Integrator™/IM-LT3

Interface Module

User Guide

ARM®

ARM DUI 0216B

Integrator/IM-LT3
User Guide

Release Information

Description	Issue	Confidentiality	Change
21 January 2005	A	Non Confidential	New document
20 July 2006	B	Non Confidential	New document

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Confidentiality Status

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

Product Status

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

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

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

CE Declaration of Conformity

The system should be powered down when not in use.

The IM-LT3 Interface Module 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
Integrator/IM-LT3 User Guide

Preface

Chapter 1 Introduction

Chapter 2 Getting Started

Chapter 3 Hardware Description

List of Tables
Integrator/IM-LT3 User Guide

List of Figures
Integrator/IM-LT3 User Guide

Preface

This preface introduces the ARM Integrator/IM-LT3 Interface Module and associated reference documentation. It contains the following sections:

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

About this document

This document describes how to set up and use the ARM Integrator/IM-LT3.

Intended audience

This document has been written to aid experienced hardware and software developers in the development of ARM-based products using the IM-LT3 in a development system.

Organization

This document is organized into the following chapters:

Chapter 1 ***Introduction***

Read this chapter for an introduction to the IM-LT3. This chapter shows the physical layout of the IM-LT3 and identifies the main components.

Chapter 2 ***Getting Started***

Read this chapter for a description of how to set up and start using the Integrator/IM-LT3 with other boards or standalone. This chapter describes how to connect the IIM-LT3 and how to apply power.

Chapter 3 ***Hardware Description***

Read this chapter for a description of the hardware architecture of the IM-LT3. This chapter describes the clocks, resets, and debug hardware provided by the IM-LT3 and the header signals used by attached boards.

Chapter 4 ***Configuring the FPGA and PLD***

Read this chapter for a description of how the Xilinx FPGA in the IM-LT3 is configured at power-up, the configuration options available, and how to download your own FPGA configurations.

Appendix A ***Signal Descriptions***

Refer to this appendix for signal descriptions and connector pinouts.

Appendix B ***Specifications***

Refer to this appendix for electrical and mechanical specifications.

Appendix C ***Using a Core Tile and an IM-LT3 as a Core Module***

This appendix describes the use of a CT7TDMI, CT926EJ-S, CT1026EJ-S, or CT1136JF-S Core Tile and an IM-LT3 to duplicate the functionality of the equivalent Core Module on an Integrator/CP running legacy applications.

Typographical conventions

The following typographical conventions are used in this book:

`monospace` Denotes text that can be entered at the keyboard, such as commands, file and program names, and source code.

`monospace` Denotes a permitted abbreviation for a command or option. The underlined text can be entered instead of the full command or option name.

`monospace italic`

Denotes arguments to commands and functions where the argument is to be replaced by a specific value.

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

bold Highlights interface elements, such as menu names and buttons. Also used for terms in descriptive lists, where appropriate.

`monospace bold`

Denotes language keywords when used outside example code and ARM processor signal names.

Further reading

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

ARM publications

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

- *ARM Integrator/CP User Guide* (ARM DUI 0159)
- *Integrator/IM-LT1 Interface Module User Guide* (ARM DUI 0187)
- *Versatile/Core Tile User Guide* (ARM DUI 0273)
- *Versatile Platform Baseboard for ARM926EJ-S User Guide* (ARM DUI 0224)

- *ARM RealView AT1 Analyzer Tile User Guide* (ARM DUI 0189)
- *ARM RealView IT1 Interface Tile User Guide* (ARM DUI 0188)
- *ARM RealView Logic Tile LT-XC2V4000+ User Guide* (ARM DUI 0186)
- *ARM Multi-ICE User Guide* (ARM DUI 0048)
- *AMBA Specification* (ARM IHI 0011)
- *ARM Architectural 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).
- *ARM RealView Logic Tile LT-XC2V4000+ User Guide* (ARM DUI 0186)
- *RealView™ ICE User Guide* (ARM DUI 0155)
- *RealView™ Debugger User Guid*e (ARM DUI 0153)
- *ETM10 Technical Reference Manual* (ARM DDI 0206)
- *RealView Compilation Tools Compilers and Libraries Guide* (ARM DUI 0205)
- *RealView Compilation Tools Developer Guide* (ARM DUI 0203)
- *RealView Compilation Tools Linker and Utilities Guid*e (ARM DUI 0206).

Note

There are also application notes that cover various combinations of development boards. See the ARM website at `www.arm.com` for details.

Other publications

The following publication provides information about the clock controller chip used on the Integrator module:

- *ICS Serially Programmable Clock Source Data Sheet* (ICS307), MicroClock Division of Integrated Circuit Systems, San Jose, CA.

The following publication provides information about the FPGA devices used on the Integrator IM-LT3, Core Modules, and Logic Tiles:

- *Virtex-II Field Programmable Gate Arrays* (DS031), Xilinx, San Jose, Ca.

Feedback

ARM Limited welcomes feedback both on the ARM Integrator/IM-LT3 Interface Module and on the documentation.

Feedback on this document

If you have any comments about this document, please send email to `errata@arm.com` giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- an explanation of your comments.

General suggestions for additions and improvements are also welcome.

Feedback on the IM-LT3 Interface Module

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

- the product name
- an explanation of your comments.

Chapter 1
Introduction

This chapter introduces the ARM Integrator/IM-LT3 Interface Module. It contains the following sections:

- *About the IM-LT3* on page 1-2
- *IM-LT3 architecture* on page 1-3
- *Typical configurations* on page 1-8
- *Precautions* on page 1-10.

1.1 About the IM-LT3

The Integrator/IM-LT3 Interface Module provides you with a development platform for using ARM Integrator products together with ARM Versatile tiles (for example, the ARM Versatile/XC2V8000 Logic Tile). Typical uses for the Interface Module and tiles include:

- providing an interface between Integrator baseboards or core modules and Versatile tile products
- developing code for an ARM core synthesized in an IM-LT3 FPGA
- developing peripherals for use with an ARM core
- prototyping a *System-On-Chip* (SOC) and developing code for it.

Figure 1-1 shows the IM-LT3 and its components.

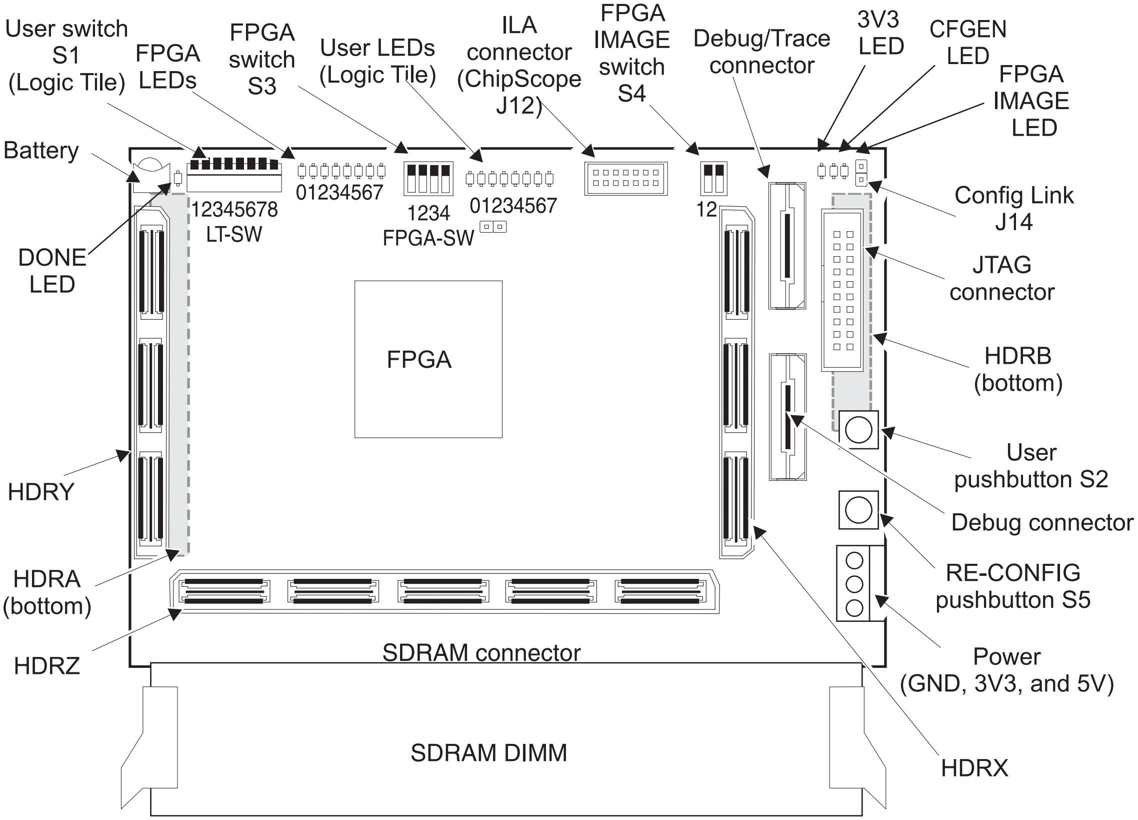


Figure 1-1 Integrator/IM-LT3 Interface Module layout

1.2 IM-LT3 architecture

The IM-LT3 consists of:

- Integrator style connectors HDRA and HDRB on the bottom of the board
- Versatile tile connectors HDRX, HDRY, and HDRZ on the top of the board
- Xilinx Virtex II *Field Programmable Gate-Array* (FPGA). The content of the FPGA depends on the configuration image loaded, but a typical system implements memory controllers and peripherals and also provides the interface logic between the Integrator style headers and the tile headers.
- Flash memory that contains the images for the FPGA. The image is selectable and loaded into the FPGA by the *Programmable Logic Device* (PLD).
- JTAG connector and routing logic for ICE and FPGA image download
- user LEDs and switches
- Logic analyzer connectors:
  - Integrated Logic Analyzer (ChipScope) for FPGA configuration
  - Mictor Logic Analyzer connector (debug signals on **XU[177:146]**)
  - Mictor Logic Analyzer connector (Trace signals on **YU[179:153]**)
- 2MB of 32-bit *Synchronous Static RAM* (SSRAM)
- up to 256MB of 32-bit wide SDRAM (optional) plugged into the *Dual In-line Memory Module* (DIMM) socket
- Power-on reset logic
- clock generators.

Figure 1-2 on page 1-4 shows the architecture of the IM-LT3 Interface Module.

See the following sections for more details on the architecture:

- *FPGA* on page 1-5
- *Memory* on page 1-5
- *Clocks* on page 1-5
- *Expansion headers* on page 1-5
- *Test and debug interfaces* on page 1-6
- *Power supplies* on page 1-7.

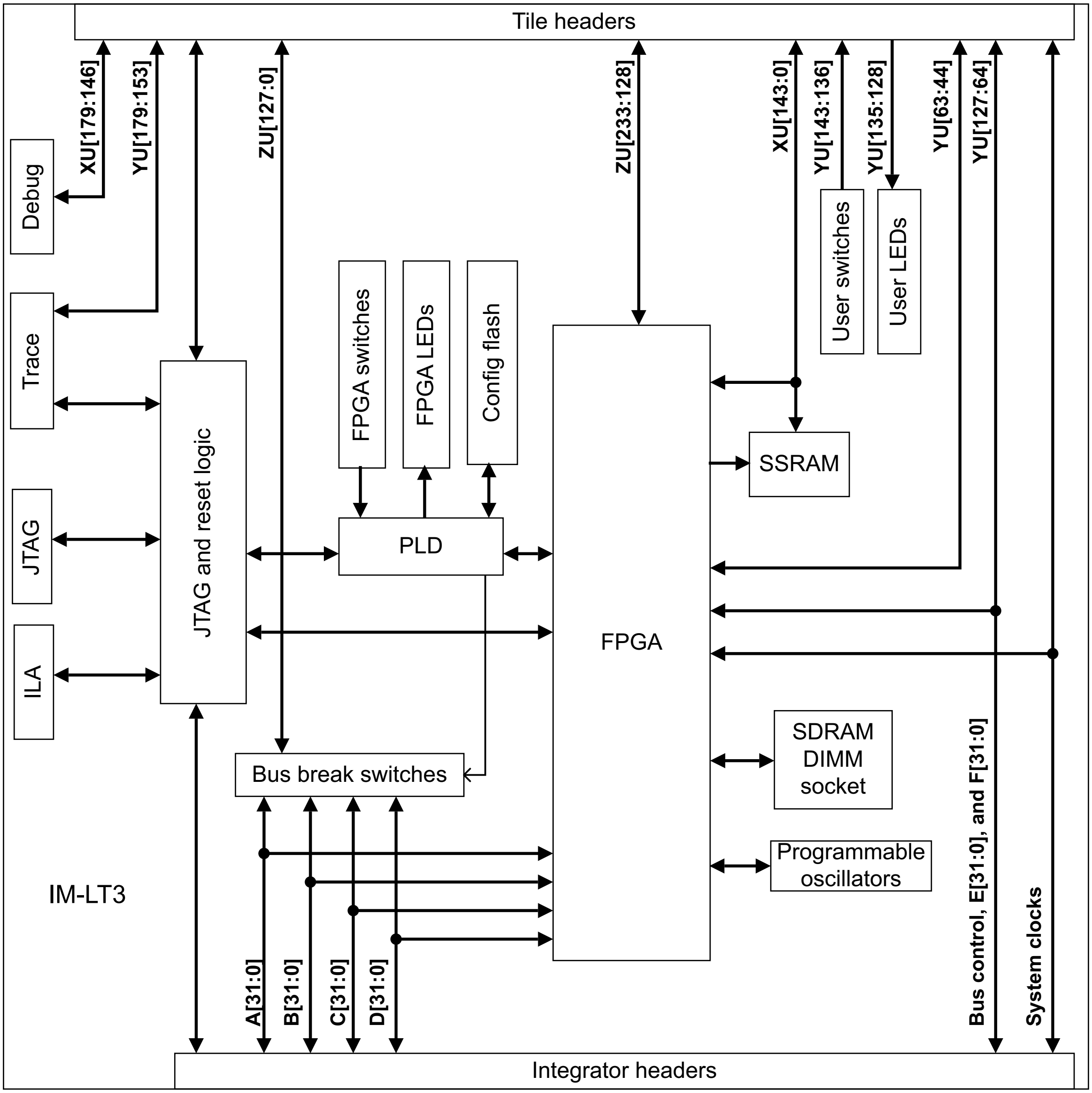


Figure 1-2 ARM Integrator IM-LT3 block diagram

1.2.1 FPGA

Field Programmable Gate-Array (FPGA). The content of the FPGA depends on the configuration image loaded, but a typical system implements:

- interconnections between the Integrator HDRA and HDRB connectors and the Logic Tile HDRX, HDRY, and HDRZ connectors.
- *Synchronous Dynamic Random Access Memory* (SDRAM) controller
- system bus bridge
- *Synchronous Static Random Access Memory* (SSRAM) controller
- interrupt controller
- clock generator controller
- status, configuration, and interrupt registers.

Note

You must program the FPGA with an appropriate image that supports the specific collection of boards that you are using. See the CD supplied with the product for information on available images, VHDL and Verilog source files, and application notes for different board combinations. A flash memory device holds the image that is loaded into the FPGA on power-on.

1.2.2 Memory

The memory system consists of SSRAM memory (32-bits wide) and a plug-in SDRAM DIMM (64-bits wide, no parity, no ECC).

The SDRAM DIMM controller can be implemented within the IM-LT3 FPGA (see the application notes for the IM-LT3). If an appropriate SDRAM controller is included in the FPGA image, the SDRAM can be accessed by the local processor, by processors on other tiles, and by other system bus masters.

1.2.3 Clocks

The IM-LT3 generates four clock sources from on-board programmable oscillators and has connections to external clock sources on the Integrator/CP baseboard. There are also connections to six clocks on attached Logic Tiles. *Delay-Locked Loops* (DLLs) within the FPGA can synchronize local clocks with clocks from other boards.

1.2.4 Expansion headers

The IM-LT3 has three tile expansion headers on the top of the board and two Integrator style headers on the bottom of the board. These connectors enable you to attach, for example, a Logic Tile, a Core Tile, and an Integrator/CP baseboard.

1.2.5 Test and debug interfaces

The IM-LT3 has connectors that enable configuration and monitoring of the board:

JTAG connector

The JTAG connector enables JTAG equipment, such as Multi-ICE or RealView ICE, to be connected to the IM-LT3. The JTAG signals are routed to all tiles stacked on the IM-LT3 and enable configuration of all programmable logic and onboard flash memory. See *JTAG support* on page 3-25.

Always use the JTAG connector on the IM-LT3 Interface Module.

Trace and logic analyzer connectors

There are two Mictor logic analyzer connectors on the IM-LT3 that are used for general-purpose debug. If an *Soft Macrocell Model* (SMM) core is implemented in the FPGA on an attached Logic Tile however, one connector can be configured as a trace port (up to 16-bits wide) and the trace signals from the ETM can be monitored by a Trace Port Analyzer. (See *Mictor logic analyzer connectors* on page A-24.)

Each Core Tile has trace connectors for the test chip on the board. In some stacking configurations, a right-angle adapter cable is required for connecting a trace unit to the trace port connectors.

Integrated Logic Analyzer connector

The IM-LT3 also has an *Integrated Logic Analyzer* (ILA) connector that can be used with Xilinx ChipScope analyzer. This connector provides access to the JTAG configuration chain when the normal JTAG chain is in debug mode.

The IM-LT3 also has LEDs and switches that can be used to configure or test images implemented in the IM-LT3 or Logic Tile FPGAs:

LEDs There are eight LEDs that can be lit by the IM-LT3 FPGA and eight LEDs that can be lit by FPGA signals from an attached Logic Tile.

switches There are four user switches that are connected to the IM-LT3 FPGA nd eight switches that can are connected to the FPGA in an attached Logic Tile.

Two image selector switches enable different FPGA images to be loaded at reset. This is useful when testing a new design.

The general purpose pushbutton S2 generates the **nPB** signal to the FPGA. For the Integrator/CP FPGA image, this signal triggers a reset of the board stack.

1.2.6 Power supplies

The IM-LT3 requires two voltage sources:

5V Power supply for expansion circuitry

3.3V Power supply for FPGA and interface circuits.

A 1.5V DC supply is derived from the 5V supply by an on-board switching regulator to supply the FPGA core voltage.

The voltages can be supplied by one of:

Screw terminals on the IM-LT3

External 5V and 3.3V power supplies can be connected to the screw terminals if the IM-LT3 is used without an Integrator/CP baseboard.

Integrator/CP baseboard

If the IM-LT3 is fitted to an Integrator/CP, the 5V and 3.3V supplies are sourced from the regulators on the CP. Connect an external power supply to the Integrator/CP only.

——— **Caution** ————

Attach the power supply to the lowest board in the stack that has a power connector.

Do not use the power connections on the IM-LT3 if the Interface Module is fitted to an Integrator/CP.

————————————

Tile header HDRY and HDRX have an option for some of the interface signals to operate at a different signal level than the 3.3V default. The **VDDIO_Y** and **VDDIO_X** supplies are normally sourced from a Logic Tile or Core Tile attached to the IM-LT3. Jumper options allow the supplies for these signals to be sourced from 3.3V, see *Voltage on header pins* on page 3-12 for details.

1.3 Typical configurations

Typical configurations for the IM-LT3 are listed in Table 1-1. 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.

Table 1-1 Typical configurations

Baseboard option	Core Module	Logic Tile	Core Tile	Description
No baseboard	No	0-3	No	Standalone SMM system. The Logic TIle FPGAs are loaded with a *Soft Macrocell Model* (SMM) implementation of a processor and is used to test processor designs. The IM-LT3 FPGA contains a design that provides control functions and enables the SMM to access the SDRAM on the IM-LT3.
No baseboard	1	0-2	No	A single-processor standalone system based on a Core Module. The optional Logic Tiles contain peripheral implementations
No baseboard	No	0-2	1	A single-processor standalone system based on a Core Tile.
No baseboard	0 or 1	0-2	1 or 2	A multi-processor standalone system based on Core Tiles. If more than one Core Tile is used, each Core Tile must be separated by a Logic Tile.
No baseboard	1-3	0-2	0, 1 or 2	A multi-processor standalone system based on Core Modules. If more than one Core Tile is used, each Core Tile must be separated by a Logic Tile.
Integrator/CP	No	0-3	No	Expanded SMM system. The Logic Tile FPGAs are loaded with an SMM. The IM-LT3 FPGA contains a design that enables the SMM to access the SDRAM on the IM-LT3 and the peripherals on the baseboard.
Integrator/CP	1	0-2	No	A single processor system with the Integrator/CP providing I/O. The IM-LT3 and optional Logic Tiles contain peripheral implementations
Integrator/CP	No	0-2	1	A single processor system with the Integrator/CP providing I/O. The optional Logic Tiles contain peripheral implementations

Note

See the applications notes provided on the CD and on the ARM web site for more details of stacking options and FPGA images. Not all board combinations have an FPGA image supplied.

The maximum number of Core Modules, Logic Tiles, and Core Tiles is limited by the power supply.

Figure 1-3 shows an Integrator/CP system consisting of an Integrator/CP, an IM-LT3, and a Core Tile.

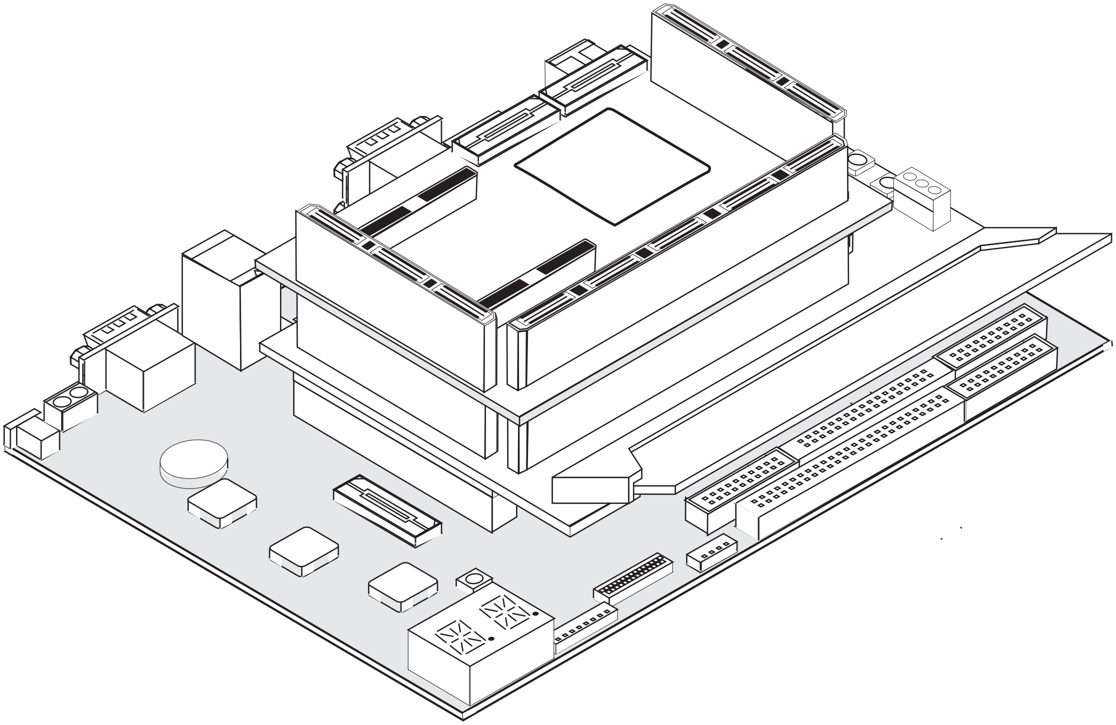


Figure 1-3 Integrator/CP system

1.4 Precautions

This section contains safety information and advice on how to avoid damage to the IM-LT3.

1.4.1 Ensuring safety

The IM-LT3 is powered from 3.3V and 5V DC supplies.

Warning

To avoid a safety hazard, only connect *Safety Extra Low Voltage* (SELV) equipment to the IM-LT3 and any attached boards.

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

1.4.2 Preventing damage

The IM-LT3 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 board, observe the following precautions.

- Never subject the board to high electrostatic potentials.
- Always wear a grounding strap when handling the board.
- Only hold the board by the edges.
- When removing the module from a baseboard, take care not to damage the connectors.
- Do not apply a twisting force to the ends of the connectors.
- Avoid touching the component pins or any other metallic element.
- Ensure that the voltage on the pins of the FPGA and interface circuitry on all connected Logic Tiles is at the correct level. Individual FPGA pins are not damaged by a clash of output levels, but this causes increased power consumption that might damage the FPGA.
- Do not use the board near a transmitter of electromagnetic emissions.
- Use the IM-LT3 and tiles in a clean environment and avoid debris fouling the connectors on the underside of the PCB. Blocked holes might damage the connectors.
- Visually inspect the baseboard pins to ensure that they are not bent and visually inspect the tile to ensure that connector holes are clear before mounting it onto another board.
- If a stack of boards are used, there must be only one connection to the power supply (see *Power supplies* on page 1-7).

Chapter 2
Getting Started

This chapter describes how to set up and prepare the IM-LT3 Interface Module for use. It contains the following sections:

- *Using the IM-LT3 without a baseboard* on page 2-2
- *Using the IM-LT3 with an Integrator/CP* on page 2-9
- *Connecting Multi-ICE, RealView ICE or Trace* on page 2-11
- *Switches, links, and indicators* on page 2-13.

——— **Note** ———

The information in this chapter covers the mechanical assembly of a development system. Different combinations of IM-LT3, Integrator/CP baseboard, Core Modules, Logic Tiles, and Core Tiles require specific FPGA images.

For details on selecting and loading the correct FPGA image, see the ARM website (at `www.arm.com`) for application notes for the product combination you are using.

2.1 Using the IM-LT3 without a baseboard

In standalone operation (without an Integrator/CP baseboard) the IM-LT3 provides the power connections and system controller.

The processor for the system can be provided by either:

- a Core Module installed below the IM-LT3
- an SMM (Soft Macrocell Model) processor implemented in the IM-LT3 (or in attached Logic Tiles)
- a Core Tile on top of the IM-LT3.

To use the IM-LT3 as a standalone system:

1. If the system is not using an SMM to provide the processor either:
   - connect a Core Module to the HDRA/HDRB connectors underneath the IM-LT3 (see *Fitting a Core Module to the IM-LT3* on page 2-3)
   - connect a Core Tile to the HDRX/HDRY/HDRZ connectors on top of the IM-LT3 (see *Fitting a Core Tile to the IM-LT3* on page 2-5).

   Note

   Create a multi-processor system by adding multiple Core Modules, Core Tiles, or a mix of Core Modules and Core Tiles.

   See the applications notes included on the CD and on the ARM website for more details of stacking options and supplied FPGA images.

2. Optionally, fit an SDRAM DIMM (see *Fitting an SDRAM DIMM* on page 2-4).
3. Optionally, connect one or more Logic Tiles to the IM-LT3. The Logic Tiles can be used for processor or peripheral development.

   An IT1 Interface Tile or AT1 Analyzer Tile can be mounted on the top of the stack to provide I/O connectors to peripherals implemented in the Logic Tile or IM-LT3 FPGA.
4. Supply power to the IM-LT3 (see *Supplying power to the IM-LT3* on page 2-8).
5. Connect a JTAG device (Multi-ICE or RealView ICE) to the IM-LT3 (see *Connecting Multi-ICE, RealView ICE or Trace* on page 2-11).
6. Download the appropriate images to the FPGAs present on the IM-LT3 and Logic Tiles (see Chapter 4 *Configuring the FPGA and PLD*).

   See also the application notes covering the combination of boards for details on FPGA and PLD images.

2.1.1 Fitting a Core Module to the IM-LT3

Connect a Core Module to the HDRA/HDRB connectors on the IM-LT3 to provide a system processor (if an SMM implemented in the IM-LT3 is not used)

——— **Note** ———

If you are using a Core Module without a baseboard below it, connect the stacking option link shorting out the motherboard detect signal (**nMBDET**) on the Core Module. This link is usually numbered LK1 or LK11 and the **nMBDET** signal is shorted if the link is in the B:C position instead of the normal A:C position. Some versions of Core Module do not have this link and cannot be used as the bottom module in a stack.

———————————

Figure 2-1 illustrates a development system with an IM-LT3 and a Core Module.

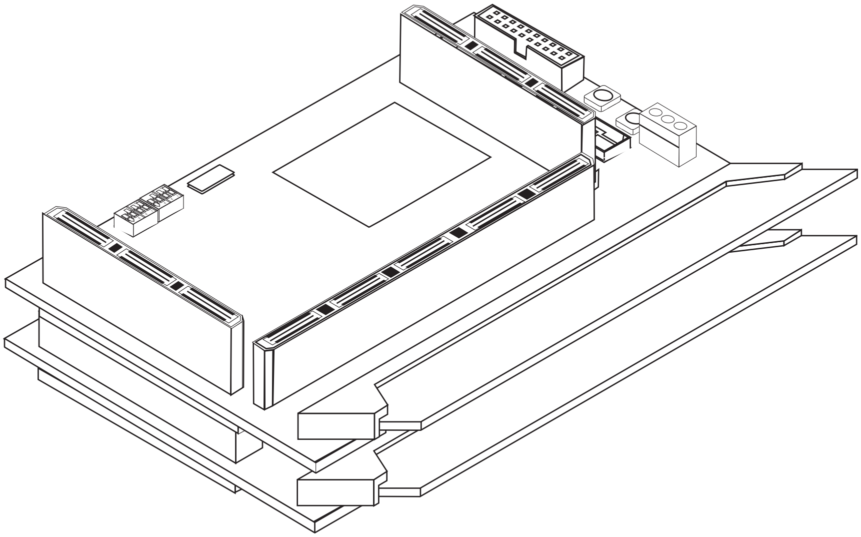


Figure 2-1 IM-LT3 and Core Module

2.1.2 SDRAM

The IM-LT3 has a socket for an SDRAM DIMM.

Fitting an SDRAM DIMM

You must fit the following type of SDRAM:

- PC66, PC100, or PC133- compliant 168-pin DIMM
- unbuffered
- 3.3V
- 16MB, 32MB, 64MB, 128MB or 256MB.

To install an SDRAM DIMM:

1. Ensure that the IM-LT3 is powered down.
2. Open the SDRAM retaining latches outwards.
3. Press the SDRAM DIMM into the edge connector until the retaining latches click into place.

Note

The DIMM edge connector has polarizing notches to ensure that it is correctly oriented in the socket.

Using the IM-LT3 without SDRAM

You can operate the IM-LT3 without an SDRAM module because it has 2MB of SSRAM permanently fitted.

Note

If you are using the RVD debugger, you must change the `top_of_memory` internal variable to match the installed memory. For further information about ARM debugger internal variables, refer to the *RealView Debugger User Guide*.

If you are using AXD, refer to the *AXD and armsd Debuggers Guide* for details on using the debugger.

2.1.3 Fitting a Core Tile to the IM-LT3

Connect a Core Tile to the IM-LT3 to provide a system processor (if an SMM or Core Module is not used).

Figure 2-2 illustrates a development system with an IM-LT3 and a Core Tile.

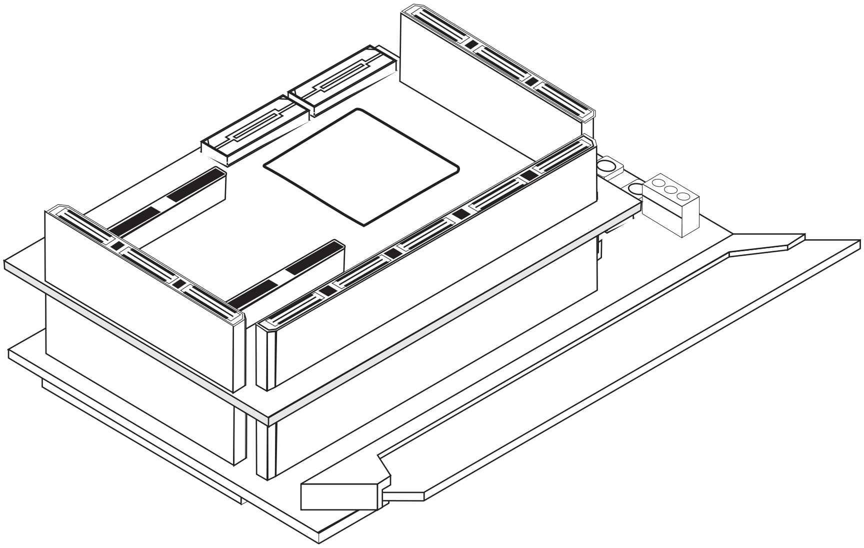


Figure 2-2 IM-LT3 and Core TIle

2.1.4 Fitting Logic Tiles to the IM-LT3

Connect a Logic Tile to the HDRX/HDRY/HDRZ connectors on the IM-LT3 to:

- provide additional capacity for implementing soft cores
- implement peripherals in the FPGAs on the Logic Tiles
- provide an interface between Core Tiles in a multi-processor system.

Up to three Logic Tiles can be stacked if you are using the tiles for SMM or peripheral development.

Figure 2-3 illustrates a development system with an IM-LT3 and a Logic Tile.

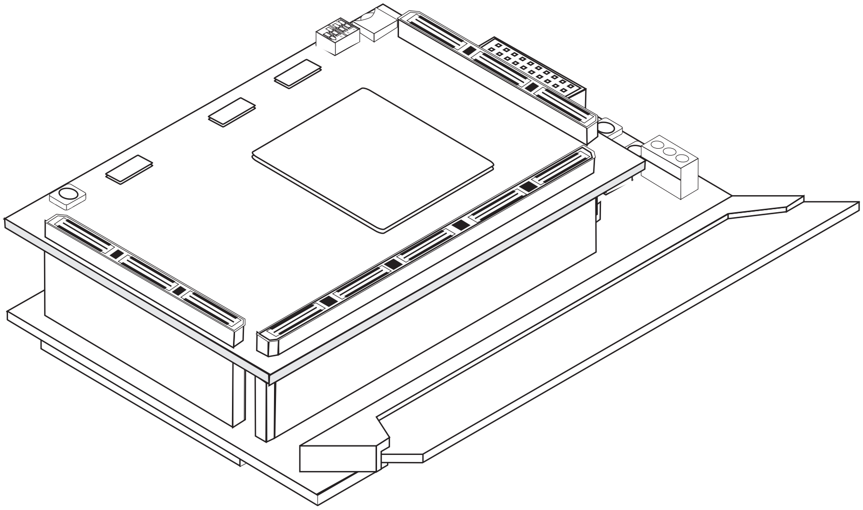


Figure 2-3 IM-LT3 and Logic Tile

2.1.5 Fitting IT1 Interface Tiles to the IM-LT3

The IT1 Interface Tile provides buffer logic, I/O connectors, and a prototyping area. Add the IT1 to the top of the stack to provide interfaces to the peripherals implemented in the Logic Tile or IM-LT3 FPGA.

Figure 2-4 illustrates a development system with an Interface Tile mounted on the top of the stack.

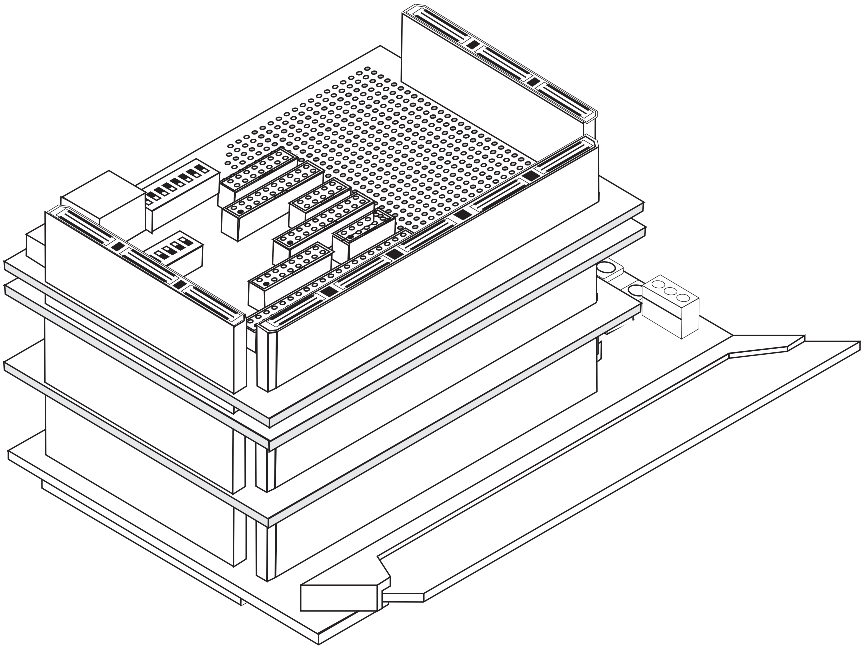


Figure 2-4 IM-LT3, Core Tile, Logic Tile and Interface TIle

2.1.6 Supplying power to the IM-LT3

When using the IM-LT3 as a standalone development system, you must connect a bench power supply with 3.3V and 5V outputs to the screw-terminal power connector as illustrated in Figure 2-5.

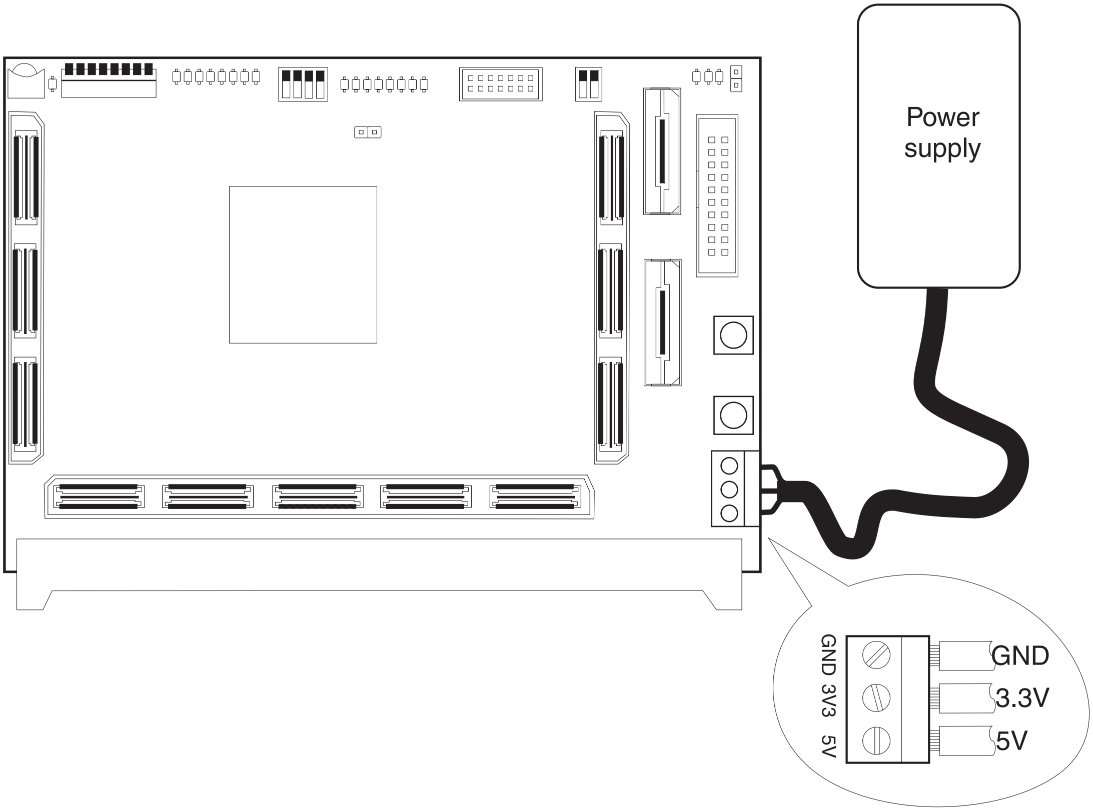


Figure 2-5 Power connector on IM-LT3

Caution

Do not use the power connection if the IM-LT3 is fitted to an Integrator/CP baseboard (see *Power supplies* on page 1-7).

Both the 3.3V and 5V supplies must be turned on and off at the same time.

Voltage supply tolerance for both supplies must be within plus or minus 5%.

2.2 Using the IM-LT3 with an Integrator/CP

Attach the IM-LT3 onto a Integrator/CP baseboard by engaging the connectors on the bottom of the Interface Module with the corresponding connectors on the top of the baseboard.

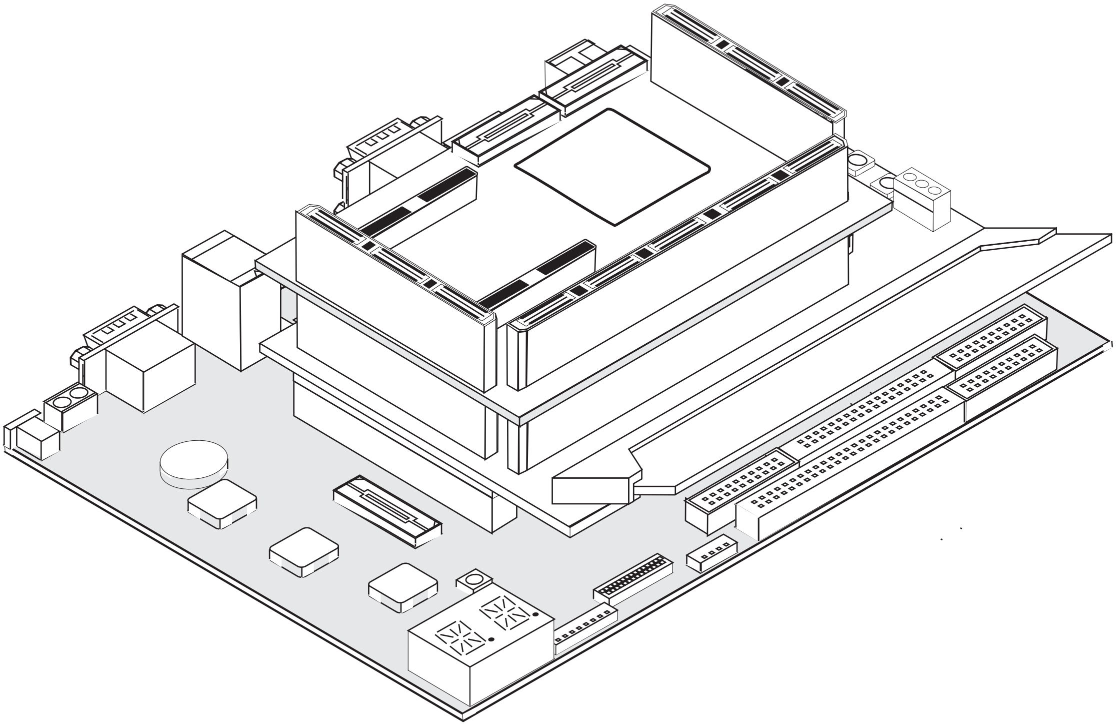


Figure 2-6 Integrator/CP with IM-LT3 and Core Tile

The processor (or processors) for the system can be provided by:

- a Core Module installed between the Integrator/CP and the IM-LT3
- as a soft core implemented in the IM-LT3 (or attached Logic Tile)
- a Core Tile on top of the IM-LT3.

To fit the IM-LT3 to the Integrator/CP:

1. If the system is not using a Core Tile or SMM to provide the processor, connect a Core Module to the HDRA/HDRB connectors on the Integrator/CP.
2. Mount the IM-LT3 on to the HDRA/HDRB connectors of the Integrator/CP (or the Core Module if used).
3. Optionally, fit an SDRAM DIMM (see *Fitting an SDRAM DIMM* on page 2-4).

4. If the system is not using a Core Module or SMM to provide the processor, connect a Core Tile to the IM-LT3 HDRX/HDRY/HDRZ connectors.
5. Optionally, connect one or more Logic Tiles to the IM-LT3.
6. Supply power to the CP baseboard by connecting a bench power supply to the Integrator/CP power terminals or the supplied brick power supply to the DC INPUT jack. (Do not connect power to both the Integrator/CP and the IM-LT3, see also *Power supplies* on page 1-7)
7. Connect a JTAG device (Multi-ICE or RealView ICE) to the IM-LT3 (see *Connecting Multi-ICE, RealView ICE or Trace* on page 2-11).
8. Download the appropriate images to the FPGAs present on the IM-LT3 and Logic Tiles (see Chapter 4 *Configuring the FPGA and PLD*).

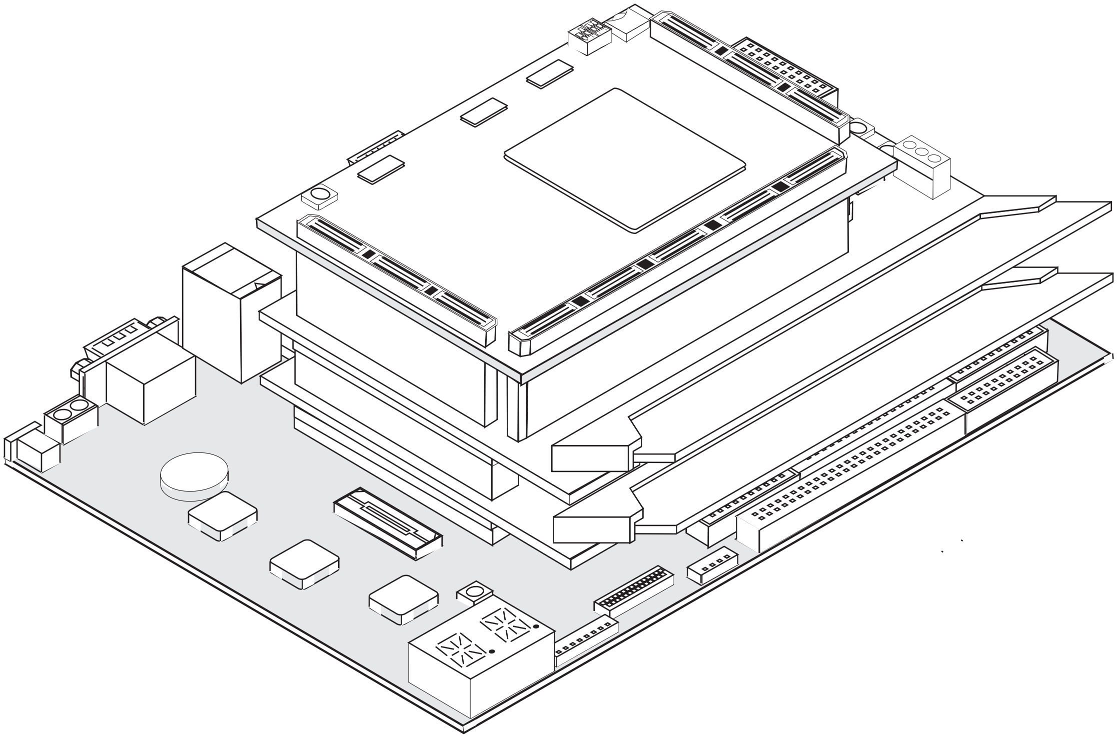


Figure 2-7 CP with Core Module, IM-LT3, and Logic Tile

The general purpose pushbutton S2 generates the **nPB** signal to the FPGA. For the supplied Integrator/CP FPGA image, this signal triggers a reset of the board stack.

2.3 Connecting Multi-ICE, RealView ICE or Trace

JTAG debugging equipment can be used to:

- download images to FPGAs and PLDs on attached Logic Tiles or Core Modules
- download images the FPGA configuration flash or to the PLD on the IM-LT3
- debug an application (the system must contain a Core Module, Core Tile, or an SMM image loaded into the IM-LT3 or an attached Logic Tile.)

The JTAG connection to the IM-LT3 is shown in Figure 2-8.

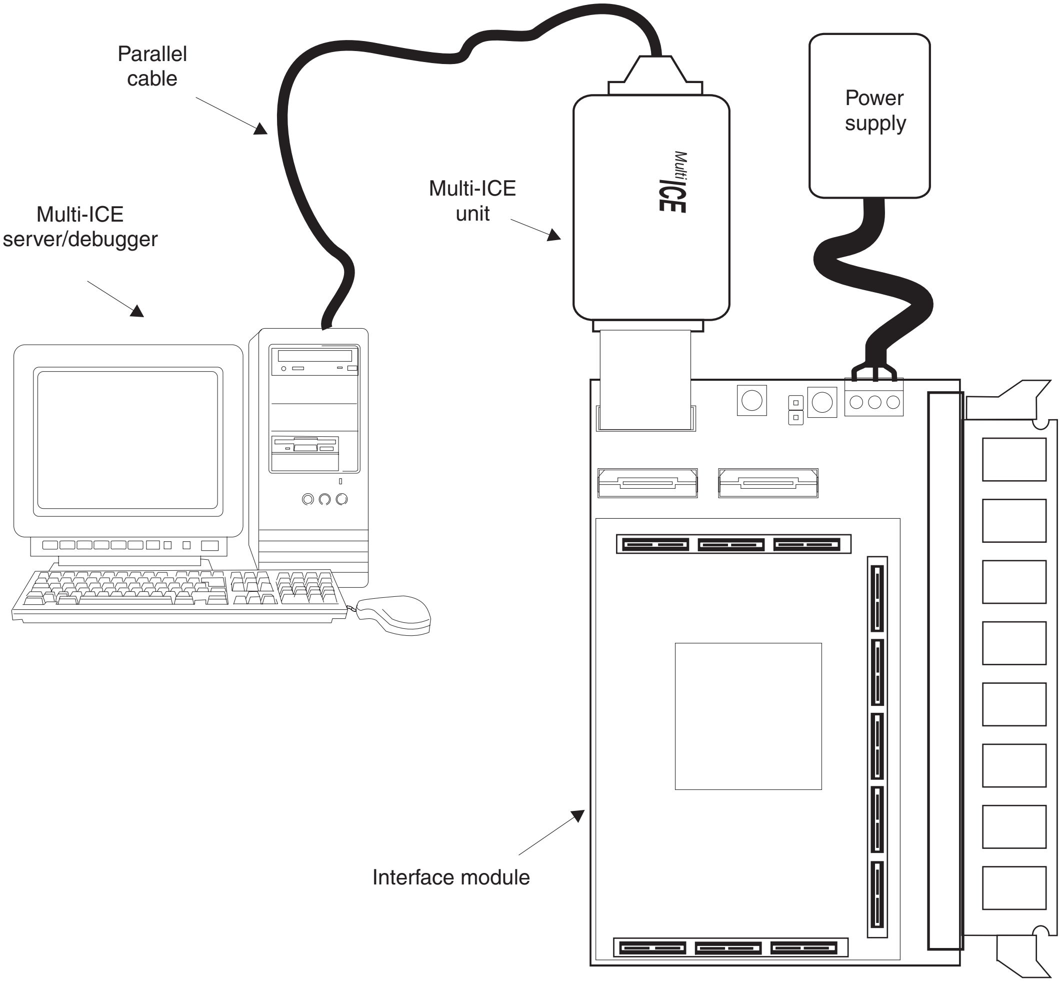


Figure 2-8 Multi-ICE connection to the Interface Module

2.3.1 Connecting Trace

Trace connection is to the Mictor trace connectors on the Core Tile or Core Module.

Depending on stacking arrangement, a right-angle adapter cable might be required to access the trace connectors (see the BLUE RIBBON coax assemblies on the Precision Interconnect website at `www.precisionint.com`).

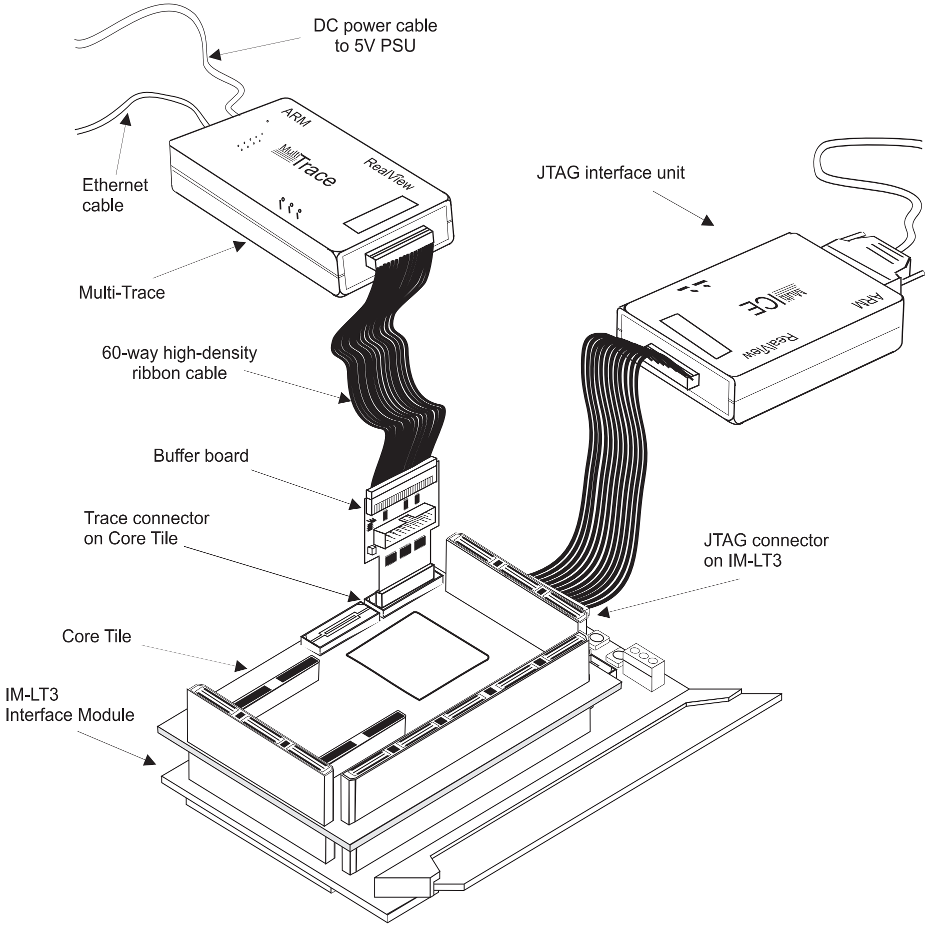


Figure 2-9 Trace connection

——— **Note** ———

If an SMM core is implemented in an attached Logic Tile FPGAs, Mictor connector J11 can be configured as a basic trace port (see *Debug connectors* on page 3-43).

2.4 Switches, links, and indicators

The IM-LT3 link, switches, and LEDs are illustrated in Figure 2-10.

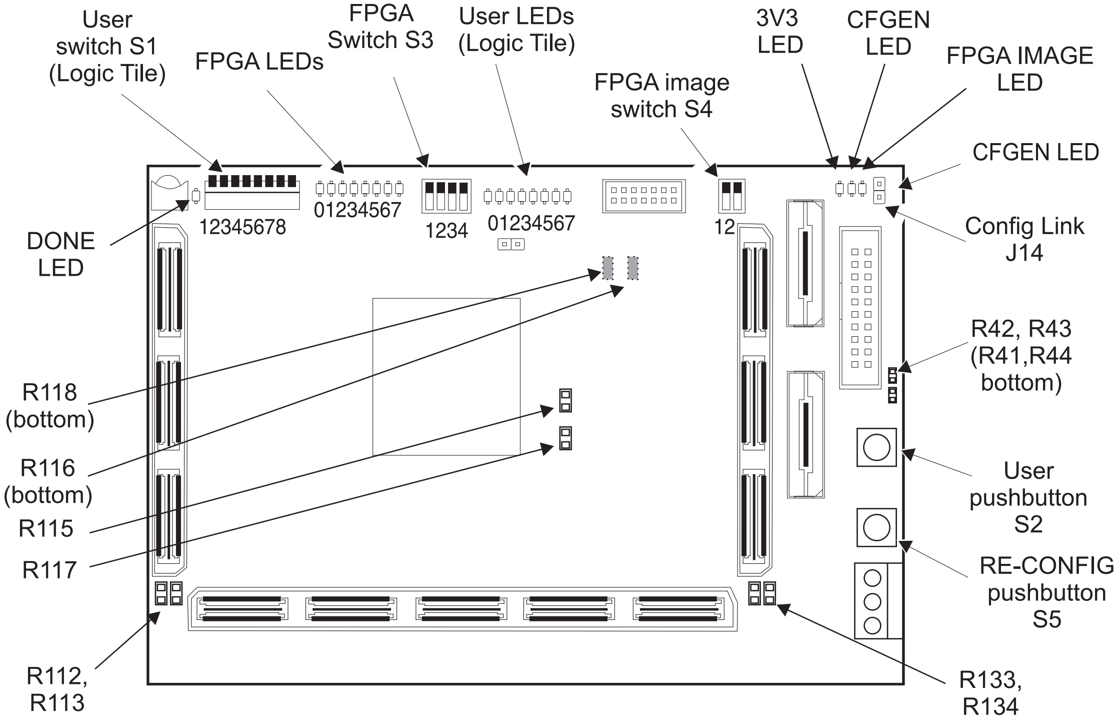


Figure 2-10 Links, switches, and indicators

See *Switches and LEDs* on page 3-39 for more details.

2.4.1 Image selector switches

The IM-LT3 has two FPGA image selection switches (marked FPGA image switch in Figure 2-10) that are used to select FPGA images under local or global control.

The **CFGSEL[1:0]** signals are set from an attached motherboard are not used to select the image in the PLD image supplied at manufacture. Resistor links R41, R42, R43, and R44 set the default value if no motherboard is present. See Chapter 4 *Configuring the FPGA and PLD*.

2.4.2 User switches

Switches on S1 (signals **YU[143:136]**) can be read from the Logic Tile fitted to the IM-LT3.

Switches on S3 (signals **SW[3:0]**) are connected to the FPGA and can be read locally.

The general purpose pushbutton S2 generates the **nPB** signal to the FPGA. For the Integrator/CP FPGA image, this signal triggers a reset of the board stack.

2.4.3 CONFIG link

The IM-LT3 has only one link for configuration enable (marked CFGEN). This is left open during normal operation. It is only fitted when downloading new FPGA or PLD configuration information.

2.4.4 User LEDs

The eight user LEDs (signals **YU[135:128]**) can be lit by a connected Logic Tile.

The eight FPGA LEDs (signals **R_LEDS[7:0]**) are connected to the FPGA and can be lit by user logic implemented in the FPGA.

2.4.5 Status indicators

The functions of the surface-mounted LEDs are summarized in Table 2-1.

Table 2-1 LED functional summary

Name	Color	Function
D20 DONE	Green	This LED illuminates when the FPGA has successfully loaded its configuration information following power-on.
D19 POWER	Green	This LED illuminates to indicate that a 3.3V supply is present.
D25 CFGEN	Orange	This LED illuminates to indicate that the CONFIG link is fitted.
D21 FPGA IMAGE	Green	This LED illuminates to indicate that image one or three from the flash memory is loaded into the IM-LT3 FPGA. This also indicates the state of the **FPGA_IMAGE** signal to any attached tiles.

2.4.6 Voltage selection links

Tile header HDRY and HDRX have an option for some of the interface signals to operate at a different signal level than the 3.3V default. The **VDDIO_Y** and **VDDIO_X** supplies are normally sourced from a Logic Tile or Core Tile attached to the IM-LT3. **VDDIO_X** and **VDDIO_Y** are connected to 3.3V by MOSFET transistors if no tile is present.**VDDIO_X** and **VDDIO_Y** can be manually connected to 3.3V by fitting resistor links on the PCB, but normally it is supplied only by the tile above See *Voltage on header pins* on page 3-12 for details.

Resistor links R115 and R117 select between 3.3V and **VDDIO_X** for the SSRAM IO signal level and R117 and R118 select the SSRAM supply voltage. These resistors are fitted at manufacture to match the SSRAM present on the board.

Chapter 3
Hardware Description

This chapter describes the on-board hardware. It contains the following sections:

3.1 Differences between IM-LT3 and IM-LT1

The IM-LT3 Interface Module is similar to the IM-LT1 Interface Module. The IM-LT1 allows the Integrator system to access resources on the Logic Tiles, but it does not have an FPGA fitted. The differences between the two modules are listed in Table 3-1. For more information on the IM-LT1, see the *Integrator/IM-LT1 User Guide*.

Note

The signal names for the same pin differ between the IM-LT1 and IM-LT3 because the IM-LT1 does not have an FPGA fitted.

Table 3-1 Differences between IM-LT1 and IM-LT3

IM-LT3 logic	Fitted on IM-LT1	Reference for IM-LT3
FPGA	No	*FPGA* on page 3-3
PLD	No	*PLD circuitry* on page 3-5
Image select switches and LED indicator	No	*Switches and LEDs* on page 3-39
User switches and LEDs	Yes	
JTAG connector	Yes	*JTAG support* on page 3-25
Trace connectors	Yes	*Debug connectors* on page 3-43
Local clock sources	No	*Clock architecture* on page 3-15
HDRZ isolation switches	No	*Header connectors* on page 3-12
SSRAM and SDRAM	No	*Memory interfaces* on page 3-23
Reset and POR circuitry	Yes	*PLD circuitry* on page 3-5

3.2 FPGA

The IM-LT3 FPGA provides sufficient functionality for the IM-LT3 to operate as a standalone development system. The IM-LT3 can also be used with an Integrator/CP baseboard to provide interface to Core Modules, Logic Tiles, and Core Tiles.

If you are using the IM-LT3 with an Integrator baseboard, the system bus arbitration, system interrupt control, and input/output resources are provided by the system controller FPGA on the baseboard. See the user guide for your baseboard for further information.

Figure 3-1 on page 3-4 illustrates the function of the IM-LT3 FPGA and shows how it connects to the other devices in the system.

At power-up the FPGA loads its configuration data from a flash memory device. Parallel data from the flash is streamed by the *Programmable Logic Device* (PLD) into the configuration ports of the FPGA. See *PLD circuitry* on page 3-5 for details of the FPGA configuration mechanism. The FPGA contains an image encryption key stored in volatile memory. A backup battery provides power to the key memory. If the battery is removed or shorted, the encryption key will be erased and the FPGA image cannot be loaded.

The JTAG interface can be used to reprogram the PLD, FPGA, and flash when the system is placed in configuration mode. See *JTAG support* on page 3-25.

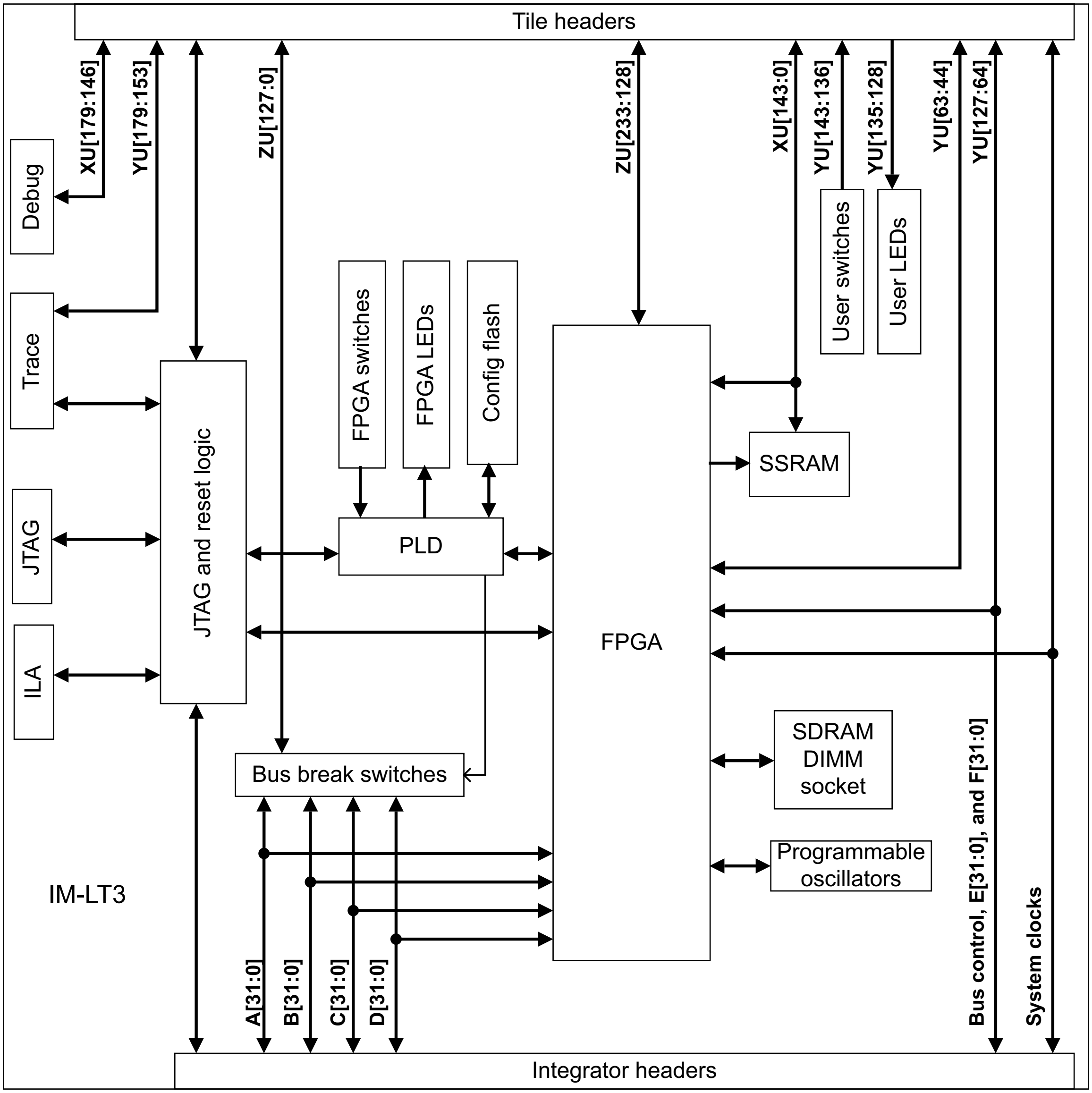


Figure 3-1 IM-LT3 block diagram

3.3 PLD circuitry

The PLD loads an image from flash into the FPGA at power-on or reset. See *Reset control* on page 3-8 for details of the reset circuitry.

The FPGA image load sequence consists of:

- Reading the **CFGSEL[1:0]** signals and the SEL1 and SEL2 image switches (SW4[1] and SW4[2]) and selecting the lower or upper flash image.

  ——— **Note** ———

  Only SW4[1] is used to select the FPGA image. SW4[2] and the **CFGSEL[1:0]** signals are reserved for future use.

  ———————

  See *Configuring the FPGA from flash* on page 4-5 for more details on image selection.

- Setting the **FPGA_IMAGE** LED signal to indicate which image is loaded. The **FPGA_IMAGE** signal to the tile headers is also driven to indicate the image selected.

- Loading the image into the FPGA from the configuration flash

  ——— **Note** ———

  The signals **LEDS[7:0]** are used during configuration as the data bus between the PLD and the FPGA. After configuration, the FPGA drives these signals to light the LEDs.

  ———————

- Lighting the DONE LED after the FPGA is configured. The **GLOBAL_DONE** signal floats high to indicate that configuration is complete. (See also *Reset control* on page 3-8.)

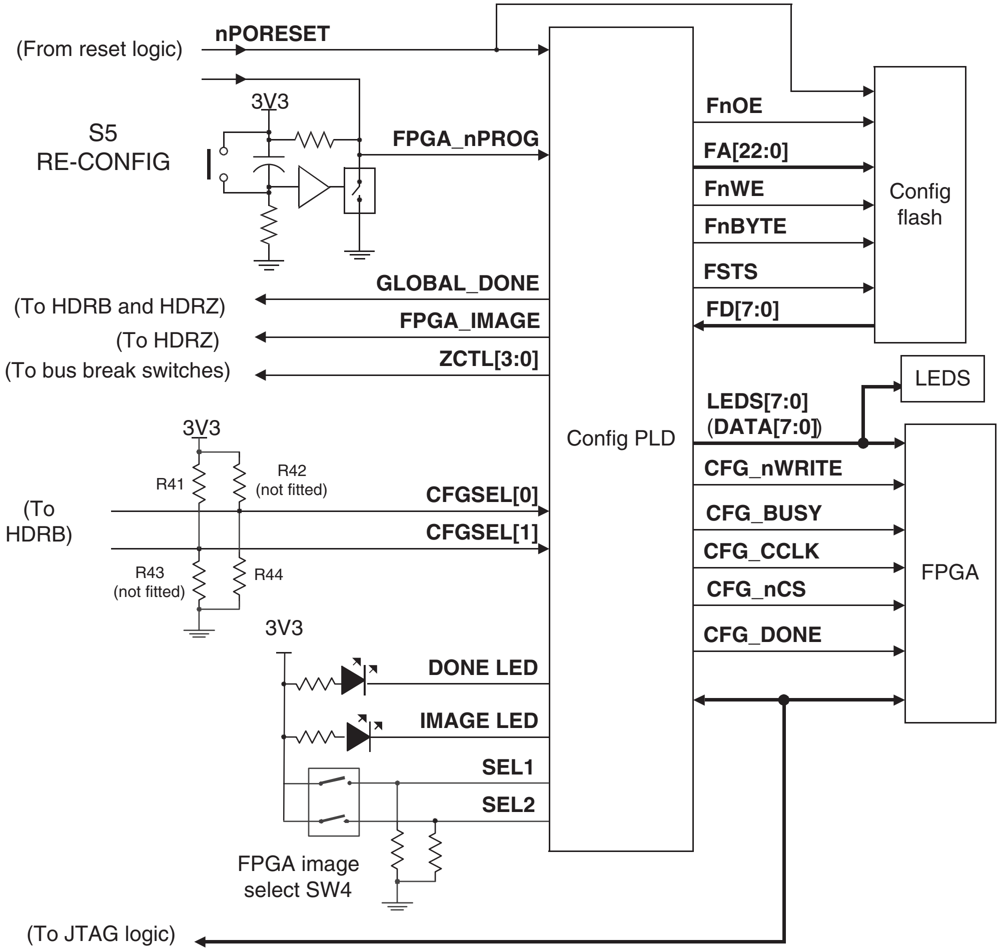


Figure 3-2 PLD logic and FPGA configuration

3.3.1 PLD configuration

The IM-LT3 PLD is preloaded with a nonvolatile image.

In addition to control the loading of the FPGA image, the PLD also provides the **nRTCKEN** signal to the motherboard and the signals to control the isolation switches. The values for these signals are loaded serially from the FPGA after it has been configured. If you are not using the example FPGA design, you must include a design to perform the serial transfer at startup.

Note

The PLD can be programmed to directly control the bus break switches. The switches, however, are normally controlled by the FPGA sending serial configuration data to the PLD (see *Isolation of HDRZ signals* on page 3-14).

Caution

Loading an incorrect image into the PLD might render the board unusable. If the image in the PLD has been accidently erased, reload the image into the PLD by inserting the CONFIG link on the Interface Module and using the Progcards utility.

Table 3-2 lists the configuration signals that are received by the PLD at power on.

Table 3-2 Power-on configuration signals by clock cycle

Clock	PLD data input	Description
`0 to` *n*	LOW	LOW transmitted until FPGA ready to send data
n+1	**nRTCKEN**	Bypass control for JTAG return clock
n+2	**ZCTL[3]**	ZCTL data is clocked into the PLD and the state of the MANID links is clocked out of the PLD.
n+3	**ZCTL[2]**	
n+4	**ZCTL[1]**	
n+5	**ZCTL[0]**	
n+6	HIGH	Indicates end of transmitted data
-	LOW	LOW transmitted until system is reset.

3.4 Reset control

Figure 3-3 on page 3-10 shows the architecture of the reset circuitry. The PLD incorporates a reset controller that enables the Core Module to be reset as a standalone unit or as part of an Integrator development system. The system can be reset from by:

- powering on the system (**nPORESET**)
- pressing pushbutton S2 (for Integrator/CP systems)
- **nSYSRST** from a motherboard (for example, from an Integrator/CP) motherboard) or a Core Module
- attached tiles driving **D_nSRST**, **C_nSRST**, **D_nTRST**, or **C_nTRST** LOW
- a JTAG device driving **nTRST** or **nSRST** LOW.

Table 3-3 describes the external reset signals.

Table 3-3 Reset signal descriptions

Signal	Description	Type	Function
nPB	User pushbutton	Local	For Integrator/CP systems, a reset signal is generated by the FPGA from the **nPB** signal from pushbutton switch S2.
nSRST	System reset	Bidirectional	The **nSRST** open collector output signal is driven LOW by the module FPGA when the signal **PBRST** or software reset (**SWRST**) is asserted. As an input, **nSRST** can be driven LOW by the JTAG interface box.
nSYSRST	System reset	Bidirectional	**nSYSRST** can be driven by an FPGA on any attached board. Typically this will be driven by the master system reset controller such as a motherboard. The **nSYSRST** signal is used by Core Modules.
nSYSPOR	System reset	Output	The **nSYSPOR** signal is generated by an RC delay on **GLOBAL_DONE**. The reset is held active LOW until **GLOBAL_DONE** goes HIGH. **nSYSPOR** is used by external modules or tiles as a general reset signal.
nPORESET	System reset	Local	The **nPORESET** signal goes HIGH 7μS after 3.3V is supplied by the supply voltage. The reset drives the **nTRST** signal LOW.
CFGnPROG	Image reload	Local	The **CFGnPROG** signal goes LOW to initiate the loading of the FPGA image from the configuration flash. The signal can be triggered by the RECONFIG pushbutton S5 or by an external reset.

Table 3-3 Reset signal descriptions (continued)

Signal	Description	Type	Function
GLOBAL_DONE	System FPGAs configured	Bidirectional	This signal is driven low until **CFG_DONE** is HIGH. Other boards in the system pull this signal LOW if the local configuration sequence has not finished. There is a delay between **CFG_DONE** and **GLOBAL_DONE** to allow serial communication and configuration of the PLD.
CFG_DONE	System FPGAs configured	Local	The FPGA will drive this signal high following a successfully image configuration.
CFG_nWRITE	FPGA and PLD configuration	Local	**CFG_nWRITE** is an input to the FPGA that initiates sending the serial stream to the PLD. The PLD drives **CFG_nWRITE** HIGH 16 CCLK cycles (64 cycles of the 24MHz clock) after **CFG_DONE** goes HIGH. After the FPGA finishes sending the serial stream to the PLD, the PLD releases **GLOBAL_DONE**.

Note

See the documentation for the other board products for more information on reset.

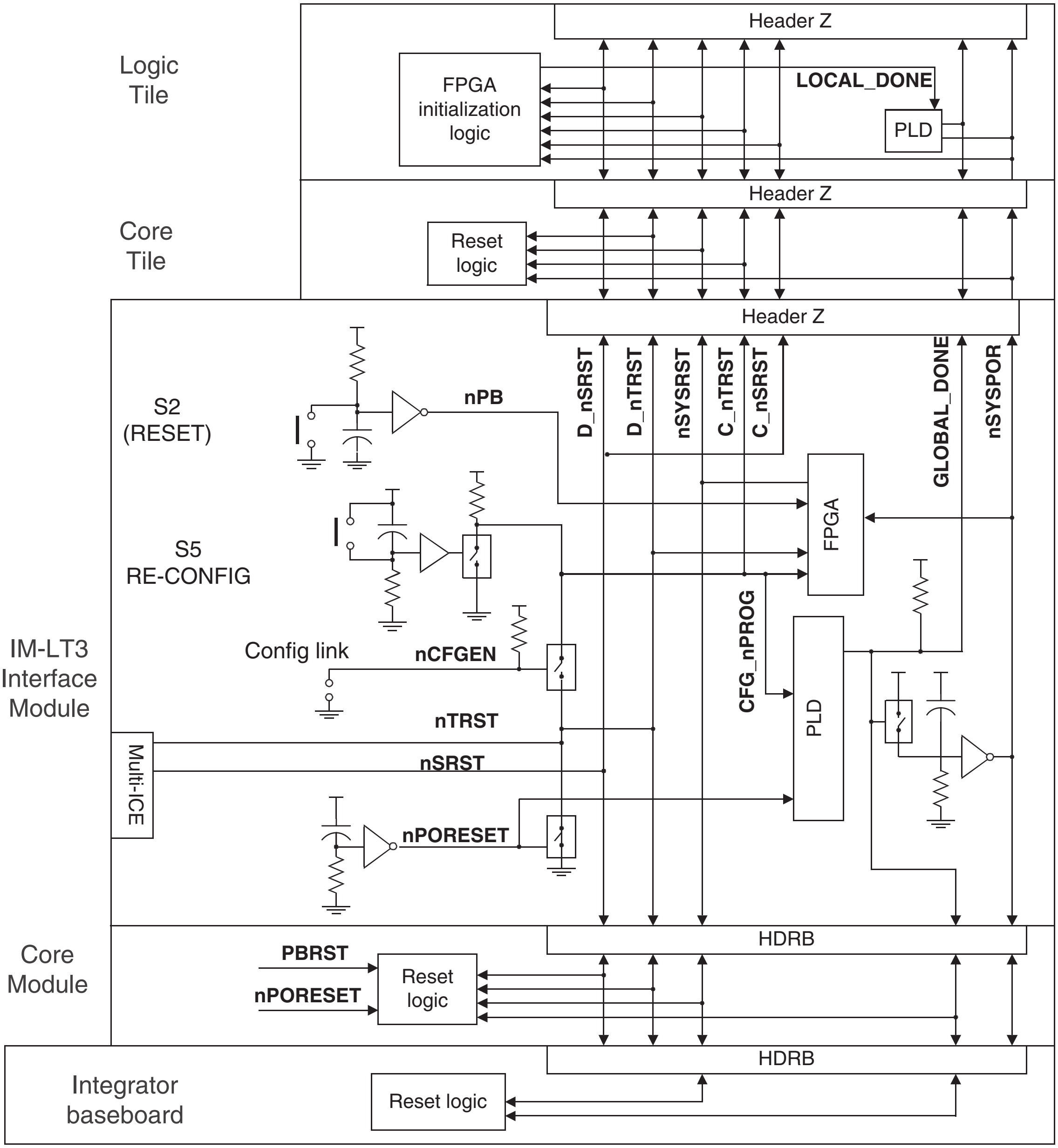


Figure 3-3 Reset control

The general purpose pushbutton S2 generates the **nPB** signal to the FPGA. For the Integrator/CP FPGA image, this signal triggers a reset of the board stack.

Figure 3-4 shows the reset timing.

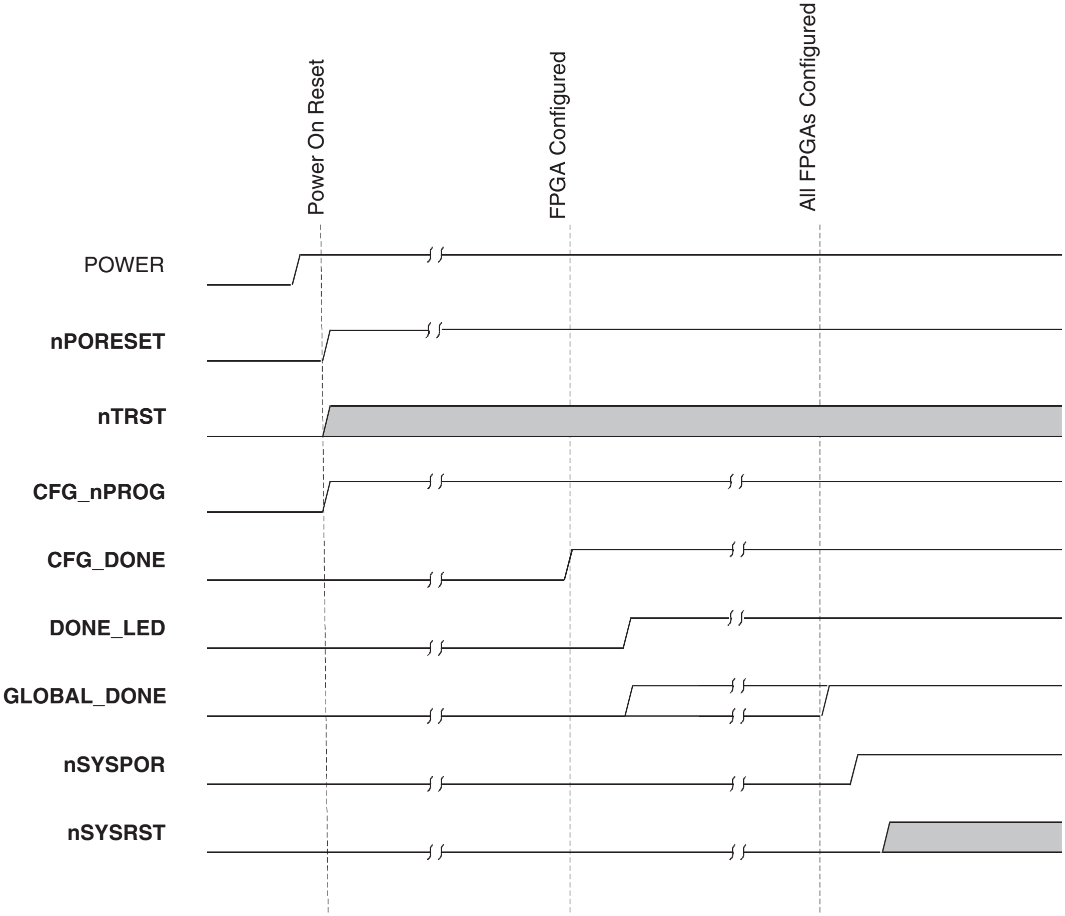


Figure 3-4 Reset sequence timing

3.5 Header connectors

The IM-LT3 provides both Integrator-style connectors (HDRA and HDRB) and tile connectors (HDRX, HDRY, and HDRZ).

3.5.1 Voltage on header pins

The voltages on the lower Integrator/CP-style connectors are the same as those on the Integrator/CM family of Core Modules. The logic signals on the lower header are standard Integrator/CM logic levels (3.3V).

The voltages on the upper connectors depends on the connector and on configuration settings:

- 5V supply is present on HDRX and HDRY. This voltage is used to generate the other voltages required for tiles.
- **VDDIO** from an attached tile is present on HDRX and HDRY.
- 3.3Vsupply is present on HDRZ.

Figure 3-5 shows the voltages present on the HDRX, HDRY, and HRDZ tile connectors.

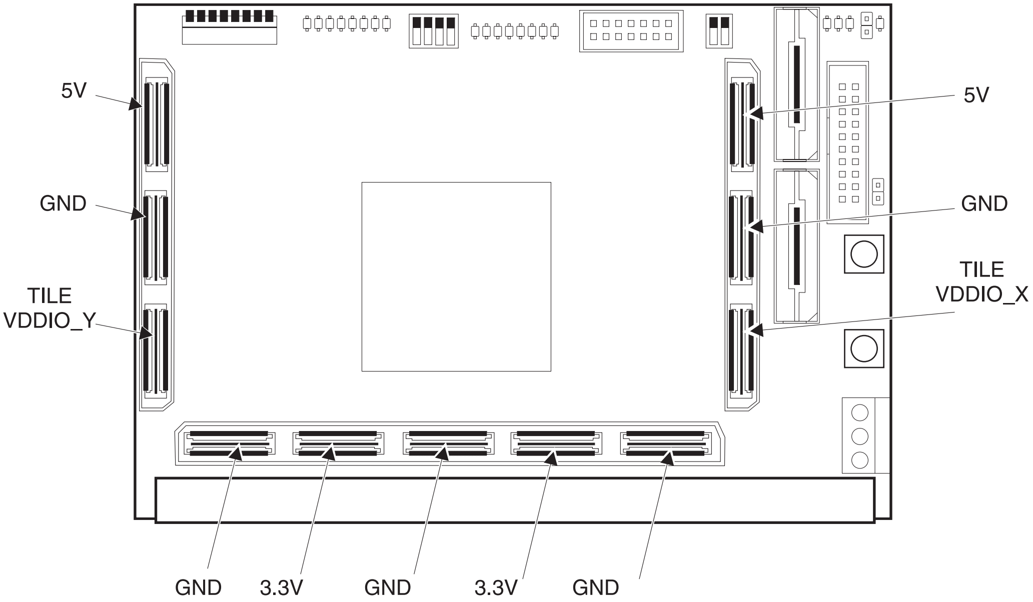


Figure 3-5 Voltages present on tile connector blades

Relationship between voltage supply and signal levels

Because of the configurable tile voltage and the types of logic used on the expansion tiles, there are different signal levels present on the connectors:

- All HDRA and HDRB signal pins are at 3.3V logic levels.
- HDRY odd pins 45 to 107 (the F bus) always use 3.3V logic levels if HDRB is connected to an Integrator system.

  All other HDRY signals are connected to the FPGA and are set by the **VDDIO_Y** level from the tile above. **VDDIO_Y** is connected to 3.3V by a MOSFET transistor if no tile is present. **VDDIO_Y** can be manually connected to 3.3V by fitting resistor links R112 and R113 on the PCB, but normally it is supplied only by the tile above.
- HDRX signals are connected to the FPGA and are set by the **VDDIO_X** level from the tile above. **VDDIO_X** is connected to 3.3V by a MOSFET transistor if no tile is present. **VDDIO_X** can be manually connected to 3.3V by fitting resistor links R133 and R134 on the PCB, but normally it is supplied only by the tile above.

  ——— **Note** ———

  If the SSRAM on the IM-LT3 is used, some of the HDRX signals must match the signal level used for the SSRAM.

  ———
- All HDRZ signals operate at a 3.3V logic level.

3.5.2 Isolation of HDRZ signals

The IM-LT3 FPGA loads the configuration image into the PLD that outputs the break switches to the switches.

The break switches can also be directly controlled by the FPGA on an attached Logic Tile.

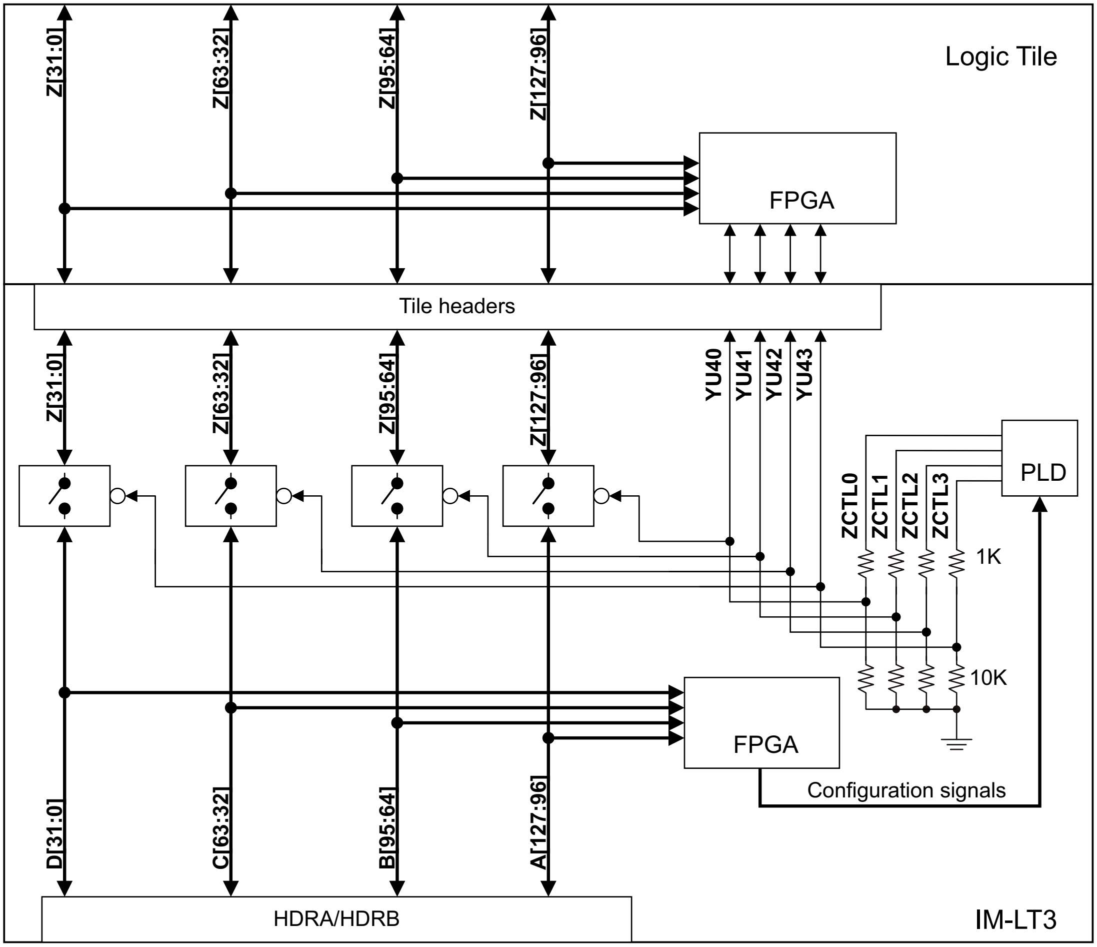


Figure 3-6 Isolation switches

3.6 Clock architecture

The IM-LT3 generates four clocks and has three connections to the **SYSCLK** signals from an attached Integrator/CP baseboard (if used) and up to seven clocks from attached tiles.

The local programmable clocks are supplied by three clock generator chips. Their frequencies are selected in oscillator control registers within the FPGA. A 24MHz reference clock is supplied to the clock generators, FPGA, and PLD.

For information on Logic Tile clocking schemes, see the *Integrator/XC2V4000+ Logic Tile User Guide*. For information on Core Module or baseboard clocking, see the documentation supplied with the product. Additional information on clocking systems might be present in the application note for the specific combination of boards.

Figure 3-7 on page 3-16 and Figure 3-7 on page 3-16 show the architecture of the clock system.

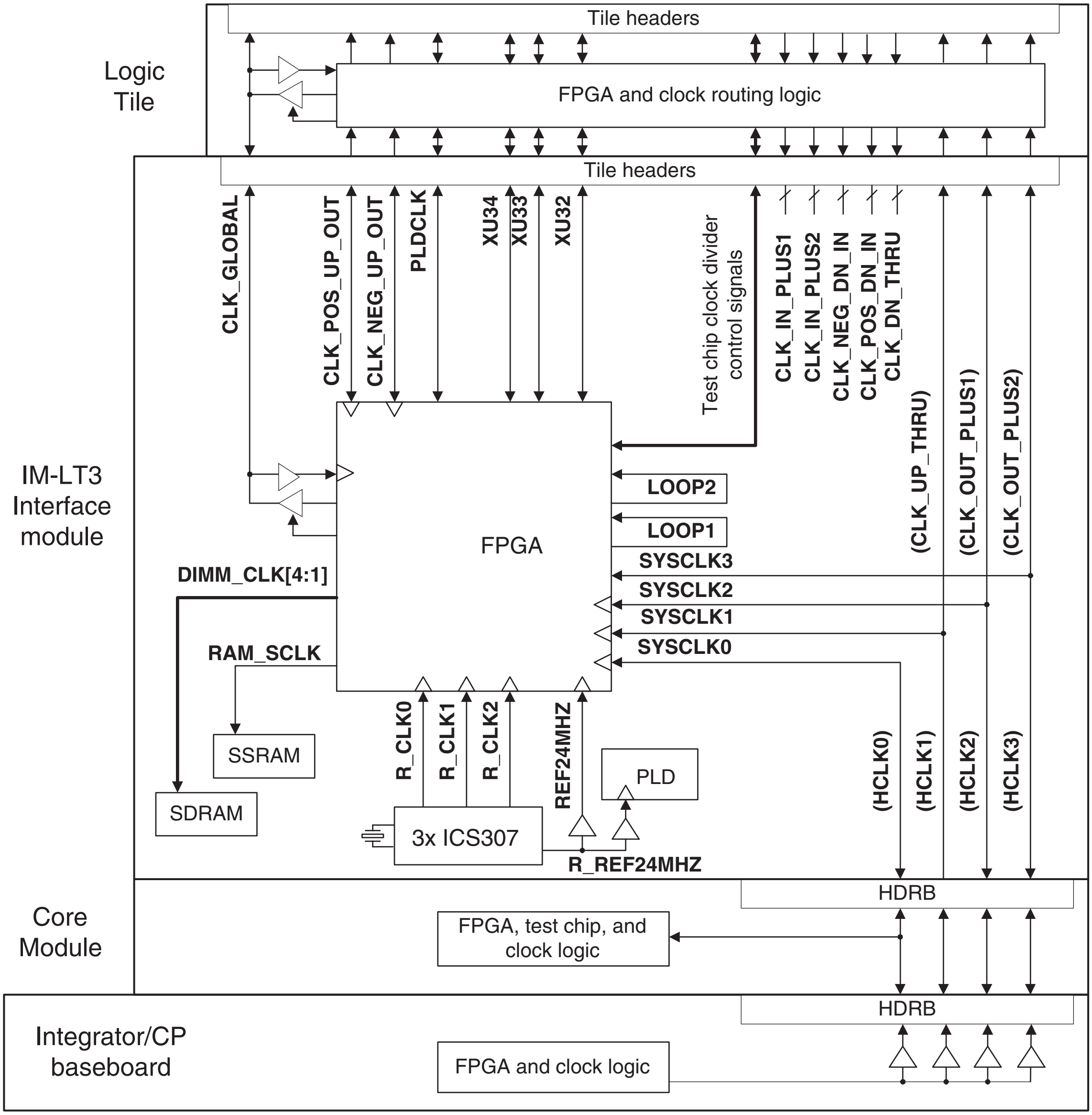


Figure 3-7 Clock signal routing (Core Module)

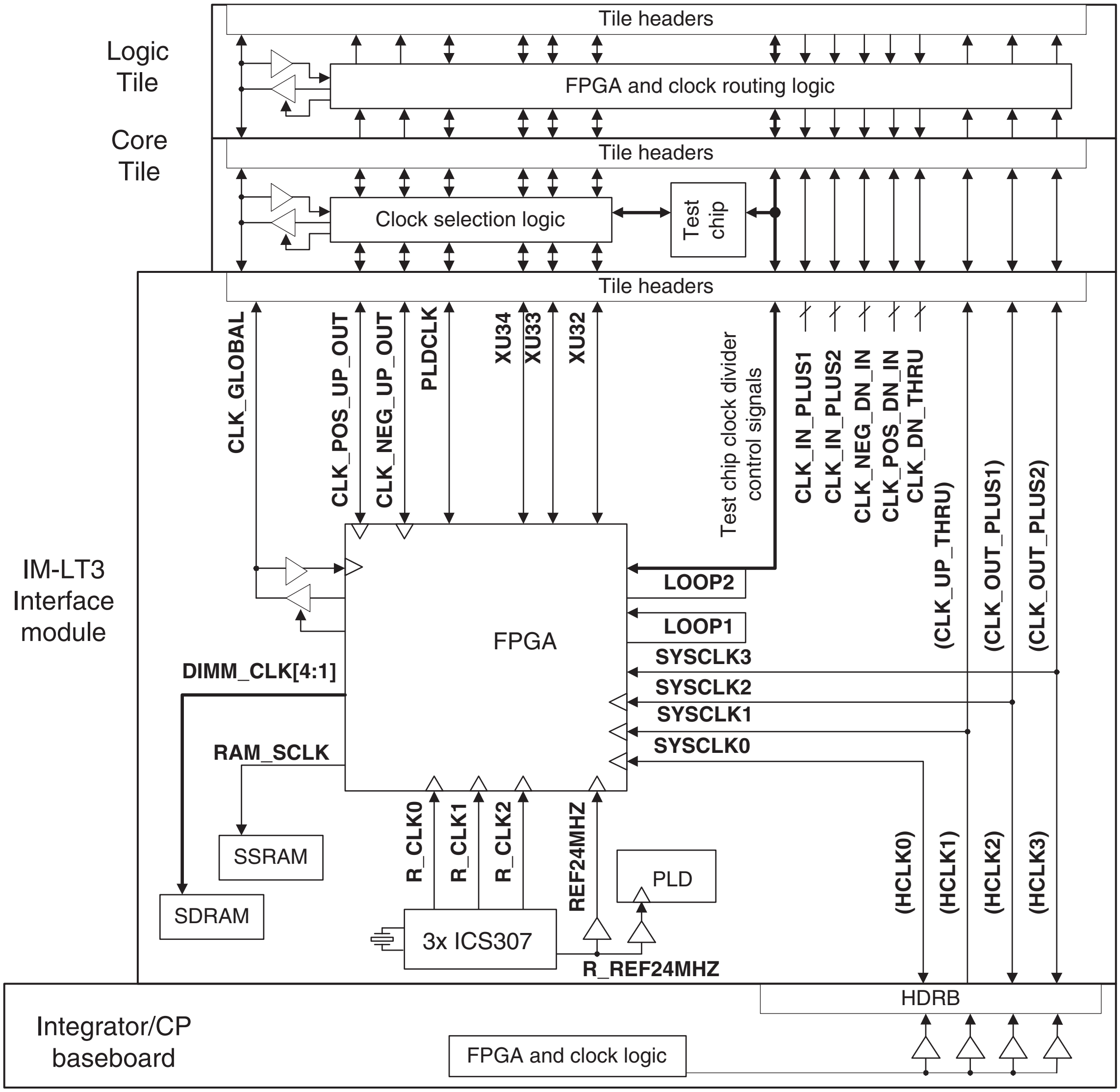


Figure 3-8 Clock signal routing (Core Tile)

3.6.1 Clock routing to and from the IM-LT3

The IM-LT3 clock signals are listed in Table 3-4. (For a description of JTAG clocks, see *JTAG support* on page 3-25.

Table 3-4 IM-LT3 clocks

Signal	Source	Description
CLK_GLOBAL	Tile header HDRZ	A clock line common to the IM-LT3 and any tiles stacked on it. The FPGAs on all boards receive the clock as **CKL_GLOBAL_IN**. The FPGA on each Logic Tile outputs a **CLK_GLOBAL_OUT** signal to a tristate buffer. The signal **CLK_GLOBAL_nEN** enables the buffer and the local signal **CLK_GLOBAL_OUT** becomes the global clock for the system. **Note** The buffers are placed close to the HDRZ connectors, but there will be some skew between tiles. To use in-phase clock signals, use the **CLK_POS_UP_OUT** and **CLK_NEG_UP_OUT** clocks and the DLLs in the FPGAs to resynchronize the clock signals between tiles.
CLK_NEG_UP_OUT, **CLK_POS_UP_OUT**	IM-LT3 FPGA	Clock signal from FPGA to the tile stacked on top of the IM-LT3. These clocks can be configured as two independent clocks or one differential clock. These signals are typically outputs from the IM-LT3, but they can be configured as inputs and be driven by an attached Logic Tile.
SYSCLK0	IM-LT3 FPGA or Integrator baseboard	This clock signal from the Integrator baseboard connects to the FPGA on the IM-LT3, but is not connected to any tiles in the stack. If the IM-LT3 is used with a Core Module but without a baseboard, this clock can be configured as an output from the FPGA to the Core Module.
SYSCLK[3:1]	IM-LT3 FPGA, tile header HDRZ, or Integrator baseboard	These clock signals are connected to the FPGA, the Integrator/CP motherboard, and the header connectors. **SYSCLK[2:1]** pass up to the first tile in the stack and **SYSCLK[3]** passes to the second tile in the stack. The IM-LT3 FPGA can accept or generate these clocks. These clocks are sometimes named **HCLK[3:0]** on Integrator Core Modules and baseboards.
R_CLK[2:0]	ICS307s	Locally generated programmable frequency clocks for the FPGA.
R_REF24MHZ	Crystal oscillator in ICS307	Buffered versions of this reference signal are available as **PLD_REF24MHZ** and **REF24MHZ**. The 24MHz oscillator is the reference for all ICS307 devices.

Table 3-4 IM-LT3 clocks (continued)

Signal	Source	Description
DIMM_HCLK[4:1]	FPGA	These are the FPGA clocks to the SDRAM memory module.
RAM_SCLK	FPGA	This is the FPGA clock to the SSRAM memory.
LOOP[2:1]	FPGA	These clock loops enable FPGA to FPGA delay matching. See the *Versatile/LT-XC2V4000+ User Guide* for more information on clock matching.
XU[32]	FPGA or Core Tile	This clock might be used as the **X_HCLK_DN** output from the clock logic on the Core Tile.
XU[33]	FPGA or Core Tile	This clock might be used as the **X_REFCLK_UP** input to the clock logic on the Core Tile.
XU[34]	FPGA or Core Tile	This clock might be used as the **X_HCLKIN_UP** input to the clock logic on the Core Tile.
CLK_IN_PLUS[2:1], **CLK_NEG_DN_IN**, **CLK_POS_DN_IN**, and **CLK_DN_THRU**	Logic Tile	These Logic Tile clocks are not connected on the IM-LT3.

3.6.2 ICS307 clock generators

The ICS307s references are supplied by a 24MHz crystal. The frequency of the outputs from the ICS307s are controlled by values loaded into the serial data pins. This enables them to produce a wide range of frequencies.

The IM-LT3 produces the fixed-frequency 24MHz clocks **REF24MHZ** (clock input to the FPGA) and **PLD_REF24MHZ** (clock input to PLD).

For more information on the ICS307, see the manufacturer's data sheet.

Programmable clocks on the IM-LT3

The IM-LT3 provides the programmable clock sources **R_CLK[2:0]**.

These clocks are supplied by three MicroClock ICS307 clock generators, as shown in Figure 3-9.

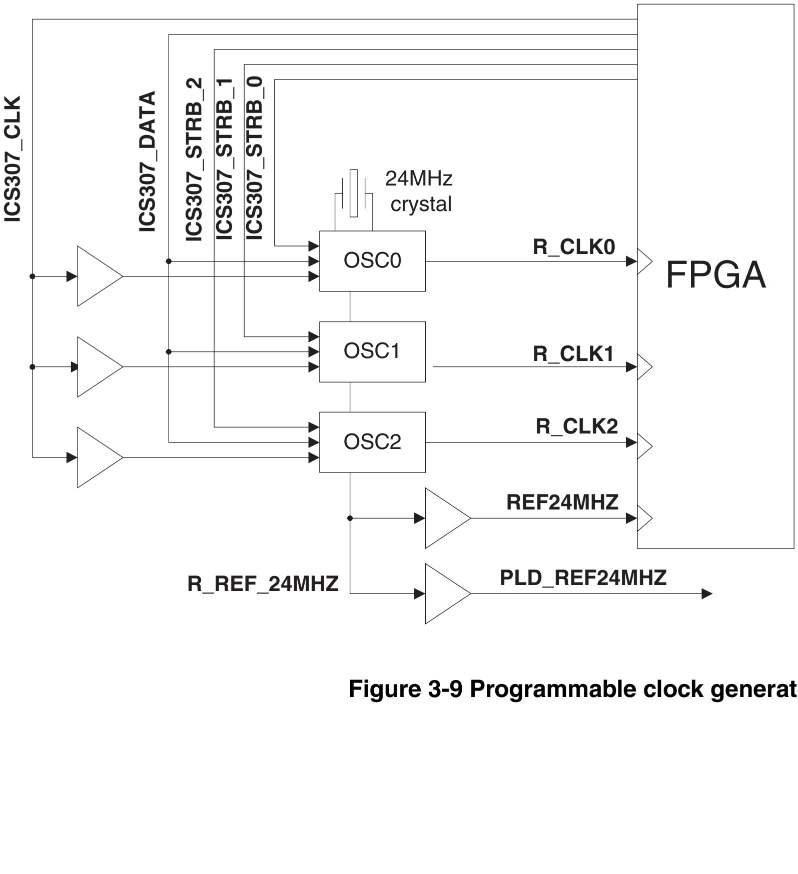


Figure 3-9 Programmable clock generators

Programming the ICS307 clocks

The frequency of clocks from an ICS307 is set by loading values for the divider and multiplier registers into the serial input port on the clock generator. These control the value of three parameters used to determine the output of the ICS307. Calculate the frequency using the formula:

$R\_CLK_{freq} = 48 * ((VDW+ 8) / ((RDW+2)*OD))$

where:

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

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

OD Is the output divider. The divider code for each bit combination is shown in Table 3-5.

Table 3-5 Output Divider code

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

The configuration data stream is shown in Figure 3-10 on page 3-22 where:

C[1:0] Internal load capacitance for crystal. If an external clock is used, set C[1:0] to 10.

T Duty cycle threshold setting:

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

F[1:0] Function of CLK2 output:

- 00 selects reference signal
- 01 selects reference signal divided by two

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

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

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

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

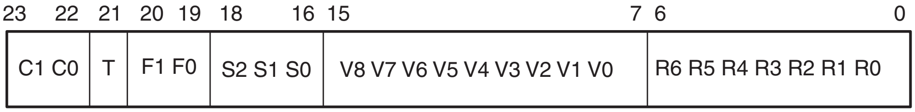


Figure 3-10 VCO configuration data

Note

Bit 23 is loaded into the shift register first and bit 0 is loaded last. Data is clocked into the register on the rising edge of **SCLK**. The **STROBE** signal is pulsed high after all bits have been shifted into the register.

See the application notes for the IM-LT3 for details of implementing the clock interface logic in the FPGA.

3.7 Memory interfaces

The IM-LT3 module has 2MB of ZBT SSRAM onboard and a socket for a single 16 to 256MB SDRAM DIMM.

Note

The description in this section refers to the controllers supplied with the example image covered by the IM-LT3 application notes. If you require a different memory controller, you must provide the HDL source. See *SDRAM interface*, and *ZBT SSRAM interface* on page 3-24 for details of the SRAM and SDRAM pin connections.

3.7.1 SDRAM interface

A socket for a single SDRAM DIMM is provided. The signals from the socket go to the FPGA and an SDRAM controller can be implemented in the FPGA.

Serial presence detect

JEDEC-compliant SDRAM DIMMs incorporate a *Serial Presence Detect* (SPD) feature. This comprises a 2048-bit serial EEPROM located on the DIMM with the first 128 bytes programmed by the DIMM manufacturer to identify the following:

- tile type
- memory organization
- timing parameters.

The EEPROM clock (SCL) operates at 93.75kHz (24MHz divided by 256). The transfer rate for read accesses to the EEPROM is 100kbit/s maximum. The data is read out serially 8 bits at a time, preceded by a start bit and followed by a stop bit. This makes reading the EEPROM a very slow process and it takes approximately 27ms to read all 256 bytes.

Write accesses to the SPD EEPROM are not supported.

SDRAM signals

The distribution of clock and control signals to the DIMM socket is shown in Figure 3-11 on page 3-24.

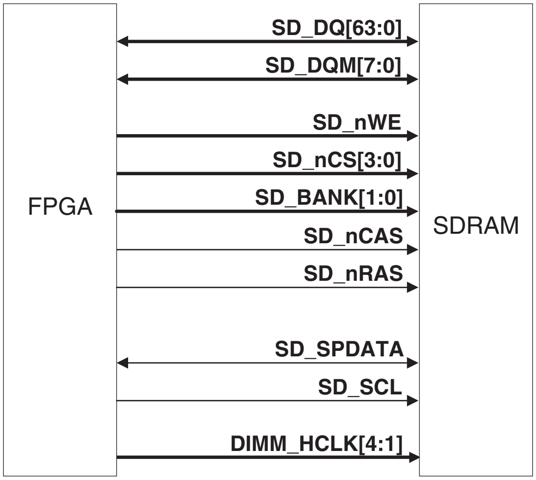


Figure 3-11 SDRAM clock and control signals

3.7.2 ZBT SSRAM interface

The ZBT SSRAM controller is implemented in the IM-LT3 FPGA. The distribution of clock, address, and data signals to the SSRAM is shown in Figure 3-12.

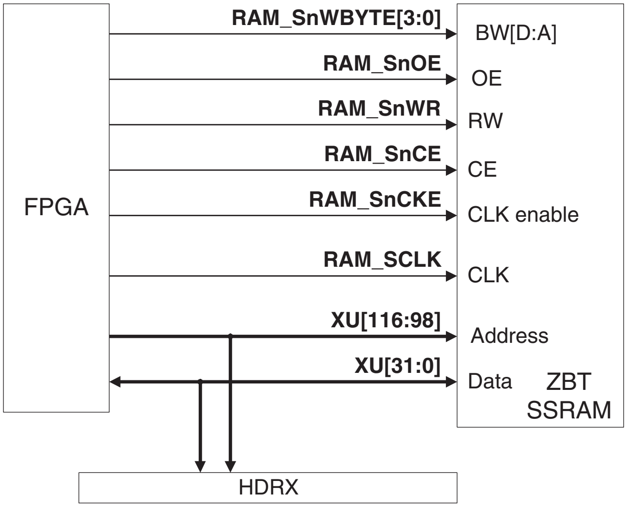


Figure 3-12 SSRAM clock and control signals

3.8 JTAG support

The IM-LT3 supports debugging and configuration using JTAG. This is described in the following subsections:

- *JTAG connection*
- *JTAG connection modes* on page 3-26
- *JTAG signals* on page 3-27
- *JTAG scan paths* on page 3-29.

3.8.1 JTAG connection

Figure 3-13 shows the JTAG connector, CFGEN link and LED.

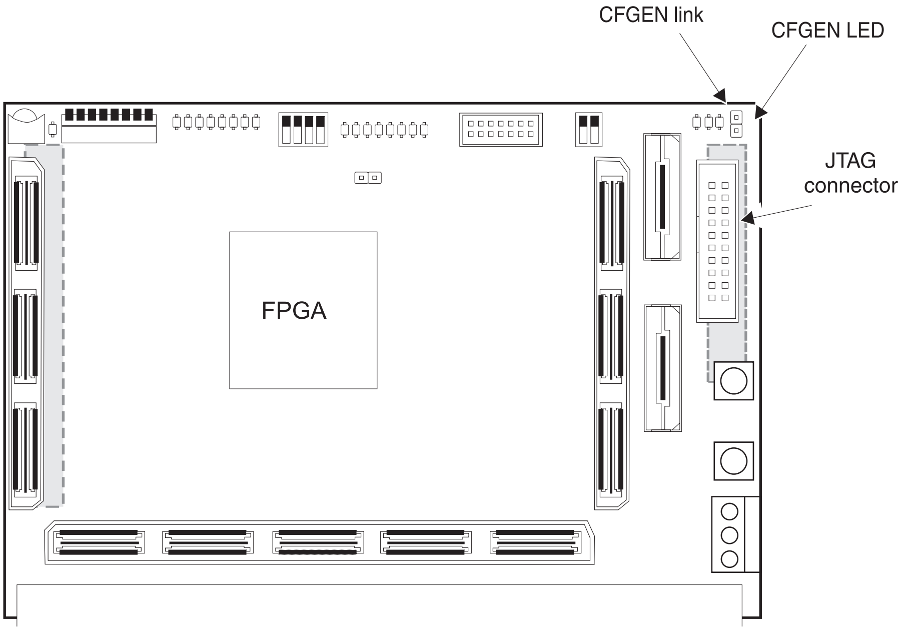


Figure 3-13 JTAG connector, CFGEN link, and LED

The CONFIG link is used to enable in-circuit programming of the FPGA and PLD using Multi-ICE and the `progcards` utility (see *JTAG connection modes* on page 3-26).

The JTAG connector provides a set of JTAG signals that allow JTAG debugging equipment to be used (see *JTAG signals* on page 3-27). If you are debugging a development system with multiple tiles, connect the JTAG debugging equipment to the IM-LT3 and the JTAG signals are routed through any connected tiles.

3.8.2 JTAG connection modes

The system is capable of operating in user mode or configuration mode.

Debug mode

Debug mode (sometimes also called user mode or normal mode) is selected by default (when a jumper is *not* fitted on the CONFIG link, see Figure 3-13 on page 3-25). In this mode, the processor cores are accessible on the scan chain, as shown in Figure 3-15 on page 3-30.

Configuration mode

In configuration mode, the configuration TAP controllers of all FPGAs and PLDs in the system are connected into the scan chain. This allows the board to be configured or upgraded using Multi-ICE or other JTAG debugging equipment.

To select configuration mode, fit a jumper to the CONFIG link on the IM-LT3 (see Figure 3-13 on page 3-25). This has the effect of pulling the **nCFGEN** signal LOW and illuminating the CFGEN LED on the IM-LT3 and rerouting the JTAG scan path. The LED provides an indication that the development system is in the configuration mode.

After upgrading the flash FPGA images:

1. Remove the CONFIG link.
2. Power cycle the development system.

The configuration mode allows FPGA and PLD code to be updated as follows:

- The FPGAs are volatile, but load their configuration from flash memory. Flash memory does not have a JTAG port, but it can be programmed by loading designs into the FPGAs and PLDs that handle the transfer of data to the flash using JTAG.
- The PLDs are nonvolatile devices that can be programmed directly by JTAG.

3.8.3 JTAG signals

Figure 3-14 shows the pinout of the JTAG connector and Table 3-6 provides a description of the JTAG and related signals.

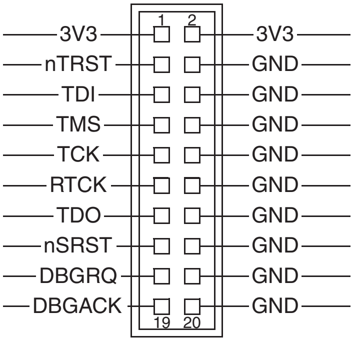


Figure 3-14 JTAG connector pinout

Note

In the description in Table 3-6, the term JTAG equipment refers to any hardware that can drive the JTAG signals to devices in the scan chain.Typically this will be Multi-ICE or RealView ICE, although hardware from other suppliers can also be used to debug ARM processors.

Table 3-6 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. It is provided for compatibility with third-party JTAG equipment.
DBGACK	Debug acknowledge (to JTAG equipment)	**DBGACK** indicates to the debugger that the processor core has entered debug mode. It is provided for compatibility with third-party JTAG equipment.
GLOBAL_DONE	All FPGAs configured	**GLOBAL_DONE** is an open-collector signal that indicates when all FPGA configurations are complete. Although this signal is not a JTAG signal, it does affect **nSRST**. The **GLOBAL_DONE** signal is routed between all boards in the system through the HDRB and HDRZ connectors. The master reset controller on the baseboard senses this signal and holds all the boards in reset (by driving **nSRST** LOW) until all FPGAs are configured.

Table 3-6 JTAG signal description (continued)

Name	Description	Function
nCFGEN	Configuration enable (from jumper on the IM-LT3)	**nCFGEN** is an active LOW signal used to put the boards into configuration mode. The **nCFGEN** signal is routed between all FPGAs in the system through the HDRB connectors. 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 baseboard)	**nRTCKEN** is an active LOW signal driven by any Core Module or tile that requires **RTCK** to be routed back to the JTAG equipment. If **nRTCKEN** is HIGH, the baseboard drives **RTCK** LOW. If **nRTCKEN** is LOW, the baseboard drives the **TCK** signal back up the stack to the JTAG equipment.
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. When the signal is driven LOW by the reset controller on the tile, the baseboard resets the whole system by driving **nSYSRST** LOW. This is also used in configuration mode to control the initialization pin (**nINIT**) on the FPGAs. Though not a JTAG signal, **nSRST** is described because it can be controlled by JTAG equipment.
nTRST	Test reset (from JTAG equipment)	This active low open-collector is used to reset the JTAG port and the associated debug circuitry on the tile. It is asserted at power-up by each tile, 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	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. The **RTCK** signal is returned by the core to the JTAG equipment, and the clock is not advanced until the core has captured the data. The JTAG debugging device can be configured to wait for 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 next device in the chain. The **RTCK** signal on the tile connectors HDRB 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 baseboard.

Table 3-6 JTAG signal description (continued)

Name	Description	Function
TCK	Test clock (from JTAG equipment)	**TCK** synchronizes all JTAG transactions. **TCK** connects to all JTAG components in the scan chain. Buffers on each board maintain good signal integrity. **TCK** flows up the stack of tiles 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**).
TDI	Test data in (from JTAG equipment)	**TDI** goes down to the baseboard and then up the stack of tiles. The signal connects each component in the scan chain.
TDO	Test data out (to JTAG equipment)	**TDO** is the return path of the data input signal **TDI**. The tile connectors HDRB have two pins labelled **TDI** and **TDO**. **TDI** refers to data flowing down the stack and **TDO** to data flowing up the stack. 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.
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 as the signal flows through the tile stack.

3.8.4 JTAG scan paths

This section describes JTAG scan chain data, clock, and TMS paths.

The JTAG chain on the IM-LT3 is split into two chains that are connected to the tiles as:

Debug The debug JTAG signals are used with a processor core (implemented as an SMM in FPGAs or an external Core Module or Core Tile).

In this mode, the JTAG signals are not routed to the configuration inputs. The signals from the ILA (Chip Scope) connector, however, can provide some configuration control even while the JTAG signals are being used for debugging.

Config The configuration JTAG signals are used to load the configuration flash for the FPGA or to reprogram the PLD.

Data path

Figure 3-15 and Figure 3-16 on page 3-31 show the data path. The switches are in debug (user) mode.

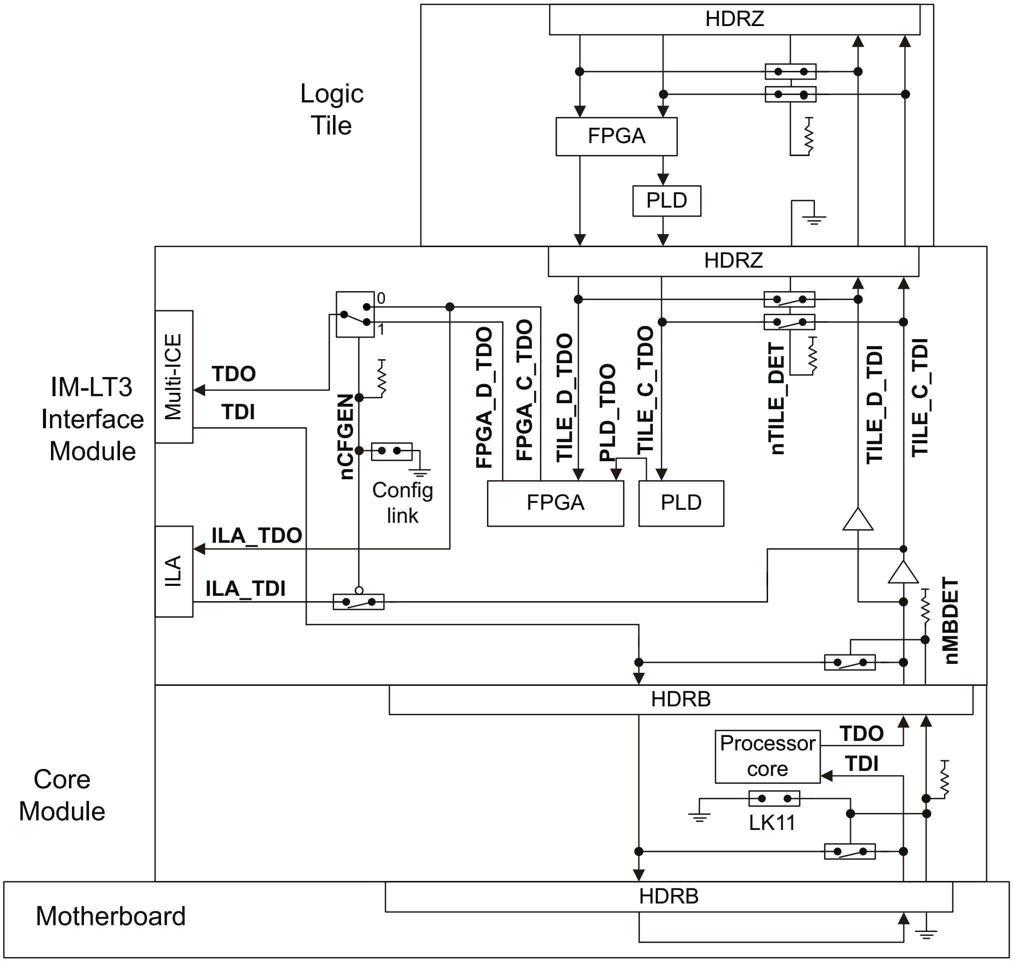


Figure 3-15 JTAG data path block diagram (Core Module)

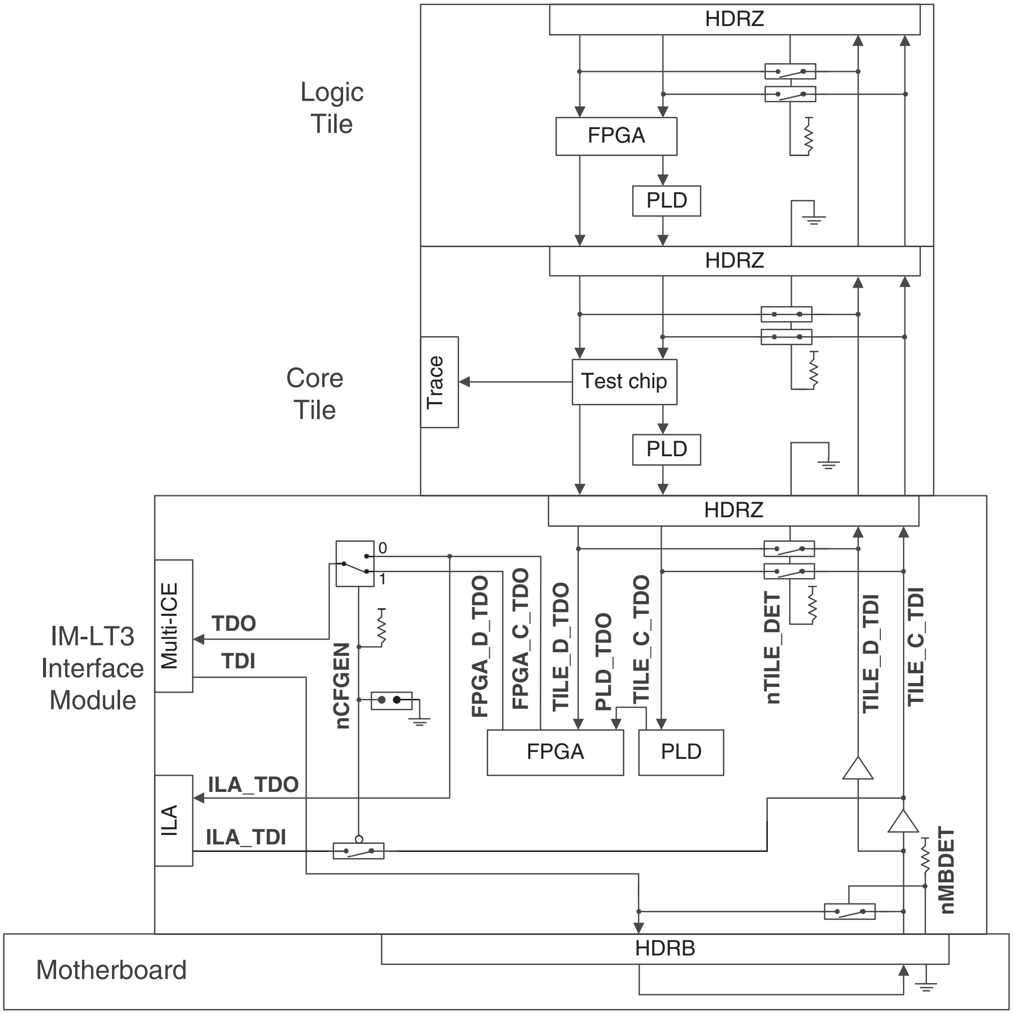


Figure 3-16 JTAG data path block diagram (Core Tile)

When you use the IM-LT3 and a tile as a standalone development system, the data path is routed to the tile and back to the JTAG connector.

Figure 3-17 on page 3-32 and Figure 3-18 on page 3-33 show a simplified diagram of the JTAG data path in debug mode.

If the IM-LT3 and tile is attached to an Integrator baseboard, the **TDI** signal from the JTAG connector is routed down through the HDRB connectors to the baseboard. From there the path is routed back up the stack through each tile, before being returned to the JTAG connector as **TDO**.

The motherboard detect signal **nMBDET** controls a switching circuit on the IM-LT3 and tile and, therefore, the routing of **TDI**.

In configuration mode, the PLDs and FPGAs are included in the scan chain as described in *JTAG connection modes* on page 3-26. In debug mode however, the JTAG signals are simply routed through the FPGA and the FPGA is not visible in the scan path. (The FPGA will be visible, however, if the re is a user-implemented TAP controller in the FPGA).

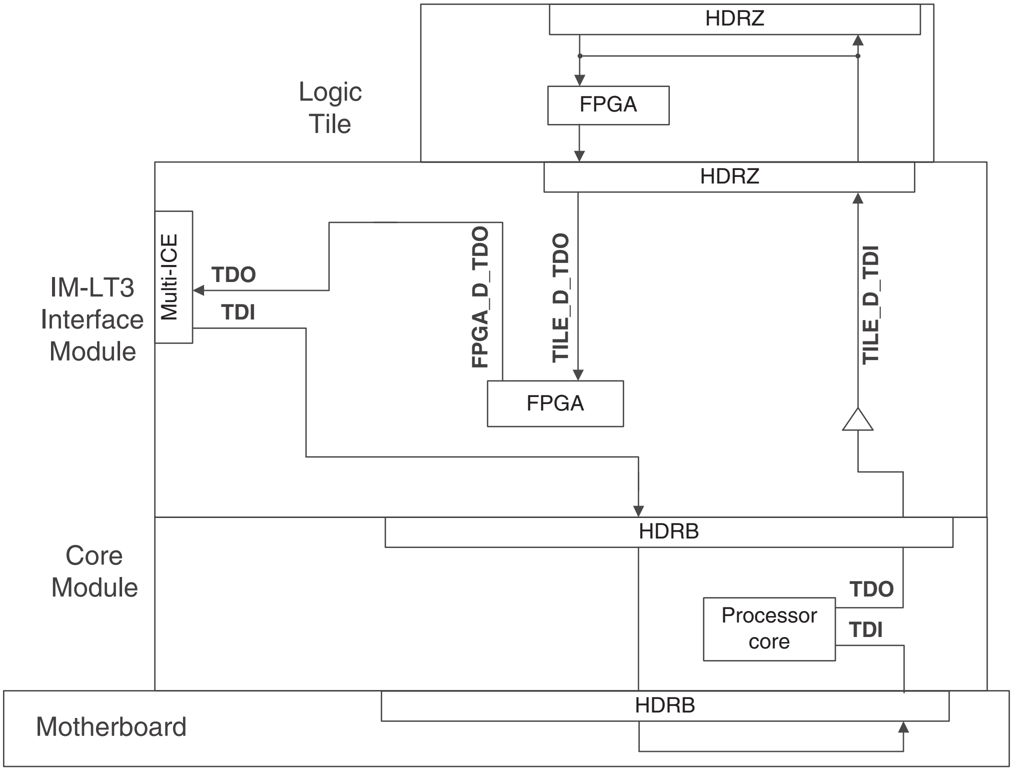


Figure 3-17 Simplified view of data path in debug mode (Core Module)

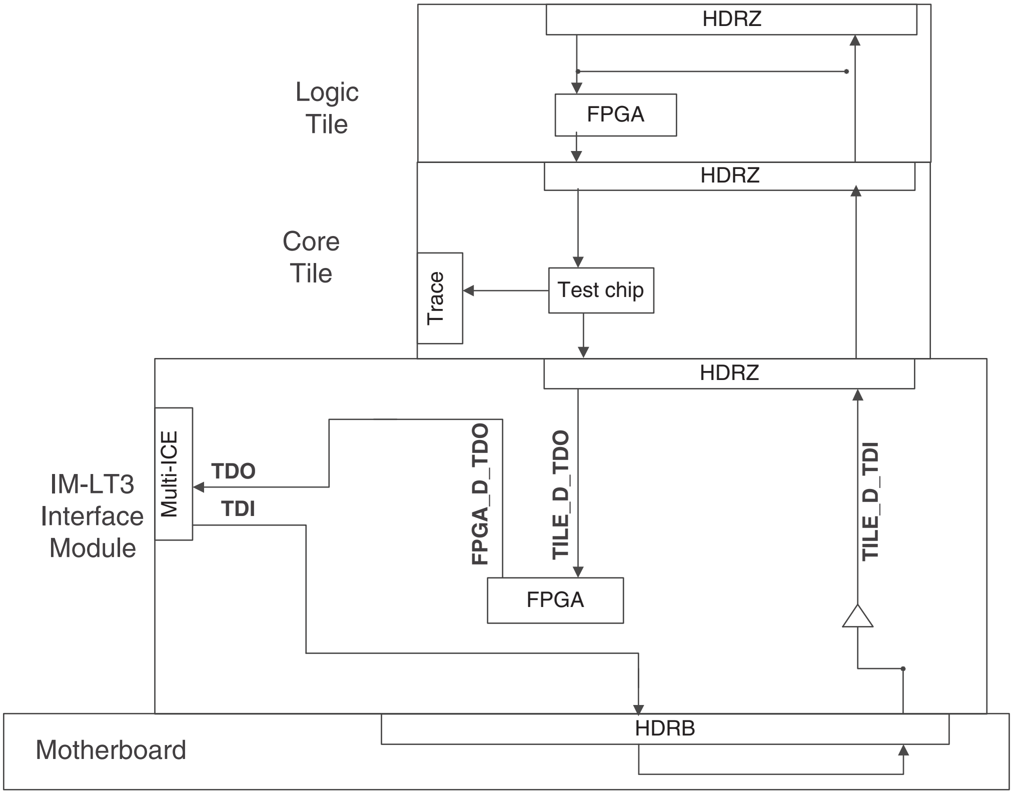


Figure 3-18 Simplified view of data path in debug mode (Core Tile)

Clock path

The clock path is routed in a similar way to the data path, although in the opposite direction. Figure 3-19 on page 3-34 and Figure 3-20 on page 3-35 show a simplified diagram of the clock path. The position for the **TILE_RTC** switch is for user mode (the CONFIG jumper is not fitted).

A number of synthesized cores sample **TCK**. This introduces a delay into the clock path. Cores of this type pass on the delayed clock signal as **RTCK**, which is fed to the **TCK** input of the next device in the chain. The **RTCK** signal at the JTAG connector is regulates the advance of **TCK** (see the *ARM Multi-ICE User Guide*).

The routing of the **TCK/RTCK** signals through the stack is controlled by switches in a similar way to the data path. The routing of **RTCK** is controlled by the signal **nRTCKEN** and an AND gate on the motherboard. See the *Integrator/XC2V4000+*

User Guide for details on control of **nRTCKEN**. If a Core Module is used without a motherboard fitted beneath it, fit the stacking option link on the Core Module to cause the **RTCK** signal to be routed back to the upper header connector (see the schematic for the Core Module for the link settings).

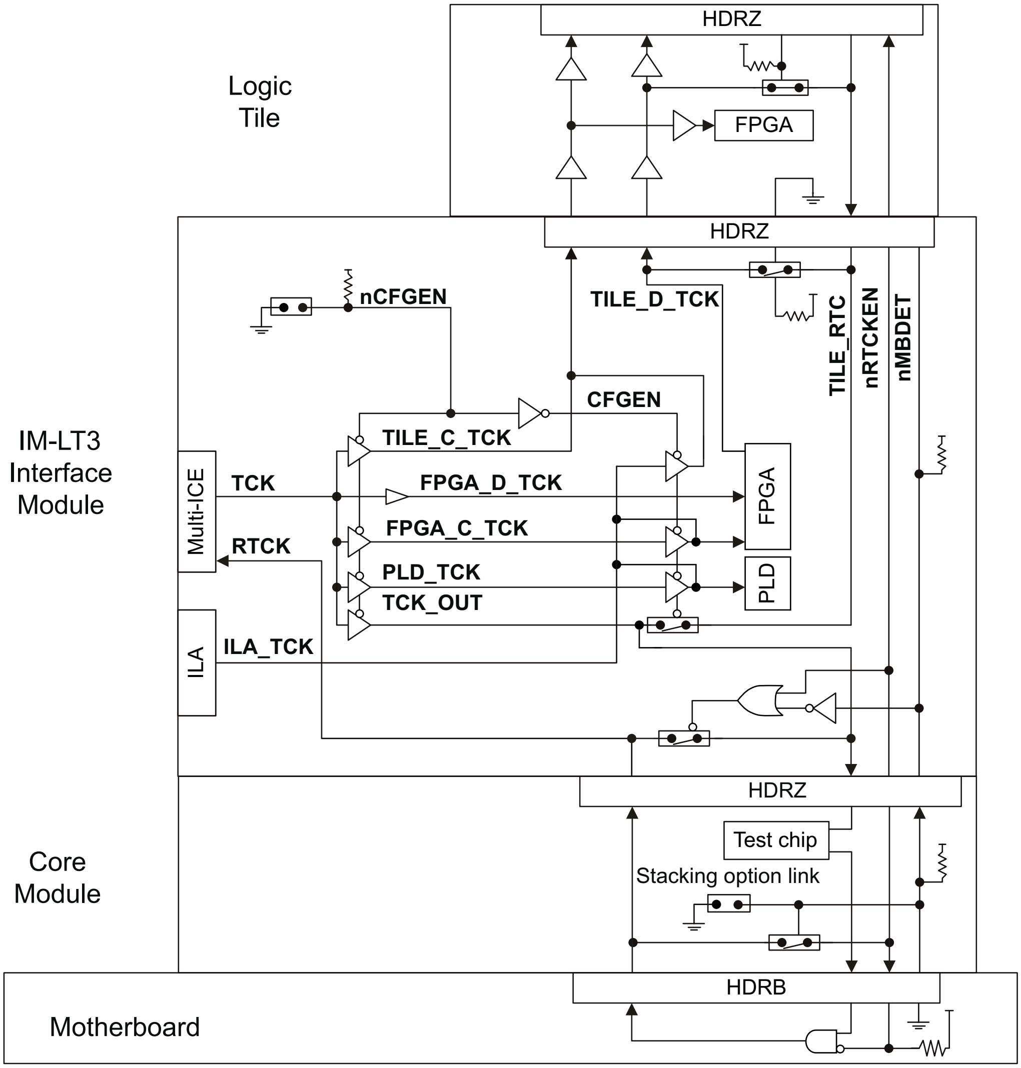


Figure 3-19 JTAG clock path (Core Module)

When the IM-LT3 is being used to support an ARM core in an adjacent Core Tile, the PLD can be set to assert the **nRTCKEN** signal. This is done via the PLD serial configuration stream that is described in *PLD circuitry* on page 3-5.

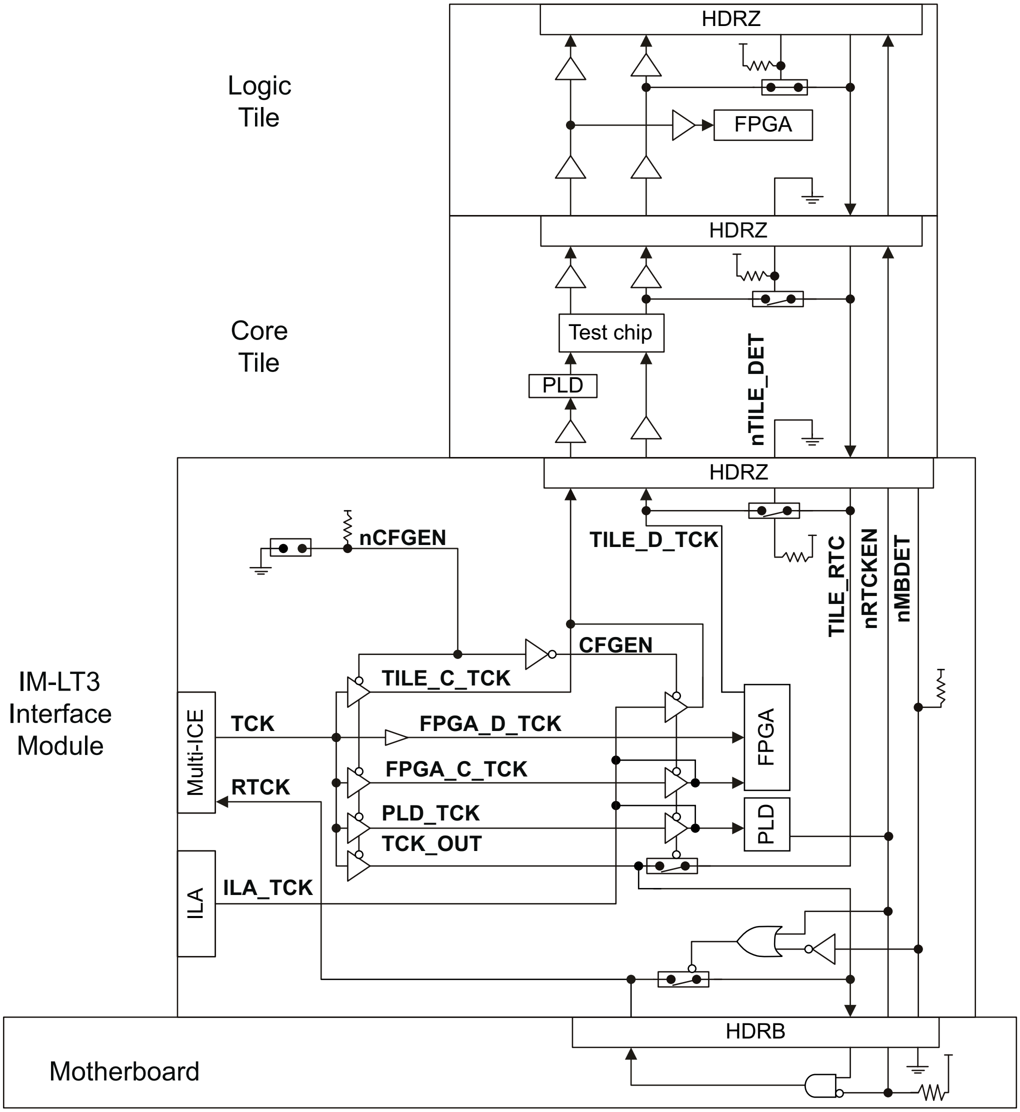


Figure 3-20 JTAG clock path (Core Tile)

TMS path

Figure 3-21 and Figure 3-22 on page 3-37 show the TMS path. The switch positions are for user mode (the CONFIG jumper is not fitted).

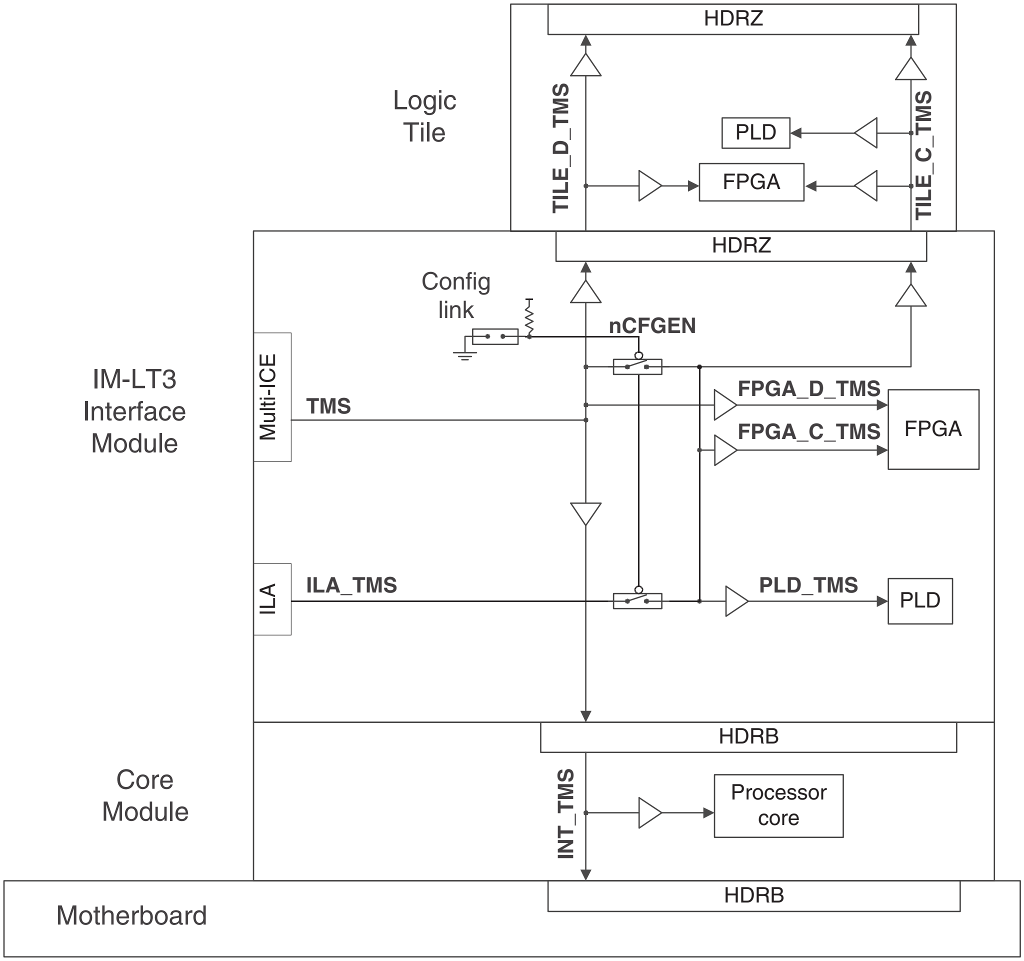


Figure 3-21 TMS path (Core Module)

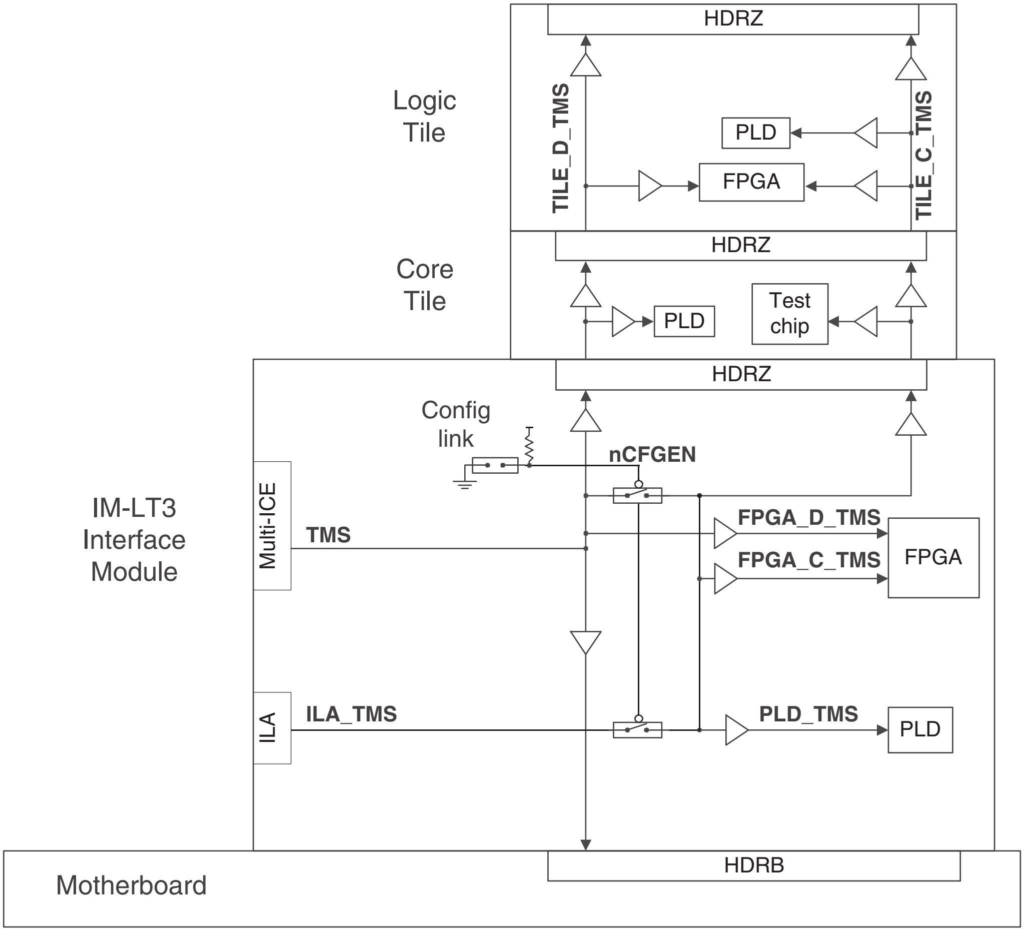


Figure 3-22 TMS path (Core Tile)

3.8.5 Integrated logic analyzer

The Inegrated logic analyzer connector J12 enables you to connect a compatible analyzer (such as ChipScope) to access the FPGA JTAG signals when the system is in debug mode (CONFIG link not fitted and a JTAG debugger is being used to examine code on a CPU).

If the board is in debug mode, the configuration scan chain is not normally accessible. The integrated logic analyzer connector, however, provides access to the IM-LT3 configuration scan chain and enables debugging of the IM-LT3 FPGA design and the software running on a Core Tile or Core Module simultaneously. For more details on the integrated logic analyzer, see the ChipScope details on the Xilinx website (www.xilinx.com).

3.9 Switches and LEDs

Figure 3-23 shows a block diagram of the switch and LED circuit.

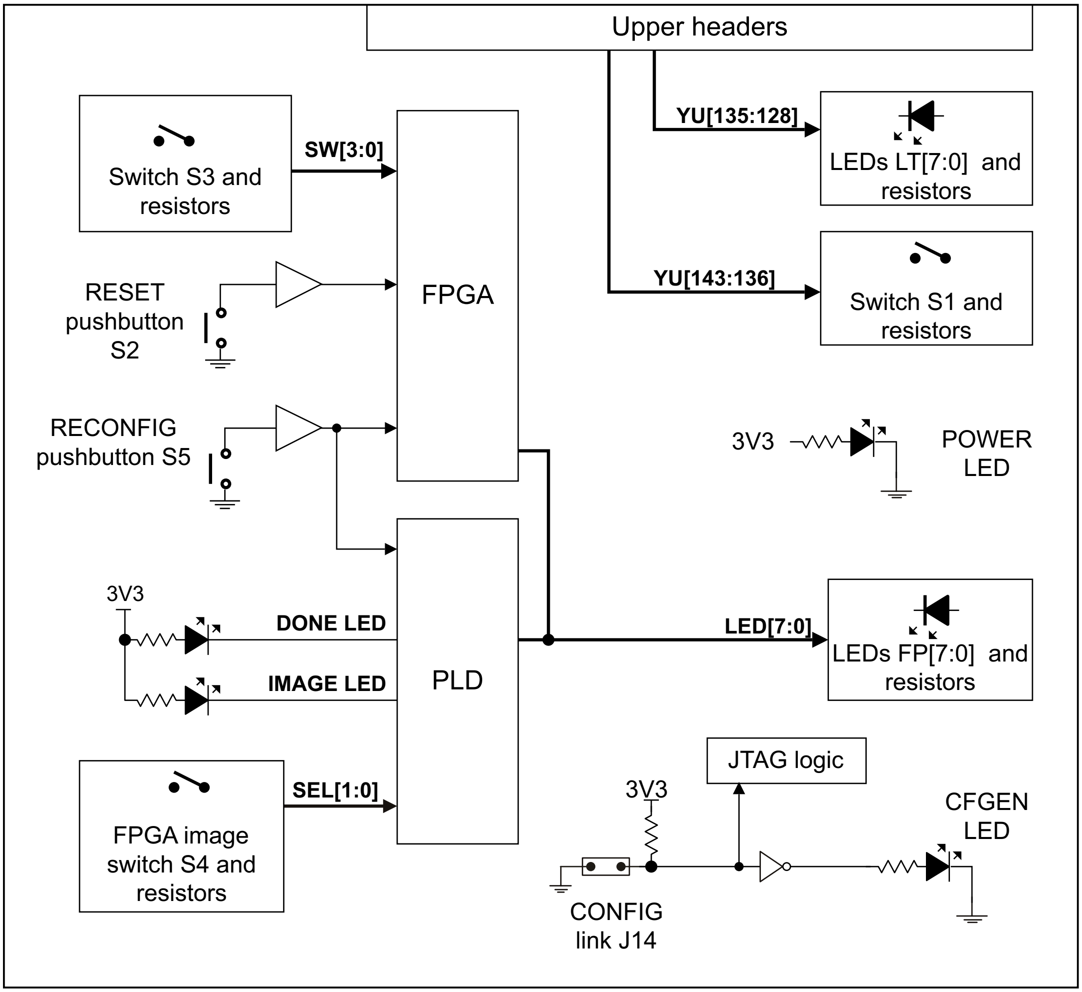


Figure 3-23 Switch and LED block diagram

Figure 3-24 on page 3-40 shows the switch and LED locations.

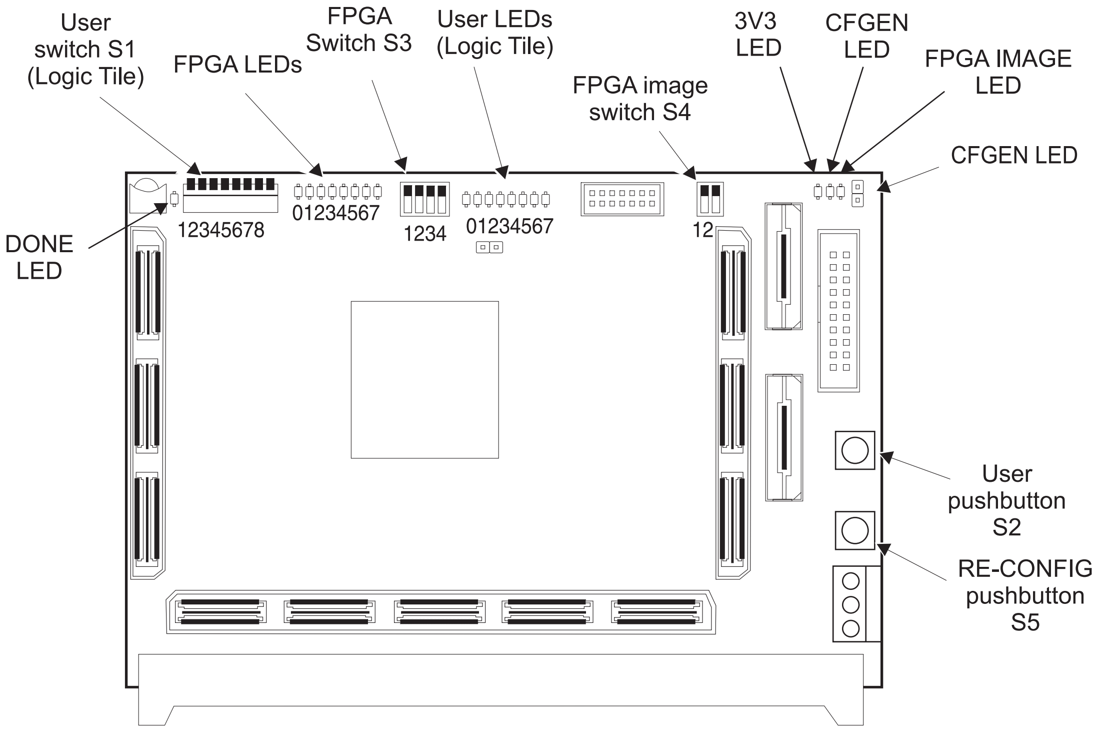


Figure 3-24 Switch and LED location

There are two pushbuttons and three switch banks fitted to the IM-LT3:

S1[8:1] These switches are connected to **YU[143:136]** and can be read under program control from a register implemented in an attached Logic Tile. (Switch S1[1] is connected to signal **YU[136]**.) If a switch is on, a logic HIGH signal is produced.

S2 This pushbutton switch pulls **nPB1** LOW and the FPGA resets the system.

S3[4:1] These switches (signals **SW[3:0]**) are connected to the FPGA. Switch S3[1] is signal **SW[0]**. If a switch is on, a logic HIGH signal is produced.

S4[2:1] These switches (signals **SEL[2:1]**) select the image to load into the FPGA at power on (see *Configuring the FPGA from flash* on page 4-5).

S5 This switch triggers (**CFG_nPROG**) reloads the selected configuration image from flash to the FPGA.

There two banks of eight user LEDs and four individual status LEDs:

3V3 LED D19 lights to indicate that 3.3V is connected to the module.

DONE LED D20 lights to indicate that FPGA configuration has completed.

CONFIG LED D25 lights to indicate that the CONFIG link is connected and the system is in configuration mode.

FPGA IMAGE

LED D21lights to indicate that image 1 in the top half of the flash memory is loaded into the FPGA. If unlit, image 0 is loaded into the FPGA.

FPGA LEDs

LEDs D11 to D18 are connected to the IM-LT3 FPGA (signals **R_LEDS[7:0]**).

A HIGH level lights the LED. (LED D11 is signal **R_LEDS[0]**.)

Logic Tile LEDs

LEDs D1 to D8 are connected to **YU[135:128]** and can be lit by an attached Logic Tile.

A HIGH level lights the LED. (LED D1 is connected to **YU[128]**.)

3.10 Test points

The IM-LT3 provides test points and debug connectors to aid diagnostics. The most useful of these are illustrated in Figure 3-25.

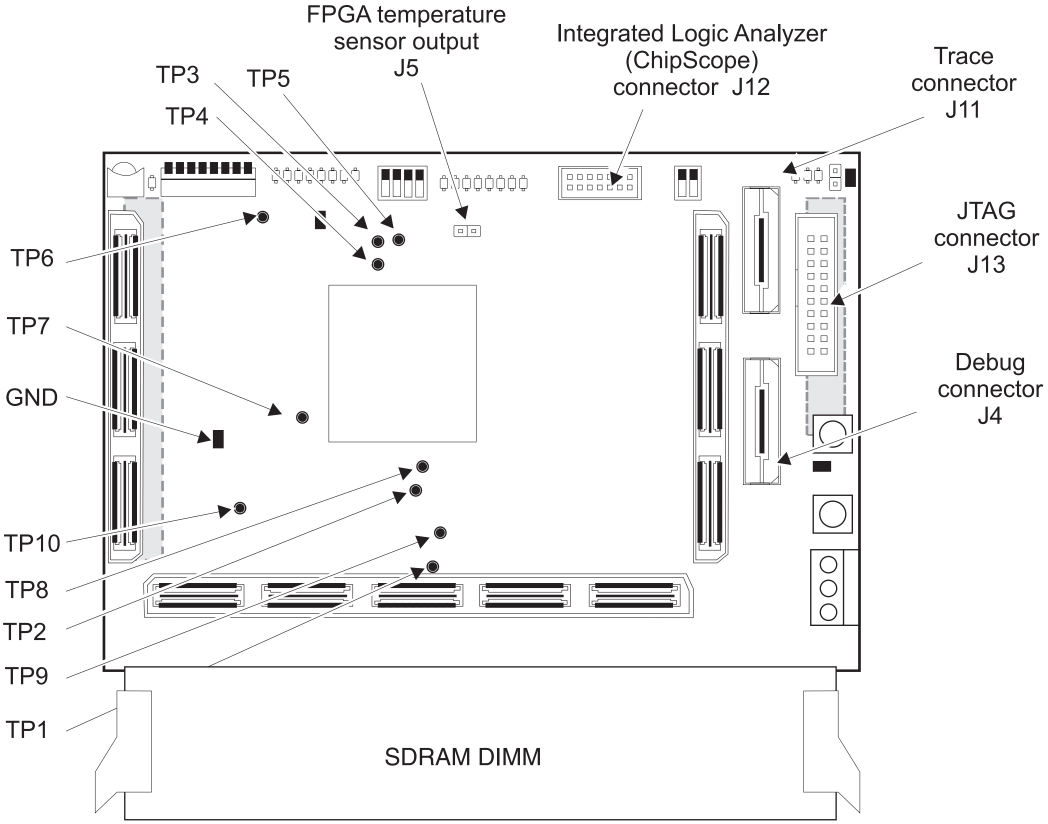


Figure 3-25 Test points

See *JTAG support* on page 3-25 for details of the JTAG connectors.

The test point signals are listed in Table 3-7.

Table 3-7 Test point functions

Test point	Signal	Signal
TP1	**TP_CLK_GLOBAL_IN**	Global clock in
TP2	**TP_CLK_GLOBAL_OUT**	Global clock out
TP3	**FPGA_TPA**	Spare signal from FPGA
TP4	**FPGA_TPB**	Spare signal from FPGA
TP5	**FPGA_TPC**	Spare signal from FPGA
TP6	**1V5**	FPGA VCCINT (core) supply voltage
TP7	**R_CLK0**	Output from ICS307 programmable oscillator U10
TP8	**R_CLK1**	Output from ICS307 programmable oscillator U11
TP9	**R_CLK2**	Output from ICS307 programmable oscillator U12
TP10	**TP_REF24MHZ**	24MHz reference for ISC307
J5	**DXN** and **DXP**	FPGA temperature sense output (connector not fitted)

3.10.1 Debug connectors

The IM-LT3 incorporates two Mictor connectors for a logic analyzer or Trace Port analyzer. These enable you to carry out real-time tests by connecting external test equipment to the IM-LT3. The Mictor connector pinout is described in *Mictor logic analyzer connectors* on page A-24.

Note

If you are connecting a Trace Port Analyzer to a Core Tile or Core Module, use the trace connectors on the Core Tile or Core Module. (See *Connecting Trace* on page 2-12.)

If an SMM is implemented in an attached Logic Tile, a basic trace port can be implemented with the Mictor connector J11 as shown in Figure 3-26 on page 3-44.

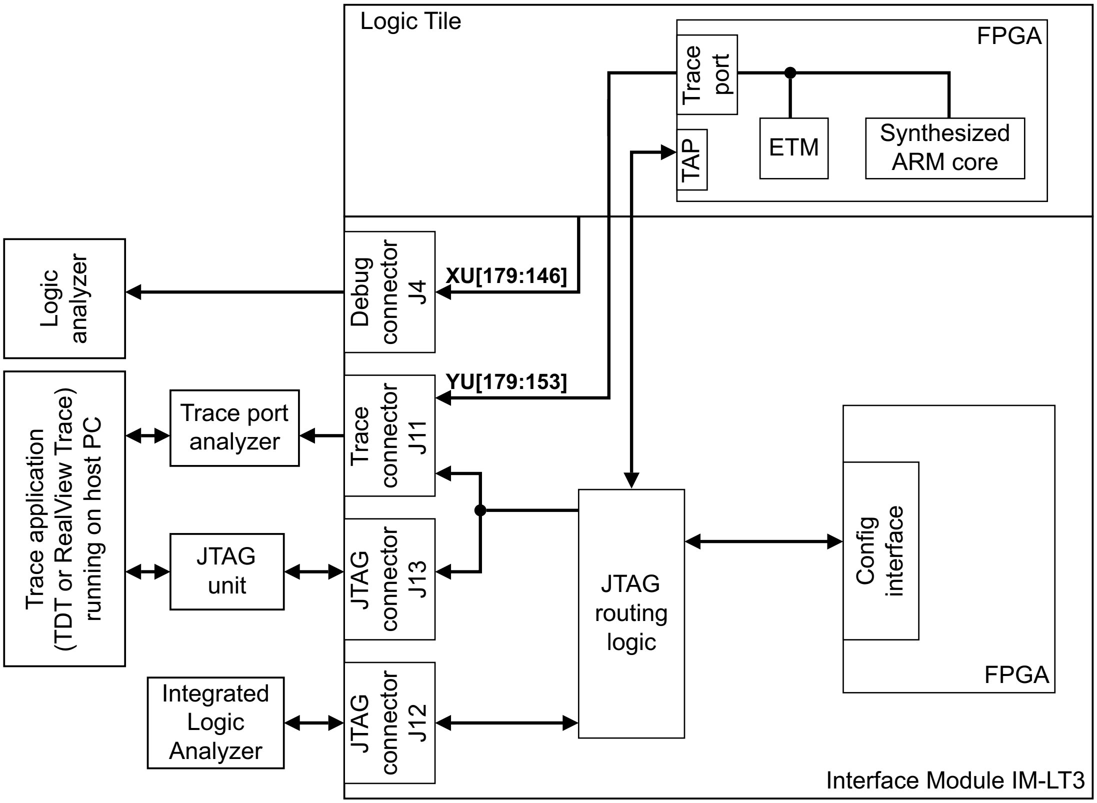


Figure 3-26 Trace and Logic analyzer connection block diagram

Chapter 4
Configuring the FPGA and PLD

This chapter describes how the PLD and the Xilinx FPGA on the IM-LT3 are configured at power-up, the configuration options available, and how to download your own configurations. It contains the following sections:

- *FPGA configuration system architecture* on page 4-2
- *FPGA tool flow* on page 4-3
- *Configuring the FPGA from flash* on page 4-5
- *Loading new FPGA configurations* on page 4-6
- *Reprogramming the PLD* on page 4-9.

——— **Note** ————

Using the IM-LT3 together with other products requires an appropriate FPGA images in both the Logic Tiles and the IM-LT3.

Using the IM-LT3 with different combinations of Integrator and tile products is described in application notes. Refer to the application notes on the CD supplied with the product (or the ARM web site at `www.arm.com`).

————————

4.1 FPGA configuration system architecture

At power-up the FPGA loads configuration data to its internal configuration memory. Figure 3-2 on page 3-6 shows the architecture of the FPGA configuration system.

There are two ways to load the FPGA image:

Debug mode The FPGA is configured from the flash memory using the select MAP mode of configuration.

This mode is the normal FPGA image loading mode. The loading process is managed by the configuration *Programmable Logic Device* (PLD). The flash must contain valid configuration data and the CONFIG link must not be fitted.

The flash memory can store two configuration images. The image is selected either by DIP switch S4[1] (see *Configuring the FPGA from flash* on page 4-5).

Configuration mode

The FPGA is configured directly by JTAG and uses the boundary scan mode of configuration. You can use the Multi-ICE JTAG port to download configurations when the CONFIG link is fitted (see *Loading new FPGA configurations* on page 4-6).

Note

You can also load a flash loading image into the FPGA and then use that image to control loading the flash memory itself with a new image.

4.2 FPGA tool flow

Preparing FPGA configuration files entails two steps:

1. Synthesis.
2. Place and route.

Figure 4-1 illustrates the basic tool flow process.

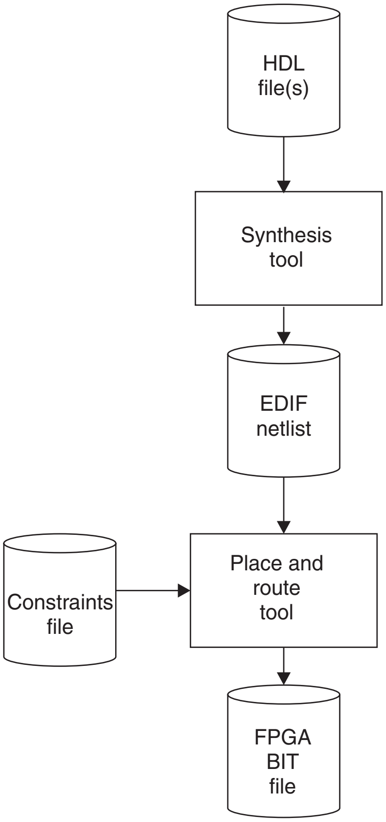


Figure 4-1 Basic tool flow

4.2.1 Synthesis

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

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

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

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

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

4.2.2 Place and route

Place and route for the IM-LT3 FPGA is performed using Xilinx-specific software. This produces a `.bit` file that is used to program the FPGA. The `.edf` file is aimed at a particular device, taking into account the device size, package type, and speed grade.

Note

Always specify **CCLK** as the start up clock for your design. The progcards utility automatically sets the startup clock to the JTAG clock option when you program the FPGA directly. Selecting **CCLK** ensures that the process always works for download into the FPGA or into flash.

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

Note

The `pinout.ucf` file for the complete FPGA pin allocation is supplied on the CD. This is intended as a starting point for any design, and must be edited before use in the place and route process.

4.3 Configuring the FPGA from flash

The flash memory can store two configurations for the XC2V2000 FPGA. The configuration image is selected according to the setting of FPGA image switch S4 as listed in Table 4-1.

Table 4-1 Image selection

FPGA IMAGE switch S4[1]	FPGA IMAGE switch S4[2]	Flash image used	Image LED	Image base address	Description
OFF	OFF	0	Unlit	`0x000000`	Use image 0
ON	OFF	1	Lit	`0x200000`	Use image 1
OFF	ON	0	Unlit	`0x000000`	Use image 0
ON	ON	1	Lit	`0x200000`	Use image 1

The **FPGA_IMAGE** signal that connects to the tile above is generated on the IM-LT3. If there is no motherboard present, it defaults to high.

See the application notes for a description of example images. See *Downloading new the FPGA configurations into flash* on page 4-8 for details of loading new images into the configuration flash. See *PLD circuitry* on page 3-5 for details of the configuration logic.

——— **Note** ———

FPGA switch S4[2] is connected to the PLD, but is reserved for future use.

If an Integrator motherboard is present, the **CFGSEL[1:0]** signals will be `b00`, `b01`, or `b10` depending on the type of motherboard. Pullup and pulldown resistors on the IM-LT3 drive **CFGSEL[1:0]** to `b01` if no motherboard is connected. These signals are reserved for future use and do not affect image selection if you are using the PLD image that is loaded at manufacture.

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

4.4 Loading new FPGA configurations

You can program the FPGA in two ways:

- writing configuration data directly to the FPGA using Multi-ICE
- writing configuration data to the flash memory using Multi-ICE.

To reconfigure the FPGA, the IM-LT3 must be in CONFIG mode. This is enabled by fitting the CONFIG link (J14). The CFGEN LED is lit as an indication that configure mode is selected.

For a description of CONFIG mode, see *Connecting Multi-ICE, RealView ICE or Trace* on page 2-11.

Caution

The 1.5V cell battery provides the **VBATT** backup voltage to the and FPGA encryption key circuitry within the FPGA. Removing the battery erases the encryption key.

Each board is provided with an encryption key that is unique to the board. The standard image supplied with the board is not encrypted. However, encrypted images might be supplied by ARM in the future, for example for synthesizable cores (SMM). If you are using encrypted images and the key is erased, you must return the board to ARM to have the key reloaded.

The battery is expected to last for approximately 10 years from manufacture of the IM-LT3. To replace the battery:

1. Power on the IM-LT3. If the battery is removed while the board is powered down, the encryption key will be erased.

2. Remove the old battery.

3. Insert the new battery and ensure that the positive terminal is facing upwards in the holder. Use care when handling the battery. Any grease that is transferred from your fingers to the positive terminal can bridge the battery contacts and reduce battery life.

4.4.1 Reconfiguring the FPGA directly

Using JTAG to program the FPGA is fast, but the configuration is lost when the power supply is removed. Programming takes approximately 30 seconds to complete using Multi-ICE on a fast computer (for example, a 400MHz Windows workstation).

You can reprogram the FPGA using Multi-ICE. A Multi-ICE client application called `Progcards` is provided to read `.bit` files and configure the FPGA using the Multi-ICE hardware. You must use a board file (`.brd`) to tell the `progcards` utility about the method of programming. Examples are provided on the CD supplied with the IM-LT3.

Note

The `Progcards` utility requires Multi-ICE release 1.4 or later.

Refer to the *Multi-ICE User Guide* for further information about using Multi-ICE.

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

To load a new configuration into the FPGA:

1. Produce a `<filename>.bit` file for your design.
2. Produce a `<filename>.brd` for your design. This is a configuration file for `progcards.exe`.
3. Start the JTAG application and autoconfigure.

   If autoconfiguration fails, load the configuration file (`.cfg`) for the board. For details on manual configuration, see the `readme.txt` file on the CD.
4. Configure the Multi-ICE server using a configuration file.
5. Run the `progcards` utility. All `.brd` files present in the current directory that match the TAP configuration are offered as options.
6. Do not power-cycle the system or press the reconfigure button. If reconfigured, the FPGA will reload the old image from the flash memory.

4.4.2 Downloading new the FPGA configurations into flash

The flash memory on the IM-LT3 configures the FPGA on power-up. Use the `progcards` utility is used to program the flash. `progcards` first loads a flash programmer design into the FPGA and PLD. It then writes the bit file to the flash memory. You can use the `progcards` utility to verify the flash image against a bit file.

The steps in writing a bit file to flash are similar to those described in *Reconfiguring the FPGA directly* on page 4-7. The only difference is the contents of the `.brd` file (examples are provided on the CD).

To load the new image from flash into the FPGA:

1. Remove the CONFIG link.
2. Power-cycle the system or press the reconfigure button.

4.5 Reprogramming the PLD

The IM-LT3 is supplied with the PLD already programmed. The information in this section is provided, however, in case of accidental erasure of the PLD or to update the PLD with a new design from ARM Ltd.

Caution

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

Program the PLD as follows:

1. Put the IM-LT3 into configuration mode by fitting the CONFIG link (J14) and power-up.
2. Start the Multi-ICE server and the load the configuration file (`.cfg`) for the IM-LT3.
3. Start a command prompt and move to the configure directory.
4. Run the `progcards` utility.
5. Choose the required PLD image.

Appendix A
Signal Descriptions

This appendix describes the pinouts and signals for the interface connectors on the IM-LT3. It contains the following sections:

A.1 HDRY tile connector

Figure A-1 shows the Samtec connector. Table A-1 shows the signals on HDRY.

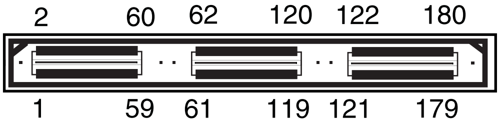


Figure A-1 180-pin Samtec connector

Table A-1 HDRY signals

Signal]	Odd Pin	Even Pin	Signal]
YU[89]	1	2	**YU[90]**
YU[88]	3	4	**YU[91]**
YU[87]	5	6	**YU[92]**
YU[86]	7	8	**YU[93]**
YU[85]	9	10	**YU[94]**
YU[84]	11	12	**YU[95]**
YU[83]	13	14	**YU[96]**
YU[82]	15	16	**YU[97]**
YU[81]	17	18	**YU[98]**
YU[80]	19	20	**YU[99]**
YU[79]	21	22	**YU[100]**
YU[78]	23	24	**YU[101]**
YU[77]	25	26	**YU[102]**
YU[76]	27	28	**YU[103]**
YU[75]	29	30	**YU[104]**
YU[74]	31	32	**YU[105]**
YU[73]	33	34	**YU[106]**
YU[72]	35	36	**YU[107]**

Table A-1 HDRY signals (continued)

Signal]	Odd Pin	Even Pin	Signal]
YU[71]	37	38	**YU[108]**
YU[70]	39	40	**YU[109]**
YU[69]	41	42	**YU[110]**
YU[68]	43	44	**YU[111]**
YU[67]	45	46	**YU[112]**
YU[66]	47	48	**YU[113]**
YU[65]	49	50	**YU[114]**
YU[64]	51	52	**YU[115]**
YU[63]	53	54	**YU[116]**
YU[62]	55	56	**YU[117]**
YU[61]	57	58	**YU[118]**
YU[60]	59	60	**YU[119]**
YU[59]	61	62	**YU[120]**
YU[58]	63	64	**YU[121]**
YU[57]	65	66	**YU[122]**
YU[56]	67	68	**YU[123]**
YU[55]	69	70	**YU[124]**
YU[54]	71	72	**YU[125]**
YU[53]	73	74	**YU[126]**
YU[52]	75	76	**YU[127]**
YU[51]	77	78	**YU[128]**
YU[50]	79	80	**YU[129]**
YU[49]	81	82	**YU[130]**
YU[48]	83	84	**YU[131]**

Table A-1 HDRY signals (continued)

Signal]	Odd Pin	Even Pin	Signal]
YU[47]	85	86	**YU[132]**
YU[46]	87	88	**YU[133]**
YU[45]	89	90	**YU[134]**
YU[44]	91	92	**YU[135]**
YU[43]	93	94	**YU[136]**
YU[42]	95	96	**YU[137]**
YU[41]	97	98	**YU[138]**
YU[40]	99	100	**YU[139]**
NC	101	102	**YU[140]**
NC	103	104	**YU[141]**
NC	105	106	**YU[142]**
NC	107	108	**YU[143]**
NC	109	110	NC
NC	111	112	NC
NC	113	114	NC
NC	115	116	NC
NC	117	118	NC
NC	119	120	NC
NC	121	122	NC
NC	123	124	NC
NC	125	126	NC
NC	127	128	**YU[153]**
NC	129	130	**YU[154]**
NC	131	132	**YU[155]**

Table A-1 HDRY signals (continued)

Signal]	Odd Pin	Even Pin	Signal]
NC	133	134	**YU[156]**
NC	135	136	**YU[157]**
NC	137	138	**YU[158]**
NC	139	140	**YU[159]**
NC	141	142	**YU[160]**
NC	143	144	**YU[161]**
NC	145	146	**YU[162]**
NC	147	148	**YU[163]**
NC	149	150	**YU[164]**
NC	151	152	**YU[165]**
NC	153	154	**YU[166]**
NC	155	156	**YU[167]**
NC	157	158	**YU[168]**
NC	159	160	**YU[169]**
NC	161	162	**YU[170]**
NC	163	164	**YU[171]**
NC	165	166	**YU[172]**
NC	167	168	**YU[173]**
NC	169	170	**YU[174]**
NC	171	172	**YU[175]**
NC	173	174	**YU[176]**
NC	175	176	**YU[177]**
NC	177	178	**YU[178]**
NC	179	180	**YU[179]**

A.2 HDRX tile connector

Figure A-1 on page A-2 shows the Samtec connector. Table A-2 shows the signals on HDRX.

Table A-2 HDRX signals

Signal	pin	pin	Signal]
XU[90]	1	2	**XU[89]**
XU[91]	3	4	**XU[88]**
XU[92]	5	6	**XU[87]**
XU[93]	7	8	**XU[86]**
XU[94]	9	10	**XU[85]**
XU[95]	11	12	**XU[84]**
XU[96]	13	14	**XU[83]**
XU[97]	15	16	**XU[82]**
XU[98]	17	18	**XU[81]**
XU[99]	19	20	**XU[80]**
XU[100]	21	22	**XU[79]**
XU[101]	23	24	**XU[78]**
XU[102]	25	26	**XU[77]**
XU[103]	27	28	**XU[76]**
XU[104]	29	30	**XU[75]**
XU[105]	31	32	**XU[74]**
XU[106]	33	34	**XU[73]**
XU[107]	35	36	**XU[72]**
XU[108]	37	38	**XU[71]**
XU[109]	39	40	**XU[70]**
XU[110]	41	42	**XU[69]**
XU[111]	43	44	**XU[68]**

Table A-2 HDRX signals (continued)

Signal	pin	pin	Signal]
XU[112]	45	46	**XU[67]**
XU[113]	47	48	**XU[66]**
XU[114]	49	50	**XU[65]**
XU[115]	51	52	**XU[64]**
XU[116]	53	54	**XU[63]**
XU[117]	55	56	**XU[62]**
XU[118]	57	58	**XU[61]**
XU[119]	59	60	**XU[60]**
XU[120]	61	62	**XU[59]**
XU[121]	63	64	**XU[58]**
XU[122]	65	66	**XU[57]**
XU[123]	67	68	**XU[56]**
XU[124]	69	70	**XU[55]**
XU[125]	71	72	**XU[54]**
XU[126]	73	74	**XU[53]**
XU[127]	75	76	**XU[52]**
XU[128]	77	78	**XU[51]**
XU[129]	79	80	**XU[50]**
XU[130]	81	82	**XU[49]**
XU[131]	83	84	**XU[48]**
XU[132]	85	86	**XU[47]**
XU[133]	87	88	**XU[46]**
XU[134]	89	90	**XU[45]**
XU[135]	91	92	**XU[44]**
XU[136]	93	94	**XU[43]**

Table A-2 HDRX signals (continued)

Signal	pin	pin	Signal]
XU[137]	95	96	**XU[42]**
XU[138]	97	98	**XU[41]**
XU[139]	99	100	**XU[40]**
XU[140]	101	102	**XU[39]**
XU[141]	103	104	**XU[38]**
XU[142]	105	106	**XU[37]**
XU[143]	107	108	**XU[36]**
NC	109	110	**XU[35]**
NC	111	112	**XU[34]**
XU[146]	113	114	**XU[33]**
XU[147]	115	116	**XU[32]**
XU[148]	117	118	**XU[31]**
XU[149]	119	120	**XU[30]**
XU[150]	121	122	**XU[29]**
XU[151]	123	124	**XU[28]**
XU[152]	125	126	**XU[27]**
XU[153]	127	128	**XU[26]**
XU[154]	129	130	**XU[25]**
XU[155]	131	132	**XU[24]**
XU[156]	133	134	**XU[23]**
XU[157]	135	136	**XU[22]**
XU[158]	137	138	**XU[21]**
XU[159]	139	140	**XU[20]**
XU[160]	141	142	**XU[19]**
XU[161]	143	144	**XU[18]**

Table A-2 HDRX signals (continued)

Signal	pin	pin	Signal]
XU[162]	145	146	**XU[17]**
XU[163]	147	148	**XU[16]**
XU[164]	149	150	**XU[15]**
XU[165]	151	152	**XU[14]**
XU[166]	153	154	**XU[13]**
XU[167]	155	156	**XU[12]**
XU[168]	157	158	**XU[11]**
XU[169]	159	160	**XU[10]**
XU[170]	161	162	**XU[9]**
XU[171]	163	164	**XU[8]**
XU[172]	165	166	**XU[7]**
XU[173]	167	168	**XU[6]**
XU[174]	169	170	**XU[5]**
XU[175]	171	172	**XU[4]**
XU[176]	173	174	**XU[3]**
XU[177]	175	176	**XU[2]**
XU[178]	177	178	**XU[1]**
XU[179]	179	180	**XU[0]**

A.3 HDRZ tile connector

Figure A-2 shows the HDRZ connector. Table A-3 shows the signals on HDRZ.

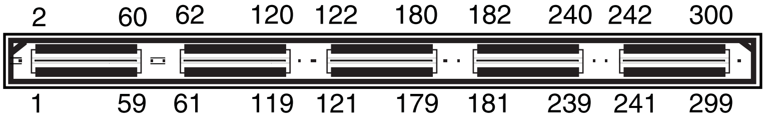


Figure A-2 300-pin Samtec connector

Table A-3 HDRZ signals

Signal	Pin	Pin	Signal
ZU[128]	1	2	NC
ZU[129]	3	4	NC
Z[130]	5	6	NC
Z[131]	7	8	NC
Z[132]	9	10	NC
Z[133]	11	12	NC
Z[134]	13	14	NC
Z[135]	15	16	NC
Z[136]	17	18	NC
Z[137]	19	20	NC
Z[138]	21	22	NC
Z[139]	23	24	NC
Z[140]	25	26	NC
Z[141]	27	28	NC
Z[142]	29	30	NC
Z[143]	31	32	NC
Z[144]	33	34	NC
Z[145]	35	36	NC
Z[146]	37	38	NC

Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
Z[147]	39	40	NC
Z[148]	41	42	NC
Z[149]	43	44	NC
Z[150]	45	46	**ZU[233]**
Z[151]	47	48	**ZU[232]**
Z[152]	49	50	**ZU[231]**
Z[153]	51	52	**ZU[230]**
Z[154]	53	54	**ZU[229]**
Z[155]	55	56	**ZU[228]**
Z[156]	57	58	**ZU[227]**
Z[157]	59	60	**ZU[226]**
Z[158]	61	62	**ZU[225]**
Z[159]	63	64	**ZU[224]**
Z[160]	65	66	**ZU[223]**
Z[161]	67	68	**ZU[222]**
Z[162]	69	70	**ZU[221]**
Z[163]	71	72	**ZU[220]**
Z[164]	73	74	**ZU[219]**
Z[165]	75	76	**ZU[218]**
Z[166]	77	78	**ZU[217]**
Z[167]	79	80	**ZU[216]**
Z[168]	81	82	**ZU[215]**
Z[169]	83	84	**ZU[214]**
Z[170]	85	86	**ZU[213]**
Z[171]	87	88	**ZU[212]**

Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
Z[172]	89	90	**ZU[211]**
Z[173]	91	92	**ZU[210]**
Z[174]	93	94	**ZU[209]**
Z[175]	95	96	**ZU[208]**
Z[176]	97	98	**ZU[207]**
Z[177]	99	100	**ZU[206]**
Z[178]	101	102	**ZU[205]**
Z[179]	103	104	**ZU[204]**
Z[180]	105	106	**ZU[203]**
Z[181]	107	108	**ZU[202]**
Z[182]	109	110	**ZU[201]**
Z[183]	111	112	**ZU[200]**
Z[184]	113	114	**ZU[199]**
Z[185]	115	116	**ZU[198]**
Z[186]	117	118	**ZU[197]**
Z[187]	119	120	**ZU[196]**
Z[188]	121	122	**ZU[195]**
Z[189]	123	124	**ZU[194]**
ZU[190]	125	126	**ZU[193]**
ZU[191]	127	128	**ZU[192]**
nSRST	129	130	NC
nTRST	131	132	NC
TILE_D_TDO	133	134	**CLK_POS_UP_OUT**
TILE_D_TDI	135	136	**CLK_NEG_UP_OUT**
TILE_D_TCK	137	138	**SYSCLK1**

Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
TILE_D_TMS	139	140	**SYSCLK2**
TILE_RTCK	141	142	**SYSCLK3**
nSRST	143	144	NC
CFG_NPROG	145	146	NC
TILE_C_TDO	147	148	NC
TILE_C_TDI	149	150	**CLK_GLOBAL**
TILE_C_TCK	151	152	**FPGA_IMAGE**
TILE_C_TMS	153	154	**nSYSPOR**
nTILE_DET	155	156	**nSYSRST**
nCFGEN	157	158	**nRTCKEN**
GLOBAL_DONE	159	160	NC
NC	161	162	NC
NC	163	164	NC
NC	165	166	NC
NC	167	168	NC
NC	169	170	NC
NC	171	172	NC
Z[63]	173	174	**Z[64]**
Z[62]	175	176	**Z[65]**
Z[61]	177	178	**Z[66]**
Z[60]	179	180	**Z[67]**
Z[59]	181	182	**Z[68]**
Z[58]	183	184	**Z[69]**
Z[57]	185	186	**Z[70]**
Z[56]	187	188	**Z[71]**

Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
Z[55]	189	190	**Z[72]**
Z[54]	191	192	**Z[73]**
Z[53]	193	194	**Z[74]**
Z[52]	195	196	**Z[75]**
Z[51]	197	198	**Z[76]**
Z[50]	199	200	**Z[77]**
Z[49]	201	202	**Z[78]**
Z[48]	203	204	**Z[79]**
Z[47]	205	206	**Z[80]**
Z[46]	207	208	**Z[81]**
Z[45]	209	210	**Z[82]**
Z[44]	211	212	**Z[83]**
Z[43]	213	214	**Z[84]**
Z[42]	215	216	**Z[85]**
Z[41]	217	218	**Z[86]**
Z[40]	219	220	**Z[87]**
Z[39]	221	222	**Z[88]**
Z[38]	223	224	**Z[89]**
Z[37]	225	226	**Z[90]**
Z[36]	227	228	**Z[91]**
Z[35]	229	230	**Z[92]**
Z[34]	231	232	**Z[93]**
Z[33]	233	234	**Z[94]**
Z[32]	235	236	**Z[95]**
Z[31]	237	238	**Z[96]**

Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
Z[30]	239	240	**Z[97]**
Z[29]	241	242	**Z[98]**
Z[28]	243	244	**Z[99]**
Z[27]	245	246	**Z[100]**
Z[26]	247	248	**Z[101]**
Z[25]	249	250	**Z[102]**
Z[24]	251	252	**Z[103]**
Z[23]	253	254	**Z[104]**
Z[22]	255	256	**Z[105]**
Z[21]	257	258	**Z[106]**
Z[20]	259	260	**Z[107]**
Z[19]	261	262	**Z[108]**
Z[18]	263	264	**Z[109]**
Z[17]	265	266	**Z[110]**
Z[16]	267	268	**Z[111]**
Z[15]	269	270	**Z[112]**
Z[14]	271	272	**Z[113]**
Z[13]	273	274	**Z[114]**
Z[12]	275	276	**Z[115]**
Z[11]	277	278	**Z[116]**
Z[0]	279	280	**Z[117]**
Z[9]	281	282	**Z[118]**
Z[8]	283	284	**Z[119]**
Z[7]	285	286	**Z[120]**
Z[6]	287	288	**Z[121]**

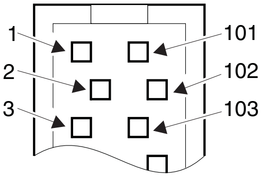
Table A-3 HDRZ signals (continued)

Signal	Pin	Pin	Signal
Z[5]	289	290	**Z[122]**
Z[4]	291	292	**Z[123]**
Z[3]	293	294	**Z[124]**
Z[2]	295	296	**Z[125]**
Z[1]	297	298	**Z[126]**
Z[0]	299	300	**Z[127]**

A.4 HDRA Integrator connector

Figure A-3 shows the pin numbers of the HDRA socket.

Pin numbers for 200-way plug, viewed from above board



Samtec TOLC series

1	A0	101	GND
2	GND	102	D0
3	A1	103	D1
4	A2	104	D2
5	A3	105	GND
6	GND	106	D3
7	A4	107	D4
8	A5	108	D5
9	A6	109	GND
10	GND	110	D6
11	A7	111	D7
12	A8	112	D8
13	A9	113	GND
14	GND	114	D9
15	A10	115	D10
16	A11	116	D11
17	A12	117	GND
18	GND	118	D12
19	A13	119	D13
20	A14	120	D14
21	A15	121	GND
22	GND	122	D15
23	A16	123	D16
24	A17	124	D17
25	A18	125	GND
26	GND	126	D18
27	A19	127	D19
28	A20	128	D20
29	A21	129	GND
30	GND	130	D21
31	A22	131	D22
32	A23	132	D23
33	A24	133	GND
34	GND	134	D24
35	A25	135	D25
36	A26	136	D26
37	A27	137	GND
38	GND	138	D27
39	A28	139	D28
40	A29	140	D29
41	A30	141	GND
42	GND	142	D30
43	A31	143	D31
44	B0	144	C0
45	B1	145	GND
46	GND	146	C1
47	B2	147	C2
48	B3	148	C3
49	B4	149	GND
50	GND	150	C4
51	B5	151	C5
52	B6	152	C6
53	B7	153	GND
54	GND	154	C7
55	B8	155	C8
56	B9	156	C9
57	B10	157	GND
58	GND	158	C10
59	B11	159	C11
60	B12	160	C12
61	B13	161	GND
62	GND	162	C13
63	B14	163	C14
64	B15	164	C15
65	B16	165	GND
66	GND	166	C16
67	B17	167	C17
68	B18	168	C18
69	B19	169	GND
70	GND	170	C19
71	B20	171	C20
72	B21	172	C21
73	B22	173	GND
74	GND	174	C22
75	B23	175	C23
76	B24	176	C24
77	B25	177	GND
78	GND	178	C25
79	B26	179	C26
80	B27	180	C27
81	B28	181	GND
82	GND	182	C28
83	B29	183	C29
84	B30	184	C30
85	B31	185	GND
86	GND	186	C31
87	5V	187	3V3
88	3V3	188	12V
89	5V	189	3V3
90	3V3	190	12V
91	5V	191	3V3
92	3V3	192	12V
93	5V	193	3V3
94	3V3	194	12V
95	5V	195	3V3
96	3V3	196	12V
97	5V	197	3V3
98	3V3	198	12V
99	5V	199	3V3
100	3V3	200	12V

Figure A-3 HDRA socket pin numbering

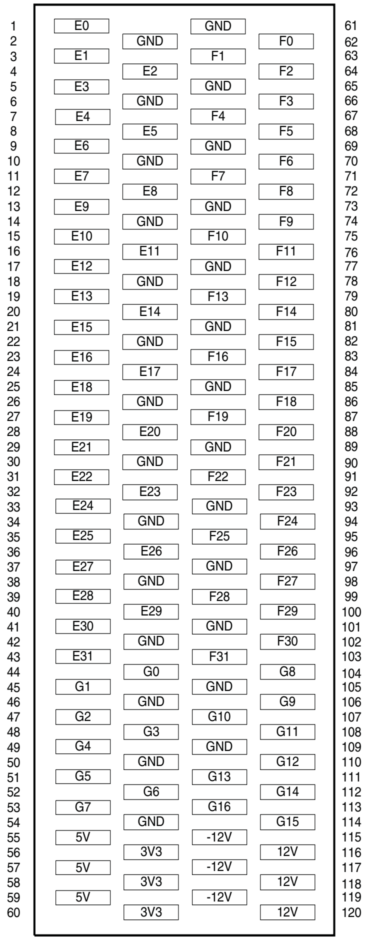
The signals are described in Table A-4.

Table A-4 Bus bit assignment

Pin label	AHB signal name	Description
A[31:0]	**HADDR[31:0]**	System address bus
B[31:0]	-	Connected to FPGA
C[31:28]	**DBGXTRIG[3:0]**	Debug cross-trigger signals
C[27:16]	Not used	-
C15	**HMASTLOCK**	Locked transaction
C14	**HRESP1**	Slave response
C13	**HRESP0**	Slave response
C12	**HREADY**	Slave wait response
C11	**HWRITE**	Write transaction
C[10:8]	**HPROT[2:0]**	Transaction protection type
C[7:5]	**HBURST[2:0]**	Transaction burst size
C4	**HPROT[3]**	Transaction protection type
C[3:2]	**HSIZE[1:0]**	Transaction width
C[1:0]	**HTRAN[1:0]**	Transaction type
D[31:0]	**HDATA[31:0]**	System data bus

A.5 HDRB Integrator connector

Figure A-4 shows the pin numbers of the socket HDRB on the underside of the IM-LT3, viewed from above the Interface Module.



1	E0	GND	61
2	GND	F0	62
3	E1	F1	63
4	E2	F2	64
5	E3	GND	65
6	GND	F3	66
7	E4	F4	67
8	E5	F5	68
9	E6	GND	69
10	GND	F6	70
11	E7	F7	71
12	E8	F8	72
13	E9	GND	73
14	GND	F9	74
15	E10	F10	75
16	E11	F11	76
17	E12	GND	77
18	GND	F12	78
19	E13	F13	79
20	E14	F14	80
21	E15	GND	81
22	GND	F15	82
23	E16	F16	83
24	E17	F17	84
25	E18	GND	85
26	GND	F18	86
27	E19	F19	87
28	E20	F20	88
29	E21	GND	89
30	GND	F21	90
31	E22	F22	91
32	E23	F23	92
33	E24	GND	93
34	GND	F24	94
35	E25	F25	95
36	E26	F26	96
37	E27	GND	97
38	GND	F27	98
39	E28	F28	99
40	E29	F29	100
41	E30	GND	101
42	GND	F30	102
43	E31	F31	103
44	G0	G8	104
45	G1	GND	105
46	GND	G9	106
47	G2	G10	107
48	G3	G11	108
49	G4	GND	109
50	GND	G12	110
51	G5	G13	111
52	G6	G14	112
53	G7	G16	113
54	GND	G15	114
55	5V	-12V	115
56	3V3	12V	116
57	5V	-12V	117
58	3V3	12V	118
59	5V	-12V	119
60	3V3	12V	120

Figure A-4 HDRB socket pin numbering (X-ray view)

A.5.1 HDRB signal descriptions

Table A-5 describes the signals for an AMBA AHB system bus on an Integrator motherboard. If the signal also connects to a tile header, the tile signal name is shown in parentheses. See also *Isolation of HDRZ signals* on page 3-14.

Table A-5 HDRB signal description (AHB)

Pin label	Integrator signal name	Description
E[31:28]	**SYSCLK[3:0]**	System clocks. The source of the clocks depend on the image loaded into the FPGA and the attached boards.
E[27:24]	**nPPRES[3:0]**	Processor present (**YU[123:120]**)
E[23:20]	**nIRQ[3:0]**	Interrupt request to processors 3, 2, 1, and 0 respectively (**YU[119:116]**)
E[19:16]	**nFIQ[3:0]**	Fast interrupt requests to processors 3, 2, 1, and 0 respectively (**YU[115:112]**)
E[15:12]	**ID[3:0]**	Core tile stack position indicator (**YU[111:108]**)
E[11:8]	**HLOCK[3:0]**	System bus lock from processor 3, 2, 1, and 0 respectively (**YU[107:104]**)
E[7:4]	**HGRANT[3:0]**	System bus grant to processor 3, 2, 1, and 0 respectively (**YU[103:100]**)
E[3:0]	**HBUSREQ[3:0]**	System bus request from processors 3, 2, 1, and 0 respectively (**YU[99:96]**)
F[31:0]	-	**YU[95:64]**
G16	**nRTCKEN**	RTCK AND gate enable
G[15:14]	**CFGSEL[1:0]**	FPGA configuration select
G13	**nCFGEN**	Sets motherboard into configuration mode
G12	**nSRST**	Multi-ICE reset (open collector)
G11	**FPGADONE**	Indicates when FPGA configuration is complete (open collector). This signal is connected to the tile signal **GLOBAL_DONE**.
G10	**RTCK**	Returned JTAG test clock
G9	**nSYSRST**	Buffered system reset

Table A-5 HDRB signal description (AHB) (continued)

Pin label	Integrator signal name	Description
G8	**nTRST**	JTAG reset
G7	**TDO**	JTAG test data out
G6	**TDI**	JTAG test data in
G5	**TMS**	JTAG test mode select
G4	**TCK**	JTAG test clock
G[3:1]	**MASTER[2:0]**	Master ID. Binary encoding of the master currently performing a transfer on the bus. Corresponds to the tile ID and to the **HBUSREQ** and **HGRANT** line numbers. (**YU[127:125]**)
G0	**nMBDET**	Motherboard detect pin (**YU[124]**)

A.6 JTAG connector signals

Figure A-5 shows the pinout of the JTAG connector and Table 3-6 on page 3-27 lists the JTAG and related signals. All JTAG active HIGH input signals have pull-up resistors except **DGBRQ** which is active LOW and has a pull-down resistor).

Note

The term JTAG equipment refers to any hardware that can drive the JTAG signals to devices in the scan chain. Typically this is RealView ICE or Multi-ICE, although hardware from other suppliers can also be used to debug ARM processors.

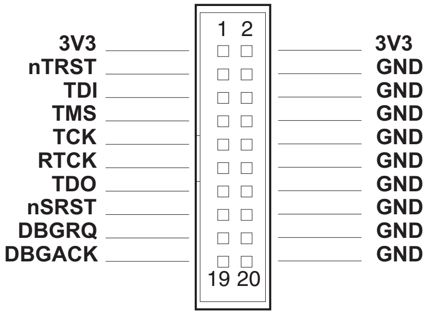


Figure A-5 Multi-ICE JTAG connector

A.7 Integrated logic analyzer connector signals

Figure A-6 shows the signals on the integrated logic analyzer connector J12. Use an embedded logic analyzer to debug FPGA designs and software (running on a physical or synthesized ARM core) at the same time. The ILA signals use a separate JTAG path to the FPGA and remain functional even if the system is in debug mode. For more information, see the documentation supplied with your analyzer. (The ChipScope product is described on the Xilinx web site at `www.xilinx.com`.)

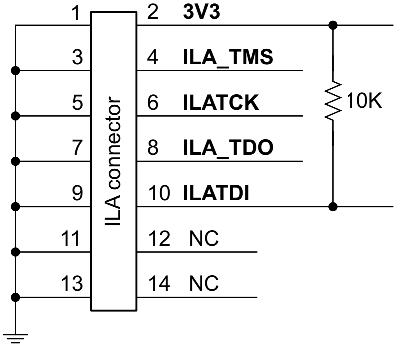


Figure A-6 Embedded logic analyzer connector J33

Note

Pins 4,6, and 10 have 10kΩ pullups to 3.3V to maintain the signals at an inactive level unless an analyzer is connected.

Connector J12 uses a 2mm pitch connector. The Xilinx JTAG cable is compatible with this connector. Use a flying probe or make an adaptor cable if a different JTAG interface is used. Using a different JTAG cable and interface might require a reduction in the JTAG clock speed (to 1MHz for example).

A.8 Mictor logic analyzer connectors

Two Mictor connectors on the IM-LT3 enable monitoring of header signals by an external logic analyzer. Figure A-7 shows the pinout of the Mictor connectors.

Table A-6 on page A-25 lists the pinouts of debug connector J11. A logic analyzer can be attached to J11 to monitor signals **YU[179:153]**. If an SMM is implemented in the FPGA on an attached Logic Tile, signals **YU[179:153]** can be used as a trace port for the synthesized core. The **Trace signal** names in Table A-6 on page A-25 refer to the signal names typically used on trace connectors. The **IM-LT3 signal** names refer to the signals names used on the on the IM-LT3 schematic.

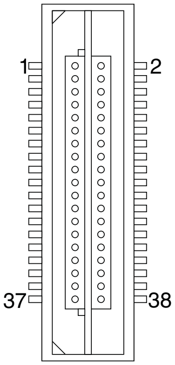


Figure A-7 Mictor connector pinout

Table A-6 Trace/Debug connector (J11)

IM-LT3 signal	Trace signal	Pin		Trace signal	IM-LT3 signal
NC	NC	1	2	NC	NC
NC	NC	3	4	NC	NC
GND	**GND**	5	6	**TRACECLK**	**YU179**
YU177	**DBGRQ**	7	8	**DBGACK**	**YU168**
nSRST	**nSRST**	9	10	**EXTTRIG**	**YU167**
TDO	**TDO**	11	12	**3V3**	**YU166**
RTCK	**RTCK**	13	14	**3V3**	**YU165**
TCK	**TCK**	15	16	**TRACEPKT7**	**YU164**
TMS	**TMS**	17	18	**TRACEPKT6**	**YU163**
TDI	**TDI**	19	20	**TRACEPKT5**	**YU162**
nTRST	**nTRST**	21	22	**TRACEPKT4**	**YU161**
YU176	**TRACEPKT15**	23	24	**TRACEPKT3**	**YU160**
YU175	**TRACEPKT14**	25	26	**TRACEPKT2**	**YU159**
YU174	**TRACEPKT13**	27	28	**TRACEPKT1**	**YU158**
YU173	**TRACEPKT12**	29	30	**TRACEPKT0**	**YU157**
YU172	**TRACEPKT11**	31	32	**PIPESTAT3**	**YU156**
YU171	**TRACEPKT10**	33	34	**PIPESTAT2**	**YU155**
YU170	**TRACEPKT9**	35	36	**PIPESTAT1**	**YU154**
YU169	**TRACEPKT8**	37	38	**PIPESTAT0**	**YU153**

Table A-7 lists the pinouts of debug connector J4. A logic analyzer can be attached to J4 to monitor signals **XU[179:146]**. (These signals are present on the upper HDRX connector.)

Table A-7 Debug connector (J4)

IM-LT3 signal	Pin	Pin	IM-LT3 signal
NC	1	2	NC
NC	3	4	NC
XU179	5	6	**XU178**
XU177	7	8	**XU161**
XU176	9	10	**XU160**
XU175	11	12	**XU159**
XU174	13	14	**XU158**
XU173	15	16	**XU157**
XU172	17	18	**XU156**
XU171	19	20	**XU155**
XU170	21	22	**XU154**
XU159	23	24	**XU153**
XU158	25	26	**XU152**
XU157	27	28	**XU151**
XU156	29	30	**XU150**
XU155	31	32	**XU149**
XU154	33	34	**XU148**
XU153	35	36	**XU147**
XU177	37	38	**XU146**

Appendix B
Specifications

This appendix describes the specifications for the IM-LT3 Interface Module. It contains the following sections:

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

B.1 Electrical specification

This section provides details of the voltage and current characteristics for the IM-LT3.

B.1.1 Bus interface characteristics

Table B-1 shows the IM-LT3 electrical characteristics for the system bus interface. The Interface Module uses 3.3V and 5V sources. The 12V inputs are supplied by the baseboard but not used by the IM-LT3 or tile.

Table B-1 IM-LT3 electrical characteristics

Symbol	Description	Min	Max	Unit
3V3	Supply voltage (interface signals)	3.1	3.5	V
5V	Supply voltage	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 typical current requirements at room temperature and nominal voltage. These measurements include the current drawn by Multi-ICE (approximately 160mA at 3.3V). The current actually drawn depends on the design in the IM-LT3 and Logic Tiles and on the clock speed.

Table B-2 Current requirements

System	At 3.3V	At 5V
Standalone operation (one tile)	1A	100mA
Integrator/CP baseboard, IM-LT3, and one tile	1.5A	500mA

B.2 Mechanical details

Figure B-1 and Figure B-2 on page B-4 show the dimensions of the IM-LT3 and the location of the header connectors.

The top header connector reference lines are for pin one and tolerances are +/- 0.01mm.

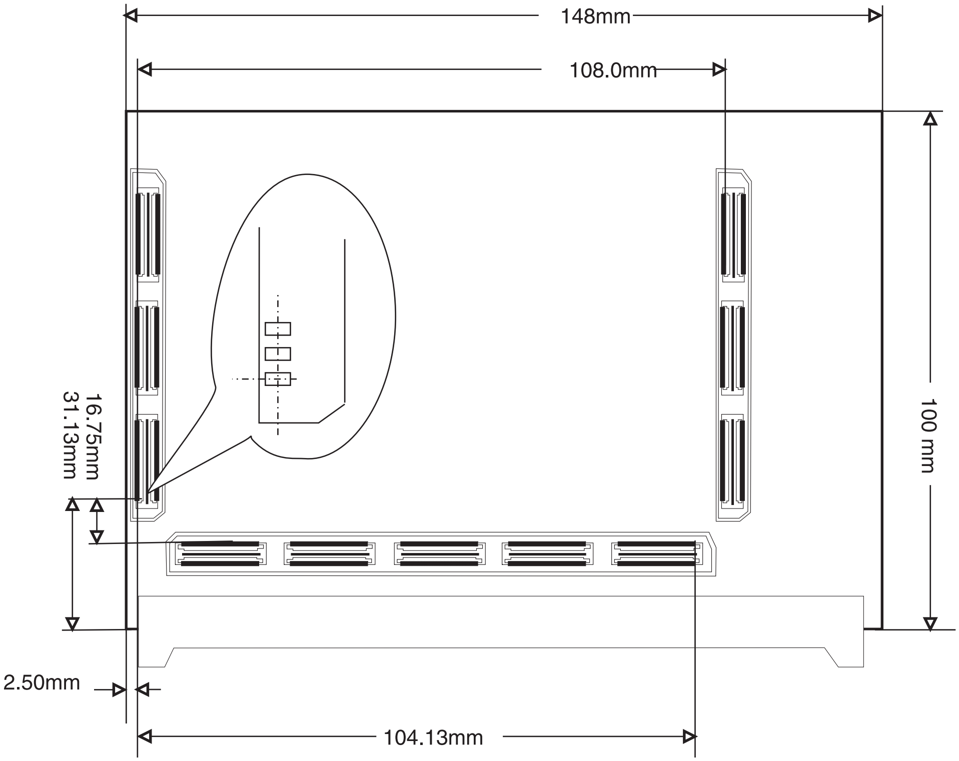


Figure B-1 Board dimensions (top view)

Bottom connector reference lines are for the center line between the pin rows and tolerances are +/- 0.01mm.

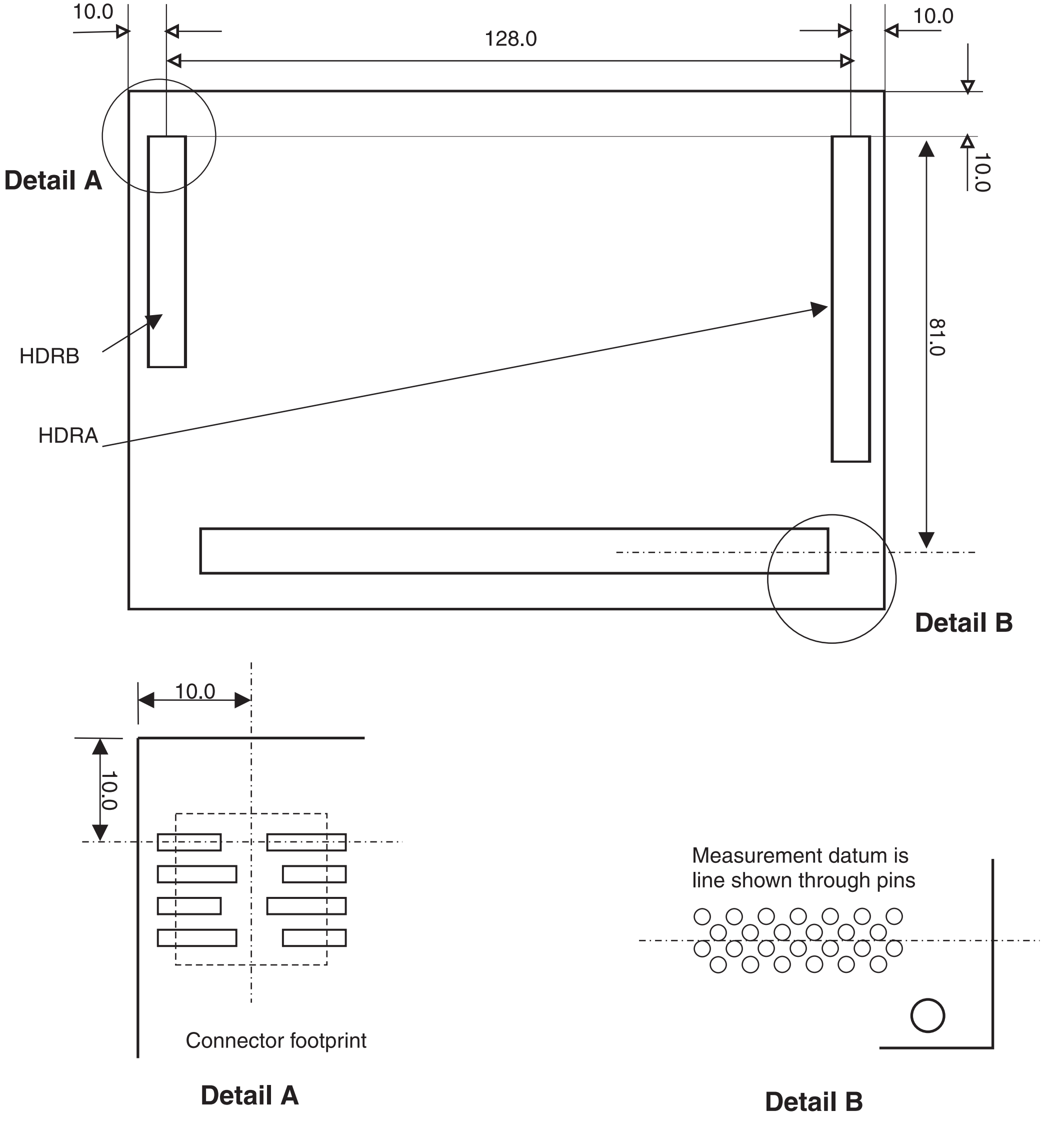


Figure B-2 Board dimensions (bottom view)

Appendix C
Using a Core Tile and an IM-LT3 as a Core Module

This appendix contains the instructions for using a CT7TDMI, CT926EJ-S, CT1026EJ-S, or CT1136JF-S Core Tile and an IM-LT3 to duplicate the functionality of the equivalent Core Module on an Integrator/CP baseboard. It contains the following sections:

Caution

Refer to the application notes for how to use Core Tile, Logic Tiles, and an IM-LT3 in new designs.

This appendix is only for the use of Core Tiles to maintain backward compatibility with legacy designs that use Core Modules on an Integrator/CP.

C.1 About the system architecture

Figure C-1 shows the architecture of a system consisting of an Integrator/CP baseboard, a Core Tile, and a IM-LT3 replacing a Core Module.

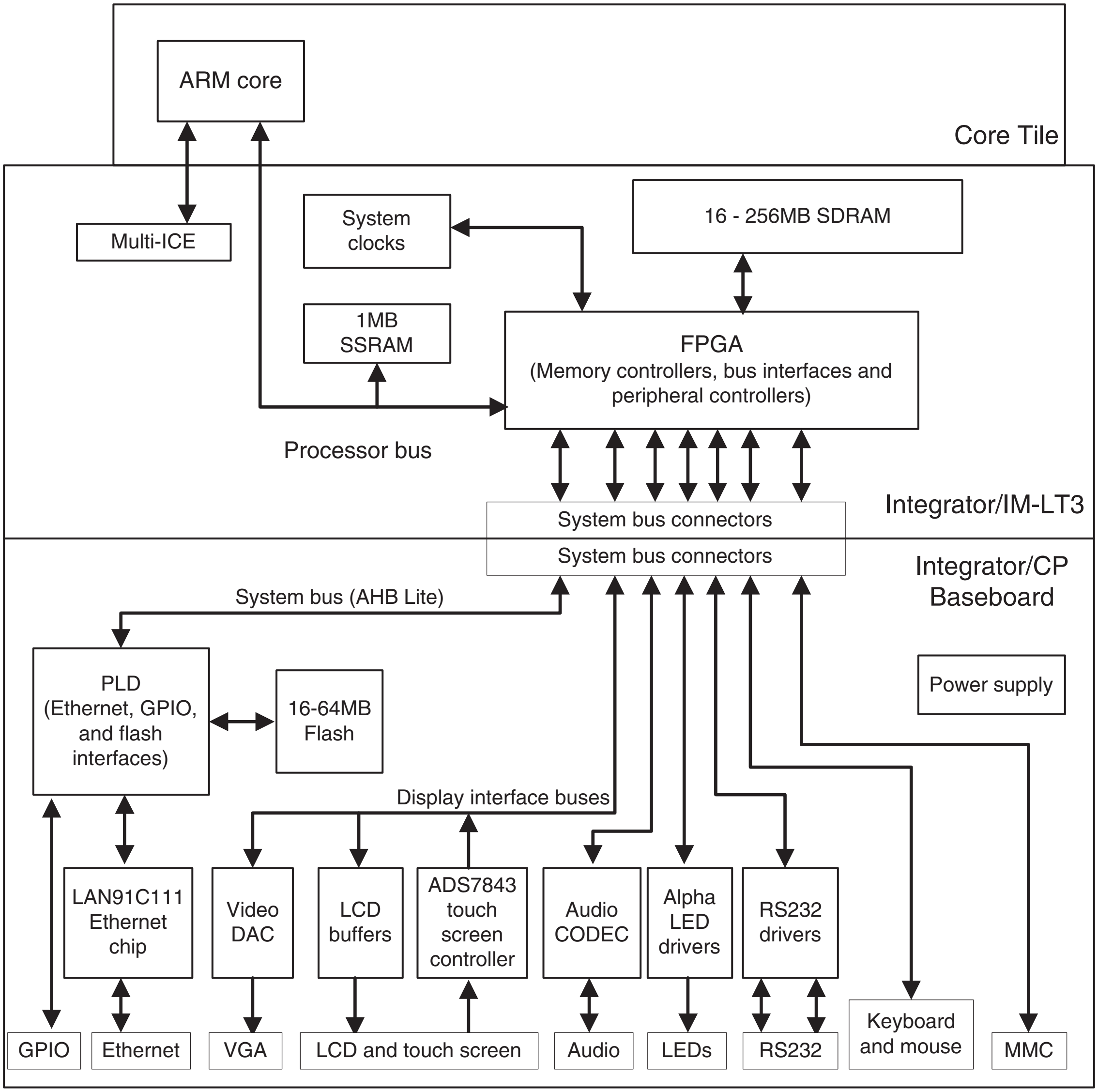


Figure C-1 Integrator/CP system architecture

C.1.1 Integrator/CP system buses

Figure C-2 shows how the external system bus and the internal buses connect the various boards together.

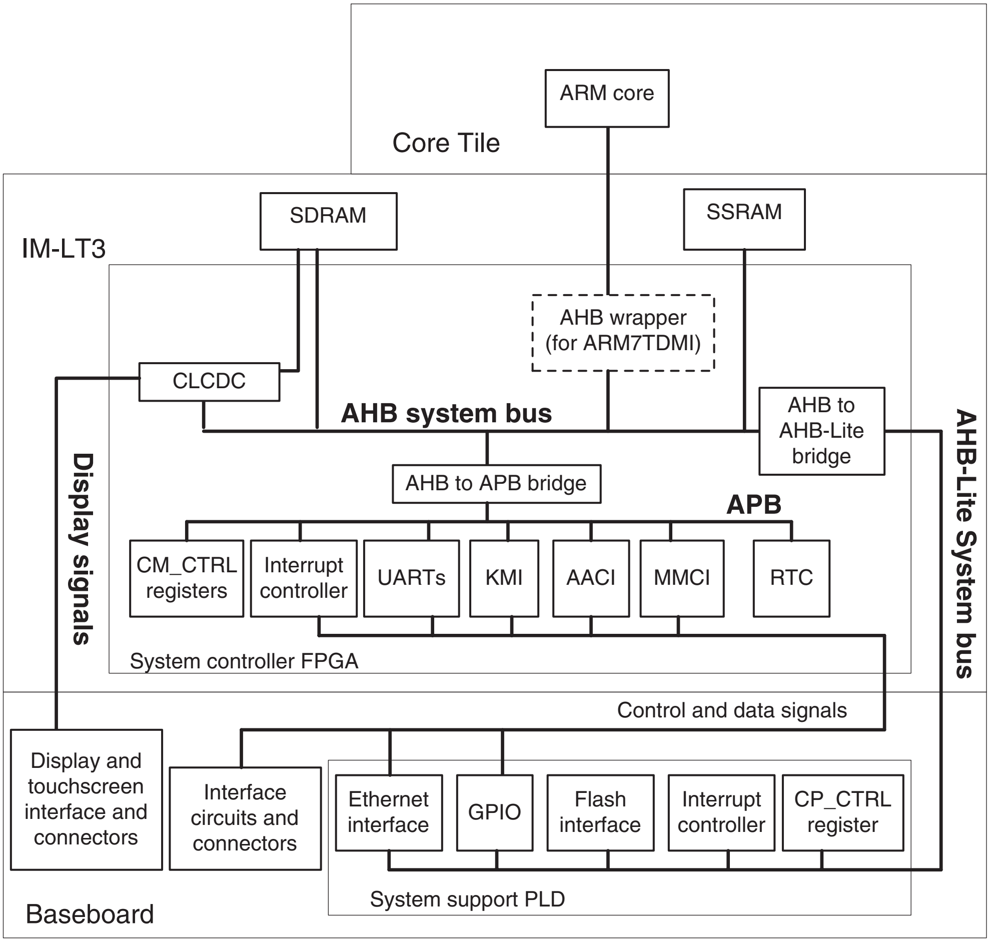


Figure C-2 Bus routing between CP baseboard, Core Tile, and IM-LT3

System bus routing and bus interfaces

The Integrator/CP, IM-LT3, and Core Tile use an AMBA system bus comprised of:

AHB system bus

This is the AHB bus within the IM-LT3 FPGA. The bus bridges in the IM-LT3 FPGA connect it to the AHB-Lite external system bus and to the APB peripheral bus.

For the CT926EJ-S or CT1136JF-S, the AHB bus from the Core Tile is directly connected to the AHB system bus in the FPGA. For the CT7TDMI, there is an AHB wrapper in the FPGA that translates between the native ARM7TDMI bus and the AHB processor bus.

AHB-Lite system bus

This is the external bus present on the HDRA and HDRB connectors. It connects the baseboard to the IM-LT3. The AHB-Lite system bus is a single-master bus with the processor core as sole bus master.

Peripheral bus

This is the APB bus present within the IM-LT3 FPGA. It connects the system bus to the APB peripherals and control registers.

APB peripheral bus

The APB is an AMBA-compliant bus optimized for minimum power and reduced interface complexity. It is used to interface peripherals, such as the UARTs and the *Keyboard and Mouse Interface* (KMI), that do not require the high performance of the AHB.

The AHB-APB bridge is an AHB slave that provides an interface between the high-speed AHB domain and the low-power APB domain. The APB is not pipelined. Wait states are added during transfers between the APB and AHB when the AHB is required to wait for the APB protocol.

Figure C-3 on page C-5 shows some example APB peripherals that are implemented using PrimeCell or ADK devices synthesized into the FPGA on the IM-LT3.

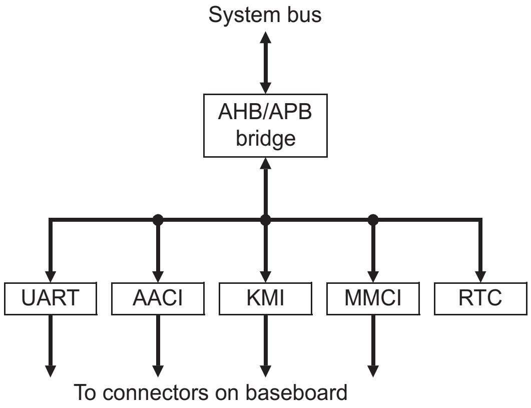


Figure C-3 APB peripherals

C.1.2 Configuring little or big-endian operation

The Integrator/CP can be configured to operate as a big-endian or little-endian system. To change to big-endian operation, write to the appropriate register in CP15 on the ARM core.

Note

The CP15 register is not present in the ARM7TDMI and systems CP systems using this processor must use a Core Tile signal to switch to big-endian mode.

The IM-LT3 FPGA image for use with the CT7TDMI is little-endian. The image must be rebuilt if big-endian operation is required. The big-endian image drives a signal to the Core Tile that selects big-endian mode.

There is a delay between changing the endian configuration of the core and the system functioning in the new endian mode. Changing endianess must only be done at the start of a debugging session. The change must have taken effect before you can perform any subword accesses.

Caution

The Integrator/CP display system and Ethernet controller cannot operate in big-endian mode.

C.2 Module ID selection and interrupt routing

The Integrator/CP can only have one master module (immediately on the baseboard). The Core Tile and IM-LT3 combination give a Core Module ID of 0.

——— **Note** ———

The signals **nPPRES[3:0]** (Core Module present) are used to signal the presence of modules to the central decoder. The IM-LT3 drives these signals as `b1110` to indicate that only one module is present.

The output from the secondary interrupt controller (SIC) is routed to the Primary Interrupt Controller (PIC) in the IM-LT3 FPGA. The PIC outputs directly drive the **nFIQ** and **nIRQ** inputs on the processor present on the Core Tile.

C.3 Top level memory map

Figure C-4 shows the top-level memory map of an Integrator/CP system. The memory map maintains compatibility with other platforms in the Integrator product family.

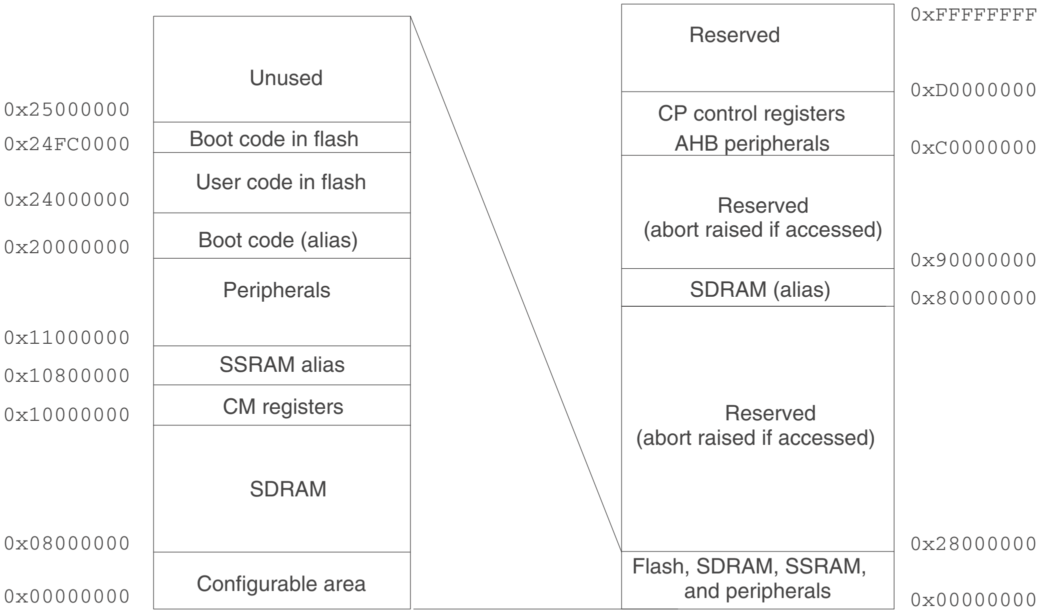


Figure C-4 Top-level memory map

The memory map contains areas for the SSRAM, SDRAM, flash memory, and peripherals. It also contains an area reserved for expansion at `0xD0000000–0xFFFFFFFF`.

Note

The processor cores on different boards might have different amounts of cache and TCM. The size of the cache and TCM on a board can be identified by reading registers in CP15.

C.3.1 Physical location of memory chips

Figure C-5 shows how the different memory types are physically located in an Integrator/CP system with a Core Tile and an IM-LT3.

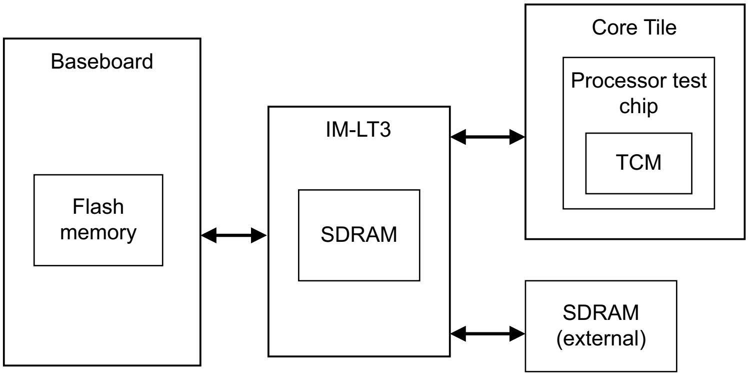


Figure C-5 Physical location of memory

C.3.2 Configurable area of memory map

The addressing of memory is partially configurable. Reasons for configuring memory addresses include:

- The flash memory is located at `0x24000000`. The memory area at `0x0` can be configured to be either RAM (REMAP =0) or flash (REMAP =1). Placing flash at `0x0` enables you to use boot code in the flash at startup, and placing RAM at `0x0` provides a mechanism for loading custom interrupt vectors and vector routines.

  Note

  The flash boot code area is at `0x24FC0000` and aliased at `0x20000000`.

- The ARM926EJ-S, the ARM1026EJ-S, and the ARM1136JF-S can have *Tightly-Coupled Memory* (TCM) located inside the chip. This provides very fast access for frequently used code or data, but the accesses are not visible on the address and data buses. Refer to the appropriate Technical Reference Manuals or Core Tile User Guides for details.

- Some applications might require large amounts of memory, or require more memory during the development phase. The IM-LT3 can be extended with SDRAM when the IM-LT3 RAM is not sufficient. An SDRAM DIMM is supplied with the IM-LT3.

Example memory map

The ARM926EJ-S and ARM1136JF-S cores have the following characteristics:

- The TCM location is configurable. Do not map the TCM to an area of memory that is used by control registers or memory-mapped IO. Also, the TCM I-RAM and TCM D-RAM regions must not overlap.

  Refer to the Technical Reference Manual for the processor core for details on using the CP15 registers to configure TCM and cache.

- The MMU can map physical memory to a different logical address.

Table C-1 shows a typical memory map with TCM present and enabled.

Table C-1 Example of 32KB TCM

Address range	Size	Description
`0x00000000–0x00007FFF`	32KB	SSRAM (REMAP=0) or flash (REMAP=1).
`0x00008000–0x0000FFFF`	32KB	TCM I-RAM (address can be assigned to any 32KB boundary) If TCM is disabled, this space is filled by SSRAM or flash.
`0x00010000–0x00017FFF`	32KB	SSRAM (REMAP=0) or flash (REMAP=1).
`0x00018000–0x0001FFFF`	32KB	TCM D-RAM (address can be assigned to any 32KB boundary) If TCM is disabled, this space is filled by SSRAM or flash.
`0x00020000–0x000FFFFF`	896KB	SSRAM (REMAP=0) or flash (REMAP=1).
`0x00100000–0x0FFFFFFF`	256MB	SDRAM (repeats physical memory to fill space) Part of the first SDRAM image is masked by the SSRAM image.
`0x10000000–0x107FFFFF`	8MB	Core Module registers.
`0x10800000–0x10FFFFFF`	8MB	On-board SSRAM alias (repeats physical memory to fill space).
`0x11000000–0x1FFFFFFF`	256MB	Peripherals
`0x20000000–0x23FFFFFF`	64MB	Boot code alias
`0x24000000–0x24FFFFFF`	16MB	Baseboard flash memory (contains user code and boot code).
`0x25000000–0x27FFFFFF`	48MB	Unused (larger sized baseboard flash devices occupy this area).
`0x28000000–0x7FFFFFFF`	2.3GB	Reserved (abort if accessed).
`0x80000000–0x8FFFFFFF`	256MB	SDRAM (alias).

Table C-1 Example of 32KB TCM (continued)

Address range	Size	Description
`0x90000000–0xBFFFFFFF`	2.3GB	Reserved (abort if accessed).
`0xC0000000–0xCFFFFFFF`	256MB	CP control registers and APB peripherals.
`0xD0000000–0xFFFFFFFF`	768MB	Reserved (abort if accessed).

Table C-2 shows a typical memory map for no TCM.

Table C-2 Example of TCM disabled or no TCM

Address range	Size	Description
`0x00000000–0x000FFFFF`	1MB	(REMAP=1) SSRAM or (REMAP=0) flash.
`0x00100000–0x0FFFFFFF`	256MB	SDRAM (repeats physical memory to fill space) Part of first SDRAM image is masked by SSRAM or flash.
`0x10000000–0x107FFFFF`	8MB	Core Module registers.
`0x10800000–0x10FFFFFF`	8MB	On-board SSRAM alias (repeats physical memory to fill space).
`0x11000000–0x1FFFFFFF`	256MB	Peripherals
`0x20000000–0x23FFFFFF`	64MB	Boot code alias
`0x24000000–0x24FFFFFF`	16MB	Baseboard flash memory (contains user code and boot code).
`0x25000000–0x27FFFFFF`	48MB	Unused (larger sized baseboard flash devices occupy this area).
`0x28000000–0x7FFFFFFF`	2.3GB	Reserved (abort if accessed).
`0x80000000–0x8FFFFFFF`	256MB	SDRAM (alias).
`0x90000000–0xBFFFFFFF`	2.3GB	Reserved (abort if accessed).
`0xC0000000–0xCFFFFFFF`	256MB	CP control registers and APB peripherals.
`0xD0000000–0xFFFFFFFF`	768MB	Reserved (abort if accessed).

C.3.3 Baseboard flash memory

The baseboard can be built with 16, 32, or 64MB of flash using 64 or 128Mb devices. (Typically, 16MB of flash is used.) Regardless of the size of flash memory, the top 256KB of the flash device is reserved for the system boot code. The remaining flash memory is available for your own code requirements See the *Integrator/CP User Guide* for more details on flash memory.

C.3.4 Memory timing

Accesses to memory require a different number of cycles depending on the type of memory as shown in Table C-3.

Table C-3 Wait states for memory access

Memory Type	Wait states
SSRAM read or write	0
SDRAM read typical (no bus conflict)	0–5
SDRAM read maximum (write in progress before read)	31
Flash read	6

C.4 Programmable logic

There are two main programmable logic parts on the Integrator/CP system. These are:

- *Baseboard system support PLD*
- *System controller FPGA*.

C.4.1 Baseboard system support PLD

The system support PLD is located on the baseboard. It provides AHB slave interfaces for the Ethernet and flash, an interrupt controller, and an 8-bit GPIO interface. Figure C-6 shows the internal architecture of the programmed PLD.

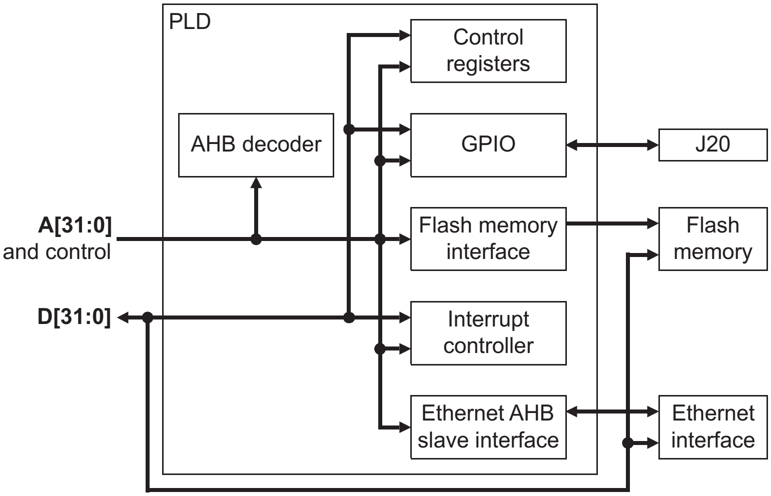


Figure C-6 Baseboard PLD

The PLD is programmed during board manufacture and is not normally reprogrammed in the field. However, if required for system upgrade, the PLD can be reprogrammed using the Multi-ICE interface. This requires the `progcards` utility (version 2.30 or later).

C.4.2 System controller FPGA

The system controller FPGA is located on the IM-LT3. It contains the main bus bridges, memory controllers, and peripheral controllers. Figure C-7 on page C-13 shows a simplified block diagram of a system controller FPGA.

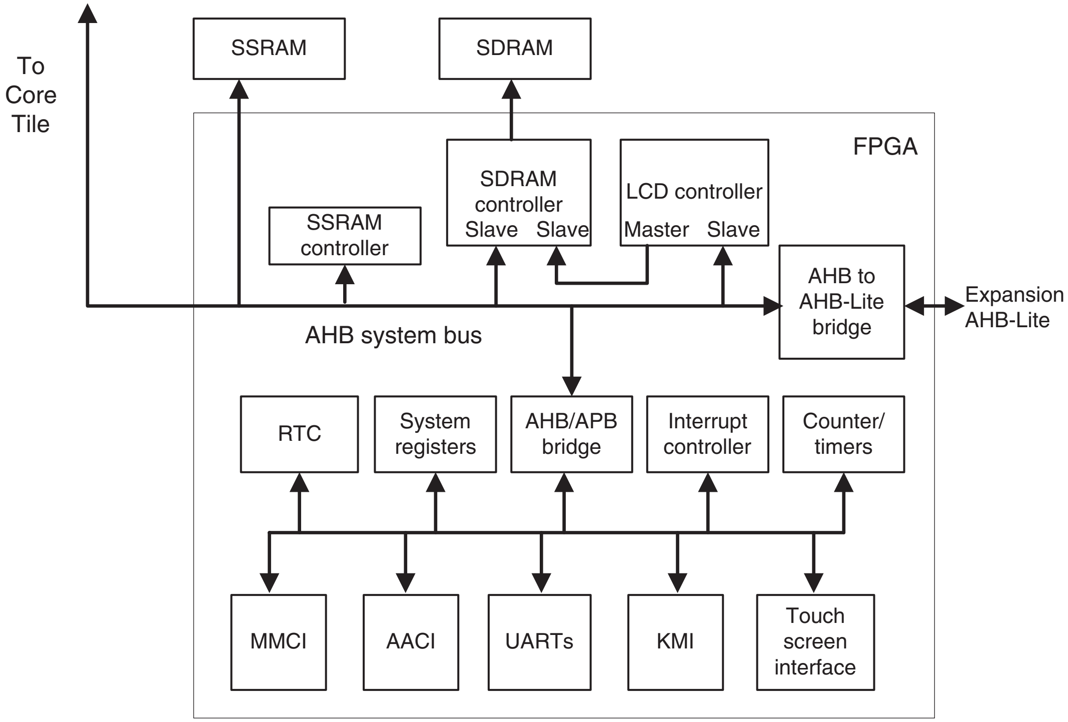


Figure C-7 System controller FPGA block diagram

——— **Note** ———

The IM-LT3 FPGA must have an appropriate Integrator/CP-compatible image in order to control the peripherals on the Integrator/CP baseboard. Images are supplied for current CT7TDMI, CT926EJ-S, and CT1136JF-S Core Tiles.

———————————

The FPGA on the IM-LT3, and on Logic Tiles added to the system, share the open-collector signal **GLOBAL_DONE**. When the system is powered on, the FPGA loads a configuration from the configuration flash memory on the IM-LT3 into the configuration inputs of the FPGA. The **GLOBAL_DONE** signal is used by the FPGAs to signal that configuration is complete and system reset can be completed.

You can use Multi-ICE to reprogram the PLD, FPGA, and flash when the system is placed in configuration mode.

The IM-LT3 configuration flash can store up to four images, but only two are used in the PLD image loaded at manufacture. The images are selected by SW4[1] as described in *PLD circuitry* on page 3-5. The **CFGSEL[1:0]** signals from the baseboardare not used to select the image.

C.4.3 HDL files

The IM-LT3 image provides the CP functional blocks as a set of HDL files. These are described in Table C-4.

Table C-4 IM-LT3 FPGA image functional FPGA block HDL file descriptions

Block	Description
`CPxx_xxx_AHBDecoder`	The decoder block provides the high-speed peripherals with select lines. The decoder blocks also contain the default slave peripheral to simplify the example structure. The Integrator family of boards uses a distributed address decoding system.
`CPAHBMUX7StoM`	This is the AHB multiplexor that connects the read data buses from all of the slaves to the AHB master(s).
`AHB2APB`	This is the bridge block required to connect APB peripherals to the high-speed AMBA AHB bus. It produces the peripheral select signals for each of the APB peripherals.
`CPxxx_xxx_regs`	The APB register peripheral provides memory-mapped registers that you can use to: • configure the two clock generators (protected by the LM_LOCK register) • write to the user LEDs • read the user switch inputs.
`CMIntcon`	The APB interrupt controller contains all of the standard interrupt controller registers and has an input port for four APB interrupts. (The example only uses one of them. The remaining three are set inactive in the AHBAPBSys block.) Four software interrupts are implemented.

C.5 Register and memory overview

Table C-5 shows a map for a typical Integrator/CP system. See the *Integrator/CP User Guide* for more details on register function.

Note

The memory map below `0x10000000` depends on the settings for TCM enabled and REMAP.

Table C-5 System control register map

Peripheral or device	Address range	Size
Boot flash. Mapped at this address only at power ON, and then disabled to allow access to SSRAM.	`0x00000000–0x000FFFFF`	1MB
SDRAM	`0x00100000–0x0FFFFFFF`	255MB
Control registers	`0x10000000-0x1000003F`	64 bytes
Interrupt controller **Note** The Integrator/CP are described in *Interrupt control* on page C-25.	`0x10000040–0x1000007F`	64 bytes
Reserved **Note** The CT7TDMI, CT926EJ-S, and CT1136JF-S use `0x10000080–0x1000008C` and `0x100000A0–0x100000AC` as voltage control registers. The CT926EJ-S also uses `0x10000094` for Core Tile PLD configuration. Refer to the documentation supplied with the Core Tiles for details.	`0x10000080–0x100000FF`	128 bytes
Serial Presence Detect memory	`0x10000100–0x100001FF`	256 bytes
Counter/timers	`0x13000000–0x13FFFFFF`	16MB
Primary interrupt controller registers	`0x14000000–0x14FFFFFF`	16MB

Table C-5 System control register map (continued)

Peripheral or device	Address range	Size
Real time clock	0x15000000–0x15FFFFFF	16MB
UART0	0x16000000–0x16FFFFFF	16MB
UART1	0x17000000–0x17FFFFFF	16MB
Keyboard	0x18000000–0x18FFFFFF	16MB
Mouse	0x19000000–0x19FFFFFF	16MB
Debug LEDs and DIP switch	0x1A000000–0x1AFFFFFF	16MB
Reserved	0x1B000000–0x1BFFFFFF	16MB
Multimedia Card Interface	0x1C000000–0x1CFFFFFF	16MB
Advanced Audio CODEC Interface	0x1D000000–0x1DFFFFFF	16MB
Touch Screen Controller Interface	0x1E000000–0x1EFFFFFF	16MB
Reserved	0x1F000000–0x1FFFFFFF	16MB
Flash (size depends on board device)	0x20000000–0x27FFFFFF	148MB
Reserved	0x28000000–0xBFFFFFFF	2416MB
CLCD regs/palette	0xC0000000–0xC0FFFFFF	16MB
Reserved	0xC1000000–0xC7FFFFFF	112MB
Ethernet	0xC8000000–0xC8FFFFFF	16MB
GPIO	0xC9000000–0xC9FFFFFF	16MB
Secondary interrupt controller	0xCA000000–0xCAFFFFFF	16MB
CP control registers	0xCB000000–0xCBFFFFFF	16MB
Reserved	0xCC000000–0xFFFFFFFF	832MB

Note

Device registers are usually mapped repeatedly to fill their assigned spaces. However, to ensure correct operation on future product versions, it is advisable to only access the register at its true address.

C.5.1 CM control registers for Integrator/CP

Table C-6 lists the CM register that are typically present on a Core Module. These registers are also implemented on the IM-LT3 when stacked on Integrator/CP. Refer to the Core Module documentation supplied on the CD provided with your IM-LT3 for more details on Core Module registers.

Note

Some registers might not function as expected on the IM-LT3 and Core Tile combination (CM_ID for example).

The CM_CTRL register is modified in the FPGA image used for the IM-LT3 on an Integrator/CP (see *Core Module control register, CM_CTRL* on page C-18).

Table C-6 Core Module status, control, and interrupt registers

Register Name	Address	Description
CM_ID	`0x10000000`	Core Module identification register
CM_PROC	`0x10000004`	Core Module processor register
CM_OSC	`0x10000008`	Core Module oscillator register
CM_CTRL	`0x1000000C`	Core Module control
CM_STAT	`0x10000010`	Core Module status
CM_LOCK	`0x10000014`	Core Module lock
CM_LMBUSCNT	`0x10000018`	Core Module local memory bus cycle counter
CM_AUXOSC	`0x1000001C`	Core Module auxiliary clock oscillator register
CM_SDRAM	`0x10000020`	SDRAM status and control register
CM_INIT	`0x10000024`	Core Module initialization register
CM_REFCT	`0x10000028`	Reference clock cycle counter
-	`0x1000002C`	Reserved

CM_xFLAGx	`0x10000030–`
	`0x1000003C`
		Core Module flag registers

Core Module control register, CM_CTRL

The Core Module and LCD control register (CM_CTRL) at `0x1000000C` is a read/write register that controls user-configurable features of the IM-LT3 and the display interface on the Integrator/CP baseboard. Refer to the Core Module documentation supplied on the CD provided with your IM-LT3 for more details on Core Module registers.

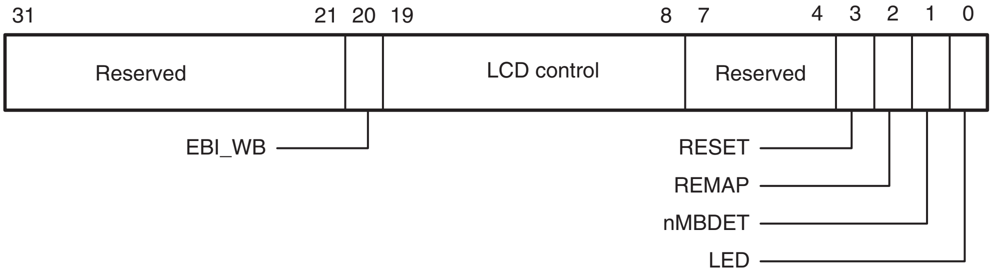


Figure C-8 CM_CTRL

Table C-7 and Figure C-8 describe the Core Module control register bits

Table C-7 CM_CTRL register

Bits	Name	Access	Function
[31:21]	Reserved	Use read-modify-write to preserve value.	
[20]	EBI_WP	Write	This bit controls the write power signal to the Core Module user flash in EBI[1] and EBI[2]. 0 = flash write protected 1= flash can be written to.
[19]	n24BITEN	Write	Select VGA depth: 0 = 24bit VGA 1 = 18bit VGA Must be 1 to enable BIAS control.
[18]	STATIC	Write	No connection on Sharp panel.
[17]	STATIC2	Write	Up/down axis flip on Sharp panel.
[16]	STATIC1	Write	Right/left axis flip on Sharp panel.
[15]	Enable LCD1	Write	Enable, active high.

Table C-7 CM_CTRL register (continued)

Bits	Name	Access	Function
[14]	Enable LCD0	Write	Enable, active high
[13:11]	LCDMUXSEL	Write	001 = Generic LCD connector, 24-bit mode 011 = Sharp LCD panel 100 = Sharp LCD panel 111 = 24-bit VGA
[10]	LCDBIASDN	Write	Low to high transition decreases LCD bias voltage (dimmer)
[9]	LCDBIASUP	Write	Low to high transition increases LCD bias voltage (brighter)
[8]	LCDBIASEN	Read/write	Enable LCD bias supply
[7:4]	Reserved	Use read-modify-write to preserve value.	
[3]	RESET	Write	This is used to reset the Core Tile, Integrator/CP, and IM-LT3. A reset is triggered by writing a 1. Reading this bit always returns a 0 allowing you to use read-modify-write operations without masking the RESET bit.
[2]	REMAP	Read/write	This bit is used to control REMAP: 0 = flash at address 0 1= SRAM at address 0.
[1]	nMBDET	Read	This bit indicates whether or not the IM-LT3 is mounted on a CP baseboard: 0 = mounted on baseboard 1 = no baseboard
[0]	LED	Read/write	This bit controls LED0 on the IM-LT3: 0 = LED OFF 1 = LED ON.

C.6 Peripherals and interfaces

This section provides a very brief description of the peripherals on the Integrator/CP:

- *Clock control*
- *Counter/timers*
- *Real-time clock*
- *UARTs* on page C-21
- *Keyboard and mouse interface* on page C-21
- *MMC interface* on page C-21
- *Audio interface* on page C-21
- *Touchscreen controller interface* on page C-22
- *Display interface* on page C-22
- *Ethernet interface* on page C-22.

For more detail, see the *Integrator/CP User Guide* and the documentation for the PrimeCell peripherals.

C.6.1 Clock control

The baseboard Micrologic IC525 clock generator provides:

- the 14.7456MHz UART clock.
- the 25MHz Ethernet clock.

A baseboard crystal oscillator provides a 12.288MHz AACI bit clock.

C.6.2 Counter/timers

The IM-LT3 FPGA provides three 32-bit counter/timers that can operate in three modes:

Free running

The timer counts down to zero and then wraps around and continues to count down from its maximum value.

Periodic The counter counts down to zero and then reloads the period value.

One shot The counter counts down to zero and does not reload a value.

C.6.3 Real-time clock

The *Real-Time Clock* (RTC) comprises the following elements:

- a 32-bit counter
- a 32-bit match register
- a 32-bit comparator.

The 32-bit counter increments on successive rising edges of a 1Hz clock generated by the IM-LT3 FPGA. You load a start value by writing to the load register RTC_LR and read the current value of the counter from the data register RTC_DR.

You program a match register by writing to the match register RTC_MR and you can read the current value at any time. When the counter and match register contents are identical, an interrupt request is asserted.

C.6.4 UARTs

The serial interface is implemented with two PrimeCell UARTs instantiated into the IM-LT3 FPGA. Transceivers and connectors for the serial interfaces are provided on the Integrator/CP baseboard and signals between the IM-LT3 and baseboard are routed through the HDRB connectors.

The UARTs are functionally similar to standard 16C550 devices.

For detailed information about the UART, see the *UART (PL011) Technical Reference Manual*.

C.6.5 Keyboard and mouse interface

The keyboard and mouse controllers are implemented with two PrimeCell *Keyboard and Mouse Interfaces* (KMIs) instantiated into the IM-LT3 FPGA. Connectors for these interfaces are provided on the baseboard and signals between the IM-LT3 and baseboard are routed through the HDRB connectors. The KMI generates an interrupt when a byte can be written or read from the data registers.

C.6.6 MMC interface

The PrimeCell *MultiMedia Card Interface* (MMCI) is instantiated into the IM-LT3 FPGA. A card connector is provided by the baseboard and the interface signals are routed between the IM-LT3 and baseboard using the HDRB connectors.

The MMCI is an APB peripheral that provides an interface between the MMC and the APB. For detailed information, see the *PrimeCell MultiMedia Card Interface (PL181) Technical Reference Manual*.

C.6.7 Audio interface

The baseboard provides a National Semiconductor LM4549 audio CODEC. The audio CODEC is compatible with AC’97 Rev 2.1 and features sample rate conversion and 3D sound. The CODEC is driven with a PrimeCell AACI (PL041) instantiated into the IM-LT3 FPGA and signals the IM-LT3 and baseboard are routed through the HDRB connectors.

Note

For a description of the audio CODEC signals, refer to the LM4549 datasheet available from National Semiconductor.

For detailed information on the AACI, see *ARM PrimeCell Advanced Audio CODEC Interface (PL041) Technical Reference Manual*.

C.6.8 Touchscreen controller interface

The touchscreen interface driven by the *TouchScreen Controller Interface* (TSCI) instantiated into the IM-LT3 FPGA and the baseboard provides external connectors. Interface signals are routed between the IM-LT3 and baseboard using the HDRB connectors.

C.6.9 Display interface

The touchscreen controller that accompanies the LCD is described in *Touchscreen controller interface*. See also the *ARM PrimeCell Color LCD Controller (PL110) Technical Reference Manual*.

The Integrator/CP provides a flexible display interface that provides support for two types of color LCD displays or a VGA display. The IM-LT3 provides a PrimeCell *Color LCD Controller* (CLCDC) instantiated into the FPGA. Display interface signals are routed using the B bus signals on the HDRA connectors.

CM_CTRL and LCD control bits

In addition to the PrimeCell LCD registers, there are some LCD control bits in the CM_CTRL register at `0x1000000C`. For more information, see *CM control registers for Integrator/CP* on page C-17.

C.6.10 Ethernet interface

The Ethernet interface is implemented with an SMC LAN91C111 10/100 Ethernet single-chip MAC and PHY on the baseboard. This is provided with a slave interface to the system bus by the FPGA on the baseboard. The Ethernet interface is described in the *Integrator/CP Baseboard User Guide*.

C.7 Reset controller

The IM-LT3 FPGA incorporates a reset controller that enables the Core Tile to be reset as a standalone unit or as part of an Integrator development system. The system can be reset from five sources:

- reset button
- motherboard
- other tiles
- JTAG debugger (RealView ICE or Multi-ICE)
- software.

Figure C-9 shows the architecture of the reset controller.

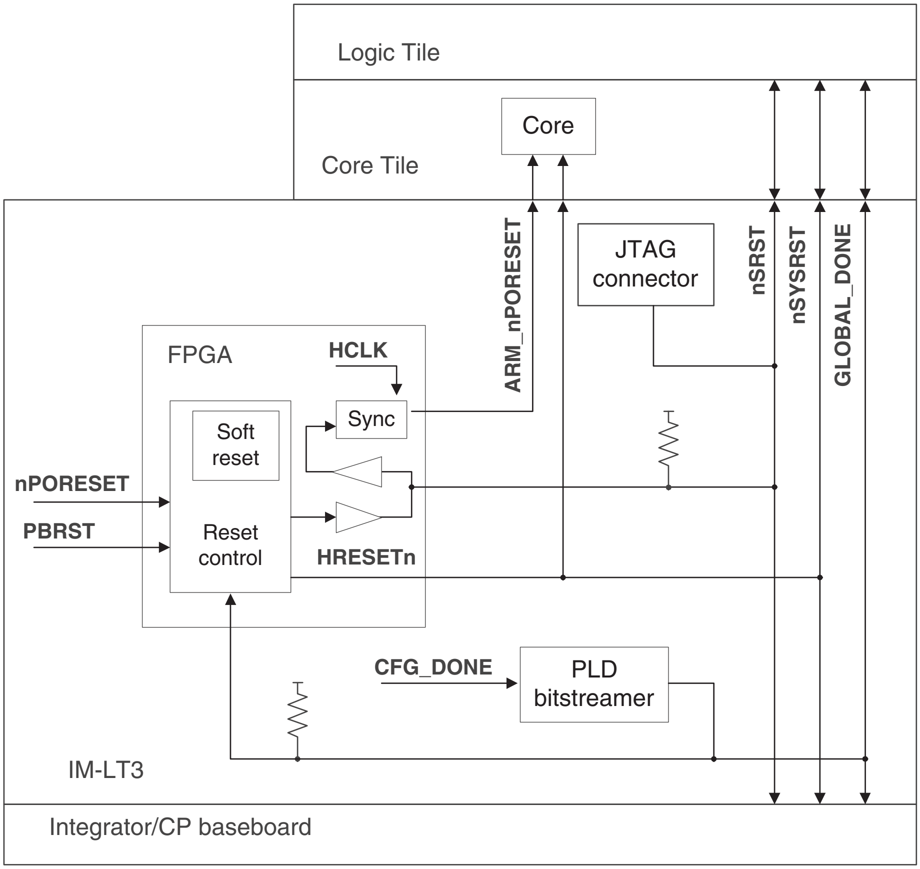


Figure C-9 IM-LT3 reset controller

C.7.1 Reset control signals

Table C-8 describes the external reset signals.

Table C-8 Reset signal descriptions

Name	Description	Type	Function
ARM_nPORESET	Processor reset	Output	The **ARM_nPORESET** signal is used to reset the processor core. It is generated from **nSRST** LOW when the Core Tile is used standalone, or **nSYSRST** LOW when the Core Tile is attached to a motherboard.
PBRST	Push-button reset	Input	The **PBRST** signal is generated by pressing the reset button.
nSRST	System reset	Bidirectional	The **nSRST** open collector output signal is driven LOW by the IM-LT3 FPGA when the signal **PBRST** or software reset (**SWRST**) is asserted. As an input, **nSRST** can be driven LOW by Multi-ICE. If there is no motherboard present, the **nSRST** signal is synchronized to the processor bus clock to generate the **ARM_nPORESET** signal.
nSYSRST	System reset	Output	The **nSYSRST** signal is generated by the system controller FPGA on the IM-LT3. It is used to generate **HRESETn** to the Core Tile and the system.

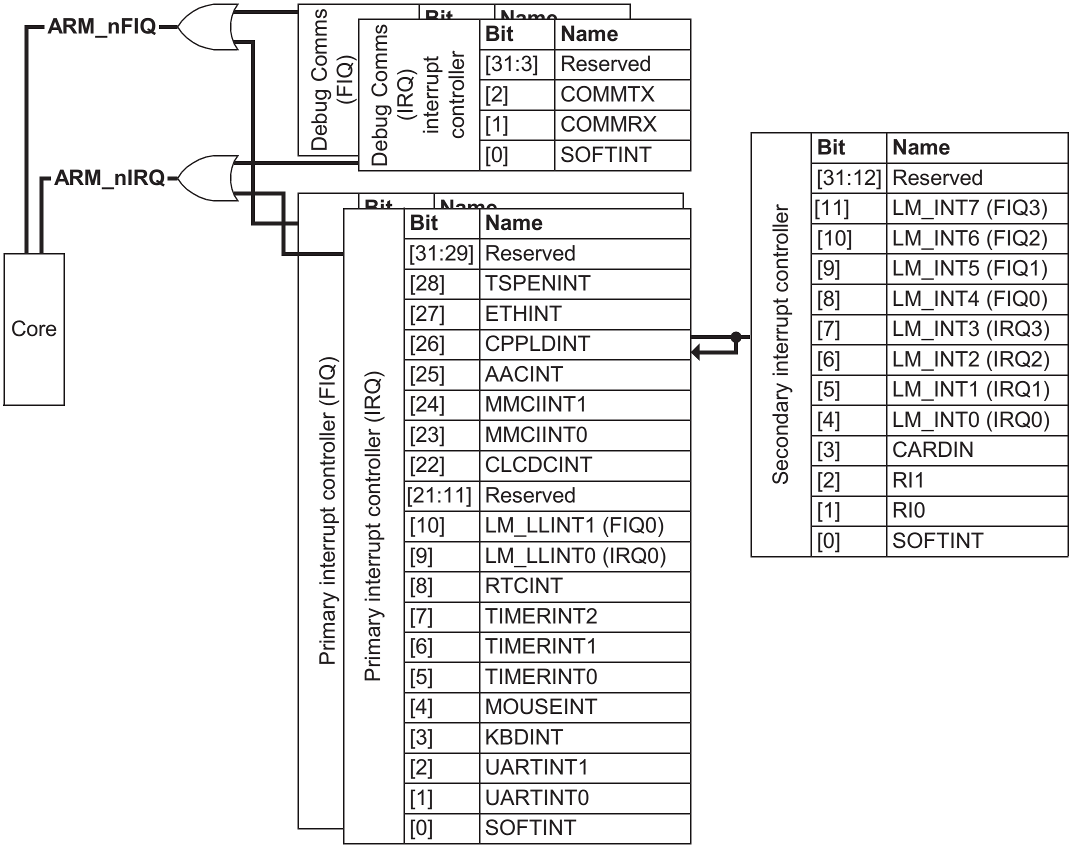
C.7.2 Software resets

The IM-LT3 FPGA provides a software reset that can be triggered by writing to the reset bit in the CM_CTRL register. This generates the internal reset signal **SWRST** that generates **nSRST** and resets the whole system.

C.8 Interrupt control

Figure C-10 shows the interrupt control architecture for the Integrator/CP system. The interrupts are described in the following sections:

- *Interrupt controllers* on page C-26
- *Interrupt routing* on page C-27
- *CP image interrupt control registers* on page C-29
- *Handling interrupts* on page C-33.



Debug Comms (IRQ) interrupt controller

Bit	Name
[31:3]	Reserved
[2]	COMMTX
[1]	COMMRX
[0]	SOFTINT

Primary interrupt controller (IRQ)

Bit	Name
[31:29]	Reserved
[28]	TSPENINT
[27]	ETHINT
[26]	CPPLDINT
[25]	AACINT
[24]	MMCIINT1
[23]	MMCIINT0
[22]	CLCDCINT
[21:11]	Reserved
[10]	LM_LLINT1 (FIQ0)
[9]	LM_LLINT0 (IRQ0)
[8]	RTCINT
[7]	TIMERINT2
[6]	TIMERINT1
[5]	TIMERINT0
[4]	MOUSEINT
[3]	KBDINT
[2]	UARTINT1
[1]	UARTINT0
[0]	SOFTINT

Secondary interrupt controller

Bit	Name
[31:12]	Reserved
[11]	LM_INT7 (FIQ3)
[10]	LM_INT6 (FIQ2)
[9]	LM_INT5 (FIQ1)
[8]	LM_INT4 (FIQ0)
[7]	LM_INT3 (IRQ3)
[6]	LM_INT2 (IRQ2)
[5]	LM_INT1 (IRQ1)
[4]	LM_INT0 (IRQ0)
[3]	CARDIN
[2]	RI1
[1]	RI0
[0]	SOFTINT

Figure C-10 Interrupt architecture (CP image)

C.8.1 Interrupt controllers

Integrator/CP system interrupts are generated by the *Primary Interrupt Controller* (PIC), the *Secondary Interrupt Controller* (SIC), and the *Communications Interrupt Controller* (CIC).

Detecting and clearing interrupts requires that each interrupt controller be correctly initialized as well as the interrupt control register in the individual peripheral.

Primary interrupt controller

The PIC is implemented within the IM-LT3 FPGA and handles the majority of interrupts from the system. A substantial number of interrupts are reserved to maintain compatibility with other modules within the Integrator family. The PIC provides a set of registers to control and handle interrupts. These are described in *Primary interrupt controller registers* on page C-29.

Secondary interrupt controller

The SIC is implemented in the baseboard PLD and combines interrupts from MMC socket, the UART ring indicator bits, and a software generated interrupt to the CPPLDINT input of the PIC.

The MMC interrupt on the SIC is generated by the card insertion detect switch and is different from the MMCI interrupts in the PIC generated by the MMCI PrimeCell.

The secondary controller provides a set of registers to control and handle interrupts. These are described in the *Integrator/CP Baseboard User Guide*.

Debug communications interrupts

The processor on the Core Tile incorporates EmbeddedICE hardware. This provides debug communications data read and write registers that are used to pass data between the processor and JTAG equipment. The **COMMTX** and **COMMRX** signals from the Core Tile are input to the debug comms interrupt controller. For a description of the debug communications channel, see the Technical Reference Manual for your core.

C.8.2 Interrupt routing

Figure C-11 shows a legacy CP system with Core Modules and Logic Modules. In this system the IRQ signals are rotated as they are routed from the modules to the interrupt controllers in the CP and Core Module.

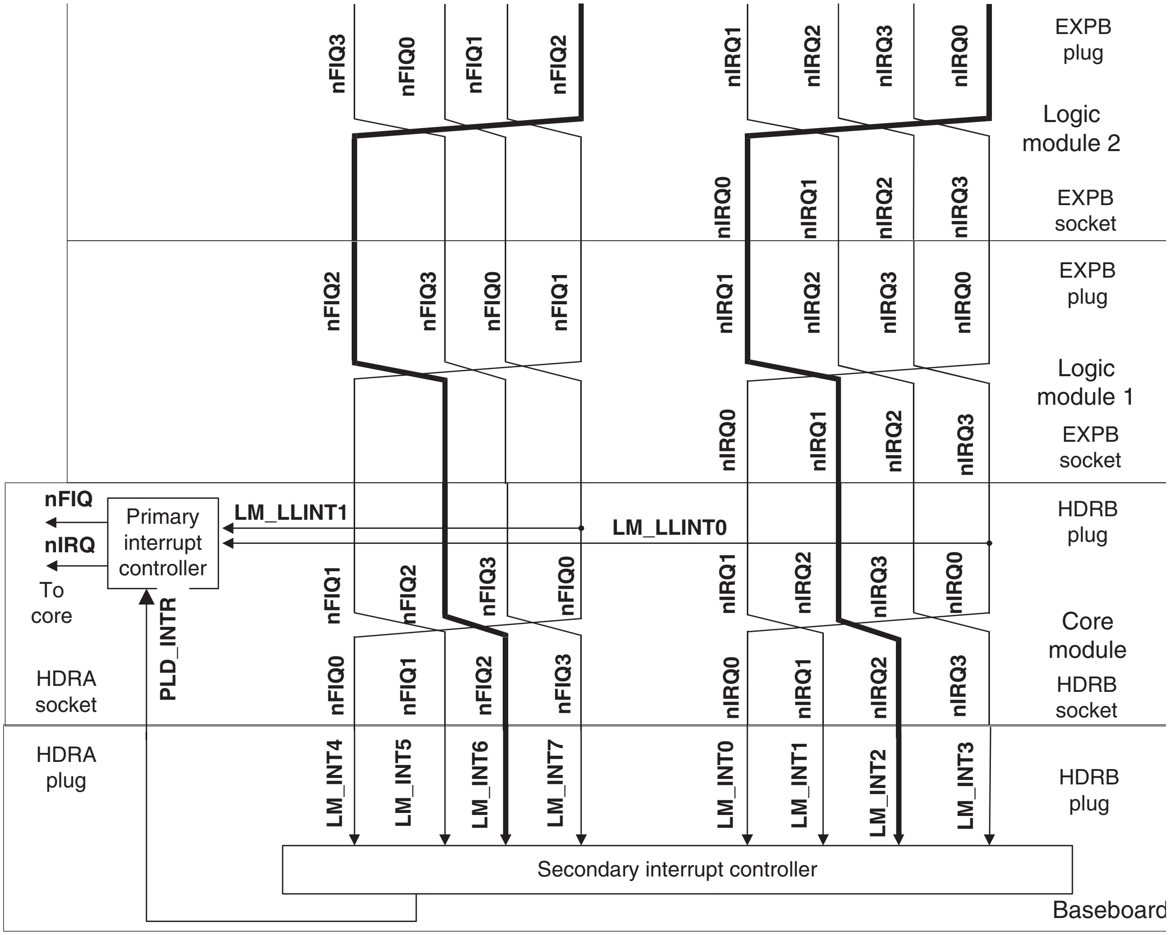


Figure C-11 Interrupt signal routing with Core Module

Figure C-11 highlights how the signals are routed so that, for example, the interrupt request from logic module 2 connects to **LM_INT2** (**IRQ[2]**) on the baseboard. A similar routing applies for logic module 1 and 3. The operation of the interrupts relies on the Core Module being mounted on the baseboard first with any logic modules on top.

Figure C-12 shows how interrupt signals are routed for a system consisting of an Integrator/CP, IM-LT3, Core Tile, and Logic Tile. The rotation of interrupt signals, if used, must be done in the Logic Tile FPGAs.

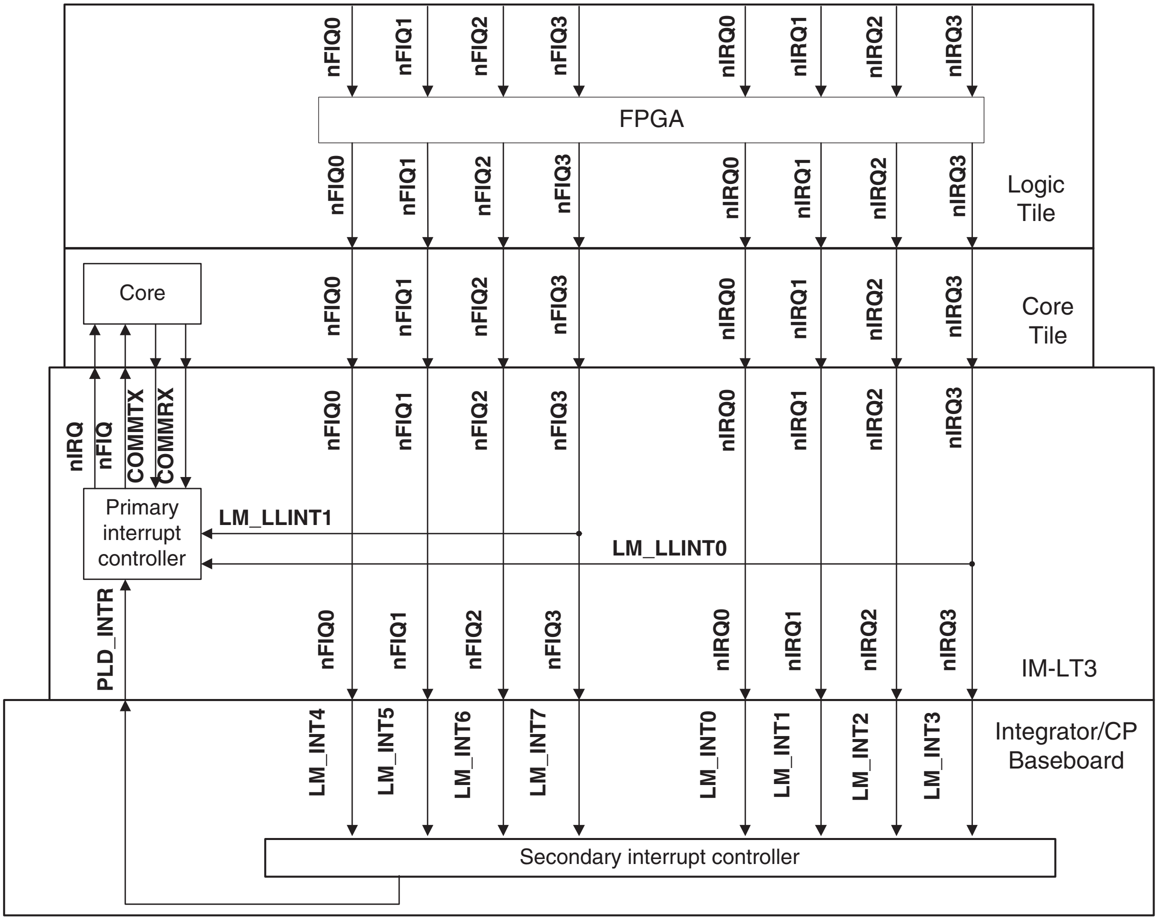


Figure C-12 Interrupt signal routing for tiles

The signals route through the SIC. Determining the source of an interrupt requires interrogating first the primary and then the secondary controller. The time required for the extra instructions might cause a problem with interrupt latency in some situations. To improve latency, **FIQ[0]** and **IRQ[0]** (as **LM_LLINT[1:0]**) are also routed to the PIC as well.

C.8.3 CP image interrupt control registers

The baseboard FPGA provides an interrupt controller to handle IRQs and FIQs from around the system. These are in addition to the primary IM-LT3 interrupt controller, and are described in:

- *Primary interrupt controller registers*
- *Debug communications interrupt registers* on page C-31
- *Secondary interrupt controller registers* on page C-31
- *Soft interrupt set and soft interrupt clear registers* on page C-32.

Primary interrupt controller registers

The CP image for the FPGA provides interrupt controllers for both IRQs and FIQs that maintain compatibility with other Integrator modules to ensure code portability. The primary interrupt control registers are listed in Table C-9

Table C-9 Primary interrupt register addresses

Address	Name	Type	Size	Function
`0x14000000`	PIC_IRQ_STATUS	Read	22	IRQ gated interrupt status
`0x14000004`	PIC_IRQ_RAWSTAT	Read	22	IRQ raw interrupt status
`0x14000008`	PIC_IRQ_ENABLESET	Read/write	22	IRQ enable set
`0x1400000C`	PIC_IRQ_ENABLECLR	Write	22	IRQ enable clear
`0x14000010`	PIC_INT_SOFTSET	Read/write	16	Software interrupt set
`0x14000014`	PIC_INT_SOFTCLR	Write	16	Software interrupt clear
`0x14000020`	PIC_FIQ_STATUS	Read-only	22	FIQ gated interrupt status
`0x14000024`	PIC_FIQ_RAWSTAT	Read-only	22	FIQ raw interrupt status
`0x14000028`	PIC_FIQ_ENABLESET	Read/write	22	FIQ enable set
`0x1400002C`	PIC_FIQ_ENABLECLR	Write-only	22	FIQ enable clear

The bit assignment for interrupts in the status, raw status, and enable registers for the IRQ and FIQ interrupt controllers is similar and is shown in Table C-10.

Table C-10 Primary interrupt register bit assignments

Bit	Name	Function
[31:29]	-	Reserved
[28]	TS_PENINT	Touchscreen pen-down event interrupt
[27]	ETH_INT	Ethernet interface interrupt
[26]	CPPLDINT	Interrupt from secondary interrupt controller, see *Secondary interrupt controller registers* on page C-31.
[25]	AACIINT	Audio interface interrupt
[24]	MMCIINT1	MultiMedia card interface
[23]	MMCIINT0	MultiMedia card interface
[22]	CLCDCINT	Display controller interrupt
[21:11]	-	Reserved
[10]	LM_LLINT1	Logic module low-latency interrupt 1(from Logic Tiles mounted above the IM-LT3)
[9]	LM_LLINT0	Logic module low-latency interrupt 0 (from Logic Tiles mounted above the IM-LT3)
[8]	RTCINT	Real time clock interrupt
[7]	TIMERINT2	Counter-timer 2 interrupt
[6]	TIMERINT1	Counter-timer 1 interrupt
[5]	TIMERINT0	Counter-timer 0 interrupt
[4]	MOUSEINT	Mouse interrupt
[3]	KBDINT	Keyboard interrupt
[2]	UARTINT1	UART 1 interrupt
[1]	UARTINT0	UART 0 interrupt
[0]	SOFTINT	Software interrupt

Debug communications interrupt registers

The bit assignments for the IRQ and FIQ status, raw status and enable register are shown in Table C-11.

Table C-11 IRQ and FIQ register bit assignment

Bit	Name	Function
[31:3]	Reserved	Write as 0. Reads undefined.
[2]	COMMTx	Debug communications transmit interrupt. This interrupt indicates that the communications channel is available for the processor to pass messages to the debugger.
[1]	COMMRx	Debug communications receive interrupt. This interrupt indicates to the processor that messages are available for the processor to read.
[0]	SOFT	Software interrupt.

Secondary interrupt controller registers

The secondary interrupt control registers are listed in Table C-12.

Table C-12 Secondary interrupt register addresses

Address	Name	Type	Size	Function
`0xCA000000`	SIC_IRQ_STATUS	Read	22	SIC gated interrupt status
`0xCA000004`	SIC_IRQ_RAWSTAT	Read	22	SIC raw interrupt status
`0xCA000008`	SIC_IRQ_ENABLESET	Read/write	22	SIC enable set
`0xCA00000C`	SIC_IRQ_ENABLECLR	Write	22	SIC enable clear
`0xCA000010`	SIC_INT_SOFTSET	Read/write	16	Software interrupt set
`0xCA000014`	SIC_INT_SOFTCLR	Write	16	Software interrupt clear

For more details, see the *Integrator/CP Baseboard User Guide*.

Soft interrupt set and soft interrupt clear registers

The primary, secondary, and comms interrupt controllers provide a register for controlling and clearing software interrupts.

This register is accessed using the software interrupt set and software interrupt clear locations. The set and clear locations are used as follows:

- Set the software interrupt by writing to the CM_SOFT_INTSET location:
  - write a 1 to SET the software interrupt
  - write a 0 to leave the software interrupt unchanged.
- Read the current state of the software interrupt register from the CM_SOFT_INTSET location. A bit set to 1 indicates that the corresponding interrupt request is active.
- Clear the software interrupt by writing to the CM_SOFT_INTCLR location:
  - write a 1 to CLEAR the software interrupt
  - write a 0 to leave the software interrupt unchanged.

The bit assignment for the software interrupt register is shown in Table C-13.

Table C-13 IRQ register bit assignment

Bit	Name	Function
`[31:1]`	Reserved	Write as 0. Reads undefined.
`[0]`	SOFT	Software interrupt

Note

The *software interrupt* described in this section is used by software to generate IRQs or FIQs. It must not be confused with the ARM SWI software interrupt instruction. See the *ARM Architecture Reference Manual*.

Refer to the documentation supplied with your Core Module for details on Core Module registers.

C.8.4 Handling interrupts

This section describes interrupt handling and clearing in general. For examples of interrupt detection and handling, see the selftest example software on the CD, the *ARM Firmware Suite User Guide*, *RealView Compilation Tools Developer Guide* (or the *ARM Developer Suite Developer Guide*), and the technical reference manual for your processor.

Enabling IRQ interrupts

The majority of peripheral interrupts are routed direct to the PIC. Each peripheral contains its own interrupt mask and clear registers. To enable interrupts, you must clear both the peripheral interrupt mask and the interrupt controller mask as well as clearing any previous interrupt flags:

1. Disable the primary interrupt by setting the appropriate bit in PIC_IRQ_ENCLR.
2. Clear the peripheral interrupt by setting the appropriate bit in the peripheral interrupt clear register.
3. Unmask the peripheral interrupt by clearing the appropriate bit in peripheral interrupt mask register.
4. Enable the primary interrupt by setting the appropriate bit in PIC_IRQ_ENSET.

The following C code stub demonstrates how the PIC UART0 CTS interrupt is cleared and re-enabled:

```
*PIC_IRQ_ENCLR    = PIC_UARTINT0;
*UART0_UARTICR    = UART_CTSINTR;
*UART0_UARTIMSC   &= ~UART_CTSINTR;
*PIC_IRQ_ENSET    |= PIC_UARTINT0;
```

The following C code stub demonstrates how the SIC MMCI CARDIN is cleared and re-enabled:

```
*PIC_IRQ_ENCLR   = PIC_CPPLDINT;
*CP_INTREG       = SIC_CARDIN;
*SIC_IRQ_ENSET   |= SIC_CARDIN;
*PIC_IRQ_ENSET   |= PIC_ CPPLDINT;
```

Note

The constants in these C code stubs must contain bit patterns necessary to set only the required interrupt mask bits. For example, `PIC_UARTINT0` must contain `0x02` to set only the UART0 bit in the PIC_IRQ_ENCLR register.

Determining and clearing IRQ interrupts

To determine an interrupt source, read the STATUS registers in the PIC and CIC to determine the interrupt controller that generated the interrupt. The sequence to determine and clear the interrupt is:

1. Determine the interrupt source by reading CM_IRQ_STATUS and PIC_STATUS.

   The interrupt handler is directed by the status register information to the particular peripheral that generated the interrupt. In the case of SIC interrupts the interrupt handler must also read the SIC STATUS register to determine the interrupt source.

2. Determine the peripheral interrupt source by reading the peripheral masked interrupt status register.

3. Clear the peripheral interrupt by setting the appropriate bit in the peripheral interrupt clear register.

The following pseudo code example demonstrates how the UART0 CTS interrupt is detected:

```
If CM_IRQ_STATUS flags set,
      . . .  CIC interrupt handler

If PIC_STATUS flags set,
      . . . PIC interrupt handler
     If PIC_CP_PLDINTR set,
           . . .  SIC output from CP active, handle SIC interrupt
     If PIC_UARTINT0 set,
           . . . UART0 interrupt handler
           If UART0_UARTMIS, UART_CTSINTR flag set,
                   . . . act on interrupt then clear flag with UARTICR
     . . . Test other PIC flags
```

Index