1]	Disables the AHBVIC block. Accesses are presented on the test chip AHB interface if this bit is configured HIGH.
BLKDISABL[0]	Disables the AHBRAM2 secondary test chip RAM block. Accesses are presented on the test chip AHB interface if this bit is configured HIGH.

5.3.5 AHB submodules

This section describes the submodules in the test chip.

Bus matrix 64

The 64-bit bus matrix, provides a connection between the Instruction Read, Data Read, and Data Write ports of the ARM1136JF-S core to the primary test chip RAM blocks. There are other connections to pass the accesses in parallel to the rest of the memory system. The 64-bit bus matrix consists of five slave ports, and three master ports.

The bus matrix 64 includes extensions to AHB for the use of byte lane strobes, providing support for **HUNALIGN** and **HBSTRB[n:0]**.

Bus matrix arbitration is fixed IR-DR-DW (highest to lowest priority).

Primary test chip RAMs

The two 64-bit wide RAMs are designed to be fast RAM blocks that can be accessed efficiently close to the core, enabling a high-performance memory system.

The three primary ports of the ARM1136JF-S core (the Instruction Read, Data Read, and Data Write ports) are each 64 bits wide, and have accesses to the two 64-bit wide primary test chip RAMs, and to other parts of the memory map. You cannot use the remaining two ports, the DMA port and Peripheral port, to access the primary test chip RAMs. The primary test chip RAMs exist to provide a relatively high capacity, high bandwidth, and low latency area of RAM that can be accessed by the primary ports. The primary RAMs are interleaved on 1KB boundaries.

AHB 1:1 bridges

The AHB 1:1 bridges connect to three of the master ports in the bus matrix 64, providing a cycle boundary between the two bus matrices, so avoiding an unacceptable critical path. These bridges include extensions to AHB for the use of byte lane strobes providing support for **HUNALIGN** and **HBSTRB[n:0]**.

Byte lane strobe converter

The byte lane strobe converter block takes in ARM11 AMBA extension signals that include **HUNALIGN** and **HBSTRB[n:0]**, and provide a mirror slave AHB port on the other side without **HBSTRB[n:0]** or **HUNALIGN**, that is AMBA v2.0 compliant signals. These blocks are inserted so that any slaves after these blocks do not have to support **HUNALIGN** and **HBSTRB[n:0]**.

The byte lane strobe converters are implemented only for ports that can do unaligned writes. These are the Data Write port and the DMA port.

Downsizer 64-bit to 32-bit

The downsizer converts an AMBA AHB v2.0 with 64-bit data bus to the same type of bus, but with a 32-bit data bus. It arranges for two accesses to be performed on the narrow side for each wide (64-bit) access.

Secondary test chip RAM

One of the AHB peripherals connected to the bus matrix 32 is a 32-bit SRAM to give a maximum of four parallel accesses being handled in the design. The secondary test chip RAM exists to increase the number of regions of RAM that can be accessed concurrently. It consists of a simple 32-bit 8KB block of RAM which is implemented in parallel with the on-chip peripherals.

Bus matrix 32

Bus matrix arbitration must be fixed IR-DR-DW-DMA-PERIPH (highest to lowest priority).

You can configure bus matrix blocks to have round-robin or fixed priority. The ARM1136JF-S test chip bus matrix (32-bit) is a fixed priority block. Input port 0 has the highest priority, and the priority decreases as the input port number increases.

AHB N:1 bridge

The AHB N:1 bridge provides the boundary between the **HCLKI** and the **HCLKE** clock domains. The clock ratio between **HCLKI** and **HCLKE** can be any integer value, including 1.

Lite2AHB converter

To convert from an AHB-Lite to full AHB functionality, a Lite2AHB converter block is used. This provides a full AHB port at the external AHB interface.

5.3.6 AHB slaves and support blocks

There are various AHB slaves connected to the bus matrix port M1. Standard AHB Decoder and AHB MuxS2M blocks are required to support the connection of multiple slaves to a single port. Table 5-5 lists the AHB slaves. See the *AMBA Specification (Rev 2.0)* for details on how to connect up an AMBA AHB system.

Table 5-5 AHB slaves

Component	Description
AHBVIC	Control port for the Vectored Interrupt Controller
AHBPCAPT	Pin capture block.
AHBRAM2	32-bit AHB SRAM. This RAM is 8KB.
AHBETB	ARM11 Embedded Trace Buffer (ETB11) Control.

The base address of these AHB slave ports is defined in *AHB address map* on page 5-10.

5.3.7 AHBPCAPT slave registers

Table 5-6 lists the addresses of the ARM1136JF-S AHBPCAPT Slave Registers.

Table 5-6 ARM1136JF-S AHBPCAPT Slave Registers

Address	Type	Register
0x3F200000	Read/write	ARM1136JF-S Pin Capture Register
0x3F200010	Read-only	Test Chip Memory Size Register
0x3F200020	Read-only	Instruction AHB Port Register 0
0x3F200024	Read-only	Instruction AHB Port Register 1
0x3F200028	Read-only	Instruction AHB Port Register 2
0x3F200030	Read-only	Data Read AHB Port Register 0
0x3F200034	Read-only	Data Read AHB Port Register 1
0x3F200038	Read-only	Data Read AHB Port Register 2
0x3F200040	Read-only	Data Write AHB Port Register 0
0x3F200044	Read-only	Data Write AHB Port Register 1
0x3F200048	Read-only	Data Write AHB Port Register 2
0x3F200050	Read-only	DMA AHB Port Register 0
0x3F200054	Read-only	DMA AHB Port Register 1
0x3F200058	Read-only	DMA AHB Port Register 2
0x3F200060	Read-only	Peripheral AHB Port0 Register
0x3F200064	Read-only	Peripheral AHB Port1 Register
0x3F200068	Read-only	Peripheral AHB Port2 Register
0x3F200080	Read/write	Clock Generator Control Register
0x3F2000C0	Read/write	Test Chip Control Register
0x3F200100	Read-only	ETM11RV Pin Capture1 Register
0x3F200104	Read-only	ETM11RV Pin Capture2 Register
0x3F200108	Read-only	ETM11RV Pin Capture3 Register
0x3F200110	Read/write	ETM11RV Pin Control1 Register

Table 5-6 ARM1136JF-S AHBPCAPT Slave Registers (continued)

Address	Type	Register
0x3F200114	Read/write	ETM11RV Pin Control2 Register
0x3F200118	Read/write	ETM11RV Pin Control3 Register
0x3F20011C	Read/write	ETM11RV Pin Control4 Register

Table 5-7 lists the Pin Capture Register bit allocation.

Table 5-7 Pin Capture Register bit allocation

Bit	Pins	Reset Value
[31:24]	Reserved, should be zero	Unpredictable
[23:16]	**DMAASID**	0
[15:8]	**COREASID**	0
[7:5]	Reserved, should be zero	Unpredictable
[4]	**DBGNOPWRDWN**	0
[3]	**CFGBIGENDPD**	0
[2]	**CFGBIGENDIRW**	0
[1]	**STANDBYWFI**	0
[0]	**Sticky STANDBYWFI**	Unpredictable

The **Sticky STANDBYWFI** bit is set if **STANDBYWFI** has been asserted in any cycle since the value was written by the core.

Table 5-8 lists the Test Chip Memory Size Register bit allocation.

Table 5-8 Test Chip Memory Size Register

Bit	Function
[31:5]	Unpredictable
[4:0]	Test chip RAM size

Table 5-9 lists the valid test chip RAM size values.

Table 5-9 Valid test chip RAM sizes

Hex	Size
0x07	64KB
0x08	128KB
0x09	256KB
0x0A	512KB
0x0B	1MB
0x0C	2MB
0x0D	4MB
0x0E	8MB

The AHB port registers read the values of the **HBSTRB[7:0]**, **HUNALIGN**, and **HSIDEBAND** signals to the last three non-sequential AHB transfers that have occurred for each port. The values are held as follows:

- The most recent transfer is held in Register 0
- The second most recent is held in Register 1
- The third most recent is held in Register 2.

The registers therefore act as FIFOs. Table 5-10 lists the contents of each AHB Port Register.

Table 5-10 Contents of each AHB Port Register

BIT	Function
[31:14]	Unpredictable
[13]	**WRITEBACK** (Data Write Port only. Other ports are UNP)
[12:9]	**HSIDEBAND[3:0]**
[8]	**HUNALIGN**
[7:0]	**HBSTRB[7:0]** for the Peripheral Port bits [7:4] are UNP

Table 5-11 lists the Test Chip Control Register bit allocations.

Table 5-11 Test Chip Control Register

Bit	Function	Reset value
[31:1]	Reserved, should be zero	Unpredictable
[0]	VIC enable	0

The VIC is disabled by default. This means that no programming of the VIC is required to connect **nFIQX** and **nIRQX** directly to the ARM1136JF-S core.

Table 5-12 lists the ETM11RV Pin Capture1 Register bit allocation.

Table 5-12 ETM11RV Pin Capture1 Register

Bit	Function	Reset value
[31:10]	Reserved, should be zero	Unpredictable
[9]	**nETMWFIREADY**	0
[8:6]	**ETMPORTMODE**	0
[5:2]	**ETMPORTSIZE**	0
[1]	**ETMEN**	0
[0]	**ETMPWRUP**	0

Table 5-13 lists the ETM11RV Pin Capture2 Register bit allocation.

Table 5-13 ETM11RV Pin Capture2 Register

Bit	Function	Reset value
[31:2]	Reserved, should be zero	Unpredictable
[1:0]	**ETMEXTOUTX**	Unpredictable

Table 5-14 lists the ETM11RV Pin Capture3 Register bit allocation.

Table 5-14 ETM11RV Pin Capture3 Register

Bit	Function	Reset value
[31:8]	Reserved, should be zero	Unpredictable
[7:0]	**ETMASICCTL**	Unpredictable

Table 5-15 lists the ETM11RV Pin Control1 Register bit allocation.

Table 5-15 ETM11RV Pin Control1 Register

Bit	Function	Reset value
[31:4]	Reserved, should be zero	Unpredictable
[3:0]	**EtmExtInRega**	`0`

Table 5-16 lists the ETM11RV Pin Control2 Register bit allocation.

Table 5-16 ETM11RV Pin Control2 Register

Bit	Function	Reset value
[31:11]	Reserved, should be zero	Unpredictable
[10]	ORed into **ETMWFIPENDING**	`0`
[9:6]	**ETMMAXPORTSIZE**	`b0100`
[5:3]	**ETMMAXEXTOUT**	`b010`
[2:0]	**ETMMAXEXTIN**	`0`

Table 5-17 lists the ETM11RV Pin Control3 Register bit allocation.

Table 5-17 ETM11RV Pin Control3 Register

Bit	Function	Reset value
[31:20]	Reserved, should be zero	Unpredictable
[19:0]	**EtmEvntBusRega**	`0`

Table 5-18 lists the ETM11RV Pin Control4 Register bit allocation.

Table 5-18 ETM11RV Pin Control4 Register

Bit	Function	Reset value
[31:6]	Reserved, should be zero	Unpredictable
[5:0]	**EtmInputSel**	0

Table 5-19 lists the EtmInputSel[1:0] encoding used to determine the source of **ETMEXTINX**.

Table 5-19 EtmInputSel[1:0] encoding

EtmInputSel[1:0]	ETMEXTINX to ETM
b00	0
b01	EtmExtInReg
b10	(**DBGACKX**, PseudoRandom[2:0])
b11	((b000, **ETMEXTINX** (test chip))

Table 5-20 lists the EtmInputSel[3:2] encoding used to determine the source of **ETMEVNTBUS**

Table 5-20 EtmInputSel[3:2] encoding

EtmInputSel[3:2]	ETMEVNTBUS to ETM
b00	0
b01	EtmEvntBusReg
b10	PseudoRandom[19:0]
b11	**ETMEVNTBUS** from ARM1136JF-S

Table 5-21 lists the EtmInputSel[5:4] encoding used to determine the source of **ETMTRACEREADY**

Table 5-21 EtmInputSel[5:4]

EtmInputSel[5:4]	ETMEVNTBUS to ETM
b00	1
b01	PseudoRandom[0]
b10	0
b11	1 whenever an instruction is executed.

Configuration Register

The AHBPCAPT block contains the configuration register space. This space is programmed when the test chip is held in its lowest level of reset (that is, when **ARM_nPORESET** is asserted). The Configuration Register is a 32-bit register that is programmed to the value on **HRDATAX** signals at the rising edge of **CONFIGINIT**.

The Configuration Register is reset by asserting **nCONFIGRST**, and only when **ARM_nPORESETX** is asserted. The Configuration Register contents cannot be modified from software. Use the Clock Generator Control Register to change the clock frequency after power on.

Table 5-22 lists the bit allocation of the Configuration Register.

Table 5-22 Configuration Register

Bit	Function	Reset value (nCONFIGRST)
[31]	Reserved, should be zero	0
[30:8]	Reset value of bits [28:8] of the Clock Generator Control Register	0
[7:2]	Test chip memory block disables	0
[1:0]	Reserved, should be zero	0

5.4 Vectored Interrupt Controller (VIC) block

The VIC has the following interfaces:

- interrupt sources
- AHB slave interface to control the VIC
- daisy chain interface to connect two or more VICs together
- scan chain interface.

There are 32 interrupt source inputs. For details of the VIC interrupts sources connections, see the *ARM PrimeCell Vectored Interrupt Controller (PL192) Technical Reference Manual*.

Note

The daisy chain interface is tied off, because only a single VIC is present in the ARM1136JF-S test chip. By default, the VIC is disabled. Use the Test Chip Control Register to enable the VIC (see Table 5-11 on page 5-18).

Table 5-23 lists the ARM1136JF-S test chip signal routing to the VIC interrupt sources.

Table 5-23 Test chip interrupt routing

VIC input	Source	Comment
VICINTSOURCE[0]	(Inverted) **nFIQ**	Pin on the test chip
VICINTSOURCE[1]	(Inverted) **nVALFIQ**	ARM1136JF-S output
VICINTSOURCE[6]	**COMMRX**	ARM1136 JF-S output
VICINTSOURCE[7]	**COMMTX**	ARM1136 JF-S output
VICINTSOURCE[10]	(Inverted) **nIRQ**	Pin on the test chip
VICINTSOURCE[11]	(Inverted) **nVALIRQ**	ARM1136 JF-S output
VICINTSOURCE[12]	(Inverted) **nDMAIRQ**	ARM1136 JF-S output
VICINTSOURCE[13]	(Inverted) **nPMUIRQ**	ARM1136 JF-S output
VICINTSOURCE[14]	(Inverted) **nVALRESET**	ARM1136 JF-S output

Note

All other **VICINTSOURCE** signals are tied to 0.

5.5 Voltage control

The CT1136JF-S Core Tile voltage connections are listed in Table 5-24.

Table 5-24 Supply voltage connections

Voltage	Source	Description
ARM_VDDCORE1, **ARM_VDDCORE2**, **ARM_VDDCORE3**, **ARM_VDDCORE4**, **ARM_VDDCORE5**, and **ARM_VDDCORE6**	VDDCOREA regulator	All core voltages are supplied from the VDDCOREA regulator.
VDDIO	3.3V	The I/O voltage is supplied from the HDRZ 3.3V power blade.
VDDPLL1	VDDPLL regulator	Supplied from the dedicated PLL regulator.
VDDPLL2	VDDPLL regulator	Supplied from the dedicated PLL regulator.

Chapter 6
HBI-0131 Signal Descriptions

This chapter provides a summary of signals present on the CT926EJ-S and CT1136JF-S Core Tile connectors and test points and the links that can be modified to change signal routing. It contains the following sections:

Note

For the Multi-ICE connector pinout and signal descriptions see *JTAG signals* on page 3-39.

6.1 Header connectors

Figure 6-1 shows the pin numbers and power-blade usage of the HDRX, HDRY, and HDRZ headers on the upper side of the tile. For details on the **VDDIO** and **VCCOY** voltages, see *Power supply control* on page 3-17.

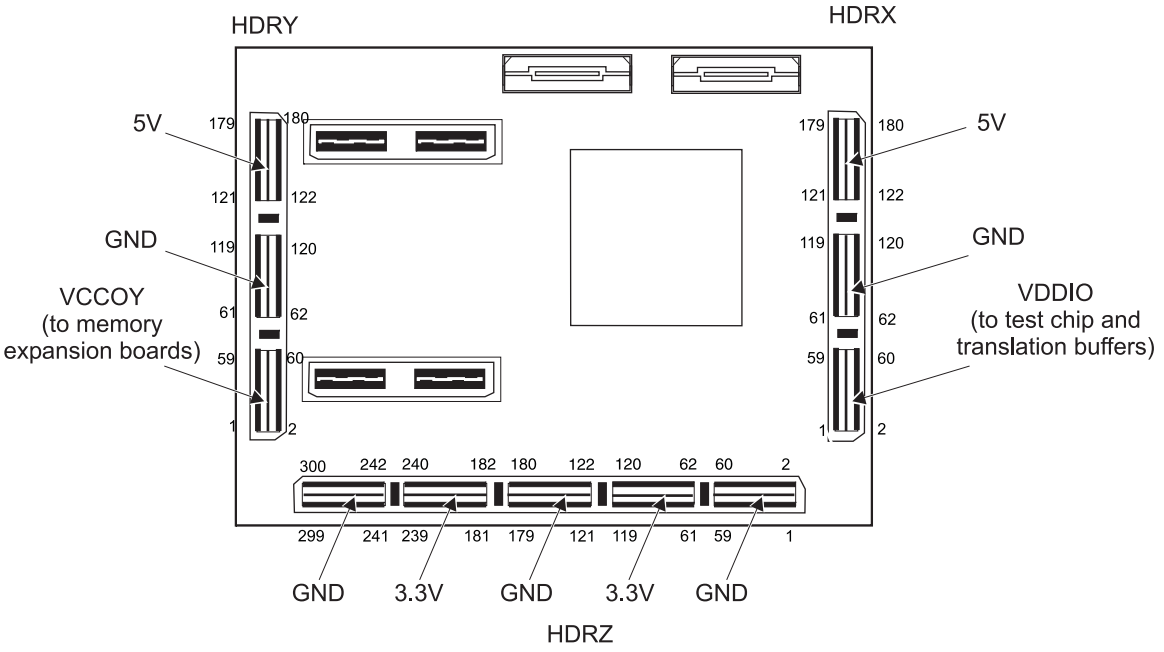

Figure 6-1 HDRX, HDRY, and HDRZ (upper) pin numbering

Table 6-1 on page 6-3 lists the Samtec part numbers.

Core Tiles use the -05- connectors on the top of the board and the -01- connectors on the bottom of the board. The total separation is 22mm. The Core Tiles have a maximum component height of 2.5mm on the bottom and 19mm on the top of the board. This ensures that there are no component interference problems with mated boards.

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 6-1 Samtec part numbers

Header	Part number	Mating height
HDRX (upper)	QTH-090-05-F-D-A-K	19mm
HDRXL (lower)	QSH-090-01-F-D-A-K	5mm
HDRY (upper)	QTH-090-05-F-D-A-K	19mm
HDRYL (lower)	QSH-090-01-F-D-A-K	5mm
HDRZ (upper)	QTH-150-05-F-D-A-K	19mm
HDRZL (lower)	QSH-150-01-F-D-A-K	5mm

6.1.1 HDRX signals

Table 6-3 on page 6-4 describes the signals on the HDRX pins for the Core Tile and Logic Tile. (Replace *x* by `L` for the lower header and by `U` for the upper header for Logic Tile signals.)

Note

Except for the **HCLK**, **REFCLK**, **HCLKIN** signals, all of the signals on the upper and lower HDRX connectors are the same for the Core Tile.

Most of the pins on the RealView Logic Tiles have different signals on the upper and lower Logic Tile HDRZ pins. If the upper signal on the Logic Tile is named `XU`*n* and the lower signal is named `XL`*n*, the table entry is `X`*xn*. Replace *x* by `L` for the signal on the lower header and by `U` for the signal on the upper header. For example, the pin 1 signal for the lower connector is **XL90** and the pin 1 signal for the upper connector is **XU90**.

There the following differences between the CT926EJ-S and CT1136JF-S header signals:

Table 6-2 Signal differences between CT926EJ-S and 1136JF-S

Signal	CT926EJ-S	CT1136JF-S
USERIN[2:0]	Reserved, should be LOW	Reserved, should be LOW
USERIN[3]	Manufacturing test for VIC and PLL bypass, should be LOW	Reserved, should be LOW
USERIN[4]	Manufacturing test, should be LOW	**UBITINTX**, Initial value of ARM1136 U bit
USERIN[5]	Manufacturing test, should be LOW	**HRESPX[2]**, AHB response signal
USEROUT[3:0]	Reserved output, do not drive	Reserved output, do not drive
USEROUT[5:4]	Reserved output, do not drive	HPROTX[5:4], AHB protection information

Table 6-3 HDRX signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
PLLCTRL0	**Xx90**	1	2	**Xx89**	**USEROUT5** (see Table 6-2)
PLLCTRL1	**Xx91**	3	4	**Xx88**	**USEROUT4**
PLLREFDIV0	**Xx92**	5	6	**Xx87**	**USEROUT3**
PLLREFDIV1	**Xx93**	7	8	**Xx86**	**USEROUT2**
PLLREFDIV2	**Xx94**	9	10	**Xx85**	**USEROUT1**
PLLREFDIV3	**Xx95**	11	12	**Xx84**	**USEROUT0**
HADDR0	**Xx96**	13	14	**Xx83**	**USERIN5**
HADDR1	**Xx97**	15	16	**Xx82**	**USERIN4**
HADDR2	**Xx98**	17	18	**Xx81**	**USERIN3**
HADDR3	**Xx99**	19	20	**Xx80**	**USERIN2**
HADDR4	**Xx100**	21	22	**Xx79**	**USERIN1**
HADDR5	**Xx101**	23	24	**Xx78**	**USERIN0**

Table 6-3 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
HADDR6	**Xx102**	25	26	**Xx77**	**IEBKPT**
HADDR7	**Xx103**	27	28	**Xx76**	**DEWPT**
HADDR8	**Xx104**	29	30	**Xx75**	**EDBGRQ**
HADDR9	**Xx105**	31	32	**Xx74**	**HCLKDIV2**
HADDR10	**Xx106**	33	34	**Xx73**	**HCLKDIV1**
HADDR11	**Xx107**	35	36	**Xx72**	**HCLKDIV0**
HADDR12	**Xx108**	37	38	**Xx71**	**INITRAM**
HADDR13	**Xx109**	39	40	**Xx70**	**PLLBYPASS**
HADDR14	**Xx110**	41	42	**Xx69**	**PLLLOCK**
HADDR15	**Xx111**	43	44	**Xx68**	**VINITHI**
HADDR16	**Xx112**	45	46	**Xx67**	**ETMEXTOUT**
HADDR17	**Xx113**	47	48	**Xx66**	**ETMEXTIN**
HADDR18	**Xx114**	49	50	**Xx65**	**NCONFIGRST**
HADDR19	**Xx115**	51	52	**Xx64**	**ARM_NPORESET**
HADDR20	**Xx116**	53	54	**Xx63**	**ARM_NRESET**
HADDR21	**Xx117**	55	56	**Xx62**	**NIRQ**
HADDR22	**Xx118**	57	58	**Xx61**	**NFIQ**
HADDR23	**Xx119**	59	60	**Xx60**	**DBGEN**
HADDR24	**Xx120**	61	62	**Xx59**	**DBGACK**
HADDR25	**Xx121**	63	64	**Xx58**	**CONFIGINIT**
HADDR26	**Xx122**	65	66	**Xx57**	**COMMTX**
HADDR27	**Xx123**	67	68	**Xx56**	**COMMRX**
HADDR28	**Xx124**	69	70	**Xx55**	**BIGENDOUT**
HADDR29	**Xx125**	71	72	**Xx54**	**BIGENDIN**

Table 6-3 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
HADDR30	**Xx126**	73	74	**Xx53**	**HRESP1**
HADDR31	**Xx127**	75	76	**Xx52**	**HRESP0**
TESTSELECT	**Xx128**	77	78	**Xx51**	**HGRANT**
TICSELECT	**Xx129**	79	80	**Xx50**	**HREADY**
PLLFBDIV0	**Xx130**	81	82	**Xx49**	**HLOCK**
PLLFBDIV1	**Xx131**	83	84	**Xx48**	**HBUSREQ**
PLLFBDIV2	**Xx132**	85	86	**Xx47**	**HWRITE**
PLLFBDIV3	**Xx133**	87	88	**Xx46**	**HBURST2**
PLLFBDIV4	**Xx134**	89	90	**Xx45**	**HBURST1**
PLLFBDIV5	**Xx135**	91	92	**Xx44**	**HBURST0**
PLLFBDIV6	**Xx136**	93	94	**Xx43**	**HPROT3**
PLLFBDIV7	**Xx137**	95	96	**Xx42**	**HPROT2**
PLLOUTDIV0	**Xx138**	97	98	**Xx41**	**HPROT1**
PLLOUTDIV1	**Xx139**	99	100	**Xx40**	**HPROT0**
PLLOUTDIV2	**Xx140**	101	102	**Xx39**	**HSIZE1**
PLLOUTDIV3	**Xx141**	103	104	**Xx38**	**HSIZE0**
EXTTRIG	**Xx142**	105	106	**Xx37**	**HTRANS1**
X143	**Xx143**	107	108	**Xx36**	**HTRANS0**
X144	**Xx144**	109	110	**Xx35**	**HSIZE2**
X145	**Xx145**	111	112 (U) 112 (L)	**Xx34**	**X_HCLK_UPX_HCLK_DN**
X146	**Xx146**	113	114 (U) 114 (L)	**Xx33**	**X_REFCLK_DN** **X_REFCLK_UP**
X147	**Xx147**	115	116 (U) 116 (L)	**Xx32**	**X_HCLKIN_DNX_HCLKIN_UP**
HDATA0	**Xx148**	117	118	**Xx31**	**HRDATA31**

Table 6-3 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
HDATA1	**Xx149**	119	120	**Xx30**	**HRDATA30**
HDATA2	**Xx150**	121	122	**Xx29**	**HRDATA29**
HDATA3	**Xx151**	123	124	**Xx28**	**HRDATA28**
HDATA4	**Xx152**	125	126	**Xx27**	**HRDATA27**
HDATA5	**Xx153**	127	128	**Xx26**	**HRDATA26**
HDATA6	**Xx154**	129	130	**Xx25**	**HRDATA25**
HDATA7	**Xx155**	131	132	**Xx24**	**HRDATA24**
HDATA8	**Xx156**	133	134	**Xx23**	**HRDATA23**
HDATA9	**Xx157**	135	136	**Xx22**	**HRDATA22**
HDATA10	**Xx158**	137	138	**Xx21**	**HRDATA21**
HDATA11	**Xx159**	139	140	**Xx20**	**HRDATA20**
HDATA12	**Xx160**	141	142	**Xx19**	**HRDATA19**
HDATA13	**Xx161**	143	144	**Xx18**	**HRDATA18**
HDATA14	**Xx162**	145	146	**Xx17**	**HRDATA17**
HDATA15	**Xx163**	147	148	**Xx16**	**HRDATA16**
HDATA16	**Xx164**	149	150	**Xx15**	**HRDATA15**
HDATA17	**Xx165**	151	152	**Xx14**	**HRDATA14**
HDATA18	**Xx166**	153	154	**Xx13**	**HRDATA13**
HDATA19	**Xx167**	155	156	**Xx12**	**HRDATA12**
HDATA20	**Xx168**	157	158	**Xx11**	**HRDATA11**
HDATA21	**Xx169**	159	160	**Xx10**	**HRDATA10**
HDATA22	**Xx170**	161	162	**Xx9**	**HRDATA9**
HDATA23	**Xx171**	163	164	**Xx8**	**HRDATA8**
HDATA24	**Xx172**	165	166	**Xx7**	**HRDATA7**

Table 6-3 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
HDATA25	**Xx173**	167	168	**Xx6**	**HRDATA6**
HDATA26	**Xx174**	169	170	**Xx5**	**HRDATA5**
HDATA27	**Xx175**	171	172	**Xx4**	**HRDATA4**
HDATA28	**Xx176**	173	174	**Xx3**	**HRDATA3**
HDATA29	**Xx177**	175	176	**Xx2**	**HRDATA2**
HDATA30	**Xx178**	177	178	**Xx1**	**HRDATA1**
HDATA31	**Xx179**	179	180	**Xx0**	**HRDATA0**

6.1.2 HDRY signals

Table 6-4 describes the signals on the HDRY pins for the Core Tile and Logic Tile.

Note

All signals on the upper and lower HDRY connectors are the same for the Core Tile.

For Logic Tile signals, most of the pins on the upper and lower connector have different signals. If the upper signal on the Logic Tile is named `YU`*n* and the lower signal is named `YL`*n*, the table entry is `Y`*xn*. Replace *x* by `L` for the signal on the lower header and by `U` for the signal on the upper header. For example, the pin 1 signal for the lower connector is **YL89** and the pin 1 signal for the upper connector is **YU89** and the corresponding Core Tile signal is **MEMEXPA25**.

Table 6-4 HDRY signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA25	**Yx89**	1	2	**Yx90**	**MEMEXPA26**
MEMEXPA24	**Yx88**	3	4	**Yx91**	**MEMEXPA27**
MEMEXPA23	**Yx87**	5	6	**Yx92**	**MEMEXPA28**
MEMEXPA22	**Yx86**	7	8	**Yx93**	**MEMEXPA29**

Table 6-4 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA21	**Yx85**	9	10	**Yx94**	**MEMEXPA30**
MEMEXPA20	**Yx84**	11	12	**Yx95**	**MEMEXPA31**
MEMEXPA19	**Yx83**	13	14	**Yx96**	**MEMEXPA32**
MEMEXPA18	**Yx82**	15	16	**Yx97**	**MEMEXPA33**
MEMEXPA17	**Yx81**	17	18	**Yx98**	**MEMEXPA34**
MEMEXPA16	**Yx80**	19	20	**Yx99**	**MEMEXPA35**
MEMEXPA15	**Yx79**	21	22	**Yx100**	**MEMEXPA36**
MEMEXPA14	**Yx78**	23	24	**Yx101**	**MEMEXPA37**
MEMEXPA13	**Yx77**	25	26	**Yx102**	**MEMEXPA38**
MEMEXPA12	**Yx76**	27	28	**Yx103**	**MEMEXPA39**
MEMEXPA11	**Yx75**	29	30	**Yx104**	**MEMEXPA40**
MEMEXPA10	**Yx74**	31	32	**Yx105**	**MEMEXPA41**
MEMEXPA9	**Yx73**	33	34	**Yx106**	**MEMEXPA42**
MEMEXPA8	**Yx72**	35	36	**Yx107**	**MEMEXPA43**
MEMEXPA7	**Yx71**	37	38	**Yx108**	**MEMEXPA44**
MEMEXPA6	**Yx70**	39	40	**Yx109**	**MEMEXPA45**
MEMEXPA5	**Yx69**	41	42	**Yx110**	**MEMEXPA46**
MEMEXPA4	**Yx68**	43	44	**Yx111**	**MEMEXPA47**
MEMEXPA3	**Yx67**	45	46	**Yx112**	**MEMEXPA48**
MEMEXPA2	**Yx66**	47	48	**Yx113**	**MEMEXPA49**
MEMEXPA1	**Yx65**	49	50	**Yx114**	**MEMEXPA50**
MEMEXPA0	**Yx64**	51	52	**Yx115**	**MEMEXPA51**
MEMEXPA103	**Yx63**	53	54	**Yx116**	**MEMEXPA52**
MEMEXPA102	**Yx62**	55	56	**Yx117**	**MEMEXPA53**

Table 6-4 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA101	**Yx61**	57	58	**Yx118**	**MEMEXPA54**
MEMEXPA100	**Yx60**	59	60	**Yx119**	**MEMEXPA55**
MEMEXPA99	**Yx59**	61	62	**Yx120**	**MEMEXPA56**
MEMEXPA98	**Yx58**	63	64	**Yx121**	**MEMEXPA57**
MEMEXPA97	**Yx57**	65	66	**Yx122**	**MEMEXPA58**
MEMEXPA96	**Yx56**	67	68	**Yx123**	**MEMEXPA59**
MEMEXPA95	**Yx55**	69	70	**Yx124**	**MEMEXPA60**
MEMEXPA94	**Yx54**	71	72	**Yx125**	**MEMEXPA61**
MEMEXPA93	**Yx53**	73	74	**Yx126**	**MEMEXPA62**
MEMEXPA92	**Yx52**	75	76	**Yx127**	**MEMEXPA63**
MEMEXPA91	**Yx51**	77	78	**Yx128**	**MEMEXPA64**
MEMEXPA90	**Yx50**	79	80	**Yx129**	**MEMEXPA65**
MEMEXPA89	**Yx49**	81	82	**Yx130**	**MEMEXPA66**
MEMEXPA88	**Yx48**	83	84	**Yx131**	**MEMEXPA67**
MEMEXPA87	**Yx47**	85	86	**Yx132**	**MEMEXPA68**
MEMEXPA86	**Yx46**	87	88	**Yx133**	**MEMEXPA69**
MEMEXPA85	**Yx45**	89	90	**Yx134**	**MEMEXPA70**
MEMEXPA84	**Yx44**	91	92	**Yx135**	**MEMEXPA71**
MEMEXPA83	**Yx43**	93	94	**Yx136**	**MEMEXPA72**
MEMEXPA82	**Yx42**	95	96	**Yx137**	**MEMEXPA73**
MEMEXPA81	**Yx41**	97	98	**Yx138**	**MEMEXPA74**
MEMEXPA80	**Yx40**	99	100	**Yx139**	**MEMEXPA75**
Y39	**Yx39**	101	102	**Yx140**	**MEMEXPA76**
Y38	**Yx38**	103	104	**Yx141**	**MEMEXPA77**

Table 6-4 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Y37	**Yx37**	105	106	**Yx142**	**MEMEXPA78**
Y36	**Yx36**	107	108	**Yx143**	**MEMEXPA79**
Y35	**Yx35**	109	110	**Yx144**	**Y144**
Y34	**Yx34**	111	112	**Yx145**	**Y145**
Y33	**Yx33**	113	114	**Yx146**	**Y146**
Y32	**Yx32**	115	116	**Yx147**	**Y147**
Y31	**Yx31**	117	118	**Yx148**	**Y148**
Y30	**Yx30**	119	120	**Yx149**	**Y149**
Y29	**Yx29**	121	122	**Yx150**	**Y150**
Y28	**Yx28**	123	124	**Yx151**	**Y151**
Y27	**Yx27**	125	126	**Yx152**	**Y152**
Y26	**Yx26**	127	128	**Yx153**	**Y153**
Y25	**Yx25**	129	130	**Yx154**	**Y154**
Y24	**Yx24**	131	132	**Yx155**	**Y155**
Y23	**Yx23**	133	134	**Yx156**	**Y156**
Y22	**Yx22**	135	136	**Yx157**	**Y157**
Y21	**Yx21**	137	138	**Yx158**	**Y158**
Y20	**Yx20**	139	140	**Yx159**	**Y159**
Y19	**Yx19**	141	142	**Yx160**	**Y160**
Y18	**Yx18**	143	144	**Yx161**	**Y161**
Y17	**Yx17**	145	146	**Yx162**	**Y162**
Y16	**Yx16**	147	148	**Yx163**	**Y163**
Y15	**Yx15**	149	150	**Yx164**	**Y164**
Y14	**Yx14**	151	152	**Yx165**	**Y165**

Table 6-4 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Y13	**Yx13**	153	154	**Yx166**	**Y166**
Y12	**Yx12**	155	156	**Yx167**	**Y167**
Y11	**Yx11**	157	158	**Yx168**	**Y168**
Y10	**Yx10**	159	160	**Yx169**	**Y169**
Y9	**Yx9**	161	162	**Yx170**	**Y160**
Y8	**Yx8**	163	164	**Yx171**	**Y171**
Y7	**Yx7**	165	166	**Yx172**	**Y172**
Y6	**Yx6**	167	168	**Yx173**	**Y173**
Y5	**Yx5**	169	170	**Yx174**	**Y174**
Y4	**Yx4**	171	172	**Yx175**	**Y175**
Y3	**Yx3**	173	174	**Yx176**	**Y176**
Y2	**Yx2**	175	176	**Yx177**	**Y177**
Y1	**Yx1**	177	178	**Yx178**	**Y178**
Y0	**Yx0**	179	180	**Yx179**	**Y179**

6.1.3 HDRZ

Table 6-5 on page 6-13 describes the signals on the HDRZ pins for the Core Tile and the RealView Logic Tile.

Note

Except for the clock and JTAG signals, the Core Tile has the same signal on both the upper and lower pins. Some of the upper and lower HDRZ signals can be isolated, see *Control of AHB data bus and HDRZ signals* on page 3-26.

Most of the pins on the RealView Logic Tiles have different signals on the upper and lower Logic Tile HDRZ pins. If the upper signal on the Logic Tile is named ZU*n* and the lower signal is named ZL*n*, the table entry is Z*xn*. Replace *x* by L for the signal on the lower header and by U for the signal on the upper header. For example, the pin 1 signal for the lower connector is **ZL128** and the pin 1 signal for the upper connector is **ZU128**.

If the upper signal on the Logic Tile has a completely different name than the lower signal, both names are listed and the signal on the upper connector is indicated by a (U) and the signal on the lower connector is indicated by a (L).

Table 6-5 HDRZ signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB0	**Zx128**	1	2	**Zx255**	**Z255**
MEMEXPB1	**Zx129**	3	4	**Zx254**	**Z254**
MEMEXPB2	**Zx130**	5	6	**Zx253**	**Z253**
MEMEXPB3	**Zx131**	7	8	**Zx252**	**Z252**
MEMEXPB4	**Zx132**	9	10	**Zx251**	**Z251**
MEMEXPB5	**Zx133**	11	12	**Zx250**	**Z250**
MEMEXPB6	**Zx134**	13	14	**Zx249**	**Z249**
MEMEXPB7	**Zx135**	15	16	**Zx248**	**Z248**
MEMEXPB8	**Zx136**	17	18	**Zx247**	**Z247**
MEMEXPB9	**Zx137**	19	20	**Zx246**	**Z246**
MEMEXPB10	**Zx138**	21	22	**Zx245**	**Z245**
MEMEXPB11	**Zx139**	23	24	**Zx244**	**Z244**
MEMEXPB12	**Zx140**	25	26	**Zx243**	**Z243**
MEMEXPB13	**Zx141**	27	28	**Zx242**	**Z242**
MEMEXPB14	**Zx142**	29	30	**Zx241**	**Z241**
MEMEXPB15	**Zx143**	31	32	**Zx240**	**Z240**
MEMEXPB16	**Zx144**	33	34	**Zx249**	**Z239**
MEMEXPB17	**Zx145**	35	36	**Zx248**	**Z238**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB18	**Zx146**	37	38	**Zx237**	**Z237**
MEMEXPB19	**Zx147**	39	40	**Zx236**	**Z236**
MEMEXPB20	**Zx148**	41	42	**Zx235**	**Z235**
MEMEXPB21	**Zx149**	43	44	**Zx234**	**Z234**
MEMEXPB22	**Zx150**	45	46	**Zx233**	**Z233**
MEMEXPB23	**Zx151**	47	48	**Zx232**	**Z232**
MEMEXPB24	**Zx152**	49	50	**Zx231**	**MEMEXPB103**
MEMEXPB25	**Zx153**	51	52	**Zx230**	**PLDCLK**
MEMEXPB26	**Zx154**	53	54	**Zx229**	**PLDRESETN**
MEMEXPB27	**Zx155**	55	56	**Zx228**	**PLDD0**
MEMEXPB28	**Zx156**	57	58	**Zx227**	**PLDD1**
MEMEXPB29	**Zx157**	59	60	**Zx226**	**MEMEXPB98**
MEMEXPB30	**Zx158**	61	62	**Zx225**	**MEMEXPB97**
MEMEXPB31	**Zx159**	63	64	**Zx224**	**MEMEXPB96**
MEMEXPB32	**Zx160**	65	66	**Zx223**	**MEMEXPB95**
MEMEXPB33	**Zx161**	67	68	**Zx222**	**MEMEXPB94**
MEMEXPB34	**Zx162**	69	70	**Zx221**	**MEMEXPB93**
MEMEXPB35	**Zx163**	71	72	**Zx220**	**MEMEXPB92**
MEMEXPB36	**Zx164**	73	74	**Zx219**	**MEMEXPB91**
MEMEXPB37	**Zx165**	75	76	**Zx218**	**MEMEXPB90**
MEMEXPB38	**Zx166**	77	78	**Zx217**	**MEMEXPB89**
MEMEXPB39	**Zx167**	79	80	**Zx216**	**MEMEXPB88**
MEMEXPB40	**Zx168**	81	82	**Zx215**	**MEMEXPB87**
MEMEXPB41	**Zx169**	83	84	**Zx214**	**MEMEXPB86**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB42	**Zx170**	85	86	**Zx213**	**MEMEXPB85**
MEMEXPB43	**Zx171**	87	88	**Zx212**	**MEMEXPB84**
MEMEXPB44	**Zx172**	89	90	**Zx211**	**MEMEXPB83**
MEMEXPB45	**Zx173**	91	92	**Zx210**	**MEMEXPB82**
MEMEXPB46	**Zx174**	93	94	**Zx209**	**MEMEXPB81**
MEMEXPB47	**Zx175**	95	96	**Zx208**	**MEMEXPB80**
MEMEXPB48	**Zx176**	97	98	**Zx207**	**MEMEXPB79**
MEMEXPB49	**Zx177**	99	100	**Zx206**	**MEMEXPB78**
MEMEXPB50	**Zx178**	101	102	**Zx205**	**MEMEXPB77**
MEMEXPB51	**Zx179**	103	104	**Zx204**	**MEMEXPB76**
MEMEXPB52	**Zx180**	105	106	**Zx203**	**MEMEXPB75**
MEMEXPB53	**Zx181**	107	108	**Zx202**	**MEMEXPB74**
MEMEXPB54	**Zx182**	109	110	**Zx201**	**MEMEXPB73**
MEMEXPB55	**Zx183**	111	112	**Zx200**	**MEMEXPB72**
MEMEXPB56	**Zx184**	113	114	**Zx199**	**MEMEXPB71**
MEMEXPB57	**Zx185**	115	116	**Zx198**	**MEMEXPB70**
MEMEXPB58	**Zx186**	117	118	**Zx197**	**MEMEXPB69**
MEMEXPB59	**Zx187**	119	120	**Zx196**	**MEMEXPB68**
MEMEXPB60	**Zx188**	121	122	**Zx195**	**MEMEXPB67**
MEMEXPB61	**Zx189**	123	124	**Zx194**	**MEMEXPB66**
MEMEXPB62	**Zx190**	125	126	**Zx193**	**MEMEXPB65**
MEMEXPB63	**Zx191**	127	128	**Zx192**	**MEMEXPB64**
D_nSRST	**D_nSRST**	129	130 (U) 130(L)	**CLK_POS_DN_INC** **LK_POS_DN_OUT**	**CLK_POS_DN**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
D_nTRST	**D_nTRST**	131	132 (U) 132 (L)	**CLK_NEG_DN_INC** **LK_NEG_DN_OUT**	**CLK_NEG_DN_INC** **LK_NEG_DN_OUT**
D_TDO_IND_TD **O_OUT**	**D_TDO_IN** **D_TDO_OUT**	133 (U) 133 (L)	134 (U) 134 (L)	**CLK_POS_UP_OUT** **CLK_POS_UP_IN**	**CLK_POS_UP**
D_TDI	**D_TDI**	135	136 (U) 136 (L)	**CLK_NEG_UP_OUT** **CLK_NEG_UP_IN**	**CLK_NEG_UP_OUT** **CLK_NEG_UP_IN**
D_TCK_OUTD_ **TCK_IN**	**D_TCK_OUT** **D_TCK_IN**	137 (U) 137 (L)	138 (U) 138 (L)	**CLK_UP_THRU** **CLK_IN_MINUS2**	**CLK_IN_MINUS2**
D_TMS_OUTD_ **TMS_IN**	**D_TMS_OUTD_** **TMS_IN**	139 (U) 139 (L)	140 (U) 140 (L)	**CLK_OUT_PLUS1C** **LK_IN_MINUS1**	**CLK_IN_MINUS1**
D_RTCK	**D_RTCK**	141	142 (U) 142 (L)	**CLK_OUT_PLUS2C** **LK_UP_THRU**	**CLK_UP_THRU**
C_NSRST	**C_nSRST**	143	144 (U) 144 (L)	**CLK_IN_PLUS2** **CLK_DN_THRU**	**CLK_IN_PLUS2**
C_NTRST	**C_nTRST**	145	146 (U) 146 (L)	**CLK_IN_PLUS1CLK** **_OUT_MINUS1**	**CLK_IN_PLUS1**
PLD_TDIC_TDO **_OUT**	**C_TDO_INC_TD** **O_OUT**	147 (U) 147 (L)	148 (U) 148 (L)	**CLK_DN_THRUCL** **K_OUT_MINUS2**	**CLK_DN_THRU**
C_TDI	**C_TDI**	149	150	**CLK_GLOBAL**	**CLK_GLOBAL**
C_TCK_OUTC_ **TCK_IN**	**C_TCK_OUT**	151 (U) 151 (L)	152	**FPGA_IMAGE**	**FPGA_IMAGE**
C_TMS_OUTC_ **TMS_IN**	**C_TMS_OUT**	153 (U) 153 (L)	154	**nSYSPOR**	**nSYSPOR**
nTILE_DET	**nTILE_DET**	155	156	**nSYSRST**	**nSYSRST**
NCFGEN	**nCFGEN**	157	158	**nRTCKEN**	**nRTCKEN**
GLOBAL_DONE	**GLOBAL_DONE**	159	160	**SPARE12** (reserved)	**SPARE12**
SPARE11	**SPARE11** (reserved)	161	162	**SPARE10** (reserved)	**SPARE10**
SPARE9	**SPARE9** (reserved)	163	164	**SPARE8** (reserved)	**SPARE8**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
SPARE7	**SPARE7** (reserved)	165	166	**SPARE6** (reserved)	**SPARE6**
SPARE5	**SPARE5** (reserved)	167	168	**SPARE4** (reserved)	**SPARE4**
SPARE3	**SPARE3** (reserved)	169	170	**SPARE2** (reserved)	**SPARE2**
SPARE1	**SPARE1** (reserved)	171	172	**SPARE0** (reserved)	**SPARE0**
Zx63	**Z63**	173	174	**Z64**	**Zx64**
Zx62	**Z62**	175	176	**Z65**	**Zx65**
Zx61	**Z61**	177	178	**Z66**	**Zx66**
Zx60	**Z60**	179	180	**Z67**	**Zx67**
Zx59	**Z59**	181	182	**Z68**	**Zx68**
Zx58	**Z58**	183	184	**Z79**	**Zx69**
Zx57	**Z57**	185	186	**Z70**	**Zx70**
Zx56	**Z56**	187	188	**Z71**	**Zx71**
Zx55	**Z55**	189	190	**Z72**	**Zx72**
Zx54	**Z54**	191	192	**Z73**	**Zx73**
Zx53	**Z53**	193	194	**Z74**	**Zx74**
Zx52	**Z52**	195	196	**Z75**	**Zx75**
Zx51	**Z51**	197	198	**Z76**	**Zx76**
Zx50	**Z50**	199	200	**Z77**	**Zx77**
Zx49	**Z49**	201	202	**Z78**	**Zx78**
Zx48	**Z48**	203	204	**Z79**	**Zx79**
Zx47	**Z47**	205	206	**Z80**	**Zx80**
Zx46	**Z46**	207	208	**Z81**	**Zx81**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Zx45	**Z45**	209	210	**Z82**	**Zx82**
Zx44	**Z44**	211	212	**Z83**	**Zx83**
Zx43	**Z43**	213	214	**Z84**	**Zx84**
Zx42	**Z42**	215	216	**Z85**	**Zx85**
Zx41	**Z41**	217	218	**Z86**	**Zx86**
Zx40	**Z40**	219	220	**Z87**	**Zx87**
Zx39	**Z39**	221	222	**Z88**	**Zx88**
Zx38	**Z38**	223	224	**Z89**	**Zx89**
Zx37	**Z37**	225	226	**Z90**	**Zx90**
Zx36	**Z36**	227	228	**Z91**	**Zx91**
Zx35	**Z35**	229	230	**Z92**	**Zx92**
Zx34	**Z34**	231	232	**Z93**	**Zx93**
Zx33	**Z33**	233	234	**Z94**	**Zx94**
Zx32	**Z32**	235	236	**Z95**	**Zx95**
Zx31	**Z31**	237	238	**Z96**	**Zx96**
Zx30	**Z30**	239	240	**Z97**	**Zx97**
Zx29	**Z29**	241	242	**Z98**	**Zx98**
Zx28	**Z28**	243	244	**Z99**	**Zx99**
Zx27	**Z27**	245	246	**Z100**	**Zx100**
Zx26	**Z26**	247	248	**Z101**	**Zx101**
Zx25	**Z25**	249	250	**Z102**	**Zx102**
Zx24	**Z24**	251	252	**Z103**	**Zx103**
Zx23	**Z23**	253	254	**Z104**	**Zx104**
Zx22	**Z22**	255	256	**Z105**	**Zx105**

Table 6-5 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Zx21	**Z21**	257	258	**Z106**	**Zx106**
Zx20	**Z20**	259	260	**Z107**	**Zx107**
Zx19	**Z19**	261	262	**Z108**	**Zx108**
Zx18	**Z18**	263	264	**Z109**	**Zx109**
Zx17	**Z17**	265	266	**Z110**	**Zx110**
Zx16	**Z16**	267	268	**Z111**	**Zx111**
Zx15	**Z15**	269	270	**Z112**	**Zx112**
Zx14	**Z14**	271	272	**Z113**	**Zx113**
Zx13	**Z13**	273	274	**Z114**	**Zx114**
Zx12	**Z12**	275	276	**Z115**	**Zx115**
Zx11	**Z11**	277	278	**Z116**	**Zx116**
Zx10	**Z10**	279	280	**Z117**	**Zx117**
Zx9	**Z9**	281	282	**Z118**	**Zx118**
Zx8	**Z8**	283	284	**Z119**	**Zx119**
Zx7	**Z7**	285	286	**Z120**	**Zx120**
Zx6	**Z6**	287	288	**Z121**	**Zx121**
Zx5	**Z5**	289	290	**Z122**	**Zx122**
Zx4	**Z4**	291	292	**Z123**	**Zx123**
Zx3	**Z3**	293	294	**Z124**	**Zx124**
Zx2	**Z2**	295	296	**Z125**	**Zx125**
Zx1	**Z1**	297	298	**Z126**	**Zx126**
Zx0	**Z0**	299	300	**Z127**	**Zx127**

6.2 Memory expansion connector pinout

The Core Tile has two 120-way Samtec QSH memory expansion connectors as shown in Figure 6-2. The connector pinout for the J6 (Z MEMEXP) is listed in Table 6-6 on page 6-21. The connector pinout for the J7 (Y MEMEXP) is listed in Table 6-7 on page 6-23.

Note

Refer to the documentation supplied with the memory boards and the *PISMO Memory Interface Specification* for details of signal assignment on the boards. Different memory boards might have different voltage levels or different arrangement of address or control signals.

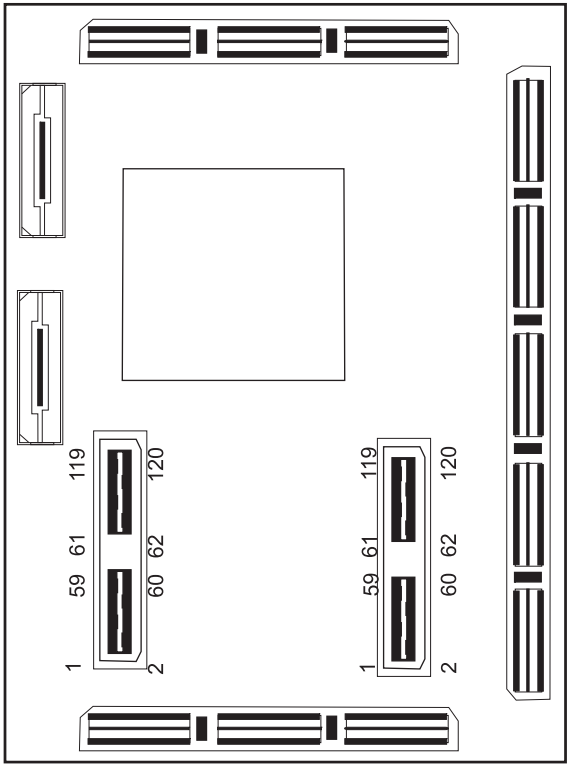

Figure 6-2 Memory expansion connectors

6.2.1 Memory expansion connector J6

The signals on the Z MEMEXP socket are listed in Table 6-6 on page 6-21.

Table 6-6 Z memory connector signals

Pin No.	Core Tile signal	Pin No.	Core Tile signal
1	**MEMEXPA0**	2	**3V3**
3	**MEMEXPA1**	4	**3V3**
5	**MEMEXPA2**	6	**3V3**
7	**MEMEXPA3**	8	**3V3**
9	**MEMEXPA4**	10	**VDDIO**[a]
11	**MEMEXPA5**	12	**VDDIO**[a]
13	**MEMEXPA6**	14	**VDDIO**[a]
15	**MEMEXPA7**	16	**VDDIO**[a]
17	**MEMEXPA8**	18	**1V8**
19	**MEMEXPA9**	20	**1V8**
21	**MEMEXPA10**	22	**1V8**
23	**MEMEXPA11**	24	**1V8**
25	**MEMEXPA12**	26	**MEMEXPA103**
27	**MEMEXPA13**	28	**MEMEXPA102**
29	**MEMEXPA14**	30	**MEMEXPA101**
31	**MEMEXPA15**	32	**MEMEXPA100**
33	**MEMEXPA16**	34	**5V**
35	**MEMEXPA17**	36	**5V**
37	**MEMEXPA18**	38	**5V**
39	**MEMEXPA19**	40	**5V**
41	**MEMEXPA20**	42	**MEMEXPA99**
43	**MEMEXPA21**	44	**MEMEXPA98**
45	**MEMEXPA22**	46	**MEMEXPA97**
47	**MEMEXPA23**	48	**MEMEXPA96**

Table 6-6 Z memory connector signals (continued)

Pin No.	Core Tile signal	Pin No.	Core Tile signal
49	**MEMEXPA24**	50	**MEMEXPA95**
51	**MEMEXPA25**	52	**MEMEXPA94**
53	**MEMEXPA26**	54	**MEMEXPA93**
55	**MEMEXPA27**	56	**MEMEXPA92**
57	**MEMEXPA28**	58	**MEMEXPA91**
59	**MEMEXPA29**	60	**MEMEXPA89**
61	**MEMEXPA30**	62	**SBSCL** (level-shifted **MEMEXPA90**)
63	**MEMEXPA31**	64	**SBSDA** (level-shifted **MEMEXPA88**)
65	**MEMEXPA32**	66	**MEMEXPA87**
67	**MEMEXPA33**	68	**MEMEXPA86**
69	**MEMEXPA34**	70	**MEMEXPA85**
71	**MEMEXPA35**	72	**MEMEXPA84**
73	**MEMEXPA36**	74	**MEMEXPA83**
75	**MEMEXPA37**	76	**MEMEXPA82**
77	**MEMEXPA38**	78	**MEMEXPA81**
79	**MEMEXPA39**	80	**MEMEXPA80**
81	**MEMEXPA40**	82	**MEMEXPA79**
83	**MEMEXPA41**	84	**MEMEXPA78**
85	**MEMEXPA42**	86	**MEMEXPA77**
87	**MEMEXPA43**	88	**MEMEXPA76**
89	**MEMEXPA44**	90	**MEMEXPA75**
91	**MEMEXPA45**	92	**MEMEXPA74**
93	**MEMEXPA46**	94	**MEMEXPA73**

Table 6-6 Z memory connector signals (continued)

Pin No.	Core Tile signal	Pin No.	Core Tile signal
95	**MEMEXPA47**	96	**MEMEXPA72**
97	**MEMEXPA48**	98	**MEMEXPA71**
99	**MEMEXPA49**	100	**MEMEXPA70**
101	**MEMEXPA50**	102	**MEMEXPA69**
103	**MEMEXPA51**	104	**MEMEXPA68**
105	**MEMEXPA52**	106	**MEMEXPA67**
107	**MEMEXPA53**	108	**MEMEXPA66**
109	**MEMEXPA54**	110	**MEMEXPA65**
111	**MEMEXPA55**	112	**MEMEXPA64**
113	**MEMEXPA56**	114	**MEMEXPA63**
115	**MEMEXPA57**	116	**MEMEXPA62**
117	**MEMEXPA58**	118	**MEMEXPA61**
119	**MEMEXPA59**	120	**MEMEXPA60**

a. **VDDIO** is the I/O voltage to host. This is not routed through on stackable boards.

6.2.2 Memory expansion connector J7

The signals on the Y MEMEXP socket are listed in Table 6-7.

Table 6-7 Y memory connector signals

Pin No.	Core Tile signal	Pin No.	Core Tile signal
1	**MEMEXPB0**	2	**3V3**
3	**MEMEXPB1**	4	**3V3**
5	**MEMEXPB2**	6	**3V3**
7	**MEMEXPB3**	8	**3V3**
9	**MEMEXPB4**	10	**3V3**

Table 6-7 Y memory connector signals (continued)

Pin No.	Core Tile signal	Pin No.	Core Tile signal
11	**MEMEXPB5**	12	**3V3**
13	**MEMEXPB6**	14	**3V3**
15	**MEMEXPB7**	16	**3V3**
17	**MEMEXPB8**	18	**1V8**
19	**MEMEXPB9**	20	**1V8**
21	**MEMEXPB10**	22	**1V8**
23	**MEMEXPB11**	24	**1V8**
25	**MEMEXPB12**	26	**MEMEXPB103**
27	**MEMEXPB13**	28	**MEMEXPB102**
29	**MEMEXPB14**	30	**MEMEXPB101**
31	**MEMEXPB15**	32	**MEMEXPB100**
33	**MEMEXPB16**	34	**5V**
35	**MEMEXPB17**	36	**5V**
37	**MEMEXPB18**	38	**5V**
39	**MEMEXPB19**	40	**5V**
41	**MEMEXPB20**	42	**MEMEXPB99**
43	**MEMEXPB21**	44	**MEMEXPB98**
45	**MEMEXPB22**	46	**MEMEXPB97**
47	**MEMEXPB23**	48	**MEMEXPB96**
49	**MEMEXPB24**	50	**MEMEXPB95**
51	**MEMEXPB25**	52	**MEMEXPB94**
53	**MEMEXPB26**	54	**MEMEXPB93**
55	**MEMEXPB27**	56	**MEMEXPB92**
57	**MEMEXPB28**	58	**MEMEXPB91**
59	**MEMEXPB29**	60	**MEMEXPB89**

Table 6-7 Y memory connector signals (continued)

Pin No.	Core Tile signal	Pin No.	Core Tile signal
61	**MEMEXPB30**	62	Serial bus clock (**MEMEXPB90**)
63	**MEMEXPB31**	64	Serial bus data (**MEMEXPB88**)
65	**MEMEXPB32**	66	**MEMEXPB87**
67	**MEMEXPB33**	68	**MEMEXPB86**
69	**MEMEXPB34**	70	**MEMEXPB85**
71	**MEMEXPB35**	72	**MEMEXPB84**
73	**MEMEXPB36**	74	**MEMEXPB83**
75	**MEMEXPB37**	76	**MEMEXPB82**
77	**MEMEXPB38**	78	**MEMEXPB81**
79	**MEMEXPB39**	80	**MEMEXPB80**
81	**MEMEXPB40**	82	**MEMEXPB79**
83	**MEMEXPB41**	84	**MEMEXPB78**
85	**MEMEXPB42**	86	**MEMEXPB77**
87	**MEMEXPB43**	88	**MEMEXPB76**
89	**MEMEXPB44**	90	**MEMEXPB75**
91	**MEMEXPB45**	92	**MEMEXPB74**
93	**MEMEXPB46**	94	**MEMEXPB73**
95	**MEMEXPB47**	96	**MEMEXPB72**
97	**MEMEXPB48**	98	**MEMEXPB71**
99	**MEMEXPB49**	100	**MEMEXPB70**
101	**MEMEXPB50**	102	**MEMEXPB69**
103	**MEMEXPB51**	104	**MEMEXPB68**
105	**MEMEXPB52**	106	**MEMEXPB67**
107	**MEMEXPB53**	108	**MEMEXPB66**
109	**MEMEXPB54**	110	**MEMEXPB65**

Table 6-7 Y memory connector signals (continued)

Pin No.	Core Tile signal	Pin No.	Core Tile signal
111	**MEMEXPB55**	112	**MEMEXPB64**
113	**MEMEXPB56**	114	**MEMEXPB63**
115	**MEMEXPB57**	116	**MEMEXPB62**
117	**MEMEXPB58**	118	**MEMEXPB61**
119	**MEMEXPB59**	120	**MEMEXPB60**

6.3 Trace Port connectors

Table 6-8 and Table 6-9 on page 6-28 lists the pinout of the trace connectors A and B. The Mictor connector is shown in Figure 6-3 on page 6-29.

Table 6-8 Trace connector A J9

Channel	Pin	Pin	Channel
Not connected	1	2	Not connected
GND	3	4	Not connected
Not connected	5	6	**TRACECLKA**
Not connected	7	8	Not connected
Not connected	9	10	**EXTTRIG**
Not connected	11	12	**ARM_VDDIO**
Not connected	13	14	**3V3**
Not connected	15	16	**TRACEPKTA7**
Not connected	17	18	**TRACEPKTA6**
Not connected	19	20	**TRACEPKTA5**
Not connected	21	22	**TRACEPKTA4**
TRACEPKTA15	23	24	**TRACEPKTA3**
TRACEPKTA14	25	26	**TRACEPKTA2**
TRACEPKTA13	27	28	**TRACEPKTA1**
TRACEPKTA12	29	30	**TRACEPKTA0**
TRACEPKTA11	31	32	**TRACESYNCA**
TRACEPKTA10	33	34	**PIPESTATA2**
TRACEPKTA9	35	36	**PIPESTATA1**
TRACEPKTA8	37	38	**PIPESTATA0**

Table 6-9 Trace connector B J10

Channel	Pin	Pin	Channel
5V	1	2	Not connected
GND	3	4	Not connected
Not connected	5	6	**TRACECLKB**
Not connected	7	8	Not connected
Not connected	9	10	Not connected
Not connected	11	12	**ARM_VDDIO**
Not connected	13	14	**3V3**
Not connected	15	16	**TRACEPKTB7**
Not connected	17	18	**TRACEPKTB6**
Not connected	19	20	**TRACEPKTB5**
Not connected	21	22	**TRACEPKTB4**
TRACEPKTB15	23	24	**TRACEPKTB3**
TRACEPKTB14	25	26	**TRACEPKTB2**
TRACEPKTB13	27	28	**TRACEPKTB1**
TRACEPKTB12	29	30	**TRACEPKTB0**
TRACEPKTB11	31	32	**TRACESYNCB**
TRACEPKTB10	33	34	**PIPESTATB2**
TRACEPKTB9	35	36	**PIPESTATB1**
TRACEPKTB8	37	38	**PIPESTATB0**

Note

Agilent (formerly HP) and Tektronix label these connectors differently, but the assignments of signals to physical pins is appropriate for both systems and pin 1 is always in the same place. The schematic is labelled according to the Agilent pin assignment.

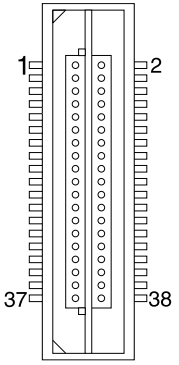

Figure 6-3 Mictor connector

6.4 Links and test points

This section describes the links and test points present on the Core Tile.

6.4.1 Links

Figure 6-4 shows the location of the links on the bottom of the Core Tile. Figure 6-4 shows the location of the links on the top of the Core Tile. The function of each link is listed in Table 6-10 on page 6-32.

There are also resistor links that are fitted for routing clock signals, enabling voltage measurements, or for specific test chip functions. These are described in Table 6-11 on page 6-32 to Table 6-14 on page 6-34.

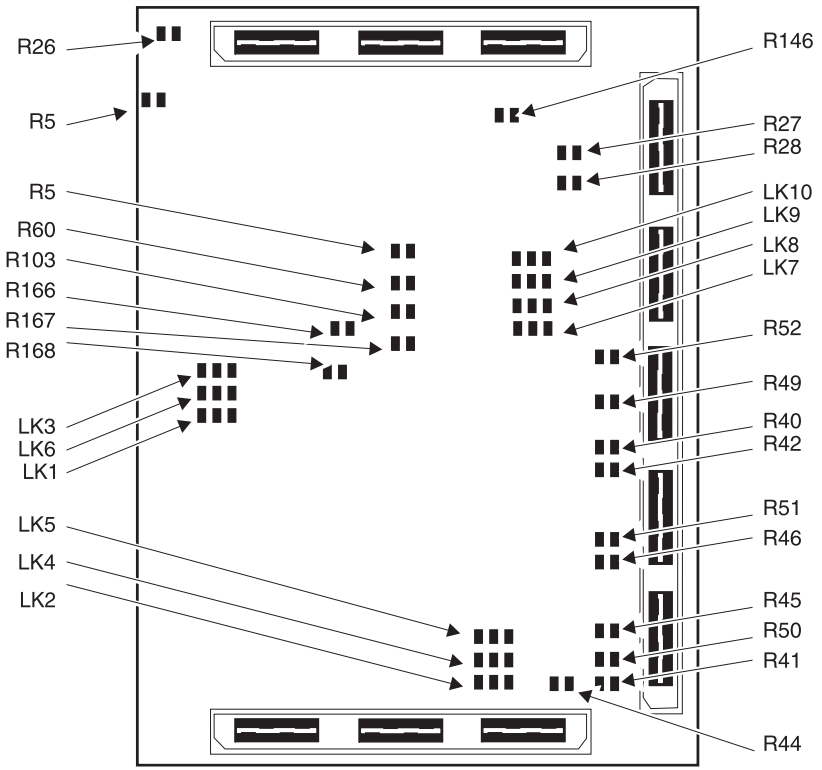

Figure 6-4 Location of links (bottom)

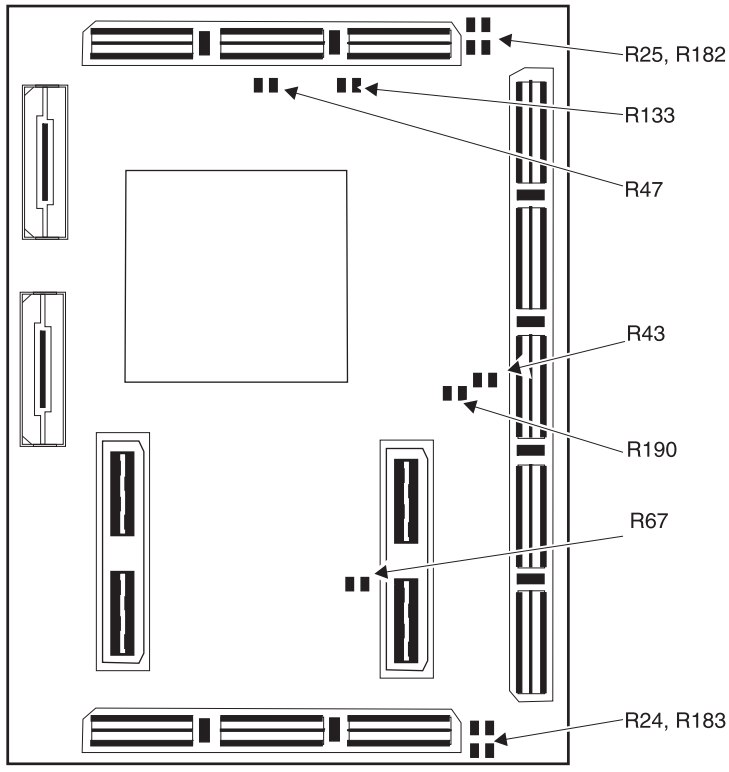

Figure 6-5 Location of links (top)

Table 6-10 Link function

Link	Description
1-3	**IMEMSIZE[2:0]**, sets size of instruction TCM memory. **Note** These links are not used. Select the memory size using the PLD configuration signals, see *Memory located inside test chip* on page 3-8 and *Core Tile PLD signals* on page 3-30,.
4-6	**DMEMSIZE[2:0]**, sets size of instruction TCM memory. **Note** These links are not used. Select the memory size using the PLD configuration signals, see *Memory located inside test chip* on page 3-8 and *Core Tile PLD signals* on page 3-30.
10-7	**MANID[3:0]**, identifies the board build variant. **Note** These links are set at manufacture and should not be changed).

JTAG links

Resistor links bypass some of the JTAG signals depending on the test chip fitted as listed in Table 6-11. See the BOM file for the fitting options used at manufacture.

Table 6-11 JTAG links

Resistor link	Description
R44	If fitted, **nRTCKEN** is connected to ground when the system is in debug mode (CONFIG link not fitted).
R188	Links **D_TCK_BYPASS** to **RTCK_ARM**
R190	Links a buffered version of **D_TCK_IN** to **D_TCK_BYPASS**

Clock selection links

Some of the clock signals (see *Clocks* on page 3-10) have associated resistor links as listed in Table 6-12. The resistor links are set at manufacture and do not normally require changing.

Table 6-12 Clock signal resistor links

Resistor link	Description
R133	Links **X_REFCLK_UP** on the lower header to **X_REFCLK_DN** on the upper header
R46	Links **X_HCLKIN_UP** on the lower header to **X_HCLKIN_DN** on the upper header
R47	Links **X_HCLK_DN** on the lower header to **X_HCLK_UP** on the upper header. This link is used if the test chip does not output both **HCLKEXT2** and **HCLKEXT3**.
R40 R49, R41 R50, R39 R52, R42 R45, and R43 R51	These links select between buffered and non-buffered signals for **HCLKEXT0**, **HCLKEXT1**, **HCLKEXT4**, **CLK_NEG_x**, and **CLK_GLOBAL_IN**. The buffer amplifiers are bypassed if short delays are more important than minimum loading. These links are set at manufacture for the specific Core Tile and do not normally require changing.

Note

The resistor links shown in Figure 3-3 on page 3-12 are not normally fitted at manufacture. Resistors R46, R47, and R133 are only used if no test chip is fitted to the Core Tile in order to, for example, test the printed-circuit board.

Voltage links

There are resistor links that select the operating voltages for the tile, see *Power supply control* on page 3-17.

Caution

The resistor links listed in Table 6-13 on page 6-34 are set at manufacture and (other than the current sense resistors for **VDDCORE[6:1]**) do not normally require changing.

Table 6-13 Voltage resistor links

Resistor link	Description
R182 and R25	Links 3.3V from HDRZ to **ARM_VDDIO** line.
R183 and R24	Links 3.3V from HDRZ to **VCCOY** line.
R60	Current sense resistor for **VDDCORE1**.
R67	Current sense resistor for **VDDCORE2**.
R103	Current sense resistor for **VDDCORE3**.
R166	Current sense resistor for **VDDCORE4**.
R167	Current sense resistor for **VDDCORE5**.
R168	Current sense resistor for **VDDCORE6**.

SDRAM links

The resistor links listed in Table 6-14 enable support for the **VREF** signals used with DDR SDRAM memory modules. These signals must be driven by an attached board such as a Logic Tile.

Table 6-14 DDR SDRAM resistor links

Resistor link	Description
R26	links **MEXPA77** to **YU3**
R27	links **MEXPA78** to **YU17**
R28	links **MEXPA79** to **YU25**

6.4.2 Test points

Figure 6-6 on page 6-35 shows the location of test points on the Core Tile. Table 6-15 on page 6-35 lists the function of each test point.

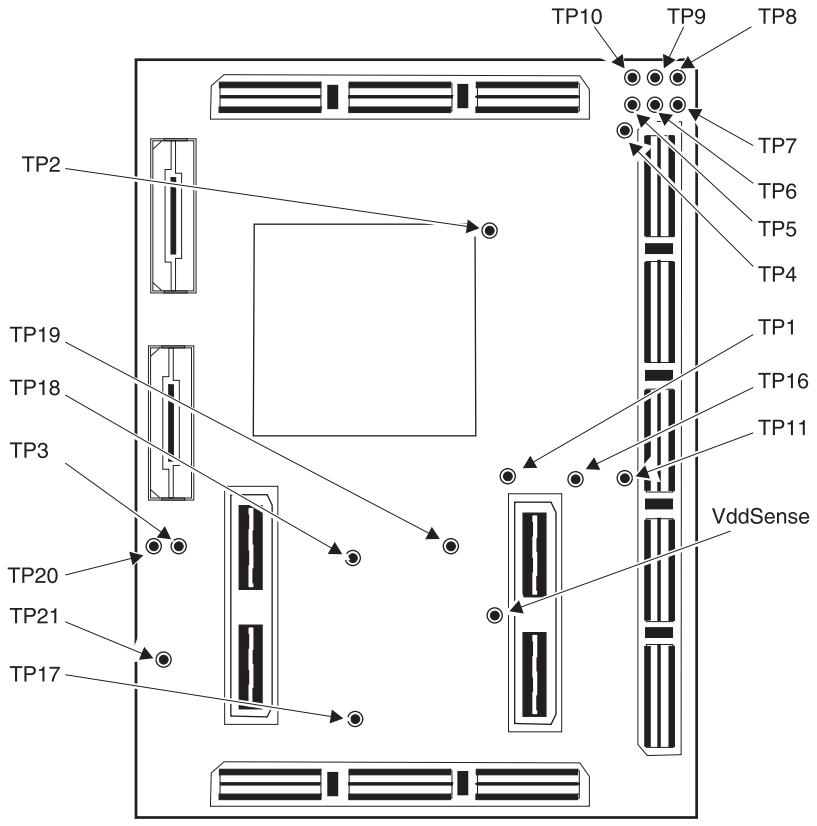

Figure 6-6 Test point location

Table 6-15 Test point function

Test point	Description
1-3	Manufacturing test
9-4	**USEROUT** from test chip
10	**PLLLOCK**
11	**CLKGLOBAL_IN**
12-15	Ground
16	**CLKGLOBAL_OUT**

Table 6-15 Test point function (continued)

Test point	Description
17	**Vdd CORE A**
18	**Vdd CORE B**
19	**Vdd CORE C**
20	**Vdd I/O**
21	**Vdd PLL**
22	**VDD SENSE**

6.5 AHB bus timing specification

This section describes the timing for Core Tiles that use an AHB system bus.

Note

Timings in this section refer to the AHB bus in general. For timings specific to the test chip used on your Core Tile, refer to the release notes for your Core Tile.

6.5.1 Core Tile timing and the AMBA Specification

The parameters listed are those specified in the AMBA Specification with the following important differences:

- only output valid and input setup times are quoted
- the required input hold time (Tih) is always less than or equal to 0ns
- the output hold time (Toh) is always greater than 2ns.

Each version and revision of the FPGA in an attached Logic Tile or baseboard has subtly different timing. The typical figures shown in Table 6-16 on page 6-38 are those you can expect under nominal conditions and must be used as a guideline when designing your own boards. The typical figures have been rounded to simplify timing analysis and constraints.

The system bus on Versatile Logic Tiles and baseboards is routed between FPGAs. These FPGAs are routed with timing constraints similar to those shown the tables in *Timing parameter tables* on page 6-38. The exact performance of a system depends on the timing parameters of the baseboard and all tiles in the system. Some allowance also has to be made for clock skew, routing delays and number of modules (that is, loading).

Not all Logic Tile or baseboard FPGA implementations meet the ideal timing parameters, due to the complexity of the design or routing congestion within the device. For this reason, the PLL clock generators on baseboards default to a safe low value that all modules can achieve. See the documentation supplied with your baseboard for details on changing the clock frequency.

A detailed timing analysis involves calculating the input/output delays between modules for all timing parameters. In general, the simplest approach to determine the maximum operating frequency is to increase the frequency of the clock generators until the system becomes unstable.

6.5.2 Timing parameter tables

Table 6-16 shows the clock and reset timing parameters.

Table 6-16 Clock and reset parameters

Parameter	Description	Typ
Tclk	**HCLK** clock period	30
Tisrst	**HRESETn** deasserted setup time before **HCLK**	15

Table 6-17 shows the AHB slave input parameters.

Table 6-17 AHB slave input parameters

Parameter	Description	Typ
Tistr	Transfer type setup time before **HCLK**	5
Tisa	**HADDR[31:0]** setup time before **HCLK**	10
Tisctl	**HWRITE**, **HSIZE[2:0]** and **HBURST[2:0]** setup time before HCLK	5
Tiswd	Write data setup time before **HCLK**	5
Tisrdy	Ready setup time before **HCLK**	5

Table 6-18 shows the AHB slave output parameters.

Table 6-18 AHB slave output parameters

Parameter	Description	Typ
Tovrsp	Response valid time after **HCLK**	15
Tovrd	Data valid time after **HCLK**	15
Tovrdy	Ready valid time after **HCLK**	15

Table 6-19 shows the bus master input parameters.

Table 6-19 Bus master input timing parameters

Parameter	Description	Typ
Tisgnt	**HGRANTx** setup time before **HCLK**	5
Tisrdy	Ready setup time before **HCLK**	5
Tisrsp	Response setup time before **HCLK**	5
Tisrd	Read data setup time before **HCLK**	5

Table 6-20 shows the bus master output timing parameters.

Table 6-20 Bus master output timing parameters

Parameter	Description	Typ
Tovtr	Transfer type valid time after **HCLK**	15
Tova	Address valid time after **HCLK**	15
Tovctl	**HWRITE**, **HSIZE[2:0]** and **HBURST[2:0]** valid time after HCLK	15
Tovwd	Write data valid time after **HCLK**	15
Tovreq	Request valid time after **HCLK**	15
Tovlck	Lock valid time after **HCLK**	15

Part C

Printed Circuit Board HBI-0141 (CT7TDMI and CT7TDMI-S)

Chapter 7
HBI-0141 Hardware Description

This chapter describes the on-board hardware on the ARM7TDMI and ARM7TDMI-S Core Tiles that use the HBI-0141 printed circuit board (the companion Logic Tile or Interface Module for these tiles are also described where relevant). It contains the following sections:

Note

The HBI-0141 board supports both the ARM7TDMI and ARM7TDMI-S processors, but the availability of boards depends on the availability of test chips. Contact your sales representatives for details on currently available Core Tiles.

Note

This chapter describes the generic hardware and is independent of the FPGA image used in an external Logic Tile, Interface Module, or baseboard.

The diagrams in this chapter (such as the block diagram shown in *Core Tile block diagram* on page 7-3 for example) are typical of Core Tiles, but your Core Tile architecture might be different depending on the test chip used.

The specific clock interconnections, voltage setting resistors and link, and bus interfaces might be different for different test chips. Different test chips might also use the configuration signals in different ways. Refer to the *Technical Reference Manual* for your processor family, the test chip reference information on the CD supplied with your Core Tile, and the appendices in this manual for more detail.

7.1 Core Tile architecture

Configuration and interface logic on the Core Tile is connected to the upper and lower header connectors as shown in Figure 7-1.

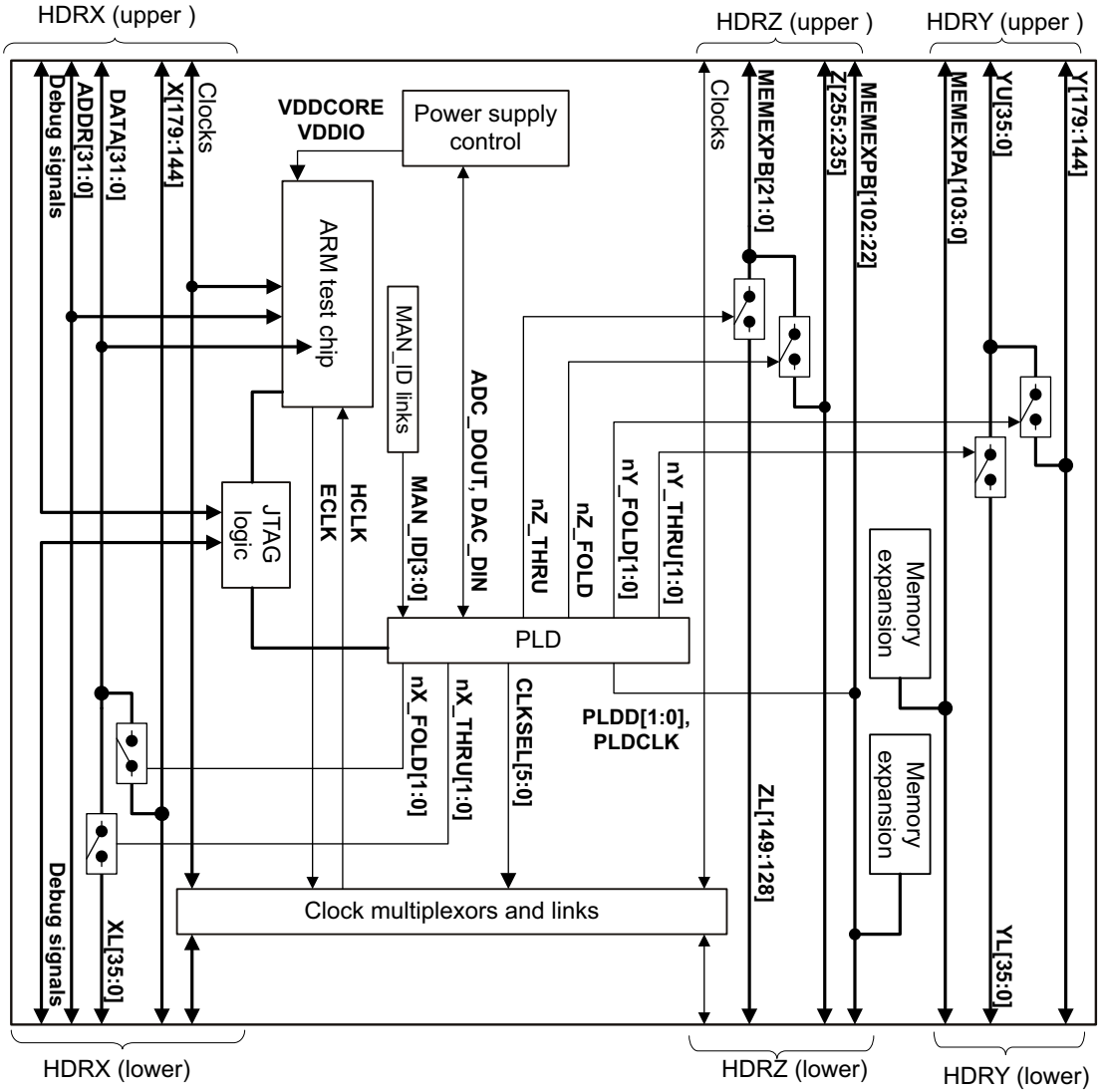

Figure 7-1 Core Tile block diagram

The signals shown in Figure 7-1 are summarized in Table 7-1 on page 7-4.

Table 7-1 Overview of Core Tile signals

Function	Signals	Description
Power	**VDDCORE**, **VDDIO**, **ADC_OUT**, **DAC_DIN**	The test chip voltages are supplied by on-board regulators. Some of the voltages can be controlled and measured by the serial data interface on the PLD. See *Power supply control* on page 7-19.
PLD configuration	**PLDD[1:0]**, **PLDCLK**	The PLD on the Core Tile is configured over a dedicated serial link to an attached Logic Tile. See *Core Tile PLD signals* on page 7-32.
Expansion memory	**MEMEXPB[102:22]**, **MEMEXPA[103:0]**	Connection to the on-board memory expansion connectors. There is no connection between these memory signals and the test chip bus. See *Memory expansion boards* on page 7-12.
Test chip signals	**ADDR[31:0]**, **DATA[31:0]**, and bus control signals	The test chip buses and control signals are brought out to header connectors. The interface to the buses is provided by an attached Logic Tile. See *ARM microprocessor test chip* on page 7-5.
Fold and isolation switches	**nZ_FOLD**, **nZ_THRU**, **nY_FOLD[1:0]**, **nY_THRU[1:0]**, **nX_FOLD[1:0]**, **nX_THRU[1:0]**,	Some of the signals have isolation switches or foldover switches that control the signal routing. See *Isolation and foldover of header signals* on page 7-26.
Debug	**TDI**, **TDO**, **TDO**, **TMS** and others	The JTAG signals connect to the test chip, the PLD, and the tile mounted above the Core Tile (if present). See *JTAG support* on page 7-40.
Clocks	**MCLK**, **ECLK**, **CLK_GLOBAL**, **CLK_NEG_UP_IN**, and others	The Core Tile does not have a clock generator. The clock for the test chip is selected from one of the external clocks provided by an attached Logic Tile. See *Clocks* on page 7-13.

7.2 ARM microprocessor test chip

Table 7-2 provides a brief overview of the signals present on test chips that follow the ARM7 bus specification. The ARM7TDMI test chip, not the ARM7TDMI-S, is normally fitted to the Core Tile. The signals on the schematic and the figures in this chapter use the ARM7TDMI signal names except when the equivalent signal does not exist on the ARM7TDMI. Refer to the *ARM7TDMI-S Technical Reference Manual* for details on the ARM7TDMI-S.

Note

Different test chips variants might use some of the pins in ways that are specific to that test chip. For more information on signals for particular test chip variants, see the chapter in this manual for the test chip family, the *Technical Reference Manual* for the test chip, and the information provided on the CD.

Test chip configuration is covered in:

- *Clocks* on page 7-13
- *Power supply control* on page 7-19
- *Through/break control for HDRX* on page 7-26.

Table 7-2 Test chip signals

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Bus	**ABE**	**MEM64M**	Address bus enable, address bus drivers are disabled if this signal is LOW	Output
Bus	**ABORT**	**ABORT**	Memory abort or bus error	Input
Bus	**ADDR[31:0]**	**ADDR[31:0]**	Address bus	Output
Bus	**ALE**	**CLKHI**	Address latch enable, the address bus, **LOCK**, **MAS[1:0]**, **nRW**, **nOPC**, and **nTRANS** signals are latched when this signal is held LOW. **Note** This signal is provided for backward compatibility. New designs should use **APE**.	Output

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Bus	**APE**	-	Address pipeline enable, selects whether the address bus, **LOCK**, **MAS[1:0**], **nRW**, **nOPC**, and **nTRANS** signals operate in pipelined (**APE** HIGH) or depipelined mode (**APE** LOW) (ARM7TDMI only)	Input
Bus	**BL[3:0]**	-	Byte latch control, values on the data bus are latched on the falling edge of **MCLK** when these signals are HIGH (ARM7TDMI only)	Output
Bus	**BC[1:0]**	**SWT_T**, **LED_N**	Cycle type, versions of **MREQ** and **SEQ** that may be modified by **F[1:0]** (ARM7TDMI-S I/O signals)	Output
Bus	**BCE**	**TRAMBE**	Enable **BC[1:0]** if LOW	Input
Bus	**CPA**	**CPA**	Coprocessor absent	Input
Bus	**CPB**	**CPB**	Coprocessor busy	Input
Bus	**TBIT**	**DRAMWE_N**	Thumb instruction set code being executed (ARM7TDMI-S memory control)	Output
Bus	**CPSEQ**	**CPSEQ**	Sequential address	Output
Bus	**DATA[31:0]**	**DATA[31:0]**	Data bus (input and output)	I/O
Bus	**DBE**	**SCANIN**	Data bus enable, if HIGH, data appears on the data bus (serial test input for ARM7TDMI-S)	Input
Bus	**EABE**	-	External Address Bus Enable, if driven LOW, the **A[31:0]**, **nRW**, **LOCK**, **nOPC**, **nTRANS**, and **MAS[1:0]** drivers are put into high impedance state. (ARM7TDMI only)	Input
Bus	**EDBE**	**DBE**	External Data Bus Enable, if driven LOW, the **D[31:0]** drivers are put into high impedance state.	Input
Bus	**EEBE**	**INPUT180**	External Enable, if driven LOW, places external outputs not controlled by **EABE** or **EDBE** into high impedance state.	Input

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Bus	**F[1:0]**	**CFGMEM1**, **CFGMEM0**	Cycle function modifiers, these signals can be used to override the requests normally made by the ARM7 processor.	Input
Bus	**LOCK**	**LOCK**	Locked transfer indication	Output
Bus	**MAS[1:0]**	**SIZE[1:0]**	Memory access size, indicates to the memory system the size of the data transfer for read and write cycles.	Output
Bus	**nCPI**	**LCDEN**	Coprocessor instruction (active LOW) (ARM7TDMI-S is I/O signal)	Output
Bus	**nENIN**	**POR_N**	Not enable input, used in conjunction with **nENOUT** to control the data bus during write cycles.	Input
Bus	**nENOUT**	**SCC_N**	Not enable output, driven LOW before the rising edge of **MCLK**.	Output
Bus	**nM[4:0]**	**RAS0_N**, **CAS3_N**, **CAS2_N**, **CAS1_N**, **CAS0_N**	Not processor mode, inverse logic level from the internal status bits that indicate processor operation mode (ARM7TDMI-S **CAS[3:0]** and **RAS[0]** are DRAM control signals)	Output
Bus	**nMREQ**	**CPnMREQ**	Memory request (active LOW)	Output
Bus	**nOPC**	**PROT0**	Opcode fetch (active LOW)	Output
Bus	**nRW**	**WRITE**	Data bus direction	Output
Bus	**nTRANS**	**PROT1**	Not memory translate, LOW if processor is in USER mode	Output
Bus	**nWAIT**	**CLKEN**	System memory clock enable. If LOW, do not advance the core clock on the next rising **CLK** input.	Input
Bus	**SEL[1:0]**	**CORESYNC**, **COREFLOW**	Data bus lane select, used to steer data to and from the macrocell and the external data bus. (**CORESYNC** usually LOW for ARM7TDMI-S, HIGH for ARM7TDMI validation**COREFLOW** usually HIGH for ARM7TDMI-S, LOW for ARM7TDMI)	Input

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Bus	**SEQ**	**CPSEQ**	Indicates sequential access to memory.	
Bus	**WAIT2**	**CFGMEM3**	WAIT (ARM7TDMI-S memory configuration)	Input
Bus	**WAITSEL**	**CFGMEM2**	Wait source select, if HIGH, selects **WAIT2** instead of **nWAIT** (ARM7TDMI-S **CFGMEM[3:0]** is usually b01)	Input
Bus	**CPnTRANS**	-	Memory translation (active LOW to indicate that the processor is in User mode) (ARM7TDMI only)	Output
Clock	**ECLKEN**	-	Clock output enable (ARM7TDMI only)	Output
Clock	**ECLK**	**CLKENCORE**	Clock output. In normal mode, this is **MCLK** output from the core. In debug mode ECLK is generated internally from TCK.	Output
Clock	**MCLK**	**CLKLO**	Clock to test chip. Times all memory accesses and internal operations.	Input
Debug	**BREAKPT**	-	Pulling this signal HIGH causes a conditional request for the processor to enter debug state (ARM7TDMI only)	Input
Debug	**COMMRX**	**RAMA4**	ICE communications receive buffer not empty (ARM7TDMI-S **RAM[4]** is a memory address)	Output
Debug	**COMMTX**	**RAMA3**	ICE communications transmit buffer empty (ARM7TDMI-S **RAM[3]** is a memory address)	Output
Debug	**DBGACK**	**DBGACK**	Debug acknowledge	Output
Debug	**DBGEN**	**DBGEN**	Debug enable	Input
Debug	-	**DBGEXT[1:0]**	ICE external inputs (ARM7TDMI-S only)	Input
Debug	**DBGINSTRVALID**	**DBGINSTRVALID**	Instruction executed. Goes HIGH for one cycle for each instruction committed to the execute stage of the pipeline	Output

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Debug	**DBGnEXEC**		Current instruction is not being executed	Output
Debug	**DBGRQ**	**DBGRQ**	Debug request	Input
Debug	**DBGRQI**	-	Internal debug request, this signal is the logical OR of **DBGRQ** and bit 1 of the debug control register (ARM7TDMI only)	Output
Debug	**DBGRNG[1:0]**	**DBGRNG[1:0]**	ICE range out	Output
Debug	**DBGTCKEN**	-	Test clock enable (ARM7TDMI only)	Input
Debug	**EXTERN[1:0]**	**DBGEXT0**, **DBGEXT1**	Embedded ICD EXTERN debug qualifiers (tie LOW if not required)	Input
Debug	**HIGHZ**	**OE_N**	Tri-state test chip signals. If the HIGHZ instruction has been loaded into the TAP controller, this signal is HIGH.	Input
Debug	**IR[3:0]**	**WBYTE[3:0]_N**	Instruction register, reflects the current instruction loaded into the TAP controller instruction register. (ARM7TDMI-S is memory write control)	Output
Debug	**ISYNC**	**CFGBOOT**	Synchronous interrupts, set HIGH if nIRQ and nFIQ are synchronous to the processor clock. (ARM7TDMI-S is configuration control)	Input
Debug	**nENOUTI**	**XIO_N**	Not enable output for coprocessor transfer, driven LOW during a coprocessor register transfer C cycle from the Embedded ICE communications channel processor	Output
Debug	**nEXEC**	**DBGnEXEC**	Not executed. Indicates that the instruction in the execution unit is not being executed	Output
Debug	**nTDOEN**	**nTDOEN**	Enable TDO tristate buffer	Output
Debug	**RANGEOUT[1:0]**	**RAMA[10:9]**	EmbeddedICE rangeout qualifier outputs for ARM7TDMI. (Equivalent ARM7S signals are **DBGRNG[1:0])** (ARM7TDMI-S **RAMA[10:9]** are RAM addresses)	Output

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Debug	**SCANENABLE**	**SCANENABLE**	Scan test path enable for test pattern generation	Input
Debug	**SCREG[3:0]**	**RAMA[8:5]**	Scan chain register, reflects the ID number of the scan chain selected by the TAP controller (ARM7TDMI-S **RAMA[8:5]** are RAM addresses)	Output
Debug	**TAPSM[3:0]**	-	TAP controller state machine, reflects the current state of the TAP controller. (ARM7TDMI only)	Output
Debug	**TBE**	**INPUT116**	Test bus enable, if LOW, **D[31:0]**, **A[31:0]**, **LOCK**, **MAS[1:0]**, **nRW**, **nTRANS**, and **nOPC** are set to high impedance.	Input
Debug	**TCK[2:1]**	**ROMB[1:0]**	**TCK** phase 1 and 2	Output
Debug	**TEST[3:0]**	**RAMDISABLE**, **RAS1_N**, **EXP_N**, **SCANOUT**	Test signals related to internal bus MD[0], **MDOUT[31]**, and chip delays. (ARM7TDMI-S signals are memory control, if **RAMDISABLE** is HIGH, the memory controller in the ARM7TDMI-S test chip is disables and the pin function reverts to the ARM7TDMI definition.)	-
Interrupt	**nFIQ**	**nFIQ**	External fast interrupt (active LOW)	Input
Interrupt	**nIRQ**	**nIRQ**	External interrupt (active LOW)	Input
JTAG	**ARM_nTRST**	**DBGnTRST**	Test reset to core, TAP and ICE	Input
JTAG	**ARM_TDI**	**DGBTDI**	Boundary scan Input	Input
JTAG	**ARM_DBGRQI**	**RTCK**	Debug request (ARM7TDMI-S return JTAG clock)	Input
JTAG	**ARM_TDO**	**DBGTDO**	Boundary scan Output	Output
JTAG	**ARM_TMS**	**DBGTMS**	ICE mode select	Input
JTAG	**DBBREAK**		ICE breakpoint/watchpoint indicator	Input
Memory	**NC[2:0]**	**RAMA0**, **RAMA1**, **RAMA2**	Reserved signals for ARM7TDMI (ARM7TDMI-S **RAM[2:1]** are memory addresses)	Output

Table 7-2 Test chip signals (continued)

Group	ARM7TDMI Signal	ARM7TDMI-S Signal	Description	Direction
Misc	**BIGEND**	**CFGBIGEND**	Selects big endian memory mode if HIGH	Input
Power	**ARM_VDDCORE**	**VDDCORE**	Voltage level used for processor core	Input
Power	**ARM_VDDIO**	**VDDIO**	Voltage level used for input and output signals	Input
Reset	**nRESET**	**nRESET**	Reset input - asynchronous	Input

Figure 7-2 shows the internal clock signals.

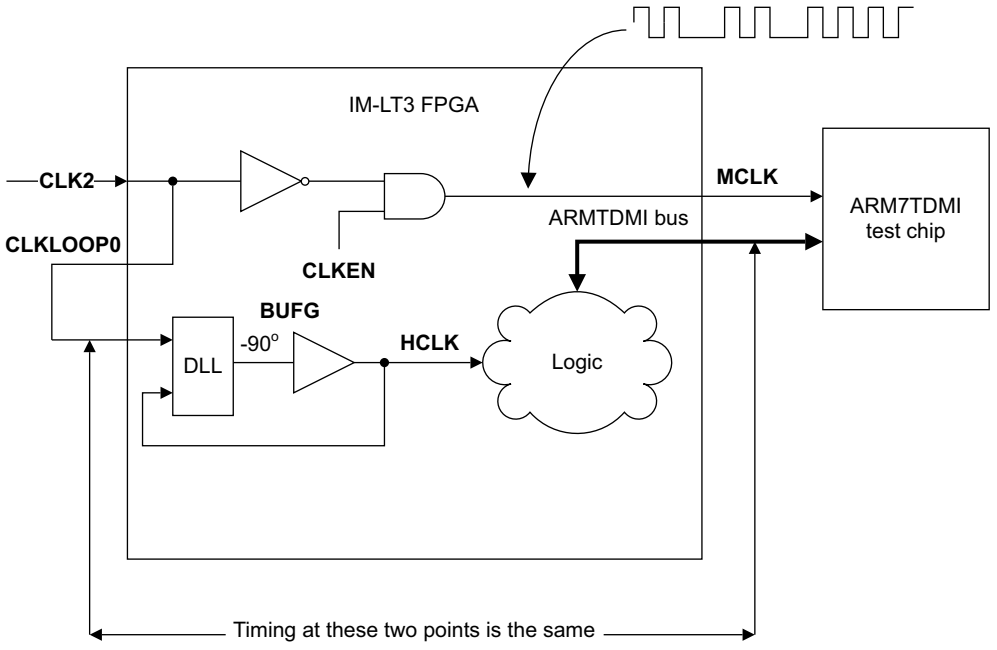

Figure 7-2 Test chip internal clock signals

7.3 Core Tile memory

This section describes the memory and memory expansion connectors present on the Core Tile.

7.3.1 Memory expansion boards

The Core Tile contains two memory connectors for expansion memory boards:

J7 Y memory expansion connector (Y MEMEXP).

J8 Z memory expansion connector (Z MEMEXP).

Note

Typically the expansion boards contain static RAM or flash memory. The memory connectors conform to the PISMO standard. Refer to the *PISMO Memory Interface Connector Specification* and Appendix A *Static Memory Expansion Board* for details on signals on the connectors and signals. Different memory boards might require special interface software or configuration settings.

The memory controller for expansion memory must be implemented in an external Logic Tile (or IM-LT3). The memory expansion signals are connected directly to HDRY and HDRZ. The Core Tile test chip is not connected to the memory expansion sockets.

To use the SSTL standard for a PISMO board on the Y MEMEXP connector, fit resistor links R26, R27, and R28.

The test chip on the CT7TDMI Core Tile can be disabled to allow the Core Tile to function as a dedicated memory tile (see *Disabling the test chip output signals* on page 7-29).

7.3.2 Memory map

All memory accesses are presented on the external 32-bit wide interface consisting of **ADDR[31:0]**, **DATA[31:0]**, and the bus control signals.

The byte order of memory can be selected by the **BIGENDIN** signal.

7.4 Clocks

Figure 7-3 shows the clock-related signals that are present on the Core Tile and an attached Logic Tile.

The selection of clock sources and outputs depends on the specific combination of Versatile products. Refer to the application notes for more detail on clock usage.

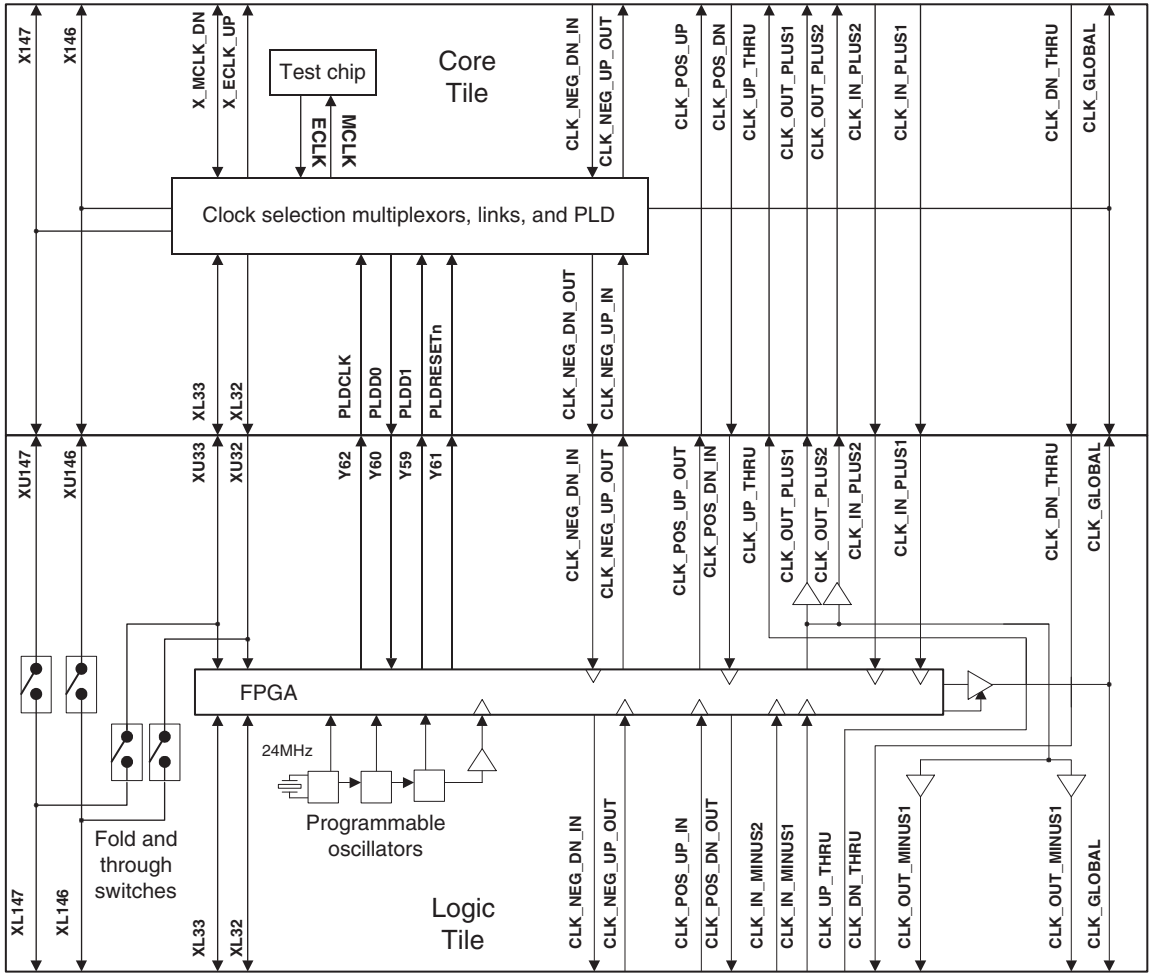

Figure 7-3 Core Tile clock signals

There is not a clock generator present on the Core Tile. The processor clock (**MCLK**) can be selected from one of the external clocks (**CLOCK_GLOBAL**, **X_MCLK_DN**, **X_MCLK_UP**, **CLK_NEG_DN_IN**, or **CLK_NEG_UP_IN**) as shown in Figure 7-4 on page 7-18. Multiplexor signals from the PLD are set from the serial configuration data.

Note

Many of the normal Logic Tile clocks are not used by the Core Tile (for example **CLK_POS_DN**). There are, however, signals input to the Core Tile from an attached Logic Tile that are not standard Logic Tile clocks (for example, **X_ECLK_DN** and **PLDCLK**). The new signals are Logic Tile FPGA outputs or inputs that are used as clocks with the Core Tile.

The resistor links shown in Figure 7-4 on page 7-18 are fitted at manufacture and do not normally require modification. For details on links, see *Links and test points* on page 8-22.

The clock sources and clock outputs might be different for different test chip variants. See the test chip information on the CD for details particular to the test chip used on the Core Tile.

For more information on clock generation and control for your test chip, see the *Technical Reference Manual* for your processor family, the test chip information supplied on the CD, and the Application Notes for using your Core Tile with other Versatile products.

The clock-related signals are summarized in Table 7-3.

Table 7-3 Clock-related signals on Core Tile

Signal	Direction	Description
MCLK	Input	Master clock for the test chip. **MCLK** is the ARM7TDMI clock input for the core and the bus, and **ECLK** is derived from it.
CLK_GLOBAL	Input/through	A global clock shared with all tiles in the stack.
CLK_NEG_DN_IN **CLK_NEG_DN_OUT**	Input/output and through	A clock signal routed from the upper header (**CLK_NEG_DN_IN**) to the lower header (**CLK_NEG_DN_OUT**).
CLK_NEG_UP_IN **CLK_NEG_UP_OUT**	Input/output and through	A clock signal routed from the lower header (**CLK_NEG_UP_IN**) to the upper header (**CLK_NEG_UP_OUT**).

Table 7-3 Clock-related signals on Core Tile (continued)

Signal	Direction	Description
CLK_POS_UP, **CLK_POS_DN**, **CLK_UP_THRU**, **CLK_OUT_PLUS1**, **CLK_OUT_PLUS2**, **CLK_IN_PLUS1**, **CLK_IN_PLUS1**, and **CLK_DN_THRU**	Through	These Logic Tile clock signals are not used by the Core Tile. They are passed through the Core Tile unmodified for use by other tiles.
CLKSEL[5:0]	Local outputs from PLD	These signals (from the PLD) control the clock-selection multiplexors.
ECLK	Output	Output clock from the test chip. The destination can be selected as either **X_ECLK_DN** or **XL33**
X_MCLK_UP, **XL32**	Input/output and through	Clock source from an external tile that is an input to the clock selection multiplexor.
X_ECLK_DN,**XL33**	Input/output and through	Clock output from test chip that is distributed to external tiles.
X146	Output and through	This signal is normally passed through the Core Tile. If **nX_FOLD1** is HIGH, however, the signal is connected to **X_MCLK_DN**.
X147	Output and through	This signal is normally passed through the Core Tile. If **nX_FOLD1** is HIGH, however, the signal is connected to **X_ECLK_UP**.

7.4.1 Clock multiplexors

Clock selection signals **CLKSEL[5:0]** and foldover control signals **nX_FOLD1** and **nX_THRU1** control the multiplexors on the Core Tile:

CLKSEL[0] This selects between clocks on the upper or lower header connectors as the source for **MCLK** and the destination for **ECLK**.

CLKSEL[2:1]

This selects the source for **MCLK**. The input is either one of the clocks on the upper or lower header connector, ground, or the buffered **CLK_GLOBAL**.

CLKSEL[3] If HIGH, the **CLK_NEG_UP_OUT** on the top header is connected to ground. If LOW, **CLK_NEG_UP_OUT** is driven by the **CLK_NEG_UP_IN** signal from the lower header on the Core Tile.

CLKSEL[4] If HIGH, the **CLK_NEG_DN_OUT** on the lower header is connected to ground. If LOW, **CLK_NEG_DN_OUT** is driven by the **CLK_NEG_DN_IN** signal from the upper header on the Core Tile.

CLKSEL[5] If HIGH, the **X_ECLK_UP** and X_MCLK_DN on the upper header are connected to the isolation switches for **XL33** and **XL32**. If **CLKSEL[5]** is LOW and **nX_THRU** is HIGH, the signals on **XL33** and **XL32** can be used by the multiplexor as alternative source for **MCLK** and destination for **ECLK**.

nX_FOLD1 Connects signals on the upper header to signals on the lower header. If HIGH, **X_ECLK_UP** and X_MCLK_DN on the upper header are connected to the **X147** and **X146** (on both the upper and lower headers).

nX_THRU1 Enables signals onto the lower header. If **CLKSEL[5]** is LOW and **nX_THRU** is HIGH, the signals on **XL33** and **XL32** are connected to the multiplexors and can be used as an alternative source for **MCLK** and destination for **ECLK**.

See *Core Tile PLD signals* on page 7-32 for details on controlling the **CLKSEL[5:0]** signals from the PLD using **PLDCLK**, **PLDD0**, and **PLDD1**. The **MCLK** and **ECLK** routing for values of **CLKSEL[5:0]** are listed in Table 7-4 and Table 7-5 on page 7-17.

Table 7-4 MCLK clock source

CLKSEL[5:0]	MCLK source
bxxx000	**CLK_NEG_UP_IN**
bxxx001	**CLK_NEG_DN_IN**
b0xx010	**X_MCLK_UP** (This signal is connected to **XL32** if **nX_THRU1** is HIGH and is floating if **nX_THRU1** is LOW)
b0xx011	**X_MCLK_DN** (If **nX_FOLD1** is HIGH **X_MCLK_DN** is also connected to **X146**.)
b1xx01x	**X_MCLK_DN** and **X_MCLK_DN** are connected together. (If **nX_THRU1** is HIGH, the signal can be sourced from either the upper or lower header. If **nX_FOLD1** is HIGH **X_MCLK_DN** is also connected to **X146**.)
bxxx100	**GND** (no clock input)
bxxx101	**GND** (no clock input)
bxxx110	**CLK_GLOBAL**
bxxx111	**CLK_GLOBAL**

Table 7-5 ECLK destination

CLKSEL[5:0]	ECLK output
b0xxxx0	**X_ECLK_UP** (If **nX_FOLD1** is HIGH **X_ECLK_UP** is connected to **X147**.)
b0xxxx1	**X_ECLK_DN** (**X_ECLK_DN** is also connected to **XL33** if **nX_THRU1** is HIGH)
b1xxxxx	Both **X_ECLK_UP and X_ECLK_DN**. (**X_ECLK_DN** is also connected to **XL33** if **nX_THRU1** is HIGH. If **nX_FOLD1** is HIGH **X_ECLK_UP** is also connected to **X147**.)

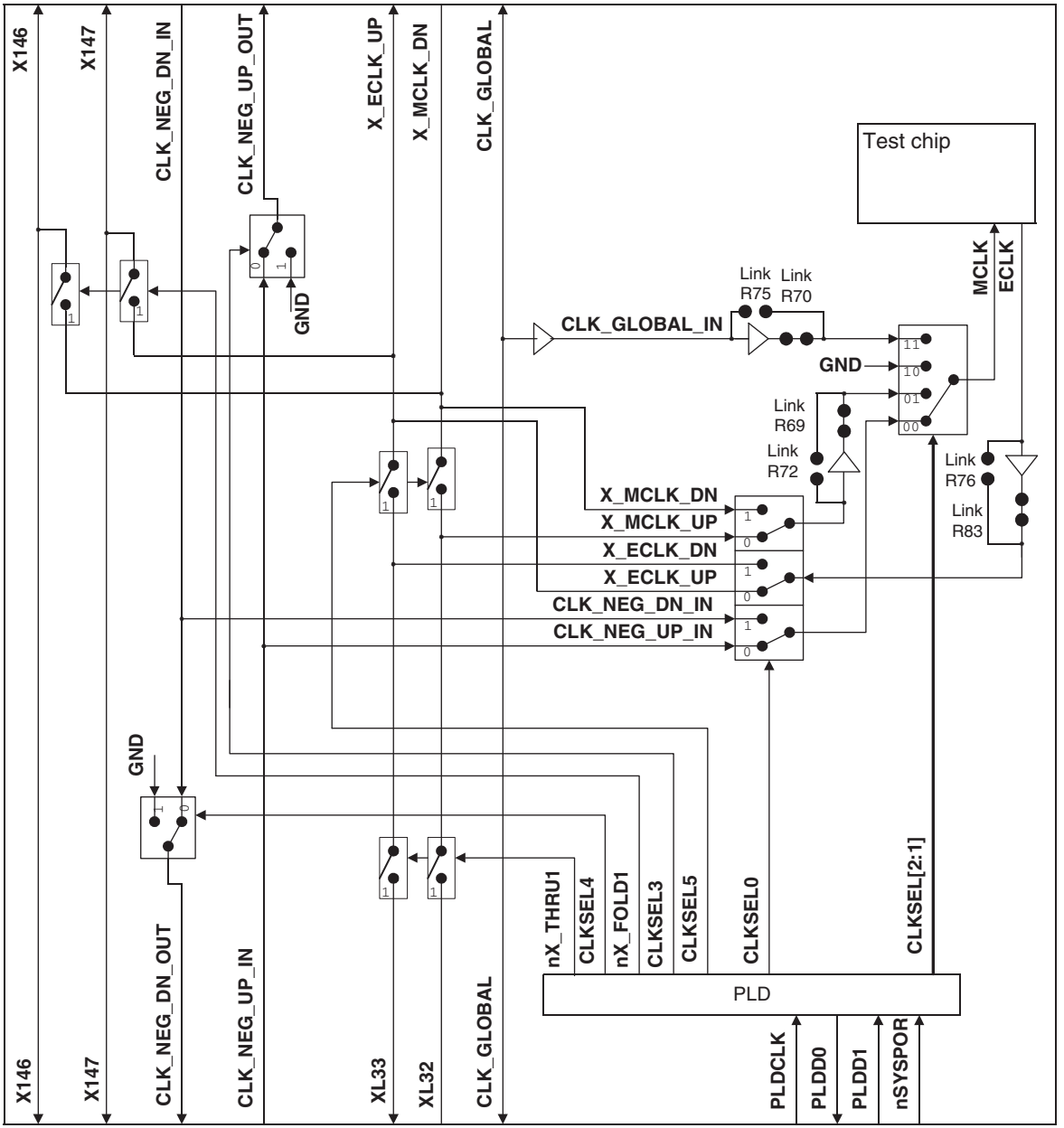

Figure 7-4 Test chip clock selection

7.5 Power supply control

The power supply on the Core Tile is controlled by the PLD and by resistor links as shown in Figure 7-5.

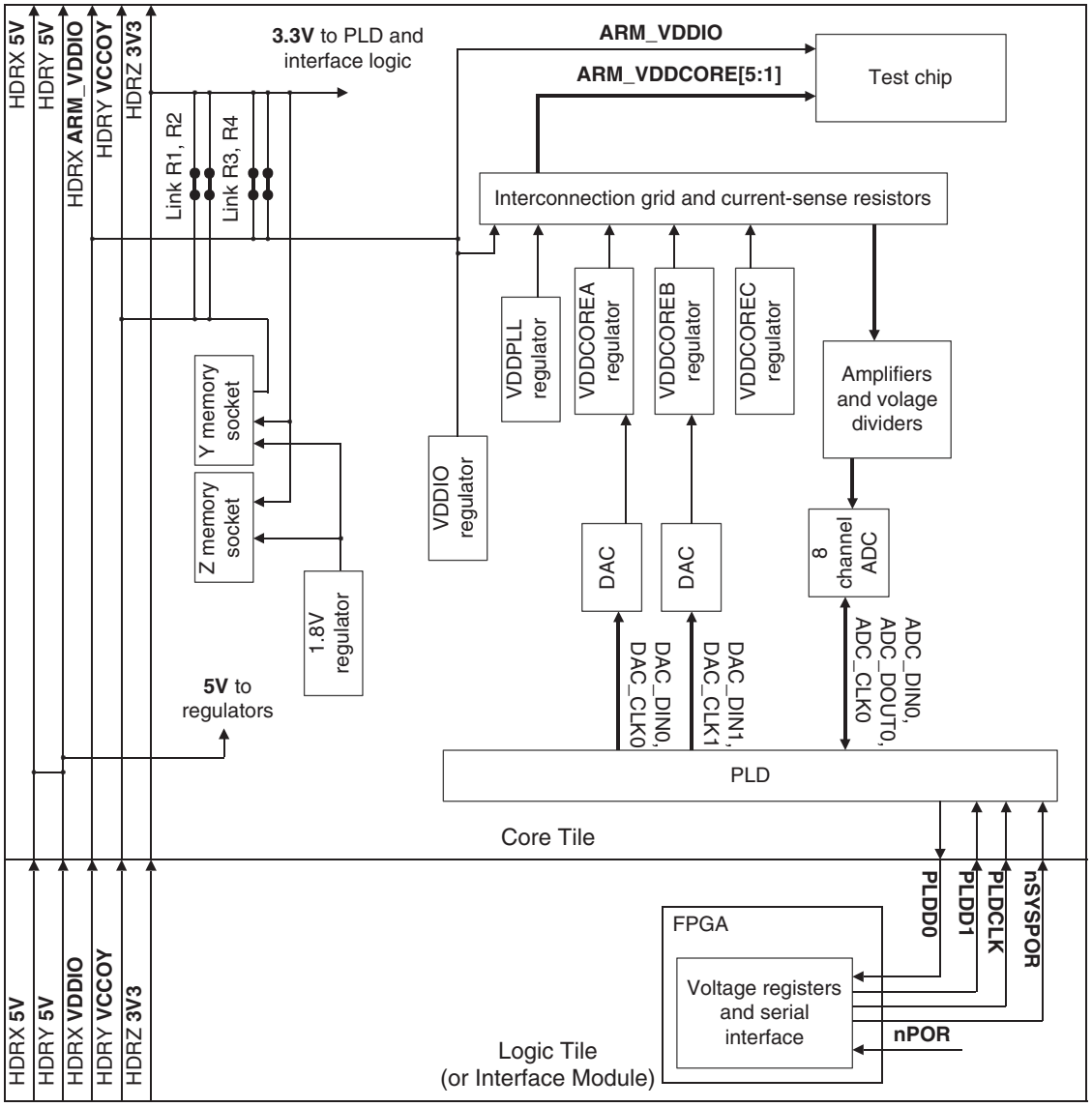

Figure 7-5 Voltage control and monitoring

Caution

Changing the resistors that control the core power-supply voltage might result in damage to the test chip (see *Setting the VDDCORE voltage* on page 7-23). The resistors that are fitted at manufacture enable you to modify the core voltage within a safe region.

Modifying the core voltage is not required for normal operation.

The resistor values and links described in this section are typical values, but the values on your Core Tile might be different. Refer to the BOM and schematic for your Core Tile for details of build options that are used for your board.

The Logic Tile FPGA implements registers and a PLD interface (as shown in Figure 7-5 on page 7-19) that control the PLD on the Core Tile to enable you to:

- Change the **VDDCOREA** and **VDDCOREB** voltages by writing values to the serially-programmed *digital to analog converters* (DACs). Resistor links allow these supplies to be connected to test chip signals **ARM_VDDCORE[5:1]**.
- Read onboard voltages and currents:
  - **ARM_VDDCORE[5:1]** and **ARM_VDDIO** are the core and I/O voltage levels
  - **VDDCOREA_DIFF** and **VDDCOREB_DIFF** are proportional to the current through current-sensing resistors
  - **TP1_SENSE** and **TP2_SENSE** are the voltages present on test points 4 and 5.

Note

The ADC controller continuously reads the values from the ADC and stores them in the registers. The serial interface between the PLD on the Core Tile and the FPGA on an attached Logic Tile continuously updates the registers in the PLD and FPGA. (See *Core Tile PLD signals* on page 7-32.)

7.5.1 Resistor links for power supply

The connections between the programmable regulators and the test chip voltage pins can be configured by resistor links on the printed circuit board as shown in Figure 7-6 on page 7-22. The Core Tile voltage sources are:

1V8 The output of this regulator is input to the memory expansion connectors.

VDDIO The output of this regulator is input to the test chip and is connected to the **ARM_VDDIO** power blade on HDRX.

Resistor links R3 and R4 connect the 3.3V supply from HDRZ with the **VDDIO** voltage blades on HDRX. Do not use this link if you require custom I/O voltages on the HDRX blade.

VDDIO can also be selected as the voltage supply for test chip signals **VDDORE2**, **VDDCORE4**, or **VDDCORE5**.

VCCOY Some memory modules have the output of an onboard regulator connected to the **VCCOY** signals. **VCCOY** is present on the power blade of HDRY.

Resistor links R1 and R2 connect the 3.3V supply from HDRZ with the **VCCOY** voltage. Do not use this link if you require custom I/O voltages on the HDRY blade.

VDDCOREA and VDDCOREB

The output of the two programmable regulators can be selected as the voltage supply for the test chip. (**VDDCOREA** can be selected for **VDDORE2**, **VDDCORE3**, or **VDDCORE5**. **VDDCOREB** can be selected for **VDDCORE[4:3]**.

VDDCORE[5:1] interconnection

There are the core voltage supplies to the test chip core. The interconnection links to connect two or more of the groups together.

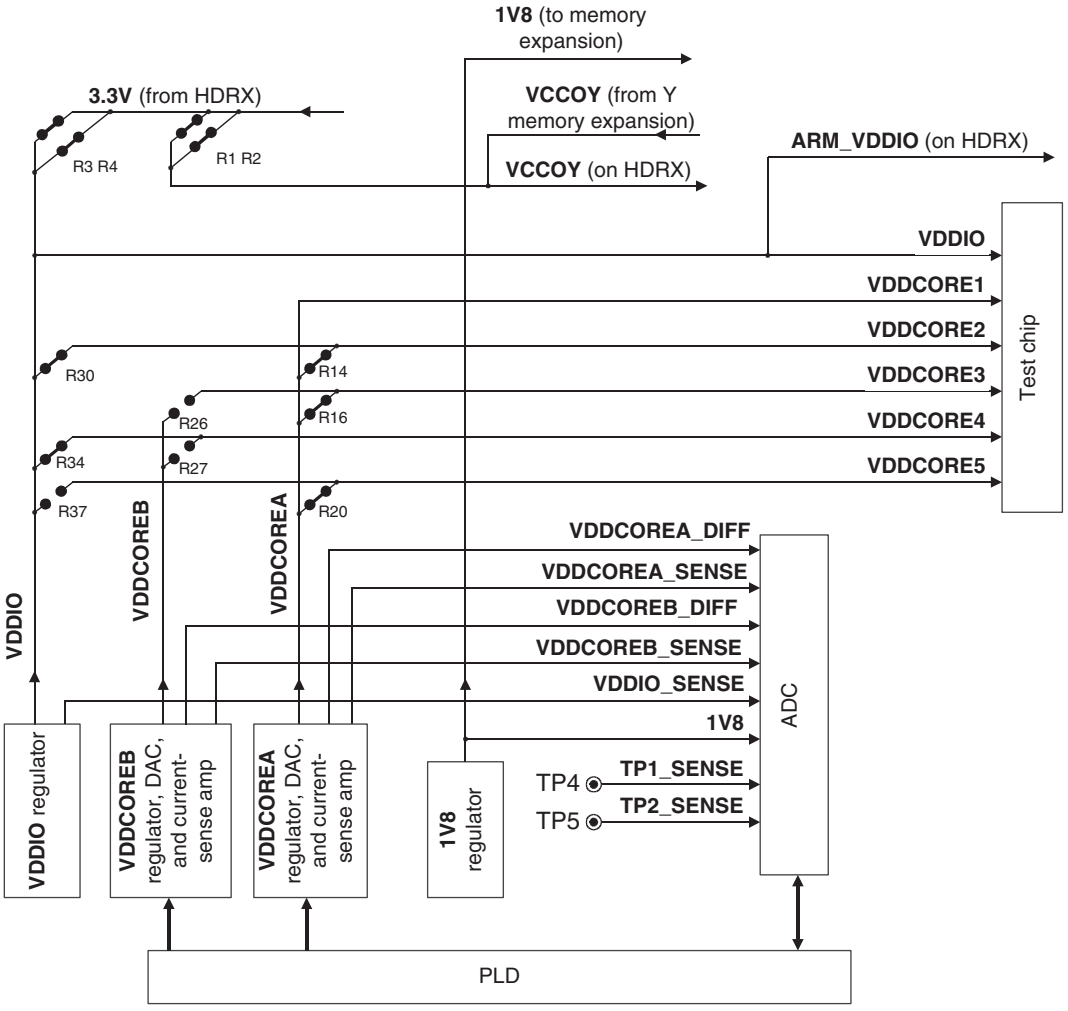

Figure 7-6 Voltage interconnection links

Note

Refer to the BOM for details of resistor links shown in Figure 7-6 that are fitted at manufacture. Also the **VDDIO** and **VDDCOREB** regulators might not be fitted for some boards. **VDDIO** is typically powered from the 3.3V supply from the HDRX power blades.

7.5.2 Setting the VDDCORE voltage

The core voltages are set by:

- Resistors R13, R18. R19, R35, R40, and R41, see Figure 7-7 on page 7-24. These define the core voltage at power-on and the range of adjustment possible for **VDDCOREA** and **VDDCOREB**. These resistors are fitted at manufacture to give the correct core voltage for the test chip fitted to the Core Tile.
- The values loaded into the PLD for the DAC settings. The values provide a positive or negative offset to the default power-on voltages.
- The interconnection links between the regulators and the **VDDCORE[5:1]** power connections (see Figure 7-6 on page 7-22).

The output voltages are given by:

VDDCOREA = 0.6V * (1 + R13/R18 + R13/R19) - CT_VOLTAGE_A[7:0]*R13*(I_{DAC}/255)

VDDCOREB = 0.6V * (1 + R13/R40 + R35/R41) - CT_VOLTAGE_B[7:0]*R13*(I_{DAC}/255)

where:

I_{DAC} The full-range DAC output current (50µA)

CT_VOLTAGEx[7:0]

The eight-bit data value loaded into the PLD from the Logic Tile FPGA. There are two DACs that drive the programmable regulators for **VDDCOREA** and **VDDCOREB**. The default value loaded into the DAC is `0x80`. A value of `0xFF` gives maximum negative offset (-0.25V) and a value of `0x0` gives maximum positive offset (+0.25V).

`Resistor values`

Refer to the BOM for the resistor values fitted to the build variant.

See *Core Tile PLD signals* on page 7-32 for a description of the PLD interface.

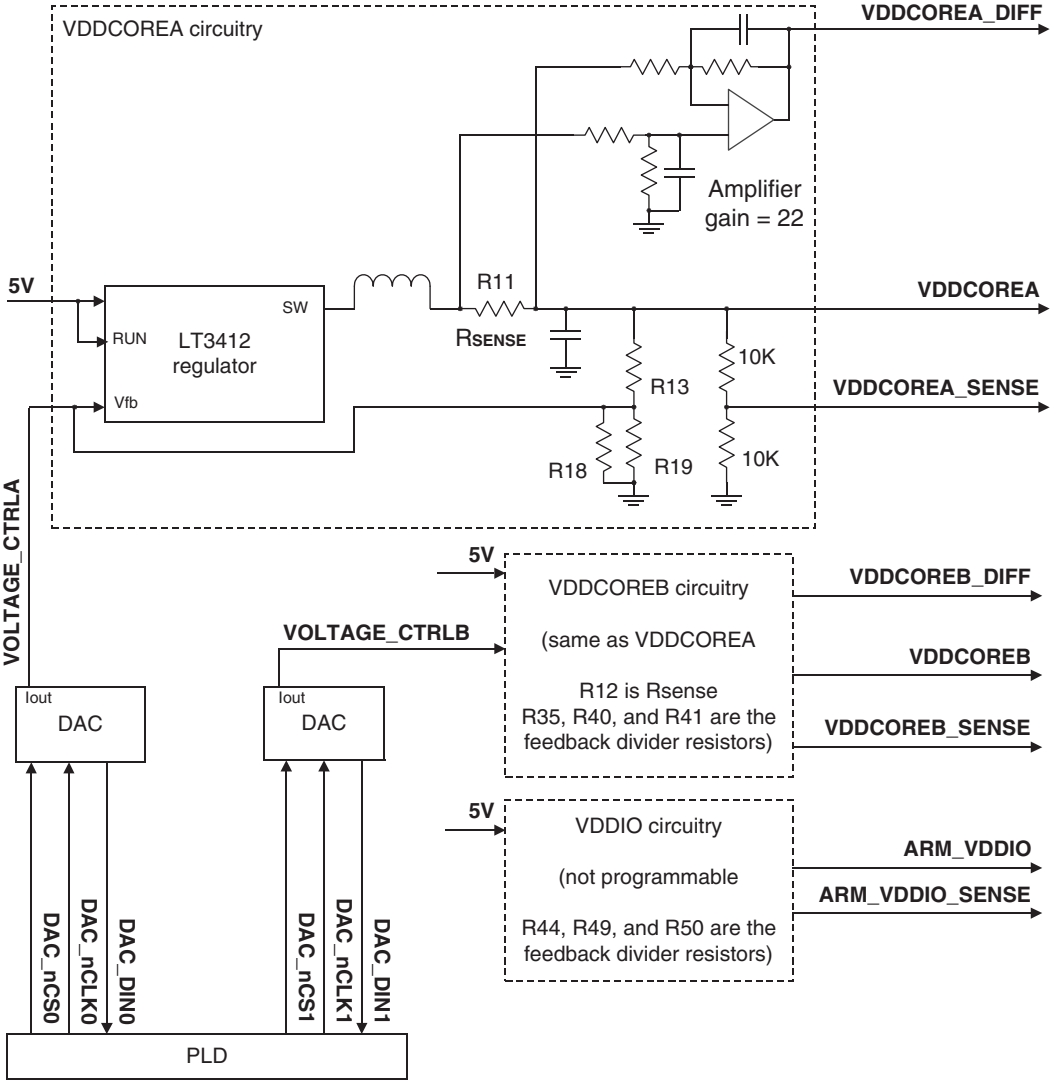

Figure 7-7 Programmable regulators

7.5.3 Reading the voltages and currents

The ADCs on the Core Tile regularly read the voltages and currents and update registers in the PLD. The PLD transmits these values over the serial interface to the FPGA in the attached Logic Tile.

The LSB of the ADC reading corresponds to 1.221 mV. (Two times the 2.5V V_{REF} of the ADC divided by 4095.) The formula for the **VDDCORE** and **VDDIO** voltages is:

V_{VDDx} = 1.221mV * VOLTAGE[11:0]

Caution

All voltages except **VDDCOREx_DIFF**, **1V8**, **ADC_VREF**, **TP1_SENSE**, and **TP2_SENSE** are divided by two before being fed to the ADC. The formula for these voltages is therefore:

V = 2.442mV * VOLTAGE[11:0]

The **TPx_SENSE** voltages should not exceed 2.5V for accurate readings and must not exceed 3.3V to prevent damage to the converters.

Voltages proportional to the **VDDCOREA** and **VDDCOREB** currents are developed across the sense resistors. Each of the sense resistors has a voltage amplifier because the sense voltage is too low to measure directly with the ADC. To calculate the current through an **VDDCOREx** line:

$I_{VDDCOREx}$ = V_{ref} * VOLTAGE[11:0] / (R_{SENSE} * GAIN * (2^{12}-1))

Note

By default, all of the R_{SENSE} resistors are 0Ω. It is not therefore possible to measure the current flow. You can replace the existing sense resistors with new ones of, for example, 1Ω.

7.6 Isolation and foldover of header signals

The Core Tile uses electronic switches to control signals to and from some of the header connectors:

- **X_MCLK_UP** can be connected to **XL[32]**.
- **X_MCLK_DN** can be connected to **X[147]**.
- **X_ECLK_DN** can be connected to **XL[33]**.
- **X_ECLK_UP** can be connected to **X[146]**.
- **SEL[1:0]** can be connected to **XL[35:34]** or **X[145:144]**.
- **DATA[31:0]** can be connected to **XL[31:0]** or **X[179:148]**.
- **YU[35:0]** can be connected to **YL[35:0]** or **Y[179:144]**.
- Signals **ZL[127:0]** can be disconnected from signals **ZU[127:0]**.

  For Core Tiles, **ZL[127:0]** signals are connected to the **ZU[127:0]** signals. The HDRZ signals can be split if, for example, there is a Logic Tile mounted on both the top and bottom of the Core Tile and the Logic Tiles have different functions for the **Zx[127:0]** signals.
- The test chip can be disabled and any header signals connected to the test chip can be used for an alternative function by tiles mounted above or below the Core Tile.

7.6.1 Through/break control for HDRX

Interconnection between some signals on HDRX of the Core Tile is controlled by signals **nX_THRU[1:0]** and **nX_FOLD[1:0]** from the Core Tile PLD (see Figure 7-8 on page 7-27). If the **nX_THRU** and **nX_FOLD** signals are LOW, the corresponding switches are closed. See *Core Tile PLD signals* on page 7-32 for a description of the PLD interface.

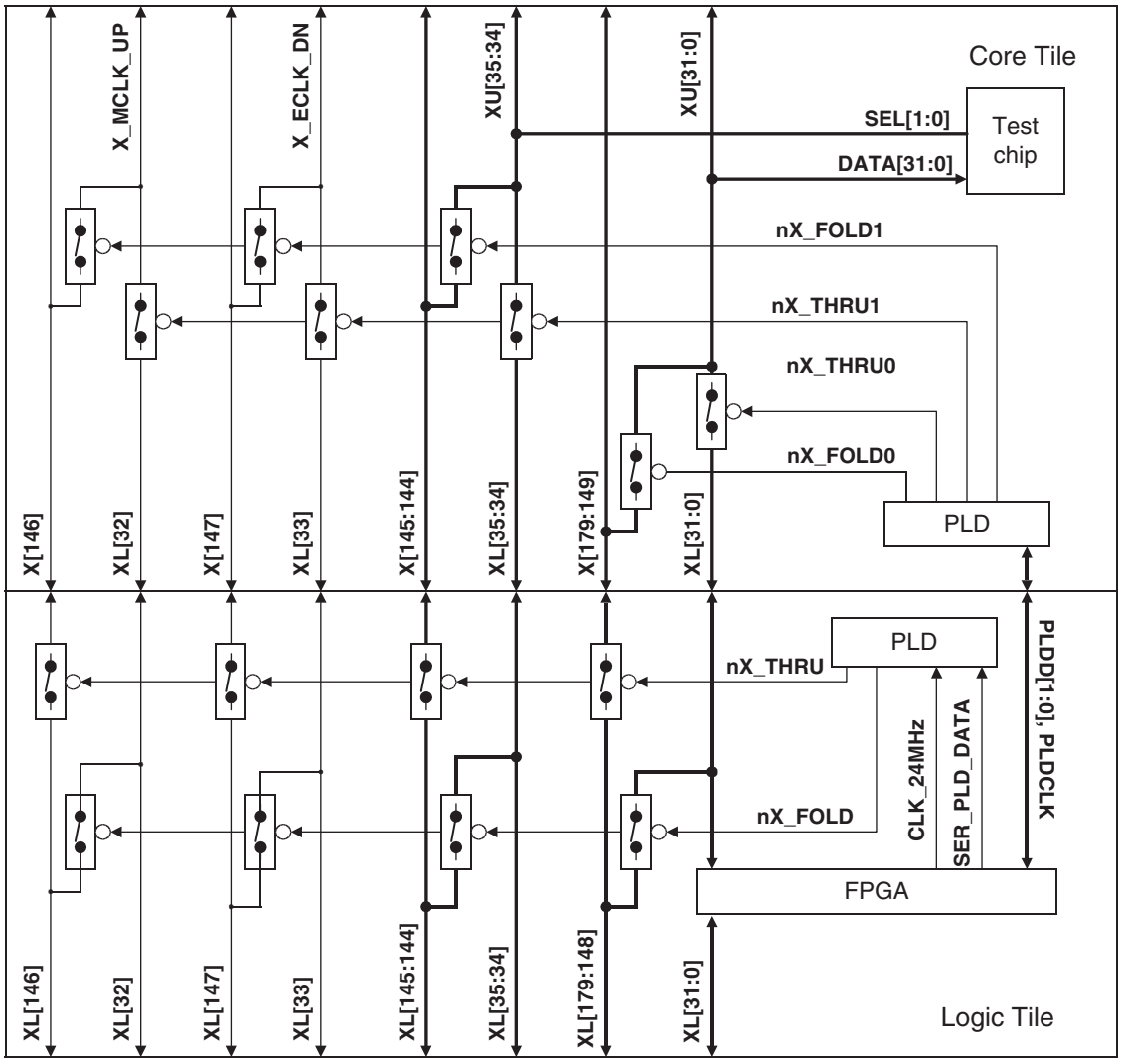

Figure 7-8 HDRX through/break control

7.6.2 Through/Break control for HDRZ

Interconnection between **ZL[127:0]** and **ZU[127:0]** is selected by control signals **ZCTL[3:0]**, Z_THRU, and Z_FOLD from the PLD as shown in Figure 7-9 on page 7-28. If the control signal is LOW, the corresponding switches are closed.

See *Core Tile PLD signals* on page 7-32 for a description of the PLD interface.

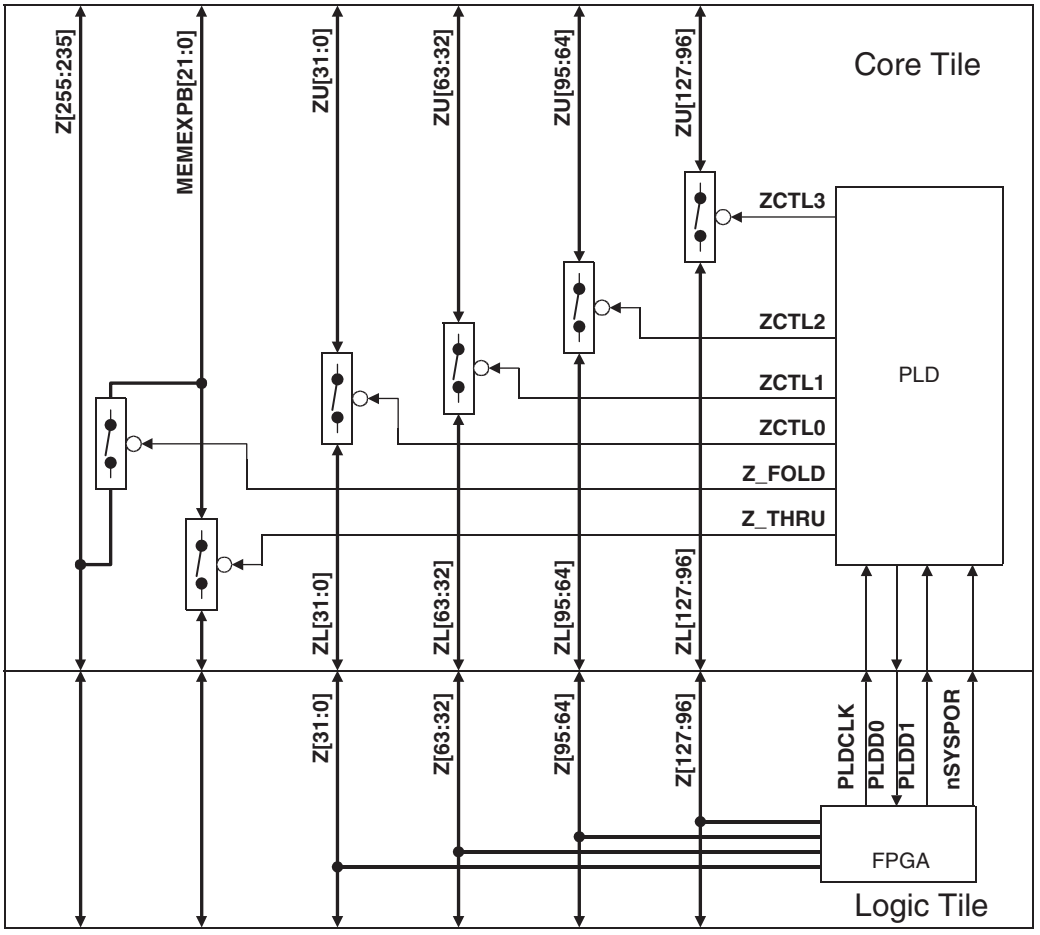

Figure 7-9 HDRZ through/break control

7.6.3 Through/Break control for HDRY

Interconnection between **Y[179:144]**, **YL[35:0]**, and **YU[35:0]** is selected by control signals **nY_FOLD[1:0]** and **nY_THRU[1:0]** from the PLD as shown in Figure 7-10 on page 7-29. If the **nY_THRU** and **nY_FOLD** signals are LOW, the corresponding switches are closed.

See *Core Tile PLD signals* on page 7-32 for a description of the PLD interface.

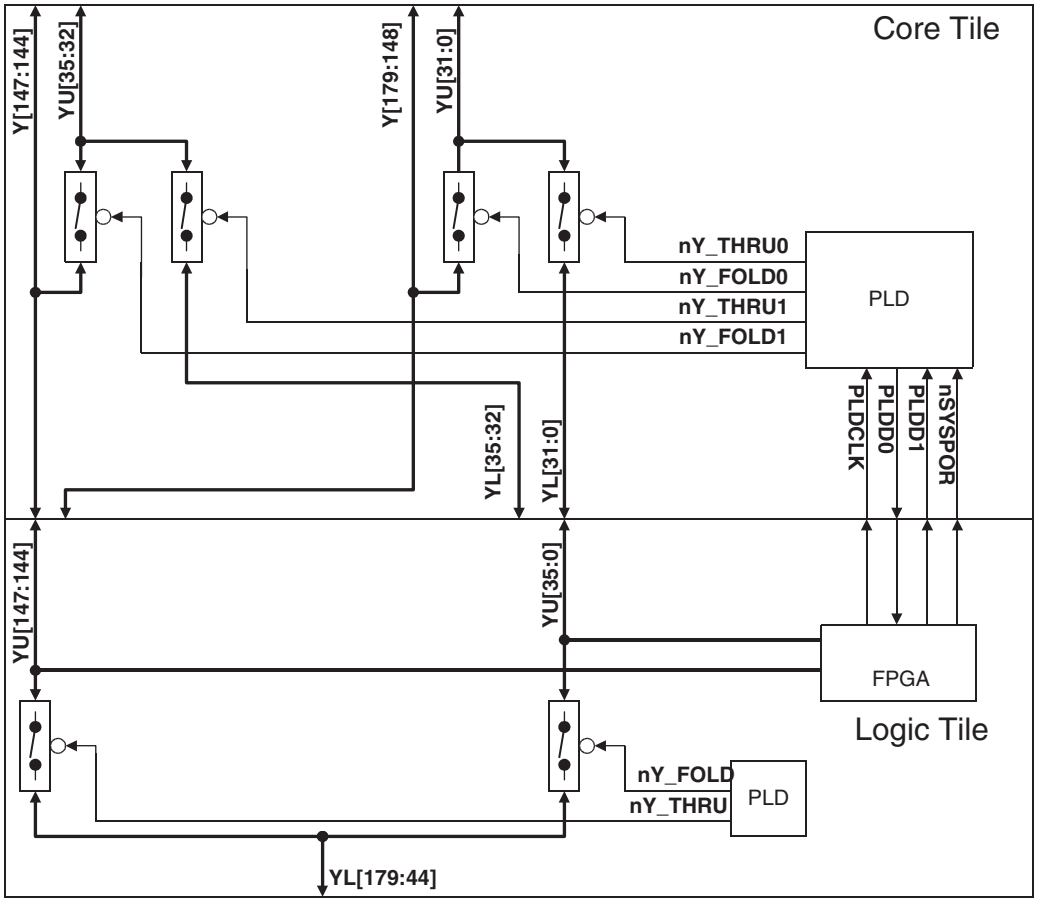

Figure 7-10 HDRY through/break control

7.6.4 Disabling the test chip output signals

The test chip on the CT7TDMI Core Tile can be disabled by driving the **BCE** signal HIGH. This tri-states all output signals from the Core Tile. Any header signals that are connected directly to the test chip can be used for other purposes by tiles connected above and below the Core Tile.

If the test chip is disabled, the PLD on the Core Tile can still be used to control header signals through the fold and thru switches and to provide access to the memory expansion connectors.

Note

The **BCE** signal is not available on the test chip used on the CT7TDMI-S Core Tile.

7.7 Overview of Core Tile configuration

The clock sources, and voltages levels on the Core Tile are configurable from input signals on the header connectors and signals output from the on-board PLD. In a production ASIC, core configuration is typically static and the core configuration signals are tied HIGH or LOW and the voltage and clocks are fixed. However, the Core Tile allows you to program these signals for experimentation.

There are several ways that Core Tile configuration occurs:

- The **BIGENDIN** signal present on HDRX selects memory byte order.

  When an external Logic Tile drives **BIGENDIN** HIGH, the processor uses big-endian memory organization. The signal typically connects to a Logic Tile (or IM-LT3 Interface Module) connected to the Core Tile.

- Configuration by the serial interface to the PLD on the Core Tile. (The PLD is normally controlled by a serial interface implemented in an FPGA present on an external Logic Tile.) The PLD controls the following signals:
  - — the DAC inputs that select the voltages for the programmable regulators
  - — the data bus interface switches
  - — the clock multiplexors
  - — tri-stating all output signals from the test chip
  - — the header isolation and fold switches.

  PLD configuration signals are described in *Core Tile PLD signals* on page 7-32.

- Registers in coprocessor 15 (if present for the test chip variant) are typically used to control endian settings and memory management units.

  See the *Technical Reference Manual* for the core used in your test chip for details of register usage.

  Note

  The ARM7TDMI does not have a coprocessor 15. Configuration is done by signals on the test chip.

  You can use an external JTAG unit such as Multi-ICE or RealView ICE to modify the contents of the test chip registers or internal memory. The contents can of course also be modified by an application program.

- Links present on the printed circuit board. The links, usually 0Ω. resistors, are normally set at manufacture to select voltage and clock options. These links are normally not modified after manufacture.

  Some of the functions performed by links are:
  - — voltage range for the programmable regulators

— clock selected for connection to multiplexors
— manufacturing identification number
— voltage source for header and memory connectors.

Caution

The resistor links are set at manufacture and do not normally require modification. Changing the voltage regulator resistors can result in excessive voltages being applied to the test chip.

To change the configuration of the processor, program the appropriate values in the control registers implemented in the FPGA on the Logic Tile (or IM-LT3). Depending on the test chip present in the Core Tile, the contents of registers in the test chip might require modification. (See the application note for your product configuration for details on the control registers.)

7.7.1 Core Tile PLD signals

The PLD on the Core Tile performs the following functions:

- loading data to the DACs that control the programmable power supplies (see *Power supply control* on page 7-19)
- reading data from the ADCs that monitor the test chip voltages (see *Reading the voltages and currents* on page 7-24)
- controlling the clock selection multiplexors (see *Clocks* on page 7-13)
- isolating HDRZ signals (see *Through/Break control for HDRZ* on page 7-27)
- tri-stating all of the test chip outputs (see *Disabling the test chip output signals* on page 7-29)
- selecting the test chip interface to the **DATA[31:0]** signals (see *Isolation and foldover of header signals* on page 7-26)

The PLD is controlled by the serial interface signals listed in Table 7-6 on page 7-33. These signals typically connect to an attached Logic Tile or IM-LT3 Interface Module. The FPGA in the external tile contains registers that hold the values to send to the PLD and received values from the PLD. The FPGA also provides the serialization and deserialization logic required by interface

Table 7-6 PLD control signals

Signal	Description
PLDCLK	Clocks data into or out of the PLD
PLDD1	Serial data input to PLD
PLDD0	Serial data output from PLD
PLDRESETn	Reset selects mode for PLD (LOW is startup, HIGH is runtime configuration)

The Core Tile PLD manages the Core Tile configuration and status signals listed in Table 7-7

Table 7-7 PLD configuration signals

Signal	Direction	Function
ZCTL[3:0]	PLD output	Z Through control (see *Through/Break control for HDRZ* on page 7-27)
CLKSEL[4:0]	PLD output	Clock selection (see *Clocks* on page 7-13)
DACnCS[1:0]	PLD output	Chip select to DAC
DAC_DIN[1:0]	PLD output	Data to DAC (see *Setting the VDDCORE voltage* on page 7-23)
ADC_nCS	PLD output	Chip select to ADC
ADC_CLK	PLD input	Clock for data from ADC
ADC_SSTRB	PLD input	Strobe for ADC. Indicates that a conversion has finished
ADC_DOUT	PLD input	Data from ADC (see *Reading the voltages and currents* on page 7-24)
ADC_DIN	PLD output	Data to ADC (see *Reading the voltages and currents* on page 7-24)
nX_FOLD[1:0], **nX_THRU[1:0]**	PLD output	Signal passthrough and fold switches for HDRX (see *Through/break control for HDRX* on page 7-26)
nY_FOLD[1:0], **nY_THRU[1:0]**	PLD output	Signal passthrough and fold switches for HDRY (see *Through/Break control for HDRY* on page 7-28)
nZ_FOLD, **nZ_THRU**	PLD output	Signal passthrough and fold switches for HDRZ (see *Through/Break control for HDRZ* on page 7-27)

Table 7-7 PLD configuration signals (continued)

Signal	Direction	Function
MAN_ID[3:0]	PLD input	Board identification from resistor links. This is set at manufacture and identifies the board build.
PWR_nSHDN[2:0]	PLD output	Shutdown to the Vdd core power supplies (see *Power supply control* on page 7-19)
BCE, **ABE**, **EEBE**, **EDBE**, **EABE**, **TBE**,	PLD output	Disable test chip by tri-stating all output pins (see *Disabling the test chip output signals* on page 7-29)

PLD function at power on

At power-on-reset, a controller in an external tile sends a configuration sequence to the PLD on the Core Tile as shown in Figure 7-11. The PLD control signals are described in Table 7-6 on page 7-33.

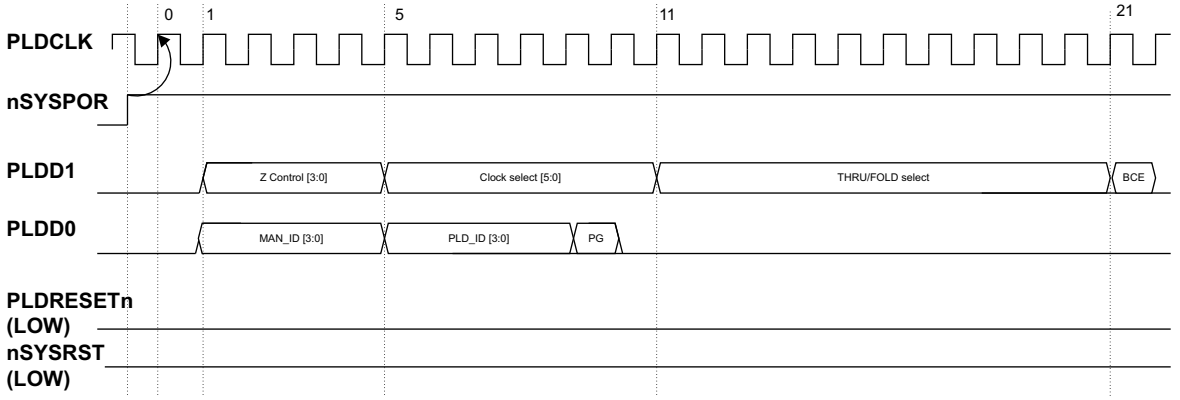

Figure 7-11 Power on signals to the Core Tile PLD

Table 7-8 lists the configuration and status signals that are received by or transmitted from the PLD at power on (after **nSYSPOR** goes HIGH).

Table 7-8 Power-on configuration signals by clock cycle

Clock	PLDD1	PLDD0	Description
-	LOW	LOW	No activity while **nSYSPOR** is LOW
0	LOW	LOW	First rising edge of clock after **nSYSPOR** goes HIGH
1	**ZCTL[3]**	**MANID[3]**	**ZCTL** data is clocked into the PLD and the state of the MANID links is clocked out of the PLD.
2	**ZCTL[2]**	**MANID[2]**	
3	**ZCTL[1]**	**MANID[1]**	
4	**ZCTL[0]**	**MANID[0]**	
5	**TCENABLE**	**PLDID[3]**	**TCENABLE** collectively controls the test chip bus output enables: **ABE** Address bus enable **BCE** BC external output enable **EEBE** External enable **EDBE** External data bus enable **EABE** Extrnal address bus enable **TBE** Test bus enable
6	**CLKSEL[4]**	**PLDID[2]**	**CLKSEL** multiplexer data is clocked into the PLD and the internal PLD ID is clocked out.
7	**CLKSEL[3]**	**PLDID[1]**	
8	**CLKSEL[2]**	**PLDID[0]**	
9	**CLKSEL[1]**	**PGOOD**	
10	**CLKSEL[0]**	–	
11	**nX_THRU[1]**	–	
12	**nX_THRU[0]**	–	
13	**nX_FOLD[1]**	–	
14	**nX_FOLD[0]**	–	
15	**nY_THRU[1]**	–	

Table 7-8 Power-on configuration signals by clock cycle (continued)

Clock	PLDD1	PLDD0	Description
16	**nY_THRU[0]**	–	
17	**nY_FOLD[1]**	–	
18	**nY_FOLD[0]**	–	
19	**nZ_THRU**	–	
20	**nZ_FOLD**	–	
21	–	–	
22	LOW	LOW	No further activity until **nSYSRST** goes HIGH

PLD function after power on

After the power-on configuration finishes, the PLD is inactive until the **nSYSRST** signal goes HIGH. If **nSYSRST** is HIGH, The PLD continuously receives DAC packets from the serial interface and transmits ADC packets to the serial interface as shown in Figure 7-12.

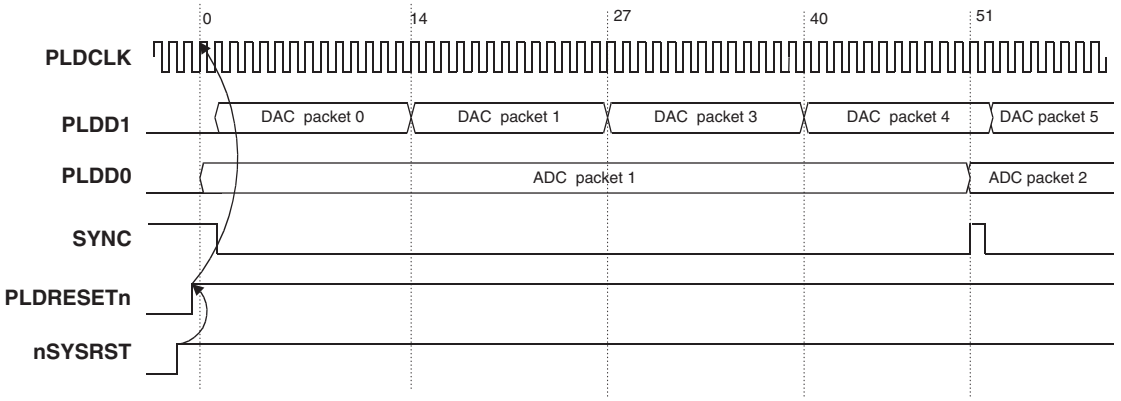

Figure 7-12 ADC and DAC data stream

Note

After **nSYSRST** goes HIGH, the ADC and DAC data packets repeat continuously.

nSYSRST is an external signal from an attached motherboard.

The format of the ADC packet is shown in Figure 7-13.

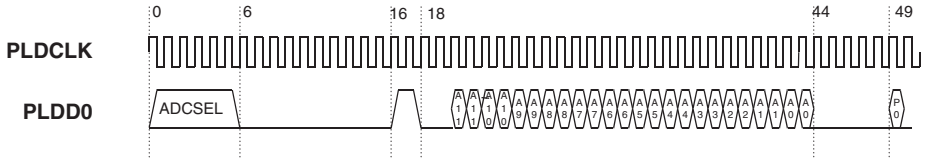

Figure 7-13 ADC packet format

The format of the DAC packet is shown in Figure 7-14.

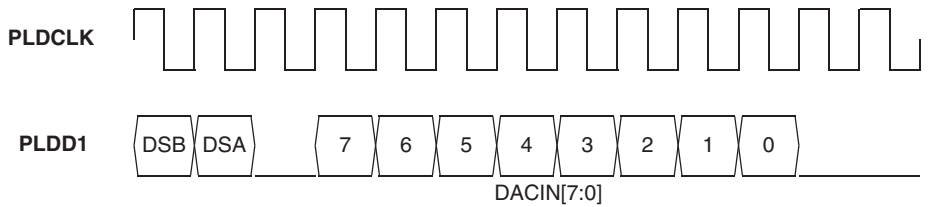

Figure 7-14 DAC packet format

Table 7-9 lists the PLD input and output signals after **nSYSPOR** goes HIGH.

Table 7-9 Run configuration signals by clock cycle

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
-	LOW	LOW	No activity while **nSYSPOR** is LOW
0	-	**ADCSELA[2]**	First cycle after **nSYSPOR** goes HIGH. The **DACSEL** signals select which DAC value is transmitted in the current packet. The value of the **ADCSEL** signals indicate the ADC value that is transmitted in the current packet.
1	**DACSELA[1]**	**ADCSELA[2]**	
2	**DACSELA[0]**	**ADCSELA[1]**	
3	LOW	**ADCSELA[1]**	
4	**DACDIN[7]**	**ADCSELA[0]**	Start of DAC data.
5	**DACDIN[6]**	**ADCSELA[0]**	

Table 7-9 Run configuration signals by clock cycle (continued)

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
6	**DACDIN[5]**	LOW	
7	**DACDIN[4]**	LOW	
8	**DACDIN[3]**	LOW	
9	**DACDIN[2]**	LOW	
10	**DACDIN[1]**	LOW	
11	**DACDIN[0]**	LOW	End of first input packet
12	-	LOW	
13	LOW	LOW	
14	**DACSELA[1]**	LOW	Second DAC packet starts with DAC selection bits.
15	**DACSELA[0]**	LOW	
16	LOW	HIGH	
17	**DACDIN[7]**	HIGH	DAC data starts.
18	**DACDIN[6]**	LOW	
19	**DACDIN[5]**	LOW	
20	**DACDIN[4]**	**ADC_DOUTA[11]**	ADC data starts.
21	**DACDIN[3]**	**ADC_DOUTA[11]**	
22	**DACDIN[2]**	**ADC_DOUTA[10]**	
23	**DACDIN[1]**	**ADC_DOUTA[10]**	
24	**DACDIN[0]**	**ADC_DOUTA[9]**	
25	-	**ADC_DOUTA[9]**	
26	LOW	**ADC_DOUTA[8]**	
27	**DACSELA[1]**	**ADC_DOUTA[8]**	DAC data ends.
28	**DACSELA[0]**	**ADC_DOUTA[7]**	
29	LOW	**ADC_DOUTA[7]**	

Table 7-9 Run configuration signals by clock cycle (continued)

Clock	PLDD1 (PLD input)	PLDD0 (PLD output)	Description
30	**DACDIN[7]**	**ADC_DOUTA[6]**	Third DAC packet starts.
31	**DACDIN[6]**	**ADC_DOUTA[6]**	
32	**DACDIN[5]**	**ADC_DOUTA[5]**	
33	**DACDIN[4]**	**ADC_DOUTA[5]**	
34	**DACDIN[3]**	**ADC_DOUTA[4]**	
35	**DACDIN[2]**	**ADC_DOUTA[4]**	
36	**DACDIN[1]**	**ADC_DOUTA[3]**	Start of DAC data
37	**DACDIN[0]**	**ADC_DOUTA[3]**	
38	-	**ADC_DOUTA[2]**	
39	LOW	**ADC_DOUTA[2]**	
40	**DACSELA[1]**	**ADC_DOUTA[1]**	
41	**DACSELA[0]**	**ADC_DOUTA[1]**	
42	LOW	**ADC_DOUTA[0]**	
43	**DACDIN[7]**	**ADC_DOUTA[0]**	Fourth DAC packet starts. End of ADC data.
44	**DACDIN[6]**	LOW	
45	**DACDIN[5]**	LOW	
46	**DACDIN[4]**	LOW	
47	**DACDIN[3]**	LOW	
48	**DACDIN[2]**	LOW	
49	**DACDIN[1]**	**PGOOD**	Power good (always HIGH)
50	**DACDIN[0]**	LOW	End of ADC packet.

7.8 JTAG support

JTAG signals are present on both the upper and lower HDRZ connectors. An external board provides the JTAG connector and the routing of the JTAG signals from the connector to HDRZ (see *JTAG signals*). The Core Tile routes the JTAG scan path through devices on the board. The logic devices that are placed in the Core Tile scan chain depend on the JTAG mode:

Debug mode Debug mode is selected by default (when a jumper is *not* fitted at the CONFIG link on the external board. It is the mode used for general system development and debug. In this mode, the JTAG signals flow through the debug scan chain (this is typically just connects to the system microprocessor).

Configuration mode

In configuration mode, all FPGAs and PLDs in the system are placed into the configuration scan chain. If the test chip implements a boundary scan chain, it is also placed in the scan chain. This mode allows the programmable logic devices in the system to be reprogrammed in the field.

To select configuration mode, fit a jumper to the CONFIG link on either the Interface Module or baseboard. This pulls the **nCFGEN** signal LOW on the system and reroutes the JTAG scan path.

7.8.1 JTAG signals

There are two separate JTAG paths through the Core Tile:

- One is used for configuration of programmable devices (the PLD for example). If the test chip has a boundary scan interface, this is included in the scan chain. These JTAG signals are identified by the **C_** prefix.
- One is used in debug mode to connect to the debug TAP controller in the test chip. 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 Core Tile connectors.

Table 7-10 on page 7-41 provides a description of the JTAG signals.

Note

In the description in Table 7-10, the term JTAG equipment refers to any hardware that can drive the JTAG signals to devices in the scan chain. Typically, Multi-ICE or RealView ICE is used, although you can also use hardware from third-party suppliers to debug ARM processors.

Note

Single-pole switches are closed if a HIGH signal is present at the control input. Multi-pole switch positions are marked with a `1` for the condition where a HIGH signal is present at the control input.

For Figure 7-15 on page 7-44, **nCFGEN** and **nTILEDET** are both HIGH indicating normal debug mode and no tile connected above the Core Tile.

Table 7-10 JTAG signal description

Name	Description	Function
DBGRQ	Debug request (from JTAG equipment)	**DBGRQ** is a request for the processor core to enter the debug state. This is not a JTAG signal, but it can be controlled by JTAG equipment.
DBGACK	Debug acknowledge (to JTAG equipment)	**DBGACK** indicates to the debugger that the processor core has entered debug mode. This is not a JTAG signal, but it can be controlled by JTAG equipment.
nCFGEN	Configuration enable (from CONFIG link on Interface Module or baseboard)	**nCFGEN** is an active LOW signal used to put the boards into configuration mode. In configuration mode all FPGAs and PLDs in the system are connected to the scan chain so that they can be configured by the JTAG equipment.
nRTCKEN	Return **TCK** enable (from Core Tile to motherboard)	**nRTCKEN** is an active LOW signal driven by any Core Tile that requires **RTCK** to be routed back to the JTAG equipment. If **nRTCKEN** is HIGH, the motherboard drives **RTCK** LOW. If **nRTCKEN** is LOW, the motherboard drives the **TCK** signal back to the JTAG equipment.

Table 7-10 JTAG signal description (continued)

Name	Description	Function
RTCK, **D_RTCK**	Return TCK for the debug scan chain (to JTAG equipment)	Some devices sample **TCK** (for example a synthesizable core with only one clock), and this has the effect of delaying the time at which a component actually captures data. **RTCK** is a mechanism for returning the sampled clock to the JTAG equipment, so that the clock is not advanced until the synchronizing device captured the data. In *adaptive clocking mode*, Multi-ICE is required to detect an edge on **RTCK** before changing **TCK**. In a multiple device JTAG chain, the **RTCK** output from a component connects to the **TCK** input of the down-stream device. The **RTCK** signal on the header connector returns **TCK** to the JTAG equipment. If there are no synchronizing components in the scan chain then it is unnecessary to use the **RTCK** signal and it is connected to ground on the motherboard. **D_RTCK** the **RTCK** signal for the debug scan chain. The config scan chain does not use **RTCK**.
nSRST	System reset (bidirectional)	**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 a board has been reset by the user. The open collector **nSRST** reset signal can be driven LOW by the reset controller on the Core Tile to cause the motherboard to reset the whole system by driving the motherboard signal **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.
nTRST	Test reset (from JTAG equipment)	This active LOW open-collector signal is used to reset the JTAG port and the associated debug circuitry on the processor. It is asserted at power-up by each module, and can be driven by the JTAG equipment. This signal is also used in configuration mode to control the programming pin (**nPROG**) on FPGAs.
TCK, **D_TCK**, **C_TCK**	Test clocks for the debug and config scan chains (from JTAG equipment)	**TCK** synchronizes all JTAG transactions. **TCK** connects to all JTAG components in the scan chain. Series termination resistors are used to reduce reflections and maintain good signal integrity. **TCK** flows down the stack of modules and connects to each JTAG component. However, if there is a device in the scan chain that synchronizes **TCK** to some other clock, then all down-stream devices are connected to the **RTCK** signal on that component (see **RTCK**). **D_TCK** is the clock for the debug mode chain and **C_TCK** is the clock for the configuration mode chain.

Table 7-10 JTAG signal description (continued)

Name	Description	Function
TDI, **D_TDI**, **C_TDI**	Test data in for the debug and config scan chains (from JTAG equipment)	**TDI** goes up the stack of modules to the motherboard and then back down the stack, labeled **TDO**, connecting to each component in the scan chain. **D_TDI** is the data signal for the debug mode chain and **C_TDI** is the data signal for the configuration mode chain.
TDO, **D_TDO**, **C_TDO**	Test data out for the debug and config scan chains (to JTAG equipment)	**TDO** is the return path of the data input signal **TDI**. The JTAG components are connected in the return path so that the length of track driven by the last component in the chain is kept as short as possible. **D_TDO** is the data signal for the debug mode chain and **C_TDO** is the data signal for the configuration mode chain.
TMS	Test mode select (from JTAG equipment)	**TMS** controls transitions in the tap controller state machine. **TMS** connects to all JTAG components in the scan chain.

JTAG data and clock logic

Figure 7-15 on page 7-44 shows the JTAG clock and data signals.

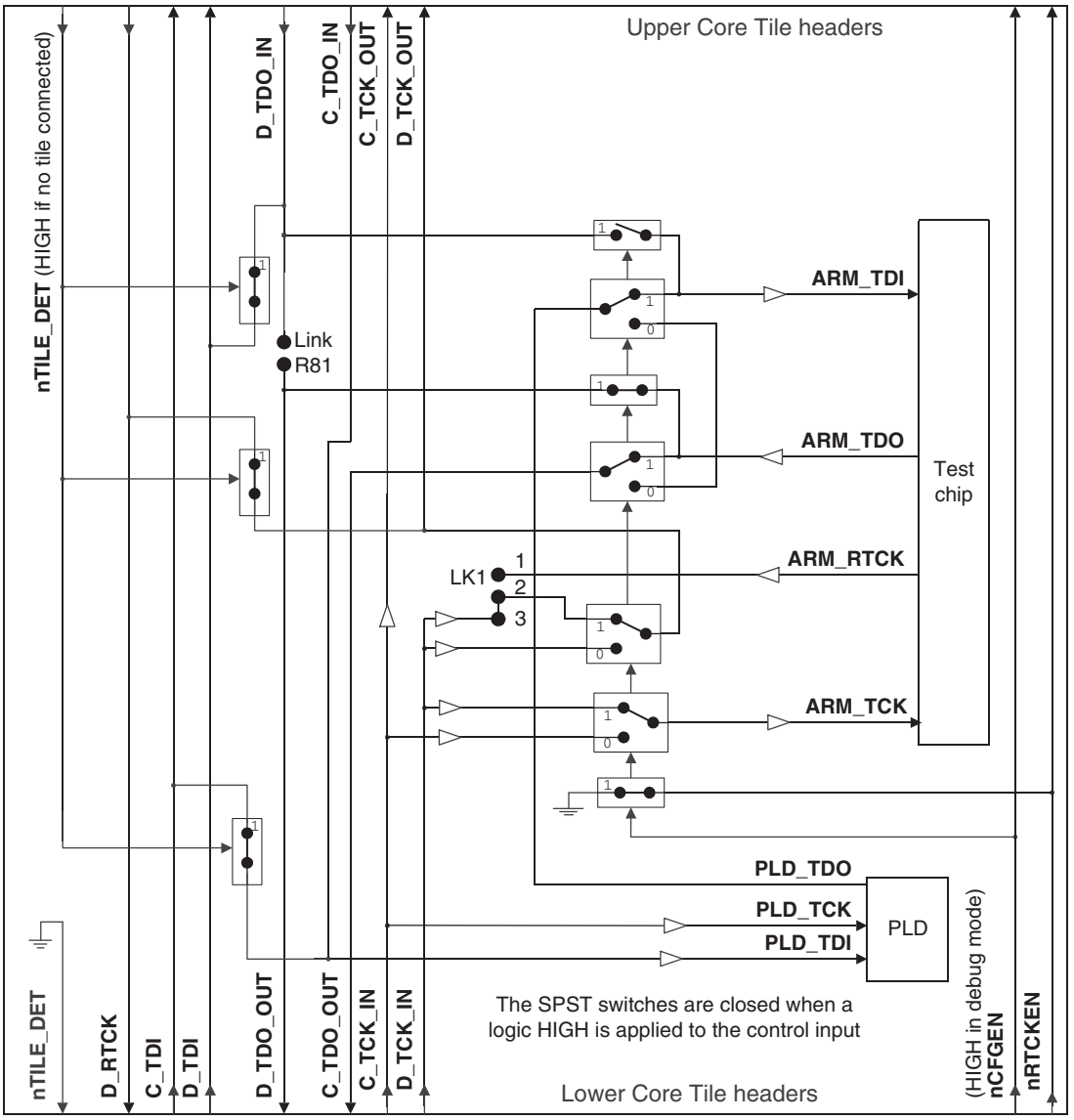

Figure 7-15 JTAG clock and data signals

Note

Unlike the ARM7TDMI-S, the ARM7TDMI does not have a separate boundary scan chain and a BS_TAP controller. The resistor links R65, R74, R77, and R81 allow both test chips to be fitted to the PCB. The resistor links route the TCK and TDI signals depending on the test chip fitted.

Figure 7-16 shows a simplified clock and data diagram for a ARM7TDMI-S in debug mode without a tile connected above the Core Tile. The two buffers in the ARM_TCK signal consist of a level-shifter that produces **D_TCK_ARM** from **D_TCK_IN**. After passing through a multiplexor, this signal is buffered again to become **ARM_TCK**.

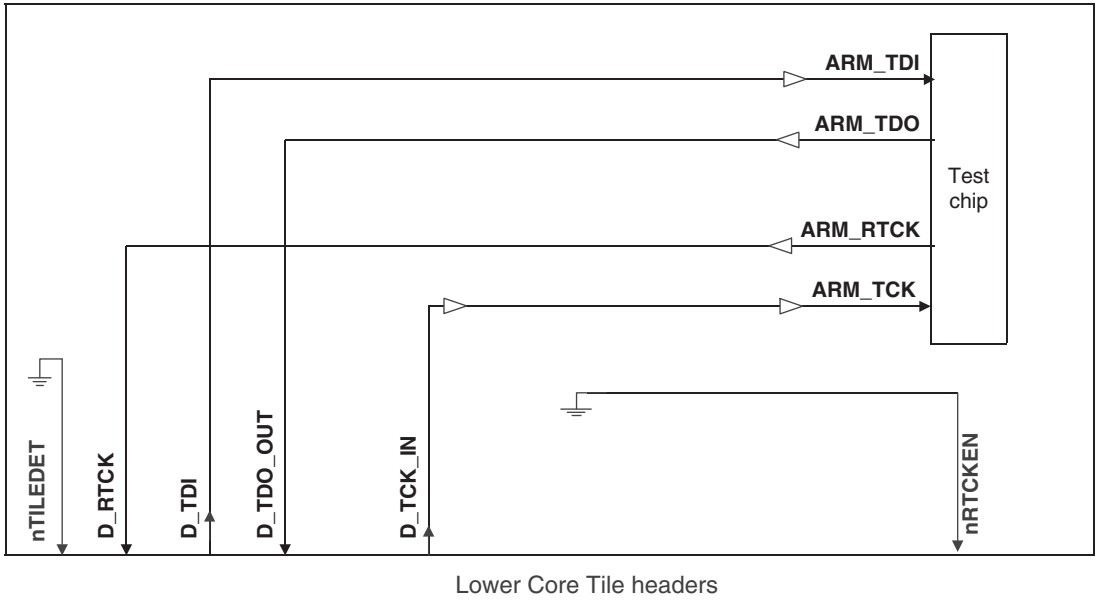

Figure 7-16 Simplified data and clocks for debug mode (ARM7TDMI-S)

The clock and data flow in debug and config mode are different. Figure 7-17 on page 7-46 and Figure 7-18 on page 7-46 show the equivalent flow for each mode. The switches have been replaced by direct connections and the diagram simplified. There is not a tile present above the Core Tile (**nTILEDET** is pulled HIGH).

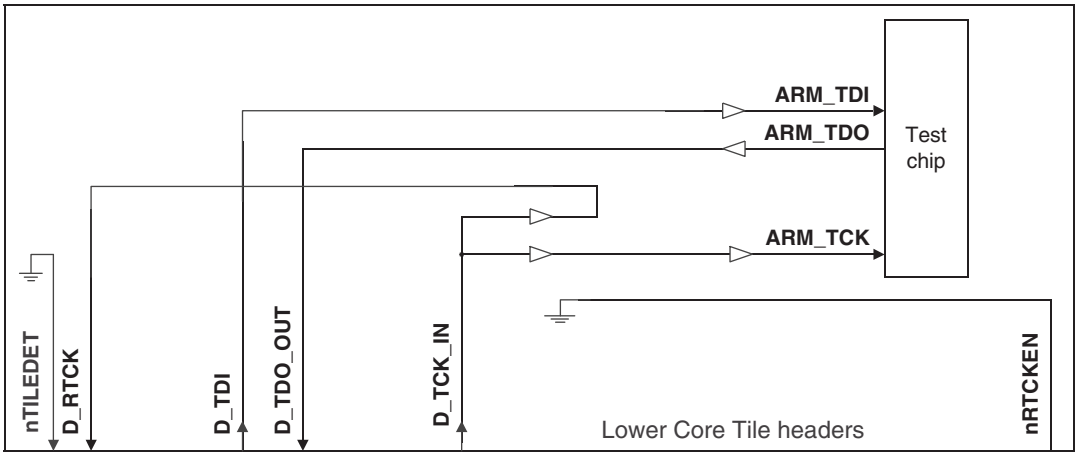

Figure 7-17 Simplified data and clocks for debug mode (ARM7TDMI)

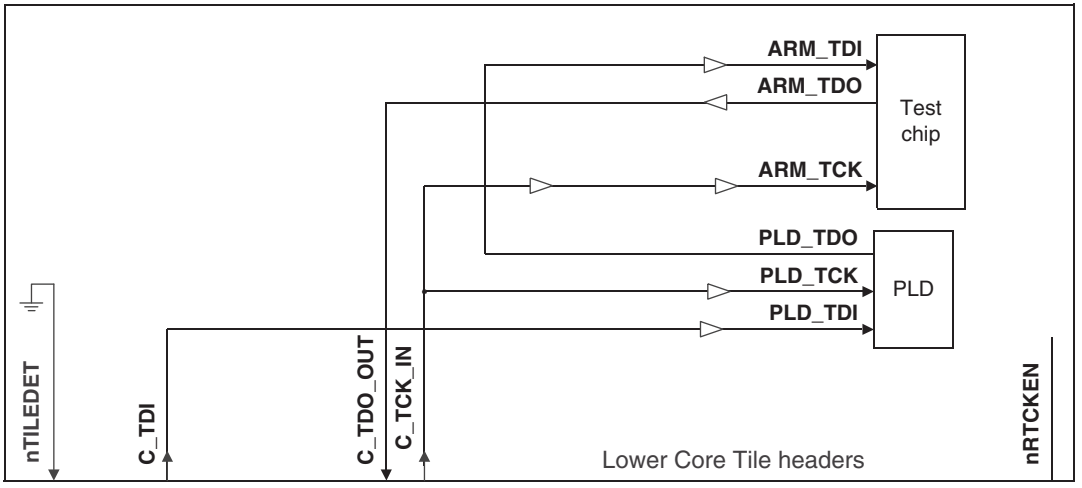

Figure 7-18 Simplified data and clocks for config mode (ARM7TDMI and ARM7TDMI-S)

JTAG reset and configure logic

Figure 7-19 on page 7-47 shows the reset and TMS signals.

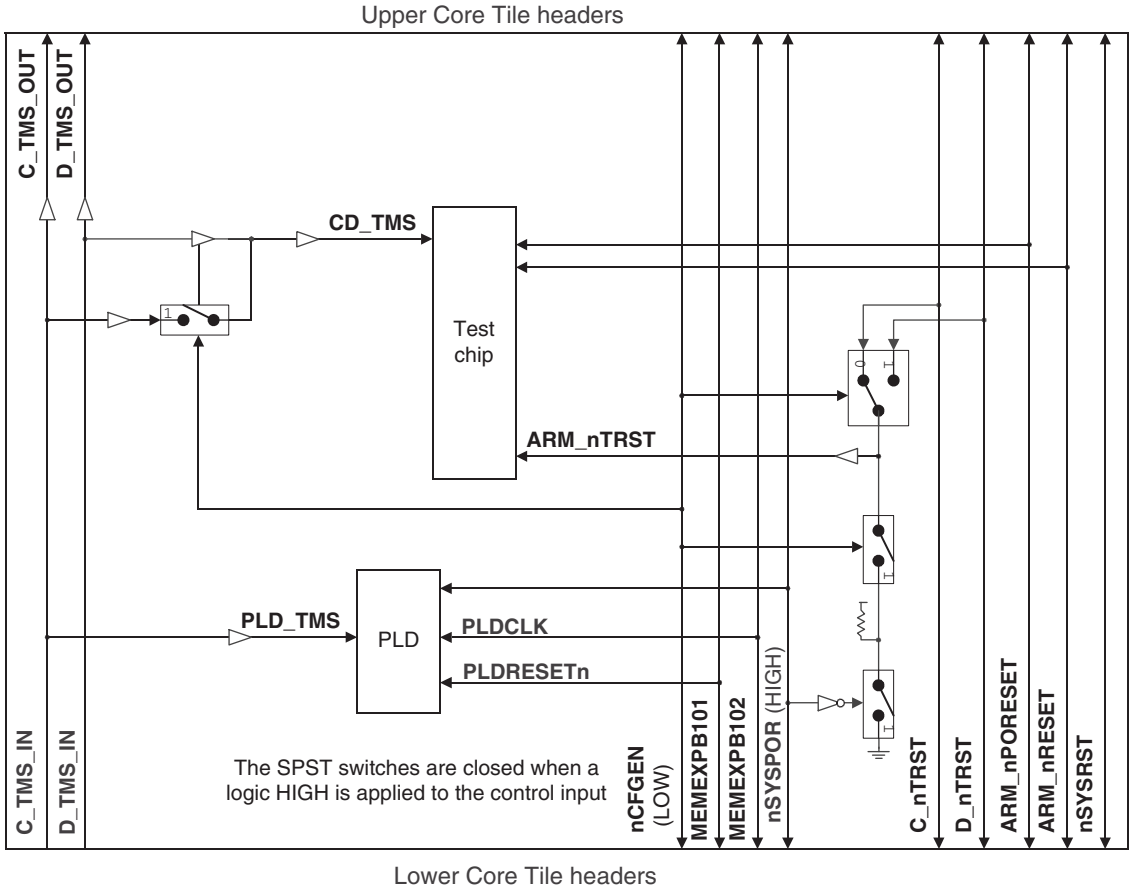

Figure 7-19 JTAG reset signals

7.8.2 Debug communications interrupts

The processor core incorporates EmbeddedICE logic that contains a communications channel used for passing information between the core and the JTAG equipment. The debug communications channel is implemented as coprocessor 14.

The processor accesses the debug communications channel registers using `MCR` and `MRC` instructions. The JTAG equipment reads and writes the register using the scan chain. For a description of the debug communications channel, see the Technical Reference Manual for your test chip.

The debug interrupt signals from the test chip are connected to HDRX pin 66 (**COMMTX**) and pin 68 (**COMMRX**).

Chapter 8
HBI-0141 Signal Descriptions

This index provides a summary of signals present on the CT7TDMI and CT7TDMI-S Core Tiles that use the HBI-0141 printed circuit board and describes the connectors and test points and the links that can be modified to change signal routing. It contains the following sections:

- *Header connectors* on page 8-2
- *Memory expansion connector pinout* on page 8-21
- *Links and test points* on page 8-22.

Note

For the Multi-ICE connector pinout and signal descriptions see *JTAG signals* on page 7-40.

The CT7TDMI and CT7TDMI-S Core Tiles do not have trace connectors.

8.1 Header connectors

Figure 8-1 shows the pin numbers and power-blade usage of the HDRX, HDRY, and HDRZ headers on the upper side of the tile. For details on the **VDDIO** and **VCCOY** voltages, see *Power supply control* on page 7-19.

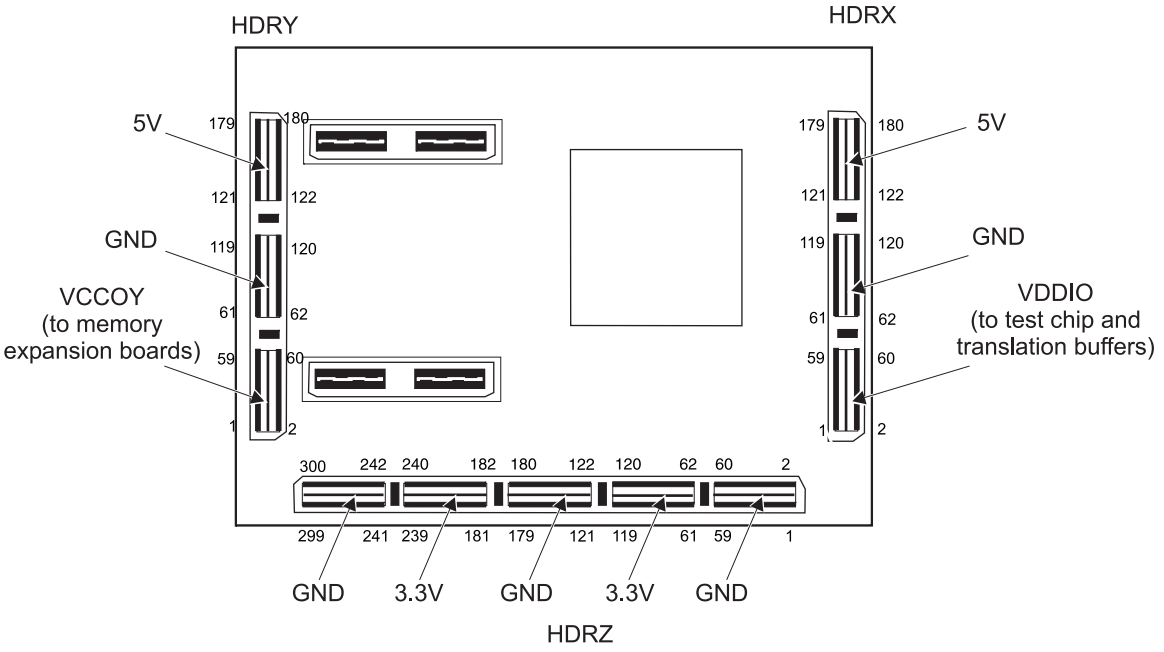

Figure 8-1 HDRX, HDRY, and HDRZ (upper) pin numbering

Table 6-1 on page 6-3 lists the Samtec part numbers.

Core Tiles use the -05- connectors on the top of the board and the -01 connectors on the bottom of the board. The total separation is 22mm. The Core Tiles have a maximum component height of 2.5mm on the bottom and 19mm on the top of the board. This ensures that there are no component interference problems with mated boards.

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.

8.1.1 HDRX signals

Table 8-1 describes the signals on the HDRX pins for the Core Tile and Logic Tile. (Replace *x* by L for the lower header and by U for the upper header for Logic Tile signals.)

Note

Except for the **SEL[1:0]**, **X_ECLK**, **X_MCLK** and **DATA[31:0]** signals, all of the signals on the upper and lower HDRX connectors are the same for the Core Tile.

Most of the pins on the RealView Logic Tiles, however, have different signals on the upper and lower Logic Tile HDRX pins. If the upper signal on the Logic Tile is named XU*n* and the lower signal is named XL*n*, the table entry is X*x*n. Replace *x* by L for the signal on the lower header and by U for the signal on the upper header.

For example, the pin 1 signal for the Logic Tile lower connector is **XL90** and the pin 1 signal for the upper connector is **XU90**. The corresponding signal for the Core Tile is **RANGEOUT1** (both upper lower pins).

If the upper signal on a tile has a completely different name than the lower signal, both names are listed and the signal on the upper connector is indicated by a (U) and the signal on the lower connector is indicated by a (L).

Table 8-1 HDRX signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
RANGEOUT1	**Xx90**	1	2	**Xx89**	**RANGEOUT0**
SCREG0	**Xx91**	3	4	**Xx88**	**NM4**
SCREG1	**Xx92**	5	6	**Xx87**	**NM3**
SCREG2	**Xx93**	7	8	**Xx86**	**NM2**
SCREG3	**Xx94**	9	10	**Xx85**	**NM1**
NTDOEN	**Xx95**	11	12	**Xx84**	**NM0**
ADDR0	**Xx96**	13	14	**Xx83**	**NEXEC**
ADDR1	**Xx97**	15	16	**Xx82**	**NENOUTI**
ADDR2	**Xx98**	17	18	**Xx81**	**NENOUT**

Table 8-1 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
ADDR3	**Xx99**	19	20	**Xx80**	**NENIN**
ADDR4	**Xx100**	21	22	**Xx79**	**IR3**
ADDR5	**Xx101**	23	24	**Xx78**	**IR2**
ADDR6	**Xx102**	25	26	**Xx77**	**IR1**
ADDR7	**Xx103**	27	28	**Xx76**	**IR0**
ADDR8	**Xx104**	29	30	**Xx75**	**DBGRQ**
ADDR9	**Xx105**	31	32	**Xx74**	**DBGRQI**
ADDR10	**Xx106**	33	34	**Xx73**	**HIGHZ**
ADDR11	**Xx107**	35	36	**Xx72**	**EXTERN1**
ADDR12	**Xx108**	37	38	**Xx71**	**EXTERN0**
ADDR13	**Xx109**	39	40	**Xx70**	**BL3**
ADDR14	**Xx110**	41	42	**Xx69**	**BL2**
ADDR15	**Xx111**	43	44	**Xx68**	**BL1**
ADDR16	**Xx112**	45	46	**Xx67**	**BL0**
ADDR17	**Xx113**	47	48	**Xx66**	**BREAKPT**
ADDR18	**Xx114**	49	50	**Xx65**	**APE**
ADDR19	**Xx115**	51	52	**Xx64**	**ALE**
ADDR20	**Xx116**	53	54	**Xx63**	**NRESET**
ADDR21	**Xx117**	55	56	**Xx62**	**NIRQ**
ADDR22	**Xx118**	57	58	**Xx61**	**NFIQ**
ADDR23	**Xx119**	59	60	**Xx60**	**DBGEN**
ADDR24	**Xx120**	61	62	**Xx59**	**DBGACK**
ADDR25	**Xx121**	63	64	**Xx58**	**ISYNC**
ADDR26	**Xx122**	65	66	**Xx57**	**COMMTX**

Table 8-1 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
ADDR27	**Xx123**	67	68	**Xx56**	**COMMRX**
ADDR28	**Xx124**	69	70	**Xx55**	**DBE**
ADDR29	**Xx125**	71	72	**Xx54**	**BIGENDIN**
ADDR30	**Xx126**	73	74	**Xx53**	**TBIT**
ADDR31	**Xx127**	75	76	**Xx52**	**ABORT**
CPA	**Xx128**	77	78	**Xx51**	**WAIT2**
CPB	**Xx129**	79	80	**Xx50**	**NWAIT**
NCPI	**Xx130**	81	82	**Xx49**	**LOCK**
TAPSM0	**Xx131**	83	84	**Xx48**	**WAITSEL**
TAPSM1	**Xx132**	85	86	**Xx47**	**NRW**
TAPSM2	**Xx133**	87	88	**Xx46**	**F1**
TAPSM3	**Xx134**	89	90	**Xx45**	**F0**
TCK1	**Xx135**	91	92	**Xx44**	**BC1**
TCK2	**Xx136**	93	94	**Xx43**	**BC0**
TEST0	**Xx137**	95	96	**Xx42**	**BCE**
TEST1	**Xx138**	97	98	**Xx41**	**NTRANS**
TEST2	**Xx139**	99	100	**Xx40**	**NOPC**
TEST3	**Xx140**	101	102	**Xx39**	**MAS1**
NC0	**Xx141**	103	104	**Xx38**	**MAS0**
NC1	**Xx142**	105	106	**Xx37**	**SEQ**
NC2	**Xx143**	107	108	**Xx36**	**NMREQ**
X144	**Xx144**	109	110 (U) 110 (L)	**XU35XL35**	**SEL1XL35**
X145	**Xx145**	111	112 (U) 112 (L)	**XU34XL34**	**SEL0XL34**

Table 8-1 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
X146	**Xx146**	113	114 (U) 114 (L)	**XU33XL33**	**X_ECLK_UPXL33**
X147	**Xx147**	115	116 (U) 116 (L)	**XU32XL32**	**X_MCLK_DNXL32**
X148	**Xx148**	117	118 (U) 118 (L)	**XU31XL31**	**DATA31XL31**
X149	**Xx149**	119	120 (U) 120(L)	**XU30XL30**	**DATA30XL30**
X150	**Xx150**	121	122 (U) 122 (L)	**XU29XL29**	**DATA29XL29**
X151	**Xx151**	123	124 (U) 124 (L)	**XU28XL28**	**DATA28XL28**
X152	**Xx152**	125	126 (U) 126 (L)	**XU27XL27**	**DATA27XL27**
X153	**Xx153**	127	128 (U) 128 (L)	**XU26XL26**	**DATA26XL26**
X154	**Xx154**	129	130 (U) 130 (L)	**XU25XL25**	**DATA25XL25**
X155	**Xx155**	131	132 (U) 132 (L)	**XU24XL24**	**DATA24XL24**
X156	**Xx156**	133	134 (U) 134 (L)	**XU23XL23**	**DATA23XL23**
X157	**Xx157**	135	136 (U) 136 (L)	**XU22XL22**	**DATA22XL22**
X158	**Xx158**	137	138 (U) 138 (L)	**XU21XL21**	**DATA21XL21**
X159	**Xx159**	139	140 (U) 140(L)	**XU20XL20**	**DATA20XL20**
X160	**Xx160**	141	142 (U) 142 (L)	**XU19XL19**	**DATA19XL19**

Table 8-1 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
X161	**Xx161**	143	144 (U) 144 (L)	**XU18XL18**	**DATA18XL18**
X162	**Xx162**	145	146 (U) 146 (L)	**XU17XL17**	**DATA17XL17**
X163	**Xx163**	147	148 (U) 148 (L)	**XU16XL16**	**DATA16XL16**
X164	**Xx164**	149	150 (U) 150 (L)	**XU15XL15**	**DATA15XL15**
X165	**Xx165**	151	152 (U) 152 (L)	**XU14XL14**	**DATA14XL14**
X166	**Xx166**	153	154 (U) 154 (L)	**XU13XL13**	**DATA13XL13**
X167	**Xx167**	155	156 (U) 156 (L)	**XU12XL12**	**DATA12XL12**
X168	**Xx168**	157	158 (U) 158 (L)	**XU11XL11**	**DATA11XL11**
X169	**Xx169**	159	160 (U) 160 (L)	**XU10XL10**	**DATA10XL10**
X170	**Xx170**	161	162 (U) 162 (L)	**XU9XL9**	**DATA9XL9**
X171	**Xx171**	163	164 (U) 164 (L)	**XU8XL8**	**DATA8XL8**
X172	**Xx172**	165	166 (U) 166 (L)	**XU7XL7**	**DATA7XL7**
X173	**Xx173**	167	168 (U) 168 (L)	**XU6XL6**	**DATA6XL6**
X174	**Xx174**	169	170 (U) 170 (L)	**XU5XL5**	**DATA5XL5**
X175	**Xx175**	171	172 (U) 172 (L)	**XU4XL4**	**DATA4XL4**

Table 8-1 HDRX signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
X176	**Xx176**	173	174 (U) 174 (L)	**XU3XL3**	**DATA3XL3**
X177	**Xx177**	175	178 (U) 178 (L)	**XU2XL2**	**DATA2XL2**
X178	**Xx178**	177	178 (U) 178 (L)	**XU1XL1**	**DATA1XL1**
X179	**Xx179**	179	180 (U) 180 (L)	**XU0XL0**	**DATA0XL0**

8.1.2 HDRY signals

Table 8-2 on page 8-9 describes the signals on the HDRY pins for the Core Tile and Logic Tile.

Note

Core Tile signals (**MEMEXPA[80:0]**, **Y[144:79]** and **Y[39:36]** are the same on the upper and lower HDRY connectors.

For the remaining Core Tile signals (and most Logic Tile signals), pins on the upper and lower connector have different signals. If the upper signal is named `YU`*n* and the lower signal is named `YL`*n*, the table entry is `Y`*xn*. Replace *x* by `L` for the signal on the lower header and by `U` for the signal on the upper header.

For example, the pin 1 signal for the lower connector on the Logic Tile is **YL89** and the pin 1 signal for the Logic Tile upper connector is **YU89**. The corresponding Core Tile signal for pin 1 (upper and lower) is **MEMEXPA25**.

Table 8-2 HDRY signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA25	**Yx89**	1	2	**Yx90**	**MEMEXPA26**
MEMEXPA24	**Yx88**	3	4	**Yx91**	**MEMEXPA27**
MEMEXPA23	**Yx87**	5	6	**Yx92**	**MEMEXPA28**
MEMEXPA22	**Yx86**	7	8	**Yx93**	**MEMEXPA29**
MEMEXPA21	**Yx85**	9	10	**Yx94**	**MEMEXPA30**
MEMEXPA20	**Yx84**	11	12	**Yx95**	**MEMEXPA31**
MEMEXPA19	**Yx83**	13	14	**Yx96**	**MEMEXPA32**
MEMEXPA18	**Yx82**	15	16	**Yx97**	**MEMEXPA33**
MEMEXPA17	**Yx81**	17	18	**Yx98**	**MEMEXPA34**
MEMEXPA16	**Yx80**	19	20	**Yx99**	**MEMEXPA35**
MEMEXPA15	**Yx79**	21	22	**Yx100**	**MEMEXPA36**
MEMEXPA14	**Yx78**	23	24	**Yx101**	**MEMEXPA37**
MEMEXPA13	**Yx77**	25	26	**Yx102**	**MEMEXPA38**
MEMEXPA12	**Yx76**	27	28	**Yx103**	**MEMEXPA39**
MEMEXPA11	**Yx75**	29	30	**Yx104**	**MEMEXPA40**
MEMEXPA10	**Yx74**	31	32	**Yx105**	**MEMEXPA41**
MEMEXPA9	**Yx73**	33	34	**Yx106**	**MEMEXPA42**
MEMEXPA8	**Yx72**	35	36	**Yx107**	**MEMEXPA43**
MEMEXPA7	**Yx71**	37	38	**Yx108**	**MEMEXPA44**
MEMEXPA6	**Yx70**	39	40	**Yx109**	**MEMEXPA45**
MEMEXPA5	**Yx69**	41	42	**Yx110**	**MEMEXPA46**
MEMEXPA4	**Yx68**	43	44	**Yx111**	**MEMEXPA47**
MEMEXPA3	**Yx67**	45	46	**Yx112**	**MEMEXPA48**

Table 8-2 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA2	**Yx66**	47	48	**Yx113**	**MEMEXPA49**
MEMEXPA1	**Yx65**	49	50	**Yx114**	**MEMEXPA50**
MEMEXPA0	**Yx64**	51	52	**Yx115**	**MEMEXPA51**
MEMEXPA103	**Yx63**	53	54	**Yx116**	**MEMEXPA52**
MEMEXPA102	**Yx62**	55	56	**Yx117**	**MEMEXPA53**
MEMEXPA101	**Yx61**	57	58	**Yx118**	**MEMEXPA54**
MEMEXPA100	**Yx60**	59	60	**Yx119**	**MEMEXPA55**
MEMEXPA99	**Yx59**	61	62	**Yx120**	**MEMEXPA56**
MEMEXPA98	**Yx58**	63	64	**Yx121**	**MEMEXPA57**
MEMEXPA97	**Yx57**	65	66	**Yx122**	**MEMEXPA58**
MEMEXPA96	**Yx56**	67	68	**Yx123**	**MEMEXPA59**
MEMEXPA95	**Yx55**	69	70	**Yx124**	**MEMEXPA60**
MEMEXPA94	**Yx54**	71	72	**Yx125**	**MEMEXPA61**
MEMEXPA93	**Yx53**	73	74	**Yx126**	**MEMEXPA62**
MEMEXPA92	**Yx52**	75	76	**Yx127**	**MEMEXPA63**
MEMEXPA91	**Yx51**	77	78	**Yx128**	**MEMEXPA64**
MEMEXPA90	**Yx50**	79	80	**Yx129**	**MEMEXPA65**
MEMEXPA89	**Yx49**	81	82	**Yx130**	**MEMEXPA66**
MEMEXPA88	**Yx48**	83	84	**Yx131**	**MEMEXPA67**
MEMEXPA87	**Yx47**	85	86	**Yx132**	**MEMEXPA68**
MEMEXPA86	**Yx46**	87	88	**Yx133**	**MEMEXPA69**
MEMEXPA85	**Yx45**	89	90	**Yx134**	**MEMEXPA70**
MEMEXPA84	**Yx44**	91	92	**Yx135**	**MEMEXPA71**
MEMEXPA83	**Yx43**	93	94	**Yx136**	**MEMEXPA72**

Table 8-2 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPA82	**Yx42**	95	96	**Yx137**	**MEMEXPA73**
MEMEXPA81	**Yx41**	97	98	**Yx138**	**MEMEXPA74**
MEMEXPA80	**Yx40**	99	100	**Yx139**	**MEMEXPA75**
Y39	**Yx39**	101	102	**Yx140**	**MEMEXPA76**
Y38	**Yx38**	103	104	**Yx141**	**MEMEXPA77**
Y37	**Yx37**	105	106	**Yx142**	**MEMEXPA78**
Y36	**Yx36**	107	108	**Yx143**	**MEMEXPA79**
Yx35	**Yx35**	109	110	**Yx144**	**Y144**
Yx34	**Yx34**	111	112	**Yx145**	**Y145**
Yx33	**Yx33**	113	114	**Yx146**	**Y146**
Yx32	**Yx32**	115	116	**Yx147**	**Y147**
Yx31	**Yx31**	117	118	**Yx148**	**Y148**
Yx30	**Yx30**	119	120	**Yx149**	**Y149**
Yx29	**Yx29**	121	122	**Yx150**	**Y150**
Yx28	**Yx28**	123	124	**Yx151**	**Y151**
Yx27	**Yx27**	125	126	**Yx152**	**Y152**
Yx26	**Yx26**	127	128	**Yx153**	**Y153**
Yx25	**Yx25**	129	130	**Yx154**	**Y154**
Yx24	**Yx24**	131	132	**Yx155**	**Y155**
Yx23	**Yx23**	133	134	**Yx156**	**Y156**
Yx22	**Yx22**	135	136	**Yx157**	**Y157**
Yx21	**Yx21**	137	138	**Yx158**	**Y158**
Yx20	**Yx20**	139	140	**Yx159**	**Y159**
Yx19	**Yx19**	141	142	**Yx160**	**Y160**

Table 8-2 HDRY signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Yx18	**Yx18**	143	144	**Yx161**	**Y161**
Yx17	**Yx17**	145	146	**Yx162**	**Y162**
Yx16	**Yx16**	147	148	**Yx163**	**Y163**
Yx15	**Yx15**	149	150	**Yx164**	**Y164**
Yx14	**Yx14**	151	152	**Yx165**	**Y165**
Yx13	**Yx13**	153	154	**Yx166**	**Y166**
Yx12	**Yx12**	155	156	**Yx167**	**Y167**
Yx11	**Yx11**	157	158	**Yx168**	**Y168**
Yx10	**Yx10**	159	160	**Yx169**	**Y169**
Yx9	**Yx9**	161	162	**Yx170**	**Y160**
Yx8	**Yx8**	163	164	**Yx171**	**Y171**
Yx7	**Yx7**	165	166	**Yx172**	**Y172**
Yx6	**Yx6**	167	168	**Yx173**	**Y173**
Yx5	**Yx5**	169	170	**Yx174**	**Y174**
Yx4	**Yx4**	171	172	**Yx175**	**Y175**
Yx3	**Yx3**	173	174	**Yx176**	**Y176**
Yx2	**Yx2**	175	176	**Yx177**	**Y177**
Yx1	**Yx1**	177	178	**Yx178**	**Y178**
Yx0	**Yx0**	179	180	**Yx179**	**Y179**

8.1.3 HDRZ

Table 8-1 on page 8-3 describes the signals on the HDRZ pins for the Core Tile and the RealView Logic Tile.

Note

Except for the clock, JTAG, and **MEMEXPB[21:0]** signals, the Core Tile has the same signal on both the upper and lower pins. Some of the upper and lower HDRZ signals can also be isolated from each other, see *Isolation and foldover of header signals* on page 7-26.

Most of the pins on the RealView Logic Tiles have different signals on the upper and lower Logic Tile HDRZ pins. If the upper signal on the Logic Tile is named `ZU`*n* and the lower signal is named `ZL`*n*, the table entry is `Z`*xn*. Replace *x* by `L` for the signal on the lower header and by `U` for the signal on the upper header.

For example, the pin 1 signal for the Logic Tile lower connector is **ZL128** and the pin 1 signal for the upper connector is **ZU128**. The corresponding signals for the Core Tile are **MEMEXPB0** (upper) and **ZL128** (lower).

If the upper signal on a tile has a completely different name than the lower signal, both names are listed and the signal on the upper connector is indicated by a (U) and the signal on the lower connector is indicated by a (L).

Table 8-3 HDRZ signals

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB0ZL128	**Zx128**	1 (U) 1 (L)	2	**Zx255**	**Z255**
MEMEXPB1ZL129	**Zx129**	3 (U) 3 (L)	4	**Zx254**	**Z254**
MEMEXPB2ZL130	**Zx130**	5 (U) 5 (L)	6	**Zx253**	**Z253**
MEMEXPB3ZL131	**Zx131**	7 (U) 7 (L)	8	**Zx252**	**Z252**
MEMEXPB4ZL132	**Zx132**	9 (U) 9 (L)	10	**Zx251**	**Z251**
MEMEXPB5ZL133	**Zx133**	11 (U) 11 (L)	12	**Zx250**	**Z250**
MEMEXPB6ZL134	**Zx134**	13 (U) 13 (L)	14	**Zx249**	**Z249**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB7ZL135	**Zx135**	15 (U) 15 (L)	16	**Zx248**	**Z248**
MEMEXPB8ZL136	**Zx136**	17 (U) 17 (L)	18	**Zx247**	**Z247**
MEMEXPB9ZL137	**Zx137**	19 (U) 19 (L)	20	**Zx246**	**Z246**
MEMEXPB10ZL138	**Zx138**	21 (U) 21 (L)	22	**Zx245**	**Z245**
MEMEXPB11ZL139	**Zx139**	23 (U) 23 (L)	24	**Zx244**	**Z244**
MEMEXPB12ZL140	**Zx140**	25 (U) 25 (L)	26	**Zx243**	**Z243**
MEMEXPB13ZL141	**Zx141**	27 (U) 27 (L)	28	**Zx242**	**Z242**
MEMEXPB14ZL142	**Zx142**	29 (U) 29 (L)	30	**Zx241**	**Z241**
MEMEXPB15ZL143	**Zx143**	31 (U) 31 (L)	32	**Zx240**	**Z240**
MEMEXPB16ZL144	**Zx144**	33 (U) 33 (L)	34	**Zx249**	**Z239**
MEMEXPB17ZL145	**Zx145**	35 (U) 35 (L)	36	**Zx248**	**Z238**
MEMEXPB18ZL146	**Zx146**	37 (U) 37 (L)	38	**Zx237**	**Z237**
MEMEXPB19ZL147	**Zx147**	39 (U) 39 (L)	40	**Zx236**	**Z236**
MEMEXPB20	**Zx148**	41 (U) 41 (L)	42	**Zx235**	**Z235**
MEMEXPB21ZL149	**Zx149**	43 (U) 43 (L)	44	**Zx234**	**Z234**
MEMEXPB22	**Zx150**	45	46	**Zx233**	**Z233**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB23	**Zx151**	47	48	**Zx232**	**Z232**
MEMEXPB24	**Zx152**	49	50	**Zx231**	**MEMEXPB103**
MEMEXPB25	**Zx153**	51	52	**Zx230**	**PLDCLK**
MEMEXPB26	**Zx154**	53	54	**Zx229**	**PLDRESETN**
MEMEXPB27	**Zx155**	55	56	**Zx228**	**PLDD0**
MEMEXPB28	**Zx156**	57	58	**Zx227**	**PLDD1**
MEMEXPB29	**Zx157**	59	60	**Zx226**	**MEMEXPB98**
MEMEXPB30	**Zx158**	61	62	**Zx225**	**MEMEXPB97**
MEMEXPB31	**Zx159**	63	64	**Zx224**	**MEMEXPB96**
MEMEXPB32	**Zx160**	65	66	**Zx223**	**MEMEXPB95**
MEMEXPB33	**Zx161**	67	68	**Zx222**	**MEMEXPB94**
MEMEXPB34	**Zx162**	69	70	**Zx221**	**MEMEXPB93**
MEMEXPB35	**Zx163**	71	72	**Zx220**	**MEMEXPB92**
MEMEXPB36	**Zx164**	73	74	**Zx219**	**MEMEXPB91**
MEMEXPB37	**Zx165**	75	76	**Zx218**	**MEMEXPB90**
MEMEXPB38	**Zx166**	77	78	**Zx217**	**MEMEXPB89**
MEMEXPB39	**Zx167**	79	80	**Zx216**	**MEMEXPB88**
MEMEXPB40	**Zx168**	81	82	**Zx215**	**MEMEXPB87**
MEMEXPB41	**Zx169**	83	84	**Zx214**	**MEMEXPB86**
MEMEXPB42	**Zx170**	85	86	**Zx213**	**MEMEXPB85**
MEMEXPB43	**Zx171**	87	88	**Zx212**	**MEMEXPB84**
MEMEXPB44	**Zx172**	89	90	**Zx211**	**MEMEXPB83**
MEMEXPB45	**Zx173**	91	92	**Zx210**	**MEMEXPB82**
MEMEXPB46	**Zx174**	93	94	**Zx209**	**MEMEXPB81**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
MEMEXPB47	**Zx175**	95	96	**Zx208**	**MEMEXPB80**
MEMEXPB48	**Zx176**	97	98	**Zx207**	**MEMEXPB79**
MEMEXPB49	**Zx177**	99	100	**Zx206**	**MEMEXPB78**
MEMEXPB50	**Zx178**	101	102	**Zx205**	**MEMEXPB77**
MEMEXPB51	**Zx179**	103	104	**Zx204**	**MEMEXPB76**
MEMEXPB52	**Zx180**	105	106	**Zx203**	**MEMEXPB75**
MEMEXPB53	**Zx181**	107	108	**Zx202**	**MEMEXPB74**
MEMEXPB54	**Zx182**	109	110	**Zx201**	**MEMEXPB73**
MEMEXPB55	**Zx183**	111	112	**Zx200**	**MEMEXPB72**
MEMEXPB56	**Zx184**	113	114	**Zx199**	**MEMEXPB71**
MEMEXPB57	**Zx185**	115	116	**Zx198**	**MEMEXPB70**
MEMEXPB58	**Zx186**	117	118	**Zx197**	**MEMEXPB69**
MEMEXPB59	**Zx187**	119	120	**Zx196**	**MEMEXPB68**
MEMEXPB60	**Zx188**	121	122	**Zx195**	**MEMEXPB67**
MEMEXPB61	**Zx189**	123	124	**Zx194**	**MEMEXPB66**
MEMEXPB62	**Zx190**	125	126	**Zx193**	**MEMEXPB65**
MEMEXPB63	**Zx191**	127	128	**Zx192**	**MEMEXPB64**
D_nSRST	**D_nSRST**	129	130 (U) 130 (L)	**CLK_POS_DN_INC** **LK_POS_DN_OUT**	**CLK_POS_DN**
D_nTRST	**D_nTRST**	131	132 (U) 132 (L)	**CLK_NEG_DN_INC** **LK_NEG_DN_OUT**	**CLK_NEG_DN_INC** **LK_NEG_DN_OUT**
D_TDO_IN **D_TDO_OUT**	**D_TDO_IN** **D_TDO_OUT**	133 (U) 133 (L)	134 (U) 134 (L)	**CLK_POS_UP_OUT** **CLK_POS_UP_IN**	**CLK_POS_UP**
D_TDI	**D_TDI**	135	136 (U) 136 (L)	**CLK_NEG_UP_OUT** **CLK_NEG_UP_IN**	**CLK_NEG_UP_OUT** **CLK_NEG_UP_IN**
D_TCK_OUT **D_TCK_IN**	**D_TCK_OUT** **D_TCK_IN**	137 (U) 137 (L)	138 (U) 138 (L)	**CLK_UP_THRU** **CLK_IN_MINUS2**	**CLK_IN_MINUS2**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
D_TMS_OUTD_ TMS_IN	**D_TMS_OUTD_ TMS_IN**	139 (U) 139 (L)	140 (U) 140 (L)	**CLK_OUT_PLUS1C LK_IN_MINUS1**	**CLK_IN_MINUS1**
D_RTCK	**D_RTCK**	141	142 (U) 142 (L)	**CLK_OUT_PLUS2C LK_UP_THRU**	**CLK_UP_THRU**
C_nSRST	**C_nSRST**	143	144 (U) 144 (L)	**CLK_IN_PLUS2 CLK_DN_THRU**	**CLK_IN_PLUS2**
C_nTRST	**C_nTRST**	145	146 (U) 146 (L)	**CLK_IN_PLUS1CLK _OUT_MINUS1**	**CLK_IN_PLUS1**
PLD_TDIC_TDO _OUT	**C_TDO_INC_TD O_OUT**	147 (U) 147 (L)	148 (U) 148 (L)	**CLK_DN_THRUCL K_OUT_MINUS2**	**CLK_DN_THRU**
C_TDI	**C_TDI**	149	150	**CLK_GLOBAL**	**CLK_GLOBAL**
C_TCK_OUTC_ TCK_IN	**C_TCK_OUT**	151 (U) 151 (L)	152	**FPGA_IMAGE**	**FPGA_IMAGE**
C_TMS_OUTC_ TMS_IN	**C_TMS_OUT**	153 (U) 153 (L)	154	**nSYSPOR**	**nSYSPOR**
nTILE_DETGND	**nTILE_DETGND**	155 (U) 155 (L)	156	**nSYSRST**	**nSYSRST**
nCFGEN	**nCFGEN**	157	158	**nRTCKEN**	**nRTCKEN**
GLOBAL_DONE	**GLOBAL_DONE**	159	160	**SPARE12** (reserved)	**SPARE12**
SPARE11	**SPARE11** (reserved)	161	162	**SPARE10** (reserved)	**SPARE10**
SPARE9	**SPARE9** (reserved)	163	164	**SPARE8** (reserved)	**SPARE8**
SPARE7	**SPARE7** (reserved)	165	166	**SPARE6** (reserved)	**SPARE6**
SPARE5	**SPARE5** (reserved)	167	168	**SPARE4** (reserved)	**SPARE4**
SPARE3	**SPARE3** (reserved)	169	170	**SPARE2** (reserved)	**SPARE2**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
SPARE1	**SPARE1** (reserved)	171	172	**SPARE0** (reserved)	**SPARE0**
Zx63	**Z63**	173	174	**Z64**	**Zx64**
Zx62	**Z62**	175	176	**Z65**	**Zx65**
Zx61	**Z61**	177	178	**Z66**	**Zx66**
Zx60	**Z60**	179	180	**Z67**	**Zx67**
Zx59	**Z59**	181	182	**Z68**	**Zx68**
Zx58	**Z58**	183	184	**Z79**	**Zx69**
Zx57	**Z57**	185	186	**Z70**	**Zx70**
Zx56	**Z56**	187	188	**Z71**	**Zx71**
Zx55	**Z55**	189	190	**Z72**	**Zx72**
Zx54	**Z54**	191	192	**Z73**	**Zx73**
Zx53	**Z53**	193	194	**Z74**	**Zx74**
Zx52	**Z52**	195	196	**Z75**	**Zx75**
Zx51	**Z51**	197	198	**Z76**	**Zx76**
Zx50	**Z50**	199	200	**Z77**	**Zx77**
Zx49	**Z49**	201	202	**Z78**	**Zx78**
Zx48	**Z48**	203	204	**Z79**	**Zx79**
Zx47	**Z47**	205	206	**Z80**	**Zx80**
Zx46	**Z46**	207	208	**Z81**	**Zx81**
Zx45	**Z45**	209	210	**Z82**	**Zx82**
Zx44	**Z44**	211	212	**Z83**	**Zx83**
Zx43	**Z43**	213	214	**Z84**	**Zx84**
Zx42	**Z42**	215	216	**Z85**	**Zx85**
Zx41	**Z41**	217	218	**Z86**	**Zx86**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Zx40	**Z40**	219	220	**Z87**	**Zx87**
Zx39	**Z39**	221	222	**Z88**	**Zx88**
Zx38	**Z38**	223	224	**Z89**	**Zx89**
Zx37	**Z37**	225	226	**Z90**	**Zx90**
Zx36	**Z36**	227	228	**Z91**	**Zx91**
Zx35	**Z35**	229	230	**Z92**	**Zx92**
Zx34	**Z34**	231	232	**Z93**	**Zx93**
Zx33	**Z33**	233	234	**Z94**	**Zx94**
Zx32	**Z32**	235	236	**Z95**	**Zx95**
Zx31	**Z31**	237	238	**Z96**	**Zx96**
Zx30	**Z30**	239	240	**Z97**	**Zx97**
Zx29	**Z29**	241	242	**Z98**	**Zx98**
Zx28	**Z28**	243	244	**Z99**	**Zx99**
Zx27	**Z27**	245	246	**Z100**	**Zx100**
Zx26	**Z26**	247	248	**Z101**	**Zx101**
Zx25	**Z25**	249	250	**Z102**	**Zx102**
Zx24	**Z24**	251	252	**Z103**	**Zx103**
Zx23	**Z23**	253	254	**Z104**	**Zx104**
Zx22	**Z22**	255	256	**Z105**	**Zx105**
Zx21	**Z21**	257	258	**Z106**	**Zx106**
Zx20	**Z20**	259	260	**Z107**	**Zx107**
Zx19	**Z19**	261	262	**Z108**	**Zx108**
Zx18	**Z18**	263	264	**Z109**	**Zx109**
Zx17	**Z17**	265	266	**Z110**	**Zx110**

Table 8-3 HDRZ signals (continued)

Core Tile signals	Logic Tile signals	Odd pins	Even pins	Logic Tile signals	Core Tile signals
Zx16	**Z16**	267	268	**Z111**	**Zx111**
Zx15	**Z15**	269	270	**Z112**	**Zx112**
Zx14	**Z14**	271	272	**Z113**	**Zx113**
Zx13	**Z13**	273	274	**Z114**	**Zx114**
Zx12	**Z12**	275	276	**Z115**	**Zx115**
Zx11	**Z11**	277	278	**Z116**	**Zx116**
Zx10	**Z10**	279	280	**Z117**	**Zx117**
Zx9	**Z9**	281	282	**Z118**	**Zx118**
Zx8	**Z8**	283	284	**Z119**	**Zx119**
Zx7	**Z7**	285	286	**Z120**	**Zx120**
Zx6	**Z6**	287	288	**Z121**	**Zx121**
Zx5	**Z5**	289	290	**Z122**	**Zx122**
Zx4	**Z4**	291	292	**Z123**	**Zx123**
Zx3	**Z3**	293	294	**Z124**	**Zx124**
Zx2	**Z2**	295	296	**Z125**	**Zx125**
Zx1	**Z1**	297	298	**Z126**	**Zx126**
Zx0	**Z0**	299	300	**Z127**	**Zx127**

8.2 Memory expansion connector pinout

The Core Tile has two 120-way Samtec QSH memory expansion connectors.

Note

See *Memory expansion connector pinout* on page 6-20 for details of the expansion connectors on the Core Tile. Refer to Appendix A *Static Memory Expansion Board*, the documentation supplied with the memory boards, and the *PISMO Memory Interface Specification* for details of signal assignment on the expansion boards. Different memory boards might have different voltage levels or different arrangement of address or control signals.

8.3 Links and test points

This section describes the links and test points present on the Core Tile.

8.3.1 Links

Figure 8-2 shows the location of the configuration resistors on the top of the Core Tile. Figure 8-3 on page 8-23 shows the location of the links and configuration resistors on the bottom of the Core Tile.

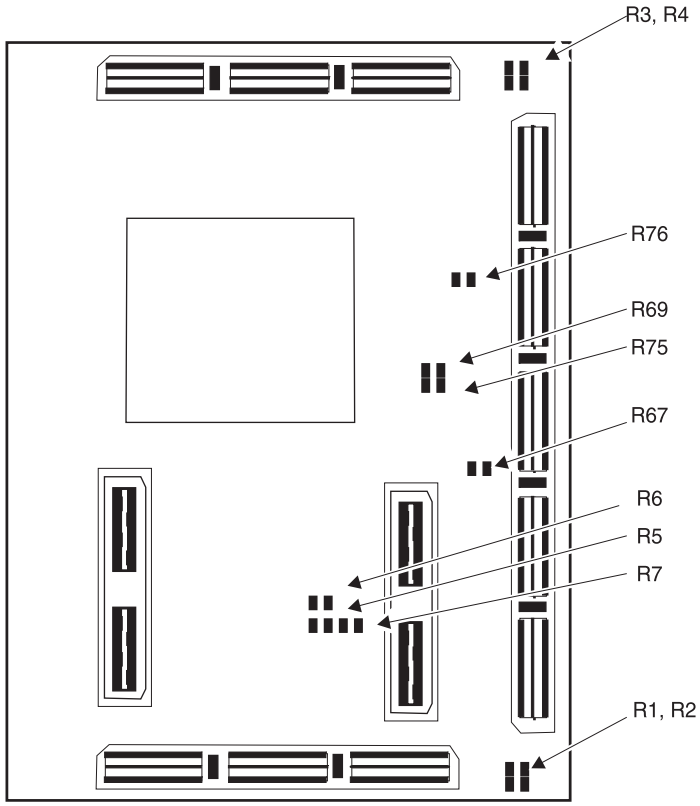

Figure 8-2 Location of links (top)

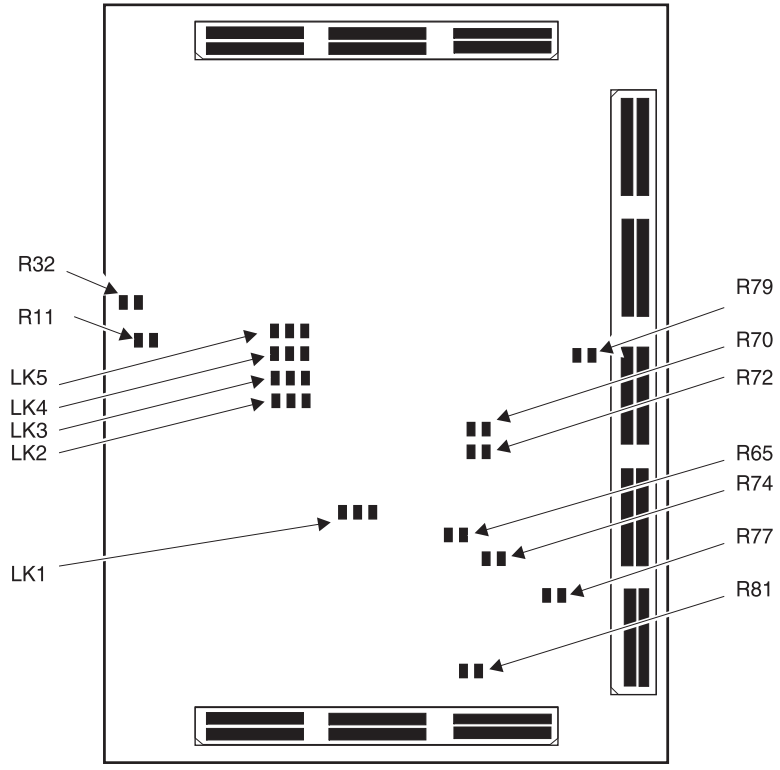

Figure 8-3 Location of links (bottom)

Board identifier links

LK5-LK2 determine the value of the **MANID[3:0]** signals that identifies the board build variant.

Caution

The board identifier links are set at manufacture and must not be changed.

JTAG links

The ARM7TDMI does not have a boundary scan chain. Resistor links bypass some of the JTAG signals depending on the test chip fitted as listed in Table 8-4. See the BOM file for the fitting options used at manufacture.

Table 8-4 Resistor links for JTAG logic

Link	Description
LK1	Selects either **ARM_RTCK** or **DBGRQI** as **ARM_DBGRQI** signal.
R67	Links **ARM_RTCK** to **RTCK_ARM**
R74	Links **D_TCK_BYPASS** to **RTCK_ARM**
R65	Links a buffered version of **D_TCK_IN** to **D_TCK_BYPASS**
R77	Links **PLD_TDO** to **CD_TDO_THRU**
R81	Links **D_TDO_IN** to **D_TDO_OUT**
R79	Links **nRTCKEN** to ground. See *JTAG signals* on page 7-40.

Clock selection links

Some of the clock signals (see also *Clocks* on page 3-10) have associated resistor links as listed in Table 8-5.

These links are set at manufacture for the specific Core Tile and do not normally require changing.

Table 8-5 Resistor links for clock signals

Link	Description
R70, R75	R70 links **GLB_MCLK** input of the **MCLK** multiplexor to a buffered version of **CLK_GLOBAL_IN**. If the buffer amplifier is not fitted, R75 bridges the amplifier input and output signals.
R69, R72	R69 links the **X_MCLK** input of the **MCLK** multiplexor to a buffered version of **X_MCLK**. If the buffer amplifier is not fitted, R72 bridges the amplifier input and output signals.
R83, R76	R83 links a buffered version of the **ECLK** output of the test chip to the **X_ECLK** multiplexor. If the buffer amplifier is not fitted, R75 bridges the amplifier input and output signals.

Voltage selection links

There are also resistors and resistor links that select the operating voltages for the tile, see *Power supply control* on page 7-19.

Caution

The resistor links listed in Table 8-6 are set at manufacture. Changing the resistors can result in voltages that are out of the normal operating range for the test chip.

Table 8-6 Resistor links for voltage selection

Link	Description
R11	Current sense resistor for **VDDCOREA**
R32	Current sense resistor for **VDDCOREB**
R1 and R2	Connects **3V3** to **VCCOY**
R3 and R4	Connects **3V3** to **ARM_VDDIO**

SDRAM resistor links

The resistor links listed in Table 8-7 enable support for the VREF signals used with DDR SDRAM memory modules. These signals must be driven by an attached board such as a Logic Tile.

Table 8-7 Resistor links for DDR SDRAM

Link	Description
R5	links MEXPA77 to YU3
R6	links MEXPA78 to YU17
R7	links MEXPA79 to YU25

8.3.2 Test points

Figure 8-4 on page 8-26 shows the location of test points on the Core Tile. Table 8-8 on page 8-26 lists the function of each test point.

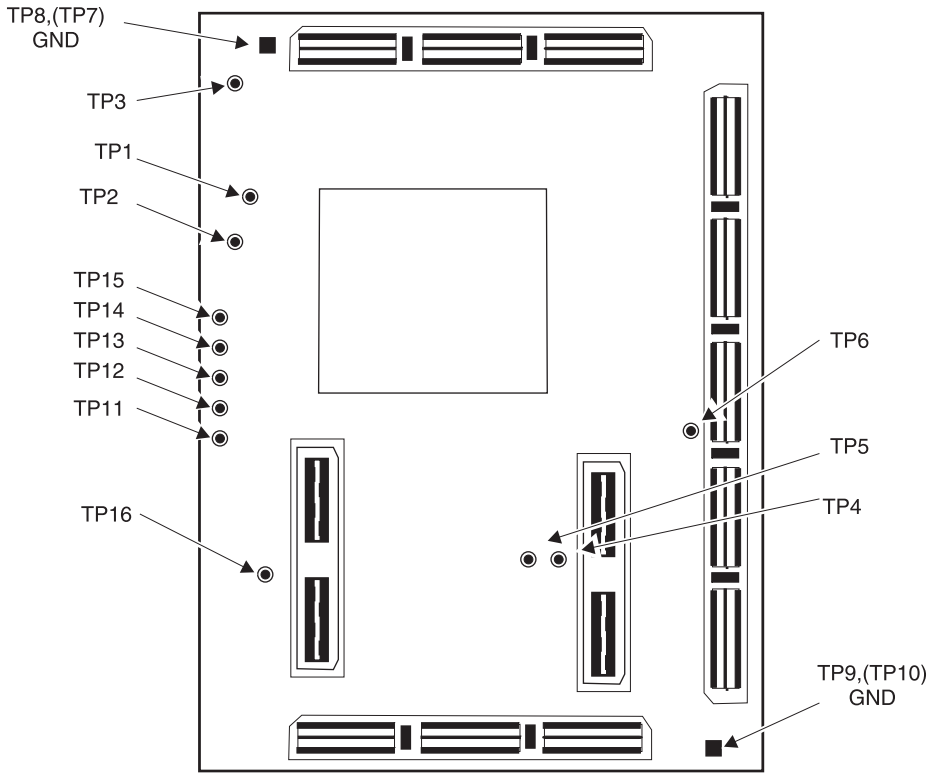

Figure 8-4 Test point location

Table 8-8 Test point function

Test point	Description
1	**VDDCOREA** voltage
2	**VDDCOREB** voltage
3	**VDDIO** voltage
4	**VDDSENSE1** voltage input to ADC
5	**VDDSENSE2** voltage input to ADC
6	**CLK_GLOBAL**

Table 8-8 Test point function (continued)

Test point	Description
7-10	Ground
15-11	Manufacturing test for PLD
16	**1V8** voltage

Appendix A
Static Memory Expansion Board

This appendix describes the static memory expansion board. It contains the following sections:

A.1 About memory expansion

You can fit one or two static expansion boards to the Core Tile.

There are five chip select signals available on the static expansion board. Each of these can select 64MB of SRAM or flash memory.

The block diagram for a typical memory board is shown in Figure A-1.

Note

Figure A-1 is an example only. Different expansion boards might have different features. For example, the links selecting which chip select to use might be omitted.

See the documentation provided with your memory board for details on signals and link options.

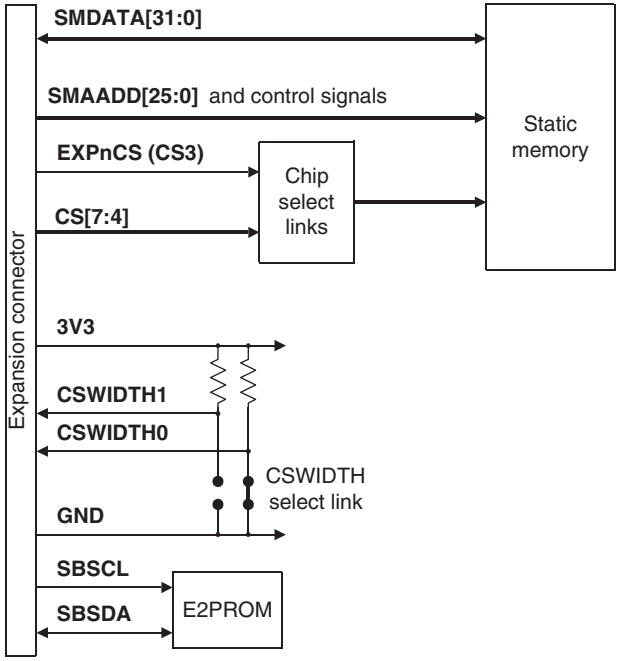

Figure A-1 Static memory board block diagram

A.1.1 Memory board configuration

The E2PROM on the memory board can be read from an attached Logic Tile to identify the type of memory on the board and how it is configured. This information can be used by the application or operating system to initialize the memory space.

Note

Additional configuration information is present in the E2PROM on the expansion board, see *EEPROM contents* on page A-5.

Memory width selection on the static memory board

The memory width on the memory board is encoded into the **CSWIDTH[1:0]** signals as shown in Table A-1. These signals are defined by links on the memory board and are fixed at manufacture and cannot be changed.

Table A-1 Memory width encoding

CSWIDTH[1:0]	Width
00	8 bit
01	16 bit
10	32 bit (default)
11	No memory present

A.2 Fitting a memory board

To install a memory expansion board:

1. Ensure that the Core Tile is powered down.
2. Align the memory board with the connectors on the interface board as shown in Figure A-2.
3. Press the memory board into the connector.

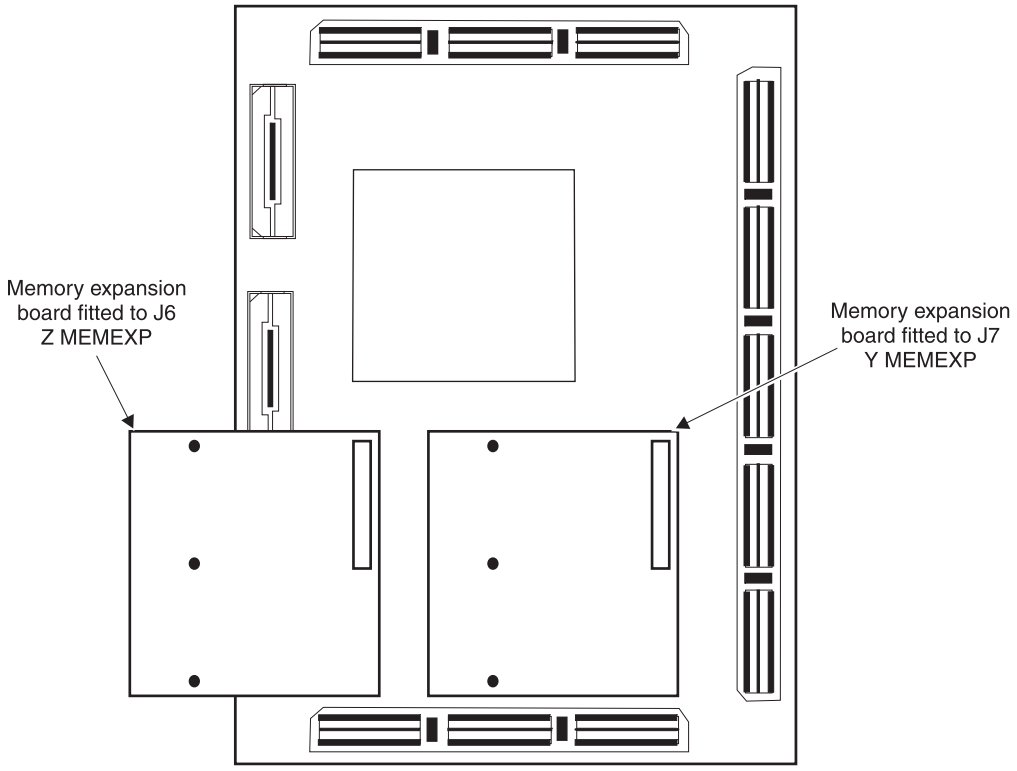

Figure A-2 Memory board installation

A.3 EEPROM contents

The Static Memory Expansion EEPROM is 256 bytes in size and contains 5 chip select information blocks, a manufacturer string and a checksum. The EEPROM is read and written using a serial bus implemented on HDRX signals **MEMEXPB88** (**SBSDA** data) and **MEMEXPB90** (**SBSCL** clock). The Logic Tile must implement a controller for these signals on Logic Tile signals **Z216** and **Z218**.

Each chip select information block contains details about the memory devices accessed with the corresponding chip select signal. The organization of a chip select information block is listed in table Table A-2 on page A-6.

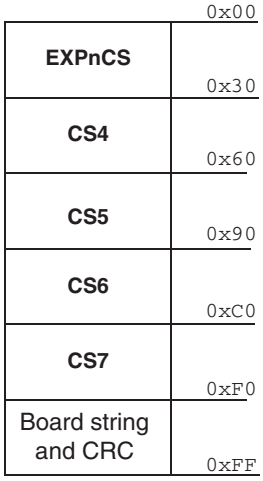

Figure A-3 Chip select information block

Table A-2 Chip Select information block

Function	Address offset	Value
Memory Type	0x0	0x0= Reserved 0x1= Static Disk On Chip 0x2= Static NOR flash 0x3= Static SRAM 0x4-0xFE = Reserved (used by dynamic memory expansion modules) 0xFF = Not fitted.
Memory Width	0x01	Bits [3:0] indicate the chip-select width: 0= byte wide1= 16-bit wide2= 32-bit wide3= Reserved. Bits [7:4] indicate the device memory width: 0= byte wide1= 16-bit wide2= 32-bit wide3= Reserved.
Access time	0x02	Two bytes containing the access time (tACC) decoded as a binary multiple of 100ps. Location 2 contains the LSB and location 3 contains the MSB. For example, a flash device with 120ns access is 1200 * 0.1ns. The decimal value is 1200 and the hex value is 0x04B0, therefore location 2 contains 0xb0 and location 3 contains 0x04.
Size	0x04	Four bytes containing the size of the memory in bytes. Location 4 is the LSB and location 7 is MSB.
Reserved	0x08-0x0F	Eight bytes reserved for future expansion
Device string	0x10-0x2F	Null terminated string of up to 32 characters (31 characters + null character) containing the manufacturer name and part number.

The base address of the information block is determined by the device chip select used.

The contents of a typical static memory expansion EEPROM with devices on **EXPnCS** and **CS4** is listed in Table A-3. Unused chip select blocks are filled with 0xFF.

Table A-3 Example contents of a static memory expansion EEPROM

Address offset	Contents	Contents
0x00	EXPnCS memory type	0x02 = Static NOR flash
0x01	EXPnCS memory width	0x12 - 32 bit chip select width, 16-bit device memory width
0x02	EXPnCS access time in 0.1ns (LSB)	0xb0 - LSB (of 1200 which 1200 * 0.1ns = 120ns access time)

Table A-3 Example contents of a static memory expansion EEPROM (continued)

Address offset	Contents	Contents
0x03	EXPnCS access time in 0.1ns (MSB)	0x04 - MSB (of 1200 which 1200 * 0.1ns = 120ns access time)
0x04	EXPnCS memory size in bytes (LSB)	0x00
0x05	EXPnCS memory size in bytes	0x00
0x06	EXPnCS memory size in bytes	0x00
0x07	EXPnCS memory size in bytes (MSB)	0x04 (0x04000000 Bytes = 64MBytes)
0x8-0xF	Reserved	0xFF
0x10-0x2F	EXPnCS memory device string	"Intel GE28F256K3C120" + null character
0x30	CS4 memory type	0x01 = Static SRAM
0x31	CS4 memory width	0x02 - 32 bit wide
0x32	CS4 access time in 0.1ps (LSB)	0x26 - LSB (of 550 which 550 * 0.1ns = 55ns access time)
0x33	CS4 access time in 0.1ps (MSB)	0x02 - MSB (of 550 which 550 * 0.1ns = 55ns access time)
0x34	CS4 memory size in bytes (LSB)	0x00
0x35	CS4 memory size in bytes	0x00
0x36	CS4 memory size in bytes	0x00
0x37	CS4 memory size in bytes (MSB)	0x20 (0x00200000 Bytes = 2MBytes)
0x38-0x3F	Reserved	0xFF
0x40-0x5F	CS4 memory device string	"Samsung K6F8016U6A-F55" + null character
0x60	CS5 memory type	0xFF - not fitted
0x61	CS5 memory width	0xFF
0x62	CS5 access time in 0.1ps (LSB)	0xFF
0x63	CS5 access time in 0.1ps (MSB)	0xFF
0x64	CS5 memory size in bytes (LSB)	0xFF
0x65	CS5 memory size in bytes	0xFF
0x66	CS5 memory size in bytes	0xFF

Table A-3 Example contents of a static memory expansion EEPROM (continued)

Address offset	Contents	Contents
0x67	CS5 memory size in bytes (MSB)	0xFF
0x68-0x6F	Reserved	0xFF
0x70-0x8F	CS5 memory device string	0xFF
0x90	CS6 memory type	0xFF - not fitted
0x91	CS6 memory width	0xFF
0x92	CS6 access time in 0.1ps (LSB)	0xFF
0x93	CS6 access time in 0.1ps (MSB)	0xFF
0x94	CS6 memory size in bytes (LSB)	0xFF
0x95	CS6 memory size in bytes	0xFF
0x96	CS6 memory size in bytes	0xFF
0x97	CS6 memory size in bytes (MSB)	0xFF
0x98-0x9F	Reserved	0xFF
0xA0-0xBF	CS6 memory device string	0xFF
0xC0	CS7 memory type	0xFF - not fitted
0xC1	CS7 memory width	0xFF
0xC2	CS7 access time in 0.1ps (LSB)	0xFF
0xC3	CS7 access time in 0.1ps (MSB)	0xFF
0xC4	CS7 memory size in bytes (LSB)	0xFF
0xC5	CS7 memory size in bytes	0xFF
0xC6	CS7 memory size in bytes	0xFF
0xC7	CS7 memory size in bytes (MSB)	0xFF
0xC8-0xCF	Reserved	0xFF

Table A-3 Example contents of a static memory expansion EEPROM (continued)

Address offset	Contents	Contents
0xD0-0xEF	CS7 memory device string	0xFF
0xF0-0xFE	Board manufacturer string	"ARM HBI0124A"+ null character
0xFF	Checksum Byte	The LSB of the sum of bytes 0x00-0xFE

A.4 Connector pinout

The static memory expansion board uses a 120-way Samtec QSH style connector. The Core Tile uses the corresponding 120-way Samtec QTH connector on the interface board, see *Memory expansion connector pinout* on page 6-20 for details of the Core Tile signals on the memory connector. The connector pinout for the static memory board is shown in Table A-4.

Note

The numbering of pins on the connector is for the connectors as viewed from below.

Table A-4 Static memory connector signals

Pin No.	Signal	Pin No.	Signal
1	**DATA[0]**	2	**3V3**
3	**DATA[1]**	4	**3V3**
5	**DATA[2]**	6	**3V3**
7	**DATA[3]**	8	**3V3**
9	**DATA[4]**	10	**VDDIO**[a]
11	**DATA[5]**	12	**VDDIO**[a]
13	**DATA[6]**	14	**VDDIO**[a]
15	**DATA[7]**	16	**VDDIO**[a]
17	**DATA[8]**	18	**1V8**
19	**DATA[9]**	20	**1V8**
21	**DATA[10]**	22	**1V8**
23	**DATA[11]**	24	**1V8**
25	**DATA[12]**	26	NC
27	**DATA[13]**	28	Reserved, do not drive
29	**DATA[14]**	30	Reserved, do not drive
31	**DATA[15]**	32	Reserved, do not drive

Table A-4 Static memory connector signals (continued)

Pin No.	Signal	Pin No.	Signal
33	**DATA[16]**	34	**5V**
35	**DATA[17]**	36	**5V**
37	**DATA[18]**	38	**5V**
39	**DATA[19]**	40	**5V**
41	**DATA[20]**	42	Reserved, do not drive
43	**DATA[21]**	44	Reserved, do not drive
45	**DATA[22]**	46	Reserved, do not drive
47	**DATA[23]**	48	Reserved, do not drive
49	**DATA[24]**	50	Reserved, do not drive
51	**DATA[25]**	52	Reserved, do not drive
53	**DATA[26]**	54	Reserved, do not drive
55	**DATA[27]**	56	Reserved, do not drive
57	**DATA[28]**	58	Reserved, do not drive
59	**DATA[29]**	60	Reserved, do not drive
61	**DATA[30]**	62	**SBSCL**, E2PROM serial interface clock (3.3V signal level)
63	**DATA[31]**	64	**SBSDA**, E2PROM serial interface data (3.3V signal level)
65	**ADDR[0]**	66	**nRESET**
67	**ADDR[1]**	68	**nBOARDPOR**, asserted on hardware power cycle
69	**ADDR[2]**	70	**nFLWP**, flash write protect. Drive HIGH to write to flash.
71	**ADDR[3]**	72	**nEARLYRESET**, Reset signal. Differs from **nRESET** in that it is not delayed by **nWAIT**.

Table A-4 Static memory connector signals (continued)

Pin No.	Signal	Pin No.	Signal
73	**ADDR[4]**	74	**nWAIT**, Wait mode input from external memory controller. Pull HIGH if not used.
75	**ADDR[5]**	76	**nBURSTWAIT**, Synchronous burst wait input. This is used by the external device to delay a synchronous burst transfer if LOW. Pull to HIGH if not used.
77	**ADDR[6]**	78	**CANCELWAIT**, If HIGH, this signal enables the system to recover from an externally waited transfer that has taken longer than expected to finish. Pull LOW if not used.
79	**ADDR[7]**	80	**nCS[4]**
81	**ADDR[8]**	82	**nCS[3]**
83	**ADDR[9]**	84	**nCS[2]**
85	**ADDR[10]**	86	**nCS[1]**
87	**ADDR[11]**	88	Reserved, do not drive
89	**ADDR[12]**	90	Reserved, do not drive
91	**ADDR[13]**	92	Reserved, do not drive
93	**ADDR[14]**	94	Reserved, do not drive
95	**ADDR[15]**	96	**nCS[0]**
97	**ADDR[16]**	98	**nBUSY**, Indicates that memory is not ready to be released from reset. If LOW, this signal holds **nRESET** active.
99	**ADDR[17]**	100	**nIRQ**, interrupt request from memory board. This signal might be used by a memory-mapped peripheral implemented on the board.
101	**ADDR[18]**	102	**nWEN**, write enable for memory on expansion board.
103	**ADDR[19]**	104	**nOEN**, Output enable for memory data.

Table A-4 Static memory connector signals (continued)

Pin No.	Signal	Pin No.	Signal
105	**ADDR[20]**	106	**nBLS[3]**, Byte Lane Select for bits [31:24]
107	**ADDR[21]**	108	**nBLS[2]**, Byte Lane Select for bits [23:16]
109	**ADDR[22]**	110	**nBLS[1]**, Byte Lane Select for bits [15:8]
111	**ADDR[23]**	111	**nBLS[0]**, Byte Lane Select for bits [7:0]
113	**ADDR[24]**	114	**CSWIDTH[0]**, Indicates bus width for fitted part. Do not route through stackable boards.
115	**ADDR[25]**	116	**CSWIDTH[1]**, Indicates bus width for fitted part. Do not route through stackable boards.
117	**ADDRVALID**, Indicates that the address output is stable during synchronous burst transfers	118	**CLK[1]**
119	**BAA**, Burst Address Advance. Used to advance the address count in the memory device	120	**CLK[0]**

a. VDDIO is the data voltage to the host. Do not route through on stackable boards

A.5 Mechanical layout

Figure A-4 shows the static memory expansion board, viewed from above.

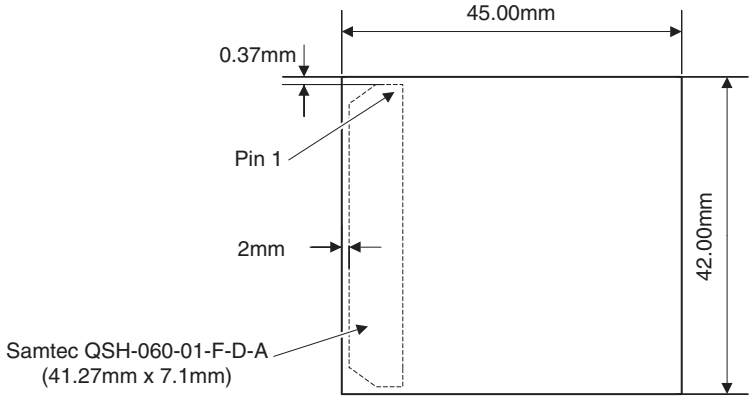

Figure A-4 Static memory board layout

Appendix B
Specifications

This appendix contains the specifications for the Core Tile. It contains the following sections:

- *Electrical specification* on page B-2
- *Mechanical details* on page B-3.

Note

See *AHB bus timing specification* on page 6-37 for the timing specifications for Core Tiles that use the HBI-0131 board.

For timings specific to Core Tiles that use an ARM7TDMI or ARM7TDMI-S test chip, refer to the release notes for the Core Tile.

B.1 Electrical specification

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

B.1.1 Bus interface characteristics

Table B-1 shows the Core Tile electrical characteristics for the system bus interface. The Core Tile uses 3.3V and 5V sources.

Table B-1 Core Tile electrical characteristics

Symbol	Description	Min	Max	Unit
3V3	Supply voltage (interface signals)	3.1	3.5	V
5V	Supply voltage (regulators)	4.75	5.25	V
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
C_{IN}	Input capacitance	-	20	pF

B.1.2 Current requirements

Table B-2 shows current requirements measured at room temperature and nominal voltage. These measurements include the current drawn by Multi-ICE (approximately 160mA at 3.3V).

The power regulators are all powered from the 5V supply. Interface logic on the board is powered from 3.3V.

Table B-2 Current requirements

	3.3V	5V
Core Tile and IM-LT3 Interface Module	1A	1A
Integrator/CP, IM-LT1 Interface Module, and one Core Tile	1.5A	1.5A

Note

The current requirements depend on the test chip used and the application being run.

B.2 Mechanical details

Figure B-1 shows the mechanical outline of the Core Tile.

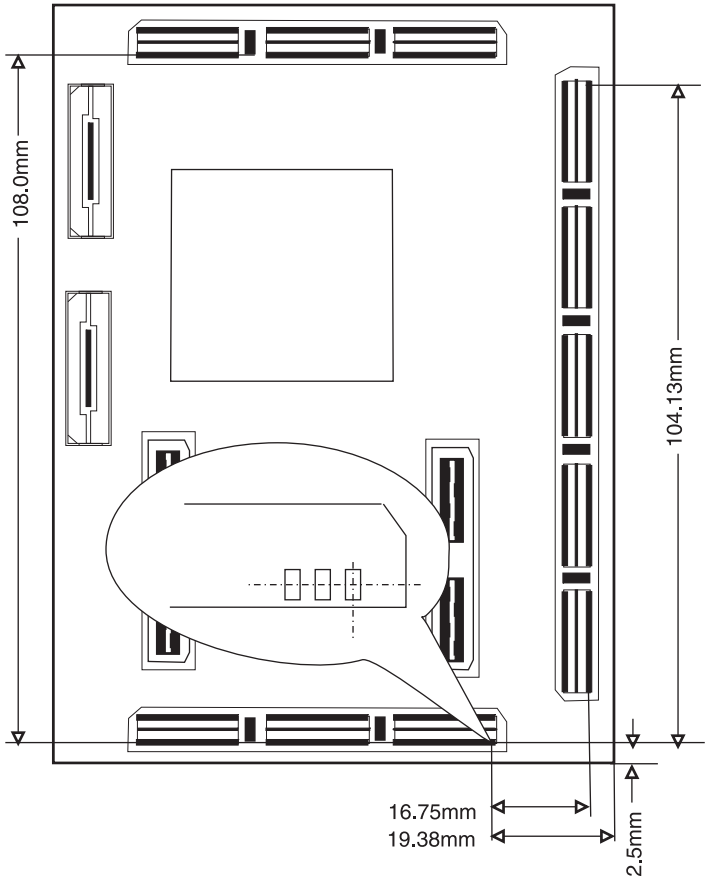

Figure B-1 Board outline (top view)

Glossary

This glossary lists all the abbreviations used in the User Guide.

ADC Analog to Digital Converter. A device that converts an analog signal into digital data.

ADK AMBA Design Kit. The ADK comprises the building blocks required to create an example system based on the low-power, generic design methodology of the Advanced Microcontroller Bus Architecture (AMBA).

AHB Advanced High-performance Bus. The ARM open standard for on-chip buses.

AMBA Advanced Microcontroller Bus Architecture.

APB Advanced Peripheral Bus. The ARM open standard for peripheral buses. This design is optimized for low power and minimal interface complexity

BOM Bill Of Materials. A list of all the parts used on the printed circuit board and any specific build instructions for board variants.

DAC Digital to Analog Converter. A device that converts digital data into analog level signals.

DCC Debug Communications Controller.

ETB Embedded Trace Buffer.

ETM	Embedded Trace Macrocell.
FPGA	Field Programmable Gate Array.
GTC	General Test Chip. A packaging and signal assignment specification for test chips.
ICE	In Circuit Emulator. A interface device for configuring and debugging processor cores.
Integrator/CP	Integrator Compact Platform.
Integrator/IM-LT1	Integrator/IM-LT1 Interface Module.
Integrator/IM-LT3	Integrator/IM-LT3 Interface Module.
I/O	Input/Output.
JTAG	Joint Test Action Group. The committee which defined the IEEE test access port and boundary-scan standard.
Multi-ICE	Multi-ICE is a system for debugging embedded processor cores using a JTAG interface.
PCB	Printed Circuit Board.
PISMO	Memory specification for plug in memory modules.
PLD	Programmable Logic Device.
PLL	Phase-Locked Loop, a type of programmable oscillator.
POR	Power On Reset.
RAM	Random Access Memory.
RVI	RealView ICE. A system for debugging embedded processor cores using a JTAG interface.
SRAM	Static Random Access Memory.
TCM	Tightly Coupled Memory. Memory present inside the test chip that typically runs at or near the processor speed.

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.

T

V