Core Tile for ARM® Cortex™-R4F

HBI-0196

User Guide

ARM®

ARM DUI 0441C (ID021412)

Core Tile for ARM Cortex-R4F
User Guide

Release Information

The following changes have been made to this book.

Change History

Date	Issue	Confidentiality	Change
9 April 2009	A	Non-Confidential	First release
06 July 2010	B	Non-Confidential	Document update
20 April 2011	C	Non-Confidential	Document update

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.

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

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

CE Declaration of Conformity

The system should be powered down when not in use.

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

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

Note

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

Contents
Core Tile for ARM Cortex-R4F User Guide

Preface

Chapter 1 Introduction

Chapter 2 Getting Started

Chapter 3 CT-R4F Hardware Description

List of Tables
Core Tile for ARM Cortex-R4F User Guide

List of Figures
Core Tile for ARM Cortex-R4F User Guide

Preface

This preface introduces the Core Tile for ARM Cortex-R4F (CT-R4F) and its reference documentation. It contains the following sections:

- *About this book* on page xii
- *Feedback* on page xvii.

About this book

This book describes how to set up and use the Core Tile for ARM Cortex-R4F (CT-R4F).

Intended audience

This book has been written for experienced hardware and software developers to aid the development of ARM-based products using the CT-R4F as part of a development system. You can use the CT-R4F to:

- enable benchmarking and software development with a fast ARM Cortex-R4F processor
- enable hardware prototyping of an ARM Cortex-R4F system operating at close to ASIC speeds.

Using this book

This book is organized into the following chapters:

Chapter 1 ***Introduction***

This chapter introduces Core Tiles and the CT-R4F.

Chapter 2 ***Getting Started***

This chapter describes how to set up and prepare the CT-R4F for use.

Chapter 3 ***CT-R4F Hardware Description***

This chapter describes the on-board hardware of the CT-R4F.

Chapter 4 ***Programmer's Reference***

This chapter describes the memory map and the configuration registers for the peripherals in the CT-R4F.

Chapter 5 ***Signal Descriptions***

This chapter provides a summary of signals present on the CT-R4F connectors.

Appendix A ***Specifications***

This appendix contains the electrical, timing, and mechanical specifications for the CT-R4F.

Typographical conventions

The following typographical conventions are used in this document:

bold Highlights ARM signal names within text, and interface elements such as menu names. This style is also used for emphasis in descriptive lists where appropriate.

italic Highlights special terminology, cross-references and citations.

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

`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 or functions where the argument is to be replaced by a specific value.

`monospace bold` Denotes language keywords when used outside example code.

Other conventions

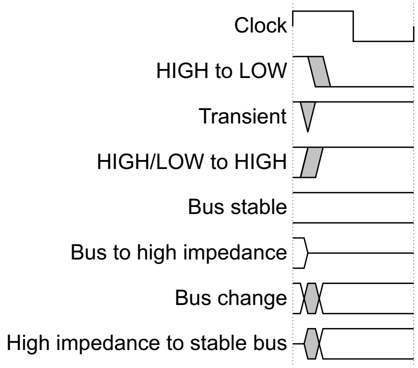
This document uses other conventions. They are described in the following sections:

- *Timing diagrams*
- *Signals* on page xiv
- *Bytes, Halfwords, and Words* on page xv
- *Bits, bytes, K, and M* on page xv
- *Register fields* on page xv.
- *Numbering* on page xvi.

Timing diagrams

The figure named *Key to timing diagram conventions* on page xiv explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.



Key to timing diagram conventions

Signals

When a signal is described as being asserted, the level depends on whether the signal is active HIGH or active LOW. Asserted means HIGH for active high signals and LOW for active low signals:

Prefix n Active LOW signals are prefixed by a lowercase n except in the case of AXI, AHB or APB reset signals. These are named **ARESETn**, **HRESETn** and **PRESETn** respectively.

Prefix A Denotes global *Advanced eXtensible Interface* (AXI) signals:

Prefix AR Denotes AXI read address channel signals.

Prefix AW Denotes AXI write address channel signals.

Prefix B Denotes AXI write response channel signals.

Prefix C Denotes AXI low-power interface signals.

Prefix H AHB signals are prefixed by an upper case H.

Prefix P APB signals are prefixed by an upper case P.

Prefix R Denotes AXI read data channel signals.

Prefix W Denotes AXI write data channel signals.

Bytes, Halfwords, and Words

Byte Eight bits.

Halfword Two bytes (16 bits).

Word Four bytes (32 bits).

Quadword 16 contiguous bytes (128 bits).

Bits, bytes, K, and M

Suffix b Indicates bits.

Suffix B Indicates bytes.

Suffix K When used to indicate an amount of memory means 1024. When used to indicate a frequency means 1000.

Suffix M When used to indicate an amount of memory means 1024^2 = 1 048 576. When used to indicate a frequency means 1 000 000.

Register fields

All reserved or unused address locations must not be accessed as this can result in unpredictable behavior of the device.

All reserved or unused bits of registers must be written as zero, and ignored on read unless otherwise stated in the relevant text.

All registers bits are reset to logic 0 by a system reset unless otherwise stated in the relevant text.

Unless otherwise stated in the relevant text, all registers support read and write accesses. A write updates the contents of the register and a read returns the contents of the register.

All registers defined in this document can only be accessed using word reads and word writes, unless otherwise stated in the relevant text.

Numbering

The document numbering convention is:

<base><number>

For example:

- `0x7B4` is a hexadecimal value of 7B4
- 9 is a decimal value of 9
- b00001111 is an eight-bit binary value of 00001111.

Further reading

The following documents provide information on the system components and interconnect in the ARM Cortex-R4F test chip:

- *ARM Cortex R4 and Cortex R4F Processor Technical Reference Manual* (ARM DDI 0363)
- *PrimeCell™ Color LCD Controller (PL111) Technical Reference Manual* (ARM DDI 0293)
- *PrimeCell DMA Controller (PL330) Technical Reference Manual* (ARM DDI 0424)
- *PrimeCell Dynamic Memory Controller (PL340) Technical Reference Manual* (ARM DDI 0331)
- *PrimeCell Vectored Interrupt Controller (PL192) Technical Reference Manual* (ARM DDI 0273)
- *PrimeCell High-Performance Matrix (PL301) Technical Reference Manual* (ARM DDI 0397)
- *AMBA Specification* (ARM IHI 0011)
- *AMBA 3 AXI Protocol Specification* (ARM IHI 0022).

The following documents provide information on the CoreSight™ system components in the ARM Cortex-R4F test chip:

- *CoreSight Technology System Design Guide* (ARM DGI 0012)
- *CoreSight Components Technical Reference Manual* (ARM DDI 0314)
- *CoreSight ETM™-R4 Technical Reference Manual* (ARM DDI 0367).

The following documents provide information on using the CT-R4F with an *Emulation Baseboard* (EB) and Logic Tile:

- *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411)
- *RealView LT-XC5VLX330 User Guide* (ARM DUI 0365)
- *ARM RealView AT1 Analyzer Tile User Guide* (ARM DUI 0189)
- *Versatile/IT1 Interface Tile User Guide* (ARM DUI 0188).

Feedback

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

Feedback on this document

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

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

General suggestions for additions and improvements are also welcome.

Feedback on the ARM Core Tiles

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

- the product name
- an explanation of your comments.

Chapter 1
Introduction

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

- *Precautions* on page 1-2.
- *About Core Tiles* on page 1-3
- *Overview of CT-R4F* on page 1-4.

Note

This guide covers the HBI-0196 (CT-R4F) printed-circuit board.

This board supports the ARM Cortex-R4F test chip. The availability of Core Tiles however, depends on the availability of individual test chips. Contact your sales representatives for details on currently available Core Tiles.

1.1 Precautions

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

1.1.1 Ensuring safety

The CT-R4F is powered from 3V3 and 5V supplies provided by the baseboard.

Warning

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

The test chip on the Core Tile can become very hot during continuous I/O activity. Take extra care when probing the test chip.

1.1.2 Preventing damage

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

Caution

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

- never subject the Core Tile to high electrostatic potentials
- always wear a grounding strap when handling the Core Tile
- only hold the Core Tile by the edges
- avoid touching the component pins or any other metallic element.

Do not use the board near equipment that could be:

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

1.2 About Core Tiles

Core Tiles are development boards that enable you to develop products based on ARM processors and AMBA® interfaces. Core Tiles are built around test chips, which are ASIC implementations of one or more ARM processors. Core Tiles provide one or more AMBA interfaces from the processor so that it can be connected to an AMBA-based system. See the *AMBA Specification (Rev 2.0)* (ARM IHI 0011) and the *AMBA AXI™ Protocol Specification* (ARM IHI 0022) for further information.

The Core Tile must be used in conjunction with a baseboard that implements the necessary system and memory controllers. For example, the CT-R4F when combined with the RealView® *Emulation Baseboard* (EB) provides a standalone system for ARM Cortex-R4F product development. Third-party or custom baseboards can also be used.

Note

The CT-R4F, like other Core Tiles, is only fully supported by ARM when used in combination with an EB, if a third-party or custom baseboard is used, ARM can provide limited support. Use of Core Tiles with a *Platform Baseboard* (PB) is not supported.

Core Tiles do not have power connectors. The tiles must be stacked on a baseboard that provides a power connection. The Core Tiles also require a reference clock (or clocks) to be supplied by the attached baseboard.

Note

Most Core Tiles require the EB or custom baseboard to provide the JTAG port connector. The CT-R4F also provides the JTAG port at the TRACE A connector. The JTAG port to be used is determined at reset.

Through-board connectors on most tile products enable stacking of multiple tiles. Multiple combinations of Core Tile and Logic Tile can be used to create a multiprocessor system.

Note

The CT-R4F does not include upper board connectors and therefore can only be used at the top of a tile stack.

1.3 Overview of CT-R4F

Figure 1-1 shows the layout of a CT-R4F.

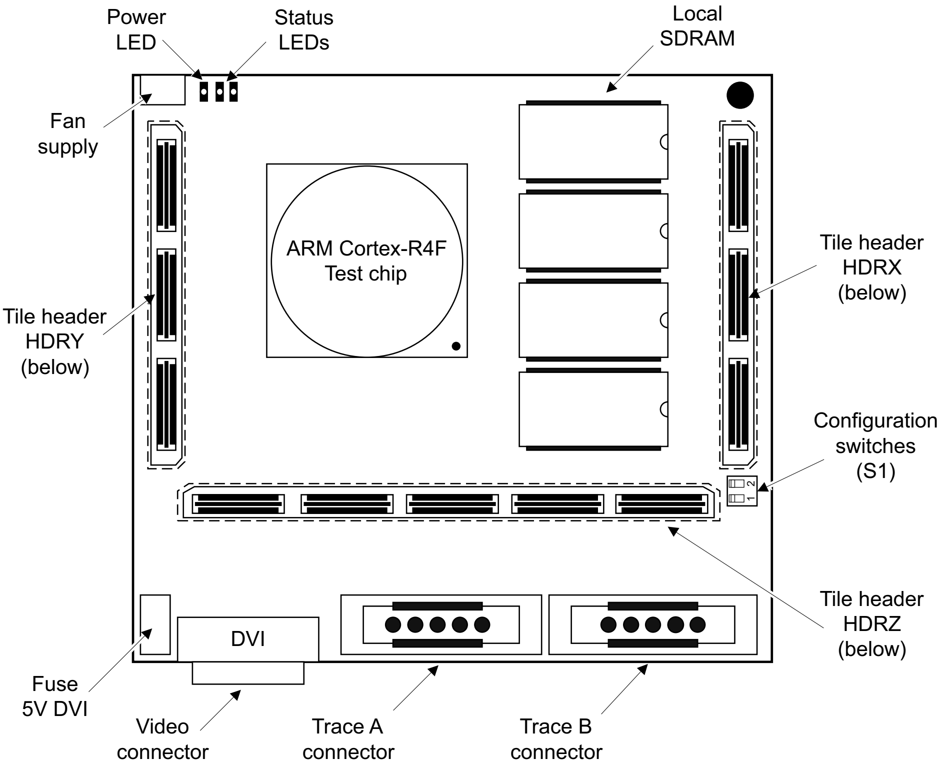


Figure 1-1 CT-R4F layout

The major components on the CT-R4F are:

- ARM Cortex-R4F test chip
- 512MB (64M x 64) of local SDRAM memory
- clock selection and initialization logic
- configuration and control PLD with 4-wire serial interface
- ARM Cortex-R4F test chip signal routing
- bus interface logic
- bottom tile headers for connection to a baseboard or Logic Tile
- DVI video interface with analog VGA support.
- Configuration switches and status LEDs
- power supplies for the test chip core, PLL, and CT-R4F PLD core

- test chip current and voltage monitoring logic.

Figure 1-2 on page 1-6 shows a CT-R4F fitted to tile site 1on an EB. Typically, tile site 2 is occupied by a Logic Tile that can be used for custom peripheral development or to hold a synthesized core. See the *LT-XC5VLX330 Logic Tile User Guide* (ARM DUI 0365) for details on a suitable Logic Tile that can be used for system prototyping.

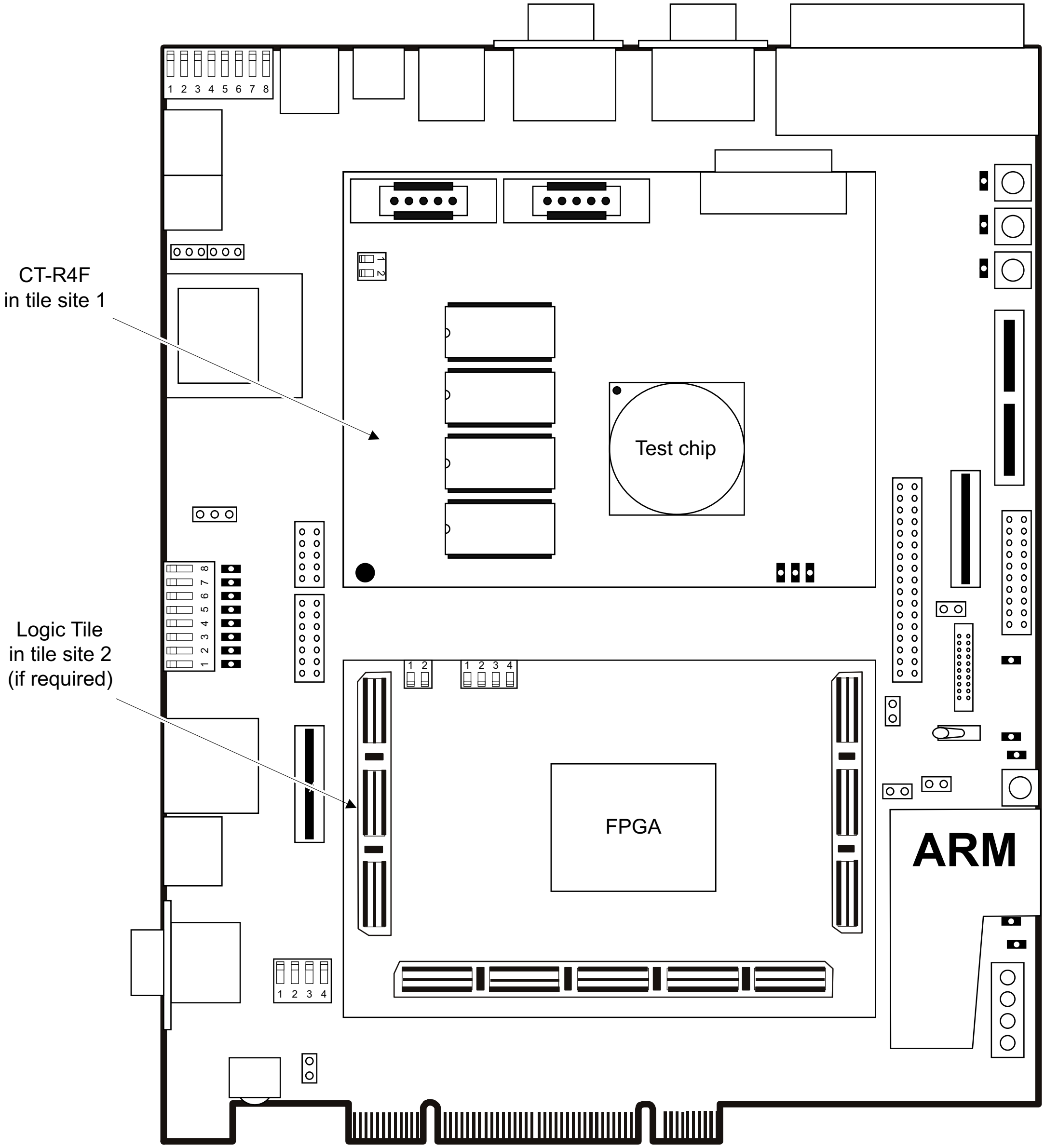


Figure 1-2 CT-R4F and Logic Tile on EB

1.3.1 System architecture

Figure 1-3 on page 1-8 shows the architecture of an ARM Cortex-R4F development system consisting of a CT-R4F and an EB. For details of the peripheral devices implemented on the EB and the available interfaces see *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411).

1.3.2 External logic

The EB provides:

- Power and JTAG connectors.
- A reference clock.
- A serial interface to the CT-R4F PLD that loads the required configuration and the initialization values for the PLL and clock divider in the test chip.
- Peripheral devices (for example, memory controller, interrupt controller, DMA controller, system and reset controllers, serial I/O).
  See *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411) for details.

The external interface connectors, control logic, and system control functions for the CT-R4F are also provided by the EB or an attached Logic Tile. See Chapter 3 *CT-R4F Hardware Description*, and the documentation for the EB for more details.

Warning

The FPGAs on the EB and an attached Logic Tile must be loaded with an appropriate image. An application note giving information on the FPGA images for the combination of products that you are using might be available. See the documentation provided on the *Versatile CD* supplied with the product. Loading an incorrect image can seriously damage the tiles and baseboard.

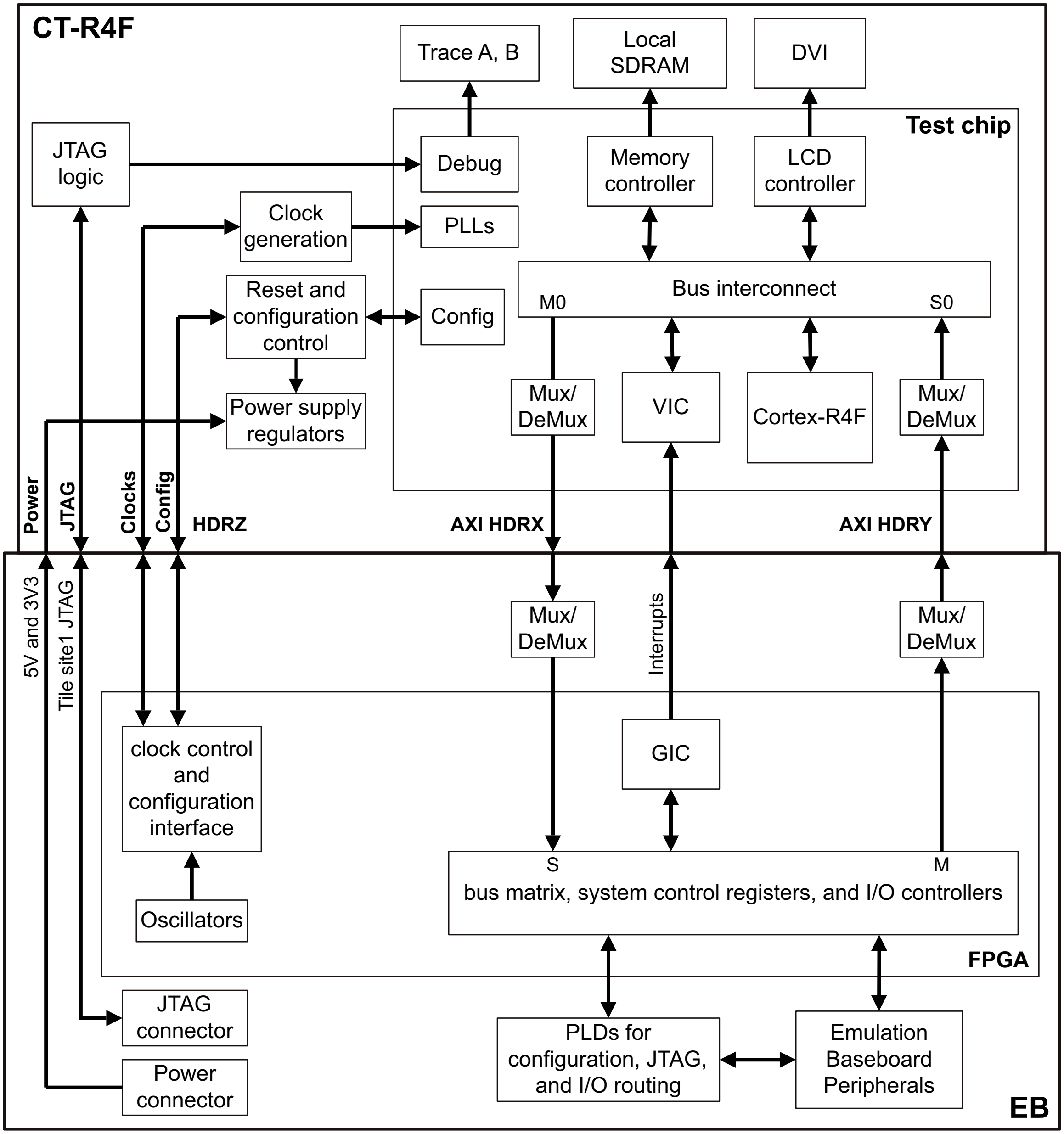


Figure 1-3 System block diagram

1.3.3 ARM Cortex-R4F test chip

The name of the Core Tile reflects the test chip fitted. The CT-R4F has an ARM Cortex-R4F test chip that implements an ARM Cortex-R4F processor and subsystem.

Note

For more details on the ARM Cortex-R4F processor, see the *ARM Cortex-R4 and Cortex-R4F Technical Reference Manual* (ARM DDI 0363).

1.3.4 CT-R4F configuration

The CT-R4F receives configuration signals from the 4-wire serial link controlled by the baseboard.

The CT-R4F configuration signals control:

- the PLLs and clock control registers that control the reference clocks for the core, test chip memory subsystem, and display interface
- the DVI and CLCD display subsystem
- the ARM Cortex-R4F and test chip configuration signals.

ARM Cortex-R4F and test chip configuration signals

The ARM Cortex-R4F and test chip configuration signals are either controlled by the baseboard, or permanently set HIGH or LOW by links on the CT-R4F. In a production ASIC, all these configuration signals are permanently tied HIGH or LOW. However, to experiment with different configurations, you can change certain CT-R4F configuration register values via a 4-wire serial link. See *Overview of Core Tile configuration* on page 3-24 for details.

Note

User control of the signals present in the serial interface is supported through the system control registers on the EB. In a custom baseboard design, provision must also be made to control these configuration signals. See *Overview of Core Tile configuration* on page 3-24 for details of the control requirements.

1.3.5 External processor bus

The test chip on the CT-R4F has two external AXI bus interfaces, one master and one slave. The test chip AXI Slave port is available at the HDRY header and the AXI Master port is available at the HDRX header. See *Header connectors* on page 5-2 for a listing of the AXI signals on the HDRX and HDRY headers and *Bus interface characteristics* on page A-2 for parametric data.

1.3.6 Memory

The CT-R4F does not support PISMO™ Expansion Memory Modules but it does have 512MB of local SDRAM memory fitted. See *Memory system* on page 3-29 for details of the ARM Cortex-R4F test chip memory system.

If additional system memory is required, you can add PISMO Expansion Memory Modules to the EB. See the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411) for details.

1.3.7 Video interface

The ARM Cortex-R4F test chip contains a CLCD controller. An onboard DVI transmitter and interface connector is provided by the CT-R4F.

1.3.8 Clock generation

The primary reference clock for the CT-R4F external AXI buses is provided by the attached baseboard. Other reference clocks are provided by programmable clock generators on the CT-R4F. See *Clocks* on page 3-4 for details of the CT-R4F clock system.

1.3.9 JTAG

The CT-R4F does not have a JTAG connector. You must either use the JTAG connector on the EB, or the JTAG port provided at the TRACE A connector. See *JTAG support* on page 3-30 for details of the CT-R4F JTAG interface.

1.3.10 Power supply monitoring

Interface logic on the CT-R4F enables you to read different supply voltages and currents. The voltages are set during board manufacture. The current drawn by the test chip core, subsystem, and PLL can be read to monitor power consumption.

Power supply is provided by the baseboard. The CT-R4F requires the baseboard to provide 3V3 and 5V. The required local subsystem voltages are generated by power modules on the CT-R4F. See *Power supply monitoring* on page 3-22 for details of the CT-R4F power supply interface.

1.3.11 Configuration switches

During configuration the JTAG source is determined by the JTAG_SRC field in the CT_R4F_CTRL register or by the configuration switch (S1-1) on the CT-R4F. See *CT_R4F_CTRL register* on page 4-27 for information on the JTAG_SRC field, and *CT-R4F layout* on page 1-4 for the location of configuration switch S1.

The S1 switch settings are listed in Table 1-1:

Table 1-1 S1 switch settings

S1	OFF	ON
S1-1	JTAG source is determined by the **JTAG_SRC** bit in the CT_R4F_CTRL register. See *CT_R4F_CTRL register* on page 4-27. Default: JTAG source is the HDRZ header.	JTAG source is the TRACE A connector
S1-2	Default	Reserved

1.3.12 LED indicators

The CT-R4F has LED indicators for power and CT-R4F status. The LED functions are listed in Table 1-2.

Table 1-2 LED indictor functions

LED	Reset state	Function
D1	OFF	ON when all ispClock generators and test chip PLLs are locked.
D2	ON	ON when 4-wire serial interface is in reset.
D7	ON	ON when 3V3 power is applied.

Note

For details on LED indicators present on a Logic Tile and on an EB, see the documentation supplied with those products.

Chapter 2
Getting Started

This chapter describes how to set up and prepare the CT-R4F for use. It contains the following sections:

- *Using the CT-R4F with an Emulation Baseboard* on page 2-2
- *Connecting power* on page 2-4
- *Connecting JTAG debugging equipment* on page 2-5
- *Using the CT-R4F with a custom baseboard* on page 2-9.

Note

The information in this chapter provides only a general overview. Details on using the CT-R4F with an EB is covered in detail by an application note provided on the *Versatile CD*. Ensure that you use the correct application note and FPGA image for this board combination.

2.1 Using the CT-R4F with an Emulation Baseboard

A typical ARM Cortex-R4F development system is shown in Figure 2-1 on page 2-3.

To set up a development system using the Emulation Baseboard (EB):

1. Fit the CT-R4F to tile site 1 on the EB.

2. If required, fit a Logic Tile to tile site 2 on the EB.

3. If required, fit additional Logic Tiles at either tile site to create a tile stack.

   Note

   The CT-R4F has no upper tile headers and can only be used either directly connected to the EB, or at the top of a tile stack. AXI bus multiplexing and demultiplexing logic must be included on any Logic Tile that connects directly to an AXI port on the EB or the CT-R4F. This might reduce the maximum operating frequency of the tile stack.

4. If required, you can connect a RealView Analyzer Tile between the baseboard and a tile, to enable monitoring of signals using a logic analyzer. See *Analyzer Tile User Guide* (ARM DUI 0189) for details.

5. You can also place an Interface Tile on top of the Logic Tile at tile site 2. The Logic Tile can be loaded with an appropriate image that contains your peripherals and the connectors on the Interface Tile can provide access to your peripherals. See *Versatile/IT1 User Guide* (ARM DUI 0188) for details.

6. Connect a JTAG debugger to J18 on the baseboard (see *Connecting a JTAG device to the EB* on page 2-7).

7. Set the CONFIG slide-switch S1 on the EB to ON.

8. Supply power to the EB (see *Supplying power to the EB* on page 2-4).

9. Load the appropriate FPGA images into the Logic Tiles. See the application note for details on the image to use and see the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411) and the user guide for the Logic Tiles fitted for detailed programming procedures.

10. Set the CONFIG slide-switch on the EB to OFF, power cycle the board, and load your application program.

    Note

    For details on how to load and run applications on an EB system, see the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411).

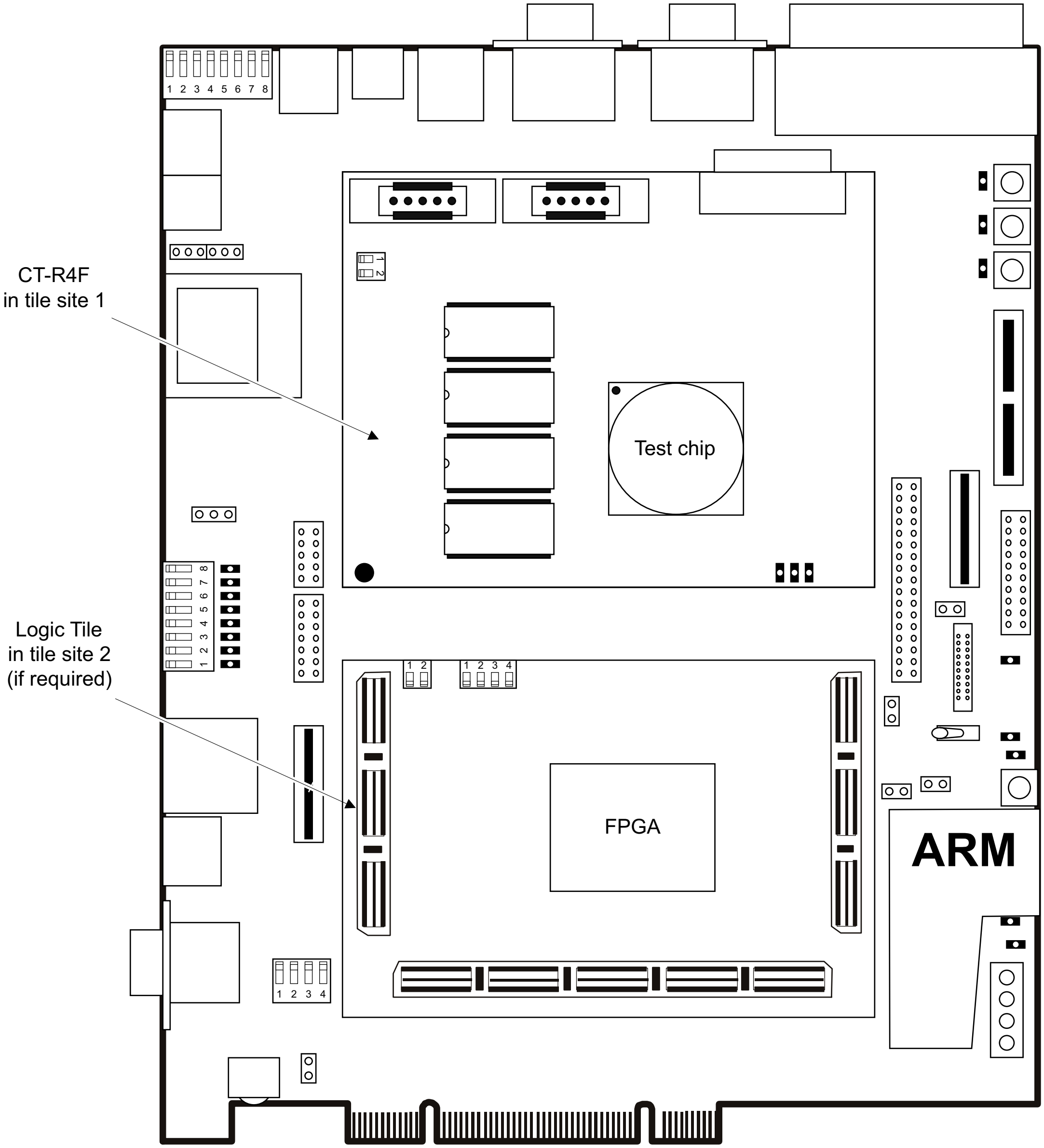


Figure 2-1 Typical ARM Cortex-R4F processor development system

2.2 Connecting power

CT-R4F power is supplied through power-blades in the header connectors. The power source is connected to the EB.

2.2.1 Supplying power to the EB

If the EB is not inserted into a PCI enclosure, you must connect the supplied 12VDC brick power supply to power socket J28 or an external bench power supply to the screw-terminal connector J27. See Figure 2-2.

Note

If you are using the supplied brick power supply connected to J28, the Standby/power push button S7 toggles the power on and off.

If you are using an external power supply connected to J27, or you are powering the board from the PCI backplane, the Standby/power switch is not used and power is controlled by shutting down the external power source.

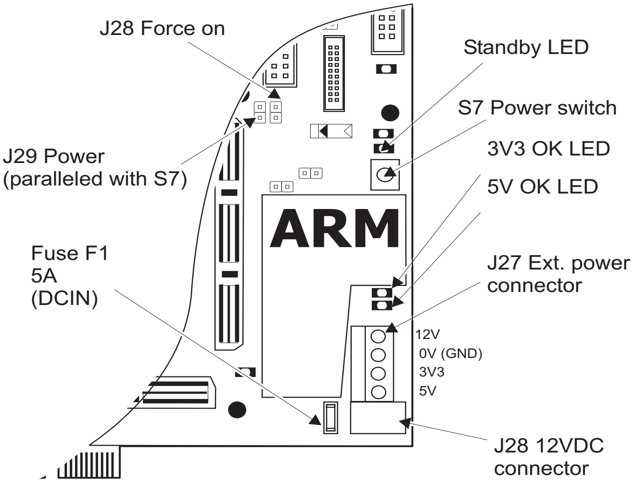


Figure 2-2 EB power connectors

Caution

You must only use one power source for the system. Use only the PCI connector, J27, or J28, do not, for example, use the PCI connector and J27 at the same time.

If you are using separate 5V and 3V3 bench supplies, always set sensible current limits. The 3V3 supply should track the 5V supply. On the EB, the 3V3 supply tracks the 5V supply after an initial 1ms delay.

2.3 Connecting JTAG debugging equipment

The CT-R4F provides a JTAG port at the HDRZ header for use with an Evaluation Baseboard (EB) or custom baseboard that provides a JTAG ICE connector. A JTAG port is also provided at the TRACE A connector for use with RealView ICE. *CoreSight Serial Wire Debug* (SWD) is also supported at the TRACE A connector.

During configuration the JTAG source is determined by the JTAG_SRC field in the CT_R4F_CTRL register or by the configuration switch (S1-1) on the CT-R4F. For further information on determining the JTAG source see *CT_R4F_CTRL register* on page 4-27 and *Configuration switches* on page 1-11.

External JTAG equipment can be used to:

- Download and debug programs.

  You can connect RealView ICE (RVI) or other JTAG debuggers to the external JTAG connector and download and debug programs.

- Download new images to the PLD on the CT-R4F or to FPGAs or PLDs present on other attached Logic Tiles.

  Note

  For a CT-R4F attached to an EB, either the JTAG ICE port (J18) or the USB config port (J16) can be used for downloading FPGA images, but only the JTAG ICE port can be used for debug.

Selection between debugging programs and downloading new images to the FPGA is controlled by the CONFIG slide-switch S1 on the EB. See the documentation supplied with the EB for more details on connecting JTAG and using the CONFIG slide-switch.

Caution

Because the CT-R4F does not provide nonvolatile memory, programs are lost if the power is removed. Use EB flash memory for nonvolatile storage.

The flash memory can be:

- any unused space in the EB NOR flash
- nonvolatile memory available in the EB PISMO memory expansion slots J35.

Do not use spare EB configuration flash space (`address range: 0x4C000000–0x4DFFFFFF`) for program storage.

See the *Programmer's Reference* chapter in the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411) for details of the EB memory map.

The JTAG ICE connector J18 provides a set of signals that enable debugging equipment to be used. When debugging a development system with multiple tiles, connect the JTAG debugging equipment to the JTAG ICE port on the EB and the JTAG signals will be routed through any connected tiles.

The JTAG debug and configuration paths are described in *JTAG support* on page 3-30.

2.3.1 CoreSight Serial Wire Debug

CoreSight *Serial Wire Debug* (SWD) replaces the 5-pin JTAG port with a clock and a single bi-directional data pin. It provides all the normal JTAG debug and test functionality plus real-time access to system memory without halting the core or requiring any target resident code. SWD uses an ARM standard bi-directional wire protocol, defined in the *ARM Debug Interface v5 Architecture Specification* (ARM IHI 0031) to pass data to and from the debugger and the target system in a highly efficient and standard way.

SWD is supported at the TRACE A connector on the CT-R4F. See *Trace connectors* on page 5-27 for the connector pinout.

Note

SWD is not supported at the JTAG ICE connector J18 on the EB.

2.3.2 Connecting a JTAG device to the EB

The JTAG setup for the EB, *RealView ICE* (RVI), and *RealView Trace* (RVT) is shown in Figure 2-3. (See the documentation supplied with your debugger for information on connecting other JTAG products.)

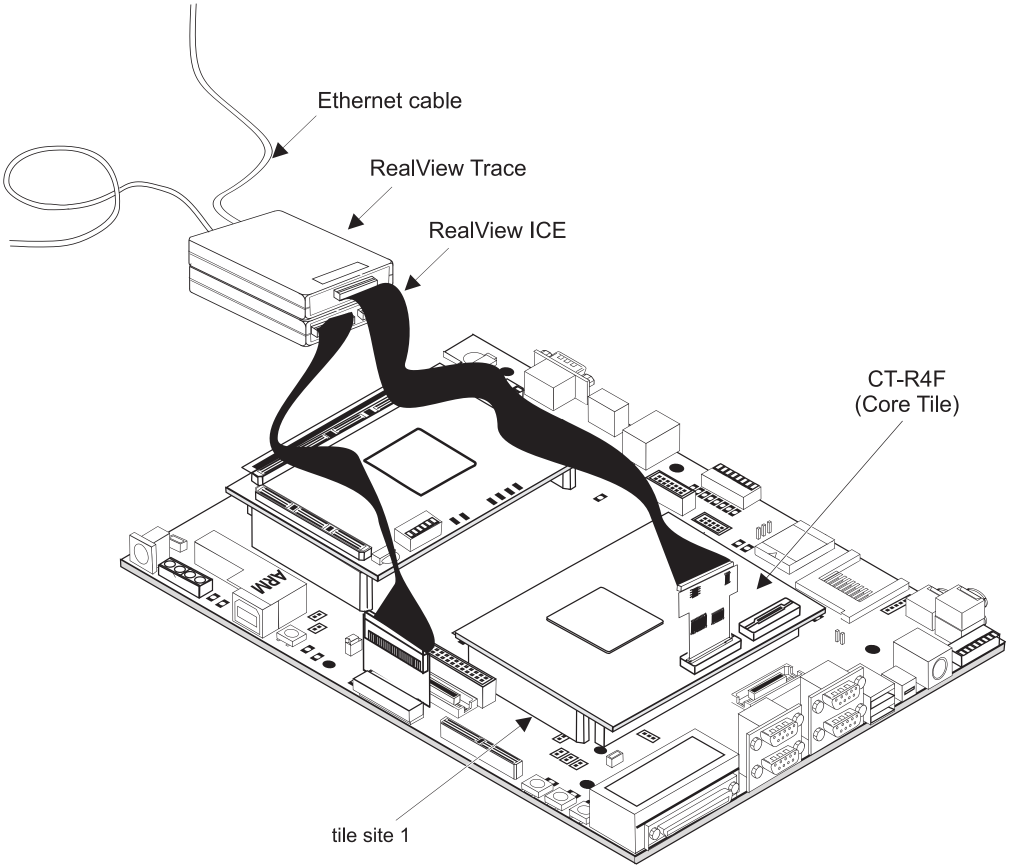


Figure 2-3 JTAG and Trace setup

2.3.3 Connecting Trace

The ARM Cortex-R4F test chip contains an *Embedded Trace Macrocell* (ETM-R4) and Trace connectors are provided on the CT-R4F:

- for 16-bit trace port width, connect the 16-bit Mictor® trace probe to the TRACE A connector on the CT-R4F
- for 32-bit trace port width, connect the 32-bit dual-Mictor probe to the TRACE A and TRACE B connectors.

Note

Trace tools using multiplexed trace packets (such as RealView Trace) require only one Mictor trace probe (Trace Port A). The CT-R4F supports both multiplexed (one trace connector) and demultiplexed mode (two trace connectors). The second connector is present to support trace tools that use demultiplexed trace packets. The trace size supported by the connectors is medium (16-bit packets).

Connect the RealView ICE run control unit to the JTAG ICE connector J18 on the EB to provide the JTAG signals that are required for controlling the ETM-R4 via the *Debug Access Port* (DAP) in the ARM Cortex-R4F test chip. The Ethernet, USB, and power supply cables connect to the RealView ICE unit.

Note

The JTAG signals are also available, including *CoreSight Serial Wire Debug* (SWD), at the TRACE A connector. See *Board level control and status* on page 4-27 for details on selecting the JTAG source and *Trace connectors* on page 5-27 for pinout details.

If your baseboard does not provide a separate JTAG ICE connector, you must setup the CT-R4F to provide the JTAG port at the TRACE A connector and use the RealView ICE socket on the Mictor trace probe.

For more details on using Trace, see the RealView ICE and RealView Trace User Guide (ARM DUI 0155) supplied with your trace hardware.

2.4 Using the CT-R4F with a custom baseboard

If you are designing a custom baseboard to accept a CT-R4F, you must ensure that your board meets the following requirements:

Mechanical layout

Mechanical layout details are available from ARM, contact ARM Support for further information.

Power supplies

The CT-R4F uses 5V and 3V3 supplies connected through power-blades in the header connectors. See *Header connectors* on page 5-2 for details. The CT-R4F uses the 5V supply to:

- generate the 1V2 supply for the ARM Cortex-R4F test chip core and 2V6 supply for the ARM Cortex-R4F test chip I/O
- generate the 2V5 supply for the ARM Cortex-R4F test chip PLL
- generate the 1V8 supply for the PLD core.

The 3V3 supply is used by the PLD I/O, SDRAM and all external CT-R4F interfaces.

The test chip and PLL voltages and currents are monitored by an *Analog to Digital Converter* (ADC) on the CT-R4F and the values are transmitted to the baseboard using the 4-wire serial link in the PLD. See *Power supply monitoring* on page 3-22 for details on the voltage and current monitoring scheme.

Clock control

Your primary reference clock must be supplied by an attached Logic Tile or baseboard. The CT-R4F clocking system is described in Chapter 3 *CT-R4F Hardware Description* and the ARM Cortex-R4F test chip specific clocking requirements are described in Chapter 4 *Programmer's Reference*. See also the *ARM Cortex-R4 and Cortex-R4F Technical Reference Manual* (ARM DDI 0363) for further information on the AXI interface clocking requirements. For an example of the clock structure design, see *Application Note 217* supplied with the product on the *Versatile CD*.

JTAG control

The CT-R4F provides a JTAG port at the TRACE A connector. The baseboard may also provide a JTAG connector and route the JTAG signals to the HDRZ header on the CT-R4F. JTAG routing is described in *JTAG support* on page 3-30.

AXI bus multiplexing

A multiplexing scheme is necessary to reduce the number of pins required on the HDRX and HDRY headers. Bus multiplexing is managed by the ARM Cortex-R4F test chip. A compatible multiplexing scheme must be implemented by the custom baseboard. See *AXI bus multiplexing* on page 3-22 for details of the multiplexing requirements.

AXI slaves

In order for the ARM Cortex-R4F to boot, the baseboard must implement AXI slaves that cover the 4GB memory map.

Chapter 3
CT-R4F Hardware Description

This chapter describes the on-board hardware of the CT-R4F. It contains the following sections:

- *Core Tile architecture* on page 3-2
- *ARM Cortex-R4F test chip* on page 3-3
- *Clocks* on page 3-4
- *Resets and interrupts* on page 3-10
- *DVI and CLCD routing* on page 3-20
- *Power supply monitoring* on page 3-22
- *AXI bus multiplexing* on page 3-22
- *Overview of Core Tile configuration* on page 3-24
- *Memory system* on page 3-29
- *JTAG support* on page 3-30.

Note

The HBI-0196 board supports the ARM Cortex-R4F test chip, but the availability of boards depends on the availability of test chips. Contact your sales representatives for details on currently available Core Tiles.

3.1 Core Tile architecture

The high level architecture of the CT-R4F is shown in Figure 3-1.

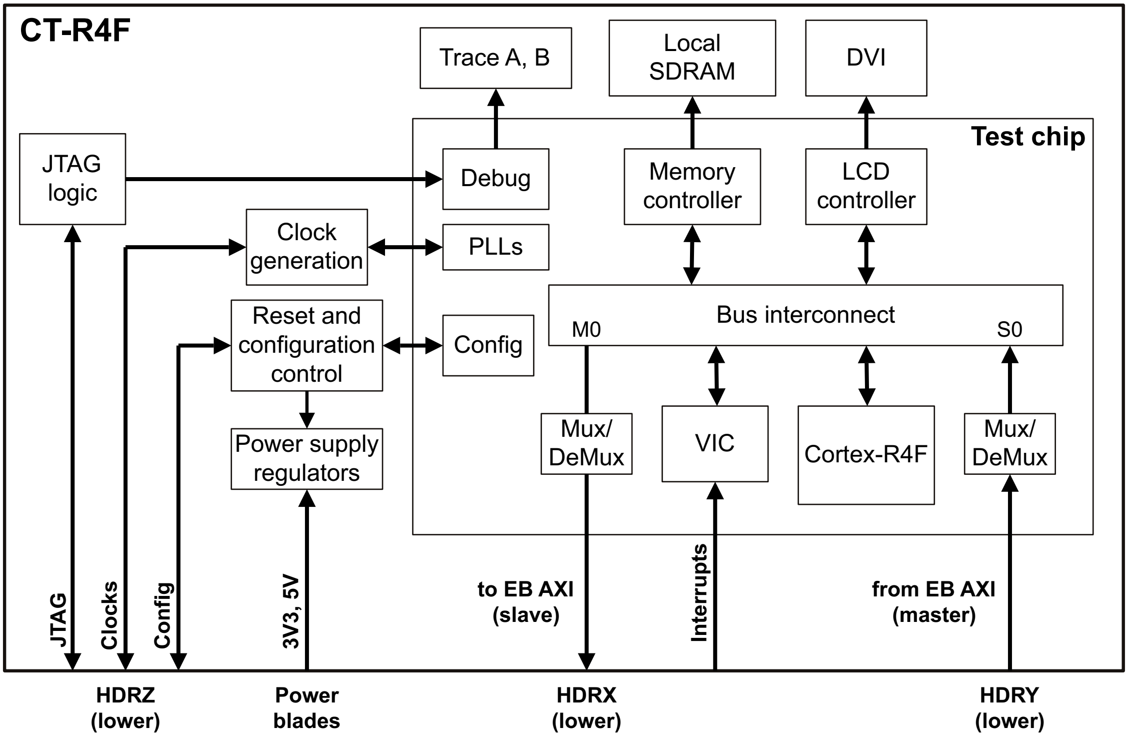


Figure 3-1 CT-R4F block diagram

3.2 ARM Cortex-R4F test chip

The ARM Cortex-R4F test chip is a proof-of-concept vehicle for the ARM Cortex-R4F macrocell.

The ARM Cortex-R4F test chip on the CT-R4F provides:

- ARM Cortex-R4F processor implementing the ARMv7-R architecture and ARMv7 debug architecture. The test chip includes:
  - A *Floating Point Unit* (FPU) implementing the VFPv3-D16 architecture and the Common VFP Sub-Architecture v2
  - 2x 16KB L1 Caches (D and I)
  - 3x 64KB TCM (A, B0 and B1)
  - A *Vectored Interrupt Controller* (VIC) port
  - A debug interface to a CoreSight *Debug Access Port* (DAP)
  - A trace interface to a CoreSight ETM-R4 and *Cross Trigger Interface* (CTI).
- SDRAM Controller (PL340)
- Direct Memory Access Controller (PL330)
- Vectored Interrupt Controller (PL192)
- CLCD Controller (PL111)
- High Performance Matrix (PL301)
- Multiplexed Master and Slave AXI bus interfaces
- Serial configuration interface
- 3x PLL local generators.

With the CT-R4F installed on an Emulation Baseboard (EB) it can be used for ARM Cortex-R4F benchmarking, software development, and validation. Installing a Logic Tile on the EB enables new peripherals and controllers to be developed and tested in an ARM Cortex-R4F based system.

Note

For more details see Chapter 4 *Programmer's Reference* and the *ARM Cortex-R4 and Cortex R4F Processor Technical Reference Manual* (ARM DDI 0363).

3.3 Clocks

The CT-R4F provides the five external clocks for the ARM Cortex-R4F test chip. Four of the clocks are generated locally and one is selected from one of two clocks sourced from the baseboard.

The four locally generated clocks are:

REFCLK (OSC0) This is the main reference clock for the ARM Cortex-R4F test chip which generates the internal clocks:

- **CPU_CLK**, the clock for the processor core
- **ACLK**, the processor system and test chip subsystem AXI clock
- **PCLK**, the AXI to APB subsystem and APB debug clock.

The frequency ratio between **CPUCLK**, **ACLK**, and **PCLK** is determined at reset by the value of the *Clock Mode* field in *Config Word 0x1* that is sent to the CT-R4F by the 4-wire serial configuration interface.

FBCLK_IN (OSC1) This clock is used to deskew the ARM Cortex-R4F internal SDRAM clock, **MCLK** , which is independent of **ACLK**. It clocks the pad side of the SDRAM memory controller ASYNC bridge in the test chip.

CLCDCLK_IO (OSC2) This clock is the **CLCDCLK** panel side clock for the CLCD controller. The remaining panel side clock **nCLCDCLK** is generated internally by the ARM Cortex-R4F test chip and is an inverted version of **CLCDCLK**.

TIMERCLK This is a fixed 24MHz clock and is the reference clock for the ARM Cortex-R4F test chip timer.

The clock sourced externally by the baseboard is:

OC_ACLK_IN This clock is used to deskew the ARM Cortex-R4F test chip internal AXI clock, **ACLK**. The clock is selected from one of two sources, **CLK_NEG_UP_IN** or **CLK_IN_MINUS1** from the baseboard. **CLK_IN_MINUS1** is selected by the ARM Cortex-R4F PLD default image. This clock is exported as **CLK_POS_DN_OUT** to the baseboard. The generated clock, **OC_AXI_CLK** clocks the pad side of the ASYNC bridges on the AXI ports. This clock is exported as **CLK_NEG_DN_OUT** to the baseboard.

A double rate clock, **OC_AXI_CLK2** is also generated by the ARM Cortex-R4F test chip and controls the AXI mux/demux operation. This clock is exported as **CLK_OUT_MINUS1** to the baseboard.

A simplified diagram of the CT-R4F clocking system is shown in Figure 3-2 on page 3-6.

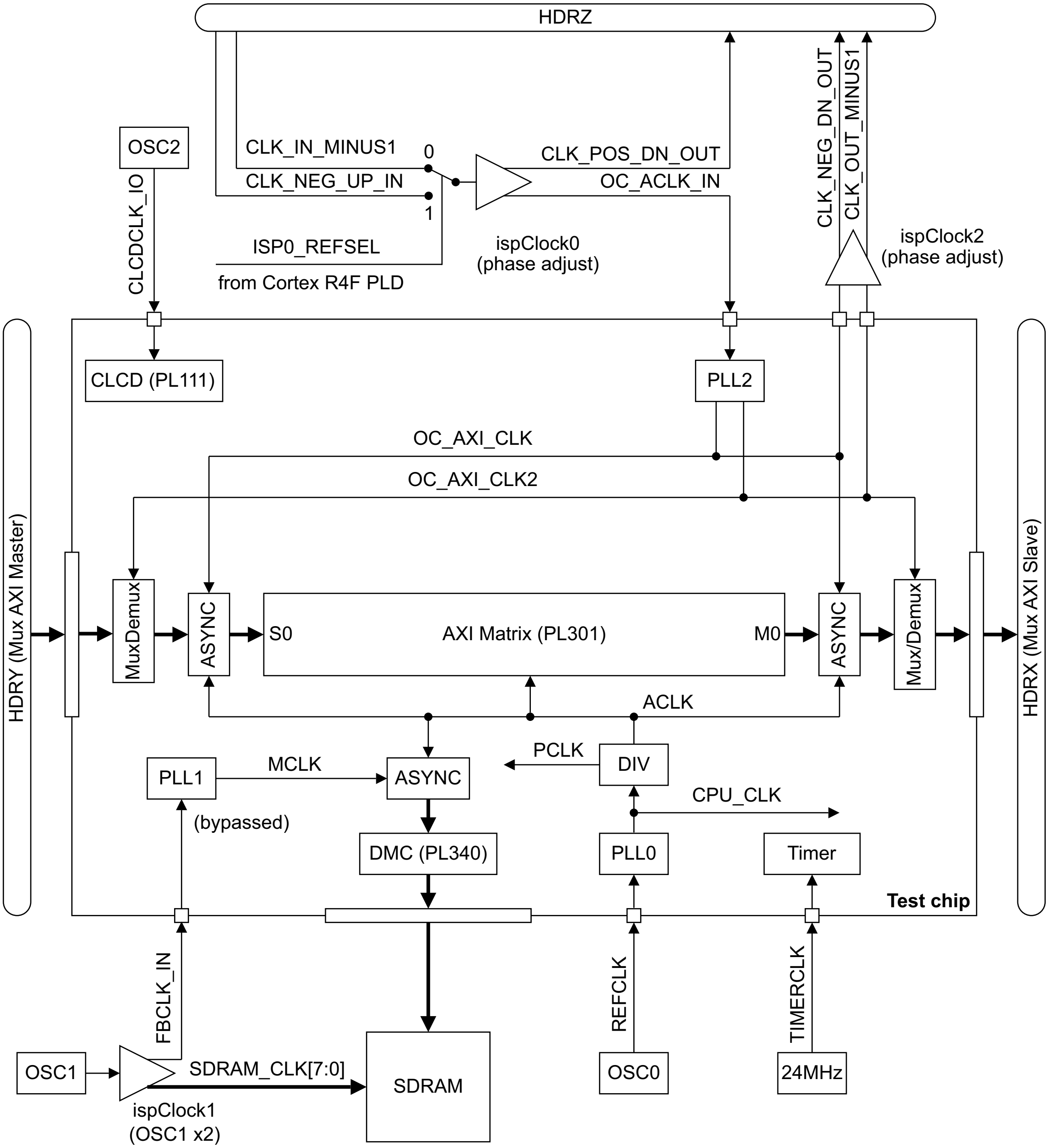


Figure 3-2 CT-R4F clock system

3.3.1 Test chip PLLs

A simplified diagram of the PLL design in the ARM Cortex-R4F test chip is shown in Figure 3-3.

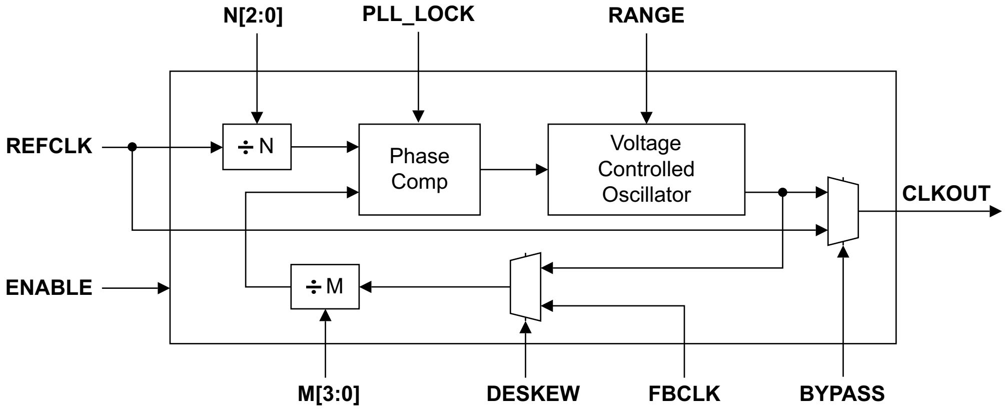


Figure 3-3 ARM Cortex-R4F test chip PLL

The PLLs are configured by the CT-R4F PLD using the *Serial Configuration Controller* (SCC) in the test chip.

The M divider shown in Figure 3-3 follows a binary divide ratio mapping, as shown in Table 3-1.

Table 3-1 M binary divide ratio

M[3]	M[2]	M[1]	M[0]	Divide ratio
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
..	..	..	..	..
1	1	1	1	15

The N divider shown in Figure 3-3 on page 3-7 follows a binary divide ratio mapping, as shown in Table 3-2.

Table 3-2 N binary divide ratio

N[2]	N[1]	N[0]	Divide ratio
0	0	0	8
0	0	1	1
0	1	0	2
..	..	..	..
1	1	1	7

The frequency of **CLKOUT** is obtained by the following formula:

$$F_{CLKOUT} = \frac{F_{REFCLK} \times M}{N}$$

The frequency of the *Voltage Controlled Oscillator* (VCO) (Fvco) is obtained by adjusting the parameters for each of the three PLLs. This is described by the following formulas:

- When **RANGE** is LOW, the Fvco is calculated by the following formula:

$$F_{VCO} = \frac{F_{REFCLK} \times M \times 4}{N}$$

- When **RANGE** is HIGH, the Fvco is calculated by the following formula:

$$F_{VCO} = \frac{F_{REFCLK} \times M}{N}$$

The absolute range limits for counter values are:

- `M = 1 to 16`

  Note

  When **RANGE** is LOW, the acceptable range for M is 1 to 9.

- `N = 1 to 8`

Table 3-3 shows the absolute range limits for the PLL frequency.

Table 3-3 Absolute range limits for PLL frequency

Frequency	Symbol	Minimum	Maximum	Unit
REFCLK	F_{REFCLK}	33	300	MHz
VCO	F_{VCO}	133	533	MHz
CLKOUT	F_{CLKOUT}	33	533	MHz

See also *CT-R4F configuration* on page 4-15.

Note

Default values are provided by the EB during reset. If you want to change these default values you are advised to contact ARM Support for assistance as not all combinations of values are always supported.

3.4 Resets and interrupts

This section describes the reset and interrupt signals and contains the following sections:

- *Resets*
- *Interrupts* on page 3-15.

3.4.1 Resets

Several resets are required by the baseboard and the CT-R4F.

This section describes the resets and contains the following subsections:

- *Baseboard resets*
- *CT-R4F resets* on page 3-11
- *ARM Cortex-R4F test chip resets* on page 3-12
- *CT-R4F reset sequencing* on page 3-14.

Baseboard resets

The reset logic on the baseboard is required to initialize the baseboard and attached Logic Tiles and Core Tiles. The baseboard has several reset sources and generates several reset signals.

The EB reset sources and the function of the EB reset signals are described in detail in the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411).

CT-R4F resets

The CT-R4F requires the following reset signals at the HDRZ header:

Table 3-4 CT-R4F Resets

Name	Description
nSYSPOR	Power-on reset from the baseboard. This reset is released approximately 7µs after the global signal **GLOBAL_DONE** goes HIGH indicating that all programmable devices in the system have been configured.
nCOLDRST	Cold reset from the baseboard. This signal is connected to **nSYSPOR** in the EB FPGA.
nSYSRST	Main reset from the baseboard. This reset is released approximately 20µs after **nSYSPOR** goes HIGH or goes LOW after **nSRST** goes LOW and returns HIGH approximately 20µs after **nSRST** goes HIGH.
nWARMRST	This is the warm reset from the baseboard. This signal is connected to **nSYSRST** in the EB FPGA.
D_nSRST	**nSYSRST** request from the debug system.
C_nSRST	**nSYSRST** request from the config system.
D_nTRST	JTAG reset to the debug JTAG scan chain.
C_nTRST	JTAG reset to the config JTAG scan chain.
PLDnRST	Resets and restarts the 4-wire serial data transfer to the CT-R4F PLD.

ARM Cortex-R4F test chip resets

Figure 3-4 shows how the resets to the ARM Cortex-R4F test chip are controlled by the CT-R4F PLD.

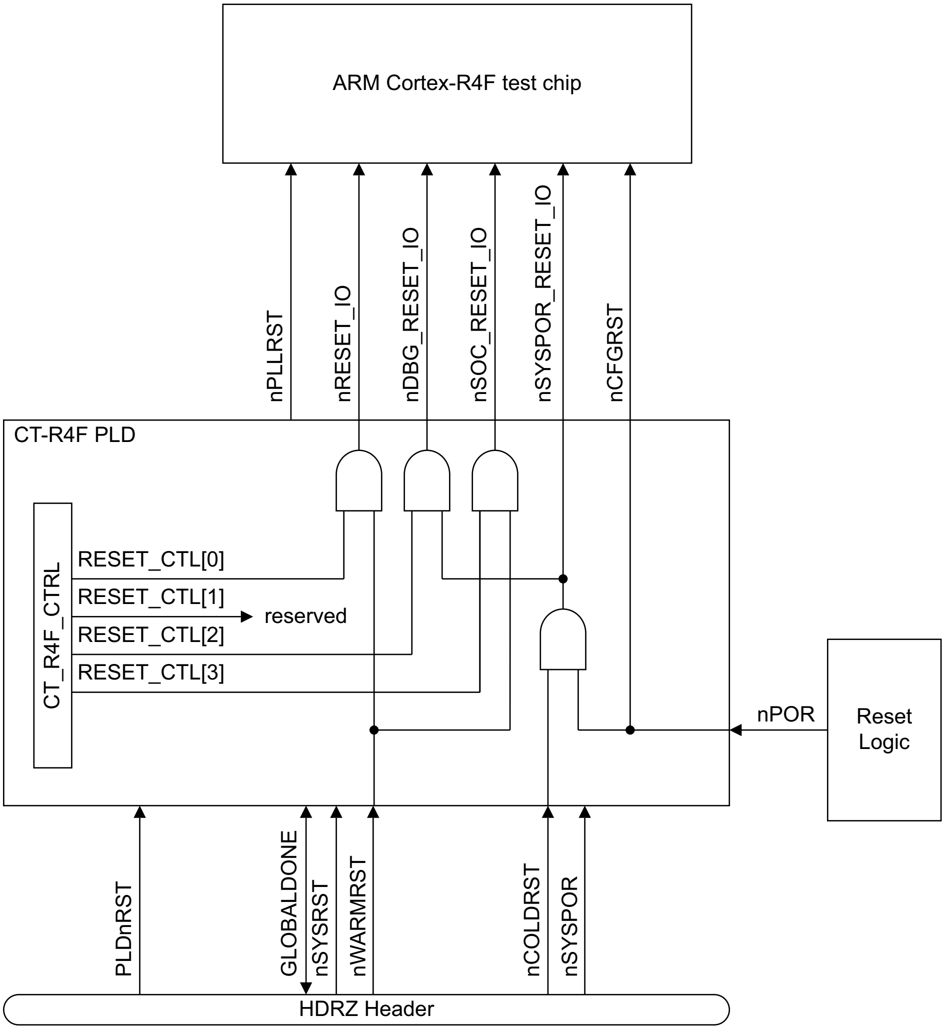


Figure 3-4 CT-R4F test chip reset control

The ARM Cortex-R4F test chip resets are listed in Table 3-5.

Table 3-5 CT-R4F test chip resets

Name	Function	Description
nSYSPOR_RESET_IO	Warm reset	Active low reset for the CP14 debug logic.
nSOC_RESET_IO	SOC reset	Active low reset for the AXI sub-system in the test chip.
nDBG_RESET_IO	Debug reset	Active low reset for the APB logic, CoreSight ETM, and CoreSight DAP but excludes the CP14 debug logic.
nRESET_IO	Processor Reset	Active low reset for the ARM Cortex-R4F processor.
nPLLRST	PLL reset	Active low reset for all the test chip PLLs.
nCFGRST	Config Block reset	Active low Reset for the Configuration Block in the test chip.

3.4.2 CT-R4F reset sequencing

Correct initialization of the CT-R4F and its associated baseboard requires a timed reset sequence. Details of the full EB reset sequence are detailed in the *RealView Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411). The main reset timing when using a CT-R4F in combination with an EB is shown in *EB and CT-R4F reset sequence*.

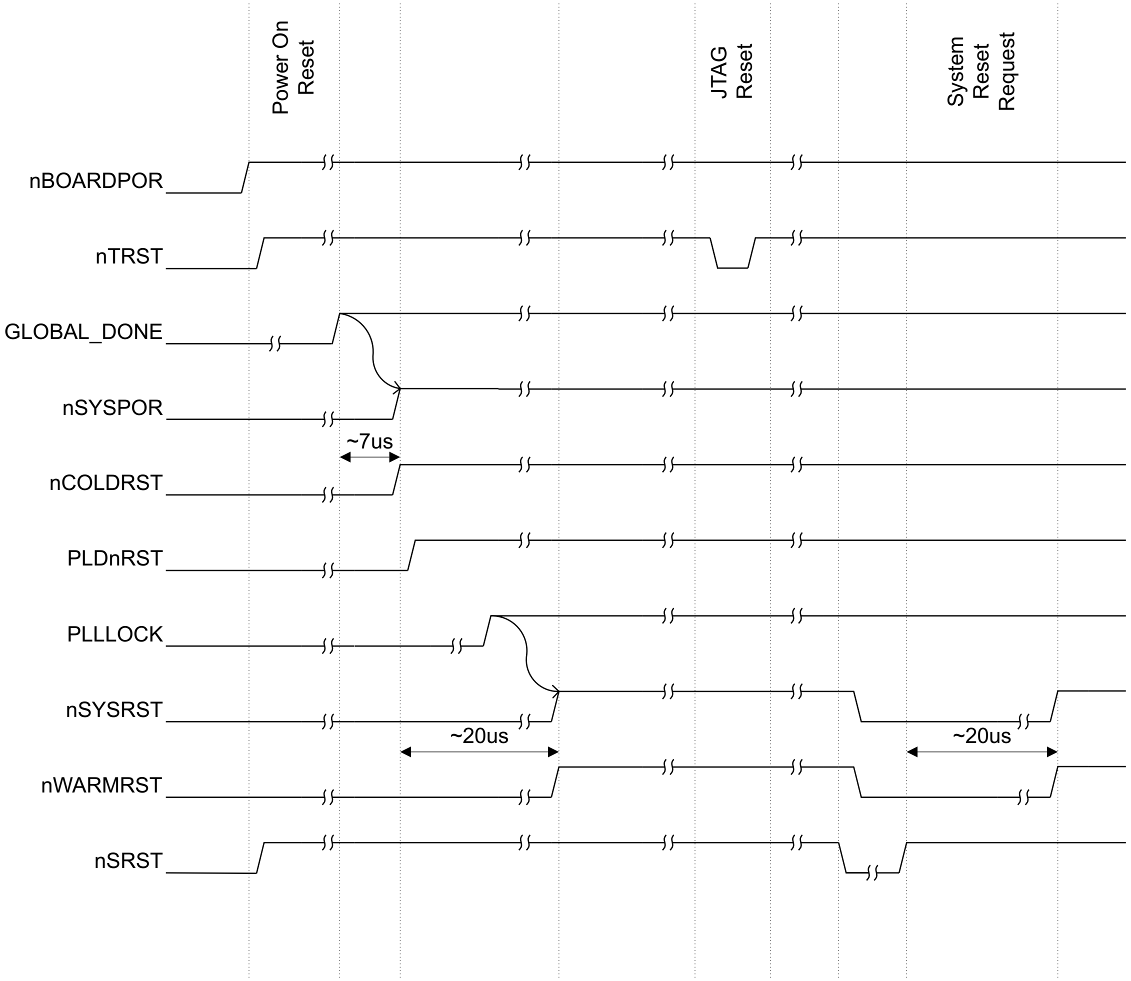


Figure 3-5 EB and CT-R4F reset sequence

3.4.3 Interrupts

A simplified diagram of the interrupt routing in the ARM Cortex-R4F test chip is shown in Figure 3-6.

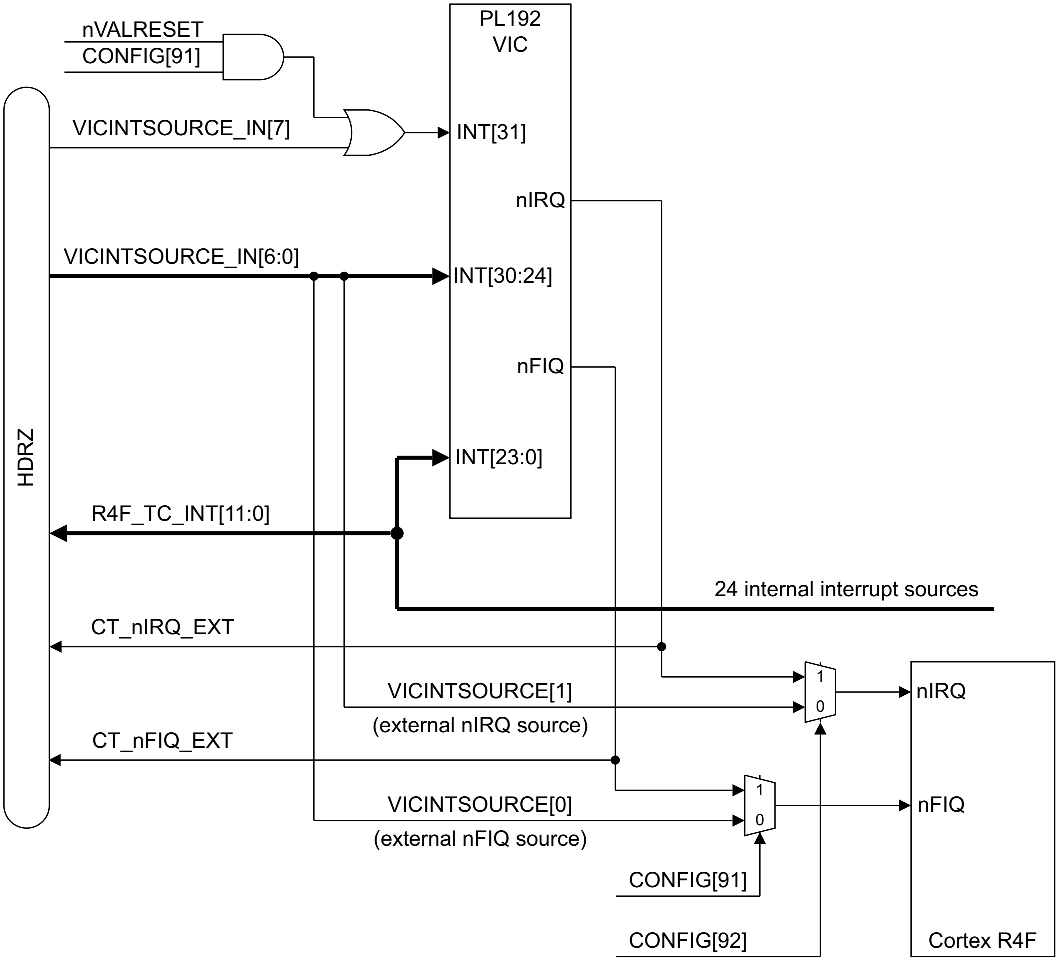


Figure 3-6 ARM Cortex-R4F interrupt routing

The PL192 VIC in the test chip can be bypassed and the **IRQ** and **FIQ** inputs to the ARM Cortex-R4F processor sourced externally. When used with the EB, the external **IRQ** and **FIQ** are sourced respectively by *Generic Interrupt Controllers*, GIC1 and GIC2 in the EB FPGA. See *RealView Emulation Baseboard User Guide (Lead Free*) (ARM DUI 0411) for further information.

——— **Note** ———

The source of the **IRQ** and **FIQ** interrupts is set during reset by the configuration register CT_R4F_TC_CFG2. See *CT_R4F_TC_CFG2 register* on page 4-22 for details. By default, the VIC in the test chip is not bypassed on the CT-R4F.

———

See *ARM PrimeCell Vectored Interrupt Controller (PL192) Technical Reference Manual* (ARM DDI 0273) for details on the PL192 VIC in the test chip.

Table 3-6 lists the ARM Cortex-R4F VIC interrupt allocations.

——— **Note** ———

The twelve internal peripheral interrupts **R4F_TC_INT[11:0]**, from the ARM Cortex-R4F test chip, are made available at the HDRZ header. See Table 3-6 for the interrupt Z bus allocations. This enables the internal PL192 VIC to be bypassed and an external interrupt controller to be used independently. With the PL192 VIC bypassed, interrupt inputs **VICINITSOURCE[0]** and **VICINITSOURCE[1]** to the CT-R4F must provide the external **nFIQ** and **nIRQ** signals respectively to the ARM Cortex-R4F processor.

———

Table 3-6 ARM Cortex-R4F test chip VIC interrupt allocations

VIC Interrupt Number	Interrupt Name	Source	Z Bus signal	Description
0	(reserved)	Test chip	**Z210**	Do not connect, reserved for production test.
1	(reserved)	Test chip	**Z211**	Do not connect, reserved for production test.

Table 3-6 ARM Cortex-R4F test chip VIC interrupt allocations (continued)

VIC Interrupt Number	Interrupt Name	Source	Z Bus signal	Description
2	**DMAC_IRQ[0]**	PL330 DMAC	**Z213**	Active HIGH interrupt outputs. The DMAC sets irq<N> HIGH when it executes a DMASEV instruction for event N, if the Interrupt Enable Register is programmed to signal an interrupt for event N.
3	**DMAC_IRQ[1]**		**Z214**	
4	**DMAC_IRQ[2]**		**Z215**	
5	**DMAC_IRQ[3]**		**Z219**	
6	**DMAC_IRQ_ABORT**	PL330 DMAC	–	The DMAC sets this signal HIGH when an abort occurs and it remains HIGH if any thread is in the Faulting completing state or Faulting state. If all threads are not in the Faulting completing state or Faulting state then the DMAC sets this signal LOW.
7	**CLCDINTR**	PL111 CLCDC	**Z212**	Active HIGH single combined interrupt for the CLCDC.
8	(reserved)	Test chip	–	–
9	**FPDZC**	ARM Cortex-R4F FPU	–	Masked floating-point divide-by-zero exception.
10	**FPIDC**		–	Masked floating-point input denormal exception.
11	**FPIOC**		–	Masked floating-point invalid operation exception.
12	**FPIXC**		–	Masked floating-point inexact exception.
13	**FPOFC**		–	Masked floating-point overflow exception.
14	**FPUFC**		–	Masked floating-point underflow exception.
15	**nPMUIRQ**	ARM Cortex-R4F	–	Interrupt request by *Performance Monitor Unit* (PMU).
16	**nVALFIQ**	ARM Cortex-R4F	–	Request for an Interrupt from the ARM Cortex-R4F system validation register.

Table 3-6 ARM Cortex-R4F test chip VIC interrupt allocations (continued)

VIC Interrupt Number	Interrupt Name	Source	Z Bus signal	Description
17	**nVALIRQ**	ARM Cortex-R4F	–	Request for a Fast Interrupt from the ARM Cortex-R4F system validation register.
18	**TIMINT1**	SP804 Timer	**Z220**	Timer1 interrupt, active HIGH.
19	**TIMINT2**		**Z221**	Timer2 interrupt, active HIGH.
20	**TIMINTC**		**Z222**	Combined timer interrupt, active HIGH.
21	**CTIRQ**	ETM-R4/CTI	–	ETM Cross Trigger interrupt, active HIGH, connects to inverted ETM CTI output **CTITRIGOUT[3]**.
22	**COMMRX**	ARM Cortex-R4F	**Z217**	Write-DTR full. Write *Data Transfer Register* (DTR) full from ARM Cortex-R4F processor debug system.
23	**COMMTX**		**Z218**	Read-DTR empty. Read *Data Transfer Register* (DTR) empty from ARM Cortex-R4F processor debug system.
24	**VICINTSOURCE_IN[0]**	HDRZ header	**Z200**	External interrupt 0, is the **FIQ** from the EB GIC. It connects to **VICINTSOURCE[24]** on the PL192 VIC. If the VIC is bypassed, it connects also to the **FIQ** input of the ARM Cortex-R4F processor.
25	**VICINTSOURCE_IN[1]**	HDRZ header	**Z201**	External interrupt 1, is the **IRQ** from the EB GIC. It connects to **VICINTSOURCE[25]** on the PL192 VIC. If the VIC is bypassed, it connects also to the **IRQ** input of the ARM Cortex-R4F processor.
26	**VICINTSOURCE_IN[2]**	HDRZ header	**Z202**	External interrupt 2, is a general purpose peripheral interrupt. It connects to **VICINTSOURCE[26]** on the PL192 VIC.

Table 3-6 ARM Cortex-R4F test chip VIC interrupt allocations (continued)

VIC Interrupt Number	Interrupt Name	Source	Z Bus signal	Description
27	**VICINTSOURCE_IN[3]**	HDRZ header	**Z203**	External interrupt 3, is a general purpose peripheral interrupt. It connects to **VICINTSOURCE[27]** on the PL192 VIC.
28	**VICINTSOURCE_IN[4]**	HDRZ header	**Z204**	General purpose external interrupt. It connects to **VICINTSOURCE[28]** on the PL192 VIC.
29	**VICINTSOURCE_IN[5]**	HDRZ header	**Z205**	General purpose external interrupt. It connects to **VICINTSOURCE[29]** on the PL192 VIC.
30	**VICINTSOURCE_IN[6]**	HDRZ header	**Z206**	General purpose external interrupt. It connects to **VICINTSOURCE[30]** on the PL192 VIC.
31	**VICINTSOURCE_IN[7]**	**nVALRESET** (default) or HDRZ header	**Z207**	Connects to **VICINTSOURCE[31]** on the PL192 VIC. If CT_R4F_TC_CF2[27] is set to bypass the PL192 VIC, the interrupt source is **nVALRESET**, a request for a reset from the ARM Cortex-R4F system validation register. If the PL192 VIC is not bypassed, it is a general purpose external interrupt from the HDRZ header.
–	**CT_nFIQ_EXT**	PL192 VIC	**Z208**	**nFIQ** interrupt from the PL192 VIC in the ARM Cortex-R4F test chip.
–	**CT_nIRQ_EXT**	PL192 VIC	**Z209**	**nIRQ** interrupt from the PL192 VIC in the ARM Cortex-R4F test chip.

3.5 DVI and CLCD routing

To maximize the performance of the CLCD interface, the CT-R4F includes a local analog color DAC, DVI transmitter, and DVI interface connector providing support for both digital and analog displays. The CLCD signals can also be routed to the baseboard CLCD interface through the HDRZ header. A simplified diagram of the CLCD signal routing is shown in Figure 3-7. CLCD signal routing is controlled by **BBDVI_nEN**. The value of **BBDVI_nEN** can be set during reset using the 4-wire interface to the CT_R4F_DVI register in the CT-R4F PLD. See *DVI and CLCD configuration* on page 4-26 for details.

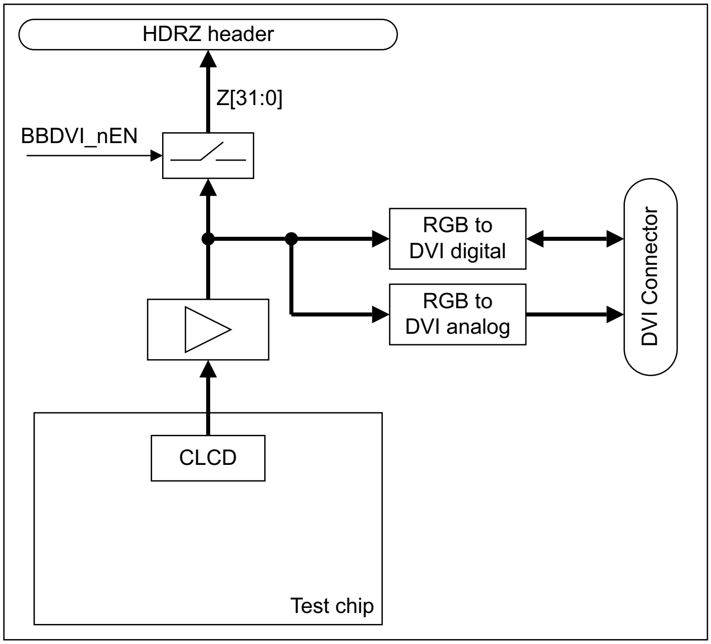


Figure 3-7 CLCD interface routing

The Z bus allocations for the CLCD signals are listed in Table 3-7. See *HDRZ signals* on page 5-18 for further information on the HDRZ header.

Table 3-7 HDRZ header CLCD signals

HDRZ bus	CLCD signal	HDRZ bus	CLCD signal
Z[0]	**R0**	Z[16]	**B0**
Z[1]	**R1**	Z[17]	**B1**
Z[2]	**R2**	Z[18]	**B2**
Z[3]	**R3**	Z[19]	**B3**
Z[4]	**R4**	Z[20]	**B4**
Z[5]	**R5**	Z[21]	**B5**
Z[6]	**R6**	Z[22]	**B6**
Z[7]	**R7**	Z[23]	**B7**
Z[8]	**G0**	Z{24]	**CLAC**
Z[9]	**G1**	Z{25]	**CLCP**
Z[10]	**G2**	Z{26]	**CLFP**
Z[11]	**G3**	Z{27]	**CLLE**
Z[12]	**G4**	Z{28]	**CLLP**
Z[13]	**G5**	Z{29]	**CLPOWER**
Z[14]	**G6**	Z{30]	–
Z[15]	**G7**	Z{31]	**CLCP**

3.6 Power supply monitoring

The EB system FPGA 4-wire serial interface to the CT-R4F PLD enables you to monitor CT-R4F on-board voltages and currents. The voltage and currents are monitored by an 8-channel 12-bit *Analog to Digital Converter* (ADC) on the CT-R4F and are held in the CT_R4F_ADCx registers in the CT-R4F PLD for continuous transfer to the baseboard via the 4-wire serial interface.
See *Power supply monitoring* on page 4-28 for further details on the CT_R4F_ADCx registers.

3.7 AXI bus multiplexing

A multiplexing scheme is necessary to reduce the number of pins required on the HDRX and HDRY headers.

3.7.1 Multiplexing scheme

The AXI bus multiplexing and demultiplexing scheme is shown in Figure 3-8.

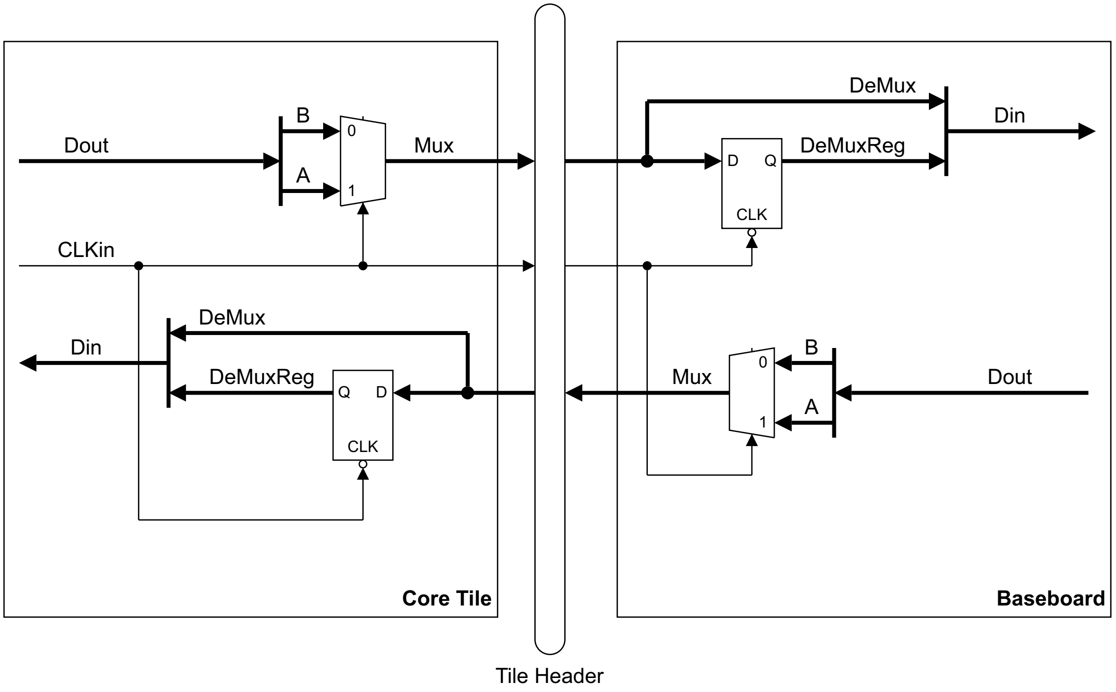


Figure 3-8 CT-R4F AXI bus multiplexing scheme

Data is multiplexed as follows:

Dout is split bitwise into data **Dout/A** and **Dout/B** and is multiplexed onto **Mux** depending on the level of **CLKin**. When **CLKin** is HIGH, **Dout/A** is selected and when **CLKin** is LOW, **Dout/B** is selected. For example, when **CLKin** is HIGH, **RDATA0[31:0]** is selected and when **CLKin** is LOW, **RDATA0[63:32]** is selected. See *HDRY signals* on page 5-11 for details of the AXI port 0 multiplexed signals, and *HDRX signals* on page 5-4 for details of the AXI port 1 multiplexed signals.

Data is de-multiplexed as follows:

Dout/A is present on **Mux** when **CLKin** is HIGH and is latched onto **DeMuxLatch** when **CLKin** goes LOW. **Dout/B** is present on **Mux** when **CLKin** is LOW and is passed straight through as **DeMux**. **DeMuxLatch** and **DeMux** are combined bitwise as **Din**.

Note

This design requires that **Dout** is generated on the rising edge of **CLKin,** and that **Din** is captured on the rising edge of **CLKin**.

AXI signal routing

The CT-R4F AXI master port signals to the baseboard are 2:1 multiplexed onto pins X0 through to X143 on the HDRX header. Similarly, the CT-R4F AXI slave port signals from the baseboard are 2:1 multiplexed onto pins Y0 through to Y143 on the HDRY header.

See *HDRX signals* on page 5-4 and *HDRY signals* on page 5-11 for pinout information.

3.8 Overview of Core Tile configuration

The ARM Cortex-R4F test chip clocks and a number of system parameters are configurable on the CT-R4F. In a final product, core configuration is static and the core configuration signals are tied HIGH or LOW and the clocks are fixed. However, the CT-R4F enables you to program these signals for experimentation.

There are several ways that CT-R4F configuration occurs:

- The CT-R4F PLD.

  This is the primary configuration source. The CT-R4F PLD serial registers are described in *CT-R4F configuration* on page 4-15.

  Note

  The extent of configuration performed by the CT-R4F PLD depends on the type of reset applied to the CT-R4F.

- CT-R4F specific configuration registers in the EB system FPGA. These registers enable you to change several of the parameters controlled by the CT-R4F PLD.

  For details of the system registers in the EB see *Application Note 217* supplied with the product on the *Versatile CD*.

  Note

  The binary image for the EB system FPGA depends on whether a tile is fitted in EB tile site 2. A Logic Tile can be fitted here for system prototyping. The application notes supplied include FPGA and PLD images for the current ARM supported tile combinations.

- Signals present on the CT-R4F HDRZ header.
- Control registers in the ARM Cortex-R4F test chip.

  The *Serial Configuration Controller* (SCC) in the test chip is described in Chapter 4 *Programmer's Reference*.

- Control registers in the ARM Cortex-R4F processor.

  For details of the registers in the ARM Cortex-R4F processor see the *Cortex-R4 and Cortex-R4F Technical Reference Manual* (ARM DDI 0363).

3.8.1 CT-R4F PLD signals

The CT-R4F PLD performs the following functions:

- controlling the individual resets to the ARM Cortex-R4F test chip
- configuring the CT-R4F and ARM Cortex-R4F test chip PLLs on power up
- configuring ARM Cortex-R4F test chip system parameters
- reading of data from the ADCs that monitor the ARM Cortex-R4F test chip voltages and currents
- configuring the CLCD and DVI interface.

The CT-R4F PLD is controlled by the serial interface signals listed in Table 3-8. These signals connect to the EB via the HDRZ header.

Table 3-8 PLD control signals

Signal	Direction	HDRZ bus	Description
PLDCLK	Output	**ZL230**	Clocks data into or out of the PLD.
PLDDIN	Input	**ZL227**	Serial data input to PLD.
PLDDOUT	Output	**ZL228**	Serial data output from PLD.
PLDnRST	Input	**ZL229**	Resets the serial interface and signals the start of transfers.

The EB system FPGA implements registers that hold values sent to or received from the CT-R4F PLD using the 4-wire serial interface. The EB system FPGA and the CT-R4F PLD provide the serialization and deserialization logic required for the interface. The interface timing is shown in Figure 3-9 on page 3-26.

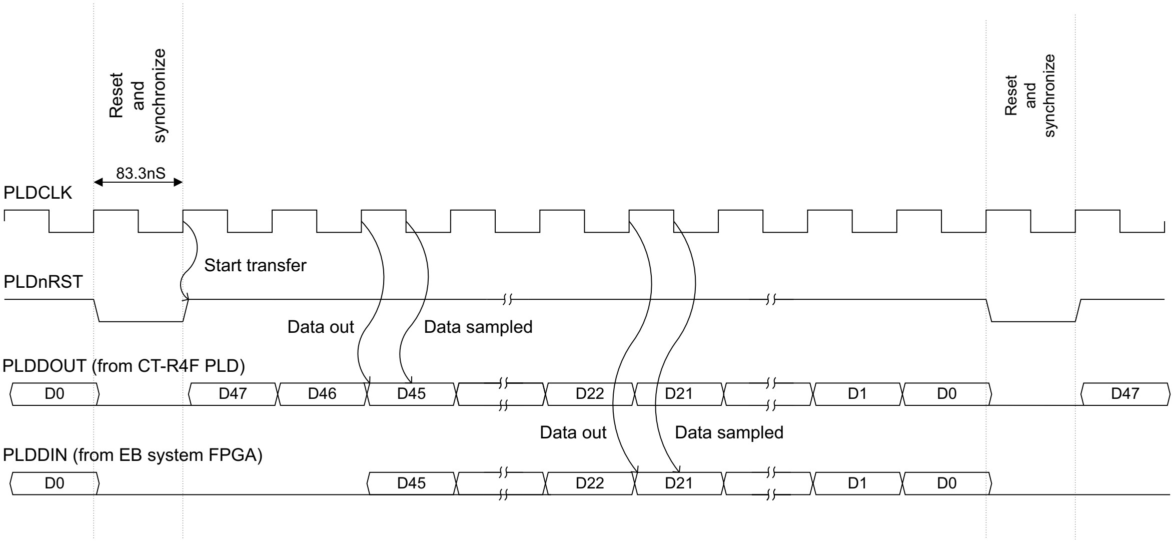


Figure 3-9 4-wire serial interface timing

Data is output on the rising edge of **PLDCLK** and sampled on the falling edge. The interface is reset and re-synchronized by **PLDnRST** after each complete serial transfer. The rising edge of **PLDnRST** also indicates the start of a transfer. The data is transferred MSB first in both directions across the interface.

There are a total of 48 data bits (timeslots) in each bidirectional serial data stream:

The serial data stream **PLDDOUT** transmits data to the EB FPGA from the CT-R4F PLD *Serial data read register*s.

The serial data stream **PLDDIN** transmits data to the CT-R4F PLD *Serial data write registers* from the EB FPGA. Bits **PLDDIN[47:46]** are reserved and tied LOW.

The generic bit allocations for the serial streams are listed in Table 3-9.

Table 3-9 Serial streams bit allocations

Bit	Serial Bits PLDDIN	Serial Bits PLDDOUT
0	DATA_DIR	BOARD_ID[0]
1	ADDR[12]	BOARD_ID[1]
2	ADDR[11]	BOARD_ID[2]
3	ADDR[10]	BOARD_ID[3]
4	ADDR[9]	BOARD_ID[4]
5	ADDR[8]	BOARD_ID[5]
6	ADDR[7]	BOARD_ID[6]
7	ADDR[6]	BOARD_ID[7]
8	ADDR[5]	BOARD_ID[8]
9	ADDR[4]	BOARD_ID[9]
10	ADDR[3]	TC_PLL0_LOCK
11	ADDR[2]	TC_PLL1_LOCK
12	ADDR[1]	TC_PLL2_LOCK
13	ADDR[0]	ISP0_LOCKn
14	DATA_IN[31]	ISP1_LOCKn
15	DATA_IN[30]	ISP2_LOCKn
16	DATA_IN[29]	DATA_OUT[0]
17	DATA_IN[28]	DATA_OUT[1]
18	DATA_IN[27]	DATA_OUT[2]
19	DATA_IN[26]	DATA_OUT[3]
20	DATA_IN[25]	DATA_OUT[4]
21	DATA_IN[24]	DATA_OUT[5]
22	DATA_IN[23]	DATA_OUT[6]
23	DATA_IN[22]	DATA_OUT[7]

Table 3-9 Serial streams bit allocations (continued)

Bit	Serial Bits PLDDIN	Serial Bits PLDDOUT
24	DATA_IN[21]	DATA_OUT[8]
25	DATA_IN[20]	DATA_OUT[9]
26	DATA_IN[19]	DATA_OUT[10]
27	DATA_IN[18]	DATA_OUT[11]
28	DATA_IN[17]	DATA_OUT[12]
29	DATA_IN[16]	DATA_OUT[13]
30	DATA_IN[15]	DATA_OUT[14]
31	DATA_IN[14]	DATA_OUT[15]
32	DATA_IN[13]	DATA_OUT[16]
33	DATA_IN[12]	DATA_OUT[17]
34	DATA_IN[11]	DATA_OUT[18]
35	DATA_IN[10]	DATA_OUT[19]
36	DATA_IN[9]	DATA_OUT[20]
37	DATA_IN[8]	DATA_OUT[21]
38	DATA_IN[7]	DATA_OUT[22]
39	DATA_IN[6]	DATA_OUT[23]
40	DATA_IN[5]	DATA_OUT[24]
41	DATA_IN[4]	DATA_OUT[25]
42	DATA_IN[3]	DATA_OUT[26]
43	DATA_IN[2]	DATA_OUT[27]
44	DATA_IN[1]	DATA_OUT[28]
45	DATA_IN[0]	DATA_OUT[29]
46	Not used	DATA_OUT[30]
47	Not used	DATA_OUT[31]

3.9 Memory system

The CT-R4F memory system consists of:

- ARM Cortex-R4F processor implemented with 16KB of L1 caches (I and D) and 64KB of TCM (A, B0 and B1).
- PrimeCell PL340 memory controller implemented in the ARM Cortex-R4F test chip with 512MB of external SDRAM provided on the CT-R4F. The SDRAM is organized as two banks of 32M x 64 wide memory.

——— **Note** ———

There is no provision for PISMO Expansion Memory Modules on the CT-R4F. If additional main memory is required, you can add PISMO Expansion Memory Modules to the EB. Additional main memory might also be available via the EB second tile site, see *Realview Emulation Baseboard User Guide (Lead Free)* (ARM DUI 0411) for system memory details.

3.9.1 Memory Remap

The SDRAM on the CT-R4F together with the memory mapped peripherals can be remapped within the ARM Cortex-R4F address space. The four available memory mappings are detailed in *Memory maps* on page 4-2.

3.10 JTAG support

JTAG ICE signals are present on the HDRZ header. An external baseboard provides the JTAG connector and the routing of the JTAG ICE signals from the connector to the HDRZ header. The CT-R4F routes the JTAG scan path through scan chain enabled devices on the board. The logic devices that are placed in the CT-R4F scan chain depend on the JTAG mode:

Debug mode Debug mode is selected when the CONFIG slide-switch on the EB is set to OFF (**nCFGEN** is HIGH). It is the default mode used for general system development and debug. In this mode, the JTAG signals flow through the Debug Scan Chain (this scan chain connects to the ARM Cortex-R4 test chip DAP only). The JTAG signals used for debug are identified by the **D_** prefix.

Configuration mode

Configuration mode is selected when the CONFIG slide-switch on the EB is set to ON (**nCFGEN** is LOW). This mode enables the programmable logic devices in the system to be reprogrammed. The CT-R4F reroutes the JTAG scan path to include the CT-R4F PLD, ispClock generators, and the test chip registers that are made available in the configuration scan chain. The JTAG signals used for configuration are identified by the **C_** prefix.

Table 3-10 on page 3-31 provides a description of the JTAG signals. See *HDRZ signals* on page 5-18 for the JTAG pinout on the HDRZ header.

Note

In the description in Table 3-10 on page 3-31, the term JTAG equipment refers to any hardware that can drive the JTAG signals to devices in the scan chain. Typically, RealView ICE is used for configuration and debug, although you can also use hardware from third-party suppliers to debug ARM processors.

Table 3-10 JTAG signal description

Name	Description	Function
EDBGRQ	External debug request (to CT-R4F PLD)	**EDBGRQ** is a request to theARM Cortex-R4F processor to enter the debug state.
DBGACK	Debug acknowledge (to JTAG equipment)	**DBGACK[3:0]** indicates to the debugger that the ARM Cortex-R4F processor has entered debug state.
nCFGEN	Configuration enable (controlled by config slide-switch on the baseboard)	**nCFGEN** is an active LOW signal used to put the boards into configuration mode. In configuration mode all FPGAs and PLDs that are connected to the Config Scan Chain can be configured by the JTAG equipment.
nTRST: **D_nTRST**, **C_nTRST**	Test reset (from JTAG equipment)	This active LOW open-collector signal is used to reset the JTAG port and the associated debug circuitry on the processor. It is asserted at power-up by each module, and can be driven by the JTAG equipment. This signal is also used in configuration mode to control the programming pin (**nPROG**) on FPGAs. **D_nTRST** is the reset for the debug mode scan chain and **C_nTRST** is the reset for the configuration mode scan chain. **Note** **D_nTRST** is always tied to **D_nSRST**. **C_nTRST** is tied to **D_nTRST** when configuration mode is enabled.
RTCK: **D_RTCK**	Return TCK (to JTAG equipment)	Some devices sample **TCK** (for example a synthesizable core with only one clock), and this has the effect of delaying the time when a component actually captures data. **RTCK** is a mechanism for returning the sampled clock to the JTAG equipment, so that the clock is not advanced until the synchronizing device has captured the data. In a multiple device JTAG chain, the **D_RTCK** output from a component connects to the **TCK** input of the down-stream device. The **RTCK** signal on the EB connector HDRZ returns **TCK** to the JTAG equipment. **D_RTCK** is the **RTCK** signal in the debug scan chain. **RTCK** is not available in the configuration mode scan chain.

Table 3-10 JTAG signal description (continued)

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

3.10.1 JTAG debug scan chain routing

In JTAG debug (normal) mode only the ARM Cortex-R4F test chip DAP is connected in the debug scan chain. JTAG debug mode routing is selected when **nCFGEN** is HIGH. This is the default baseboard setting.

Note

Serial Wire Debug (SWD) is also supported on the CT-R4F at the Trace A connector. See *CoreSight Serial Wire Debug* on page 2-6 for details.

3.10.2 JTAG configuration scan chain routing

During configuration all the JTAG configurable devices (CT-R4F PLD, ISP clock generators, and the ARM Cortex-R4F test chip DAP) are connected in the configuration scan chain. JTAG configuration routing is selected when **nCFGEN** is LOW.

Figure 3-10 shows the JTAG configuration scan chain routing

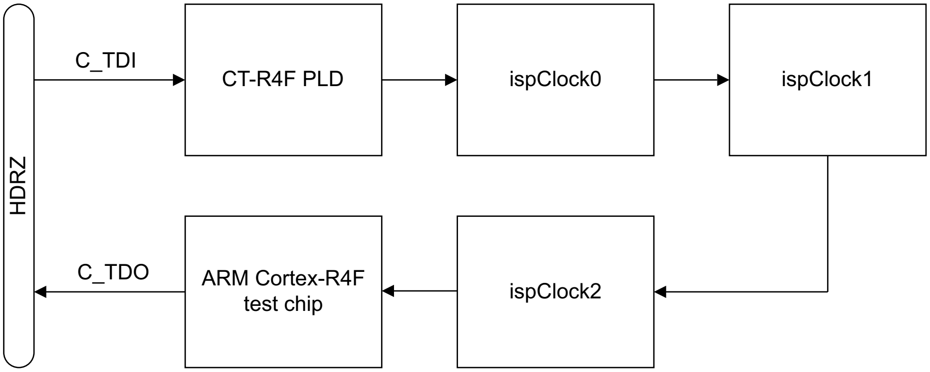


Figure 3-10 JTAG configuration routing

3.10.3 CoreSight sub-system

A simplified diagram of the CoreSight sub-system is shown in Figure 3-11. Each of the system elements is listed and briefly described in Table 3-11 on page 3-35. For information on CoreSight, see *CoreSight Technology System Design Guide* (ARM DGI 0012) and for information on the individual CoreSight components, see *CoreSight Components Technical Reference Manual* (ARM DDI 0314).

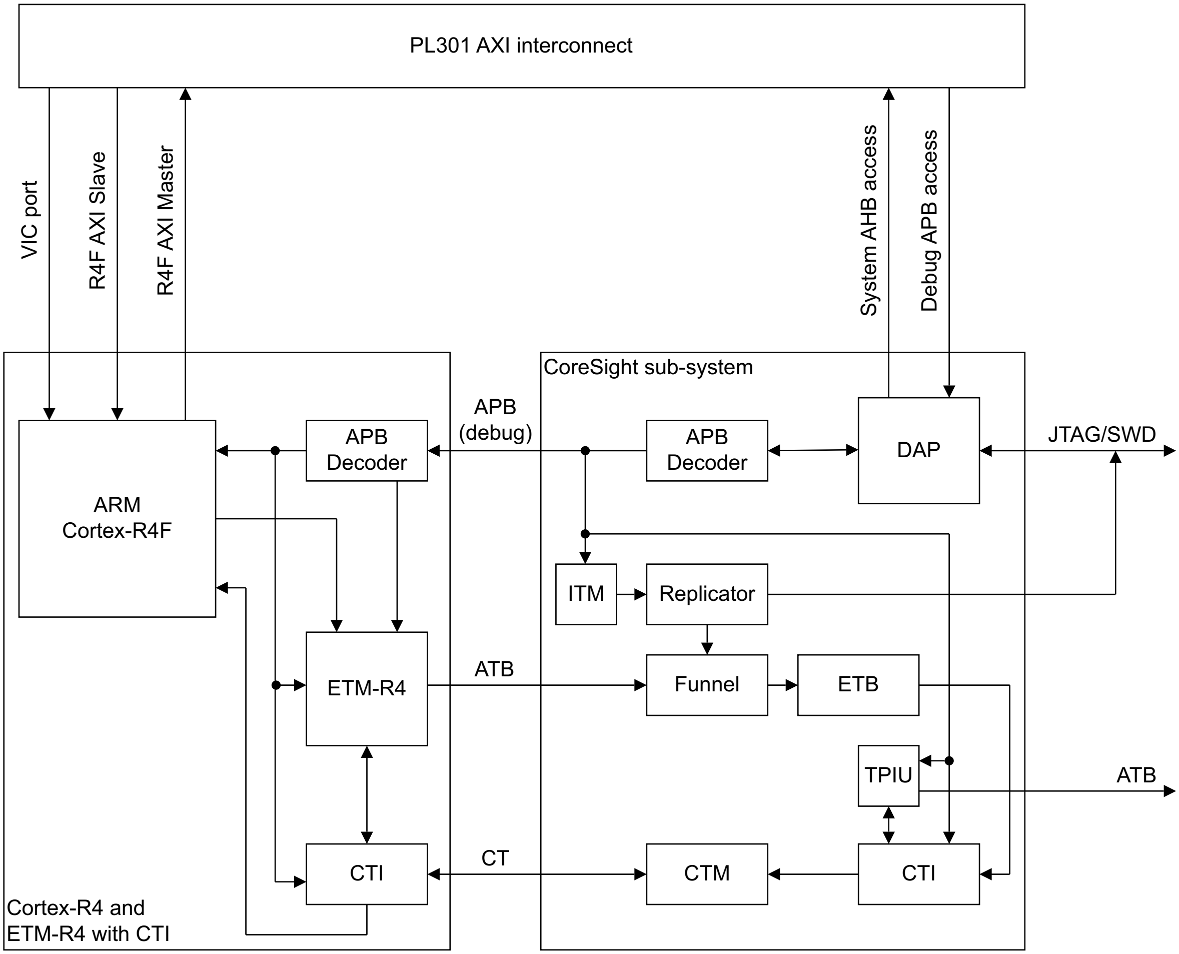


Figure 3-11 CoreSight sub-system

Note

The DAP ROM in the ARM Cortex-R4F test chip is not operational. For information on setting up RVI for use with the Debug APB see Application Note 217 *Using a CT-R4F with the RealView Emulation Baseboard* supplied on the Versatile CD or visit the FAQ section of the ARM website.

Table 3-11 CoreSight components

CoreSight Component	Base Address
	REMAP = 000
	REMAP = 010
		Base Address
		REMAP = 001
		REMAP = 100
			Description

CTI CPU_ETM)	`0xE0029000`	`0xD0029000`	Cross Trigger Interface, part of the *Embedded Cross Trigger* (ECT). See *CoreSight Components Technical Reference Manual* (ARM DDI 0314) for details.
ETM-R4	`0xE0028000`	`0xD0028000`	Embedded Trace MacroCell, provides instruction trace and data trace for the ARM Cortex-R4F processor. See *CoreSight ETM-R4 Technical Reference Manual* (ARM DDI 0367) for further details.
R4F _APB	`0xE0027000`	`0xD0027000`	The ARM Cortex-R4F processor debug unit APB slave port. See *Cortex-R4 and Cortex-R4F Technical Reference Manual* (ARM DDI 0363) for details.
SWD	`0xE0026000`	`0xD0026000`	Serial Wire Debug unit, part of the *Debug Access Port* (DAP).
ITM	`0xE0025000`	`0xD0025000`	Instrumentation Trace Macrocell, with supporting code, generates *SoftWare Instrumentation Trace* (SWIT).
Funnel	`0xE0024000`	`0xD0024000`	CoreSight Trace Funnel, used when there is more than one trace source. The CSTF combines multiple trace streams into a single *AMBA Trace Bus* (ATB).
TPIU	`0xE0023000`	`0xD0023000`	Trace Port Interface Unit, acts as a bridge between the on-chip trace data and the data stream captured by the external *Trace Port Analyzer* (TPA).
CTI (CoreSight sub-system)	`0xE0022000`	`0xD0022000`	Cross Trigger Interface, part of the *Embedded Cross Trigger* (ECT). See *CoreSight Components Technical Reference Manual* (ARM DDI 0314) for details.

Table 3-11 CoreSight components (continued)

CoreSight Component	Base Address
	REMAP = 000
	REMAP = 010
		Base Address
		REMAP = 001
		REMAP = 100
			Description

ETB	`0xE0021000`	`0xD0021000`	Embedded Trace Buffer, provides on-chip storage of trace data using 32-bit RAM.
DAP_ROM	`0xE0020000`	`0xD0020000`	Internal ROM table, stores the locations of the components on the Debug APB.
Replicator	–	–	ATB replicator, sends identical trace data from an incoming ATB slave port interface to two outgoing ATB master port interfaces.

Chapter 4
Programmer's Reference

This chapter describes the memory map and the 4-wire serial configuration interface for the CT-R4F. It contains the following sections.

- *Memory maps* on page 4-2
- *CT-R4F configuration* on page 4-15
- *Board level control and status* on page 4-27
- *Power supply monitoring* on page 4-28
- *Board identification* on page 4-30.

Note

See the *Cortex-R4 and Cortex-R4F Technical Reference Manual* (ARM DDI 0363) for details on the ARM Cortex-R4F processor implemented in the test chip.

4.1 Memory maps

There are four memory maps available for the CT-R4F local memory and memory mapped peripherals. The memory map selected is determined by the REMAP[2:0] value in the CT_R4F_TC_CFG2 register. The value is set by the baseboard using the 4-wire serial interface to the CT-R4F. The remap options are:

REMAP = b000 This is the regular configuration memory map. In this configuration, the 512MB of local SDRAM memory appears aliased at the top of memory beginning at `0xE10A0000` and ending at `0xFFFFFFFF`. The memory map is shown in Figure 4-1 on page 4-3.

REMAP = b001 This configuration moves the local SDRAM and memory mapped peripherals down the memory map 1G, mapping the top 1G of memory to the CT-R4F AXI master port. This mode supports another slave tile at `0xF0000000`. The 512MB of local SDRAM memory appears aliased beginning at `0xD10A0000` and ending at `0xEFFFFFFF`. The memory map is shown in Figure 4-2 on page 4-4.

REMAP = b010 In this configuration, the bottom 256MB of the memory map is mapped to the on-board SDRAM. The 512MB of local SDRAM memory is also aliased at the top of memory beginning at `0xE10A0000` and ending at `0xFFFFFFFF`. The memory map is shown in Figure 4-3 on page 4-5.

REMAP = b100 This configuration moves the local SDRAM and memory mapped peripherals down the memory map 1G, mapping the top 1G of memory to the CT-R4F AXI master port. This mode supports another slave tile at `0xF0000000`. The bottom 256MB of the memory map is mapped to the on-board SDRAM. The 512MB of local SDRAM memory appears aliased beginning at `0xD10A0000` and ending at `0xEFFFFFFF`. The memory map is shown in Figure 4-4 on page 4-6.

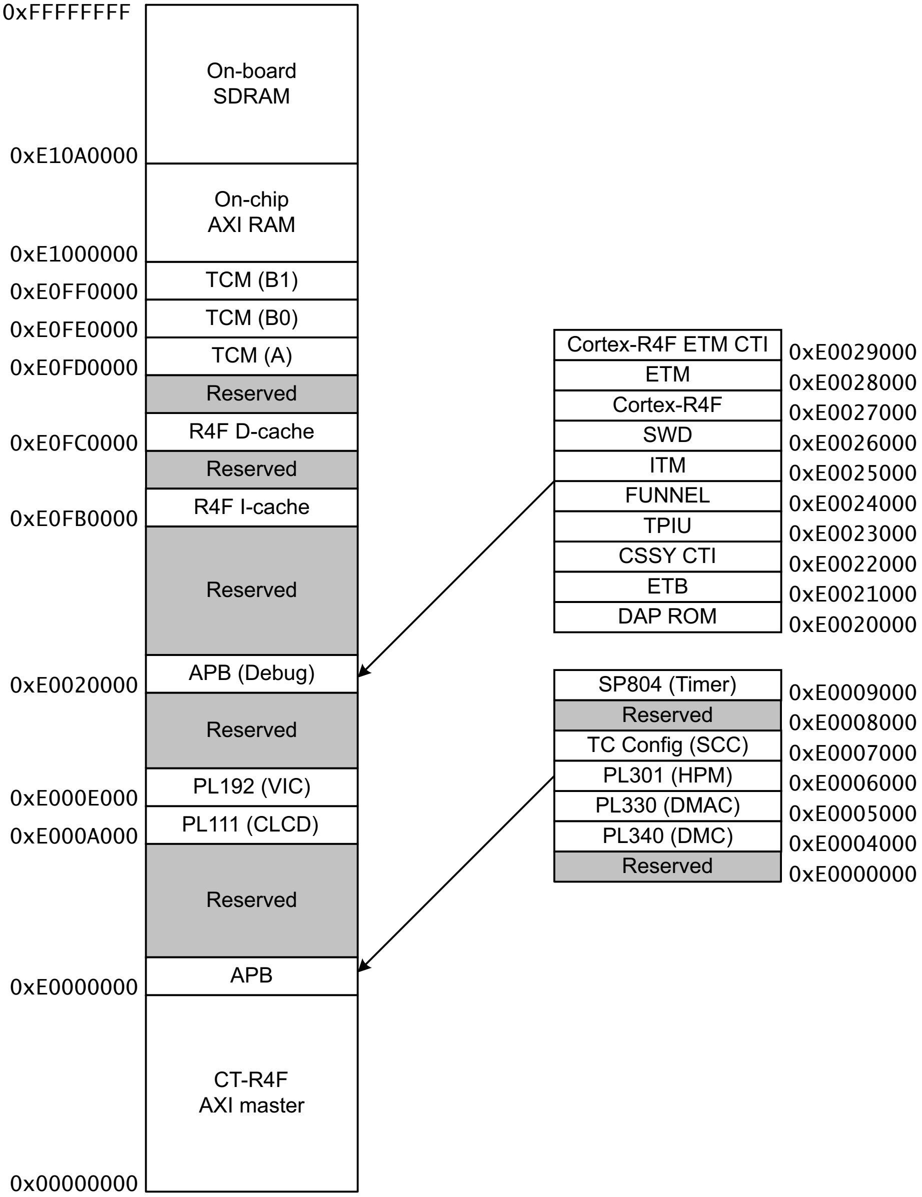


Figure 4-1 Memory map (REMAP = 000)

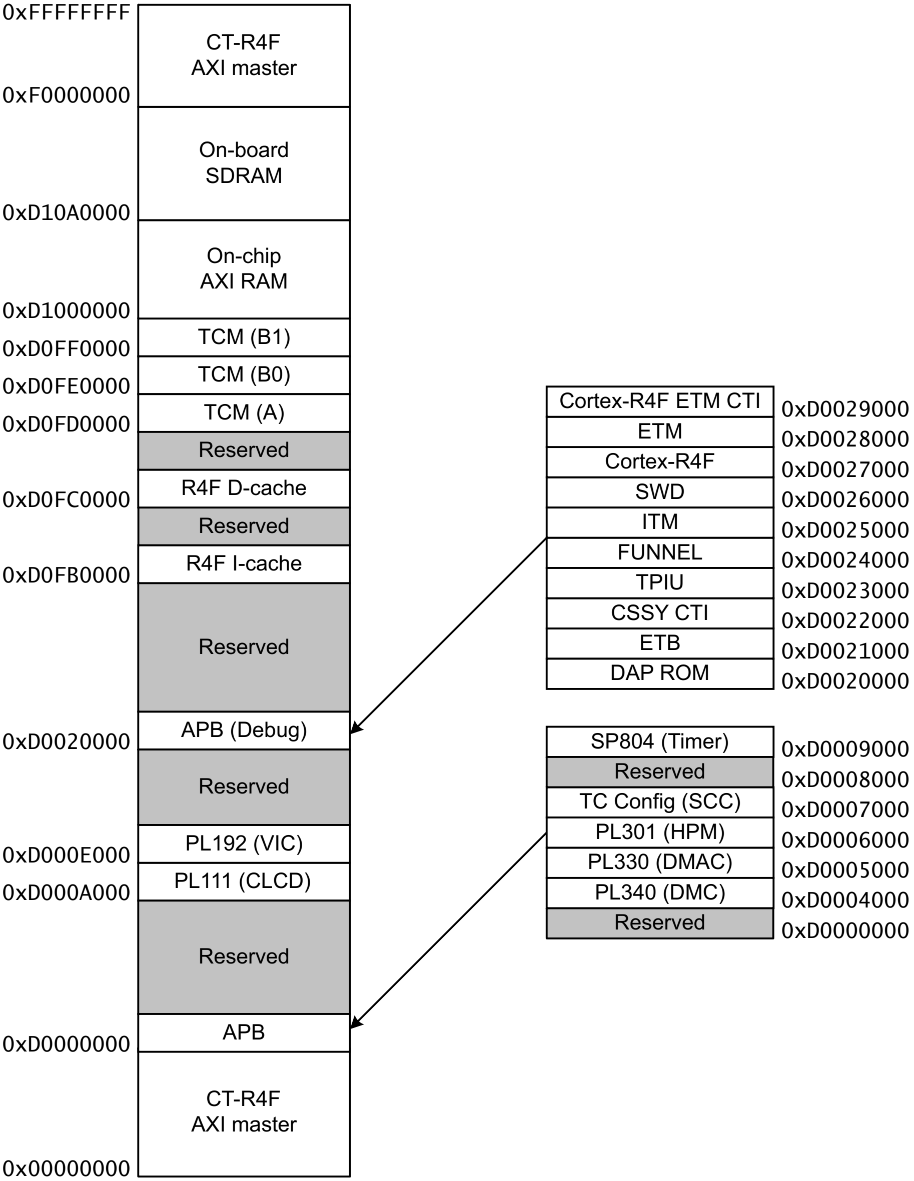


Figure 4-2 Memory map (REMAP = 001)

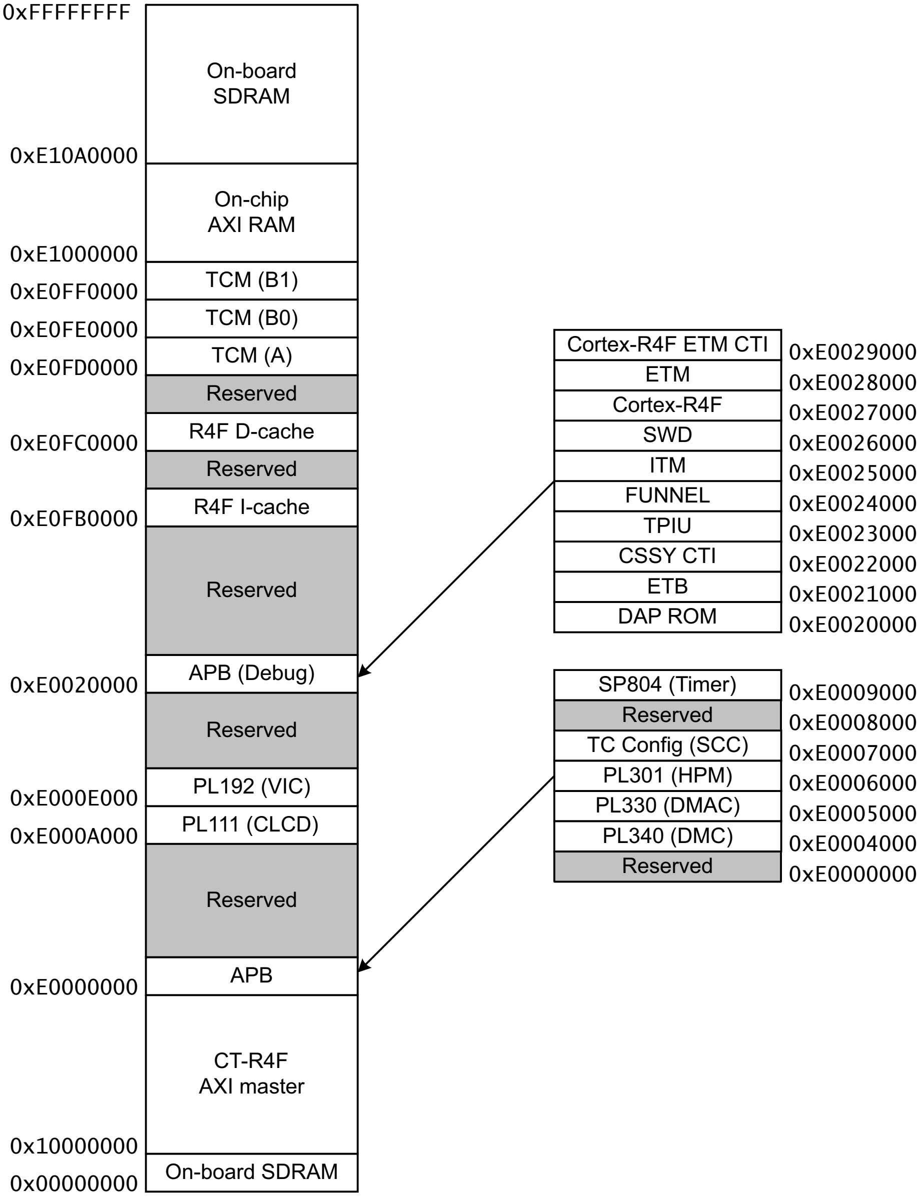


Figure 4-3 Memory map (REMAP = 010)

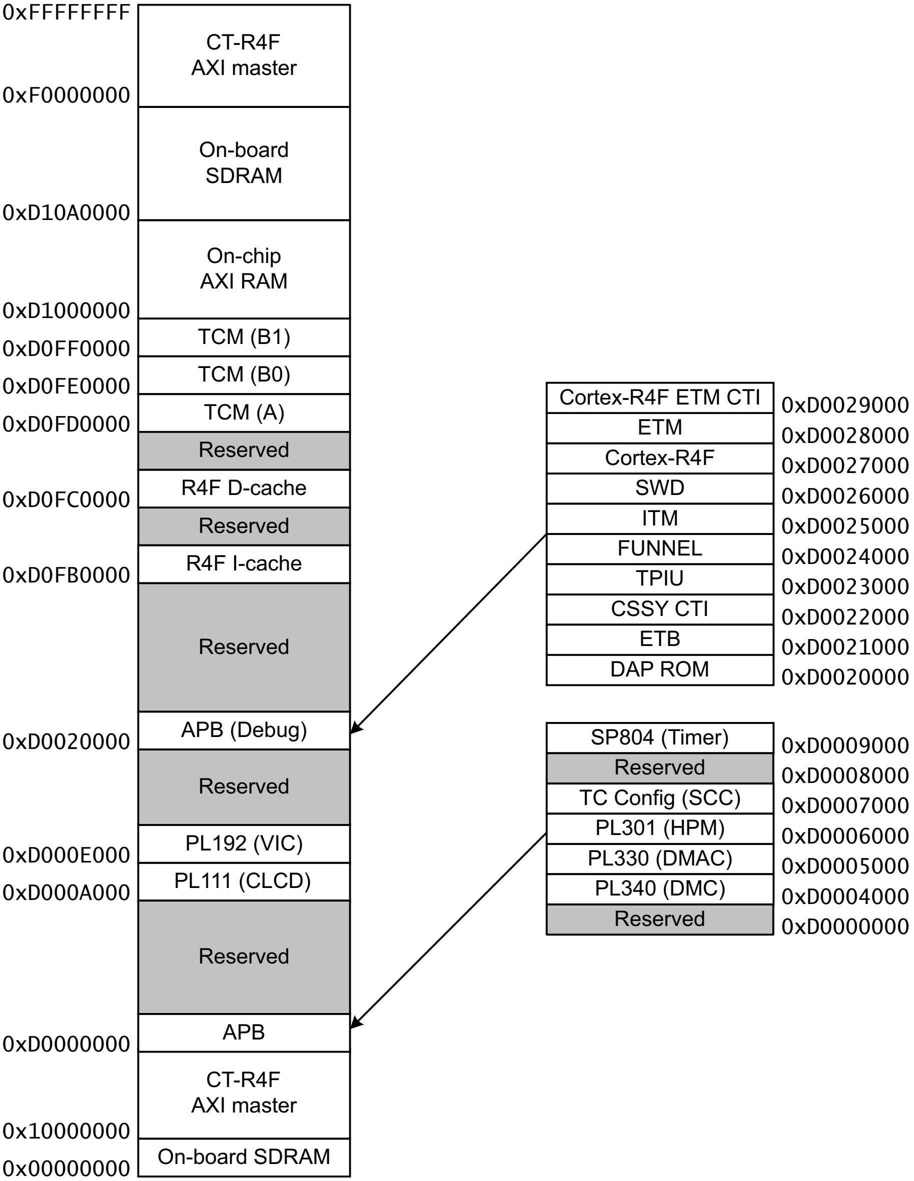


Figure 4-4 Memory map (REMAP = 100)

4.1.1 Peripheral memory maps

The peripheral memory maps for the four remap options are listed in Table 4-1 and Table 4-2 on page 4-8. Not all configuration options for the ARM Cortex-R4F processor can be set using the *Serial Configuration Controller* (SCC) APB port listed in Table 4-1 and Table 4-2 on page 4-8, some can only be set from the SCC serial configuration port during system reset.

Table 4-1 Peripheral memory map (REMAP = 000 or 010)

Peripheral	Address range (REMAP = 000 or 010)	Bus type	Region size
Reserved	`0xE0000000-0xE0003FFF`	APB	16KB
Dynamic Memory Controller (PL340)	`0xE0004000-0xE0004FFF`	APB	4KB
Direct Memory Access Controller (PL330)	`0xE0005000-0xE0005FFF`	APB	4KB
High-Performance Matrix (PL301)	`0xE0006000-0xE0006FFF`	APB	4KB
Serial Configuration Controller (SCC)	`0xE0007000-0xE0007FFF`	APB	4KB
Reserved	`0xE0008000-0xE0008FFF`	APB	4KB
Timer (SP804)	`0xE0009000-0xE0009FFF`	APB	4KB
Color LCD Controller (PL111)	`0xE000A000-0xE000AFFF`	AHB	4KB
Reserved (PL111 aliased)	`0xE000B000-0xE000DFFF`	AHB	12KB
Vectored Interrupt Controller (PL192)	`0xE000E000-0xE000EFFF`	AHB	4KB
Reserved (PL192 aliased)	`0xE000F000-0xE0011FFF`	AHB	12KB

Table 4-2 Peripheral memory map (REMAP = 001 or 100)

Peripheral	Address range (REMAP = 001 or 100)	Bus type	Region size
Reserved	`0xD0000000-0xD0003FFF`	APB	16KB
Dynamic Memory Controller (PL340)	`0xD0004000-0xD0004FFF`	APB	4KB
Direct Memory Access Controller (PL330)	`0xD0005000-0xD0005FFF`	APB	4KB
High-Performance Matrix (PL301)	`0xD0006000-0xD0006FFF`	APB	4KB
Serial Configuration Controller (SCC)	`0xD0007000-0xD0007FFF`	APB	4KB
Reserved	`0xD0008000-0xD0008FFF`	APB	4KB
Timer	`0xD0009000-0xD0009FFF`	APB	4KB
Color LCD Controller (PL111)	`0xD000A000-0xD000AFFF`	AHB	4KB
Reserved (PL111 aliased)	`0xD000B000-0xD000DFFF`	AHB	12KB
Vectored Interrupt Controller (PL192)	`0xD000E000-0xD000EFFF`	AHB	4KB
Reserved (PL192 aliased)	`0xD000F000-0xD0011FFF`	AHB	12KB

Dynamic Memory Controller (DMC)

The PL340 PrimeCell *Dynamic Memory Controller* (DMC) is an AMBA compliant SoC peripheral that is developed, tested, and licensed by ARM Limited.

The DMC implementation controls 512MB of 64-bit wide SDRAM organized as two banks.

Table 4-3 DMC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE0004000` • REMAP = 001 or 100: `0xD0004000`
Interrupt	–
DMA	–
Release version	ARM DMC PL340 r1p0
Reference documentation	*PrimeCell Dynamic Memory Controller (PL340) Technical Reference Manual* DDI 0331.

The DMC controls 2 banks of 512MB SDRAM memory on the CT-R4F. Sample programs that configure and use dynamic memory can be found on the CD that accompanies the Core Tile.

Direct Memory Access Controller (DMAC)

The PL330 PrimeCell *Direct Memory Access Controller* (DMAC) is an AMBA compliant SoC peripheral that is developed, tested, and licensed by ARM Limited.

The DMAC can access the local SDRAM, test chip AXI RAM, the ARM Cortex-R4F processor slave port, and the CT-R4F slave port. The PL330 is configured to support 4 banks of SDRAM with a memory bus width of 64-bit. The peripheral request interface is not implemented.

Table 4-4 DMAC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE0005000` • REMAP = 001 or 100: `0xD0005000`
Interrupt	**VICINTSOURCE[2]** TC_DMAC_IRQ0 **VICINTSOURCE[3]** TC_DMAC_IRQ1 **VICINTSOURCE[4]** TC_DMAC_IRQ2 **VICINTSOURCE[5]** TC_DMAC_IRQ3 **VICINTSOURCE[6]** TC_DMAC_IRQ_ABORT
Release version	ARM DMAC PL330 r0p0
Reference documentation	*PrimeCell DMA Controller (PL330) Technical Reference Manual* DDI 0424.

High Performance Matrix (HPM)

The PL301 PrimeCell *High Performance Matrix* (HPM) is an AMBA compliant SoC peripheral that is developed, tested, and licensed by ARM Limited.

The HPM is a highly configurable auto-generated AMBA 3 bus subsystem, based around a high-performance AXI cross-bar switch known as the AXI bus matrix, and extended by AMBA infrastructure components.

Table 4-5 DMAC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE0006000` • REMAP = 001 or 100: `0xD0006000`
Interrupt	–
Release version	ARM HPM PL301 r1p0
Reference documentation	*PrimeCell High-Performance Matrix (PL301) Technical Reference Manual* DDI 0397.

Serial Configuration Controller (SCC)

The *Serial Configuration Controller* (SCC) is a custom AMBA compliant SoC peripheral.

During reset, the ARM Cortex-R4F test chip is configured by the CT-R4F PLD via the SCC serial configuration port. After reset, there is limited access to the SCC internal configuration registers at the SCC APB port.

Table 4-6 SCC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE0007000` • REMAP = 001 or 100: `0xD0007000`
Interrupt	–
Release version	ARM Custom IP
Reference documentation	See *CT-R4F configuration* on page 4-15 for configuration register details.

Timer

The SP804 *Dual-Timer Module* is an AMBA compliant SoC peripheral that is developed and tested by ARM Limited.

The Dual-Timer Module consists of two programmable 32/16-bit down counters that can generate interrupts on reaching zero.

Table 4-7 Timer implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE0009000` • REMAP = 001 or 100: `0xD0009000`
Interrupt	**VICINTSOURCE[18]** TC_TIMINT1 **VICINTSOURCE[19]** TC_TIMINT2 **VICINTSOURCE[20]** TC_TIMINTC
Release version	ARM Dual-Timer SP804 r1p2
Reference documentation	*ARM Timer Module (SP804) Technical Reference Manual* ARM DDI 0271

Color LCD Controller (CLCDC)

The PL111 PrimeCell *Color LCD Controller* (CLCDC) is an AMBA compliant SoC peripheral that is developed, tested, and licensed by ARM Limited.

The CLCDC master can access the local SDRAM, test chip AXI RAM, and the CT-R4F slave port.

Table 4-8 CLCDC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE000A000` • REMAP = 001 or 100: `0xD000A000`
Interrupt	**VICINTSOURCE[7]** CLCDINTR
Release version	ARM CLCDC PL111 (version r0p0)
Reference documentation	*ARM PrimeCell Color LCD Controller (PL111) Technical Reference Manual* DDI 0293.

Vectored Interrupt Controller (VIC)

The PL192 PrimeCell *Vectored Interrupt Controller* (VIC) is an AMBA compliant SoC peripheral that is developed, tested, and licensed by ARM Limited.

Table 4-9 CLCDC implementation

Property	Value
Location	ARM Cortex-R4F test chip
Memory base address	• REMAP = 000 or 010: `0xE000E000` • REMAP = 001 or 100: `0xD000E000`
Interrupt	**VICINTSOURCE[7]** CLCDINTR
Release version	ARM CLCDC PL111 (version r0p0)
Reference documentation	*ARM PrimeCell Color LCD Controller (PL111) Technical Reference Manual* DDI 0293.

4.2 CT-R4F configuration

To maintain compatibility with other board products, the standard 4-wire serial interface implemented on ARM baseboards sends and receives data to and from registers in the CT-R4F PLD. See *Overview of Core Tile configuration* on page 3-24 for details of the 4-wire serial interface implementation and the format of the address and data stream. The CT-R4F PLD registers and the serial stream addresses are listed in Table 4-10.

Table 4-10 CT-R4F PLD registers

Address	Name	Access	Description
`0x0000`	CT_R4F_TC_CFG0	Write	CT-R4F test chip config word 0. See *ARM Cortex-R4F test chip configuration at reset* on page 4-16 for details.
`0x0004`	CT_R4F_TC_CFG1	Write	CT-R4F test chip config word 1.
`0x0008`	CT_R4F_TC_CFG2	Write	CT-R4F test chip config word 2.
`0x1000`	CT_R4F_PLD_ID	Read	CT-R4F PLD ID. See *CT-R4F ID register* on page 4-30 for details.
`0x1004`	CT_R4F_OSC0	Write	CT-R4F test chip **CPU_CLK** PLL reference clock.
`0x1005`	CT_R4F_OSC1	Write	CT-R4F test chip **MCLK** PLL reference clock.
`0x1006`	CT_R4F_OSC2	Write	CT-R4F test chip CLCD reference clock.
`0x1008`	CT_R4F_DVI	Write	CT-R4F DVI and CLCD control. See *CT_RF_DVI register* on page 4-26 for details.
`0x100A`	CT_R4F_CTRL	Read/Write	CT-R4F control. See *CT_R4F_CTRL register* on page 4-27 for details.
`0x1010`	CT_R4F_CTRL_ADC0	Read	CT-R4F ADC channel 0 – ARM Cortex-R4F processor current. See *CT_R4F_ADC registers* on page 4-28 for details.
`0x1011`	CT_R4F_CTRL_ADC1	Read	CT-R4F ADC channel 1 – Reserved.
`0x1012`	CT_R4F_CTRL_ADC2	Read	CT-R4F ADC channel 2 – test chip sub-system current.
`0x1013`	CT_R4F_CTRL_ADC3	Read	CT-R4F ADC channel 3 – ARM Cortex-R4F processor and test chip sub-system supply voltage.
`0x1014`	CT_R4F_CTRL_ADC4	Read	CT-R4F ADC channel 4 – test chip PLLs supply voltage.

Table 4-10 CT-R4F PLD registers (continued)

Address	Name	Access	Description
`0x1015`	CT_R4F_CTRL_ADC5	Read	CT-R4F ADC channel 5 – test chip I/O supply voltage.
`0x1016`	CT_R4F_CTRL_ADC6	Read	CT-R4F ADC channel 6 – Reserved.
`0x1017`	CT_R4F_CTRL_ADC7	Read	CT-R4F ADC channel 7 – Reserved.

4.2.1 ARM Cortex-R4F test chip configuration at reset

Because of the limited pin count and extensibility of the ARM Cortex-R4F test-chip design, all static configuration inputs are programmed serially. During reset, the CT-R4F PLD configures the ARM Cortex-R4F test chip by transferring data from the CT_R4F_TC_CFG[0:2] registers in the CT-R4F PLD using the serial interface port of the *Serial Configuration Controller* (SCC) in the test chip.

The default values are supplied by the EB via the 4-wire serial interface during reset.

Note

Some configuration data can be changed using the APB port of the SCC after reset. See *ARM Cortex-R4F test chip configuration after reset* on page 4-20 for details.

The CT_R4F_TC_CFG0 register configuration signals that are sent serially during reset are listed in Table 4-11. The CT_R4F_TC_CFG0 register address is: `0x0000`.

Table 4-11 CT_R4F_TC_CFG0 configuration signals

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[3:0]	**CONFIG[3:0]**	**M_PLL0**	b0101	ARM Cortex-R4F test chip PLL0 setup
[6:4]	**CONFIG[6:4]**	**N_PLL0**	b001	
[7]	**CONFIG[7]**	**BYPASS_PLL0**	b0	
[8]	**CONFIG[8]**	**ENABLE_PLL0**	b1	
[9]	**CONFIG[9]**	**DESKEW_PLL0**	b0	
[10]	**CONFIG[10]**	**RANGE_PLL0**	b1	
[11]	**CONFIG[11]**	**MACRO_BYPASS_PLL0**	b0	

Table 4-11 CT_R4F_TC_CFG0 configuration signals (continued)

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[15:12]	**CONFIG[15:12]**	**M_PLL1**	b0001	ARM Cortex-R4F test chip PLL1 setup PLL1 is bypassed by default
[18:16]	**CONFIG[18:16]**	**N_PLL1**	b001	
[19]	**CONFIG[19]**	**BYPASS_PLL1**	b1	
[20]	**CONFIG[20]**	**ENABLE_PLL1**	b0	
[21]	**CONFIG[21]**	**DESKEW_PLL1**	b1	
[22]	**CONFIG[22]**	**RANGE_PLL1**	b0	
[23]	**CONFIG[23]**	**MACRO_BYPASS_PLL1**	b0	
[27:24]	**CONFIG[27:24]**	**M_PLL2**	b0001	ARM Cortex-R4F test chip PLL2 setup
[30:28]	**CONFIG[30:28]**	**N_PLL2**	b001	
[31]	**CONFIG[31]**	**BYPASS_PLL2**	b0	

The CT_R4F_TC_CFG1 register configuration signals that are sent serially during reset are listed in Table 4-12. The CT_R4F_TC_CFG1 register address is: `0x0004`.

Table 4-12 CT_R4F_TC_CFG1 configuration signals

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[0]	**CONFIG[32]**	**ENABLE_PLL2**	b1	ARM Cortex-R4F test chip PLL2 setup
[1]	**CONFIG[33]**	**DESKEW_PLL2**	b1	
[2]	**CONFIG[34]**	**RANGE_PLL2**	b0	
[3]	**CONFIG[35]**	**MACRO_BYPASS_PLL2**	b0	
[5:4]	**CONFIG[37:36]**	**CLOCK_MODE**	b01	**CPU_CLK**:**ACLK**:**PCLK** frequency ratio: 00 = 8:8:1 01 = 16:8:1 10 = 16:4:1 11 = 16:2:1
[6]	**CONFIG[38]**	**CFGEE**	b0	
[7]	**CONFIG[39]**	**CFGIE**	b0	

Table 4-12 CT_R4F_TC_CFG1 configuration signals (continued)

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[8]	**CONFIG[40]**	**CPUHALTn**	b1	
[9]	**CONFIG[41]**	**ENTCM1F**	b1	
[10]	**CONFIG[42]**	**ETMDBGEN**	b1	Invasive Debug Enable (ETM)
[11]	**CONFIG[43]**	**ETMNIDEN**	b1	Non-Invasive Debug Enable (ETM)
[12]	**CONFIG[44]**	**INITRAMA**	b0	Enable ATCM boot
[13]	**CONFIG[45]**	**INITRAMB**	b0	Enable BTCM boot
[14]	**CONFIG[46]**	**LOCZRAMA**	b1	
[15]	**CONFIG[47]**	**NIDEN**	b1	Non-Invasive Debug Enable (CPU)
[16]	**CONFIG[48]**	**SLBTCMSB**	b0	
[17]	**CONFIG[49]**	**TEINIT**	b0	
[18]	**CONFIG[50]**	**VINITHI**	b0	
[19]	**CONFIG[51]**	**CFGNMFI**	b0	
[20]	**CONFIG[52]**	**DBGNOCLKSTOP**	b0	The processor does not stop the clocks when entering WFI state
[21]	**CONFIG[53]**	**EDBGRQ**	b1	A request to theARM Cortex-R4F processor to enter the debug state.
[25:22]	**CONFIG[57:54]**	**CFGATCMSZ**	b0111	Soft ATCM size config (maximum ATCM size is 64K). Valid values are: 0000 = 0KB 0011 = 4KB 0100 = 8KB 0101 = 16KB 0110 = 32KB 0111 = 64KB

Table 4-12 CT_R4F_TC_CFG1 configuration signals (continued)

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[29:26]	**CONFIG[61:58]**	**CFGBTCMSZ**	b0111	Soft BTCM size config (maximum BTCM size is 64K). Valid values are: 0000 = 0KB 0011 = 4KB 0100 = 8KB 0101 = 16KB 0110 = 32KB 0111 = 64KB
[30]	**CONFIG[62]**	**DBGEN**	b1	Non-Invasive Debug Enable (CPU)
[31]	**CONFIG[63]**	**CTINIDEN**	b1	Non-Invasive Debug Enable (CTI)

The CT_R4F_TC_CFG2 register configuration signals that are sent serially during reset are listed in Table 4-13. The CT_R4F_TC_CFG2 register address is: `0x0008`.

Table 4-13 CT_R4F_TC_CFG2 configuration signals

Config word bit	Config signal name	ARM Cortex-R4F test chip signal name	Default value	Description
[0]	**CONFIG[64]**	**CTIDBGEN**	b1	Invasive Debug Enable (CTI)
[1]	**CONFIG[65]**	Reserved	b0	–
[26:2]	**CONFIG[90:66]**	Reserved	`0x000000`	–
[27]	**CONFIG[91]**	**CPU_nFIQEXT_SEL**	b0	Source for **CPU_nFIQEXT** for VIC bypass.
[28]	**CONFIG[92]**	**CPU_nIRQEXT_SEL**	b0	Source for **CPU_nIRQEXT** for VIC bypass.
[29]	**CONFIG[93]**	**REMAP_0**	b0	Memory REMAP selection See *Memory maps* on page 4-2 for details.
[30]	**CONFIG[94]**	**REMAP_1**	b0	
[31]	**CONFIG[95]**	**REMAP_2**	b0	

4.2.2 ARM Cortex-R4F test chip configuration after reset

The CT_R4F_TC_CFG0 register can only be accessed at reset using the 4-wire serial interface. Some of the bit fields in the CT_R4F_TC_CFG1 and CT_R4F_TC_CFG2 registers can be accessed after reset using the APB port of the SCC.

The base address of the SCC APB port is set by the value of **REMAP**, which is read from the CT_R4F_TC_CFG2[31:29] register field during reset. See Table 4-13 on page 4-19 for details.

The SCC registers APB port addresses for all REMAP options are listed in Table 4-14.

Table 4-14 SCC APB port addresses

SCC register	REMAP = 000	REMAP = 001	REMAP = 010	REMAP = 100
CT_R4F_TC_CFG1	`0xE0007004`	`0xD0007004`	`0xE0007004`	`0xD0007004`
CT_R4F_TC_CFG2	`0xE0007008`	`0xD0007008`	`0xE0007008`	`0xD0007008`

CT_R4F_TC_CFG1 register

The CT_R4F_TC_CFG1 register is at offset `0x0004`.
Figure 4-5 shows the SCC APB port accessible fields within the register.

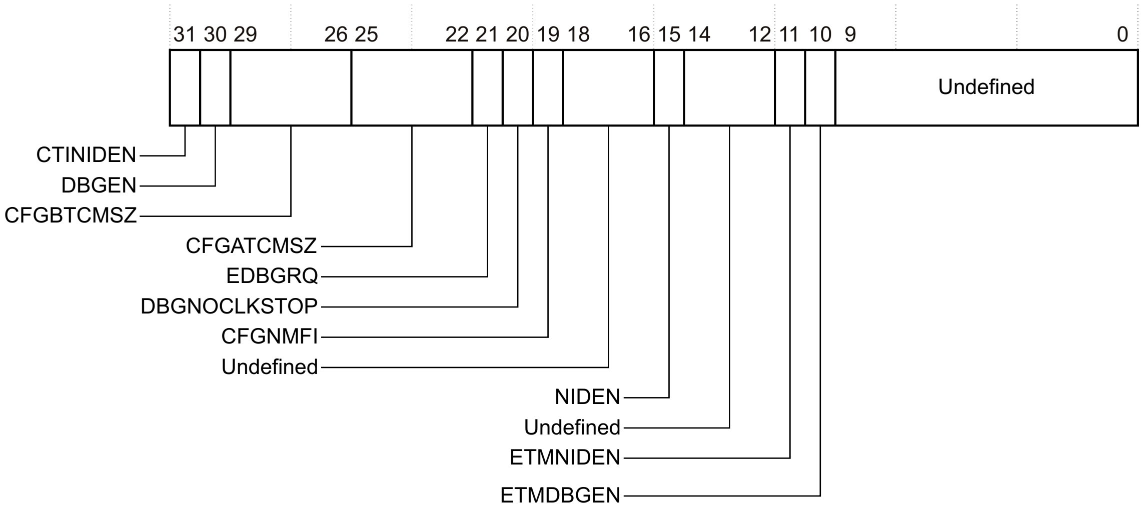


Figure 4-5 CT_R4F_TC_CFG1 register

The function of the register bits are listed in Table 4-15 on page 4-21.

Table 4-15 CT_R4F_TC_CFG1 register bit assignments

Bits	Access	Name	Reset	Description
[31]	Read/Write	**CTINIDEN**	b1	Non-Invasive Debug Enable (CTI)
[30]	Read/Write	**DBGEN**	b1	Non-Invasive Debug Enable (CPU)
[29:26]	Read/Write	**CFGBTCMSZ**	b0111	Soft BTCM size config (maximum BTCM size is 64K). Valid values are: 0000 = 0KB 0011 = 4KB 0100 = 8KB 0101 = 16KB 0110 = 32KB 0111 = 64KB
[25:22]	Read/Write	**CFGATCMSZ**	b0111	Soft ATCM size config (maximum ATCM size is 64K). Valid values are: 0000 = 0KB 0011 = 4KB 0100 = 8KB 0101 = 16KB 0110 = 32KB 0111 = 64KB
[21]	Read/Write	**EDBGRQ**	b0	
[20]	Read/Write	**DBGNOCLKSTOP**	b0	The ARM Cortex-R4F processor does not stop the clocks when entering WFI state
[19]	Read/Write	**CFGNMFI**	b0	
[18:16]	Write ignored, read undefined	–	–	Undefined
[15]	Read/Write	**NIDEN**	b1	Non-Invasive Debug Enable (CPU)
[14:12]	Write ignored, read undefined	–	–	Undefined
[11]	Read/Write	**ETMNIDEN**	b1	Non-Invasive Debug Enable (ETM)
[10]	Read/Write	**ETMDBGEN**	b1	Invasive Debug Enable (ETM)
[9:0]	Write ignored, read undefined	–	–	Undefined

CT_R4F_TC_CFG2 register

The CT_R4F_TC_CFG2 register is at offset `0x0008`.
Figure 4-6 shows the SCC APB port accessible fields within the register.

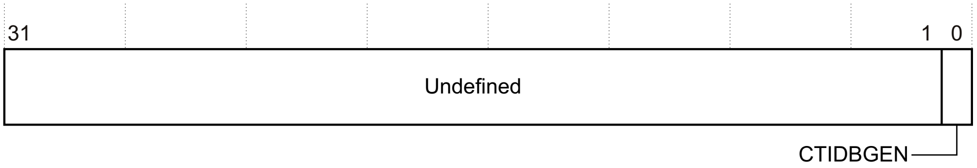


Figure 4-6 CT_R4F_TC_CFG2 register

The function of the register bits are listed in Table 4-16.

Table 4-16 CT_R4F_TC_CFG1 register bit assignments

Bits	Access	Name	Reset	Description
[31:4]	Write ignored, Read as zero	–	`0x0000000`	Undefined
[3:1]	Write ignored, Read as zero	–	b000	Undefined
[0]	Read/Write	**CTIDBGEN**	b1	Invasive Debug Enable (CTI)

4.2.3 CT-R4F Oscillator configuration

There are three ICS307 programmable oscillators on the CT-R4F and one fixed 24MHz oscillator. The 24MHz oscillator is the reference clock, **TIMCLK** for the timer in the ARM Cortex-R4F test chip. See *ARM Dual-Timer Module (SP804) Technical Reference Manual* (ARM DDI 0271) for details.

Note

The default clock frequencies are determined at reset by the EB sending programming information to the CT-R4F. Suitable default values are not specified by the CT-R4F itself. If a custom baseboard is used, it is a requirement that the custom baseboard provides suitable default values. See *ARM Cortex-R4F test chip configuration at reset* on page 4-16 for programming details.

The programmable oscillators are:

OSC0 The reference clock (**REFCLK**) for the test chip PLL (PLL0) that generates **CPU_CLK**.

For default frequency **REFCLK** = 50MHz and
default **CLOCK_MODE** = b01:

- **CPU_CLK** = 250MHz
- **ACLK** = 125MHz
- **PCLK** = 15.625MHz

OSC1 The reference clock (**FBCLK_IN**) for the test chip PLL (PLL1) that generates **MCLK**.

For default frequency 67MHz:
FBCLK_IN = **SDRAM_CLK** = **MCLK** = 134MHz (OSC1 x2).

OSC2 The reference clock for the test chip CLCD controller (PL111).

For default frequency = 25MHz:
CLCDCLK_IO = 25MHz.

See *Clocks* on page 3-4 for details of the clock routing on the CT-R4F and within the ARM Cortex-R4F test chip.

Setting the programmable oscillator frequency

The output frequencies of the ICS307 programmable oscillators are controlled by divider values loaded into the serial data input pins on the oscillators. The divider values required are transferred during reset from the CT_R4F_OSCx registers in the CT-R4F PLD. The CT_R4F_OSCx registers are accessed via the CT-R4F 4-wire serial interface.

See *Overview of Core Tile configuration* on page 3-24 for details of the 4-wire serial interface address and data stream format, and *CT-R4F PLD registers* on page 4-15 for the location of all the CT_R4F_OSCx registers.

You can calculate the oscillator output frequency from the formula:

$$\text{OSCCLKx} = \frac{48 \text{ x } (\text{VDW}+8)}{(\text{RDW}+2) \text{ x OD}} \text{ MHz}$$

where:

VDW Is the VCO divider word (4 – 511) read from CT_R4F_OSCx[8:0]

RDW Is the reference divider word (1 – 127) read from CT_R4F_OSCx[15:9]

OD Is the output divider select (2 to 10) read from CT_R4F_OSCx[18:16]:

- `b000` selects divide by 10
- `b001` selects divide by 2
- `b010` selects divide by 8
- `b011` selects divide by 4
- `b100` selects divide by 5
- `b101` selects divide by 7
- `b110` selects divide by 3
- `b111` selects divide by 6.

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

CT_R4F_OSC registers

Figure 4-7 shows the bit allocations within each register.

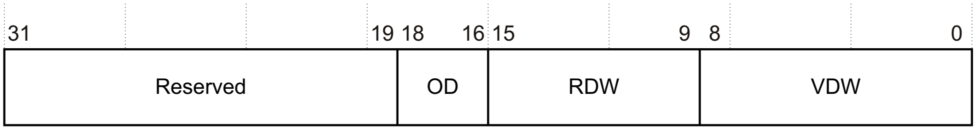


Figure 4-7 CT_R4F_OSC register

The CT_R4F_OSC registers hold the values that set the OSC[0:2] reference clock frequencies. The registers are accessed using the 4-wire serial interface to the CT-R4F PLD. The function of the register bits are listed in Table 4-17.

Table 4-17 CT_R4F_OSC register bit assignments

Bits	Access	Name	Description
[31:20]	Write ignored, Read as zero	–	Reserved
[19]	Write ignored, Read as zero	–	Reserved
[18:16]	Write	**OD**	Output Divider value
[15:9]	Write	**RDW**	Reference Divider Word value
[8:0]	Write	**VDW**	VCO Divider word value

The serial stream address, the default divider values sent by the EB, and the resulting frequency of each reference oscillator are listed in Table 4-18.

Table 4-18 CT_R4F_OSC register default values

CT_R4F_OSC register	Serial stream address	OD divider	RDW divider	VDW divider	Frequency
CT_R4F_OSC0	`0x1004`	b011	b0010110	b001011100	50MHz
CT_R4F_OSC1	`0x1005`	b011	b0010110	b001111110	67MHz
CT_R4F_OSC2	`0x1006`	b000	b0010110	b001110101	25MHz

4.2.4 DVI and CLCD configuration

To maximize the performance of the CLCD interface, the CT-R4F includes a local triple video DAC, DVI transmitter, and DVI interface connector to provide direct support for both digital and analog displays. The CLCD panel side signals can also be routed to a baseboard CLCD interface through the HDRZ header. A simplified diagram of the CLCD signal routing is shown in Figure 3-7 on page 3-20. The routing of the CLCD signals is controlled by the baseboard using the 4-wire interface to the CT_R4F_DVI register in the CT-R4F PLD.

CT_RF_DVI register

The CT_R4F_DVI register is at serial stream address `0x1008`.
Figure 4-8 shows the bit allocations within the register.

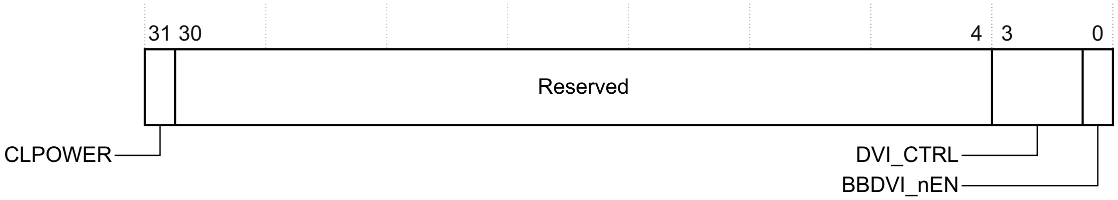


Figure 4-8 CT_R4F_DVI register

The function of the register bits are listed in Table 4-19.

Table 4-19 CT_R4F_DVI register bit assignments

Bits	Access	Name	Reset	Description
[31]	Read only	**CLPOWER**	b0	CLCD power enable
[30:28]	–	–	b000	Reserved
[27:4]	–	–	`0x000000`	Reserved
[3:1]	Read/Write	**DVI_CTRL**	b000	Set to b000 for normal operation
[0]	Read/Write	**BBDVI_nEN**	b0	Enables the CT-R4F CLCD controller panel side outputs to the HDRZ header bus ZL[0:31]. See *DVI and CLCD routing* on page 3-20 and *HDRZ signals* on page 5-18 for further details.

4.3 Board level control and status

The CT-R4F PLD enables control of a number of board level functions and the monitoring of board status. The routing of resets to the ARM Cortex-R4F test chip is controlled by the baseboard using the 4-wire interface to the CT_R4F_CTRL register in the CT-R4F PLD.

4.3.1 CT_R4F_CTRL register

The CT_R4F_CTRL register is at serial stream address `0x100A`.
Figure 4-9 shows the bit allocations within the register.

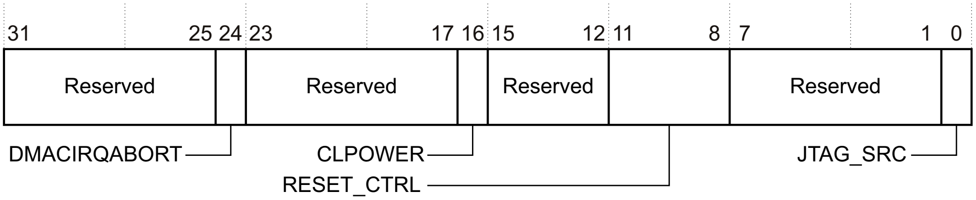


Figure 4-9 CT_RF_CTRL register

The function of the register bits are listed in Table 4-20.

Table 4-20 CT_R4F_CTRL register bit assignments

Bits	Access	Name	Reset	Description
[31:25]	Read only	–	b0000000	Reserved
[24]	Read only	**DMACIRQABORT**	bX	Set HIGH by the ARM Cortex-R4F test chip DMAC when an abort occurs.
[23:17]	Read only	–	b0000000	Reserved
[16]	Read only	**CLPOWER**	bx	Status of CLCD Power Enable.
[15:12]	Read only	–	b0000	Reserved
[11]	Read/Write	**RESET_CTRL[3]**	b0	Enables **nWARMRST** from the baseboard to control **nSOC_RESET_IO** to the ARM Cortex-R4F test chip.
[10]	Read/Write	**RESET_CTRL[2]**	b0	Enables **nWARMRST** from the baseboard to control **nDBG_RESET_IO** to the ARM Cortex-R4F test chip.
[9]	Read/Write	**RESET_CTRL[1]**	b0	Set to b0 for normal operation.

Table 4-20 CT_R4F_CTRL register bit assignments (continued)

Bits	Access	Name	Reset	Description
[8]	Read/Write	**RESET_CTRL[0]**	b0	Enables **nWARMRST** from the baseboard to control **nRESET_IO** to the ARM Cortex-R4F test chip.
[7:1]	Read only	–	b0000000	Reserved
[0]	Read/Write	**JTAG_SRC**	b0	When **nCFGEN** is LOW, selects the JTAG source as the HDRZ header or the Trace A connector: b0 = JTAG source is at the HDRZ header. b1 = JTAG source is at the TRACE A connector.

4.4 Power supply monitoring

There is an *Analog to Digital Converter* (ADC) and eight CT_R4F_ADC registers in the CT-R4F PLD that enable specific ARM Cortex-R4F test chip currents and voltages to be monitored. Each channel of the ADC is allocated a register to provides a digital value proportional to the current or the voltage supplied. The contents of these registers is transferred continually to the baseboard using the 4-wire serial interface.

4.4.1 CT_R4F_ADC registers

Figure 4-10 shows the bit allocations within the CT_R4_ADC registers.

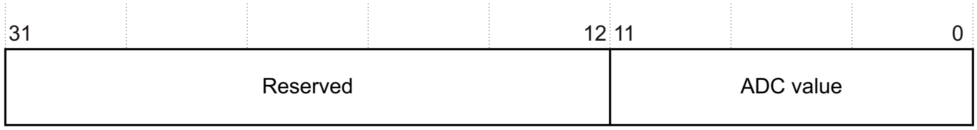


Figure 4-10 CT_R4F_ADC register

The CT_R4F_ADC register serial stream addresses and the currents and voltages that are monitored are listed in Table 4-21 together with the measurement scaling factor applied.

Table 4-21 CT_R4F_ADC bit assignments

Address	Access	Name	Description
`0x1010`	Read only	CT_R4F_ADC0	Monitors ARM Cortex-R4F processor current @ 0.5mA per bit
`0x1011`	Read only	CT_R4F_ADC1	Reserved
`0x1012`	Read only	CT_R4F_ADC2	Monitors the test chip sub-system current @ 0.2mA per bit
`0x1013`	Read only	CT_R4F_ADC3	Monitors the ARM Cortex-R4F processor and CT-R4F test chip sub-system voltage @ 0.5mV per bit
`0x1014`	Read only	CT_R4F_ADC4	Monitors the CT-R4F test chip PLL voltage @ 1mV per bit
`0x1015`	Read only	CT_R4F_ADC5	Monitors the CT-R4F test chip I/O voltage @1mV per bit
`0x1016`	Read only	CT_R4F_ADC6	Reserved
`0x1017`	Read only	CT_R4F_ADC7	Reserved

4.5 Board identification

The CT-R4F PLD includes an ID register. The contents of the register is transferred from the CT-R4F PLD to the baseboard using the 4-wire serial interface during CT-R4F initialization.

4.5.1 CT-R4F ID register

The CT_R4F_ID register provides information that identifies the revision status of the CT-R4F Core Tile.

Note

This information will be useful if you need to contact ARM Support for assistance.

The CT_R4F_ID register is at serial stream address `0x1000`.
Figure 4-11 shows the bit allocations within the CT_R4F_ID register.

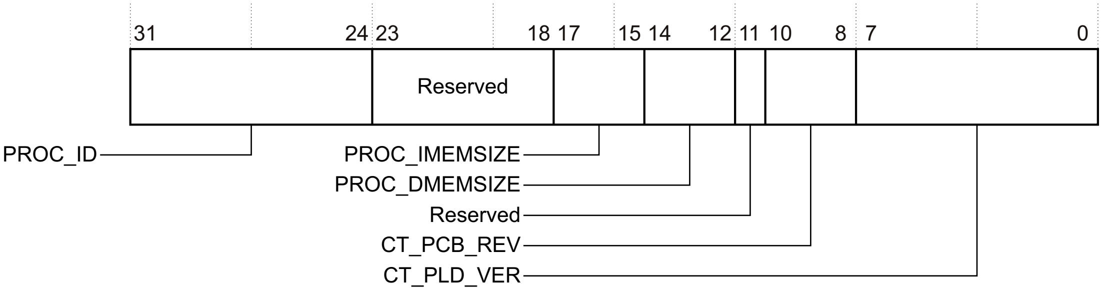


Figure 4-11 CT_R4F_ID register

Table 4-22 shows the bit assignments within the register.

Table 4-22 CT_R4F_ID register bit assignments

Bits	Access	Name	Value	Description
[31:24]	Read only	PROC_ID	`0x11`	Processor ID (ARM Cortex-R4F)
[23:18]	Read only	–	–	Reserved
[17:15]	Read only	PROC_IMEMSIZE	b100	Processor Instruction Cache size (16KB)

Table 4-22 CT_R4F_ID register bit assignments (continued)

Bits	Access	Name	Value	Description
[14:12]	Read only	PROC_DMEMSIZE	b100	Processor Data Cache size (16KB)
[11]	Read only	–	–	Reserved
[10:8]	Read only	CT_PCB_REV	b010	CT-R4F board revision
[7:0]	Read only	CT_PLD_VER	0x01	CT-R4F PLD version

Chapter 5
Signal Descriptions

This chapter provides a summary of signals present on the CT-R4F connectors, the links that can be modified to change signal routing, test points, and the LED indicators.
It contains the following sections:

5.1 Header connectors

Figure 5-1 on page 5-3 shows the pin numbers and power-blade usage of the HDRX, HDRY, and HDRZ headers on the bottom of the board.

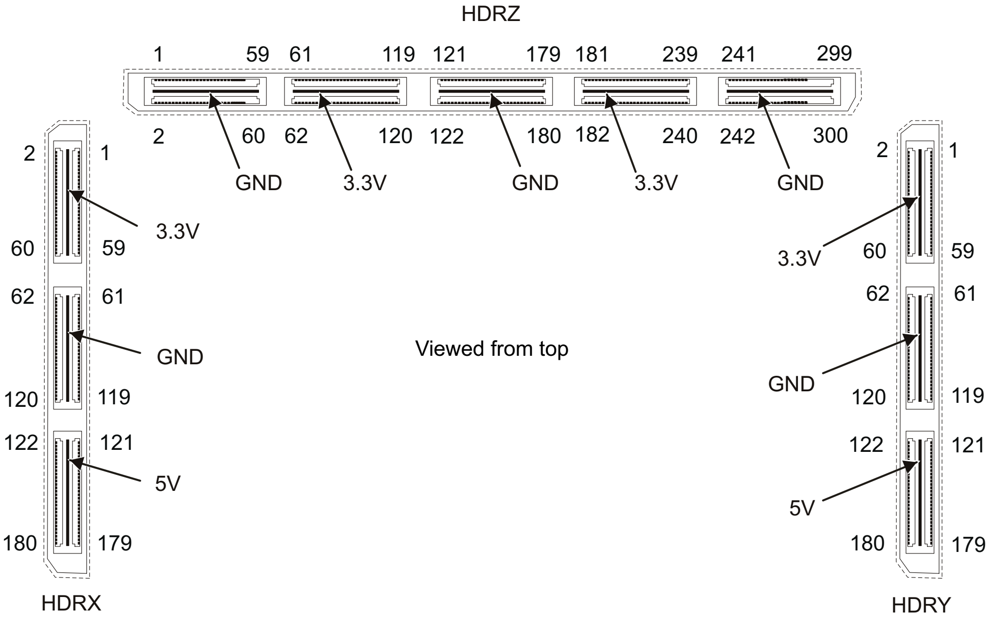


Figure 5-1 HDRX, HDRY, and HDRZ (lower) pin numbering

Table 5-1 lists the Samtec part numbers.

The CT-R4F uses Samtec connectors on the bottom of the board only. The total board separation is 8mm when mated. As the Core Tiles have a maximum component height of 2.5mm on the bottom of the board this ensures that there are no component interference problems with mated boards.

Note

Samtec describes the QSH connector used on the bottom of the board as a socket.

Table 5-1 Samtec part numbers

Header	Part number	Mating height
HDRX	QSH-090-01-F-D-A-K	5mm
HDRY	QSH-090-01-F-D-A-K	5mm
HDRZ	QSH-150-01-F-D-A-K	5mm

5.1.1 HDRX signals

Table 5-2 on page 5-5 lists the signals on the HDRX pins for the CT-R4F and EB baseboard. The CT-R4F AXI master port signals are multiplexed onto the pins on this header.

For a description of the AXI bus signals see *AMBA 3 AXI Protocol Specification* (ARM IHI 0022).

Note

The signal on the lower header of a Core Tile is named **XLn**. For example, the pin 1 signal for the lower header is **XL90** and the corresponding CT-R4F signal is **ARADDR13 / ARADDR29**. This example also illustrates the designation used for a multiplexed signal, `X / Y` where signal `X` is present when **CLK_OUT_MINUS1** is HIGH and signal `Y` is present when **CLK_OUT_MINUS1** is LOW. See *AXI bus multiplexing* on page 3-22 for further details of the AXI signal multiplexing.

For the EB there are two HDRX headers (tile site 1 and tile site 2). For tile site1 the HDRX pin number prefix is **T1X**.

Table 5-2 HDRX signals

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ARADDR12 /ARADDR28	**T1X89**	2	1	**T1X90**	**ARADDR13 /ARADDR29**
ARADDR11 /ARADDR27	**T1X88**	4	3	**T1X91**	**ARADDR14 /ARADDR30**
ARADDR10 /ARADDR26	**T1X87**	6	5	**T1X92**	**ARADDR15 /ARADDR31**
ARADDR9 /ARADDR25	**T1X86**	8	7	**T1X93**	**ARID0 /ARID2**
ARADDR8 /ARADDR24	**T1X85**	10	9	**T1X94**	**ARID1 /ARID3**
ARADDR7 /ARADDR23	**T1X84**	12	11	**T1X95**	**ARLEN0 /ARLEN2**
ARADDR6 /ARADDR22	**T1X83**	14	13	**T1X96**	**ARLEN1 /ARLEN3**
ARADDR5 /ARADDR21	**T1X82**	16	15	**T1X97**	**ARSIZE0 /ARSIZE1**
ARADDR4 /ARADDR20	**T1X81**	18	17	**T1X98**	**ARID4 /ARPROT2**
ARADDR3 /ARADDR19	**T1X80**	20	19	**T1X99**	**ARPROT0 /ARPROT1**
ARADDR2 /ARADDR18	**T1X79**	22	21	**T1X100**	**ARBURST0 /ARBURST1**
ARADDR1 /ARADDR17	**T1X78**	24	23	**T1X101**	**ARLOCK0 /ARLOCK1**
ARADDR0 /ARADDR16	**T1X77**	26	25	**T1X102**	**ARCACHE0 /ARCACHE2**
BREADY	**T1X76**	28	27	**T1X103**	**ARCACHE1 /ARCACHE3**
BVALID	**T1X75**	30	29	**T1X104**	**ARVALID /ARID5**

Table 5-2 HDRX signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
BRESP0 /BRESP1	**T1X74**	32	31	**T1X105**	**ARREADY**
BID4 /BID5	**T1X73**	34	33	**T1X106**	**RDATA0 /RDATA32**
BID1 /BID3	**T1X72**	36	35	**T1X107**	**RDATA1 /RDATA33**
BID0 /BID2	**T1X71**	38	37	**T1X108**	**RDATA2 /RDATA34**
AWREADY	**T1X70**	40	39	**T1X109**	**RDATA3 /RDATA35**
AWVALID /AWID5	**T1X69**	42	41	**T1X110**	**RDATA4 /RDATA36**
AWCACHE1 /AWCACHE3	**T1X68**	44	43	**T1X111**	**RDATA5 /RDATA37**
AWCACHE0 /AWCACHE2	**T1X67**	46	45	**T1X112**	**RDATA6 /RDATA38**
AWLOCK0 /AWLOCK1	**T1X66**	48	47	**T1X113**	**RDATA7 /RDATA39**
AWBURST0 /AWBURST1	**T1X65**	50	49	**T1X114**	**RDATA8 /RDATA40**
AWPROT0 /AWPROT1	**T1X64**	52	51	**T1X115**	**RDATA9 /RDATA41**
ARM_nRESET	**T1X63**	54	53	**T1X116**	**RDATA10 /RDATA42**
AWID4 /AWPROT2	**T1X62**	56	55	**T1X117**	**RDATA11 /RDATA43**
AWSIZE0 /AWSIZE1	**T1X61**	58	57	**T1X118**	**RDATA12 /RDATA44**
AWLEN1 /AWLEN3	**T1X60**	60	59	**T1X119**	**RDATA13 /RDATA45**

Table 5-2 HDRX signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
AWLEN0 /AWLEN2	**T1X59**	62	61	**T1X120**	**RDATA14 /RDATA46**
AWID1 /AWID3	**T1X58**	64	63	**T1X121**	**RDATA15 /RDATA47**
AWID0 /AWID2	**T1X57**	66	65	**T1X122**	**RDATA16 /RDATA48**
AWADDR15 /AWADDR31	**T1X56**	68	67	**T1X123**	**RDATA17 /RDATA49**
AWADDR14 /AWADDR30	**T1X55**	70	69	**T1X124**	**RDATA18 /RDATA50**
AWADDR13 /AWADDR29	**T1X54**	72	71	**T1X125**	**RDATA19 /RDATA51**
AWADDR12 /AWADDR28	**T1X53**	74	73	**T1X126**	**RDATA20 /RDATA52**
AWADDR11 /AWADDR27	**T1X52**	76	75	**T1X127**	**RDATA21 /RDATA53**
AWADDR10 /AWADDR26	**T1X51**	78	77	**T1X128**	**RDATA22 /RDATA54**
AWADDR9 /AWADDR25	**T1X50**	80	79	**T1X129**	**RDATA23 /RDATA55**
AWADDR8 /AWADDR24	**T1X49**	82	81	**T1X130**	**RDATA24 /RDATA56**
AWADDR7 /AWADDR23	**T1X48**	84	83	**T1X131**	**RDATA25 /RDATA57**
AWADDR6 /AWADDR22	**T1X47**	86	85	**T1X132**	**RDATA26 /RDATA58**
AWADDR5 /AWADDR21	**T1X46**	88	87	**T1X133**	**RDATA27 /RDATA59**
AWADDR4 /AWADDR20	**T1X45**	90	89	**T1X134**	**RDATA28 /RDATA60**

Table 5-2 HDRX signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
AWADDR3 /AWADDR19	**T1X44**	92	91	**T1X135**	**RDATA29 /RDATA61**
AWADDR2 /AWADDR18	**T1X43**	94	93	**T1X136**	**RDATA30 /RDATA62**
AWADDR1 /AWADDR17	**T1X42**	96	95	**T1X137**	**RDATA31 /RDATA63**
AWADDR0 /AWADDR16	**T1X41**	98	97	**T1X138**	**RID0 /RID2**
WREADY	**T1X40**	100	99	**T1X139**	**RID1 /RID3**
WVALID /WID5	**T1X39**	102	101	**T1X140**	**RRESP0 /RRESP1**
WLAST /WID4	**T1X38**	104	103	**T1X141**	**RLAST /RID4**
WSTRB3 /WSTRB7	**T1X37**	106	105	**T1X142**	**RVALID /RID5**
WSTRB2 /WSTRB6	**T1X36**	108	107	**T1X143**	**RREADY**
WSTRB1 /WSTRB5	**T1X35**	110	109	**T1X144**	**XL144**
WSTRB0 /WSTRB4	**T1X34**	112	111	**T1X145**	**XL145**
WID1 /WID3	**T1X33**	114	113	**T1X146**	**XL146**
WID0 /WID2	**T1X32**	116	115	**T1X147**	**XL147**
WDATA31 /WDATA63	**T1X31**	118	117	**T1X148**	**XL148**
WDATA30 /WDATA62	**T1X30**	120	119	**T1X149**	**XL149**

Table 5-2 HDRX signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
WDATA29 /WDATA61	**T1X29**	122	121	**T1X150**	**XL150**
WDATA28 /WDATA60	**T1X28**	124	123	**T1X151**	**XL151**
WDATA27 /WDATA59	**T1X27**	126	125	**T1X152**	**XL152**
WDATA26 /WDATA58	**T1X26**	128	127	**T1X153**	**XL153**
WDATA25 /WDATA57	**T1X25**	130	129	**T1X154**	**XL154**
WDATA24 /WDATA56	**T1X24**	132	131	**T1X155**	**XL155**
WDATA23 /WDATA55	**T1X23**	134	133	**T1X156**	**XL156**
WDATA22 /WDATA54	**T1X22**	136	135	**T1X157**	**XL157**
WDATA21 /WDATA53	**T1X21**	138	137	**T1X158**	**XL158**
WDATA20 /WDATA52	**T1X20**	140	139	**T1X159**	**XL159**
WDATA19 /WDATA51	**T1X19**	142	141	**T1X160**	**XL160**
WDATA18 /WDATA50	**T1X18**	144	143	**T1X161**	**XL161**
WDATA17 /WDATA49	**T1X17**	146	145	**T1X162**	**XL162**
WDATA16 /WDATA48	**T1X16**	148	147	**T1X163**	**XL163**
WDATA15 /WDATA47	**T1X15**	150	149	**T1X164**	**XL164**

Table 5-2 HDRX signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
WDATA14 /WDATA46	**T1X14**	152	151	**T1X165**	**XL165**
WDATA13 /WDATA45	**T1X13**	154	153	**T1X166**	**XL166**
WDATA12 /WDATA44	**T1X12**	156	155	**T1X167**	**XL167**
WDATA11 /WDATA43	**T1X11**	158	157	**T1X168**	**XL168**
WDATA10 /WDATA42	**T1X10**	160	159	**T1X169**	**XL169**
WDATA9 /WDATA41	**T1X9**	162	161	**T1X170**	**XL170**
WDATA8 /WDATA40	**T1X8**	164	163	**T1X171**	**XL171**
WDATA7 /WDATA39	**T1X7**	166	165	**T1X172**	**XL172**
WDATA6 /WDATA38	**T1X6**	168	167	**T1X173**	**XL173**
WDATA5 /WDATA37	**T1X5**	170	169	**T1X174**	**XL174**
WDATA4 /WDATA36	**T1X4**	172	171	**T1X175**	**XL175**
WDATA3 /WDATA35	**T1X3**	174	173	**T1X176**	**XL176**
WDATA2 /WDATA34	**T1X2**	176	175	**T1X177**	**XL177**
WDATA1 /WDATA33	**T1X1**	178	177	**T1X178**	**XL178**
WDATA0 /WDATA32	**T1X0**	180	179	**T1X179**	**XL179**

5.1.2 HDRY signals

Table 5-3 lists the signals on the HDRY pins for the CT-R4F and EB baseboard. The CT-R4F AXI slave port signals are multiplexed onto the pins on this header.

For a description of the AXI bus signals see *AMBA 3 AXI Protocol Specification* (ARM IHI 0022).

——— **Note** ———

The signal on the lower header of a Core Tile is named **YLn**. For example, the pin 1 signal for the lower header is **YL89** and the corresponding CT-R4F signal is **ARADDR12 / ARADDR28**. This example also illustrates the designation used for a multiplexed signal, `X` / `Y` where signal `X` is present when **CLK_OUT_MINUS1** is HIGH and signal `Y` is present when **CLK_OUT_MINUS1** is LOW. See *AXI bus multiplexing* on page 3-22 for further details of the AXI signal multiplexing.

For the EB there are two HDRY headers (tile site 1 and tile site 2). For tile site1 the HDRY pin number prefix is **T1Y**.

Table 5-3 HDRY signals

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ARADDR13 /ARADDR29	**T1Y90**	2	1	**T1Y89**	**ARADDR12 /ARADDR28**
ARADDR14 /ARADDR30	**T1Y91**	4	3	**T1Y88**	**ARADDR11 /ARADDR27**
ARADDR15 /ARADDR31	**T1Y92**	6	5	**T1Y87**	**ARADDR10 /ARADDR26**
ARID0 /ARID2	**T1Y93**	8	7	**T1Y86**	**ARADDR9 /ARADDR25**
ARID1 /ARID3	**T1Y94**	10	9	**T1Y85**	**ARADDR8 /ARADDR24**
ARLEN0 /ARLEN2	**T1Y95**	12	11	**T1Y84**	**ARADDR7 /ARADDR23**
ARLEN1 /ARLEN3	**T1Y96**	14	13	**T1Y83**	**ARADDR6 /ARADDR22**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ARSIZE0 /ARSIZE1	**T1Y97**	16	15	**T1Y82**	**ARADDR5 /ARADDR21**
ARID4 /ARPROT2	**T1Y98**	18	17	**T1Y81**	**ARADDR4 /ARADDR20**
ARPROT0 /ARPROT1	**T1Y99**	20	19	**T1Y80**	**ARADDR3 /ARADDR19**
ARBURST0 /ARBURST1	**T1Y100**	22	21	**T1Y79**	**ARADDR2 /ARADDR18**
ARLOCK0 /ARLOCK1	**T1Y101**	24	23	**T1Y78**	**ARADDR1 /ARADDR17**
ARCACHE0 /ARCACHE2	**T1Y102**	26	25	**T1Y77**	**ARADDR0 /ARADDR16**
ARCACHE1 /ARCACHE3	**T1Y103**	28	27	**T1Y76**	**BREADY**
ARVALID /ARID5	**T1Y104**	30	29	**T1Y75**	**BVALID**
ARREADY	**T1Y105**	32	31	**T1Y74**	**BRESP0 /BRESP1**
RDATA0 /RDATA32	**T1Y106**	34	33	**T1Y73**	**BID4 /BID5**
RDATA1 /RDATA33	**T1Y107**	36	35	**T1Y72**	**BID1 /BID3**
RDATA2 /RDATA34	**T1Y108**	38	37	**T1Y71**	**BID0 /BID2**
RDATA3 /RDATA35	**T1Y109**	40	39	**T1Y70**	**AWREADY**
RDATA4 /RDATA36	**T1Y110**	42	41	**T1Y69**	**AWVALID /AWID5**
RDATA5 /RDATA37	**T1Y111**	44	43	**T1Y68**	**AWCACHE1 /AWCACHE3**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
RDATA6 /RDATA38	**T1Y112**	46	45	**T1Y67**	**AWCACHE0 /AWCACHE2**
RDATA7 /RDATA39	**T1Y113**	48	47	**T1Y66**	**AWLOCK0 /AWLOCK1**
RDATA8 /RDATA40	**T1Y114**	50	49	**T1Y65**	**AWBURST0 /AWBURST1**
RDATA9 /RDATA41	**T1Y115**	52	51	**T1Y64**	**AWPROT0 /AWPROT1**
RDATA10 /RDATA42	**T1Y116**	54	53	**T1Y63**	**ARM_nRESET**
RDATA11 /RDATA43	**T1Y117**	56	55	**T1Y62**	**AWID4 /AWPROT2**
RDATA12 /RDATA44	**T1Y118**	58	57	**T1Y61**	**AWSIZE0 /AWSIZE1**
RDATA13 /RDATA45	**T1Y119**	60	59	**T1Y60**	**AWLEN1 /AWLEN3**
RDATA14 /RDATA46	**T1Y120**	62	61	**T1Y59**	**AWLEN0 /AWLEN2**
RDATA15 /RDATA47	**T1Y121**	64	63	**T1Y58**	**AWID1 /AWID3**
RDATA16 /RDATA48	**T1Y122**	66	65	**T1Y57**	**AWID0 /AWID2**
RDATA17 /RDATA49	**T1Y123**	68	67	**T1Y56**	**AWADDR15 /AWADDR31**
RDATA18 /RDATA50	**T1Y124**	70	69	**T1Y55**	**AWADDR14 /AWADDR30**
RDATA19 /RDATA51	**T1Y125**	72	71	**T1Y54**	**AWADDR13 /AWADDR29**
RDATA20 /RDATA52	**T1Y126**	74	73	**T1Y53**	**AWADDR12 /AWADDR28**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
RDATA21 /RDATA53	**T1Y127**	76	75	**T1Y52**	**AWADDR11 /AWADDR27**
RDATA22 /RDATA54	**T1Y128**	78	77	**T1Y51**	**AWADDR10 /AWADDR26**
RDATA23 /RDATA55	**T1Y129**	80	79	**T1Y50**	**AWADDR9 /AWADDR25**
RDATA24 /RDATA56	**T1Y130**	82	81	**T1Y49**	**AWADDR8 /AWADDR24**
RDATA25 /RDATA57	**T1Y131**	84	83	**T1Y48**	**AWADDR7 /AWADDR23**
RDATA26 /RDATA58	**T1Y132**	86	85	**T1Y47**	**AWADDR6 /AWADDR22**
RDATA27 /RDATA59	**T1Y133**	88	87	**T1Y46**	**AWADDR5 /AWADDR21**
RDATA28 /RDATA60	**T1Y134**	90	89	**T1Y45**	**AWADDR4 /AWADDR20**
RDATA29 /RDATA61	**T1Y135**	92	91	**T1Y44**	**AWADDR3 /AWADDR19**
RDATA30 /RDATA62	**T1Y136**	94	93	**T1Y43**	**AWADDR2 /AWADDR18**
RDATA31 /RDATA63	**T1Y137**	96	95	**T1Y42**	**AWADDR1 /AWADDR17**
RID0 /RID2	**T1Y138**	98	97	**T1Y41**	**AWADDR0 /AWADDR16**
RID1 /RID3	**T1Y139**	100	99	**T1Y40**	**WREADY**
RRESP0 /RRESP1	**T1Y140**	102	101	**T1Y39**	**WVALID /WID5**
RLAST /RID4	**T1Y141**	104	103	**T1Y38**	**WLAST /WID4**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
RVALID /RID5	**T1Y142**	106	105	**T1Y37**	**WSTRB3 /WSTRB7**
RREADY	**T1Y143**	108	107	**T1Y36**	**WSTRB2 /WSTRB6**
YL144	**T1Y144**	110	109	**T1Y35**	**WSTRB1 /WSTRB5**
YL145	**T1Y145**	112	111	**T1Y34**	**WSTRB0 /WSTRB4**
YL146	**T1Y146**	114	113	**T1Y33**	**WID1 /WID3**
YL147	**T1Y147**	116	115	**T1Y32**	**WID0 /WID2**
YL148	**T1Y148**	118	117	**T1Y31**	**WDATA31 /WDATA63**
YL149	**T1Y149**	120	119	**T1Y30**	**WDATA30 /WDATA62**
YL150	**T1Y150**	122	121	**T1Y29**	**WDATA29 /WDATA61**
YL151	**T1Y151**	124	123	**T1Y28**	**WDATA28 /WDATA60**
YL152	**T1Y152**	126	125	**T1Y27**	**WDATA27 /WDATA59**
YL153	**T1Y153**	128	127	**T1Y26**	**WDATA26 /WDATA58**
YL154	**T1Y154**	130	129	**T1Y25**	**WDATA25 /WDATA57**
YL155	**T1Y155**	132	131	**T1Y24**	**WDATA24 /WDATA56**
YL156	**T1Y156**	134	133	**T1Y23**	**WDATA23 /WDATA55**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
YL157	**T1Y157**	136	135	**T1Y22**	**WDATA22 /WDATA54**
YL158	**T1Y158**	138	137	**T1Y21**	**WDATA21 /WDATA53**
YL159	**T1Y159**	140	139	**T1Y20**	**WDATA20 /WDATA52**
YL160	**T1Y160**	142	141	**T1Y19**	**WDATA19 /WDATA51**
YL161	**T1Y161**	144	143	**T1Y18**	**WDATA18 /WDATA50**
YL162	**T1Y162**	146	145	**T1Y17**	**WDATA17 /WDATA49**
YL163	**T1Y163**	148	147	**T1Y16**	**WDATA16 /WDATA48**
YL164	**T1Y164**	150	149	**T1Y15**	**WDATA15 /WDATA47**
YL165	**T1Y165**	152	151	**T1Y14**	**WDATA14 /WDATA46**
YL166	**T1Y166**	154	153	**T1Y13**	**WDATA13 /WDATA45**
YL167	**T1Y167**	156	155	**T1Y12**	**WDATA12 /WDATA44**
YL168	**T1Y168**	158	157	**T1Y11**	**WDATA11 /WDATA43**
YL169	**T1Y169**	160	159	**T1Y10**	**WDATA10 /WDATA42**
YL160	**T1Y170**	162	161	**T1Y9**	**WDATA9 /WDATA41**
YL171	**T1Y171**	164	163	**T1Y8**	**WDATA8 /WDATA40**

Table 5-3 HDRY signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
YL172	**T1Y172**	166	165	**T1Y7**	**WDATA7 /WDATA39**
YL173	**T1Y173**	168	167	**T1Y6**	**WDATA6 /WDATA38**
YL174	**T1Y174**	170	169	**T1Y5**	**WDATA5 /WDATA37**
YL175	**T1Y175**	172	171	**T1Y4**	**WDATA4 /WDATA36**
YL176	**T1Y176**	174	173	**T1Y3**	**WDATA3 /WDATA35**
YL177	**T1Y177**	176	175	**T1Y2**	**WDATA2 /WDATA34**
YL178	**T1Y178**	178	177	**T1Y1**	**WDATA1 /WDATA33**
YL179	**T1Y179**	180	179	**T1Y0**	**WDATA0 /WDATA32**

5.1.3 HDRZ signals

Table 5-4 lists the signals on the HDRZ pins for the CT-R4F and EB baseboard. The external clocks, resets, interrupts, CLCD panel signals, the 4-wire serial interface for Core Tile configuration, and the JTAG debug and configuration signals are all routed to the CT-R4F using the HDRZ header.

For a description of these miscellaneous signals, see the relevant sections of the user guide.

Note

The signal on the lower header of a Core Tile is named **ZLn**. For example, the pin 1 signal for the lower header is **ZL128**.

For the EB there are two HDRZ headers (tile site 1 and tile site 2). For tile site 1 the HDRZ pin number prefix is **T1Z**.

Table 5-4 HDRZ signals

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ZL255	**T1Z255**	2	1	**T1Z128**	**ZL128**
ZL254	**T1Z254**	4	3	**T1Z129**	**ZL129**
ZL253	**T1Z253**	6	5	**T1Z130**	**ZL130**
ZL252	**T1Z252**	8	7	**T1Z131**	**ZL131**
ZL251	**T1Z251**	10	9	**T1Z132**	**ZL132**
ZL250	**T1Z250**	12	11	**T1Z133**	**ZL133**
ZL249	**T1Z249**	14	13	**T1Z134**	**ZL134**
ZL248	**T1Z248**	16	15	**T1Z135**	**ZL135**
ZL247	**T1Z247**	18	17	**T1Z136**	**ZL136**
ZL246	**T1Z246**	20	19	**T1Z137**	**ZL137**
ZL245	**T1Z245**	22	21	**T1Z138**	**ZL138**
ZL244	**T1Z244**	24	23	**T1Z139**	**ZL139**
ZL243	**T1Z243**	26	25	**T1Z140**	**ZL140**

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ZL242	**T1Z242**	28	27	**T1Z141**	**ZL141**
ZL241	**T1Z241**	30	29	**T1Z142**	**ZL142**
ZL240	**T1Z240**	32	31	**T1Z143**	**ZL143**
ZL239	**T1Z249**	34	33	**T1Z144**	**ZL144**
ZL238	**T1Z248**	36	35	**T1Z145**	**ZL145**
ZL237	**T1Z237**	38	37	**T1Z146**	**ZL146**
ZL236	**T1Z236**	40	39	**T1Z147**	**ZL147**
ZL235	**T1Z235**	42	41	**T1Z148**	**ZL148**
ZL234	**T1Z234**	44	43	**T1Z149**	**ZL149**
ZL233	**T1Z233**	46	45	**T1Z150**	**ZL150**
ZL232	**T1Z232**	48	47	**T1Z151**	**ZL151**
nWARMRST	**T1Z231**	50	49	**T1Z152**	**ZL152**
PLDCLK	**T1Z230**	52	51	**T1Z153**	**ZL153**
PLDnRST	**T1Z229**	54	53	**T1Z154**	**ZL154**
PLDDOUT	**T1Z228**	56	55	**T1Z155**	**ZL155**
PLDDIN	**T1Z227**	58	57	**T1Z156**	**ZL156**
CTR4F_SPARE[3]	**T1Z226**	60	59	**T1Z157**	**ZL157**
CTR4F_SPARE[2]	**T1Z225**	62	61	**T1Z158**	**ZL158**
CTR4F_SPARE[1]	**T1Z224**	64	63	**T1Z159**	**ZL159**
CTR4F_SPARE[0]	**T1Z223**	66	65	**T1Z160**	**ZL160**
TC_TIMINTC	**T1Z222**	68	67	**T1Z161**	**ZL161**
TC_TIMINT2	**T1Z221**	70	69	**T1Z162**	**ZL162**
TC_TIMINT1	**T1Z220**	72	71	**T1Z163**	**ZL163**
TC_DMAC_IRQ[3]	**T1Z219**	74	73	**T1Z164**	**ZL164**

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
CT_COMMTX	**T1Z218**	76	75	**T1Z165**	**ZL165**
CT_COMMRX	**T1Z217**	78	77	**T1Z166**	**ZL166**
nCOLDRST	**T1Z216**	80	79	**T1Z167**	**ZL167**
TC_DMAC_IRQ[2]	**T1Z215**	82	81	**T1Z168**	**ZL168**
TC_DMAC_IRQ[1]	**T1Z214**	84	83	**T1Z169**	**ZL169**
TC_DMAC_IRQ[0]	**T1Z213**	86	85	**T1Z170**	**ZL170**
TC_CLCDIRQ	**T1Z212**	88	87	**T1Z171**	**ZL171**
(reserved)	**T1Z211**	90	89	**T1Z172**	**ZL172**
(reserved)	**T1Z210**	92	91	**T1Z173**	**ZL173**
CT_nIRQ_EXT	**T1Z209**	94	93	**T1Z174**	**ZL174**
CT_nFIQ_EXT	**T1Z208**	96	95	**T1Z175**	**ZL175**
VICINTSOURCE[7]	**T1Z207**	98	97	**T1Z176**	**ZL176**
VICINTSOURCE[6]	**T1Z206**	100	99	**T1Z177**	**ZL177**
VICINTSOURCE[5]	**T1Z205**	102	101	**T1Z178**	**ZL178**
VICINTSOURCE[4]	**T1Z204**	104	103	**T1Z179**	**ZL179**
VICINTSOURCE[3]	**T1Z203**	106	105	**T1Z180**	**ZL180**
VICINTSOURCE[2]	**T1Z202**	108	107	**T1Z181**	**ZL181**
VICINTSOURCE[1]	**T1Z201**	110	109	**T1Z182**	**ZL182**
VICINTSOURCE[0]	**T1Z200**	112	111	**T1Z183**	**ZL183**
ZL199	**T1Z199**	114	113	**T1Z184**	**ZL84**
ZL198	**T1Z198**	116	115	**T1Z185**	**ZL185**
ZL197	**T1Z197**	118	117	**T1Z186**	**ZL186**
ZL196	**T1Z196**	120	119	**T1Z187**	**ZL187**
ZL195	**T1Z195**	122	121	**T1Z188**	**ZL188**

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ZL194	**T1Z194**	124	123	**T1Z189**	**ZL189**
ZL193	**T1Z193**	126	125	**T1Z190**	**ZL190**
ZL192	**T1Z192**	128	127	**T1Z191**	**ZL191**
CLK_POS_DN_OUT	**CLK_POS_DN_IN**	130	129	**D_nSRST**	**D_nSRST**
CLK_NEG_DN_OUT	**CLK_NEG_DN_IN**	132	131	**D_nTRST**	**D_nTRST**
–	**CLK_POS_UP_OUT**	134	133	**D_TDO_IN**	**D_TDO_OUT**
CLK_NEG_UP_IN	**CLK_NEG_UP_OUT**	136	135	**D_TDI**	**D_TDI**
–	**CLK_UP_THRU**	138	137	**D_TCK_OUT**	**D_TCK_IN**
CLK_IN_MINUS1	**CLK_OUT_PLUS1**	140	139	**D_TMS_OUT**	**D_TMS_IN**
–	**CLK_OUT_PLUS2**	142	141	**D_RTCK**	**D_RTCK**
–	**CLK_IN_PLUS2**	144	143	**C_nSRST**	–
CLK_OUT_MINUS1	**CLK_IN_PLUS1**	146	145	**C_nTRST**	**C_nTRST**
–	**CLK_DN_THRU**	148	147	**C_TDO_IN**	**C_TDO_OUT**
–	**CLK_GLOBAL**	150	149	**C_TDI**	**C_TDI**
–	**FPGA_IMAGE**	152	151	**C_TCK_OUT**	**C_TCK_IN**
nSYSPOR	**nSYSPOR**	154	153	**C_TMS_OUT**	**C_TMS_IN**
nSYSRST	**nSYSRST**	156	155	**nTILE_DET**	**GND**
nRTCKEN	**nRTCKEN**	158	157	**nCFGEN**	**nCFGEN**
–	**SPARE12**	160	159	**GLOBAL_DONE**	**GLOBAL_DONE**
–	**SPARE10**	162	161	**SPARE11**	–
–	**SPARE8**	164	163	**SPARE9**	–
–	**SPARE6**	166	165	**SPARE7**	–
–	**SPARE4**	168	167	**SPARE5**	–
–	**SPARE2**	170	169	**SPARE3**	–

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
–	**SPARE0**	172	171	**SPARE1**	–
ZL64	**T1Z64**	174	173	**T1Z63**	**ZL63**
ZL65	**T1Z65**	176	175	**T1Z62**	**ZL62**
ZL66	**T1Z66**	178	177	**T1Z61**	**ZL61**
ZL67	**T1Z67**	180	179	**T1Z60**	**ZL60**
ZL68	**T1Z68**	182	181	**T1Z59**	**ZL59**
ZL69	**T1Z79**	184	183	**T1Z58**	**ZL58**
ZL70	**T1Z70**	186	185	**T1Z57**	**ZL57**
ZL71	**T1Z71**	188	187	**T1Z56**	**ZL56**
ZL72	**T1Z72**	190	189	**T1Z55**	**ZL55**
ZL73	**T1Z73**	192	191	**T1Z54**	**ZL54**
ZL74	**T1Z74**	194	193	**T1Z53**	**ZL53**
ZL75	**T1Z75**	196	195	**T1Z52**	**ZL52**
ZL76	**T1Z76**	198	197	**T1Z51**	**ZL51**
ZL77	**T1Z77**	200	199	**T1Z50**	**ZL50**
ZL78	**T1Z78**	202	201	**T1Z49**	**ZL49**
ZL79	**T1Z79**	204	203	**T1Z48**	**ZL48**
ZL80	**T1Z80**	206	205	**T1Z47**	**ZL47**
ZL81	**T1Z81**	208	207	**T1Z46**	**ZL46**
ZL82	**T1Z82**	210	209	**T1Z45**	**ZL45**
ZL83	**T1Z83**	212	211	**T1Z44**	**ZL44**
ZL84	**T1Z84**	214	213	**T1Z43**	**ZL43**
ZL85	**T1Z85**	216	215	**T1Z42**	**ZL42**
ZL86	**T1Z86**	218	217	**T1Z41**	**ZL41**

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ZL87	**T1Z87**	220	219	**T1Z40**	**ZL40**
ZL88	**T1Z88**	222	221	**T1Z39**	**ZL39**
ZL89	**T1Z89**	224	223	**T1Z38**	**ZL38**
ZL90	**T1Z90**	226	225	**T1Z37**	**ZL37**
ZL91	**T1Z91**	228	227	**T1Z36**	**ZL36**
ZL92	**T1Z92**	230	229	**T1Z35**	**ZL35**
ZL93	**T1Z93**	232	231	**T1Z34**	**ZL34**
ZL94	**T1Z94**	234	233	**T1Z33**	**ZL33**
ZL95	**T1Z95**	236	235	**T1Z32**	**ZL32**
ZL96	**T1Z96**	238	237	**T1Z31**	**VGA_CLK**
ZL97	**T1Z97**	240	239	**T1Z30**	**ZL30**
ZL98	**T1Z98**	242	241	**T1Z29**	**CLCD_CLPOWER**
ZL99	**T1Z99**	244	243	**T1Z28**	**CLCD_CLLP**
ZL100	**T1Z100**	246	245	**T1Z27**	**CLCD_CLLE**
ZL101	**T1Z101**	248	247	**T1Z26**	**CLCD_CLFP**
ZL102	**T1Z102**	250	249	**T1Z25**	**CLCD_CLCP**
ZL103	**T1Z103**	252	251	**T1Z24**	**CLCD_CLAC**
ZL104	**T1Z104**	254	253	**T1Z23**	**CLCD_B7**
ZL105	**T1Z105**	256	255	**T1Z22**	**CLCD_B6**
ZL106	**T1Z106**	258	257	**T1Z21**	**CLCD_B5**
ZL107	**T1Z107**	260	259	**T1Z20**	**CLCD_B4**
ZL108	**T1Z108**	262	261	**T1Z19**	**CLCD_B3**
ZL109	**T1Z109**	264	263	**T1Z18**	**CLCD_B2**
ZL110	**T1Z110**	266	265	**T1Z17**	**CLCD_B1**

Table 5-4 HDRZ signals (continued)

Core Tile signals	Baseboard signals	Even pins	Odd pins	Baseboard signals	Core Tile signals
ZL111	**T1Z111**	268	267	**T1Z16**	**CLCD_B0**
ZL112	**T1Z112**	270	269	**T1Z15**	**CLCD_G7**
ZL113	**T1Z113**	272	271	**T1Z14**	**CLCD_G6**
ZL114	**T1Z114**	274	273	**T1Z13**	**CLCD_G5**
ZL115	**T1Z115**	276	275	**T1Z12**	**CLCD_G4**
ZL116	**T1Z116**	278	277	**T1Z11**	**CLCD_G3**
ZL117	**T1Z117**	280	279	**T1Z10**	**CLCD_G2**
ZL118	**T1Z118**	282	281	**T1Z9**	**CLCD_G1**
ZL119	**T1Z119**	284	283	**T1Z8**	**CLCD_G0**
ZL120	**T1Z120**	286	285	**T1Z7**	**CLCD_R7**
ZL121	**T1Z121**	288	287	**T1Z6**	**CLCD_R6**
ZL122	**T1Z122**	290	289	**T1Z5**	**CLCD_R5**
ZL123	**T1Z123**	292	291	**T1Z4**	**CLCD_R4**
ZL124	**T1Z124**	294	293	**T1Z3**	**CLCD_R3**
ZL125	**T1Z125**	296	295	**T1Z2**	**CLCD_R2**
ZL126	**T1Z126**	298	297	**T1Z1**	**CLCD_R1**
ZL127	**T1Z127**	300	299	**T1Z0**	**CLCD_R0**

5.2 CLCD DVI display interface

The CT-R4F DVI combined connector (J4) is shown in Figure 5-2. The digital CLCD data from the PL111 in ARM Cortex-R4F test chip is passed to a T.M.D.S. transmitter to provide the DVI digital data and to a triple video DAC to provide the analogue RGB signals. The DDC2B interface is provided by a custom *Two-wire Interface* implemented in the CT-R4F PLD.

Note

The mechanical interconnect includes 29 signal contacts, that are divided into two sections. The first section is organized as three rows of eight contacts. The second section contains five signals that are designed specifically for analog video signals. Horizontal Sync, Vertical Sync, R, G, and B are all required for analog implementations.

A fused (1A anti-surge) +5V supply (pin 14) is provided by the CT-R4F.

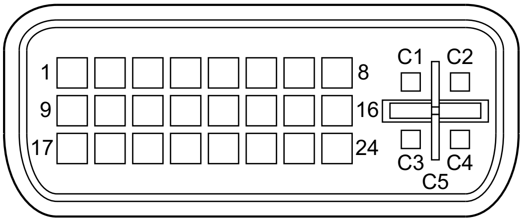


Figure 5-2 DVI Digital/Analog connector

The DVI connector signals are listed in Table 5-5 on page 5-26.

Table 5-5 DVI connector signals

Pin	Signal	Pin	Signal	Pin	Signal
1	T.M.D.S. Data2-	9	T.M.D.S. Data1-	17	T.M.D.S. Data0-
2	T.M.D.S. Data2+	10	T.M.D.S. Data1+	18	T.M.D.S. Data0+
3	T.M.D.S. Data2/4 Shield	11	T.M.D.S. Data1/3 Shield	19	T.M.D.S. Data0/5 Shield
4	T.M.D.S. Data4-	12	T.M.D.S. Data3-	20	T.M.D.S. Data5-
5	T.M.D.S. Data4+	13	T.M.D.S. Data3+	21	T.M.D.S. Data5+
6	DDC Clock	14	+5V Power	22	T.M.D.S. Clock Shield
7	DDC Data	15	Ground (for +5V)	23	T.M.D.S. Clock+
8	Analog Vertical sync	16	Hot Plug Detect	24	T.M.D.S. Clock-
C1	Analog Red	C2	Analog Green	C3	Analog Blue
C4	Analog Horizontal Sync	C5	Analog Ground (analog R, G, and B return)		

Note

The CT-R4F implements a single link, T.M.D.S. Data 3, 4, and 5 connections are not used.

5.3 Trace connectors

Trace connectors are provided on the CT-R4F. The trace connectors provide access to the *Trace Port Interface Unit* (TPIU) implemented in the ARM Cortex-R4F test chip.

The Mictor connector (part number AMP 2-767004-2) is shown in Figure 5-3.

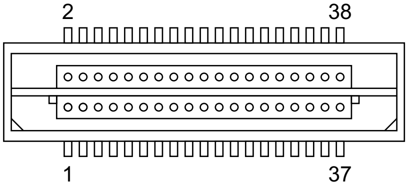


Figure 5-3 Trace Connector

Table 5-6 lists the pinout for the CT-R4F Trace Port A (J6) connector.

Table 5-6 Trace Port A (J6) connector

Trace Port signal	Pin	Pin	Trace Port signal
Not connected	1	2	Not connected
Not connected	3	4	Not connected
GND	5	6	**TRACECLKA**
Not connected	7	8	Not connected
T_nSRST	9	10	**EXTTRIGX**
T_TDO / TRACESW	11	12	**VTREFA_R**
T_RTCK	13	14	**VSUPPLYA_R**
T_TCK_IN	15	16	**TRACEDATA7**
T_TMS / SWDIO	17	18	**TRACEDATA6**
T_TDI	19	20	**TRACEDATA5**
T_nTRST	21	22	**TRACEDATA4**
TRACEDATA15	23	24	**TRACEDATA3**
TRACEDATA14	25	26	**TRACEDATA2**

Table 5-6 Trace Port A (J6) connector (continued)

Trace Port signal	Pin	Pin	Trace Port signal
TRACEDATA13	27	28	**TRACEDATA1**
TRACEDATA12	29	30	**GND**
TRACEDATA11	31	32	**GND**
TRACEDATA10	33	34	**VDDIO** (pullup)
TRACEDATA9	35	36	**TRACECTL**
TRACEDATA8	37	38	**TRACEDATA0**

Table 5-7 lists the pinout for the CT-R4F Trace Port B (J7) connector.

Table 5-7 Trace Port B (J7) connector

Trace Port signal	Pin	Pin	Trace Port signal
Not connected	1	2	Not connected
Not connected	3	4	Not connected
GND	5	6	**TRACECLKB**
Not connected	7	8	Not connected
Not connected	9	10	Not connected
Not connected	11	12	**VTREFB_R**
Not connected	13	14	Not connected
Not connected	15	16	**TRACEDATA23**
Not connected	17	18	**TRACEDATA22**
Not connected	19	20	**TRACEDATA21**
Not connected	21	22	**TRACEDATA20**
TRACEDATA31	23	24	**TRACEDATA19**
TRACEDATA30	25	26	**TRACEDATA18**
TRACEDATA29	27	28	**TRACEDATA17**
TRACEDATA28	29	30	**GND**

Table 5-7 Trace Port B (J7) connector (continued)

Trace Port signal	Pin	Pin	Trace Port signal
TRACEDATA27	31	32	**GND**
TRACEDATA26	33	34	**VDDIO** (pullup)
TRACEDATA25	35	36	**GND**
TRACEDATA24	37	38	**TRACEDATA16**

5.4 AXI bus timing specification

This section describes the timing for the CT-R4F multiplexed AXI system bus.

5.4.1 Core Tile timing and the AMBA 3 AXI Protocol

The worst case timing figures at the CT-R4F headers are shown in Table 5-8 on page 5-31 and Table 5-9 on page 5-33. You must use these figures as a guideline when designing your own boards.

The system bus on RealView Logic Tiles and baseboards is routed between FPGAs. The exact performance of a system depends on the timing parameters of the baseboard and all tiles in the system. Some allowance also has to be made for clock skew, routing delays and number of modules (that is, loading).

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

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

AXI Clock

The AMBA 3 AXI protocol includes a single clock signal, **ACLK**. All input signals are sampled on the rising edge of **ACLK**. All output signal changes must occur after the rising edge of **ACLK**.

There must be no combinatorial paths between input and output signals on both master and slave interfaces.

The ARM Cortex-R4F test chip has two external clocks:

OC_AXI_CLK2

This clock drives the test chip AXI ports signal multiplexers and is exported at the HDRZ header as **CLK_OUT_MINUS1**.

OC_AXI_CLK

This clock drives the pad side of the test chip AXI matrix and is exported to the HDRZ header as **CLK_NEG_DN_OUT**.

AXI Reset

The AMBA 3 AXI protocol includes a single active LOW reset signal, **ARESETn**. The reset signal can be asserted asynchronously, but de-assertion must be synchronous after the rising edge of **ACLK**.

Note

nSYSPOR is the CT-R4F global AXI reset (**ARESETn**).

5.4.2 Timing parameters

The following timings are based on the grouped delays for the CT-R4F. They include the test chip input and output delay, and the delay because of the AXI bus multiplexing.

Table 5-8 shows the Global timing parameters.

Table 5-8 Global timing parameters

Parameter	Description	Min	Max
T_{clk}	**CLK_POS_DN_OUT** clock frequency	33MHz	40MHz
T_{isrst}	**nSYSPOR** de-asserted setup time before **CLK_POS_DN_OUT**	5ns	–

Note

During reset the following interface requirements apply:

- a master interface must drive **ARVALID**, **AWVALID**, and **WVALID** LOW
- a slave interface must drive **RVALID** and **BVALID** LOW.

All other signals can be driven to any value.

A master interface must begin driving **ARVALID**, **AWVALID**, or **WVALID** HIGH only at a rising **CLK_IN_MINUS1** edge after **nSYSPORESET** is HIGH.

The CT-R4F uses a multiplexed AXI bus. Figure 5-4 on page 5-32 shows the CT-R4F multiplexed AXI bus input setup timing.

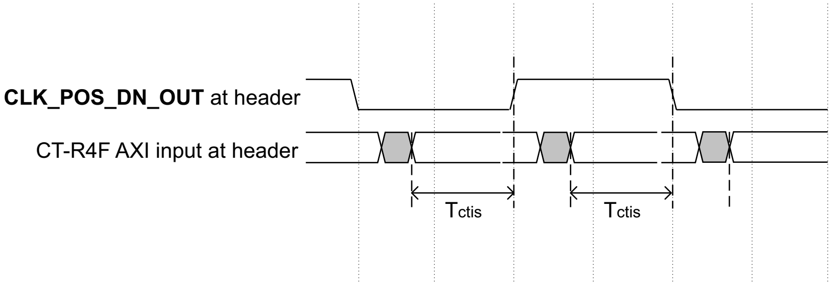


Figure 5-4 CT-R4F multiplexed AXI bus input setup timing

T_{ctis} Input setup to clock

This is the longest time that the CT-R4F requires a valid AXI bus signal level to be presented at the Core Tile header before a rising or falling edge of **CLK_POS_DN_OUT**.

Figure 5-5 shows the CT-R4F multiplexed AXI bus output valid timing.

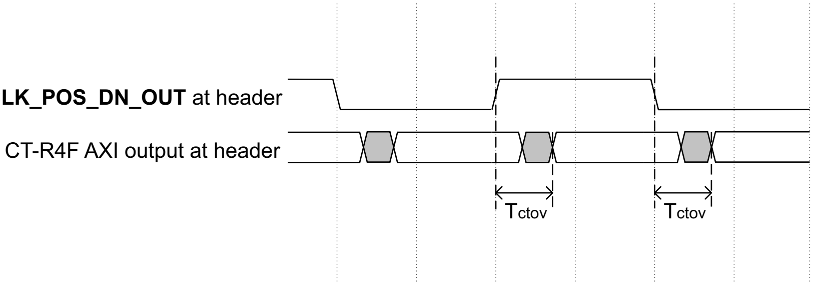


Figure 5-5 CT-R4F multiplexed AXI bus output valid timing

T_{ctov} Core Tile output valid after clock

This is the longest delay between a rising or falling edge of **CLK_POS_DN_OUT** and a valid AXI bus output signal level arriving at the CT-R4F header.

Table 5-9 shows the CT-R4F AXI bus worst case timings when characterized for a slow process, a **VDDCORE** voltage of 0.95V, and a core temperature of 85°C.

Table 5-9 CT-R4F multiplexed AXI bus timing parameters

Parameter	Description	Max
T_{ctis}	Input setup time to either edge of **CLK_POS_DN_OUT**	6ns
T_{ctov}	Output valid time from either edge of **CLK_POS_DN_OUT**	5ns

Appendix A
Specifications

This appendix contains the specifications for the CT-R4F. It contains the following sections:

- *Electrical specification* on page A-2
- *Mechanical details* on page A-4.

Note

See *AXI bus timing specification* on page 5-30 for the timing specifications for the CT-R4F AXI interfaces.

A.1 Electrical specification

This section provides details of the voltage and current characteristics for the CT-R4F.

A.1.1 Bus interface characteristics

Table A-1 shows the Core Tile electrical characteristics for the system bus interface. The CT-R4F uses 3.3V and 5V sources.

Table A-1 Core Tile electrical characteristics

Symbol	Description	Min	Max	Unit
3V3	Supply voltage (interface signals)	3.1	3.5	V
5V	Supply voltage (regulators)	4.75	5.25	V
V_{IH}	High-level input voltage	2.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{OH}	High-level output voltage	2.4	–	V
V_{OL}	Low-level output voltage	–	0.4	V
C_{IN}	Input capacitance	–	20	pF

A.1.2 Current requirements

Table A-2 shows current requirements measured at room temperature and nominal voltage.

The power regulators are all powered from the 5V supply. The maximum current drawn from this supply is 3A. The maximum current requirements for the individual supplies are listed in Table A-2. Interface logic, the ADC, and the on board SDRAM memory are powered directly from the 3V3 supply.

Table A-2 Current requirements (max)

Component	VDDCORE 1V0	VDDPLL 2V5	VDDTC 2V6	VDDPLD 1V8	3V3
ARM Cortex-R4F core	4A	–	–	–	–
Test chip PLL	–	250mA	–	–	–

Table A-2 Current requirements (max) (continued)

Component	VDDCORE 1V0	VDDPLL 2V5	VDDTC 2V6	VDDPLD 1V8	3V3
Test chip I/O	–	–	2A	–	–
CT-R4F PLD core	–	–	–	250mA	–
CT-R4F PLD I/O, ADC, and SDRAM	–	–	–	–	1A

A.2 Mechanical details

If you require mechanical details of the CT-R4F for use with a custom baseboard, you must contact ARM Support for assistance.

Glossary

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

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

AXI Advanced eXtensible Interface. An ARM open standard bus protocol.

AMBA 3 Advanced Microcontroller Bus Architecture version 3.

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

DCC Debug Communications Controller. The DCC is the set of resources that the external DBGTAP debugger uses to communicate with a piece of software running on the core.

DMA Direct Memory Access.

EB RealView Emulation Baseboard. A hardware platform used for system prototyping and debugging of ARM microprocessors.

FPGA Field Programmable Gate Array.

ICE In Circuit Emulator. A interface device for configuring and debugging processor cores.

I/O Input/Output.

JTAG Joint Test Action Group. The committee that defined the IEEE test access port and boundary-scan standard.

LED Light Emitting Diode.

Multi-ICE A system for debugging embedded processor cores using a JTAG interface.

PCI Peripheral Component Interconnect. A circuit board level bus interconnect.

PISMO Memory specification for plug in memory modules.

PLD Programmable Logic Device.

PLL Phase-Locked Loop, a type of programmable oscillator.

RAM Random Access Memory.

RVI RealView ICE. A system for debugging embedded processor cores using a JTAG interface.

SCU Snoop Control Unit. A system block that maintains coherency across the L1 cache memory systems in the ARM Cortex-R4.

USB Universal Serial Bus. Hardware interface for connecting peripheral devices.