ARM® CoreTile Express A15x2

Cortex®-A15 MPCore (V2P-CA15)

Technical Reference Manual

ARM®

ARM DUI 0604F (ID121616)

ARM CoreTile Express A15x2
Technical Reference Manual

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
02 December 2011	A	Non-Confidential	First release for V2P-CA15 TRM
15 July 2012	B	Non-Confidential	Second release for V2P-CA15 TRM
06 August 2012	C	Non-Confidential	Third release for V2P-CA15 TRM
12 October 2012	D	Non-Confidential	Fourth release for V2P-CA15 TRM
31 March 2013	E	Non-Confidential	Fifth release for V2P-CA15 TRM
16 December 2016	F	Non-Confidential	Sixth release for V2P-CA15 TRM

Proprietary Notice

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

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

Product Status

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

Web Address

`http://www.arm.com`

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

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

CE Declaration of Conformity

CE

The system should be powered down when not in use.

The daughterboard 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
ARM CoreTile Express A15x2 Technical Reference Manual

Preface

This preface introduces the *CoreTile Express A15×2 Technical Reference Manual*. It contains the following sections:

About this book

This book is for the CoreTile Express A15×2 daughterboard.

Intended audience

This document is written for experienced hardware and software developers to aid the development of ARM-based products using the CoreTile Express A15×2 daughterboard with the Motherboard Express µATX as part of a development system.

Using this book

This book is organized into the following chapters:

Chapter 1 *Introduction*

Read this for an introduction to the CoreTile Express A15×2 daughterboard.

Chapter 2 *Hardware Description*

Read this for a description of the hardware present on the daughterboard.

Chapter 3 *Programmers Model*

Read this for a description of the configuration registers present on the daughterboard.

Appendix A *Signal Descriptions*

Read this for a description of the signals present on the daughterboard.

Appendix B *HDLCD controller*

Read this for a description of the HDLCD controller in the Cortex-A15 test chip.

Appendix C *Electrical Specifications*

Read this for a description of the electrical specifications of the daughterboard.

Appendix D *Revisions*

Read this for a description of the technical changes between released issues of this book.

Glossary

The *ARM Glossary* is a list of terms used in ARM documentation, together with definitions for those terms. The *ARM Glossary* does not contain terms that are industry standard unless the ARM meaning differs from the generally accepted meaning.

See *ARM Glossary*, `http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html`.

Conventions

This book uses the conventions that are described in:

- *Typographical conventions* on page ix
- *Timing diagrams* on page ix
- *Signals* on page x.

Typographical conventions

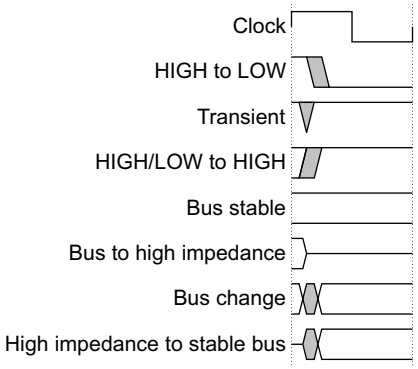
The following table describes the typographical conventions:

Style	Purpose
italic	Introduces special terminology, denotes cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`monospace italic`	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The figure named *Key to timing diagram conventions* 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

Timing diagrams sometimes show single-bit signals as HIGH and LOW at the same time and they look similar to the bus change shown in *Key to timing diagram conventions*. If a timing diagram shows a single-bit signal in this way then its value does not affect the accompanying description.

Signals

The signal conventions are:

Signal level The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:
- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lower-case n At the start or end of a signal name denotes an active-LOW signal.

Additional reading

This section lists publications by ARM and by third parties.

See Infocenter, http://infocenter.arm.com, for access to ARM documentation.

ARM publications

This book contains information that is specific to this product. See the following documents for other relevant information:

- *Motherboard Express µATX Technical Reference Manual* (ARM DUI 0447)
- *Versatile™ Express Configuration Technical Reference Manual* (ARM DDI 0496)
- *Programmer Module (V2M-CP1)* (ARM DDI 0495)
- *LogicTile Express 3MG Technical Reference Manual* (ARM DUI 0449)
- *LogicTile Express 13MG Technical Reference Manual* (ARM DUI 0556)
- *Versatile™ Express Boot Monitor Technical Reference Manual* (ARM DUI 0465)
- *Cortex®-A15 Technical Reference Manual* (ARM DDI 0438)
- *AMBA® Network Interconnect (NIC-301) Technical Reference Manual* (ARM DDI 0397)
- *ARM PrimeCell™ DDR2 Dynamic Memory Controller (PL341) Technical Reference Manual* (ARM DDI 0418)
- *ARM PrimeCell Static Memory Controller (PL350 series) Technical Reference Manual* (ARM DDI 0380)
- *AMBA® DMA Controller DMA-330 Technical Reference Manual* (ARM DDI 0424)
- *ARM Watchdog Module (SP805) Technical Reference Manual* (ARM DDI 0270)
- *ARM PrimeCell External Bus Interface (PL220) Technical Reference Manual* (ARM DDI 0249)

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

- *ARM® DSTREAM™ System and Interface Design Reference* (ARM DUI 0449)
- *ARM® DSTREAM™ Setting up the Hardware* (ARM DUI 0481)
- *ARM® DSTREAM™and RVT™ Using the Debug Hardware Configuration Utilities* (ARM DUI 0498)
- *ARM CoreSight Components Technical Reference Manual* (ARM DDI 0314)

- Application note AN283, *Example LogicTile Express 3MG design for a CoreTile Express A15×2*.

Feedback

ARM welcomes feedback on this product and its documentation.

Feedback on this product

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

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to `errata@arm.com`. Give:

- The title.
- The number, ARM DUI 0604F.
- The page numbers to which your comments apply.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Chapter 1
Introduction

This chapter provides an introduction to the CoreTile Express A15×2 daughterboard. It contains the following sections:

1.1 About the CoreTile Express A15x2 daughterboard

The CoreTile Express A15×2 daughterboard is designed as a platform for developing systems based on the *Advanced Microcontroller Bus Architecture* (AMBA®) that use the *Advanced eXtensible Interface* (AXI™) or custom logic for use with ARM cores.

You can use the CoreTile Express A15×2 daughterboard to create prototype systems.

Note

You can use the CoreTile Express A15×2 daughterboard with a Motherboard Express µATX. See *System interconnect signals* on page 2-6 for information about interconnection.

You can also use the CoreTile Express daughterboard with a custom-design motherboard. See the *Programmer Module (V2M-CP1)*.

The daughterboard includes the following features:

- Cortex-A15 MPCore test chip, with NEON, the advanced *Single Instruction Multiple Data* (SIMD) extension and FPU, that contains a dual-core A15 cluster operating at 1GHz.
- Cortex-A15 MPCore test chip internal AXI subsystem operating at 600MHz.
- Simple configuration with V2M-P1 motherboard:
  - Configuration EEPROM.
  - *Daughterboard Configuration Controller*.
- Nine programmable oscillators.
- 2GB of daughterboard DDR2 32-bit memory operating at 400MHz.
- *High Definition LCD* (HDLCD) controller that supports up to 1920×1080p video at 60Hz, 165MHz pixel clock.
- CoreSight software debug and 32-bit trace ports.
- HDRX header with one AMBA AXI master bus port that connects to the other daughterboard site on the V2M-P1 motherboard.
- HDRY header with four buses to the motherboard:
  - *Static Memory Bus* (SMB).
  - *MultiMedia Bus* (MMB).
  - *Configuration Bus* (CB).
  - *System Bus* (SB).
- *Power Supply Units* (PSUs) for the Cortex-A15 test chip and DDR2 memory.
- CPU voltage control and current, temperature, and power monitoring.

Note

The Cortex-A15 test chip does not support TrustZone®.

Figure 1-1 on page 1-3 shows the layout of the daughterboard.

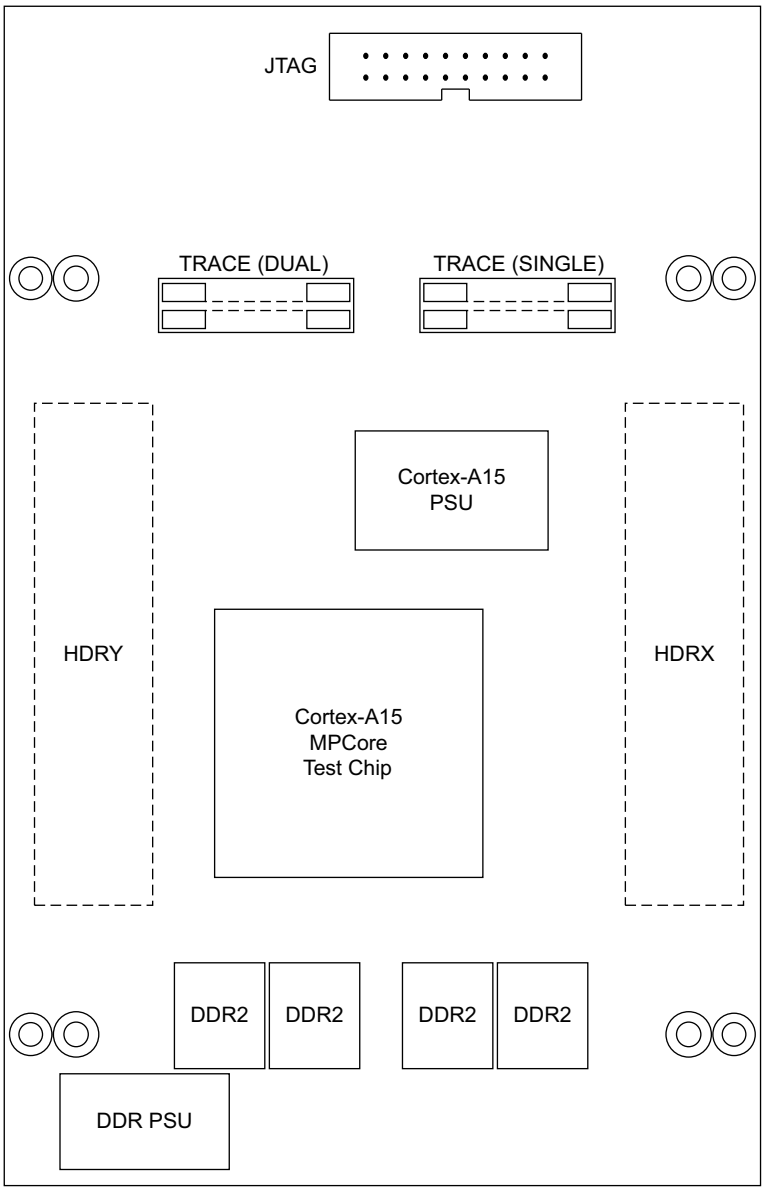


Figure 1-1 CoreTile Express A15×2 daughterboard layout

Note

All components in Figure 1-1, except the HDRY and HDRX headers, are on the upper face of the daughterboard facing away from the motherboard.

The HDRX and HDRY headers are on the lower face of the daughterboard facing towards the motherboard.

1.2 Precautions

This section contains advice about how to prevent damage to your daughterboard.

1.2.1 Ensuring safety

The daughterboard is supplied with a range of DC voltages. Power is supplied to the daughterboard through the header connectors.

Warning

Do not use the board near equipment that is sensitive to electromagnetic emissions, for example, medical equipment.

1.2.2 Preventing damage

The daughterboard 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 permits electromagnetic emissions.

Caution

To avoid damage to the daughterboard, observe the following precautions:

- Never subject the board to high electrostatic potentials. Observe *ElectroStatic Discharge* (ESD) precautions when handling any board.
- Always wear a grounding strap when handling the board.
- Only hold the board by the edges.
- Avoid touching the component pins or any other metallic element.
- Do not use the board near a transmitter of electromagnetic emissions.

Chapter 2
Hardware Description

This chapter describes the hardware on the CoreTile Express A15×2 daughterboard. It contains the following sections:

2.1 CoreTile Express A15x2 daughterboard architecture

Figure 2-1 shows a block diagram of the daughterboard.

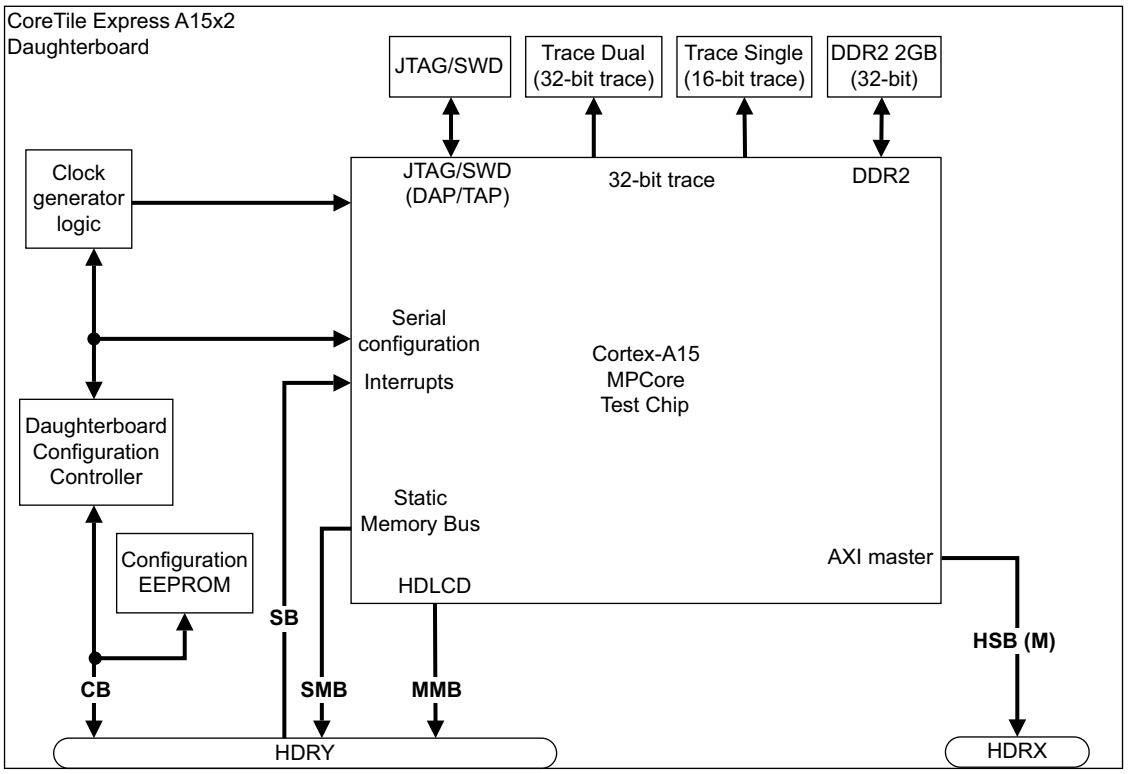


Figure 2-1 CoreTile Express A15×2 daughterboard block diagram

The daughterboard contains the following devices and interfaces:

Cortex-A15 MPCore test chip

The *Motherboard Configuration Controller* (MCC) and the *Daughterboard Configuration Controller* configure the test chip at power-up or reset.

Daughterboard Configuration Controller

The *Daughterboard Configuration Controller* initiates, controls, and configures the test chip.

An MCC on the Motherboard Express µATX configures the daughterboard and communicates with the *Daughterboard Configuration Controller* to configure the test chip.

Configuration EEPROM

The daughterboard EEPROM contains information for identification and detection of the daughterboard.

DDR2 memory

The daughterboard supports 2GB of 32-bit DDR2 on-board memory.

Clock generators

The daughterboard provides nine on-board OSCCLKS to drive the CPU and internal AXI, AXIM, DDR2, SMC, and HDLCD interfaces.

CoreSight software debug and trace ports

The Cortex-A15 MPCore test chip CoreSight system supports the SWD and JTAG protocols.

A 32-bit trace interface is provided through the standard dual 16-bit *Matched Impedance ConnecTOR* (MICTOR) connectors.

System interconnect buses

The HDRX header connects the AXI master bus, HSB M, to the external AXI slave on the other daughterboard in Site 2 of the V2M-P1 Motherboard Express.

Note

ARM recommends that you fit the CoreTile Express A15×2 daughterboard in site 1, and any optional FPGA daughterboard, for example the V2F-1XV5, in site 2. See *System interconnect signals* on page 2-6.

The HDRY header connects the *Configuration Bus* (CB), the *System Bus* (SB), the *Static Memory Bus* (SMB), and the *MultiMedia Bus* (MMB) to the V2M-P1 Motherboard Express.

See *System interconnect signals* on page 2-6.

2.2 Cortex-A15 MPCore test chip

Figure 2-2 shows the main components of the test chip.

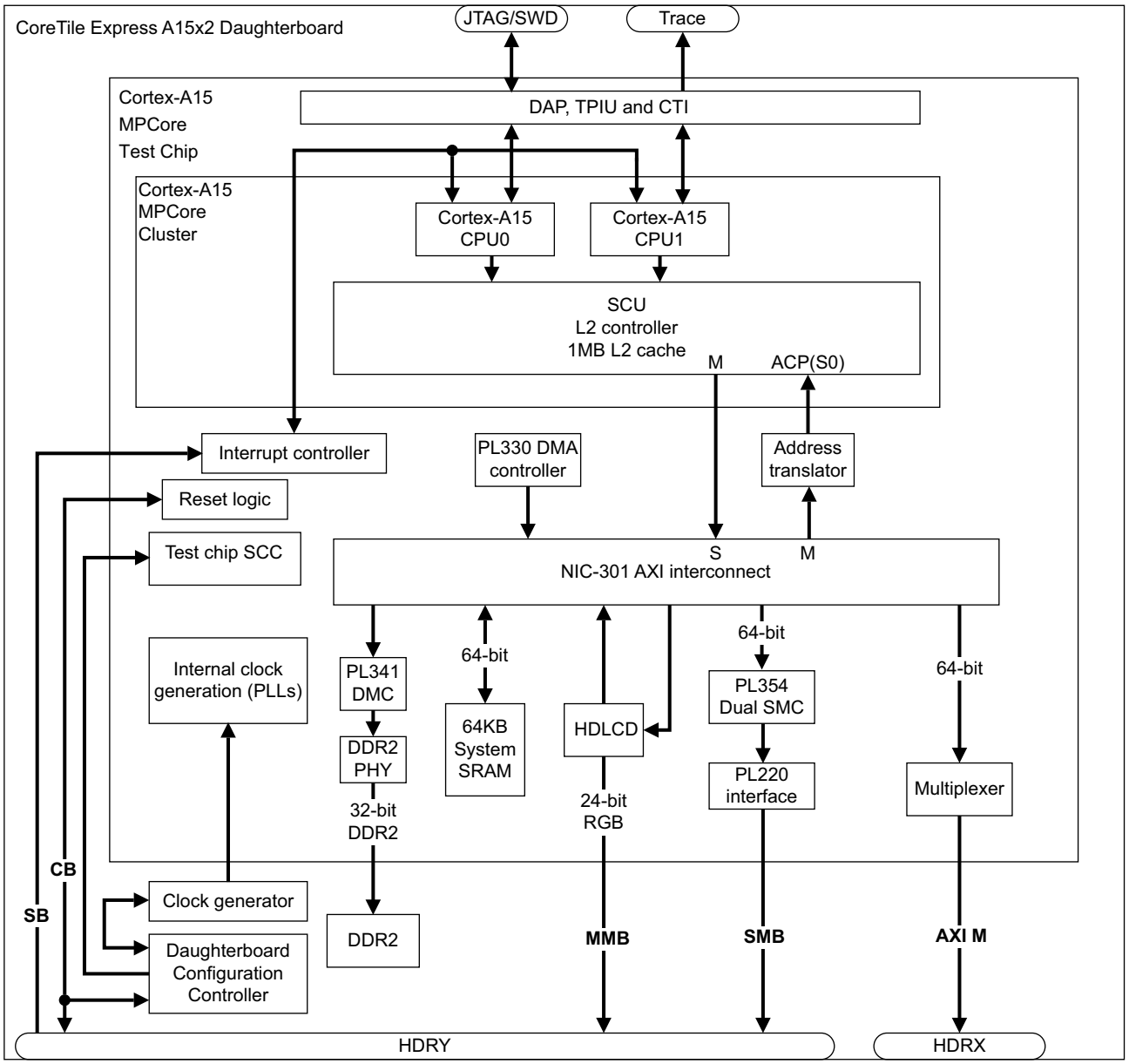


Figure 2-2 Top-level view of the Cortex-A15 MPCore test chip components

Note

Bus lines with single-headed arrows indicate the direction of control, not the direction of data flow. That is, each arrow points from the bus master to the bus slave.

Cortex-A15 MPCore test chip

The test chip includes the following components and interfaces:

- Cortex-A15 dual-core cluster with 32KB I/D cache, NEON and FPU, operating at 1GHz.
- 1MB L2 cache.

- PL341 32-bit *Double Data Rate* 2 (DDR2) *Dynamic Memory Controller* (DMC) interface to the onboard 2GB DDR2 memory.
- PL354 32-bit SMB controller. This connects to the motherboard peripherals.
- 24-bit HDLCD video controller that drives the MMB to the MUXFPGA on the V2M-P1 Motherboard Express.
- Multiplexed 64-bit AXI master interface.
- CoreSight debug and trace interface to the onboard connectors.
- *Serial Configuration Controller* (SCC) interface:
  - Configures the test chip PLLs during power-up or reset.
  - Connects to *Daughterboard Configuration Controller*.
- Interrupts interface:
  - Connects interrupt signals from the V2M-P1 motherboard to the *Generic Interrupt Controller* (GIC) in the test chip.

2.3 System interconnect signals

This section provides an overview of the signals present on the header connectors. It contains the following subsections:

2.3.1 Overview of system interconnect signals

Figure 2-3 on page 2-7 shows the daughterboard system interconnect to the Motherboard Express µATX development system, and to an optional LogicTile Express FPGA daughterboard. For more information about the global interconnect scheme, see the *Motherboard Express µATX Technical Reference Manual*.

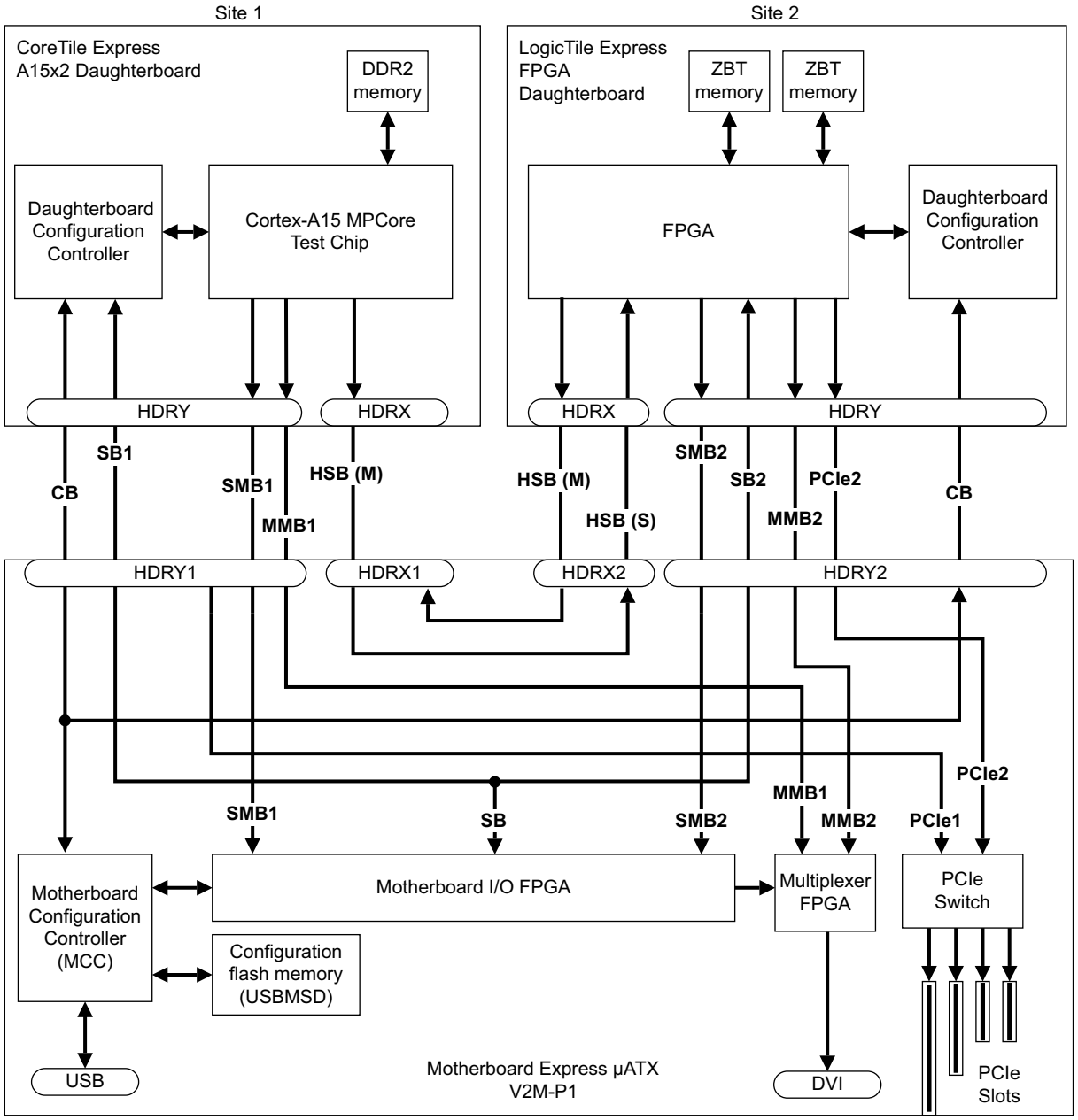


Figure 2-3 System connect example with optional LogicTile Express FPGA daughterboard

Note

- ARM recommends that you fit CoreTile Express daughterboards, including the CoreTile Express A15×2 daughterboard, on site 1, and that you fit LogicTile Express FPGA daughterboards on site 2.
- The CoreTile Express A15×2 daughterboard does not have a HSB slave port or a PCI express root complex.

- Application note AN283, *Example LogicTile Express 3MG design for a CoreTile Express A15×2*, that ARM provides, implements an example AMBA system using the LogicTile Express 3MG daughterboard to interconnect with the CoreTile Express A15×2 daughterboard. See the documentation supplied on the accompanying media and the *Application Notes* listing for more information at, http://infocenter.arm.com.

2.3.2 *High-Speed Bus* (HSB) to other daughterboard

The HSB link to the other daughterboard consists of one bus between the Cortex-A15 MPCore test chip and the daughterboard fitted in the other site on the motherboard. This is a 64-bit multiplexed AXI master bus, HSB M, to the external AXI slave on the other daughterboard in Site 2.

The HSB connection to the other daughterboard is through the HDRX header on the daughterboard, dedicated headers HDRX1 and HDRX2 on the motherboard, and the HDRX header on the other daughterboard.

Note

For information about the multiplexing scheme for the AXI buses, see Appendix A *Signal Descriptions*.

Note

Application note AN283, *Example LogicTile Express 3MG design for a CoreTile Express A15×2*, provides an example AXI design implementing an external multiplexed AXI master bus at the HDRX header on the FPGA daughterboard in site 2.

2.3.3 *Static Memory Bus* (SMB)

The SMB connects the Cortex-A15 test chip SMC to the motherboard. You can use it to access the motherboard peripherals such as the following:

- NOR Flash
- SRAM
- Ethernet
- USB
- *MultiMedia Card* (MMC)
- *Compact Flash* (CF)
- *Keyboard and Mouse Interface* (KMI)
- CLCD
- UARTs
- System registers.

2.3.4 *MultiMedia Bus* (MMB)

The *MultiMedia Bus* (MMB) consists of a video bus that connects the 24-bit RGB HDLCD controller directly to the motherboard MUXFPGA. The motherboard IOFPGA implements a CLCD controller. The motherboard MUXFPGA selects the source for the motherboard DVI connector from:

- The MMB from the CoreTile Express A15×2, that is, the 24-bit RGB from the HDLCD controller.
- The MMB from the LogicTile Express daughterboard in site 2.

- The CLCD controller in the motherboard IOFPGA.

2.3.5 *System Bus* (SB)

The SB connects 26 interrupt lines from motherboard peripherals to the GIC in the MPCore cluster in the test chip.

2.3.6 *Configuration Bus* (CB)

The CB connects the *Daughterboard Configuration Controller* to the MCC. The *Daughterboard Configuration Controller* configures the OSCCLKs on the daughterboard and the Cortex-A15 test chip SCC registers. The *Daughterboard Configuration Controller* also controls daughterboard resets and temperature monitoring. The CB connects the daughterboard EEPROM directly to the MCC and enables automatic detection and identification of the daughterboard at power-up or reset.

2.4 Power-up configuration and resets

This section describes the CoreTile Express A15×2 daughterboard power-up configuration and resets. It contains the following subsections:

- *Configuration architecture*
- *Resets* on page 2-13
- *Configuration and reset signals* on page 2-14.

2.4.1 Configuration architecture

Figure 2-4 shows the power-up, configuration, and reset architecture when the CoreTile Express A15×2 daughterboard is fitted to a Motherboard Express, V2M-P1.

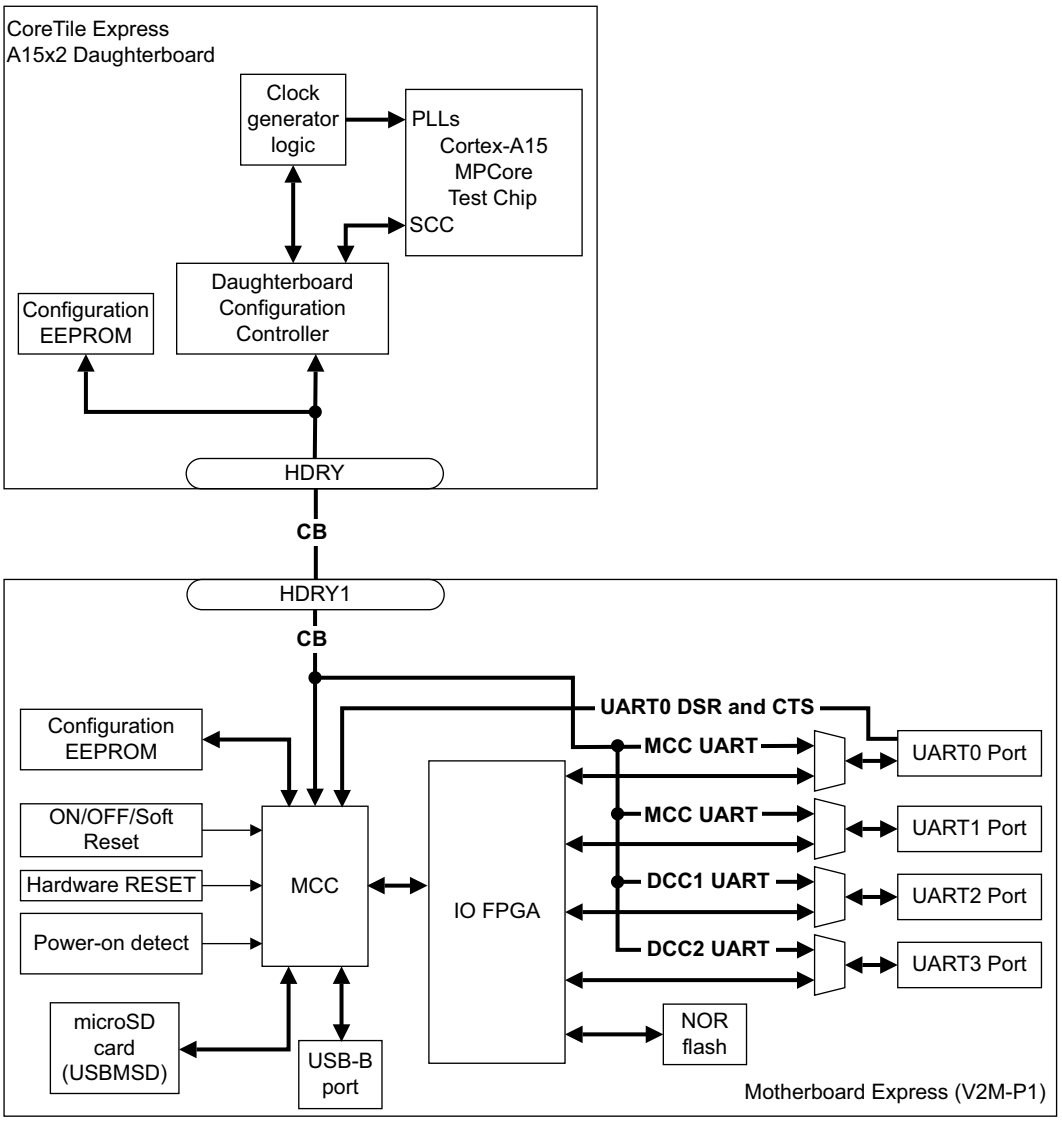


Figure 2-4 CoreTile Express A15×2 configuration architecture with Motherboard Express, V2M-P1

Note

ARM recommends that you fit the CoreTile Express A15×2 daughterboard in site 1.

The configuration environment of the CoreTile Express A15×2 daughterboard fitted to the Motherboard Express, V2M-P1, consists of the following hardware components:

- MCC on the Motherboard Express, V2M-P1.
- *Daughterboard Configuration Controller* on the CoreTile Express daughterboard and on the LogicTile Express daughterboard.
- Configuration microSD card or *Universal Serial Bus Mass Storage Device* (USBMSD) on the Motherboard Express, V2M-P1.
- Configuration EEPROM on the Motherboard Express, V2M-P1.
- Clock generator logic on the V2P-CA15 daughterboard.
- ON/OFF/Soft Reset and Hardware RESET buttons on the Motherboard Express, V2M-P1.
- USB-B port on the Motherboard Express, V2M-P1.
- Four UART ports on the Motherboard Express, V2M-P1.
- NOR flash on the Motherboard Express, V2M-P1.
- Power-on detect on the Motherboard Express, V2M-P1.
- Configuration EEPROM on the V2P-CA15 daughterboard.
- HDRY headers on the Motherboard Express, V2M-P1, and V2P-CA15 daughterboard.

Figure 2-5 on page 2-12 shows the power-up, configuration, and reset architecture when the CoreTile Express A15×2 daughterboard is fitted to a V2M-CP1 Programmer Module and custom motherboard built under the ARM *Design Assist Program*.

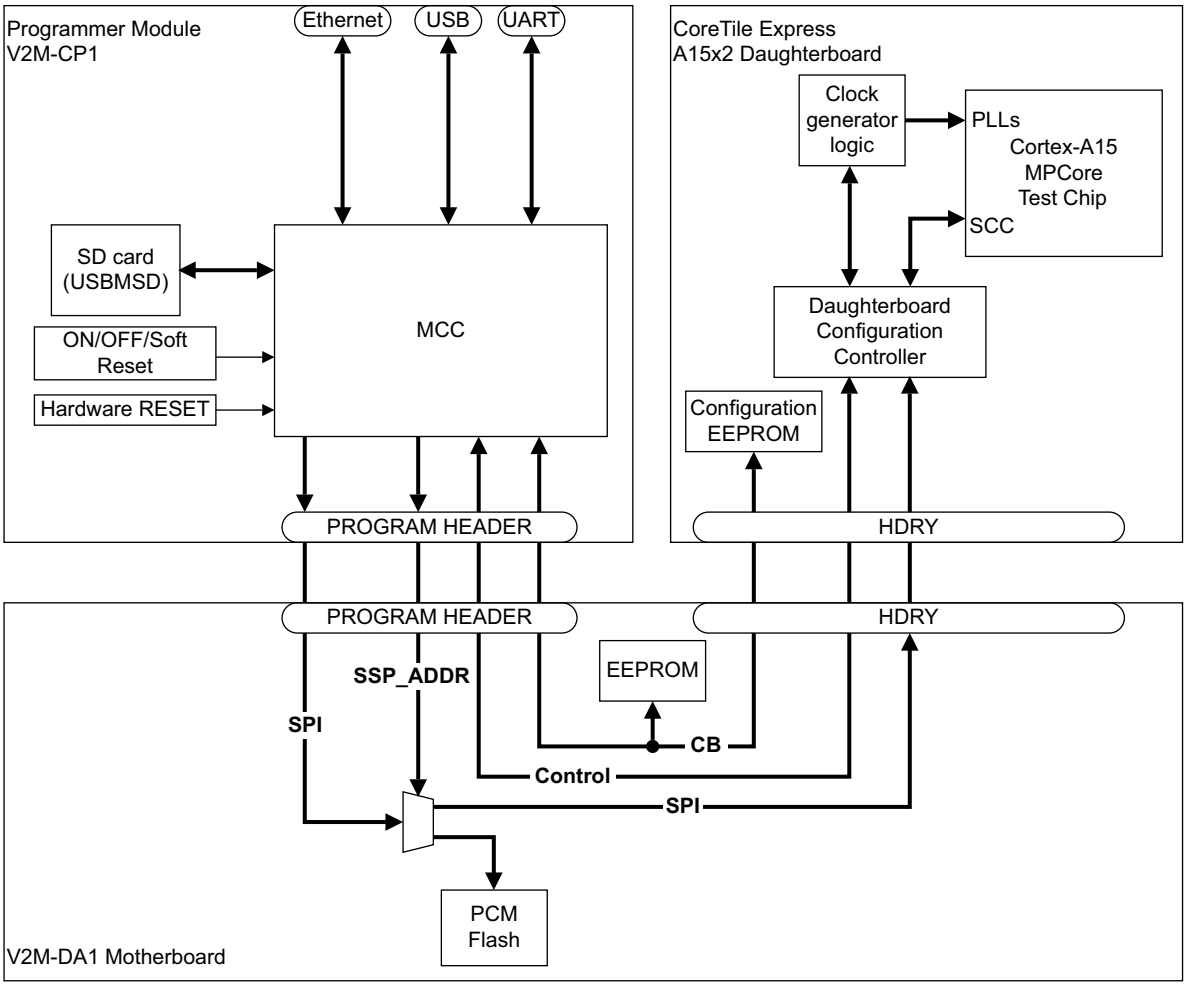


Figure 2-5 CoreTile Express A15×2 configuration architecture with custom motherboard

The configuration environment of a V2P-CA15 daughterboard fitted to a custom motherboard and a V2M-CP1 Programmer Module consists of:

- MCC on the V2M-CP1.
- *Daughterboard Configuration Controller* on the V2P-CA15 daughterboard.
- Configuration microSD card or USBMSD on the V2M-CP1 Programmer Module.
- ON/OFF/Soft Reset and Hardware RESET buttons on the V2M-CP1 Programmer Module.
- USB port on the V2M-CP1 Programmer Module.
- Ethernet port on the V2M-CP1 Programmer Module.

  Note

  The V2M-CP1 Programmer Module does not support the ethernet port.

- Configuration EEPROM on the custom motherboard.
- Configuration EEPROM on the V2P-CA15 daughterboard.
- Clock generator logic on the V2P-CA15 daughterboard.

2.4.2 System configuration

See the *Versatile™ Express Configuration Technical Reference Manual* and *Motherboard Express (µATX)* or *Programmer Module (V2M-CP1)* for information on how to configure the V2P-CA15 daughterboard using the configuration files on the motherboard.

Caution

ARM recommends that you use the configuration files for all system configuration. *Test chip SCC registers* on page 3-11 and *Programmable peripherals and interfaces* on page 3-35 describe registers that directly modify the test chip configuration.

2.4.3 Resets

The *Daughterboard Configuration Controller* manages reset signals between the motherboard and the daughterboard test chip in response to reset requests from the motherboard MCC as a result of, for example, pressing the motherboard Power/Reset push button.

Figure 2-6 shows an overview of the daughterboard reset signals, and includes configuration signals.

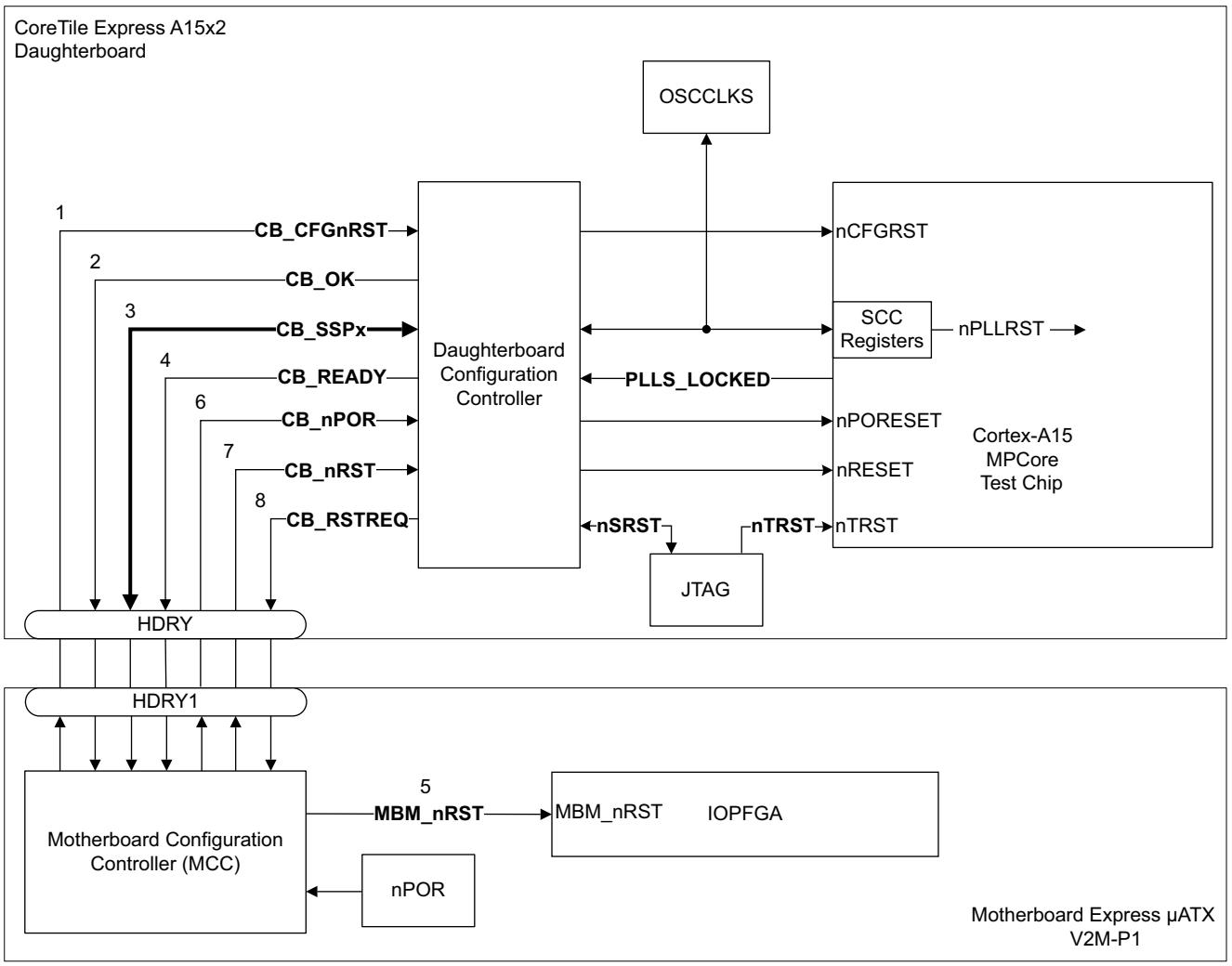


Figure 2-6 CoreTile Express A15×2 daughterboard resets

Note

The numbers in Figure 2-6 on page 2-13 represent the stages in the reset and configuration process. See Figure 2-7.

2.4.4 Configuration and reset signals

This section describes the configuration and reset signals and their sequences and timings.

Figure 2-7 shows the CoreTile Express A15×2 daughterboard power-up configuration and reset timing cycle.

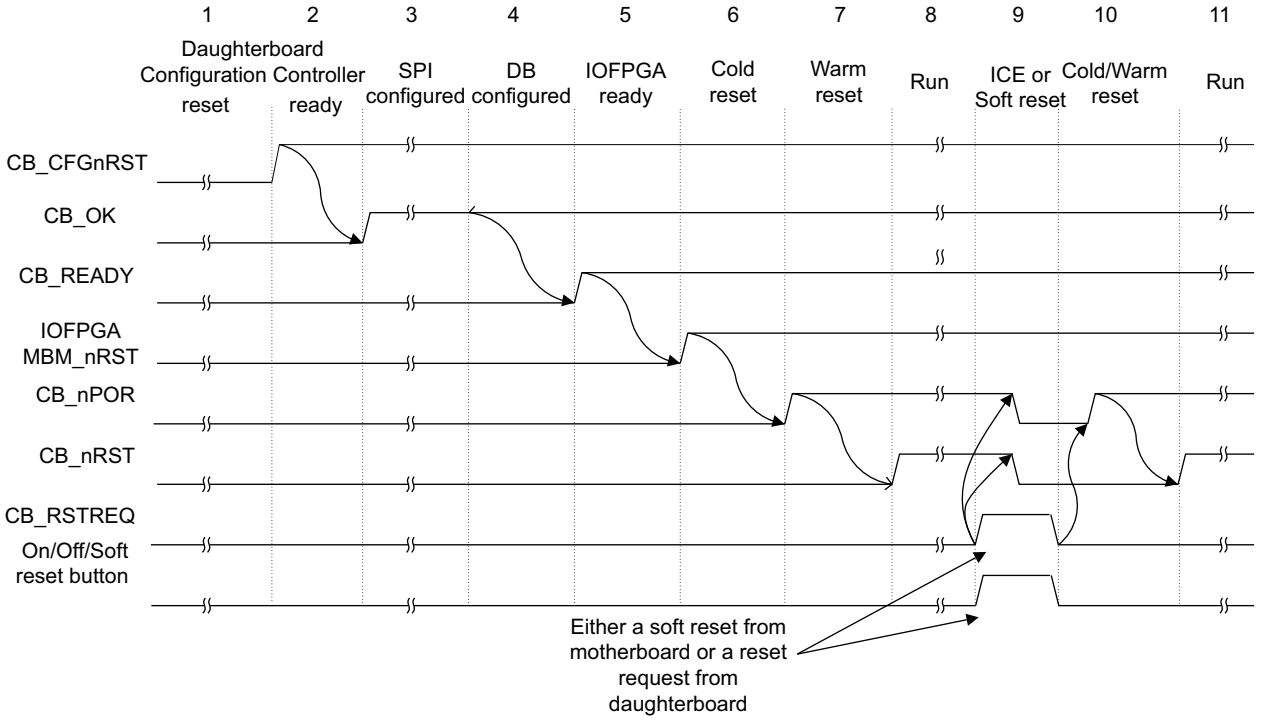


Figure 2-7 CoreTile Express A15×2 daughterboard configuration and reset timing cycle

Table 2-1 shows the Cortex-A15 MPCore test chip configuration and reset signals.

Table 2-1 Configuration and reset signals

Reset source	Destination	Description
CB_CFGnRST	*Daughterboard Configuration Controller*	Initiate the daughterboard configuration process.
CB_OK	MCC	Daughterboard configuration system ready.
CB_SSPx[a]	*Daughterboard Configuration Controller*-MCC	Signal transactions during time period 3, *DB configuration*, that cause the *Daughterboard Configuration Controller* to configure the Cortex-A15 MPCore test chip SCC registers and daughterboard OSCCLKs.

Table 2-1 Configuration and reset signals (continued)

Reset source	Destination	Description
CB_READY	MCC	Daughterboard configured and ready. The **PLLS_LOCKED**[a] signal and the control signals[a] between the *Daughterboard Configuration Controller*, Cortex-A15 MPCore test chip, and the daughterboard OSCLCLKS indicate to the *Daughterboard Configuration Controller* and MCC that the PLLs are locked and that the daughterboard OSCCLKS are operating.
CB_nPOR	Test chip internal signal **nPORESET**	This is the main power-on-reset that resets the entire test chip logic and the clock generation logic, except for the PLLs.
CB_nRST	Test chip internal signal **nRESET**	This is a hard reset from the motherboard that resets the Cortex-A15 dual core cluster. The following internal resets, that the SCC registers control, enable reset of specific sections of the test chip independently of **nRESET**: **nCPURESET[1:0]** Resets the entire CPU including NEON-VPF, debug, PTM, break-point, and watch-point logic in the CLK domain. **nCORERESET[1:0]** Resets the entire CPU including NEON-VFP, debug, PTM, but excludes break-point and watch-point logic. **nCXRESET[1:0]** Resets NEON-VFP. **nDBGRESET[1:0]** Resets the debug, PTM, break-point, and watch-point in the CLK domain. **nPRESETDBG** Resets initialized shared debug, APB, CTI, and CTM logic in the PCLKDBG domain. **nL2RESET** Resets the L2 logic, interrupt controller and timer logic. See *Test chip SCC Register 1* on page 3-16 for information on controlling these internal resets in the Cortex-A15 MPCore test chip. See the *Cortex-A15 Technical Reference Manual* for information on using these internal resets.
JTAG **nTRST**[a]	Test chip internal signal **nTRST**	This is the test logic reset to the Cortex-A15 MPCore test chip TAP controller and the *Daughterboard Configuration Controller*.
JTAG **nSRST**[a]	**CB_RSTREQ** to motherboard MCC	If an external source asserts the JTAG **nSRST** signal, the daughterboard generates a reset request to the motherboard MCC. The motherboard hardware is reset, and the MCC asserts **CB_nRST** and, optionally, **CB_nPOR**[b]. **nSRST** remains LOW until the reset sequence completes.
CB_RSTREQ[b]	MCC	Reset request from the daughterboard to the motherboard.
Test Chips Watchdogs	Test chip internal **nPORESET**	If the internal test chip watchdog timers are configured and trigger, they force an internal test chip **nPORESET**. The external system components on the motherboard are not reset.

a. Figure 2-7 on page 2-14 does not include these signals.

b. The reset request from the daughterboard results in the motherboard asserting **CB_nRST**. You can optionally assert **CB_nPOR** by correctly defining the **ASSERTNPOR** control value, that can be TRUE or FALSE in the `config.txt` motherboard configuration file. See the *Motherboard Express µATX Technical Reference Manual* for an example `config.txt` file.

2.5 Serial Configuration Controller

The SCC is a block of registers internal to the test chip that the *Daughterboard Configuration Controller* writes through a serial interface. The *Daughterboard Configuration Controller* uses the SCC during power-up configuration to set the test chip PLLs and other default values from the board configuration files stored on the motherboard microSD card. The MCC also uses the SCC to read values from the test chip, for example, the device ID.

You can also read and write to the SCC registers while the system is running using the motherboard SYS_CFG register interface. See *Test chip SCC registers* on page 3-11 for a full description of the SCC registers.

Figure 2-8 shows an overview of the SCC connectivity.

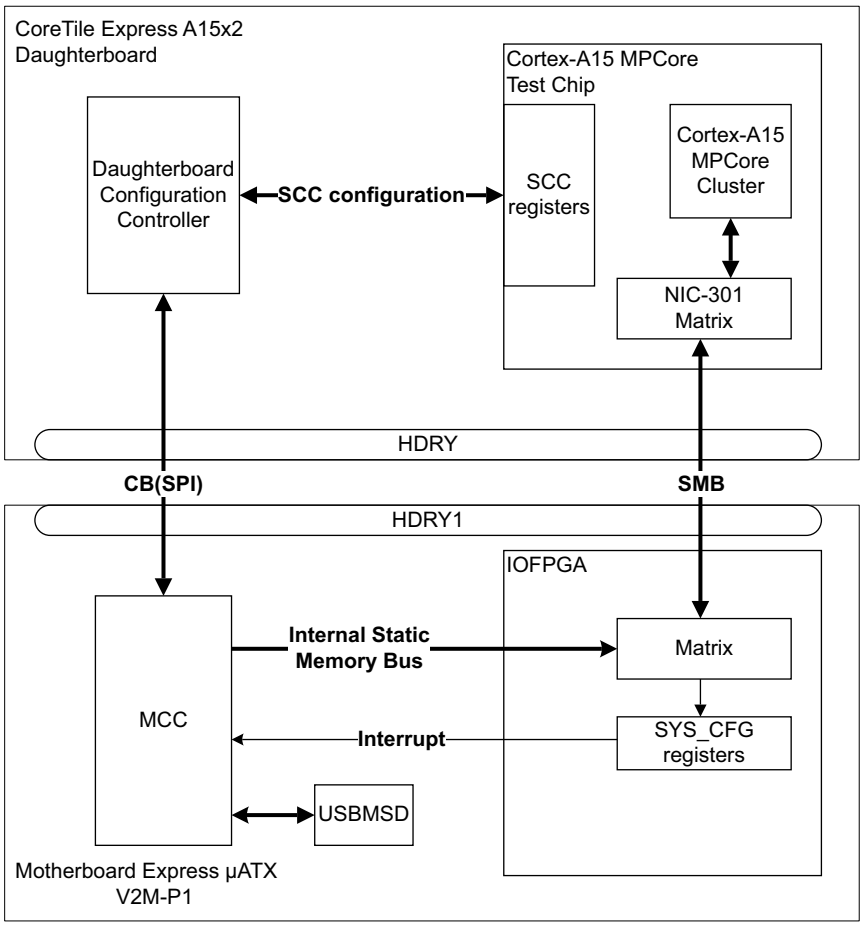


Figure 2-8 Overview of SCC connectivity

2.6 Voltage, current, power, temperature, and energy monitoring

This section describes the V2P_CA15 daughterboard CPU voltage control and current, temperature, power, and energy monitoring. It contains the following sections:

- *Voltage control and current, temperature, and power monitoring*
- *Energy meter*.

2.6.1 Voltage control and current, temperature, and power monitoring

The *Daughterboard Configuration Controller* on the daughterboard transmits voltage and current measurements for the CPU cores, and a temperature measurement for the *Daughterboard Configuration Controller* to the motherboard, where they can be read from the SYS_CFG register interface.

Table 2-2 shows the device number for monitoring and setting the CPU core voltage.

Table 2-2 Monitoring and setting the CPU cluster voltage

Device	Voltage supply	Default voltage	Description
0	V_{core}	0.9V +/-5%	Sets and reads CPU core voltage

Table 2-3 shows the device number for monitoring the CPU core current.

Table 2-3 Monitoring the CPU cluster current

Device	Voltage supply	Description
0	V_{core}	Current measurement device for the two cores, total current

Table 2-4 shows the device number for monitoring the CPU core power.

Table 2-4 Monitoring the CPU cluster power

Device	Voltage supply	Description
0	V_{core}	Power measurement device for the two cores, total power

Table 2-5 shows the device number for monitoring the *Daughterboard Configuration Controller* temperature.

Table 2-5 Monitoring the *Daughterboard Configuration Controller* temperature

Device	Description
0	Measurement device for the *Daughterboard Configuration Controller* internal operating temperature

2.6.2 Energy meter

The *Daughterboard Configuration Controller* firmware supports an on-board energy meter that enables you to measure the energy consumption of the core supply. The energy meter is a background task and does not load the main CPU software except when you request a reading through the SYS_CFG interface.

You can use the energy meter to determine the amount of energy that a specific software function consumes. You read the energy before and after the function operates and the difference between the two readings represents the amount of energy that the function consumes.

The energy meter accumulates core power supply at a rate of 10000 samples per second. You can read the accumulated energy, since reset, in Micro-Joules through the SYS_CFGCTRL register interface.

A 64-bit value represents the total energy. This requires two 32-bit reads from the SYS_CFG interface. You must read the lower 32 bits followed by the upper 32 bits. When you read the lower 32 bits, the 64-bit data transfers to two temporary storage registers in the *Daughterboard Configuration Controller*, and you then receive the lower 32 bits from the lower storage register. You then read the upper 32 bits and receive the value from the upper storage register.

If you read the upper 32 bits first, no data transfers to the storage registers and you receive a false energy reading.

The energy value is the energy since reset. You can also clear the energy value by writing `0x0000_0000` to the lower 32 bits.

Table 2-6 shows the device numbers for monitoring the energy consumption of the CPU clusters.

Table 2-6 Energy Monitoring

Device	CPU cluster energy	Description
0	J_{core}	Energy meter for the two Cortex-A15 cores, lower 32 bits, total energy
1	J_{core}	Energy meter for the two Cortex-A15 cores, upper 32 bits, total energy

See the *Motherboard Express µATX Technical Reference Manual* for more information on the SYS_CFGCTRL registers.

2.7 Clocks

This section describes the daughterboard clocks. It contains the following subsections.

- *Overview of clocks*
- *Daughterboard programmable clock generators* on page 2-22
- *Test chip PLLs and clock divider logic* on page 2-26
- *External clocks* on page 2-28.

2.7.1 Overview of clocks

The daughterboard sends and receives clocks to and from the motherboard, and also generates local clocks that are imported into the Cortex-A15 MPCore test chip. Additional PLLs inside the test chip provide phase-shifted and frequency-multiplied versions of these imported clocks as Figure 2-10 on page 2-21 shows.

Figure 2-9 on page 2-20 shows a functional overview of the CoreTile Express A15×2 daughterboard clocks and their connections to the motherboard, and a LogicTile Express FPGA daughterboard.

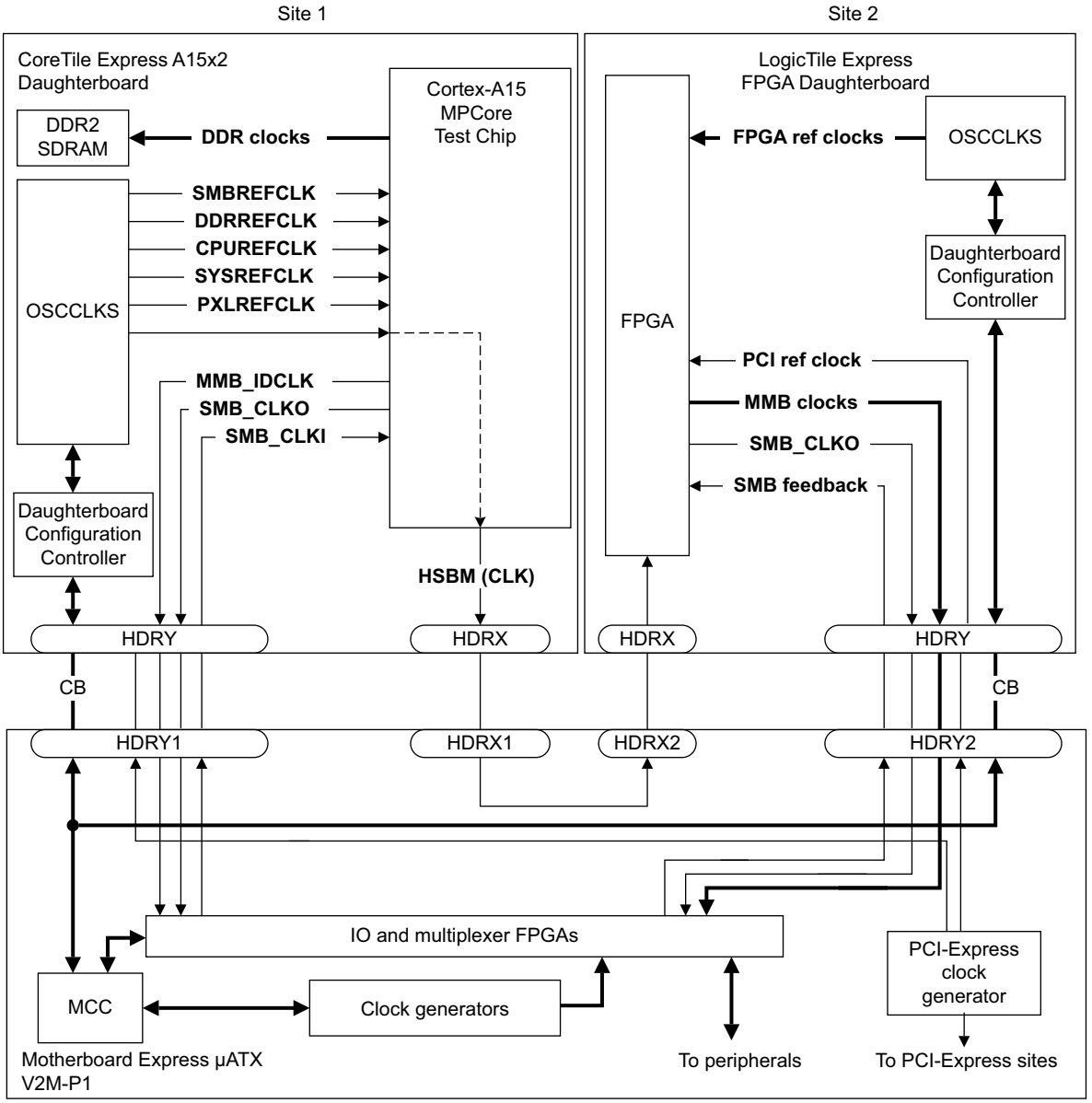


Figure 2-9 System clocks overview

Note

In the CoreTile Express A15×2 daughterboard, MMB clocks are generated from **PXLREFCLK**, that originates from OSC 5 on the daughterboard. See Figure 2-10 on page 2-21.

Figure 2-10 on page 2-21 shows a functional view of the CoreTile Express A15×2 daughterboard clocks and the internal test chip PLLs and clocks.

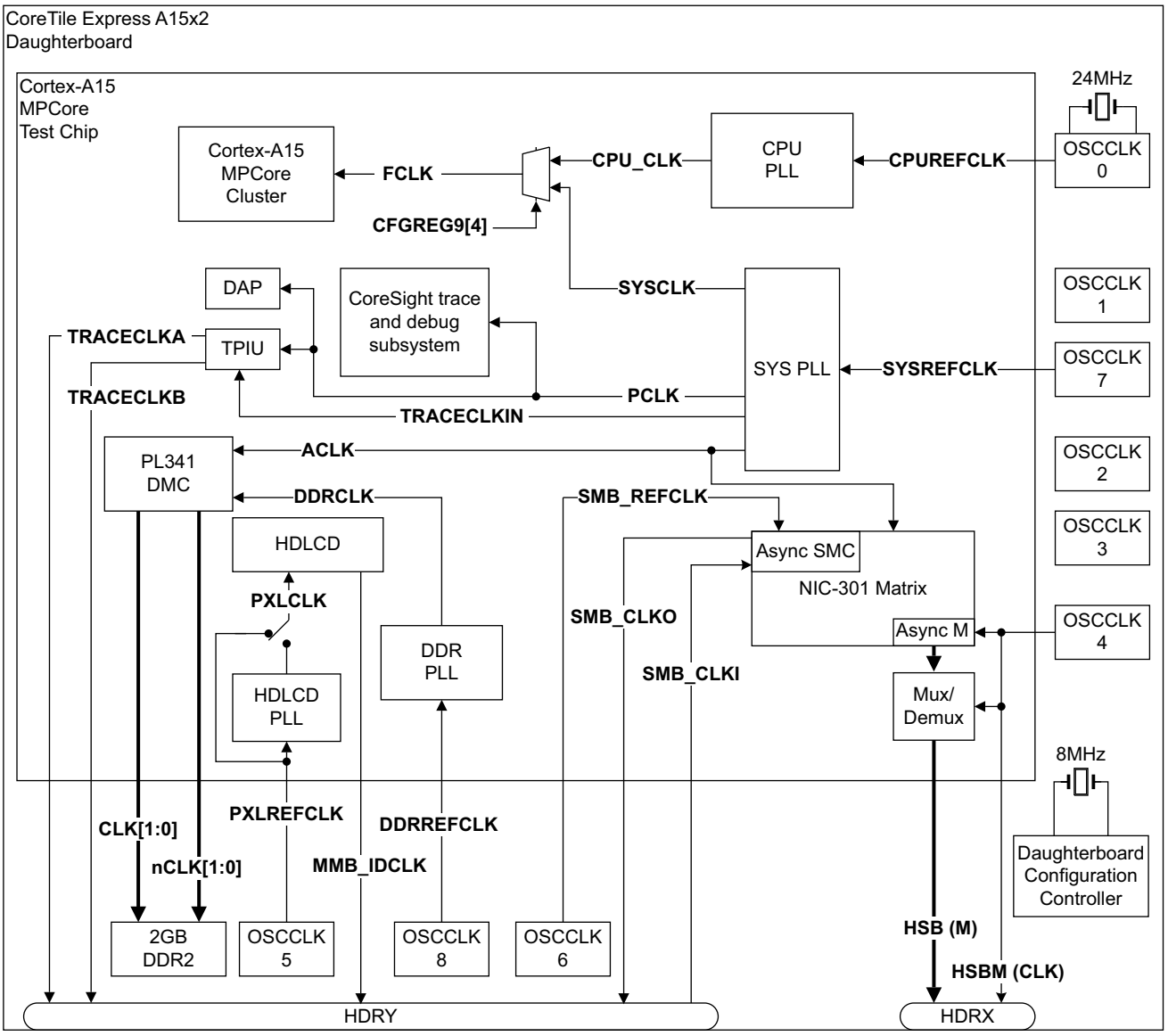


Figure 2-10 CoreTile Express A15×2 daughterboard clocks

Figure 2-10 shows the default inputs for the PLLs. You can independently select **SYSREFCLK** as the inputs to either HDLCD PLL or DDR PLL.

You can also independently select **CPUREFCLK** as the inputs to any or all of SYS PLL, DDR PLL, and HDLCD PLL.

CPU_CLK is the default source for **FCLK**.

See *Test chip SCC Register 9* on page 3-23 for information on how to select the PLL inputs and the source for **CPU_CLK**.

Note

OSCCLK 1, OSCCLK 2, and OSCLK 3 are reserved.

You can select the polarity of **MMB_IDCLK** relative to **PXLREFCLK**. It can be either in phase with **PXLREFCLK**, or inverted. See *Polarities Register bit assignments* on page B-17.

You can select **ACLK** to clock the PL341 memory interface and the DDR2 physical interface (PHY) so that the DMC and DDR2 memory operate synchronously to the AXI matrix. See *Test chip SCC Register 9* on page 3-23.

You can also use the SCC registers to exercise other options, for example, to select *External Bypass* that bypasses the PLL and drives the reference clock into the design.

Note

The configuration process bypasses the HDLCD by default.

ARM recommends that you do not select non-default options for the other PLLs. Figure 2-10 on page 2-21 and Figure 2-11 on page 2-27 do not show these options.

See:

- *Test chip SCC Register 9* on page 3-23
- *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26
- *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.

The motherboard MCC and daughterboard *Daughterboard Configuration Controller* use the `board.txt` configuration file for the daughterboard to set the frequency of the daughterboard clock generators, and to configure the SCC registers on power-up or reset. You can also use the motherboard SYS_CFG register interface to adjust the daughterboard clocks.

For more information see:

- *Power-up configuration and resets* on page 2-10.
- The *Versatile™ Express Configuration Technical Reference Manual* for an example `board.txt` file.
- The *Motherboard Express µATX Technical Reference Manual*.

2.7.2 Daughterboard programmable clock generators

This section describes the daughterboard clock generators and the clocks that the test chip generates from them to drive the on-chip systems.

The SCC registers control the PLLs, clock divider blocks, and PLL input select multiplexers. See:

- *Test chip SCC Register 9* on page 3-23
- *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26
- *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.

Table 2-7 shows the maximum clock operating frequencies.

Table 2-7 Cortex-A15 test chip maximum clock operating frequencies

Clock	Maximum operating frequency
FCLK	1200MHz
PCLK	1200MHz
TRACECLKIN	140MHz
SMB_REFCLK	50MHz
ACLK	600MHz

Table 2-7 Cortex-A15 test chip maximum clock operating frequencies (continued)

Clock	Maximum operating frequency
PXLCLK and **MMB_IDCLK**	165MHz
DDRCLK	320MHz
HSBM (CLK)	40MHz

Note

ARM derives the values in Table 2-7 on page 2-22 from a random sample of test chips operating at room temperature, and from temperature testing and RTL simulations. These are best estimate ARM can provide of the maximum operating frequencies. However, the maximum operating frequencies of the test chip on your daughterboard might be different from the frequencies in Table 2-7 on page 2-22.

Table 2-8 shows the daughterboard OSCCLK clock sources.

Table 2-8 Daughterboard OSCCLK clock sources

Test chip signal	Function	Source	OSCCLK frequency range, and default	Description, clocks derived from test chip signal
CPUREFCLK	CPU PLL reference clock	OSCCLK 0	20MHz-60MHz Default is 60MHz	Reference for **CPU_CLK** that is the default source for **FCLK**, the clock for the Cortex-A15 CPU 0 and CPU 1 cores. Default **CPU_CLK** to OSCCLK 0 ratio: 20:1.
SYSREFCLK	SYS PLL reference clock	OSCCLK 7	20MHz-60MHz Default is 60MHz	Reference for **SYSCLK** that is the alternative source for **FCLK**, the clock for the Cortex-A15 CPU 0 and CPU 1 cores. Default **SYSCLK** to OSCCLK 7 ratio: 20:1. **Note** Programmable dividers generate the following clocks from **SYSCLK**, see Figure 2-11 on page 2-27: **ACLK** The clock for the internal AXI components and the PL341 *Dynamic Memory Controller* (DMC). The default value for the programmable divider is 2. See *Test chip SCC Register 9* on page 3-23. **PCLK** The CoreSight subsystem clock. The default value for the programmable divider is 2. See *Test chip SCC Register 9* on page 3-23. **TRACECLKIN** The CoreSight *Trace Port Interface Unit* (TPIU) clock. The default value for the programmable divider is 4. See *Test chip SCC Register 9* on page 3-23. These clocks are synchronous to **SYSCLK**. If you select **SYSCLK** as the source for **FCLK**, the Cortex-A15 CPU cores operates synchronously with the NIC-301 matrix. See Figure 2-10 on page 2-21.
HSBM (CLK)	Multiplexed AXI master clock	OSCCLK 4	20MHz-40MHz Default is 40MHz	External AXI master clock, default 40MHz.

Table 2-8 Daughterboard OSCCLK clock sources (continued)

Test chip signal	Function	Source	OSCCLK frequency range, and default	Description, clocks derived from test chip signal
DDRREFCLK	DDR2 PLL reference clock	OSCCLK 8	Default is 40MHz	Reference for **DDRCLK** that is the default clock for the PL341 memory interface and DDR2 physical interface (PHY), the DDR2 pad interface. This operates asynchronously to the AXI clock, **ACLK**. See Figure 2-14 on page 2-33. Default **DDRCLK** to OSCCLK 8 ratio: 8:1. **Note** You can also select **CPUREFCLK** as the reference clock for the DDR2 PLL. See *Test chip SCC Register 9* on page 3-23. ARM recommends that you do not use this option, and Figure 2-10 on page 2-21 does not show this connection. **Note** You can select **ACLK** to clock the DDR2 pad interface. If you select this option, the DMC and DDR2 memory operate synchronously to the AXI domain. See Figure 2-14 on page 2-33 and *Test chip SCC Register 9* on page 3-23.
PXLREFCLK	HDLCD PLL reference clock	OSCCLK 5	23.75MHz-165MHz Default is 23.75MHz	The HDLCD PLL is bypassed by default. Reference for **PXLCLK**, the HDLCD controller clock in the test chip. You must adjust the frequency of this clock to match your target screen resolution. The HDLCD controller is capable of displaying up to 1920×1080p pixel resolution at 60Hz with **PXLCLK** set to 165MHz. See Appendix B *HDLCD controller*. **Note** You can also select **CPUREFCLK** as the reference clock for the HDLCD PLL. See *Test chip SCC Register 9* on page 3-23. ARM recommends that you do not use this option, and Figure 2-10 on page 2-21 does not show this connection.
SMB_REFCLK	SMB clock	OSCCLK 6	20MHz-50MHz Default is 50MHz	Static memory controller clock

Note

The clock generators have an absolute accuracy of better than 1%. If you enter a setting that cannot be precisely generated, the value is approximated to the nearest usable value.

2.7.3 Test chip PLLs and clock divider logic

This section describes the test chip PLLs and dividers that generate the internal clocks that drive the on-chip systems.

The test chip contains four *Phase Locked Loops* (PLLs) and their associated frequency dividers. They use the programmable clocks from the daughterboard to generate some of the clocks that the internal systems on the test chip use. See Figure 2-10 on page 2-21.

- CPU PLL generates **CPU_CLK**.
- SYS PLL generates **SYSCLK**.

  Note

  Programmable dividers generate the **PCLK**, **TRACECLKIN**, and **ACLK** clocks from the PLL output **SYSCLK**. See Figure 2-11 on page 2-27 and *Test chip SCC Register 9* on page 3-23.

- HDLCD PLL generates **PXLCLK**.
- DDR PLL generates the DDR2 pad interface clock **DDRCLK**.

The SCC registers control the output frequency of each PLL. Control bits from the registers set the PLL dividers to the values that achieve the required output frequency. Two SCC registers control each PLL and send:

- A 6-bit word, **CLKR**, to the input divider.
- A 13-bit word, **CLKF**, to the feedback divider.
- A 4-bit word, **CLKOD**, to the output divider.

The following equation provides the output frequency of each PLL:

$$\text{Output frequency} = \text{Input frequency} \times \frac{(\text{CLKF} + 1)}{(\text{CLKOD} + 1) \times (\text{CLKR} + 1)}$$

Figure 2-11 on page 2-27 shows the test chip PLLs and clock divider logic.

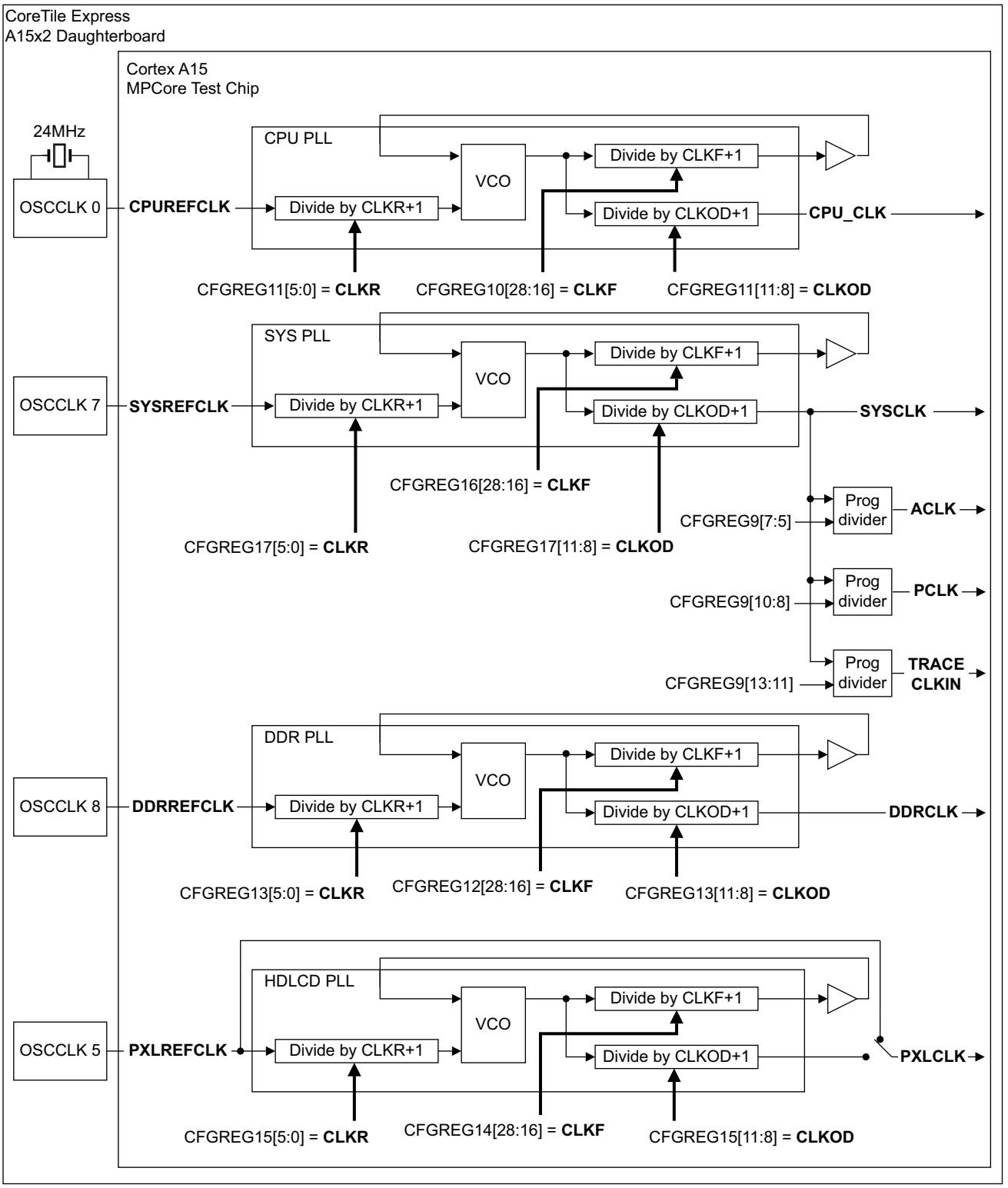


Figure 2-11 Test chip PLLs and clock divider logic

2.7.4 External clocks

Table 2-9 shows the external clocks that connect:

- Between the CoreTile Express A15×2 daughterboard and the motherboard.
- Between the CoreTile Express A15×2 daughterboard and the optional FPGA daughterboard in Site 2.

Table 2-9 External clock sources

Test chip signal	Function	Source	Frequency default, range	Description
HSBM (CLK)	-	-	-	See Table 2-8 on page 2-24
MMB_IDCLK	MMB clock	OSCCLK 5	Default is 23.7MHz	Clock for HDLCD 24-bit RGB data and synch signals
SMB_CLKI	SMC uses this to adjust for optimum timing	Motherboard	See the *Motherboard Express µATX Technical Reference Manual*	A skew-controlled version of the SMB clock that is sent from the motherboard to the SMC in the test chip
SMB_CLKO	SMC clock to motherboard	OSCCLK 6	See the *Motherboard Express µATX Technical Reference Manual*	Clock to motherboard SMB derived from **SMB_MCLK**
TRACECLKA **TRACECLKB**	Trace clocks	TPIU	140MHz maximum	Clocks to external debug trace equipment

2.8 Interrupts

This section describes the daughterboard interrupts. It contains the following subsections:

- *Overview of interrupts*
- *Test chip interrupts* on page 2-30.

2.8.1 Overview of interrupts

The Cortex-A15 MPCore test chip implements a GIC with 32 internal interrupts and 128 external interrupts.

The 32 internal interrupts connect internally between the CPUs and the GIC.

The 128 external interrupts are as follows:

- The SB connects 22 of the 128 external interrupts between the motherboard and the daughterboard.
- The test chip peripherals connect to 21 of the external interrupts.
- 85 external interrupts are reserved.

The CoreTile Express A15×2 daughterboard does not send interrupts to the second daughterboard. You can route interrupts to the CoreTile Express A15×2 daughterboard from the second daughterboard through the motherboard IOFPGA using **SB2_INT[3:0]** that loop back to **SB_IRQ**. Table 2-10 on page 2-30 shows the interrupt mapping.

Figure 2-12 shows an overview of the Cortex-A15 MPCore test chip interrupt signals.

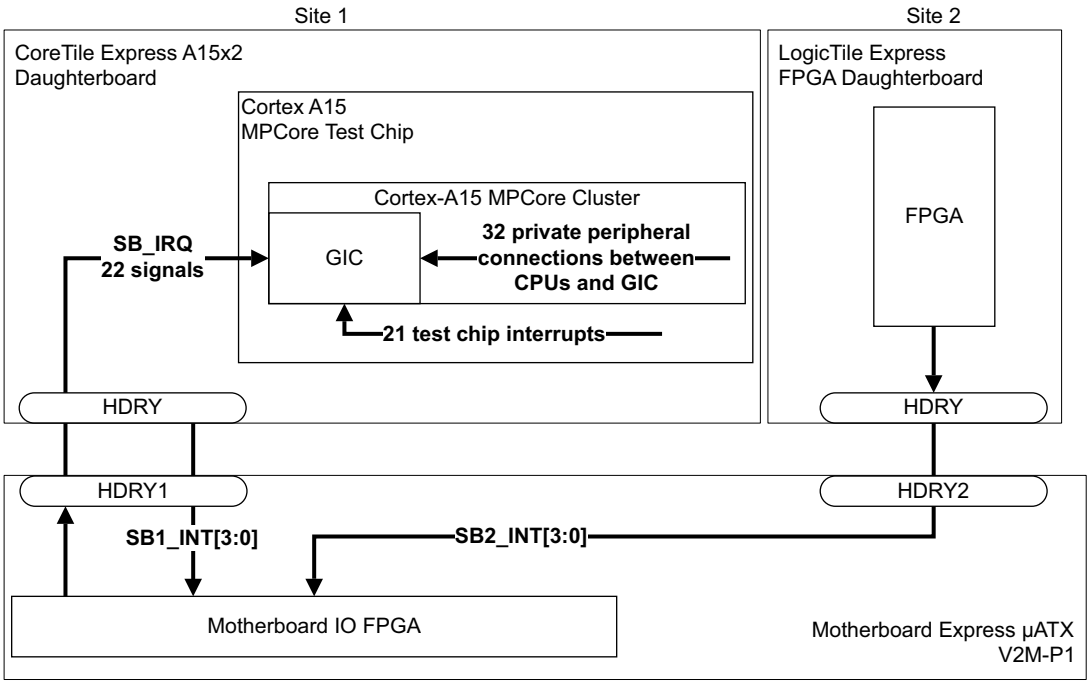


Figure 2-12 CoreTile Express A15×2 daughterboard interrupt overview

2.8.2 Test chip interrupts

Table 2-10 shows the interrupts from the motherboard IOFPGA, the interrupts from the test chip peripherals, the interrupts from the MPCore cluster, and the reserved interrupts.

Table 2-10 Test chip interrupts

GIC interrupt	SB_IRQ[] interrupt from the motherboard	Source	Signal	Description
0:31	Not applicable	MPCore cluster	-	Private peripheral connections between CPUs and GIC
32	0	IOFPGA	**WDOG0INT**	Watchdog timer
33	1	IOFPGA	**SWINT**	Software interrupt
34	2	IOFPGA	**TIM01INT**	Dual Timer 0, 1 interrupt
35	3	IOFPGA	**TIM23INT**	Dual Timer 2, 3 interrupt
36	4	IOFPGA	**RTCINTR**	Real time clock interrupt
37	5	IOFPGA	**UART0INTR**	UART0 interrupt
38	6	IOFPGA	**UART1INTR**	UART1 interrupt
39	7	IOFPGA	**UART2INTR**	UART2 interrupt
40	8	IOFPGA	**UART3INTR**	UART3 interrupt
41	9	IOFPGA	**MCI_INTR[0]**	Media card interrupt
42	10	IOFPGA	**MCI_INTR[1]**	Media card interrupt
43	11	IOFPGA	**AACI_INTR**	Audio CODEC interrupt
44	12	IOFPGA	**KMI0_INTR**	Keyboard, mouse interrupt
45	13	IOFPGA	**KMI1_INTR**	Keyboard, mouse interrupt
46	14	IOFPGA	**CLCDINTR**	Display interrupt
47	15	IOFPGA	**ETH_INTR**	Ethernet interrupt
48	16	IOFPGA	**USB_nINT**	USB interrupt
49	17	IOFPGA	**PCIE_GPEN**	PCI express
63:50	31:18	b0	-	Reserved
67:64	35:32	IOFPGA	**SB1_INT[3:0]**	Interrupts from daughterboard in site 1
71:68	39:36	IOFPGA	**SB2_INT[3:0]**	Interrupts from daughterboard in site 2
95:72	63:40	b0	-	Reserved
99:96	Not applicable	b0	-	Reserved
101:100	Not applicable	Test chip	**PMU[1:0]**	Performance monitor unit[1:0]
103:102	Not applicable	Test chip	-	Reserved
105:104	Not applicable	Test chip	**CTIIRQ[1:0]**	Cross trigger interrrupt[1:0]
107:106	Not applicable	Test chip	-	Reserved

Table 2-10 Test chip interrupts (continued)

GIC interrupt	SB_IRQ[] interrupt from the motherboard	Source	Signal	Description
109:108	Not applicable	Test chip	**COMMTX[1:0]**	CP14 DTR COMMTX Interrupt[1:0]
111:110	Not applicable	Test chip	-	Reserved
113:112	Not applicable	Test chip	**COMMRX[1:0]**	CP14 DTR COMMRX Interrupt[1:0]
115:114	Not applicable	Test chip	-	Reserved
116	Not applicable	b0	-	Reserved
117	Not applicable	Test chip	-	HDLCD interrupt
119:118	Not applicable	Test chip	**SMC[1:0]**	SMC interface interrupt[1:0]
123:120	Not applicable	Test chip	**DMA_IRQ[3:0]**	DMA interrupts[3:0]
124	Not applicable	Test chip	**DMA_IRQ_ABORT**	-
127:125	Not applicable	b0	-	Reserved
128	Not applicable	Test chip	-	AXI error
129	Not applicable	Test chip	-	RAM double bit error
130	Not applicable	Test chip	-	System watchdog interrupt
159:131	Not applicable	b0	-	Reserved

Note

- For more information on the motherboard peripherals that generate interrupts to the test chip, see the *Motherboard Express µATX Technical Reference Manual*.

2.9 HDLCD

An ARM HDLCD controller in the Cortex-A15 MPCore test chip provides graphic display capabilities. The controller is a frame buffer device that is capable of displaying up to 1920×1080p pixel resolution at 60Hz with a 165MHz pixel clock from OSCCLK 5.

The MMB connects the 24-bit RGB data directly between the test chip and the motherboard through the HDRY header. The multiplexer FPGA on the motherboard can select this bus to drive the analog and digital interfaces for the DVI connector using the motherboard SYS_CFG register interface. See the *Motherboard Express µATX Technical Reference Manual*.

The HDLCD frame buffer is located in DDR2 memory serviced by the DMC from the test chip bus matrix. This ensures maximum data bandwidth between the Cortex-A15 MPCore cluster, the HDLCD controller, and DDR2 memory without accessing off-chip devices.

Figure 2-13 shows a functional overview of the HDLCD controller and its connections to the Cortex-A15 test chip and the motherboard.

See Appendix B *HDLCD controller* for a full description of the HDLCD controller.

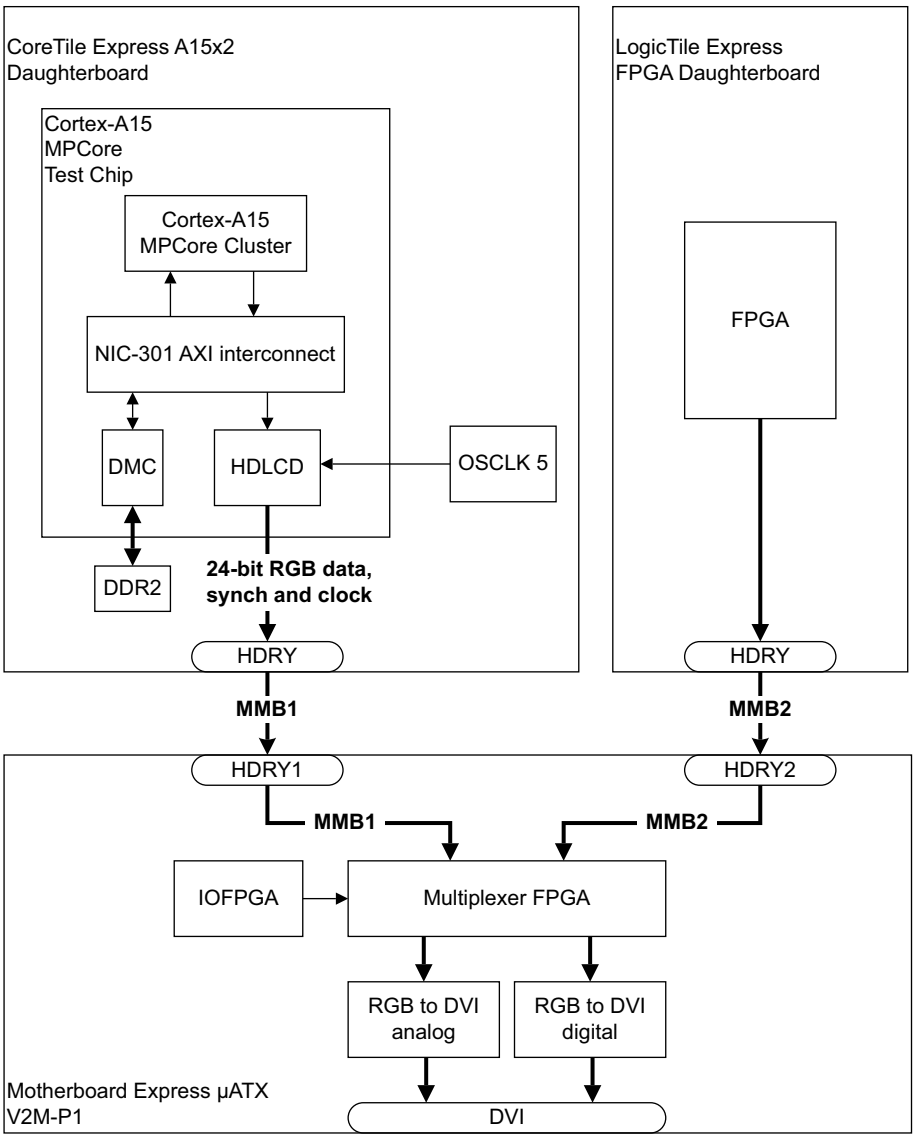


Figure 2-13 HDLCD graphics system interconnect

2.10 DDR2 memory interface

The Cortex-A15 DDR2 memory interface uses a PL341 *Dynamic Memory Controller* (DMC). By default, the DMC runs asynchronously to the AXI matrix so the AXI sub-system does not impose frequency limitations on the DMC interface.

You can select **ACLK** to clock the PL341 memory interface and the DDR2 physical interface, PHY, so that the DMC and DDR2 memory operate synchronously to the AXI matrix. See *Test chip SCC Register 9* on page 3-23.

Figure 2-14 shows a functional overview of the DDR2 memory interface.

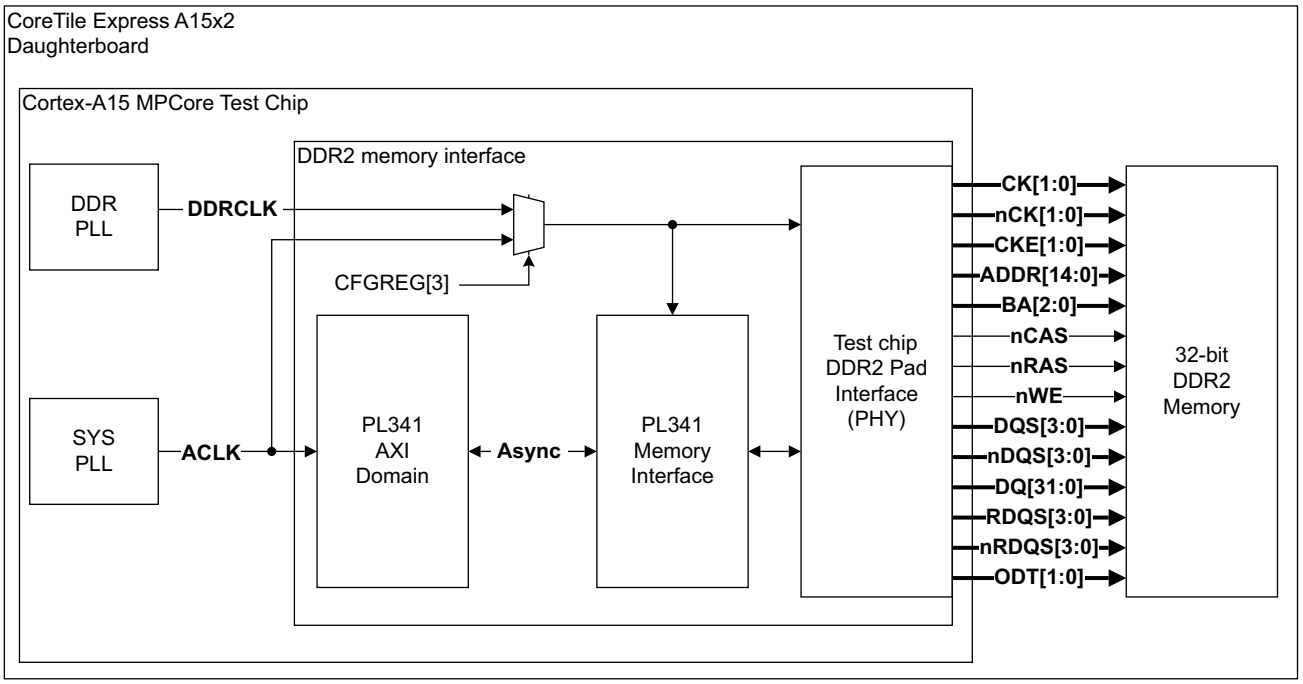


Figure 2-14 DDR2 memory interface

Note

OSCCLK 8 is the source for **DDRCLK**, and OSCCLK 7 is the source for **ACLK**. See Figure 2-10 on page 2-21 and Table 2-8 on page 2-24.

2.11 Debug

You can attach a JTAG debugger to the daughterboard JTAG connector to execute programs to the daughterboard and debug them. For convenience, connect the cable from the rear panel JTAG connector to the daughterboard JTAG. For example, you can connect a software debugger, such as the DS-5 debugger, to this debug interface using an external DSTREAM debug and trace unit.

Note

The daughterboard does not support adaptive clocking. The **RTCK** signal is tied LOW on the JTAG ICE connector.

See Figure 1-1 on page 1-3 for the location of the JTAG ICE connector.

Figure 2-15 shows an overview of the CoreSight system.

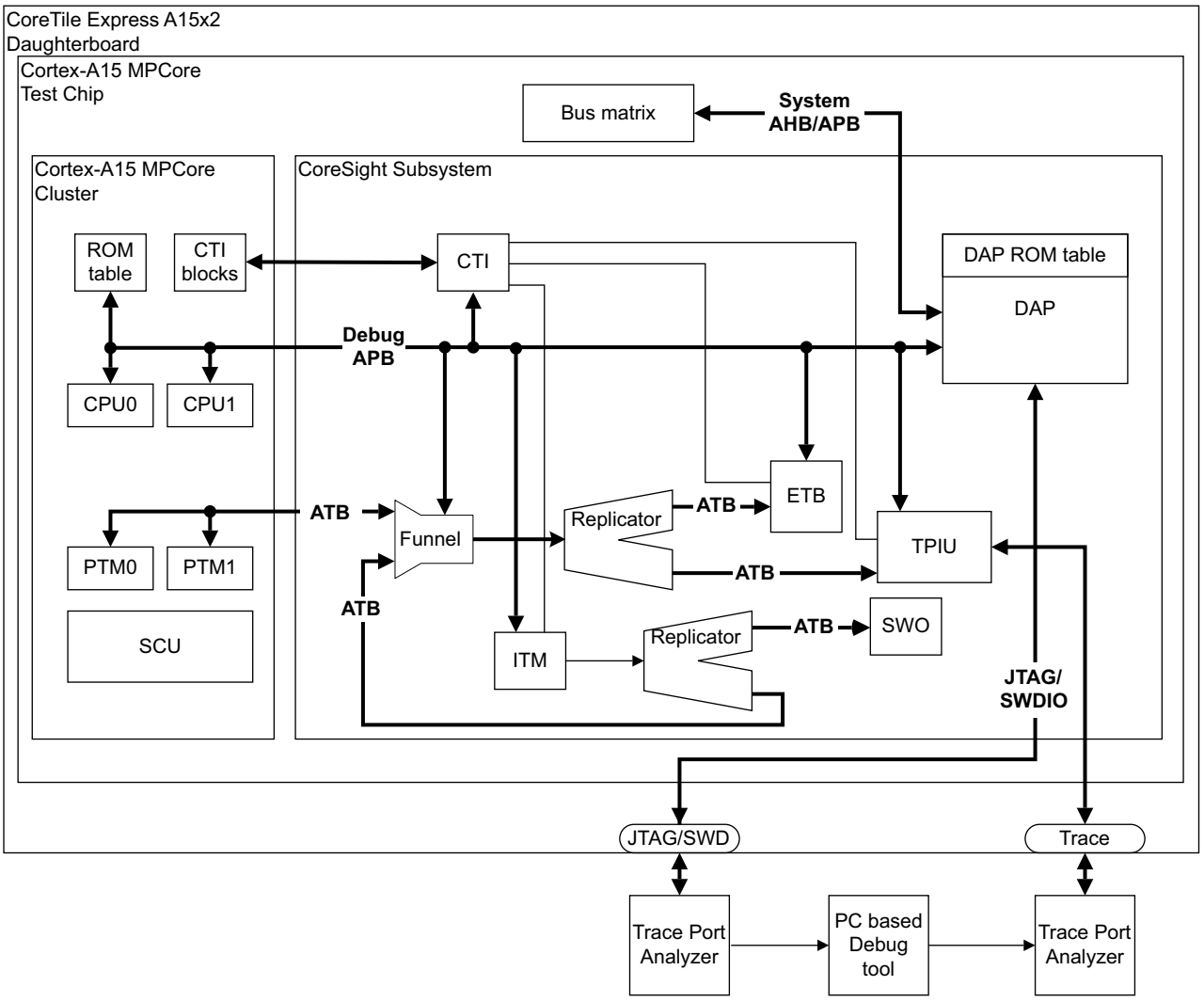


Figure 2-15 CoreTile Express A15×2 CoreSight and trace

For information on CoreSight components, see the *CoreSight Components Technical Reference Manual*.

The daughterboard supports up to 32-bit trace in *continuous* mode. There are two MICTOR connectors for JTAG and trace. Figure 1-1 on page 1-3 shows the location of these connectors.

To set up a trace connection to either of the cores on the test chip, you must know the funnel port number and PTM base address connection information associated with the cores. Table 2-11 defines the funnel port numbers and base addresses.

Table 2-11 Test chip Trace connection addresses

Core	Core base address	Funnel port	PTM base address
Core 0	0x00_2003_0000	0	0x00_2003_C000
Core 1	0x00_2003_2000	1	0x0x_2003_D000

See Figure 3-3 on page 3-9, Table 3-4 on page 3-9, and *Daughterboard memory map* on page 3-3.

Chapter 3
Programmers Model

This chapter describes the memory map and the configuration registers for the peripherals on the daughterboard. It contains the following sections:

3.1 About this programmers model

The following information applies to the SCC and to the system counter registers:

- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in UNPREDICTABLE behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - All register bits are reset to a logic 0 by a system or power-on reset.
  - Access type in Table 3-5 on page 3-12 is described as follows:

    RW Read and write.

    RO Read only.

    WO Write only.

3.2 Daughterboard memory map

The Cortex-A15 test chip supports the 40-bit *Large Physical Address Extension* (LPAE). The programmers model is based on the *Cortex-A Series* memory map with support for 2GB of contiguous DDR memory that is located at `0x00_8000_0000`.

The test chip SMC is located at `0x00_0000_0000` and supports up to six chip selects. The test chip internal peripherals are located at `0x00_2000_0000`, Cortex-A15 ACP at `0x00_3000_0000`, and external AXI at `0x00_4000_0000`.

The memory map supports a remap option at `0x00_0000_0000` that can select SMC or external AXI.

A typical system boots from SMC CS0 that addresses the motherboard NOR flash 0. See *Remapping memory* on page 3-5.

3.2.1 Overview of daughterboard memory map

Figure 3-1 on page 3-4 shows the daughterboard memory map. SCC register *Test chip SCC Register 0* on page 3-15 controls the remap option. See *Remapping memory* on page 3-5.

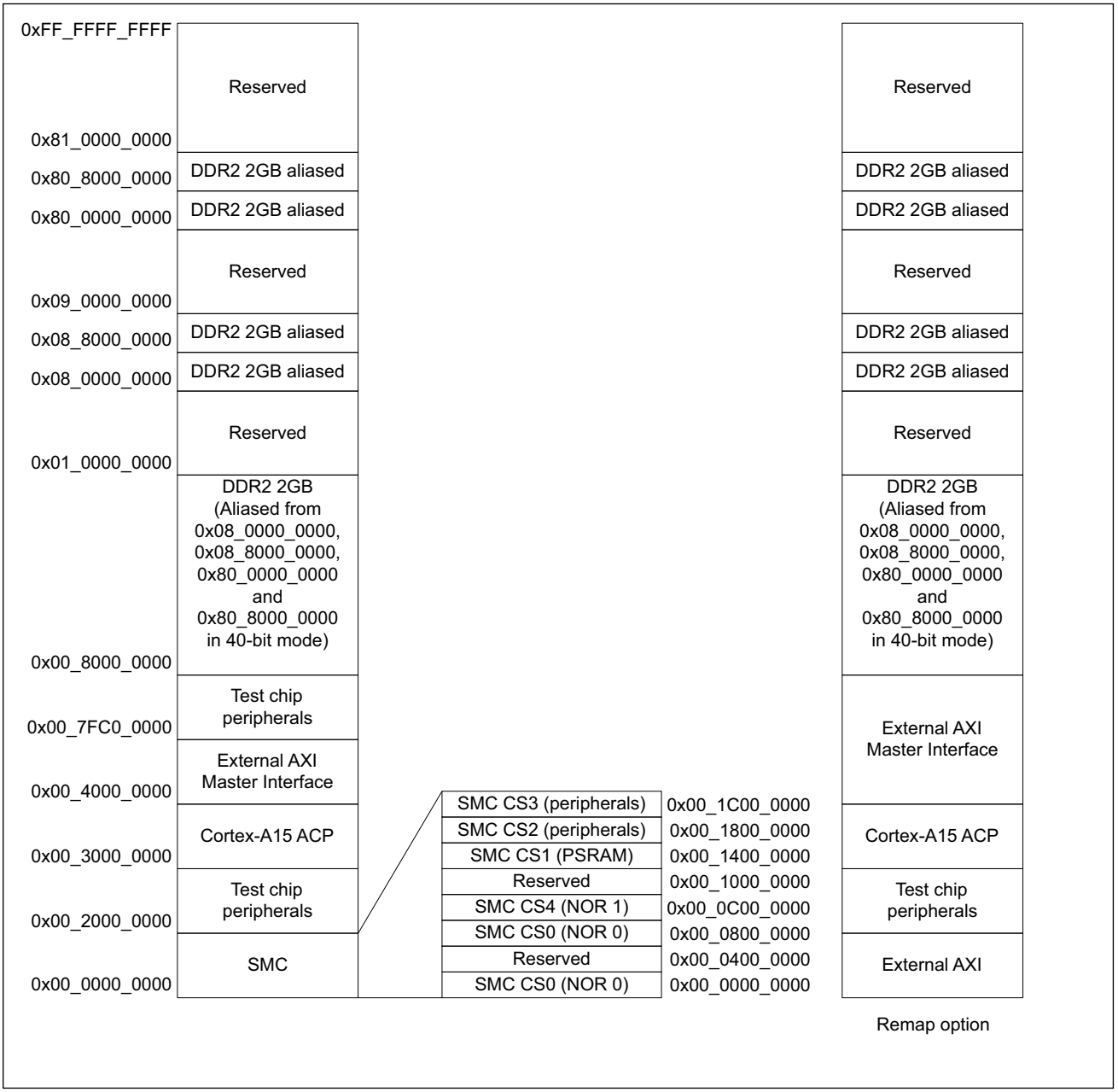


Figure 3-1 CoreTile Express A15×2 daughterboard memory map

Table 3-1 shows the daughterboard peripheral interfaces.

Table 3-1 Daughterboard memory map

Address range	Size	Description
0x00_0000_0000 - 0x00_03FF_FFFF	64MB	CS0-Motherboard NOR flash 0
0x00_0400_0000 - 0x00_07FF_FFFF	64MB	Reserved
0x00_0800_0000 - 0x00_0BFF_FFFF	64MB	CS0-Motherboard NOR flash 0

Table 3-1 Daughterboard memory map (continued)

Address range	Size	Description
0x00_0C00_0000 - 0x00_0FFF_FFFF	64MB	CS4-Motherboard NOR flash 1
0x00_1000_0000 - 0x00_13FF_FFFF	64MB	CS5-Reserved
0x00_1400_0000 - 0x00_17FF_FFFF	64MB	CS1-Pseudostatic RAM, PSRAM, on the motherboard
0x00_1800_0000 - 0x00_1BFF_FFFF	64MB	CS2-Video/ETH/USB on the motherboard
0x00_1C00_0000 - 0x00_1FFF_FFFF	64MB	CS3-system registers and peripherals on the motherboard
0x00_0000_0000 - 0x00_1FFF_FFFF	512MB	External AXI master interface, remap option
0x00_2000_0000 - 0x00_2FFF_FFFF	256MB	Test chip peripherals (some areas Reserved). See *Overview of the memory map for the on-chip peripherals* on page 3-6.
0x00_3000_0000 - 0x00_3FFF_FFFF	256MB	Cortex-A15 *Accelerator Coherency Port* (ACP)
0x00_4000_0000 - 0x00_7FBF_FFFF	1020MB	External AXI between daughterboards
0x00_7FC0_0000 - 0x00_7FFF_FFFF	4MB	Test chip peripherals (some areas Reserved). See *Overview of the memory map for the on-chip peripherals* on page 3-6.
0x00_8000_0000 - 0x00_FFFF_FFFF	2GB	CoreTile Express A15×2 daughterboard DDR2, block 1
0x01_0000_0000 - 0x07_FFFF_FFFF	28GB	Reserved
0x08_0000_0000 - x08_7FFF_FFFF	2GB	CoreTile Express A15×2 daughterboard DDR2, alias to block 1
0x08_8000_0000 - 0x08_FFFF_FFFF	2GB	CoreTile Express A15×2 daughterboard DDR2, alias to block 1
0x09_0000_0000 - 0x7F_FFFF_FFFF	476GB	Reserved
0x80_0000_0000 - 0x80_7FFF_FFFF	2GB	CoreTile Express A15×2 daughterboard DDR2, alias to block 1
0x80_8000_0000 - 0x80_FFFF_FFFF	2GB	CoreTile Express A15×2 daughterboard DDR2, alias to block 1
0x81_0000_0000 - 0xFF_FFFF_FFFF	508GB	Reserved

3.2.2 Remapping memory

This section describes the remap options for address 0x00_0000_0000. It contains the following subsection:

- *SMC or AXI*
- *Chip-select Remap*
- *Memory remapping at power-on during run-time* on page 3-6.

SMC or AXI

The SCC registers control whether AXI or SMC is mapped to the lower 512MB of memory. This enables booting from external AXI. See *Test chip SCC Register 0* on page 3-15.

Chip-select Remap

SCC register bits CFGREG0[31:30] control whether CS0 or CS4 is addressed at address 0x00_0000_0000. See *Test chip SCC Register 0* on page 3-15.

Note

Chip-select remap operates only when you map SMC to 0x00_0000_0000.

The processor fetches its first instructions from address `0x00_0000_0000`, but the actual memory read depends on the remapped memory region.

Memory remapping at power-on during run-time

You can configure the remap option at power-on or during run time.

Remapping at power-on

Use the `board.txt` file if the remap option is required when the processor starts from a reset. The `SCC: 0x000` entry in the `board.txt` file controls the settings for the *Test chip SCC Register 0* on page 3-15. See the *Versatile™ Express Configuration Technical Reference Manual*.

Remapping during run time

Write directly to the SCC register CFGREG0 to change remapping after power-on. See *Test chip SCC Register 0* on page 3-15. See the Boot Monitor `sys_boot.s` file for an example of reconfiguring while running.

Caution

ARM recommends that you use the configuration file rather than directly writing to the control registers. You must perform remapping during run-time with care. Do not do this when code is running from the remapped area.

3.2.3 Overview of the memory map for the on-chip peripherals

Figure 3-2 on page 3-7 shows the on-chip peripheral memory map.

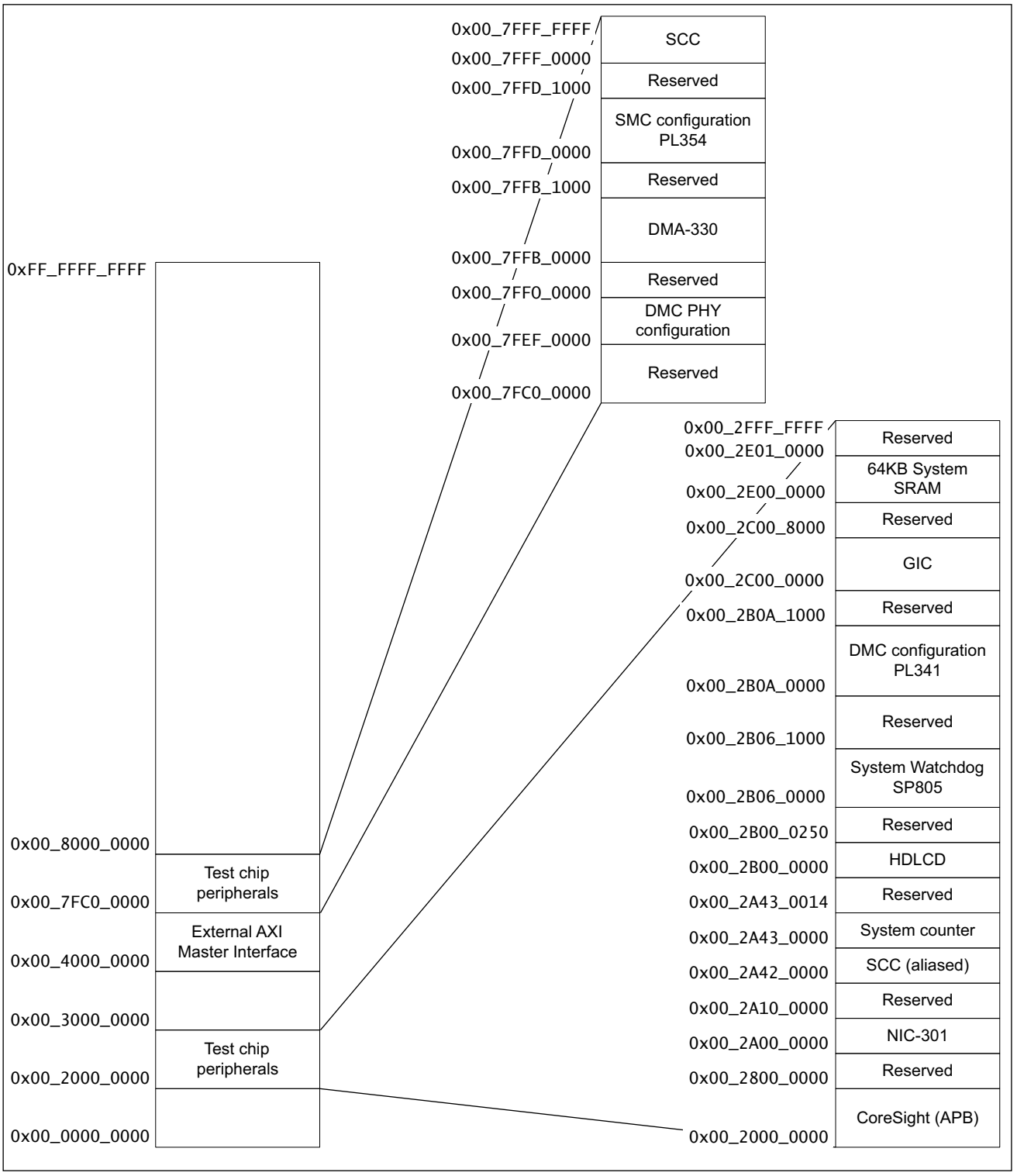


Figure 3-2 Cortex-A15 MPCore on-chip peripheral memory map

Table 3-2 and Table 3-3 show the areas of the Cortex-A15 MPCore memory map occupied by test chip peripherals.

Table 3-2 Cortex-A15 MPCore on-chip peripheral memory map

Address range	Size	Description
0x00_2000_0000 - 0x00_27FF_FFFF	128MB	CoreSight components. See Table 3-4 on page 3-9
0x00_2800_0000 - 0x00_29FF_FFFF	32MB	Reserved
0x00_2A00_0000 - 0x00_2A0F_FFFF	1MB	AXI Network Interconnect, NIC-301
0x00_2A10_0000 - 0x00_2A41_FFFF	3.125MB	Reserved
0x00_2A42_0000 - 0x00_2A42_FFFF	64KB	SCC, aliased
0x00_2A43_0000 - 0x00_2A43_0013	20B	System counter
0x00_2A43_0014 - 0x00_2AFF_FFFF	12386284B	Reserved
0x00_2B00_0000 - 0x00_2B00_024F	592B	HDLCD
0x00_2B00_0250 - 0x00_2B05_FFFF	392624B	Reserved
0x00_2B06_0000 - 0x00_2B06_0FFF	4KB	System Watchdog, SP805
0x00_2B06_1000 - 0x00_2B09_FFFF	252KB	Reserved
0x00_2BOA_0000 - 0x00_2B0A_0FFF	4KB	DMC configuration, PL341
0x00_2B0A_1000 - 0x00_2BFF_FFFF	15740KB	Reserved
0x00_2C00_0000 - 0x00_2C00_7FFF	32KB	*Generic Interrupt Controller*, GIC
0x00_2C00_8000 - 0x00_2DFF_FFFF	32736KB	Reserved
0x00_2E00_0000 - 0x00_2E00_FFFF	64KB	64KB System SRAM
0x00_2E01_0000 - 0x00_2FFF_FFFF	32704KB	Reserved

Table 3-3 Cortex-A15 MPCore on-chip peripheral memory map

Address range	Size	Description
0x00_7FC0_0000 - 0x00_7FEE_FFFF	3008KB	Reserved
0x00_7FEF_0000 - 0x00_7FEF_FFFF	64KB	DMC PHY configuration
0x00_7FF0_0000 - 0x00_7FFA_FFFF	704KB	Reserved
0x00_7FFB_0000 - 0x00_7FFB_0FFF	4KB	*Direct Memory Access* controller, DMA-330
0x00_7FFB_1000 - 0x00_7FFC_FFFF	124KB	Reserved
0x00_7FFD_0000 - 0x00_7FFD_0FFF	4KB	Static Memory Controller (SMC), PL354
0x00_7FFD_1000 - 0x00_7FFE_FFFF	124KB	Reserved
0x00_7FFF_0000 - 0x00_7FFF_FFFF	64KB	SCC

Figure 3-3 on page 3-9 shows the on-chip CoreSight component memory map.

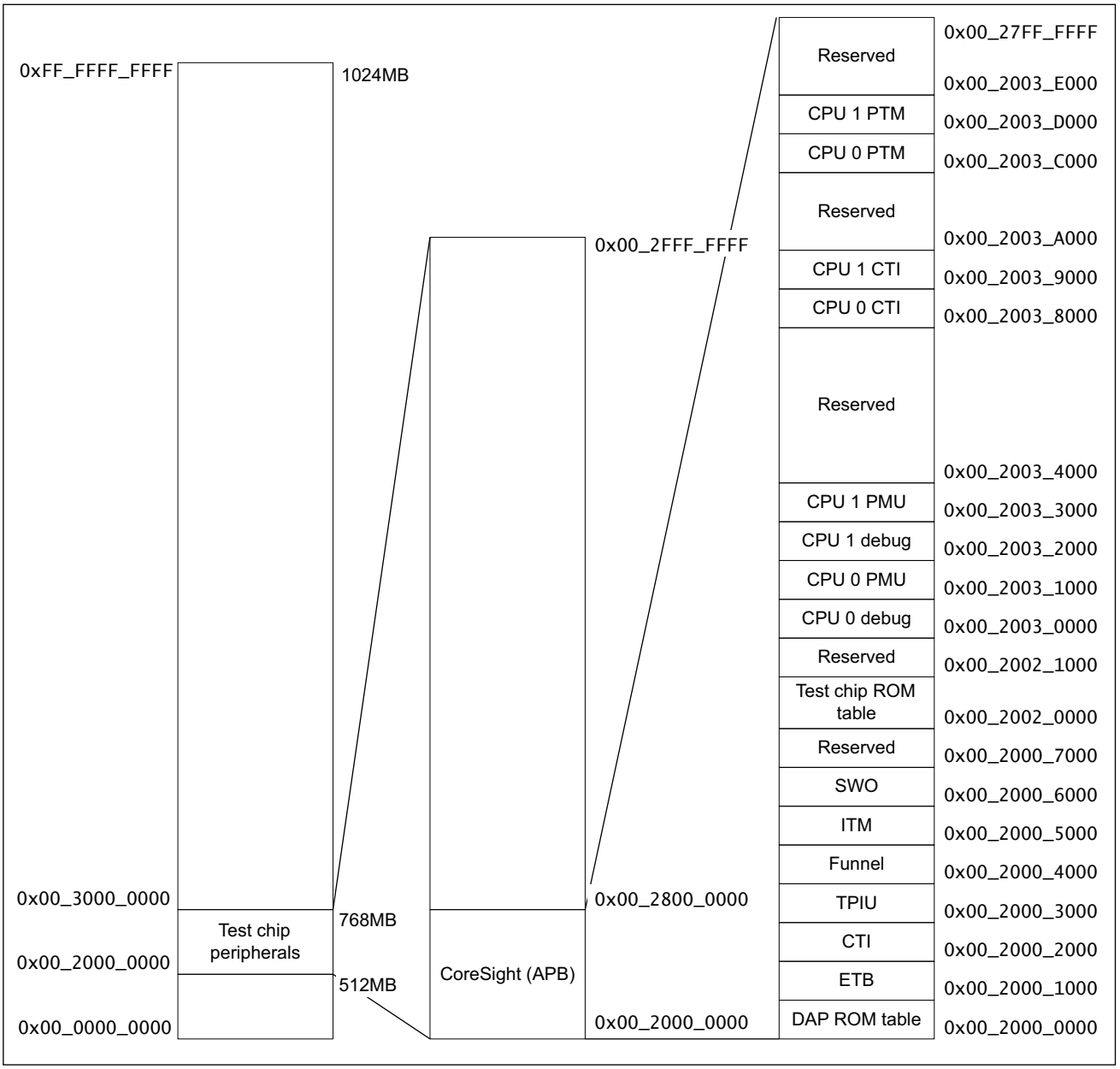


Figure 3-3 Cortex-A15 MPCore CoreSight component memory map

Table 3-4 shows the CoreSight component memory map.

Table 3-4 CoreSight component memory map

Address range	Size	Description
0x00_2000_0000 - 0x00_2000_0FFF	4KB	DAP ROM table
0x00_2000_1000 - 0x00_2000_1FFF	4KB	ETB
0x00_2000_2000 - 0x00_2000_2FFF	4KB	CTI
0x00_2000_3000 - 0x00_2000_3FFF	4KB	TPIU
0x00_2000_4000 - 0x00_2000_4FFF	4KB	Funnel

Table 3-4 CoreSight component memory map (continued)

Address range	Size	Description
0x00_2000_5000 - 0x00_2000_5FFF	4KB	ITM
0x00_2000_6000 - 0x00_2000_6FFF	4KB	SWO
0x00_2000_7000 - 0x00_2001_FFFF	100KB	Reserved
0x00_2002_0000 - 0x00_2002_0FFF	4KB	Test chip ROM table
0x00_2002_1000 - 0x00_2002_FFFF	60KB	Reserved
0x00_2003_0000 - 0x00_2003_0FFF	4KB	CPU 0 debug
0x00_2003_1000 - 0x00_2003_1FFF	4KB	CPU 0 PMU
0x00_2003_2000 - 0x00_2003_2FFF	4KB	CPU 1 debug
0x00_2003_3000 - 0x00_2003_3FFF	4KB	CPU 1 PMU
0x00_2003_4000 - 0x00_2003_7FFF	16KB	Reserved
0x00_2003_8000 - 0x00_2003_8FFF	4KB	CPU 0 CTI
0x00_2003_9000 - 0x00_2003_9FFF	4KB	CPU 1 CTI
0x00_2003_A000 - 0x00_2003_BFFF	8KB	Reserved
0x00_2003_C000 - 0x00_2003_CFFF	4KB	CPU 0 PTM
0x00_2003_D000 - 0x00_2003_DFFF	4KB	CPU 1 PTM
0x00_2003_E000 - 0x00_27FF_FFFF	130824KB	Reserved

3.3 Test chip SCC registers

This section describes the SCC registers. It contains the following sections:

- *Test chip SCC register overview*
- *Test chip SCC register summary* on page 3-12
- *Test chip SCC register descriptions* on page 3-15.

3.3.1 Test chip SCC register overview

The SCC register interface enables configuration at power-up in addition to the reading and writing of system parameters. The SCC registers use a dedicated interface to the *Daughterboard Configuration Controller* that communicates with the MCC over an SPI interface.

The MCC on the motherboard reads the `config.txt` and `board.txt` configuration files and uses the *Daughterboard Configuration Controller* to configure the motherboard and attached daughterboards. The *Daughterboard Configuration Controller* loads some of the registers in the test chip SCC.

Run-time read and write operations are performed either through the SYS_CFG registers, or directly through the APB interface at base address `0x00_7FFF_0000`.

Note

ARM recommends that, where possible, you perform all system configuration by loading configuration files into the microSD card on the motherboard rather than writing directly to the test chip controller. The settings in the `board.txt` file are applied to the daughterboard before reset is released.

You can read and write to the Cortex-A15 MPCore test chip SCC registers:

- The interface supports word writes to the configuration controller registers.
- Writes to read-only registers are ignored.
- Writes to unused words fail.
- ARM recommends that you use a read-modify-write sequence to update the configuration controller registers.
- Read accesses to the peripheral support reading back 32 bits of the register at a time.
- Reads from unused words in the register return zero.
- The base address of the SCC registers is `0x00_7FFF_0000`.

3.3.2 Test chip SCC register summary

Table 3-5 shows the configuration registers and corresponding offsets from the base memory address. The offsets are also their `board.txt` entries. The configuration process, see *Versatile™ Express Configuration Technical Reference Manual*, overwrites the test chip reset or default values in Table 3-5.

Table 3-5 Test chip SCC register summary

Offset	Name	Type	Test chip reset	Width	Description
0x000	CFGREG0	RW	0x400F0008	32	Miscellaneous configuration register 0. See *Test chip SCC Register 0* on page 3-15.
0x004	CFGREG1	RW	0x03FFC3FF	32	Miscellaneous configuration register 1. See *Test chip SCC Register 1* on page 3-16.
0x010	CFGREG2	RW	0x14FC00FC	32	SMC CS0/1. See *Test chip SCC Register 2* on page 3-19.
0x014	CFGREG3	RW	0x1CFC18FC	32	SMC CS2/3. See *Test chip SCC Register 3* on page 3-19.
0x018	CFGREG4	RW	0x10FC0CFC	32	SMC CS4/5. See *Test chip SCC Register 4* on page 3-20.
0x01C	CFGREG5	RW	0x06FC04FC	32	SMC CS6/7. See *Test chip SCC Register 5* on page 3-21.
0x020 - 0X03C	-	-	-	-	Reserved. Do not write to or read from these registers.
0x040	CUSTOMER_ID	RO	0x10041002	32	Revision, designer ID, and part number information register. See *Test chip SCC CUSTOMER_ID Register* on page 3-21.
0x044-0X18C	Reserved	-	-	-	Reserved. Do not write to or read from these registers.
0x190	CFGREG6	RO	0x07230477	32	DAP target ID, DAP ROM default target ID. See *Test chip SCC Register 6* on page 3-22.
0X194	CFGREG7	RO	0x00000000	32	DAP instance ID, DAP ROM default instance ID. See *Test chip SCC Register 7* on page 3-23.
0x198	Reserved	-	-	-	Reserved. Do not write to or read from this register.
0X19C	CFGREG9	RW	0xFF201920	32	Clock control register. See *Test chip SCC Register 9* on page 3-23.
0X1A0	CFGREG10	RW	0x022F1000	32	CPU PLL settings register 0. See *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26.
0X1A4	CFGREG11	RW	0x0011710D	32	CPU PLL settings register 1. See *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.
0X1A8	CFGREG12	RW	0x00BF1000	32	DDR PLL settings register 0. See *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26.

Table 3-5 Test chip SCC register summary (continued)

Offset	Name	Type	Test chip reset	Width	Description
0X1AC	CFGREG13	RW	0x00005F503	32	DDR PLL settings register 1. See *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.
0X1B0	CFGREG14	RW	0x01CD1000	32	HDLCD PLL settings register 0. See *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26.
0X1B4	CFGREG15	RW	0x000E6D09	32	HDLCD PLL settings register 1. See *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.
0X1B8	CFGREG16	RW	0x022F1000	32	SYS PLL settings register 0. See *Test chip SCC Registers 10, 12, 14 and 16* on page 3-26.
0X1BC	CFGREG17	RW	0x0011710D	32	SYS PLL settings register 1. See *Test chip SCC Registers 11, 13, 15 and 17* on page 3-28.
0x1C0-0x1CC	-	-		-	Reserved. Do not write to or read from these registers.
0X1D0	CFGREG22	RW	0x00000000	32	DMAC boot settings register 0. See *Test chip SCC Register 22* on page 3-30.
0X1D4	CFGREG23	RW	0x00001000	32	DMAC boot settings register 1. *Test chip SCC Register 23* on page 3-32.
0X1D8-0xFF0	-	-	-	-	Reserved. Do not write to or read from these registers.
0XFF4	APB_CLEAR	WO	-	32	Write 0xA50FF0FA to this register to revert serial control to the values loaded through the SCC. See *Test chip SCC Register APB_CLEAR* on page 3-32.
0XFF8	TC_ID	RO	0x00050176	32	Test chip-specific device ID register. See *Test chip SCC Register TC_ID* on page 3-33.
0XFFC	CPU_ID	RO	0x410FC0F0	32	Cortex-A15 CPU ID register. See *Test chip SCC Register CPU_ID* on page 3-33.

3.3.3 Mask operation to define SMC chip select address ranges

The following enable you to define SMC *Chip Select* (CS) base addresses by defining the 7th and 8th hexadecimal digits of each base address:

- *Test chip SCC Register 2* on page 3-19
- *Test chip SCC Register 3* on page 3-19
- *Test chip SCC Register 4* on page 3-20
- *Test chip SCC Register 5* on page 3-21.

Each register defines two base addresses, and each base address is defined by 8 *match* bits and by 8 *mask* bits.

In each register:

- Bits[31:24] and bits[15:8] are the *match* bits.
- Bits[23:16] and bits[7:0] are the *mask* bits.

The 8 match bits modified by the 8 *mask* bits represent the 7th and 8th digits of the SMC base address:

`b0` in *mask* register

Masked. Corresponding bit in *match* register is *Don't Care*, X.

`b1` in *mask* register

Not masked. Corresponding bit in *match* register is unchanged.

Example 3-1 Defining SMC CS4 base address

The match bits of SMC CS4 base address are `b00001100`, and the mask bits are `b11111100`. See *Test chip SCC Register 4* on page 3-20. The result of the mask operation is `b000011XX`.

This means that the 8th and 7th digits of the base address of SMC CS4 have the value `0x0C`. Test chip circuitry generates the address range of SMC CS4 as `0x00_0C00_0000` to `0x00_0FFF_FFFF` that is, a range of 64MB. See Figure 3-1 on page 3-4 and Table 3-1 on page 3-4.

Example 3-2 Defining SMC CS0 base address

The match bits of SMC CS0 base address are `b00000000`, and the mask bits are `b11110100`. See *Test chip SCC Register 2* on page 3-19. The result of the mask operation is `b0000X0XX`.

This means that the 8th and 7th digits of the base address of SMC CS0 have two values that are `0x00` and `0x08`. Test chip circuitry generates the address ranges of SMC CS0 as `0x00_0000_0000` to `0x00_03FF_FFFF`, and `0x00_0800_0000` to `0x00_0BFF_FFFF`. Each range is 64MB. See Figure 3-1 on page 3-4 and Table 3-1 on page 3-4.

3.3.4 Test chip SCC register descriptions

This section describes the SCC registers.

Test chip SCC Register 0

The CFGREG0 Register characteristics are:

Purpose Miscellaneous Configuration Register 0 that enables you to read and write test chip configuration settings.

Usage constraints CFGREG0[31:30] = `00` or `11` are reserved and must not be used. These bits are valid only if CFGREG0[0] is `b0`.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-4 shows the bit assignments.

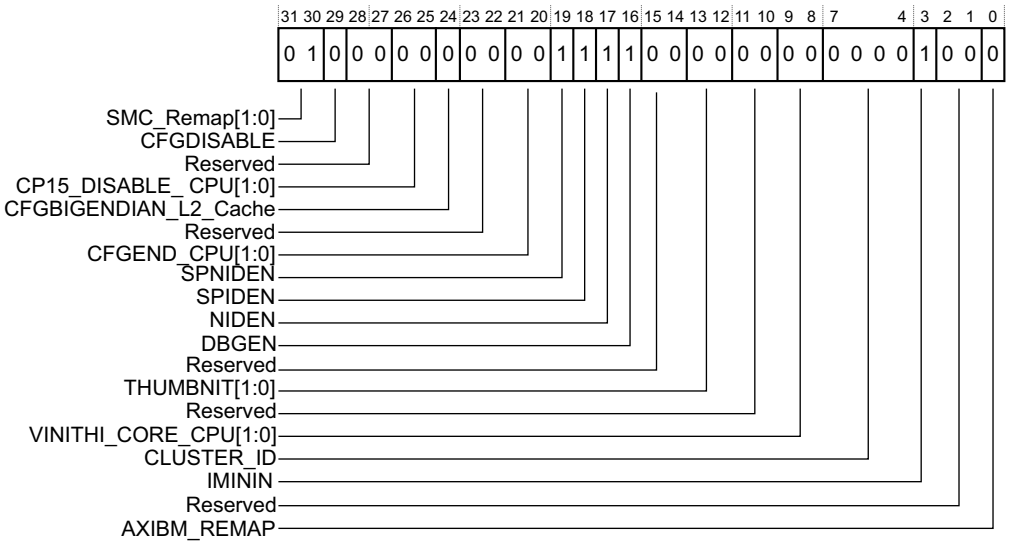


Figure 3-4 Test chip CFGREG0 Register bit assignments

Table 3-6 shows the bit assignments.

Table 3-6 Test chip CFGREG0 Register bit assignments

Bits	Name	Function
[31:30]	SMC remap	These map to the **SMC_REMAP[1:0]** bus: `b00` Reserved. `b01` CS0. `b10` CS4. `b11` Reserved. These bits are valid only if SMC is mapped to `0x0`, that is CFGREG0[0] = `b0`
[29]	CFGDISABLE	GIC configuration disable.
[28:27]	-	Reserved. Do not modify.

Table 3-6 Test chip CFGREG0 Register bit assignments (continued)

Bits	Name	Function
[26:25]	CP15_DISABLE_CPU[1:0]	Maps to the **CP15_DISABLE_CPU[1:0]** bus: b0 Co-processor 15 enable. b1 Co-processor 15 disable.
[24]	CFGBIGENDIAN_L2_Cache	Maps to the **CFGBIGENDIAN_L2_Cache** signal. Reserved. Do not modify.
[23:22]	-	Reserved. Do not modify.
[21:20]	CFGEND[1:0]	Maps to the **CFGEND[1:0]** bus. Configures CPUs as bigend.
[19]	SPNIDEN	Maps to the **SPNIDEN** secure non-invasive debug signal for both CPUs.
[18]	SPIDEN	Maps to the **SPIDEN** secure invasive debug signal for both CPUs.
[17]	NIDEN	Maps to the **NIDEN** non-invasive debug enable signal.
[16]	DBGEN	Maps to the **DBGEN** invasive debug enable signal.
[15:14]	-	Reserved. Do not modify.
[13:12]	THUMBNIT_CORE[1:0]	Thumbnit input for CPU[1:0].
[11:10]	-	Reserved. Do not modify.
[9:8]	VINITHI_CORE[1:0]	Vinithi input for CPU[1:0].
[7:4]	CLUSTER_ID	Maps to the **CLUSTERID[3:0]** bus.
[3]	IMININ	ICache minimum line size: b0 32 bytes. b1 64 bytes.
[2:1]	-	Reserved. Do not modify.
[0]	AXI_REMAP	NIC-301 AMBA AXI memory map: b0 SMC mapped to 0x00_0000_0000. b1 AXI Master interface mapped to 0x00_0000_0000.

Test chip SCC Register 1

The CFGREG1 Register characteristics are:

Purpose Miscellaneous Configuration Register 1 that enables you to read and write test chip configuration settings.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-5 on page 3-17 shows the bit assignments.

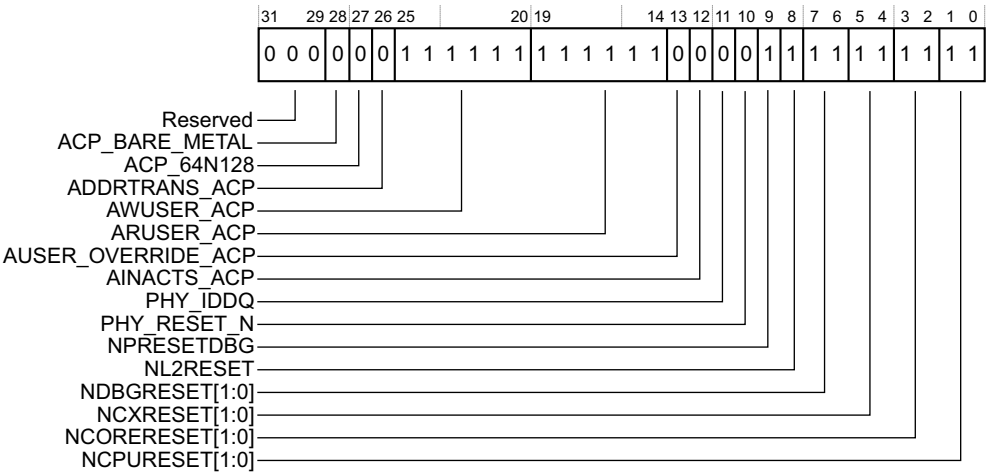


Figure 3-5 Test chip CFGREG1 Register bit assignments

Table 3-7 shows the bit assignments.

Table 3-7 Test chip CFGREG1 Register bit assignments

Bits	Name	Function
[31:29]	-	Reserved. Do not modify.
[28]	ACP_BARE_METAL	Selects DMA bare metal access to ACP interface: `b0` DMA accesses ACP through NIC-301 fabric. `b1` DMA directly accesses ACP interface.
[27]	ACP_64N128	Selects 64-bit or 128-bit operation for the ACP interface: `b0` 128-bit ACP. `b1` 64-bit ACP.
[26]	ADDRTRANS_ACP	Translates ACP addresses `0x00_3000_0000` to `0x00_3FFF_FFFF` by overriding bit 28 of the AxADDR of the cluster ACP: `b0` `00x_E000_0000`. `b1` `00x_F000_0000`.
[25:20]	AWUSER_ACP[5:0]	ACP AWCACHE override: AWUSER[5:2] = AWCACHE[3:0] if AUSER_OVERRIDE_ACP = `b0` AWUSER[5:2] = CFGREG1[25:22] if AUSER_OVERRIDE_ACP = `b1` AWUSER[1:0] = CFGREG1[21:20]
[19:14]	ARUSER_ACP[5:0]	ACP ARCACHE override: ARUSER[5:2] = ARCACHE[3:0] if AUSER_OVERRIDE_ACP = `b0` ARUSER[5:2] = CFGREG1[19:16] if AUSER_OVERRIDE_ACP = `b1` ARUSER[1:0] = CFGREG1[15:14]
[13]	AUSER_OVERRIDE_ACP	ACP AxUSER[5:2] override enable.
[12]	AINACTS_ACP	ACP disable: `b0` ACP enabled. `b1` ACP disabled.
[11]	PHY_IDDQ	Forces DDR PHY into IDDQ power-down mode: `b0` DDR PHY enabled. `b1` DDR PHY in IDDQ power-down mode.

Table 3-7 Test chip CFGREG1 Register bit assignments (continued)

Bits	Name	Function
[10]	PHY_RESET_N	PHY reset. Release after programming the PHY through APB and LOCK signal/register is asserted.
[9]	NPRESETDBG	Resets the initialized shared debug, APB, CTI, and CTM logic in the **PCLK** domain. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b1`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.
[8]	NL2RESET	Resets the L2 logic, interrupt controller, and timer logic. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b1`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.
[7:6]	NDBGRESET[1:0]	Resets the debug, PTM, break-point, and watch-point in the **FCLK** domain. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b11`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.
[5:4]	NCXRESET[1:0]	Resets NEON-VFP. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b11`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.
[3:2]	NCORERESET[1:0]	Resets the entire CPU including NEON-VPF, debug, and PTM, but excludes break-point and watch-point logic. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b11`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.
[1:0]	NCPURESET[1:0]	Resets the entire CPU including NEON-VPF, debug, PTM, break-point, and watch-point logic in the **FCLK** domain. This reset operates independently of **nRESET**: `b0` Reset. `b1` Non-reset. The default is `b11`. See the *Cortex-A15 Technical Reference Manual* for information on using this internal reset.

Test chip SCC Register 2

The CFGREG2 Register characteristics are:

Purpose SMC CS0 and CS1 Register that enables you to read and write match and mask bits for the SMC CS0 and SMC CS1. See *Mask operation to define SMC chip select address ranges* on page 3-13.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-6 shows the bit assignments.

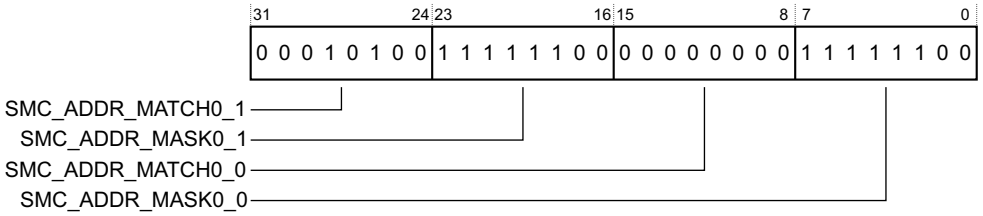


Figure 3-6 Test chip CFGREG2 Register bit assignments

Table 3-8 shows the bit assignments.

Table 3-8 Test chip CFGREG2 Register bit assignments

Bits	Name	Function
[31:24]	SMC_ADDR_MATCH0_1	SMC CS1 address match of top 8 bits
[23:16]	SMC_ADDR_MASK0_1	SMC CS1 address mask of top 8 bits
[15:8]	SMC_ADDR_MATCH0_0	SMC CS0 address match of top 8 bits
[7:0]	SMC_ADDR_MASK0_0	SMC CS0 address mask of top 8 bits

Test chip SCC Register 3

The CFGREG3 Register characteristics are:

Purpose SMC CS2 and CS3 Register that enables you to read and write match and mask bits for the SMC CS2 and SMC CS3. See *Mask operation to define SMC chip select address ranges* on page 3-13.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-7 on page 3-20 shows the bit assignments.

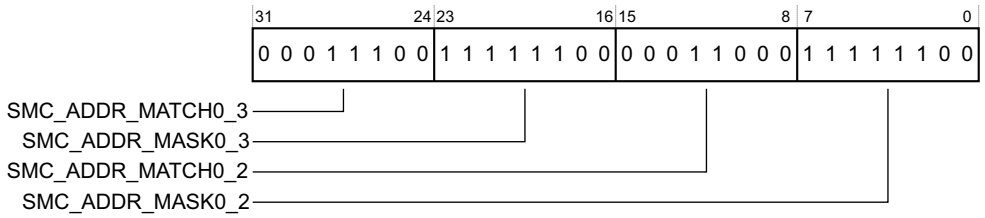


Figure 3-7 Test chip CFGREG3 Register bit assignments

Table 3-9 shows the bit assignments.

Table 3-9 Test chip CFGREG3 Register bit assignments

Bits	Name	Function
[31:24]	SMC_ADDR_MATCH0_3	SMC CS3 address match of top 8 bits
[23:16]	SMC_ADDR_MASK0_3	SMC CS3 address mask of top 8 bits
[15:8]	SMC_ADDR_MATCH0_2	SMC CS2 address match of top 8 bits
[7:0]	SMC_ADDR_MASK0_2	SMC CS2 address mask of top 8 bits

Test chip SCC Register 4

The CFGREG4 Register characteristics are:

Purpose SMC CS4 and CS5 Register that enables you to read and write match and mask bits for the SMC CS4. See *Mask operation to define SMC chip select address ranges* on page 3-13. SMC CS5 is reserved, and the match and mask bits for CS5 are also reserved.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-8 shows the bit assignments.

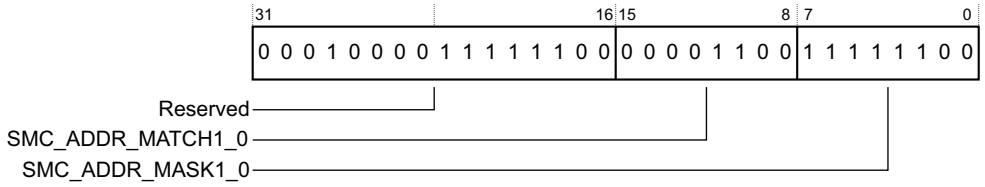


Figure 3-8 Test chip CFGREG4 Register bit assignments

Table 3-10 shows the bit assignments.

Table 3-10 Test chip CFGREG4 Register bit assignments

Bits	Name	Function
[31:24]	-	Reserved. Do not modify.

Table 3-10 Test chip CFGREG4 Register bit assignments (continued)

Bits	Name	Function
[23:16]	-	Reserved. Do not modify.
[15:8]	SMC_ADDR_MATCH1_0	SMC CS4 address match of top 8 bits.
[7:0]	SMC_ADDR_MASK1_0	SMC CS4 address mask of top 8 bits.

Test chip SCC Register 5

The CFGREG5 Register characteristics are:

Purpose SMC CS6 and CS7 Register that controls the match and mask bits of CS6 and CS7. See *Mask operation to define SMC chip select address ranges* on page 3-13. CS6 and CS7 are reserved, and the bits in this register are reserved.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-9 shows the bit assignments.

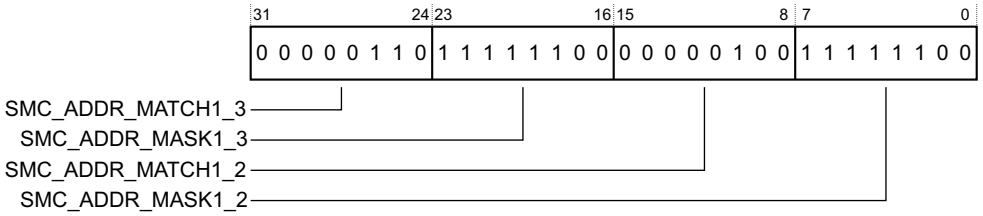


Figure 3-9 Test chip CFGREG5 Register bit assignments

Table 3-11 shows the bit assignments.

Table 3-11 Test chip CFGREG5 Register bit assignments

Bits	Name	Function
[31:24]	-	Reserved. Do not modify.
[23:16]	-	Reserved. Do not modify.
[15:8]	-	Not used. Reserved. Do not modify.
[7:0]	-	Not used. Reserved. Do not modify.

Test chip SCC CUSTOMER_ID Register

The CUSTOMER_ID Register characteristics are:

Purpose Enables you to read information about the Cortex-A15 test chip.

Usage constraints This register is read only.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-10 shows the bit assignments.

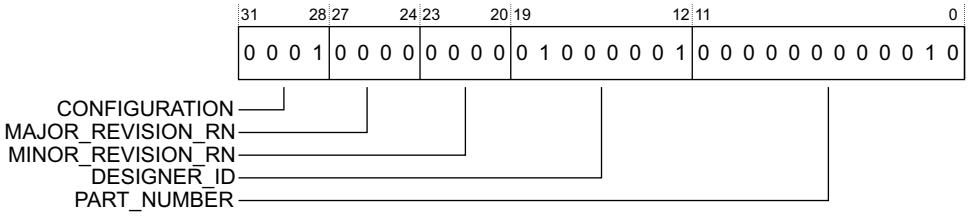


Figure 3-10 Test chip CUSTOMER_ID Register bit assignments

Table 3-12 shows the bit assignments.

Table 3-12 Test chip CUSTOMER_ID Register bit assignments

Bits	Name	Function
[31:28]	CONFIGURATION	Cortex-A15 test chip configuration
[27:24]	MAJOR_REVISION_RN	Cortex-A15 test chip major revision number
[23:20]	MINOR_REVISION_RN	Cortex-A15 test chip minor revision number
[19:12]	DESIGNER_ID	Cortex-A15 test designer
[11:0]	PART_NUMBER	Cortex-A15 part number

Note

Figure 3-10 and Table 3-12 show a test chip that has revision number r0p0 and designer ID 41, representing ARM, and part number `0x002`, representing LPAE.

Test chip SCC Register 6

The CFGREG6 Register characteristics are:

Purpose Enables you to read the DAP ROM default target ID.

Usage constraints This register is read only.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-11 shows the bit assignments.

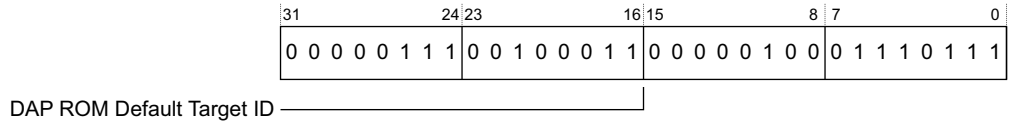


Figure 3-11 Test chip CFGREG6 Register bit assignments

Table 3-13 shows the bit assignments.

Table 3-13 Test chip CFGREG6 Register bit assignments

Bits	Name	Function
[31:0]	-	DAP ROM default target ID

Test chip SCC Register 7

The CFGREG7 Register characteristics are:

Purpose Enables you to read the DAP ROM default instance ID.

Usage constraints This register is read only.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-12 shows the bit assignments.

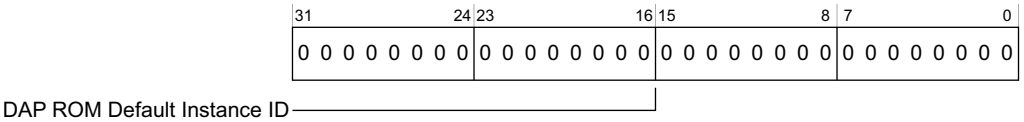


Figure 3-12 Test chip CFGREG7 Register bit assignments

Table 3-14 shows the bit assignments.

Table 3-14 Test chip CFGREG7 Register bit assignments

Bits	Name	Function
[31:0]	-	DAP ROM default instance ID

Test chip SCC Register 9

The CFGREG9 Register characteristics are:

Purpose Clock Control Register that enables you to read and write test chip clock control configuration settings.

Usage constraints Some bit combinations are invalid. See Table 3-15 on page 3-24.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-13 on page 3-24 shows the bit assignments.

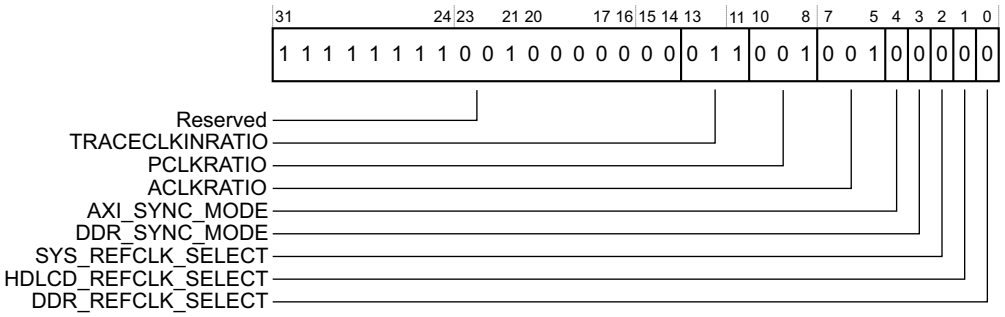


Figure 3-13 Test chip CFGREG9 Register bit assignments

Table 3-15 shows the bit assignments.

Table 3-15 Test chip CFGREG9 Register bit assignments

Bits	Name	Function
[31:14]	-	Reserved. Do not modify.
[13:11]	TRACECLKINRATIO	**TRACECLKIN** divider ratio. This divides the output of SYS PLL to derive **TRACECLKIN**: `b000` Divider ratio 1. `b001` Divider ratio 2. `b010` Divider ratio 3. `b011` Divider ratio 4. The default is `b011`. All other bit combinations are invalid. See Figure 2-11 on page 2-27. **Note** The maximum operating frequency of **TRACECLKIN** is 140MHz.
[10:8]	PCLKRATIO	**PCLK** divider ratio. This divides the output of SYS PLL to derive **PCLK**. `b000` Divider ratio 1. `b001` Divider ratio 2. `b010` Divider ratio 3. `b011` Divider ratio 4. The default is `b001` All other bit combinations are invalid. See Figure 2-11 on page 2-27. **Note** The maximum operating frequency of **PCLK** is 1500MHz.

Table 3-15 Test chip CFGREG9 Register bit assignments (continued)

Bits	Name	Function
[7:5]	ACLKRATIO	**ACLK** divider ratio: This divides the output of SYS PLL to derive **ACLK**. `b000` Divider ratio 1. `b001` Divider ratio 2. `b010` Divider ratio 3. `b011` Divider ratio 4. The default is `b001`. All other bit combinations are invalid. See Figure 2-11 on page 2-27. **Note** The maximum operating frequency of **ACLK** is 770MHz.
[4]	AXI_SYNC_MODE	Select **SYSCLK** or **CPU_CLK** as the source for **FCLK**: `b0` Select **CPU_CLK** as the source for **FCLK**. The NIC-301 matrix operates asynchronously to CPU cores. `b1` Select **SYSCLK** as the source for **FCLK**. The NIC-301 matrix operates synchronously to CPU cores. The default is `b0`. See Figure 2-10 on page 2-21. **Note** The maximum frequency of **FCLK** is 1400MHz.
[3]	DDR_SYNC_MODE	Select **ACLK** or **DDRCLK** to clock the PL341 memory interface and the DDR2 physical interface, PHY: `b0` Select **DDRCLK**. The PL341 memory interface and the DDR2 physical interface (PHY) operate asynchronously to AXI domain. `b1` Select **ACLK**. The PL341 memory interface and the DDR2 physical interface (PHY) operate synchronously with AXI domain. The default is `b0`. See Figure 2-14 on page 2-33. **Note** The maximum frequency of the PL341 memory interface and DDR2 PHY clock is 400MHz.

Table 3-15 Test chip CFGREG9 Register bit assignments (continued)

Bits	Name	Function
[2]	SYS_REFCLK_SELECT	Select reference clock to SYS PLL: b0 Select **SYSREFCLK** as the input reference for SYS PLL. b1 Select **CPUREFCLK** as the input reference for SYS PLL. The default is b0. See Table 2-8 on page 2-24. **Note** ARM does not recommend using the non-default option and Figure 2-10 on page 2-21 and Figure 2-11 on page 2-27 do not show this connection.
[1]	HDLCD_REFCLK_SELECT	Select reference clock to HDLCD PLL: b0 Select **PXLREFCLK** as the input reference for SYS PLL. b1 Select **CPUREFCLK** as the input reference for SYS PLL. The default is b0. See Table 2-8 on page 2-24. **Note** The maximum frequency of **PXLCL**K and **MMB_IDCLK** is 261MHz. ARM does not recommend using the non-default option and Figure 2-10 on page 2-21 and Figure 2-11 on page 2-27 do not show this connection.
[0]	DDR_REFCLK_SELECT	Select reference clock to DDR PLL: b0 Select **DDRREFCLK** as the input reference for SYS PLL. b1 Select **CPUREFCLK** as the input reference for SYS PLL. The default is b0. See Table 2-8 on page 2-24. **Note** The maximum value of **DDRCLK** is 400MHz. ARM does not recommend using the non-default option and Figure 2-10 on page 2-21 and Figure 2-11 on page 2-27 do not show this connection.

Test chip SCC Registers 10, 12, 14 and 16

The CFGREG10, CFGREG12, CFGREG14, and CFGREG16 Register characteristics are:

Purpose

The CFGREG10, CFGREG12, CFGREG14, and CFGREG16 registers, together with CFGREG11, CFGREG13, CFGREG15, and CFGREG17, control the PLLs that generate some of the internal clocks inside the test chip.

Two registers control each PLL as follows:

- CFGREG10 and CFGREG11 control CPU PLL.
- CFGREG12 and CFGREG13 control DDR PLL.
- CFGREG14 and CFGREG15 control HDLCD PLL.
- CFGREG16 and CFGREG17 control SYS PLL.

These registers enable you to read and write test chip CPU PLL, SYS PLL, DDR PLL, and HDLCD PLL configuration settings.

See *Clocks* on page 2-19.

Usage constraints Bit 5 is read only.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-14 shows the bit assignments.

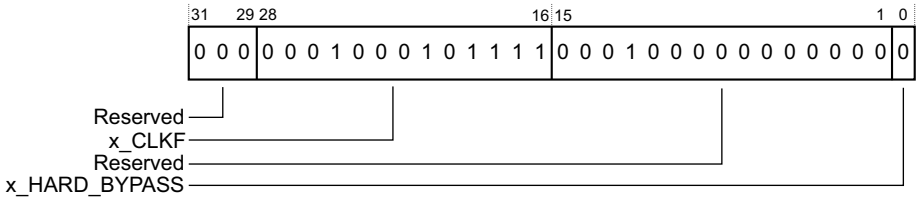


Figure 3-14 Test chip CFGREG10, CFGREG12, CFGREG14, and CFGREG16 Register bit assignments

Table 3-16 shows the bit assignments.

Table 3-16 CFGREG10, CFGREG12, CFGREG14, and CFGREG16 Register bit assignments

Bits	Name	Function
[31:29]	-	Reserved. Do not modify.

Table 3-16 CFGREG10, CFGREG12, CFGREG14, and CFGREG16 Register bit assignments (continued)

Bits	Name	Function
[28:16]	x_CLKF	PLL feedback clock divider settings. Divisor = x_CLKF+1. These bits have the following default values: **CFGREG10 - CPU PLL** b0001000101111, decimal 559. **CFGREG12 - DDR PLL** b0000010111111, decimal 191. **CFGREG14 - HDLCD PLL** b0000111001101, decimal 461. **CFGREG16 - SYS PLL** b0001000101111, decimal 559. See Figure 2-11 on page 2-27. **Note** See Table 2-7 on page 2-22 for the maximum clock operating frequencies.
[15:1]	-	Reserved. Do not modify.
[0]	x_HARD_BYPASS	This bit forces the reference input clock to bypass the PLL to enable it to be driven directly into the design: b0 Do not bypass PLL. b1 Bypass PLL. This bit has the following default values: **CFGREG10 - CPU PLL** b0. **CFGREG12 - DDR PLL** b0. **CFGREG14 - HDLCD PLL** b0. **CFGREG16 - SYS PLL** b0. **Note** The board.txt file sets the HDLCD PLL to bypass by setting CFGREG14[0] to b1 during the configuration process. You can set CFGREG14[0] to b0 if you want to use the HDLCD PLL. ARM does not recommend using the other non-default options and Figure 2-10 on page 2-21 and Figure 2-11 on page 2-27 do not show these connections.

Test chip SCC Registers 11, 13, 15 and 17

The CFGREG11, CFGREG13, CFGREG15, and CFGREG17 Register characteristics are:

Purpose The CFGREG11, CFGREG13, CFGREG15, and CFGREG17 registers, together with CFGREG10, CFGREG12, CFGREG14, and CFGREG16, control the PLLs that generate some of the internal clocks inside the test chip.

Two registers control each PLL as follows:

- CFGREG10 and CFGREG11 control CPU PLL.
- CFGREG12 and CFGREG13 control DDR PLL.
- CFGREG14 and CFGREG15 control HDLCD PLL.

- CFGREG16 and CFGREG17 control SYS PLL.

These registers enable you to read and write test chip CPU PLL, SYS PLL, DDR PLL, and HDLCD PLL configuration settings.

See *Clocks* on page 2-19.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-15 shows the bit assignments.

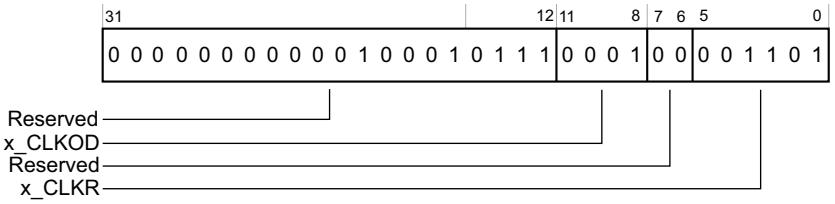


Figure 3-15 Test chip CFGREG11, CFGREG13, CFGREG15, and CFGREG17 Register bit assignments

Table 3-17 shows the bit assignments.

Table 3-17 Test chip CFGREG11, CFGREG13, CFGREG15, and CFGREG17 Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved. Do not modify.

Table 3-17 Test chip CFGREG11, CFGREG13, CFGREG15, and CFGREG17 Register bit assignments (continued)

Bits	Name	Function
[11:8]	x_CLKOD	PLL output divider settings. Divisor = x_CLOD+1. These bits have the following default values: **CFGREG11- CPU PLL** b0001. **CFGREG13 - DDR PLL** b0101. **CFGREG15 - HDLCD PLL** b1101. **CFGREG17 - SYS PLL** b1101. See Figure 2-11 on page 2-27. **Note** See Table 2-7 on page 2-22 for the maximum clock operating frequencies.
[7:6]	-	Reserved. Do not modify.
[5:0]	x_CLKR	PLL reference clock divider settings. Divisor = x_CLKR+1. These bits have the following default values: **CFGREG11 - CPU PLL** b001101. **CFGREG13 - DDR PLL** b000011. **CFGREG15 - HDLCD PLL** b001001. **CFGREG17 - SYS PLL** b001101. See Figure 2-11 on page 2-27. **Note** See Table 2-7 on page 2-22 for the maximum clock operating frequencies.

Test chip SCC Register 22

The CFGREG22 Register characteristics are:

Purpose DMAC register 0 that enables you to read from and write control signals to the *Dynamic Memory Access Controller* (DMAC), DMA-330.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-16 on page 3-31 shows the bit assignments.

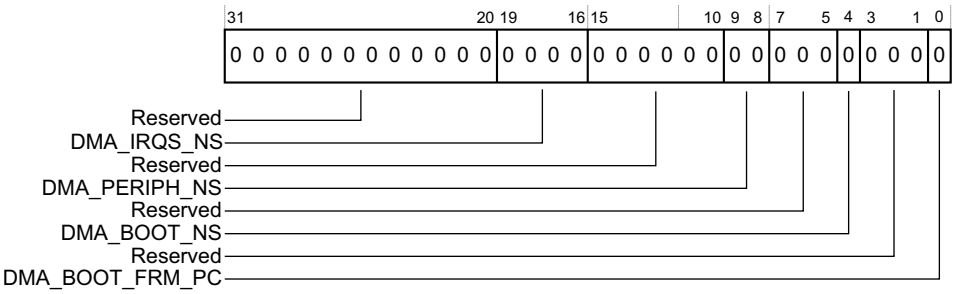


Figure 3-16 Test chip CFGREG22 Register bit assignments

Table 3-18 shows the bit assignments.

Table 3-18 Test chip CFGREG22 Register bit assignments

Bits	Name	Function
[31:20]	-	Reserved. Do not modify.
[19:16]	DMA_IRQS_NS[3:0]	These bits control the security state of the interrupt outputs when the DMAC exits from reset: **DMA_IRQ_NS[x]** = `b0` The DMAC assigns IRQ[x] to the Secure state. **DMA_IRQ_NS[x]** = `b1` The DMAC assigns IRQ[x] to the Non-secure state. See the *AMBA DMA Controller DMA-330 Technical Reference Manual*.
[15:10]	-	Reserved. Do not modify.
[9:8]	DMA_PERIPH_NS[1:0]	Boot peripherals security state. These bits control the security state of the peripheral request interface when the DMAC exits from reset: **DMA_PERIPH_NS[x]** = `b0` The DMAC assigns peripheral request interface[x] to the Secure state. **DMA_PERIPH_NS[x]** = `b1` The DMAC assigns peripheral interface request interface[x] to the Non-secure state.
[7:5]	-	Reserved. Do not modify.
[4]	DMA_BOOT_NS	Boot manager security state. When the DMAC exits from reset, this bit controls the security state of the DMA manager thread: `b0` Assigns DMA manager to the Secure state. `b1` Assigns DMA manager to the Non-secure state.
[3:1]	-	Reserved. Do not modify.
[0]	DMA_BOOT_FRM_PC	Boot from program counter. This controls the location of where the DMAC executes its initial instruction after it exits from reset: `b0` The DMAC waits for an instruction from the APB interface. `b1` The DMAC executes the instruction that is located at the address that DMAC register 1 provides. See *Test chip SCC Register 23* on page 3-32.

Test chip SCC Register 23

The CFGREG23 Register characteristics are:

Purpose DMAC register 1 that enables you to read from and write control signals to the *Dynamic Memory Access Controller* (DMAC), DMA-330.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-17 shows the bit assignments.

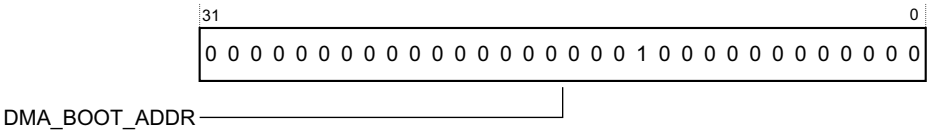


Figure 3-17 Test chip CFGREG23 Register bit assignments

Table 3-19 shows the bit assignments.

Table 3-19 Test chip CFGREG23 Register bit assignments

Bits	Name	Function
[31:0]	DMA_BOOT_ADDR	Boot address for the DMAC. This configures the address location that contains the first instruction that the DMAC executes when it exits from reset. **Note** The DMAC uses this address only when DMA_BOOT_FRM_PC is HIGH. See *Test chip SCC Register 22* on page 3-30.

Test chip SCC Register APB_CLEAR

The APB_CLEAR Register characteristics are:

Purpose Writing `0xA50FF0FA to the` APB_CLEAR register reverts all SCC registers to their test chip default values.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-18 shows the bit assignments.

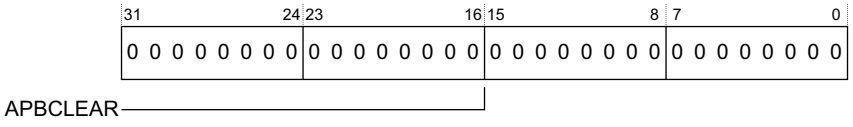


Figure 3-18 Test chip APB_CLEAR Register bit assignments

Table 3-20 shows the bit assignments.

Table 3-20 Test chip APB_CLEAR Register bit assignments

Bits	Name	Function
[31:0]	-	Writing `0xA50FF0FA` to this register reverts the SCC registers to their test chip default values. **Note** ARM recommends that you do not write to this register.

Test chip SCC Register TC_ID

The DEVICEID Register characteristics are:

Purpose Enables you to read the Cortex-A15 MPCore test chip specific device ID.

Usage constraints This register is read-only.

Configurations Not applicable.

Attributes See Table 3-5 on page 3-12.

Figure 3-19 shows the bit assignments.

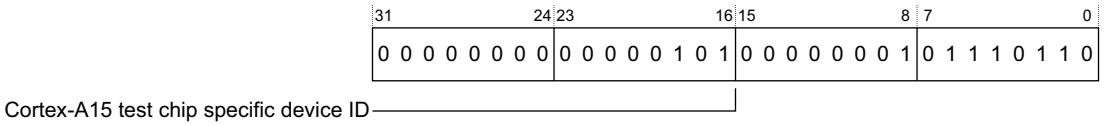


Figure 3-19 Test chip TC_ID Register bit assignments

Table 3-21 shows the bit assignments.

Table 3-21 Test chip TC_ID Register bit assignments

Bits	Name	Function
[31:0]	Cortex-A15 test chip specific device ID	Cortex-A15 test chip device ID. The default is `0x00050176`.

Test chip SCC Register CPU_ID

The CPUID Register characteristics are:

Purpose Enables you to read the Cortex-A15 MPCore test chip CPU ID.

Usage constraints This register is read-only.

Configurations Not applicable.

Attributes Table 3-5 on page 3-12.

Figure 3-20 shows the bit assignments.

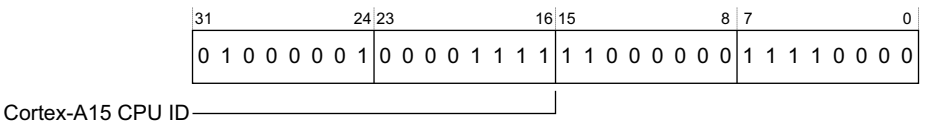


Figure 3-20 Test chip CPU_ID Register bit assignments

Table 3-22 shows the bit assignments.

Table 3-22 Test chip CPUID Register bit assignments

Bits	Name	Function
[31:0]	-	Cortex-A15 CPU ID

3.4 Programmable peripherals and interfaces

The following sections describe the configurable modules in the test chip:

- *Cortex-A15 MPCore cluster*
- *AXI network interconnect, NIC-301* on page 3-36
- *HDLCD controller* on page 3-36
- *Cortex-A15 L2 cache controller* on page 3-36
- *DDR2 memory controllers, PL341* on page 3-37
- *Static memory controller, PL354* on page 3-38
- *Direct memory access controller, DMA-330* on page 3-39
- *Watchdog, SP805* on page 3-39
- *System counter* on page 3-40.

See also the reference manual for each peripheral for more information on programming these devices.

3.4.1 Cortex-A15 MPCore cluster

The Cortex-A15 MPCore cluster consists of a Cortex-A15 MPCore multiprocessor that includes two Cortex-A15 CPUs with NEON media processing technology, SIMD extension, and FPU. The L1 memory subsystem has 32KB of instruction cache, and 32KB of data cache for each CPU.

The A15 MPCore cluster contains a *Generic Interrupt Controller* (GIC). Table 3-23 shows information on the implementation of the GIC.

Table 3-23 Interrupt controller implementation

Interrupt controller block	Property	Value
GIC	Memory base address	`0x00_2C00_0000`
Reserved	Offset from base address	`0x00_0000_0000` - `0x00_0000_0FFF`
Interrupt controller distributor	Offset from base address	`0x00_0000_1000` - `0x00_0000_1FFF`
Interrupt controller physical CPU interface	Offset from base address	`0x00_0000_2000` - `0x00_0000_3FFF`
Interrupt controller virtual CPU interface, hypervisor view	Offset from base address	`0x00_0000_4000` - `0x00_0000_5FFF`
Interrupt controller virtual CPU interface, virtual machine view	Offset from base address	`0x00_0000_6000` - `0x00_0000_7FFF`

For information about the programmable devices within the Cortex-A15 MPCore processor, see the *Cortex-A15 Technical Reference Manual*.

3.4.2 AXI network interconnect, NIC-301

The internal AXI is clocked by **ACLK**. The internal AXI operates asynchronously to the Cortex-A15 MPCore cluster by default. You can operate the internal AXI synchronously to the cluster by selecting **SYSCLK** to clock the cluster. See *Clocks* on page 2-19 and *Test chip SCC register descriptions* on page 3-15.

Table 3-24 provides information on the AXI interconnect implementation.

Table 3-24 AXI network interconnect implementation

Property	Value
Memory base address	`0x00_2A00_0000`
Interrupt	128
Release version	ARM PL301 r2p1
Reference documentation	*AMBA® Network Interconnect (NIC-301) Technical Reference Manual*

3.4.3 HDLCD controller

Table 3-25 provides information on the HDLCD controller implementation.

Table 3-25 HDLCD controller implementation

Property	Value
Memory base address	`0x00_2B00_0000`
Interrupt	117

See Appendix B *HDLCD controller* for a full description of the HDLCD video controller.

3.4.4 Cortex-A15 L2 cache controller

Table 3-26 provides information on the Cortex-A15 Level-2 cache controller implementation.

Table 3-26 Cortex-A15 Level-2 cache controller implementation

Property	Value
Memory base address	Not applicable. The L2-cache controller is integrated into the Cortex-A15 MPCore cluster.
Cache size	1024KB.
Number of Ways	16.
Line size	64 bytes.
Master port	Single 128-bit ACE interface.
Slave port	Single 128-bit AXI 4 ACP interface.
Physically indexed and tagged cache	-
Parity	*Error Correction Code* (ECC) if enabled.

Table 3-26 Cortex-A15 Level-2 cache controller implementation (continued)

Property	Value
Default Data and Tag RAM latencies	2 cycles.
Release version	Not applicable.
Reference documentation	*Cortex-A15 Technical Reference Manual*.

Note

The default Level-2 data and tag RAM latency values in the Level-2 control register in the Cortex-A15 MPCore cluster are set to 2 cycles. You must increase these values to 3 cycles to match the high-density RAMs used for the Level-2 memory. You must perform this initialization in your OS boot code. See the *Cortex-A15 Technical Reference Manual* for more information on the Cortex-A15 MPCore Level-2 control register.

3.4.5 DDR2 memory controllers, PL341

The PL341 AXI interface runs asynchronously to the internal AXI interconnect, NIC-301, by default at the frequency that the daughterboard oscillator OSCLK 8 defines. See Figure 2-10 on page 2-21. See the example `board.txt` file in the *Versatile™ Express Configuration Technical Reference Manual* for information on how to set OSCCLK 8.

You can operate the PL341 memory interface and the DDR2 physical interface, PHY, synchronously with the AXI interconnect by selecting **ACLK** to clock these blocks. See *Clocks* on page 2-19 and *Test chip SCC registers* on page 3-11.

Table 3-27 provides information on the DDR2 memory controller implementation.

Table 3-27 DDR2 memory controllers implementation

Property	Value
Memory base address	Controller 0: `0x__00_2B0A_0000`.
Configuration	The configuration settings of the PL341 memory controllers are as follows: **AXI width** 64. **MEMIF width** 32. **Banks per chip** 8. **Exclusion monitors** 2. **Arbitration depth** 16. **CFIFO depth** 8. **WFIFO depth** 32. **RFIFO depth** 32. **AID width** 15. **DFI** 1. **ECC** FALSE.
Release version	ARM PL341 r1p0.
Reference documentation	*ARM PrimeCell DDR2 Dynamic Memory Controller (PL341) Technical Reference Manual*.

3.4.6 Static memory controller, PL354

The PL354 *Static Memory Controller* (SMC) connects between the NIC-AXI interconnect and the PL220 interface that drives the SMB.

SMC organization

The SMC accesses these devices on the motherboard using the following chip selects:

CS0 NOR flash 0.
CS1 PSRAM.
CS2 Video/ETH/USB.
CS3 system registers and peripherals.
CS4 NOR flash 1.
CS5 Reserved.
CS6 Reserved.
CS7 Reserved.

Static memory controller implementation

Table 3-28 provides information on the static memory controller implementation.

Table 3-28 Static memory controller implementation

Property	Value
Memory base address	`0x00_7FFD_0000`.
Interrupts	118, 119.
Configuration	The configuration settings of the PL354 static memory controller are as follows: **AXI width** 64. **MEMIF width** 32. **MEMIF CS** 8. **Exclusion monitors** 2. **CFIFO depth** 8. **WFIFO depth** 16. **RFIFO depth** 16. **AID width** 15. **ECC** FALSE. **Pipeline** TRUE.
Release version	ARM PL354 r2p1.
Reference documentation	*ARM PrimeCell Static Memory Controller (PL350 series) Technical Reference Manual*.

PrimeCell External Bus Interface, PL220

A PL220 *External Bus Interface* (EBI) multiplexes the dual SRAM memory interface to reduce pin count and to facilitate board layout.

See the *PrimeCell External Bus Interface (PL220) Technical Reference Manual* for more information.

3.4.7 Direct memory access controller, DMA-330

Table 3-29 shows information relating to the DMA implementation.

Table 3-29 DMA implementation

Property	Value
Memory base address	0x00_7FFB_0000
Interrupts	**Direct memory access controller** 120-123. **Direct memory access abort** 124.
Configuration	**ICache lines** 8. **ICache words** 4. **Number of VChannels** 4. **Number of PChannels** 4. **Write Issue Cap** 8. **Read Issue Cap** 8. **Read LSQ depth** 8. **Write LSQ depth** 8. **FIFO depth** 512. **FIFO implementation** Register. **Number of INTR** 4.
Release version	ARM PL330 r1p0.
Reference documentation	*AMBA® DMA-330 Technical Reference Manual*.

3.4.8 Watchdog, SP805

The SP805 watchdog module is an AMBA-compliant SoC peripheral developed and tested by ARM.

The watchdog module consists of a 32-bit down counter with a programmable time-out interval that can generate an interrupt and a reset signal when it times out. It can be used to apply a reset to a system in the event of a software failure.

Table 3-30 provides information on the Watchdog implementation.

Table 3-30 Watchdog implementation

Property	Value
Memory base address	0x00_2B06_0000
Interrupt	130
Release version	ARM WDOG SP805 r2p0
Reference documentation	*ARM Watchdog Module (SP805) Technical Reference Manual*

Note

The watchdog counter is disabled if the core is in the debug state.

3.4.9 System counter

The generic timers are part of the Cortex-A15 MPCore cluster. The system counter provides the Cortex-A15 MPCore timers with a real time reference.

Table 3-31 provides information on the system counter implementation.

Table 3-31 System counter implementation

Property	Value
Memory base address	`0x00_2A43_0000`.
Clock	Clocked by **ACLK**. See Figure 2-10 on page 2-21.

System counter register summary

Table 3-32 shows the counter system registers and corresponding offsets from the system counter base memory address.

Table 3-32 System counter register summary

Offset	Name	Type	Test chip reset	Width	Description
`0x00`	REGLWR	RW	`0x00000000`	32	System counter Load Register lower 32 bits. See *REGLWR* on page 3-41.
`0x04`	REGUPR	RW	`0x00000000`	32	System counter Load Register upper 32 bits. See *REGUPR* on page 3-41.
`0x08`	CNTLWR	RO	`0x00000000`	32	System counter lower 32 bits. See *CNTLWR* on page 3-42.
`0x0C`	CNTUPR	RO	`0x00000000`	32	System counter upper 32 bits. See *CNTUPR* on page 3-42.
`0x10`	CNTCTL	RW	-	32	Write any value to this register to load the system counter with the load register value. See *CNTCTL* on page 3-43.

System counter register descriptions

The following sections describes the system counter registers:

- *REGLWR* on page 3-41
- *REGUPR* on page 3-41
- *CNTLWR* on page 3-42
- *CNTUPR* on page 3-42
- *CNTUPR* on page 3-42.

REGLWR

The REGLWR Register characteristics are:

Purpose System counter load register lower 32 bits. A write of any value to the CNTCL register loads these bits into the CNTLWR register.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-32 on page 3-40.

Figure 3-21 shows the bit assignments.

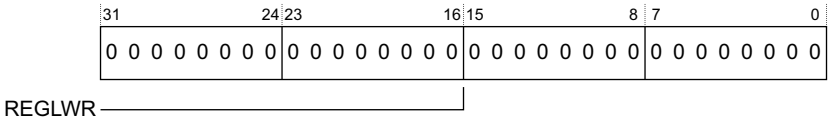


Figure 3-21 System counter REGLWR Register bit assignments

Table 3-33 shows the bit assignments.

Table 3-33 System counter REGLWR Register bit assignments

Bits	Name	Function
[31:0]	REGLWR	System counter lower 32 bits

REGUPR

The REGUPR Register characteristics are:

Purpose System counter load register upper 32 bits. A write of any value to the CNTCL register loads these bits into the CNTUPR register.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-32 on page 3-40.

Figure 3-22 shows the bit assignments.

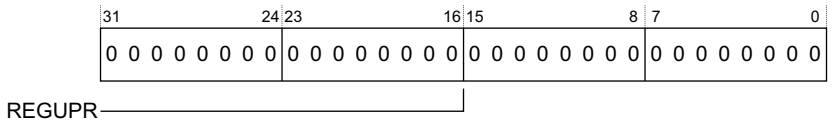


Figure 3-22 System counter REGUPR Register bit assignments

Table 3-34 shows the bit assignments.

Table 3-34 System counter REGUPR Register bit assignments

Bits	Name	Function
[31:0]	REGUPR	System counter upper 32 bits

CNTLWR

The CNTLWR Register characteristics are:

Purpose System counter lower 32 bits.

Usage constraints This register is read only.

Configurations Not applicable.

Attributes See Table 3-32 on page 3-40.

Figure 3-23 shows the bit assignments.

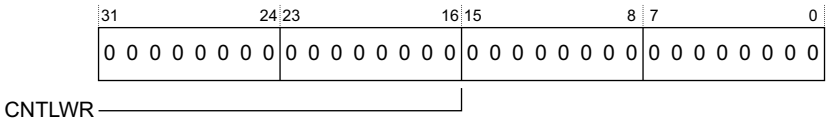


Figure 3-23 System counter CNTLWR Register bit assignments

Table 3-35 shows the bit assignments.

Table 3-35 System counter CNTLWR Register bit assignments

Bits	Name	Function
[31:0]	CNTLWR	System counter lower 32 bits

CNTUPR

The CNTUPR Register characteristics are:

Purpose System counter upper 32 bits.

Usage constraints This register is read only.

Configurations Not applicable.

Attributes See Table 3-32 on page 3-40.

Figure 3-24 shows the bit assignments.

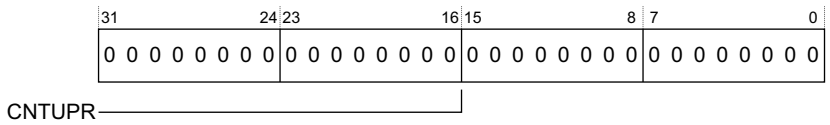


Figure 3-24 System counter CNTUPR Register bit assignments

Table 3-36 shows the bit assignments.

Table 3-36 System counter CNTUPR Register bit assignments

Bits	Name	Function
[31:0]	CNTUPR	System counter upper 32 bits

CNTCTL

The CNTCTL Register characteristics are:

Purpose System counter upper 32 bits.

Usage constraints There are no usage constraints.

Configurations Not applicable.

Attributes See Table 3-32 on page 3-40.

Table 3-37 shows the bit assignments.

Table 3-37 System counter CNTCTL Register bit assignments

Bits	Name	Function
[31:0]	CNTCTL	Write any value to this register to load the system counter registers CNTLWR and CNTUPR registers with the values in REGLWR and REGUPR.

Appendix A
Signal Descriptions

This appendix describes the signals present at the interface connectors. It contains the following sections:

Note

See also the *Motherboard Express µATX Technical Reference Manual*.

A.1 Daughterboard connectors

Figure A-1 shows the connectors fitted to the daughterboard.

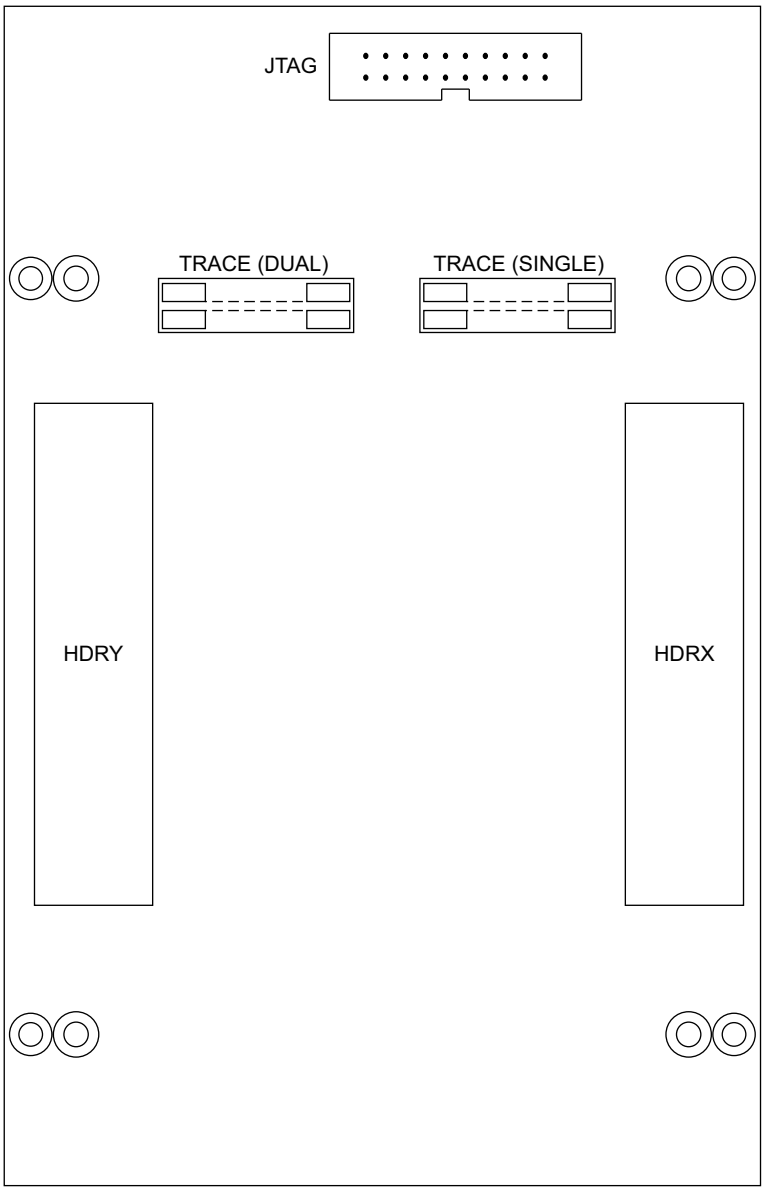


Figure A-1 CoreTile Express A15×2 daughterboard connectors

Note

All connectors in Figure A-1, except for the HDRY and HDRX headers, are on the upper face of the daughterboard facing away from the motherboard.

The HDRX and HDRY headers are on the lower face of the daughterboard facing towards the motherboard.

A.2 HDRX HSB multiplexing scheme

A bus multiplexing scheme is necessary to reduce the number of pins required on the HDRX header for the 64-bit AXI master on the HSBM bus. The LogicTile Express daughterboard must implement a similar multiplexing scheme to be compatible with the CoreTile Express signals.

OSCCLK 4 on the CoreTile Express A15×2 daughterboard generates the AXI master bus clock. This clocks the test chip master multiplex and demultiplex logic. This logic is positive- and negative-edge triggered to avoid the requirement for a PLL or double-rate clock in the test chip. Double-edge clocking also enables operation at low speed for use with emulation systems.

Note

All signals on the **HSB (M)** bus are 1.8V.

Figure A-2 shows a simplified block diagram of the multiplexing scheme for the AXI bus.

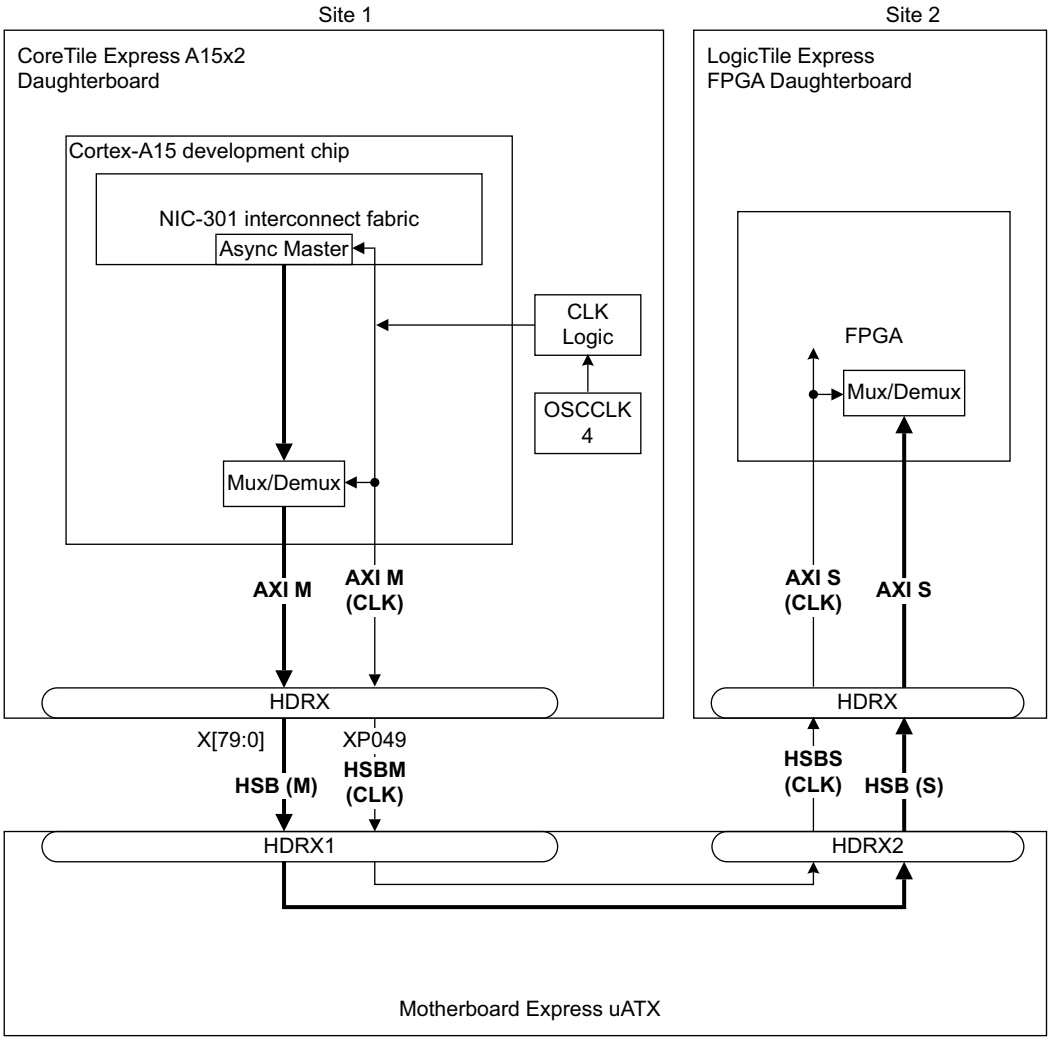


Figure A-2 HSB multiplexing

The AN283, *Example LogicTile Express 3MG design for a CoreTile Express A15×2* application note, provided by ARM, implements an example AMBA system using a LogicTile Express 3MG daughterboard to interconnect with the CoreTile Express A15×2 daughterboard. See the documentation supplied on the accompanying media and the *Application Notes* listing for more information, at, http://infocenter.arm.com.

A.3 Header connectors

The following high density headers are fitted to the underside of the daughterboard:

Header HDRY, J2 Routes the buses and power interconnect between the V2P-CA15 daughterboard and the V2M-P1 motherboard.

Header HDRX, J1 Routes the HSB buses between the V2P-CA15 daughterboard and the daughterboard on the other motherboard site.

The `an283_revx.ucf` constraints file available in application note AN283, *Example LogicTile Express 3MG design for a CoreTile Express A15×2* lists the HDRX signals.

A.4 Debug and trace connectors

This section describes the debug and trace connectors on the daughterboard. It contains the following subsections:

- *JTAG connector*
- *Trace connectors* on page A-7.

Caution

- Your external debug interface unit must adapt its interface voltages to the voltage level of the daughterboard JTAG. All the trace and JTAG signals operate at 1.8V.

A.4.1 JTAG connector

The daughterboard provides the JTAG connector to enable connection of DSTREAM or a compatible third-party debugger. Figure A-3 shows the JTAG connector.

Note

- **EDBGRQ** has a pull-down resistor to 0V. The daughterboard does not support adaptive clocking and **RTCK** has a pull-down resistor to 0V. All other signal connections on the P-JTAG connector have pull-up resistors to 1V8.
- Pins 7 and 9 of the JTAG connector are dual mode pins that enable the Cortex-A15 MPCore test chip to support both the JTAG and SWD protocols.

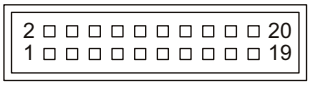


Figure A-3 JTAG connector, J6

Table A-1 shows the JTAG pin mapping for each JTAG signal.

Table A-1 JTAG connector, J6, signal list

Pin	Signal	Pin	Signal
1	**VTREFC**	2	**VSUPPLYA**
3	**nTRST**	4	**GND**
5	**TDI**	6	**GND**
7	**TMS/SWDIO**	8	**GND**
9	**TCK/SWCLK**	10	**GND**
11	**RTCK**	12	**GND**
13	**TDO**	14	**GND**
15	**nSRST**	16	**GND**
17	**EDBGRQ**	18	**GND**
19	**DBGACK**	20	**GND**

A.4.2 Trace connectors

The test chip supports up to 32-bit trace output from the CoreSight *Trace Port Interface Unit* (TPIU) and enables connection of a compatible trace unit. See Figure 2-15 on page 2-34. Two MICTOR trace connectors labeled *Trace Dual* and *Trace Single*, are connected to the TPIU. The two connectors used together support 32-bit trace and the connector labeled *Trace Single*, used alone, supports 16-bit trace.

Note

- DSTREAM is an example of a trace module that can be used.
- All the trace and JTAG signals operate at 1.8V.
- The trace connector cannot supply power to a trace unit.
- The interface does not support the **TRACE_CTL** signal. This is always driven LOW.

Figure A-4 shows the MICTOR connector, part number AMP 2-5767004-2.

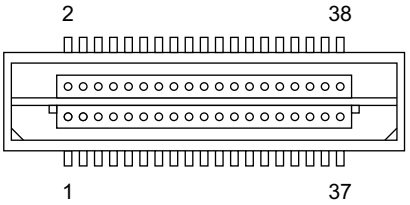


Figure A-4 Trace Connector, J4 and J5

Table A-2 shows the trace pin mapping for each *Trace Single* signal.

Table A-2 Trace Single connector, J4, signal list

Pin	Signal	Pin	Signal
1	Not connected	2	Not connected
3	Not connected	4	Not connected
5	**GND**	6	**TRACE_CLKA**
7	**TRACEDBGRQ**	8	**TRACE_DBGACK**
9	**nSRST**	10	**TRACE_EXTTRIGX**
11	**TDO**	12	**VTREFA**, 1V8
13	**RTCK**	14	**VSUPPYLYA**, 1V8
15	**TCK**	16	**TRACE_DATA7**
17	**TMS**	18	**TRACE_DATA6**
19	**TDI**	20	**TRACE_DATA5**
21	**nTRST**	22	**TRACE_DATA4**
23	**TRACE_DATA15**	24	**TRACE_DATA3**
25	**TRACE_DATA14**	26	**TRACE_DATA2**
27	**TRACE_DATA13**	28	**TRACE_DATA1**
29	**TRACE_DATA12**	30	**GND**

Table A-2 Trace Single connector, J4, signal list (continued)

Pin	Signal	Pin	Signal
31	**TRACE_DATA11**	32	**GND**
33	**TRACE_DATA10**	34	**VTREFA**, 1V8
35	**TRACE_DATA9**	36	**TRACE_CTL**
37	**TRACE_DATA8**	38	**TRACE_DATA0**

Table A-3 shows the trace pin mapping for each *Trace Dual* signal.

Table A-3 Trace Dual connector, J5, signal list

Pin	Signal	Pin	Signal
1	Not connected	2	Not connected
3	Not connected	4	Not connected
5	**GND**	6	**TRACECLKB**
7	Not connected	8	Not connected
9	Not connected	10	Not connected
11	Not connected	12	**VTREF**, 1V8
13	Not connected	14	Not connected
15	Not connected	16	**TRACE_DATA23**
17	Not connected	18	**TRACE_DATA22**
19	Not connected	20	**TRACE_DATA21**
21	Not connected	22	**TRACE_DATA20**
23	**TRACE_DATA31**	24	**TRACE_DATA19**
25	**TRACE_DATA30**	26	**TRACE_DATA18**
27	**TRACE_DATA29**	28	**TRACE_DATA17**
29	**TRACE_DATA28**	30	**GND**
31	**TRACE_DATA27**	32	**GND**
33	**TRACE_DATA26**	34	**VTREF**, 1V8
35	**TRACE_DATA25**	36	**GND**
37	**TRACE_DATA24**	38	**TRACE_DATA16**

Appendix B
HDLCD controller

This appendix describes the HDLCD controller. It contains the following sections:

B.1 Introduction

This appendix describes the LCD controller supporting *High Definition* (HD) resolutions.

The HDLCD controller has the following features:

- Resolution:
  - 2048x2048, sufficient for full 1080p HDTV resolution.
- Frame buffer:
  - Supports all common non-indexed RGB formats.
  - Frame buffer can be placed anywhere in memory.
  - Scan lines must be a multiple of 8 bytes long, and aligned to 8-byte boundaries. No other restrictions on size or placement. Line pitch is configurable in multiples of 8 bytes.
- Management:
  - Frame buffer address can be updated at any time, and applies from the next full frame.
  - Frame buffer size, color depth, and timing can only be changed while the display is disabled.
- Maskable interrupts:
  - DMA-end, last part of frame read from bus.
  - VSYNC.
  - Underrun.
  - Bus error.
- Color depths:
  - Supports 8 bits per color. Frame buffers with other color depths are truncated or interpolated to 8 bits per component.
- Interfaces:
  - AMBA 3 APB interface for configuration.
  - Read-only AXI bus for frame buffer reads.
  - Standard LCD external interface. All timings and polarities are configurable.
  - APB, AXI, and pixel clock can run on separate asynchronous clocks.
- Buffering:
  - Internal 2KB buffer.
  - After underrun, it blanks the rest of the frame and resynchronizes from the next frame.

B.2 HDLCD programmers model

This section describes the programmers model. It contains the following subsections:

- *About the HDLCD controller programmers model*
- *Register summary*
- *Register descriptions* on page B-4.

B.2.1 About the HDLCD controller programmers model

The following information applies to the HDLCD controller registers:

- The base address is not fixed, and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in UNPREDICTABLE behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - All register bits are reset to a logic 0 by a system or power-on reset.
- Access type in Table B-1 is described as follows:
  RW Read and write.
  RO Read only.
  WO Write only.

B.2.2 Register summary

Table B-1 shows the registers in offset order from the base memory address. The base memory address of the HDLCD controller on the Cortex-A15 MPCore test chip is `0x00_2B00_0000`.

Table B-1 Register summary

Offset	Name	Type	Reset	Width	Description
`0x0000`	VERSION	RO	VERSION	32	*Version Register* on page B-4
`0x0010`	INT_RAWSTAT	RW	`0x0`	32	*Interrupt Raw Status Register* on page B-5
`0x0014`	INT_CLEAR	WO	N/A	32	*Interrupt Clear Register* on page B-6
`0x0018`	INT_MASK	RW	`0x0`	32	*Interrupt Mask Register* on page B-7
`0x001C`	INT_STATUS	RO	`0x0`	32	*Interrupt Status Register* on page B-8
`0x0100`	FB_BASE	RW	`0x0`	32	*Frame Buffer Base Address Register* on page B-9
`0x0104`	FB_LINE_LENGTH	RW	`0x0`	32	*Frame Buffer Line Length Register* on page B-9
`0x0108`	FB_LINE_COUNT	RW	`0x0`	32	*Frame Buffer Line Count Register* on page B-10
`0x010C`	FB_LINE_PITCH	RW	`0x0`	32	*Frame Buffer Line Pitch Register* on page B-11
`0x0110`	BUS_OPTIONS	RW	`0x408`	32	*Bus Options Register* on page B-11
`0x0200`	V_SYNC	RW	`0x0`	32	*Vertical Synch Width Register* on page B-12
`0x0204`	V_BACK_PORCH	RW	`0x0`	32	*Vertical Back Porch Width Register* on page B-13
`0x0208`	V_DATA	RW	`0x0`	32	*Vertical Data Width Register* on page B-13

Table B-1 Register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x020C	V_FRONT_PORCH	RW	0x0	32	*Vertical Front Porch Width Register* on page B-14
0x0210	H_SYNC	RW	0x0	32	*Horizontal Synch Width Register* on page B-14
0x0214	H_BACK_PORCH	RW	0x0	32	*Horizontal Back Porch Width Register* on page B-15
0x0218	H_DATA	RW	0x0	32	*Horizontal Data Width Register* on page B-15
0x021C	H_FRONT_PORCH	RW	0x0	32	*Horizontal Front Porch Width Register* on page B-16
0x0220	POLARITIES	RW	0x0	32	*Polarities Register* on page B-17
0x0230	COMMAND	RW	0x0	32	*Command Register* on page B-17
0x0240	PIXEL_FORMAT	RW	0x0	32	*Pixel Format Register* on page B-18
0x0244	RED_SELECT	RW	0x0	32	*Color Select Registers* on page B-19
0x0248	GREEN_SELECT	RW	0x0	32	*Color Select Registers* on page B-19
0x024C	BLUE_SELECT	RW	0x0	32	*Color Select Registers* on page B-19

B.2.3 Register descriptions

This section describes the HDLCD controller registers. Table B-1 on page B-3 provides cross references to individual registers.

Version Register

The VERSION Register characteristics are:

Purpose Holds a static version number for the LCD controller. Changes to the processor that affect registers and data structures increment the VERSION_MAJOR value and reset the VERSION_MINOR value.

Other changes that do not affect the binary compatibility only increment the VERSION_MINOR number.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD configurations.

Attributes See Table B-1 on page B-3.

Figure B-1 shows the bit assignments.

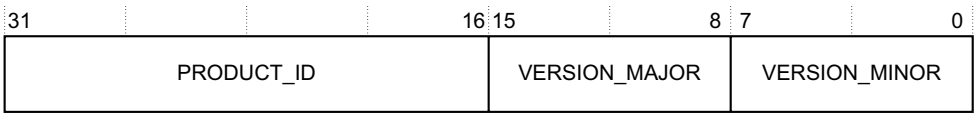


Figure B-1 Version Register bit assignments

Table B-2 shows the bit assignments.

Table B-2 Version Register bit assignments

Bits	Name	Function
[31:16]	PRODUCT_ID	Product ID number `0x1CDC`
[15:8]	VERSION_MAJOR	These bits provide the major product version information. For release r0p0, the value is `0x00`.
[7:0]	VERSION_MINOR	These bits provide the minor product version information. For release r0p0, the value is `0x00`.

Interrupt Raw Status Register

The INT_RAWSTAT Register characteristics are:

Purpose Shows the unmasked status of the interrupt sources. Writing a 1 to the bit of an interrupt source forces this bit to be set and generate an interrupt if it is not masked by the corresponding bit in the *Interrupt Mask Register* on page B-7. Writing a 0 to the bit of an interrupt source has no effect.

Use the *Interrupt Clear Register* on page B-6 to clear interrupts.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-2 shows the bit assignments.

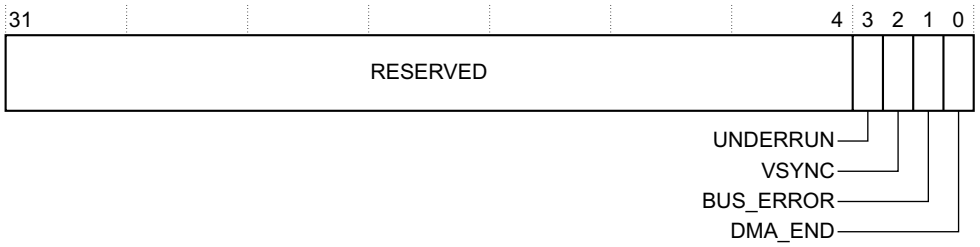


Figure B-2 Interrupt Raw Status Register bit assignments

Table B-3 shows the bit assignments.

Table B-3 Interrupt Raw Status Register bit assignments

Bits	Name	Function
[31:4]	-	Reserved, write as zero, read undefined.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.

Table B-3 Interrupt Raw Status Register bit assignments (continued)

Bits	Name	Function
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Clear Register

The INT_CLEAR Register characteristics are:

Purpose Clears interrupt sources. Writing a 1 to the bit of an asserted source clears the interrupt in the *Interrupt Raw Status Register* on page B-5, and in the *Interrupt Status Register* on page B-8, if it is not masked.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-3 shows the bit assignments.

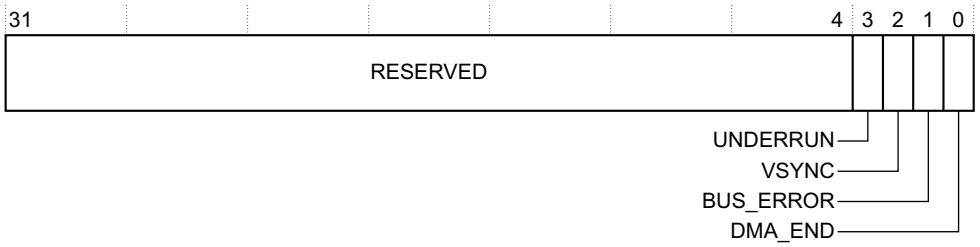


Figure B-3 Interrupt Clear Register bit assignments

Table B-4 shows the bit assignments.

Table B-4 Interrupt Clear Register bit assignments

Bits	Name	Function
[31:4]	-	Reserved, write as zero.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.

Table B-4 Interrupt Clear Register bit assignments (continued)

Bits	Name	Function
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Mask Register

The INT_MASK Register characteristics are:

Purpose Holds the bit mask that enables an interrupt source if the corresponding mask bit is set to 1.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-4 shows the bit assignments.

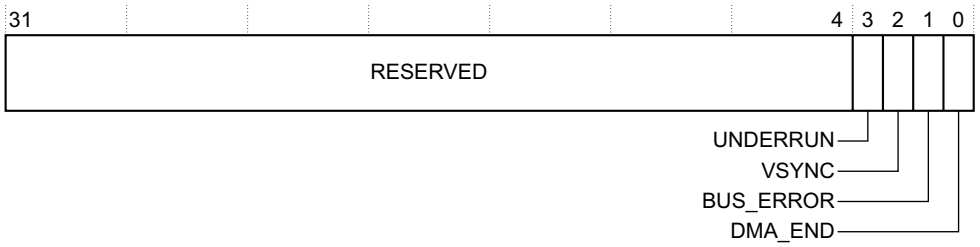


Figure B-4 Interrupt Mask Register bit assignments

Table B-5 shows the bit assignments.

Table B-5 Interrupt Mask Register bit assignments

Bits	Name	Function
[31:4]	-	Reserved, write as zero, read undefined.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.

Table B-5 Interrupt Mask Register bit assignments (continued)

Bits	Name	Function
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Status Register

The INT_STATUS Register characteristics are:

Purpose *Interrupt Raw Status Register* on page B-5 ANDed with the *Interrupt Mask Register* on page B-7 and shows the active and masked interrupt sources. Bits selected by the Interrupt Mask Register are active in the *Interrupt Status Register*. These bits show the status of the interrupt sources. Bits not selected by the Interrupt Mask Register are inactive. If any of the sources are asserted in the Interrupt status Register, then the external **IRQ** line is asserted.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-5 shows the bit assignments.

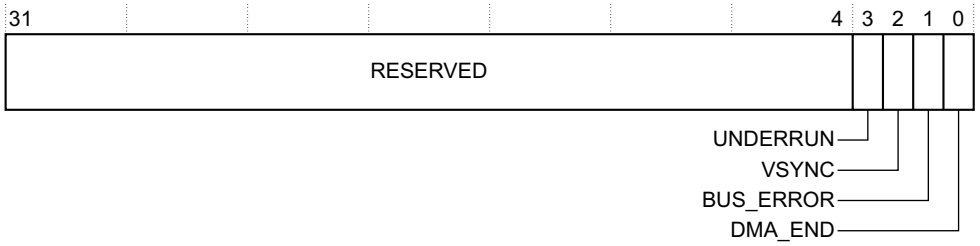


Figure B-5 Interrupt Status Register bit assignments

Table B-6 shows the bit assignments.

Table B-6 Interrupt Status Register bit assignments

Bits	Name	Function
[31:4]	-	Reserved, read undefined.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Frame Buffer Base Address Register

The FB_BASE Register characteristics are:

Purpose Holds the address of the first pixel of the first line in the frame buffer.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-6 shows the bit assignments.

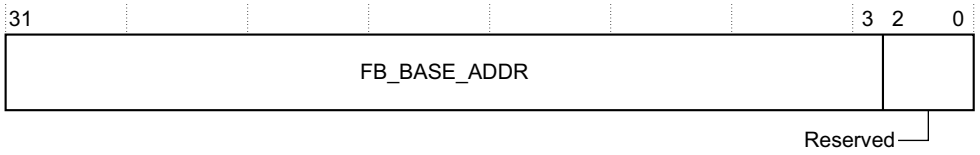


Figure B-6 Frame Buffer Base Address Register bit assignments

Table B-7 shows the bit assignments.

Table B-7 Frame Buffer Base Address Register bit assignments

Bits	Name	Function
[31:3]	FB_BASE_ADDR	Frame buffer base address
[2:0]	Reserved	Reserved, write as zero, read undefined

Frame Buffer Line Length Register

The FB_LINE_LENGTH Register characteristics are:

Purpose Holds the length of each frame buffer line in bytes.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-7 shows the bit assignments.

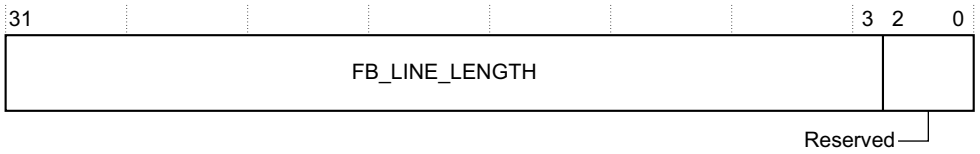


Figure B-7 Frame Buffer Line Length Register bit assignments

Table B-8 shows the bit assignments.

Table B-8 Frame Buffer Line Length Register bit assignments

Bits	Name	Function
[31:3]	FB_LINE_LENGTH	Frame buffer line length
[2:0]	-	Reserved, write as zero, read undefined

Frame Buffer Line Count Register

The FB_LINE_COUNT Register characteristics are:

Purpose Holds the number of lines to read from the frame buffer.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-8 shows the bit assignments.

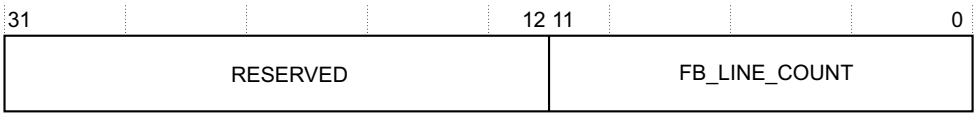


Figure B-8 Frame Buffer Line Count Register bit assignments

Table B-9 shows the bit assignments.

Table B-9 Frame Buffer Line Count Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	FB_LINE_COUNT	Frame buffer line count

Frame Buffer Line Pitch Register

The FB_LINE_PITCH Register characteristics are:

Purpose Holds the number of bytes between the start of one line in the frame buffer and the start of the next line. This value is treated as a signed 2's complement number, enabling negative pitch if required.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-9 shows the bit assignments.

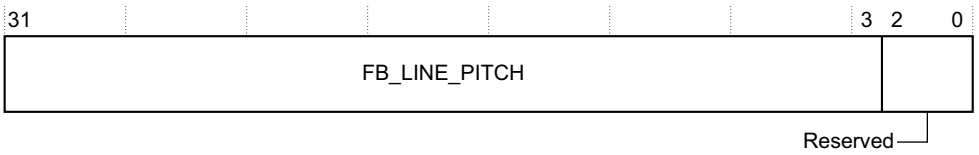


Figure B-9 Frame Buffer Line Count Register bit assignments

Table B-10 shows the bit assignments.

Table B-10 Frame Buffer Line Count Register bit assignments

Bits	Name	Function
[31:3]	FB_LINE_PITCH	Frame buffer line pitch
[2:0]	-	Reserved, write as zero, read undefined

Bus Options Register

The BUS_OPTIONS Register characteristics are:

Purpose Controls aspects of how the LCD controller accesses the bus. This value can be tuned to better match the characteristics of the memory controller and other units in the system.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-10 on page B-12 shows the bit assignments.

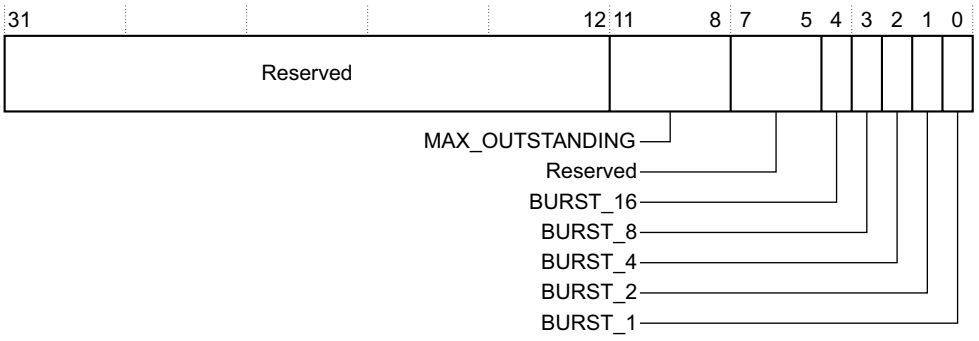


Figure B-10 Bus Options Register bit assignments

Table B-11 shows the bit assignments.

Table B-11 Bus Options Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined.
[11:8]	MAX_OUTSTANDING	Maximum number of outstanding requests the LCD controller is permitted to have on the bus at any time. **Caution** A value of zero disables all bus transfers.
[7:5]	-	Reserved, write as zero, read undefined.
[4]	BURST_16	Permit the use of 16-beat bursts.
[3]	BURST_8	Permit the use of 8-beat bursts.
[2]	BURST_4	Permit the use of 4-beat bursts.
[1]	BURST_2	Permit the use of 2-beat bursts.
[0]	BURST_1	Permit the use of 1-beat bursts.

Note

- If the scan line length does not end up at a multiple of the permitted burst lengths, the controller uses smaller bursts to read the remaining few pixels in each scan line. If no bursts are permitted, this mechanism also triggers, and has the same effect as permitting all bursts.
- Incorrectly configuring this register can degrade the performance of both the LCD controller and the rest of the system.

Vertical Synch Width Register

The V_SYNC Register characteristics are:

Purpose Holds the width of the vertical synch signal, counted in number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-11 shows the bit assignments.

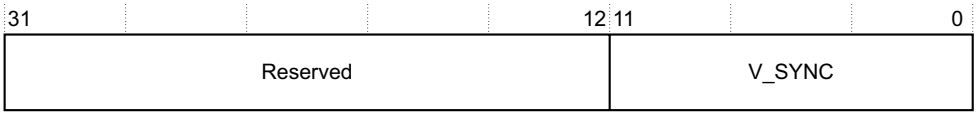


Figure B-11 Vertical Synch Width Register bit assignments

Table B-12 shows the bit assignments.

Table B-12 Vertical Synch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	V_SYNC	Vertical synch width -1

Vertical Back Porch Width Register

The V_BACK_PORCH Register characteristics are:

Purpose Holds the width of the interval between the vertical sync and the first visible line, counted in the number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-12 shows the bit assignments.

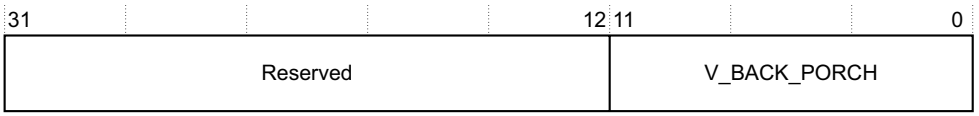


Figure B-12 Vertical Back Porch Width Register bit assignments

Table B-13 shows the bit assignments.

Table B-13 Vertical Back Porch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	V_BACK_PORCH	Vertical back porch width -1

Vertical Data Width Register

The V_DATA Register characteristics are:

Purpose Holds the width of the vertical data area, that is, the number of visible lines, counted in the number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-13 shows the bit assignments.

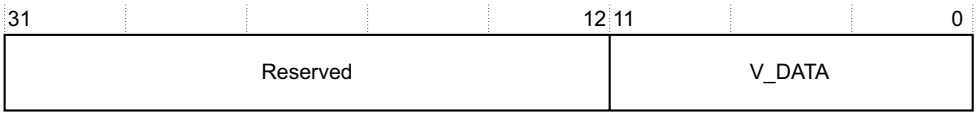


Figure B-13 Vertical Data Width Register bit assignments

Table B-14 shows the bit assignments.

Table B-14 Vertical Data Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	V_DATA	Vertical data width -1

Vertical Front Porch Width Register

The V_FRONT_PORCH Register characteristics are:

Purpose Holds the width of the interval between the last visible line and the next vertical synchronization, counted in the number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-14 shows the bit assignments.

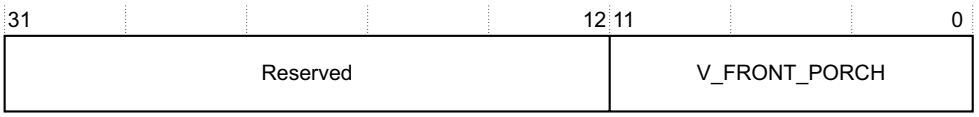


Figure B-14 Vertical Front Porch Width Register bit assignments

Table B-15 shows the bit assignments.

Table B-15 Vertical Front Porch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	V_FRONT_PORCH	Vertical front porch width -1

Horizontal Synch Width Register

The H_SYNCH Register characteristics are:

Purpose Holds the width of the horizontal synch signal, counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-15 shows the bit assignments.

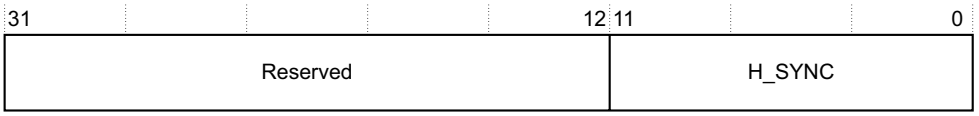


Figure B-15 Horizontal Synch Width Register bit assignments

Table B-16 shows the bit assignments.

Table B-16 Horizontal Synch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	H_SYNC	Horizontal synch width -1

Horizontal Back Porch Width Register

The H_BACK_PORCH Register characteristics are:

Purpose Holds the width of the interval between the horizontal sync and the first visible column, counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-16 shows the bit assignments.

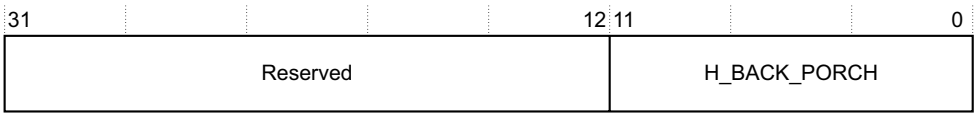


Figure B-16 Horizontal Back Porch Width Register bit assignments

Table B-17 shows the bit assignments.

Table B-17 Horizontal Back Porch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	H_BACK_PORCH	Horizontal back porch width -1

Horizontal Data Width Register

The H_DATA Register characteristics are:

Purpose Holds the width of the horizontal data area, that is, the number of visible columns counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-17 shows the bit assignments.

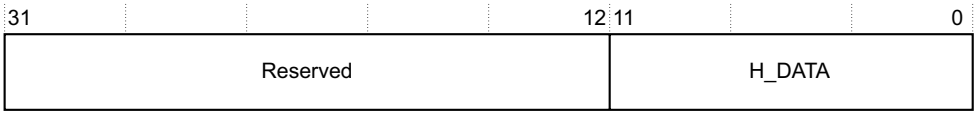


Figure B-17 Horizontal Data Width Register bit assignments

Table B-18 shows the bit assignments.

Table B-18 Horizontal Data Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	H_DATA	Horizontal data width -1

Horizontal Front Porch Width Register

The H_FRONT_PORCH Register characteristics are:

Purpose Holds the width of the interval between the last visible column and the next horizontal synchronization, counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-18 shows the bit assignments.

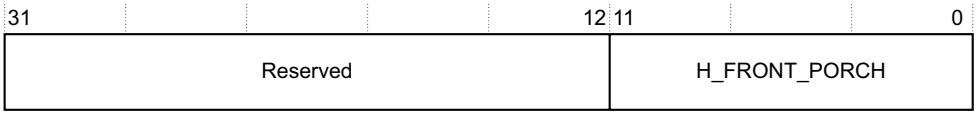


Figure B-18 Horizontal Front Porch Width Register bit assignments

Table B-19 shows the bit assignments.

Table B-19 Horizontal Front Porch Width Register bit assignments

Bits	Name	Function
[31:12]	-	Reserved, write as zero, read undefined
[11:0]	H_FRONT_PORCH	Horizontal front porch width -1

Polarities Register

The POLARITIES Register characteristics are:

Purpose Controls the polarities of the synchronization signals and **PXLCLK** that is exported from the CoreTile Express A15×2 daughterboard.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-19 shows the bit assignments.

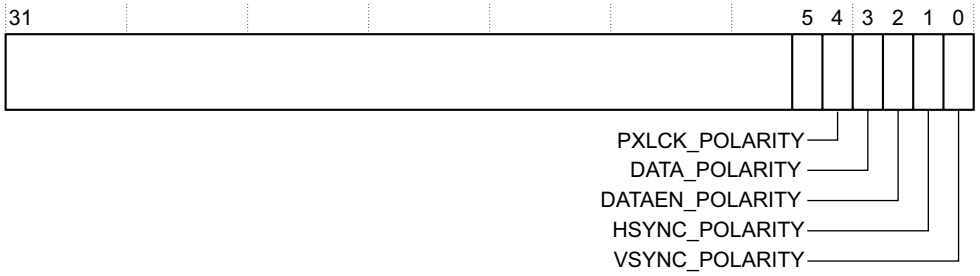


Figure B-19 Polarities Register bit assignments

Table B-20 shows the bit assignments.

Table B-20 Polarities Register bit assignments

Bits	Name	Function
[31:5]	-	Reserved, write as zero, read undefined.
[4]	PXLCLK_POLARITY	Holds the value of the PXLCLKPOL output. This is intended to be used to control the polarity of the pixel clock that is exported from the CoreTile Express A15×2 daughterboard. **Note** **PXLCLK_POLARITY=b1;** **PXLCLK** (HDLCD) and **MMB_IDCLK**, export, are the same polarity. **PXLCLK_POLARITY=b0;** **PXLCLK** (HDLCD) and **MMB_IDCLK**, export, are the opposite polarity.
[3]	DATA_POLARITY	Holds the active level of the DATA output.
[2]	DATAEN_POLARITY	Holds the active level of the DATAEN output.
[1]	HSYNC_POLARITY	Holds the active level of the HSYNC output.
[0]	VSYNC_POLARITY	Holds the active level of the VSYNC output.

Command Register

The COMMAND Register characteristics are:

Purpose Starts and stops the LCD controller.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-20 shows the bit assignments.

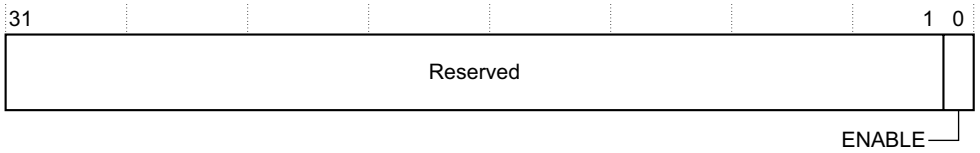


Figure B-20 Command Register bit assignments

Table B-21 shows the bit assignments.

Table B-21 Command Register bit assignments

Bits	Name	Function
[31:1]	-	Reserved, write as zero, read undefined
[0]	ENABLE	Enable the LCD controller

Pixel Format Register

The PIXEL_FORMAT Register characteristics are:

Purpose BYTES_PER_PIXEL plus one bytes are extracted from the internal buffer. The extracted bytes are used to form a 32-bit value. The individual bytes are then optionally reordered if BIG_ENDIAN is set before the color components are extracted.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-21 shows the little endian byte layout.

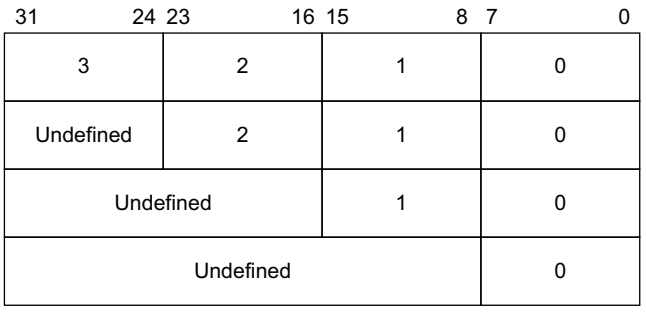


Figure B-21 Little endian byte layout

Figure B-22 on page B-19 shows the big endian byte layout.

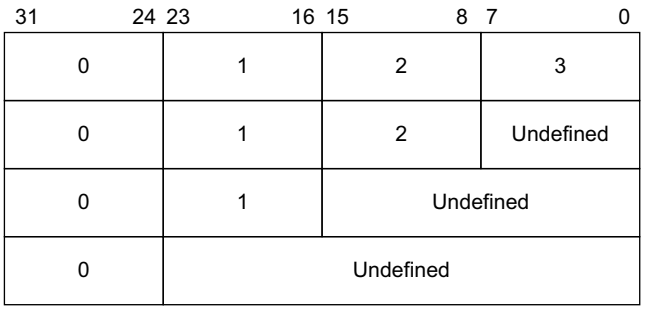


Figure B-22 Big endian byte layout

Figure B-23 shows the bit assignments for the big endian format.

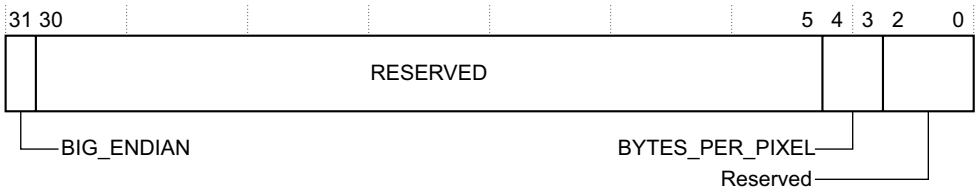


Figure B-23 Pixel Format Register bit assignments

Table B-22 shows the bit assignments for the big endian format.

Table B-22 Pixel Format Register bit assignments

Bits	Name	Function
[31]	BIG_ENDIAN	Use big endian byte order
[30:5]	-	Reserved, write as zero, read undefined
[4:3]	BYTES_PER_PIXEL	Number of bytes to extract from the buffer for each pixel to display, minus one
[2:0]	-	Reserved, write as zero, read undefined

Color Select Registers

The RED_SELECT, GREEN_SELECT, and BLUE_SELECT Registers characteristics are:

Purpose The bytes extracted from the internal buffer are presented as a 32-bit value. These registers select how many bits, and at which position, are used to extract and use as the red, green, and blue color components. If no bits are extracted, or no data is available, the default color is used.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table B-1 on page B-3.

Figure B-24 on page B-20 shows the bit assignments.

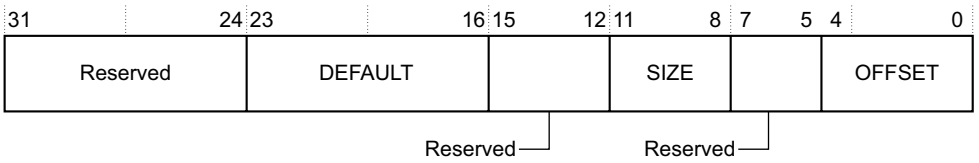


Figure B-24 Color Select Register bit assignments

Table B-23 shows the bit assignments.

Table B-23 Color Select Register bit assignments

Bits	Name	Function
[31:24]	-	Reserved, write as zero, read undefined.
[23:16]	DEFAULT	Default color. This color is used if SIZE is zero, if the buffer underruns, or while outside the visible frame area.
[15:12]	-	Reserved, write as zero, read undefined.
[11:8]	SIZE	Number of bits to extract: If this value is zero, the default color is used. If this value is in the range 1-7, the extracted MSBs are repeated for the LSBs until eight bits are reached. If this value is larger than eight, the behavior is UNDEFINED.
[7:5]	-	Reserved, write as zero, read undefined.
[4:0]	OFFSET	Index of the lowest bit to extract. If OFFSET + SIZE >= 8 + 8 x BYTES_PER_PIXEL, the behavior is UNDEFINED.

Appendix C
Electrical Specifications

This appendix contains the electrical specification for the daughterboard. It contains the following section:

- *AC characteristics*.

C.1 AC characteristics

Table C-1 shows the recommended AC operating characteristics for the Cortex-A15 MPCore test chip.

For more information on each interface that Table C-1 describes, see the appropriate technical reference manual that *Additional reading* on page x lists.

Table C-1 AC characteristics

Interface	Parameter	Symbol	Minimum	Maximum	Description
Multiplexed Master AXI port	Clock cycle	t_{SPcyc}	25ns	-	C_{max}=47.8pF C_{min}=23.88pF
	Output valid time before clock edge	t_{SPov}	-	6.35ns	
	Output hold time after clock edge	t_{SPoh}	2.53ns		
	Input setup time to clock edge	t_{SPis}	-	10.4ns	
	Input hold time after clock edge	t_{SPih}	6.38ns	-	
Trace	Clock cycle	$t_{TRACEcyc}$	7.14ns	-	C_{max}=22.5pF C_{min}=16.5pF
JTAG	Clock cycle	$t_{JTAGcyc}$	40ns	-	C_{max}=48.5pF C_{min}=10.5pF
	Output valid time before clock rising edge	t_{JTAGov}	-	13.35ns	
	Output hold time after clock rising edge	t_{JTAGoh}	4.9ns	-	
	Input setup time to clock rising edge	t_{JTAGis}	-	34.45ns	
	Input hold time after clock rising edge	t_{JTAGih}	2.8ns	-	
SMC	Clock cycle	t_{SMCcyc}	12.3ns	-	C_{max}=47.8pF C_{min}=23.88pF
	Output valid time before clock rising edge	t_{SMCov}	-	14ns	
	Input setup time to clock edge	t_{SMCis}		10ns	
HDLCD (MMB_IDCLK)	Clock cycle	$t_{HDLCDcyc}$	6.06ns	-	C_{max}=47.8pF C_{min}=23.88pF
	Output hold time after clock rising edge	$t_{HDLCDoh}$	-	2.5ns	
	Input setup time to clock edge	$t_{HDLCDis}$		0.5ns	

Appendix D
Revisions

This appendix describes the technical changes between released issues of this book.

Table D-1 Issue A

Change	Location	Affects
First release	-	-

Table D-2 Differences between issue A and issue B

Change	Location	Affects
Updated daughterboard memory map figure. Updated daughterboard peripheral memory map table.	Figure 3-1 on page 3-4 and Table 3-1 on page 3-4.	All revisions
Updated table title.	Table 3-4 on page 3-9.	All revisions
Added new table to document.	Table 3-2 on page 3-8.	All revisions
Updated default value of HSBM (CLK).	Table 2-8 on page 2-24.	All revisions
Shortened configuration chapter. Information is now in a new document called *Versatile™ Express Configuration Technical Reference Manual.* See *Additional reading* on page x. Added information about custom motherboard.	*Configuration architecture* on page 2-10.	All revisions

Table D-2 Differences between issue A and issue B (continued)

Change	Location	Affects
Added new documents to *Additional reading* on page x: • *Versatile™ Express Configuration Technical Reference Manual* • *LogicTile Express 13MG Technical Reference Manual*.	*Additional reading* on page x.	All revisions
Updated PLL settings register description.	Figure 3-14 on page 3-27 and Table 3-16 on page 3-27.	All revisions
Added Energy Meter section.	*Energy meter* on page 2-17.	All revisions
Replaced PL310 L2 cache controller with integrated L2 cache controller in the Cortex-A15 MPCore cluster.	Figure 2-2 on page 2-4.	All revisions
Updated description of Cortex-A15 L2 controller.	*Cortex-A15 L2 cache controller* on page 3-36.	All revisions

Table D-3 Differences between issue B and issue C

Change	Location	Affects
Updated address offset range for CPU 0 PTM.	Table 3-4 on page 3-9.	All revisions

Table D-4 Differences between issue C and issue D

Change	Location	Affects
Updated interrupt table.	Table 2-10 on page 2-30.	All revisions

Table D-5 Differences between issue D and issue E

Change	Location	Affects
Updated introduction to daughterboard memory map section.	*Daughterboard memory map* on page 3-3.	All revisions
Corrected typographical error in daughterboard memory map.	Table 3-1 on page 3-4.	All revisions

Table D-6 Differences between issue E and issue F

Change	Location	Affects
Corrected memory test chip peripheral areas of memory maps.	*Overview of daughterboard memory map* on page 3-3. *Overview of the memory map for the on-chip peripherals* on page 3-6.	All revisions