LogiCORE IP AXI Chip2Chip v4.1

Product Guide For Vivado Design Suite

PG067 December 18, 2013

XILINX®

Table of Contents

Introduction

The LogiCORE™ IP AXI Chip2Chip is a soft Xilinx IP core for use with the Vivado Design Suite. The adaptable block provides bridging between AXI systems for multi-device System on Chip solutions. The core supports multiple device-to-device interfacing options and provides a low pin count, high performance AXI chip-to-chip bridging solution.

Features

- Supports AXI4 Memory Map interface data width of 32 and 64 bits
- Supports optional AXI4-Lite data width of 32 bits
- Supports Single Ended or Differential SelectIO™ FPGA interface and Aurora FPGA interface
- Independent Master or Slave mode selection for AXI4 and AXI4-Lite interfaces
- Supports Common Clock or Independent Clock operations
- Supports multiple Width Conversion options for reduced I/O utilization
- Supports Link Detect FSM with deskew operation
- Allows all five AXI4 channels to operate independently
- Supports an additional high-priority cut through channel for communicating interrupts
- Provides a dedicated high-priority internal channel for link status monitoring and reporting
- Generates Link Error and Multi-bit Error interrupts
- Supports ECC with single-bit error correction for Aurora interface

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq®-7000, 7 Series
Supported User Interfaces	AXI4, AXI4-Lite
Resources	See Table 2-2.
Provided with Core	
Design Files	Verilog and VHDL
Example Design	Verilog
Test Bench	Verilog
Constraints File	XDC
Simulation Model	Not Provided
Supported S/W Driver	N/A
Tested Design Flows(2)	
Design Entry	Vivado Design Suite IP Integrator
Simulation	For support simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete list of supported devices, see Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

The LogiCORE™ IP AXI Chip2Chip core functions like a bridge to seamlessly connect two devices over an AXI interface. The core transparently bridges transactions in compliance with AXI protocol specifications. The bridging function allows all AXI channels to operate independently by forwarding per-channel data and control information in compliance with AXI per-channel Valid-Ready handshake.

The optional AXI4-Lite functions in the core are independent of AXI4 Interface functions. The AXI4-Lite interface can be used for low-bandwidth access such as configuration/status registers of a peripheral Master/Slave.

Two instances of the AXI Chip2Chip core are required for performing the bridging function. Figure 1-1 provides block diagrams for both Master and Slave AXI Chip2Chip IP cores.

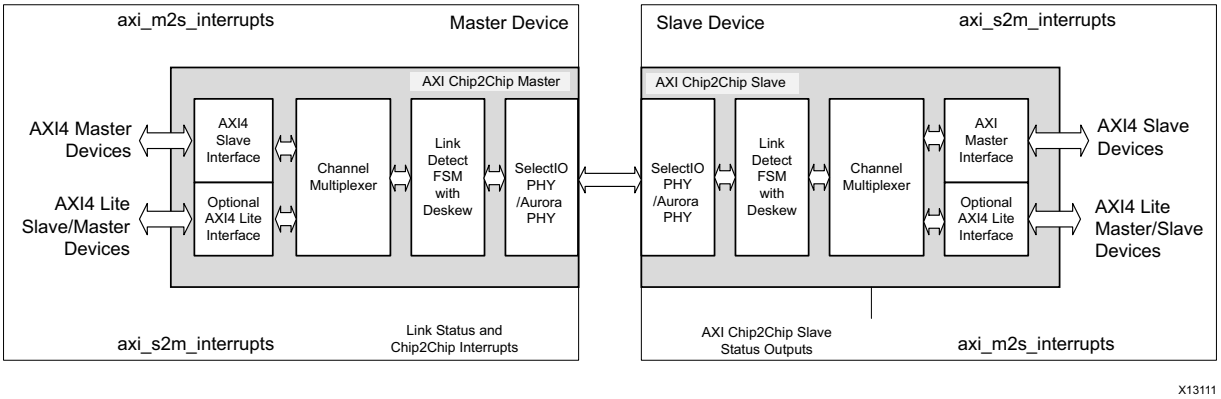

Figure 1-1: **AXI Chip2Chip Block Diagram**

The AXI Chip2Chip Master instance provides an AXI4 slave interface that can be directly connected to AXI Master or AXI interconnect devices. The AXI Chip2Chip Slave instance provides an AXI4 Master interface that can be connected to AXI Slave or AXI interconnect devices. The bridging functions in AXI Chip2Chip cores convert the wide on-chip AXI signaling to a compact device-to-device interfacing by utilizing a minimum set of FPGA I/Os. The AXI Chip2Chip bridging also implements functions that provide error-free communication over the FPGA I/Os.

The AXI4-Lite configuration option allows master or slave mode selection. For example, when the processor is connected to an AXI Chip2Chip Master instance, then an AXI4-Lite instance can be set to master mode; this setup will provide an AXI4-Lite slave interface. When peripheral Masters are connected to an AXI Chip2Chip Master instance, then an

AXI4-Lite instance can be set to slave mode and it will provide an AXI4-lite master interface. For more details on AXI4-Lite configuration options, see User Tab in Chapter 4.

AXI Chip2Chip operations can be categorized into five modules: AXI4 Interface, AXI4-Lite Interface, Channel Multiplexer, Link Detect FSM, and PHY interface.

Feature Summary

This section summarizes the functionality of the five modules.

AXI4 Interface

The AXI Chip2Chip core provides an AXI4 interface to map to AXI Memory Mapped devices in the device fabric. AXI Memory Mapped devices can be AXI Master, AXI interconnect, or AXI Slave functions. The AXI Interface can operate in either Common Clock or in Independent clock modes. For more details on clocking and latencies, see General Design Guidelines in Chapter 3.

TIP: *The AXI4 interface of the Chip2Chip core provides WUSER signals to maintain compatibility with AXI3 interface specifications. Any AXI3 master that supports write interleaving can use the AXI4 WUSER[3:0] signals to map the WID[3:0] signals in AXI3 write data channel.*

AXI4-Lite Interface

The AXI4-Lite functions in the core are implemented with a shared address and data bus approach. This allows AXI4-Lite Master to accept a new write transaction only on completion of previous write transaction. This means it only accepts new writes on receiving a write response from the AXI4-Lite Slave. Similarly, the AXI4-Lite Master accepts a new read transaction only on completion of previous read transaction. This means it only accepts reads after receiving a read response and data from the AXI4-Lite Slave. For more details on the AXI4 Lite clocking and reset, see General Design Guidelines in Chapter 3.

Channel Multiplexer

The Channel Multiplexer multiplexes AXI Address and Data channels over FPGA I/Os. In addition, the AXI Chip2Chip core internally determines a 2:1 or 4:1 width conversion based on the Chip2Chip PHY Width option selected for the cores. Width conversion is used for reduced I/O utilization between the two devices. For more details on width conversion, see User Tab in Chapter 4.

The Channel Multiplexer also multiplexes AXI, AXI4-Lite and interrupt interfaces over the same set of FPGA I/Os. The priority round-robin multiplexing in the Chip2Chip core assigns the highest priority to interrupt signals, second highest priority to the low-bandwidth

AXI4-Lite interface, and last priority to the AXI interface. The priority round-robin multiplexing is in effect when more than one of these interfaces are active simultaneously.

SelectIO Link Detect FSM with Deskew

The SelectIO Link Detect FSM with deskew operation ensures that the AXI Chip2Chip Master core initiates transactions only when both Master and Slave cores are out of reset and deskew patterns are exchanged without any bit errors. Deskew operations align data until an optimized sampling point is determined for the data. The nibble level deskew operation also enhances the maximum frequency of operation for the SelectIO™ interface. For more details on Link Detect FSM and deskew operations, see Calibration and Link Error Detection in Chapter 3.

Aurora Link Detect FSM

The Link Detect FSM ensures that the AXI Chip2Chip Master core initiates transactions only after the Aurora channel initialization is complete. For more details on Aurora Link Detect FSM, see Auto-Negotiation in Chapter 3.

SelectIO PHY Interface

The AXI Chip2Chip core provides the SelectIO FPGA interface as an interfacing option between the devices. The SelectIO provides minimum latency between the devices and provides SDR or DDR operations. When the SelectIO interface is used, the I/O type and I/O location must be specified in the Xilinx Design Constraints file (XDC).

Aurora PHY Interface

AXI Chip2Chip provides an optional AXI4-Stream interface to connect to the Aurora 64B66B core. Aurora provides a high-speed serial communication. The ECC module performs bit error correction for single-bit error detected in the received data. When multiple bits are detected with errors, the Multibit Error interrupt signal is asserted.

Interrupt Signals

The AXI Chip2Chip core allows level interrupts to be communicated through a high-priority internal channel. Interrupts can be independently communicated between AXI Masters and AXI Slaves. On detecting a value change in the interrupt inputs, the AXI Chip2Chip Master core initiates a high-priority transfer to update the interrupt outputs of the AXI Chip2Chip Slave core. Similarly, on detecting a value change in the interrupt inputs, the AXI Chip2Chip Slave core initiates a high-priority transfer to update the interrupt outputs of the AXI Chip2Chip Master core.

The AXI Chip2Chip Master core also generates interrupts for link error conditions. Interrupt signals are asserted by the AXI Chip2Chip Master core. For this, the error conditions

detected in the AXI Chip2Chip Slave core are communicated to the Master device through a high-priority internal channel.

The following interrupt signals are supported in the AXI Chip2Chip Master core:

- **Link Error Interrupt**: Asserted when the AXI Chip2Chip Slave core is reset during normal operation. For more details on Link Error Interrupt, see Resets in Chapter 3.
- **Multibit Error Interrupt**: When asserted, a Multibit Error interrupt indicates multiple bits are received in error in the Master or Slave AXI Chip2Chip core. For the SelectIO interface, a multibit error is determined during deskew operations and indicates deskew operation failure. Multibit Error for the Aurora interface is determined by the ECC function.
- **Configuration Error Interrupt**: The Configuration Error Interrupt is set when a mismatch is detected between the Master and Slave AXI Chip2Chip core configurations.

After being asserted, interrupt flags can be cleared only with a reset.

Applications

Figure 1-2 shows an example of the AXI Chip2Chip use case with SelectIO PHY.

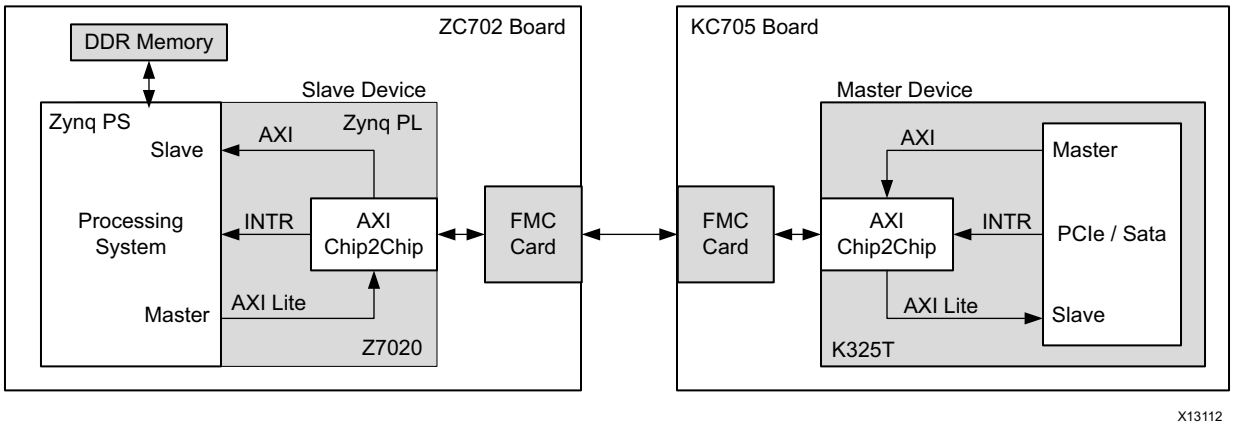

Figure 1-2: **AXI Chip2Chip Core Application Diagram**

In this use case, a Kintex™-7 device implementing a PCIe peripheral Master is connected to a Zynq™-7020 device over an AXI Memory Mapped interface. Because it implements the Peripheral Master on the Chip2Chip AXI interface, the Kintex-7 device is the Master Device. Because it implements an AXI DDR Memory slave, the Zynq-7020 device is the Slave device. In this use case, the processing subsystem in the Zynq-7020 device uses the AXI4-Lite interface of the Chip2Chip core to access the control and status registers of the Peripheral Master in the Kintex-7 device. The PCIe Peripheral Master uses the AXI interface of the Chip2Chip core for writing and reading data from the DDR memory connected to the

Zynq-7020 device. The PCIe Master in this case uses the Chip2Chip core interrupt signaling to trigger any PCIe interrupt service routines in the host processor.

Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado Design Suite under the terms of the Xilinx End User License.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards Compliance

This core has bus interfaces that comply with the *ARM® AMBA® AXI4 Protocol Specification Version 1.0*.

Performance

Maximum Frequencies

The AXI Chip2Chip core operates up to 200 MHz on Artix-7 devices with -1 and higher speed grades.

Latency

Table 2-1 lists the latencies and performance measurements on the AXI4 interface of the Chip2Chip Master core with SelectIO interface. The measurements were taken with simultaneous read and write operations. The measurement setup issued up to four AXI4 outstanding transactions. The AXI (system) clock frequency was set to 100 MHz, and ALEN was set to 16 beats. The measured latency can have up to 5-10% variation and does not account for system latencies outside of the AXI Chip2Chip core.

Table 2-1: **Latency for AXI4 Interface of the AXI Chip2Chip Master Core**

Features			Latencies (AXI Clocks)		Performance (Mb/s)	
AXI Data Width	Number of I/Os	PHY Clock / PHY Type(1)(2)	AW_Valid to B_Valid	AR_Valid to R_Valid	Write Data Channel	Read Data Channel
32-bit	38	200 MHz / SDR	77	44	1190	1280
	32	150 MHz / DDR	69	47	1770	1920
	58	100 MHz / DDR	57	42	2350	2550

Table 2-1: **Latency for AXI4 Interface of the AXI Chip2Chip Master Core *(Cont'd)***

Features			Latencies (AXI Clocks)		Performance (Mb/s)	
AXI Data Width	Number of I/Os	PHY Clock / PHY Type[(1)(2)]	AW_Valid to B_Valid	AR_Valid to R_Valid	Write Data Channel	Read Data Channel
64-bit	28	150 MHz / DDR	95	59	1780	1920
	46	100 MHz / DDR	77	51	2370	2560

Notes:

1. Common Clock mode of operation was selected for configurations having the same PHY clock and AXI clock frequencies (100 MHz).
2. The number of I/Os is determined by the PHY Type and PHY Width configurations. See Table 4-2, page 33 for more details.

Resource Utilization

Table 2-2 shows the resource utilization numbers for the AXI Chip2Chip core with a SelectIO PHY interface and Table 2-3 shows resource utilization when using an Aurora interface. These values have been generated using the Xilinx Vivado Design Suite. The values are derived from actual hardware validation systems.

Table 2-2: **Resource Utilization for the AXI Chip2Chip Core with a SelectIO Interface**

Family	Feature			AXI Chip2Chip Master Core			AXI Chip2Chip Slave Core		
	AXI Data Width	Number of I/Os	Clocking	Slice LUTs	Slice Registers	Block RAMs	Slice LUTs	Slice Registers	Block RAMs
Virtex-7	32	38	Independent	919	1507	4	846	1524	4
	64	46	Common	1197	1608	5	1092	1417	5
Artix-7	32	38	Independent	932	1507	4	860	1524	4
	64	46	Common	1252	1608	5	1147	1417	5
Kintex-7	32	38	Independent	919	1507	4	846	1524	4
	64	46	Common	1197	1608	5	1092	1417	5

Table 2-3: **Resource Utilization for the Chip2Chip with Aurora (Kintex-7 FPGAs)**

Features			AXI Chip2Chip Master Core + Aurora 64B66B Duplex Core			AXI Chip2Chip Slave Core + Aurora 64B66B Duplex Core		
AXI Data Width	Aurora Width	Clocking	LUTs	FFs	Block RAMs	LUTs	FFs	Block RAMs
32	64	Independent	1637	2833	5	1635	2831	5
64	64	Independent	1804	2943	6	1794	2943	6

Resource usage for the AXI4-Lite mode is shown in Table 2-4.

Table 2-4: **Resource Utilization for AXI4-Lite Mode**

Family	Feature			AXI Chip2Chip Master Core			AXI Chip2Chip Slave Core		
	AXI Data Width	Number of I/Os	Clocking	Slice LUTs	Slice Registers	Block RAMs	Slice LUTs	Slice Registers	Block RAMs
Virtex-7	32	32	Independent	283	436	0	320	562	0
Artix-7	32	32	Independent	283	436	0	320	562	0
Kintex-7	32	32	Independent	283	436	0	320	562	0

Port Descriptions

This section provides port descriptions for the AXI Chip2Chip core. It is organized according to Master and Slave ports.

AXI Chip2Chip Master

This section describes the ports when the AXI Chip2Chip core is a Master. It contains these subsections:

- Global Signals
- AXI Interface Signals
- Master Device Interface Signals
- Interrupt and Status Signals

Global Signals

Table 2-5 describes the global signals for the AXI Chip2Chip Master core.

Table 2-5: **Global Interface Signals**

Name	Direction	Description
s_aclk	Input	Global Slave Interface Clock. For Independent Clock mode, all signals on the AXI Slave interface of an AXI Chip2Chip Master core are synchronous to s_aclk. For Common Clock mode, all AXI Chip2Chip Master core operations are synchronous to s_aclk.
axi_c2c_phy_clk	Input	Physical Interface Clock. The axi_c2c_phy_clk signal is applicable only when Independent Mode operation is selected for the core. AXI Chip2Chip Master core operations excluding the AXI Slave Interface are synchronous to axi_c2c_phy_clk. For the Aurora interface, axi_c2c_phy_clk needs to be provided by the Aurora PHY.

Table 2-5: **Global Interface Signals *(Cont'd)***

Name	Direction	Description
idelay_ref_clk	Input	SelectIO Interface I/O Reference Clock. This signal is applicable only when the SelectIO interface is selected as the FPGA interfacing option and when deskew is enabled for the SelectIO interface. The applicable frequency for idelay_ref_clk is 200 MHz or 300 MHz (±10 MHz).
s_aresetn	Input	Global Reset. This signal is active-Low and synchronous. All applicable clock inputs to the AXI Chip2Chip Master core must be stable when s_aresetn input is de-asserted.
Optional AXI4-Lite Signals		
m_axi_lite_aclk	Input	Master Interface AXI4-Lite Clock. Applicable only when Slave Mode of AXI4-Lite is selected. All signals are sampled on the rising edge of this clock.
s_axi_lite_aclk	Input	Slave Interface AXI4-Lite Clock. Applicable only when Master Mode of AXI-Lite is selected. All signals are sampled on the rising edge of this clock.

AXI Interface Signals

Table 2-6 describes the AXI Interface signals for the AXI Chip2Chip Master core.

Table 2-6: **AXI Interface Signals**

Name	Direction	Description
Write Address Channel		
s_axi_awid[C_AXI_ID_WIDTH-1:0]	Input	Write address ID. This signal is the identification tag for the write address group of signals. ID width can be configured from 1 to 6 bits.
s_axi_awaddr[31:0]	Input	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
s_axi_awlen[7:0]	Input	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
s_axi_awsize[1:0]	Input	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.
s_axi_awburst[1:0]	Input	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.

Table 2-6: **AXI Interface Signals *(Cont'd)***

Name	Direction	Description
s_axi_awvalid	Input	Write address valid. This signal indicates that valid write address and control information are available: 1 = address and control information available. 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes HIGH
s_axi_awready	Output	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready 0 = slave not ready.
Write Data Channel		
s_axi_wuser[C_AXI_WUSER_WIDTH-1:0]	Input	Write Data Channel User signals. The WUSER width can be configured from 1 to 4 bits.
s_axi_wdata[C_AXI_DATA_WIDTH-1:0]	Input	Write data. The write data bus can be 32 or 64 bits wide.
s_axi_wstrb[C_AXI_DATA_WIDTH/8-1:0]	Input	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for every eight bits of the write data bus. WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].
s_axi_wlast	Input	Write last. This signal indicates the last transfer in a write burst.
s_axi_wvalid	Input	Write valid. This signal indicates that valid write data and strobes are available: 1 = write data and strobes available. 0 = write data and strobes not available.
s_axi_wready	Output	Write ready. This signal indicates that the slave can accept the write data: 1 = slave ready. 0 = slave not ready.
Write Response Channel		
s_axi_bid[C_AXI_ID_WIDTH-1:0]	Output	Response ID. The identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding. ID width can be configured from 1 to 6 bits.
s_axi_bresp[1:0]	Output	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
s_axi_bvalid	Output	Write response valid. This signal indicates that a valid write response is available: 1 = write response available. 0 = write response not available.

Table 2-6: **AXI Interface Signals *(Cont'd)***

Name	Direction	Description
s_axi_bready	Input	Response ready. This signal indicates that the Master can accept the response information. 1 = Master ready. 0 = Master not ready.
Read Address Channel		
s_axi_arid[C_AXI_ID_WIDTH-1:0]	Input	Read address ID. This signal is the identification tag for the read address group of signals. ID width can be configured from 1 to 6 bits.
s_axi_araddr[31:0]	Input	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided. The control signals that are issued with the address detail how the address is calculated for the remaining transfers in the burst.
s_axi_arlen[7:0]	Input	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
s_axi_arsize[1:0]	Input	Burst size. This signal indicates the size of each transfer in the burst.
s_axi_arburst[1:0]	Input	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
s_axi_arvalid	Input	Read address valid. When High, this signal indicates that the read address and control information is valid and remains stable until the address acknowledge signal, ARREADY, is High. 1 = address and control information valid. 0 = address and control information not valid.
s_axi_arready	Output	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready. 0 = slave not ready.
Read Data Channel		
s_axi_rid[C_AXI_ID_WIDTH-1:0]	Output	Read ID tag. This signal is the ID tag of the read data group of signals. The RID value is generated by the slave. It must match the ARID value of the read transaction to which it is responding. ID width can be configured from 1 to 6 bits.
s_axi_rdata[C_AXI_DATA_WIDTH-1:0]	Output	Read data. The read data bus can be 32 or 64 bits wide.
s_axi_rresp[1:0]	Output	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.

Table 2-6: **AXI Interface Signals *(Cont'd)***

Name	Direction	Description
s_axi_rlast	Output	Read last. This signal indicates the last transfer in a read burst.
s_axi_rvalid	Output	Read valid. This signal indicates that the required read data is available and the read transfer can complete: 1 = read data available. 0 = read data not available.
s_axi_rready	Input	Read ready. This signal indicates that the master can accept the read data and response information: 1= master ready. 0 = master not ready.
AXI Master to AXI Slave Interrupts		
axi_c2c_m2s_intr_in[3:0]	Input	Level Interrupt signaling from AXI Master to AXI Slave.
axi_c2c_s2m_intr_out[3:0]	Output	Level Interrupt signaling from AXI Slave to AXI Master.

AXI4-Lite Master Mode Signals

The signals shown in Table 2-7 are applicable only when Master Mode is selected for the AXI4-Lite Interface.

Table 2-7: **AXI4-Lite Master Signals**

Name	Direction	Description
Write Address Channel		
s_axi_lite_awaddr[31:0]	Input	Write address. The write address bus gives the address of the transfer.
s_axi_lite_awprot[3:0]	Input	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
s_axi_lite_awvalid	Input	Write address valid. This signal indicates that valid write address and control information are available: • 1 = address and control information available • 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes High.
s_axi_lite _awready	Input	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Write Data Channel		
s_axi_lite _wdata[31:0]	Input	Write data. The write data bus can be 32 bits wide

Table 2-7: **AXI4-Lite Master Signals *(Cont'd)***

Name	Direction	Description
s_axi_lite _wstrb[m/8-1:0]	Input	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].
s_axi_lite _wvalid	Input	Write valid. This signal indicates that valid write data and strobes are available: • 1 = write data and strobes available • 0 = write data and strobes not available
s_axi_lite _wready	Input	Write ready. This signal indicates that the slave can accept the write data: • 1 = slave ready • 0 = slave not ready
Write Response Channel		
s_axi_lite _bresp[1:0]	Input	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This Signal is mapped to the input data bus of the FIFO.
s_axi_lite _bvalid	Input	Write response valid. This signal indicates that a valid write response is available: • 1 = write response available • 0 = write response not available
s_axi_lite _bready	Input	Response ready. This signal indicates that the master can accept the response information. • 1 = master ready • 0 = master not ready
Read Address Channel		
s_axi_lite _araddr[31:0]	Input	Read address. The read address bus gives the address of a read transaction.
s_axi_lite _arprot[3:0]	Input	Protection type. This signal provides protection unit information for the transaction.
s_axi_lite _arvalid	Input	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and remains stable until the address acknowledge signal, ARREADY, is High. • 1 = address and control information valid • 0 = address and control information not valid
s_axi_lite _arready	Input	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready.
Read Data Channel		
s_axi_lite _rdata[31:0]	Input	Read data. The read data bus can be 32 bits wide.

Table 2-7: **AXI4-Lite Master Signals *(Cont'd)***

Name	Direction	Description
s_axi_lite _rresp[1:0]	Input	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This Signal is mapped to the input data bus of the FIFO.
s_axi_lite _rvalid	Input	Read valid. This signal indicates that the required read data is available and the read transfer can complete: • 1 = read data available • 0 = read data not available
s_axi_lite _rready	Input	Read ready. This signal indicates that the master can accept the read data and response information: • 1= master ready • 0 = master not ready

AXI4-Lite Slave Mode Signals

The signals shown in Table 2-8 are applicable only when Slave Mode is selected for the AXI4-Lite Interface.

Table 2-8: **AXI4-Lite Slave Signals**

Name	Direction	Description
Write Address Channel		
m_axi_lite_awaddr[m:0]	Output	Write address. The write address bus gives the address of the transfer.
m_axi_lite _awprot[3:0]	Output	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
m_axi_lite _awvalid	Output	Write address valid. This signal indicates that valid write address and control information are available: • 1 = address and control information available • 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes High.
m_axi_lite _awready	Output	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Write Data Channel		
m_axi_lite _wdata[m-1:0]	Output	Write data. The write data bus can be 32 bits wide
m_axi_lite _wstrb[m/8-1:0]	Output	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].

Table 2-8: **AXI4-Lite Slave Signals *(Cont'd)***

Name	Direction	Description
m_axi_lite _wvalid	Output	Write valid. This signal indicates that valid write data and strobes are available: • 1 = write data and strobes available • 0 = write data and strobes not available
m_axi_lite _wready	Output	Write ready. This signal indicates that the slave can accept the write data: • 1 = slave ready • 0 = slave not ready
Write Response Channel		
m_axi_lite _bresp[1:0]	Output	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This signal is mapped to the input data bus of the FIFO.
m_axi_lite _bvalid	Output	Write response valid. This signal indicates that a valid write response is available: • 1 = write response available • 0 = write response not available
m_axi_lite _bready	Output	Response ready. This signal indicates that the master can accept the response information. • 1 = master ready • 0 = master not ready
Read Address Channel		
m_axi_lite _araddr[m:0]	Output	Read address. The read address bus gives the address of a read transaction.
m_axi_lite _arprot[3:0]	Output	Protection type. This signal provides protection unit information for the transaction.
m_axi_lite _arvalid	Output	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledge signal, ARREADY, is High. • 1 = address and control information valid • 0 = address and control information not valid
m_axi_lite _arready	Output	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Read Data Channel		
m_axi_lite _rdata[m-1:0]	Output	Read data. The read data bus can be 32 bits wide.
m_axi_lite_rresp[1:0]	Output	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This signal is mapped to the input data bus of the FIFO.

Table 2-8: **AXI4-Lite Slave Signals *(Cont'd)***

Name	Direction	Description
m_axi_lite _rvalid	Output	Read valid. This signal indicates that the required read data is available and the read transfer can complete: • 1 = read data available • 0 = read data not available
m_axi_lite _rready	Output	Read ready. This signal indicates that the master can accept the read data and response information: • 1= master ready • 0 = master not ready

Master Device Interface Signals

Table 2-9 describes the Master Device Interface signals for the AXI Chip2Chip Master core.

Table 2-9: **Master Device Interface Signals**

Name	Direction	Description
Single Ended SelectIO Interface		
axi_c2c_selio_tx_clk_out	Output	SelectIO FPGA interface clock from Master device to Slave device.
axi_c2c_selio_tx_data_out[m-1:0]	Output	SelectIO FPGA Interface Data from Master device to Slave device. 'm' is the number of Output I/Os required for Master-to-Slave device interfacing. For details, see User Tab in Chapter 4.
axi_c2c_selio_rx_clk_in	Input	SelectIO FPGA interface clock from Slave device to Master device.
axi_c2c_selio_rx_data_in[m-1:0]	Input	SelectIO FPGA interface signals from Slave device to Master device. 'm' is number of Input I/Os required for Slave to Master device interfacing. For details, see User Tab in Chapter 4.
Differential SelectIO Interface		
axi_c2c_selio_tx_diff_clk_out_p axi_c2c_selio_tx_diff_clk_out_n	Output	Select IO differential clock from Master to Slave device. Differential clocking is valid when C_USE_DIFF_CLK is set to 1.
axi_c2c_selio_tx_diff_data_out_p[m-1:0] axi_c2c_selio_tx_diff_data_out_n[m-1:0]	Output	SelectIO differential Data from Master to Slave device. 'm' is the number of Output I/Os required for Master-to-Slave device interfacing. For details, see Chip2Chip PHY Width in Chapter 4. Differential data is valid when C_USE_DIFF_IO is set to 1.
axi_c2c_selio_rx_diff_clk_in_p axi_c2c_selio_rx_diff_clk_in_n	Input	Select IO differential clock from Slave to Master device. Differential clocking is valid when C_USE_DIFF_CLK is set to 1.

Table 2-9: **Master Device Interface Signals *(Cont'd)***

Name	Direction	Description
axi_c2c_selio_rx_diff_data_in_p[m-1:0] axi_c2c_selio_rx_diff_data_in_n[m-1:0]	Input	SelectIO differential data signals from Slave to Master device. 'm' is number of Input I/Os required for Slave to Master device interfacing. For details, see User Tab in Chapter 4. Differential data is valid when C_USE_DIFF_IO is set to 1.
Master FPGA Aurora Interface Signals		
axi_c2c_aurora_channel_up	Input	Asserted (active-High) when Aurora channel initialization is complete, and the channel is ready to send data.
axi_c2c_aurora_tx_tready	Input	Asserted (active-High) during clock edges when signals from the AXI Chip2Chip core are accepted (if axi_c2c_aurora_tx_tvalid is also asserted). De-asserted (Low) on clock edges when signals from the AXI Chip2Chip core are ignored.
axi_c2c_aurora_tx_tdata	Output	Streaming Data output from AXI Chip2Chip core.
axi_c2c_aurora_tx_tvalid	Output	Asserted (active-High) when AXI4-Stream signals from the AXI Chip2Chip core are valid. Deasserted (Low) when AXI4-Stream control signals and/or data from the AXI Chip2Chip core should be ignored.
axi_c2c_aurora_rx_tdata	Input	Streaming Data input to AXI Chip2Chip core.
axi_c2c_aurora_rx_tvalid	Input	Asserted (active-High) when data and control signals from an Aurora 64B/66B core are valid. De-asserted (active-Low) when data and/or control signals from an Aurora 64B/66B core should be ignored.

Interrupt and Status Signals

Table 2-10 describes the interrupt and status signals for the AXI Chip2Chip Master core.

Table 2-10: **Interrupt and Status Signals**

Name	Direction	Description
axi_c2c_link_status_out	Output	Link Status: Asserted when Link Detect FSM is in the SYNC state. Deasserted when either the Master or Slave AXI Chip2Chip core is under reset or when the Link Detect FSM is not in the SYNC state.
axi_c2c_link_error_out	Output	Link Error Interrupt: Asserted when the AXI Chip2Chip Slave core is reset during normal operations.

Table 2-10: **Interrupt and Status Signals *(Cont'd)***

Name	Direction	Description
axi_c2c_multi_bit_error_out	Output	Multi-bit Error Interrupt: When asserted, this interrupt indicates multiple bits are received with errors in the Master or Slave AXI Chip2Chip core. For the SelectIO interface, a multi-bit error is determined during deskew operations and indicates failure of those operations. For the Aurora interface, multi-bit errors are determined by the ECC function.
axi_c2c_config_error_out	Output	Configuration Error Interrupt: Applicable only when Aurora is selected as the FPGA interface for the core. When asserted, this interrupt indicates that Link Detect FSM failed due to a configuration mismatch of Master and Slave AXI Chip2Chip cores.

AXI Chip2Chip Slave

This section describes the ports when the AXI Chip2Chip core is a Slave. It contains these subsections:

- Global Signals
- AXI4 Interface Signals
- Slave Device Interface Signals
- Status Signals

Global Signals

Table 2-11 describes the global signals for the AXI Chip2Chip Slave core.

Table 2-11: **Global Interface Signals**

Name	Direction	Description
m_aclk	Input	Global Master Interface Clock (Independent Clock). The m_aclk signal is an input when the Independent Clock mode of operation is selected for the core. Note: In Common Clock mode, the m_aclk input does not drive any logic in the AXI Chip2Chip Slave core; however, for Vivado Design Suite to build the system, this clock input needs to be connected. For this, the m_aclk input pin can be connected to m_aclk_out driven by the AXI Chip2Chip Slave core.
m_aclk_out	Output	Global Master Interface Clock (Common Clock). The m_aclk_out signal is output when the Common Clock Mode of operation is selected for the core.
idelay_ref_clk	Input	SelectIO Interface I/O Reference Clock. This signal is applicable only when the SelectIO interface is selected as the FPGA interfacing option and when deskew is enabled for the SelectIO interface. The applicable frequency for `idelay_ref_clk` is 200 MHz or 300 MHz (± 10 MHz)
m_aresetn	Input	Global Reset. This signal is active-Low and synchronous. All applicable clock inputs to the AXI Chip2Chip Slave core must be stable when m_aresetn input is de-asserted.
Optional AXI4-Lite Signals		
m_axi_lite_aclk	Input	Master Interface AXI4-Lite Clock. Applicable only when Slave Mode of AXI4-Lite is selected. All signals are sampled on the rising edge of this clock.
s_axi_lite_aclk	Input	Slave Interface AXI4-Lite Clock. Applicable only when Master Mode of AXI-Lite is selected. All signals are sampled on the rising edge of this clock.
Aurora Interface Signals		
axi_c2c_phy_clk	Input	Physical Interface Clock. Applicable only when Aurora is selected as the FPGA interface for the core. This signal is provided by the Aurora PHY.

AXI4 Interface Signals

Table 2-12 describes the AXI4 Interface signals for the AXI Chip2Chip Slave core.

Table 2-12: **AXI4 Interface Signals**

Name	Direction	Description
Write Address Channel		
m_axi_awid[C_AXI_ID_WIDTH-1:0]	Output	Write address ID. This signal is the identification tag for the write address group of signals. ID width can be configured from 1 to 6 bits.
m_axi_awaddr[31:0]	Output	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
m_axi_awlen[7:0]	Output	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
m_axi_awsize[1:0]	Output	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate exactly which byte lanes to update.
m_axi_awburst[1:0]	Output	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
m_axi_awvalid	Output	Write address valid. This signal indicates that valid write address and control information are available: 1 = address and control information available. 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes HIGH.
m_axi_awready	Input	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready. 0 = slave not ready.
Write Data Channel		
m_axi_wuser[C_AXI_WUSER_WIDTH-1:0]	Output	Write Data Channel User signals. The WUSER width can be configured from 1 to 4 bits.
m_axi_wdata[C_AXI_DATA_WIDTH-1:0]	Output	Write data. The write data bus can be 32 or 64 bits wide.
m_axi_wstrb[C_AXI_DATA_WIDTH/8-1:0]	Output	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].
m_axi_wlast	Output	Write last. This signal indicates the last transfer in a write burst.

Table 2-12: **AXI4 Interface Signals *(Cont'd)***

Name	Direction	Description
m_axi_wvalid	Output	Write valid. This signal indicates that valid write data and strobes are available: 1 = write data and strobes available. 0 = write data and strobes not available.
m_axi_wready	Input	Write ready. This signal indicates that the slave can accept the write data: 1 = slave ready. 0 = slave not ready.
Write Response Channel		
m_axi_bid[C_AXI_ID_WIDTH-1:0]	Input	Response ID. The identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding. ID width can be configured from 1 to 6 bits.
m_axi_bresp[1:0]	Input	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
m_axi_bvalid	Input	Write response valid. This signal indicates that a valid write response is available: 1 = write response available. 0 = write response not available.
m_axi_bready	Output	Response ready. This signal indicates that the master can accept the response information. 1 = master ready. 0 = master not ready.
Read Address Channel		
m_axi_arid[C_AXI_ID_WIDTH-1:0]	Output	Read address ID. This signal is the identification tag for the read address group of signals. ID width can be configured from 1 to 6 bits.
m_axi_araddr[31:0]	Output	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst.
m_axi_arlen[7:0]	Output	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
m_axi_arsize[1:0]	Output	Burst size. This signal indicates the size of each transfer in the burst.
m_axi_arburst[1:0]	Output	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.

Table 2-12: **AXI4 Interface Signals *(Cont'd)***

Name	Direction	Description
m_axi_arvalid	Output	Read address valid. When High, this signal indicates that the read address and control information is valid and remains stable until the address acknowledge signal, ARREADY, is High. 1 = address and control information valid. 0 = address and control information not valid.
m_axi_arready	Input	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: 1 = slave ready. 0 = slave not ready.
Read Data Channel		
m_axi_rid[C_AXI_ID_WIDTH-1:0]	Input	Read ID (RID) tag. This signal is the ID tag of the read data group of signals. The RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding. ID width can be configured from 1 to 6 bits.
m_axi_rdata[C_AXI_DATA_WIDTH-1:0]	Input	Read data. The read data bus can 32 or 64 bits wide.
m_axi_rresp[1:0]	Input	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR.
m_axi_rlast	Input	Read last. This signal indicates the last transfer in a read burst.
m_axi_rvalid	Input	Read valid. This signal indicates that the required read data is available and the read transfer can complete: 1 = read data available. 0 = read data not available.
m_axi_rready	Output	Read ready. This signal indicates that the master can accept the read data and response information: 1= master ready. 0 = master not ready.
AXI Slave to AXI Master Interrupts		
axi_c2c_m2s_intr_out[3:0]	Output	Level Interrupt signaling from the AXI Master to the AXI Slave.
axi_c2c_s2m_intr_in[3:0]	Input	Level Interrupt signaling from the AXI Slave to the AXI Master.

AXI4-Lite Master Mode Signals

The signals shown in Table 2-7 are applicable only when Master Mode is selected for the AXI4-Lite Interface.

Table 2-13: **AXI4-Lite Master Signals**

Name	Direction	Description
Write Address Channel		
s_axi_lite_awaddr[31:0]	Input	Write address. The write address bus gives the address of the transfer.
s_axi_lite_awprot[3:0]	Input	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
s_axi_lite_awvalid	Input	Write address valid. This signal indicates that valid write address and control information are available: • 1 = address and control information available • 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes High.
s_axi_lite _awready	Input	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Write Data Channel		
s_axi_lite _wdata[31:0]	Input	Write data. The write data bus can be 32 bits wide
s_axi_lite _wstrb[m/8-1:0]	Input	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].
s_axi_lite _wvalid	Input	Write valid. This signal indicates that valid write data and strobes are available: • 1 = write data and strobes available • 0 = write data and strobes not available
s_axi_lite _wready	Input	Write ready. This signal indicates that the slave can accept the write data: • 1 = slave ready • 0 = slave not ready
Write Response Channel		
s_axi_lite _bresp[1:0]	Input	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This Signal is mapped to the input data bus of the FIFO.

Table 2-13: **AXI4-Lite Master Signals *(Cont'd)***

Name	Direction	Description
s_axi_lite _bvalid	Input	Write response valid. This signal indicates that a valid write response is available: • 1 = write response available • 0 = write response not available
s_axi_lite _bready	Input	Response ready. This signal indicates that the master can accept the response information. • 1 = master ready • 0 = master not ready
Read Address Channel		
s_axi_lite _araddr[31:0]	Input	Read address. The read address bus gives the address of a read transaction.
s_axi_lite _arprot[3:0]	Input	Protection type. This signal provides protection unit information for the transaction.
s_axi_lite _arvalid	Input	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and remains stable until the address acknowledge signal, ARREADY, is High. • 1 = address and control information valid • 0 = address and control information not valid
s_axi_lite _arready	Input	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready.
Read Data Channel		
s_axi_lite _rdata[31:0]	Input	Read data. The read data bus can be 32 bits wide.
s_axi_lite _rresp[1:0]	Input	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This Signal is mapped to the input data bus of the FIFO.
s_axi_lite _rvalid	Input	Read valid. This signal indicates that the required read data is available and the read transfer can complete: • 1 = read data available • 0 = read data not available
s_axi_lite _rready	Input	Read ready. This signal indicates that the master can accept the read data and response information: • 1= master ready • 0 = master not ready

AXI4-Lite Slave Mode Signals

The signals shown in Table 2-8 are applicable only when Slave Mode is selected for the AXI4-Lite Interface.

Table 2-14: **AXI4-Lite Slave Signals**

Name	Direction	Description
Write Address Channel		
m_axi_lite_awaddr[m:0]	Output	Write address. The write address bus gives the address of the transfer.
m_axi_lite _awprot[3:0]	Output	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
m_axi_lite _awvalid	Output	Write address valid. This signal indicates that valid write address and control information are available: • 1 = address and control information available • 0 = address and control information not available. The address and control information remain stable until the address acknowledge signal, AWREADY, goes High.
m_axi_lite _awready	Output	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Write Data Channel		
m_axi_lite _wdata[m-1:0]	Output	Write data. The write data bus can be 32 bits wide
m_axi_lite _wstrb[m/8-1:0]	Output	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus. Therefore, WSTRB[n] corresponds to WDATA[(8 × n) + 7:(8 × n)].
m_axi_lite _wvalid	Output	Write valid. This signal indicates that valid write data and strobes are available: • 1 = write data and strobes available • 0 = write data and strobes not available
m_axi_lite _wready	Output	Write ready. This signal indicates that the slave can accept the write data: • 1 = slave ready • 0 = slave not ready
Write Response Channel		
m_axi_lite _bresp[1:0]	Output	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This signal is mapped to the input data bus of the FIFO.
m_axi_lite _bvalid	Output	Write response valid. This signal indicates that a valid write response is available: • 1 = write response available • 0 = write response not available
m_axi_lite _bready	Output	Response ready. This signal indicates that the master can accept the response information. • 1 = master ready • 0 = master not ready

Table 2-14: **AXI4-Lite Slave Signals *(Cont'd)***

Name	Direction	Description
Read Address Channel		
m_axi_lite _araddr[m:0]	Output	Read address. The read address bus gives the address of a read transaction.
m_axi_lite _arprot[3:0]	Output	Protection type. This signal provides protection unit information for the transaction.
m_axi_lite _arvalid	Output	Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledge signal, ARREADY, is High. • 1 = address and control information valid • 0 = address and control information not valid
m_axi_lite _arready	Output	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: • 1 = slave ready • 0 = slave not ready
Read Data Channel		
m_axi_lite _rdata[m-1:0]	Output	Read data. The read data bus can be 32 bits wide.
m_axi_lite_rresp[1:0]	Output	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR, and DECERR. This signal is mapped to the input data bus of the FIFO.
m_axi_lite _rvalid	Output	Read valid. This signal indicates that the required read data is available and the read transfer can complete: • 1 = read data available • 0 = read data not available
m_axi_lite _rready	Output	Read ready. This signal indicates that the master can accept the read data and response information: • 1= master ready • 0 = master not ready

Slave Device Interface Signals

Table 2-15 describes the Slave Device Interface signals for the AXI Chip2Chip Slave core.

Table 2-15: **Slave Device Interface Signals**

Name	Direction	Description
axi_c2c_selio_tx_clk_out	Output	SelectIO FPGA interface clock from Slave device to Master device.
axi_c2c_selio_tx_data_out[*m*-1:0]	Output	SelectIO FPGA Interface Data from Slave device to Master device. *m* is the number of input I/Os required for Slave-to-Master device interfacing. For details, see User Tab in Chapter 4.

Table 2-15: **Slave Device Interface Signals *(Cont'd)***

Name	Direction	Description
axi_c2c_selio_rx_clk_in	Input	SelectIO FPGA interface clock from Master device to Slave device.
axi_c2c_selio_rx_data_in[*m*-1:0]	Input	SelectIO FPGA interface signals from Master device to Slave device. *m* is number of output I/Os required for Master-to-Slave device interfacing. For details, see User Tab in Chapter 4.
axi_c2c_selio_tx_diff_clk_out_p axi_c2c_selio_tx_diff_clk_out_n	Output	Select IO differential clock from Slave to Master device. Differential clocking is valid when C_USE_DIFF_CLK is set to 1.
axi_c2c_selio_tx_diff_data_out_p[m-1:0] axi_c2c_selio_tx_diff_data_out_n[m-1:0]	Output	SelectIO differential Data from Slave to Master device. 'm' is the number of Output I/Os required for Slave-to-Master device interfacing. For details, see User Tab in Chapter 4. Differential data is valid when C_USE_DIFF_IO is set to 1.
axi_c2c_selio_rx_diff_clk_in_p axi_c2c_selio_rx_diff_clk_in_n	Input	Select IO differential clock from Master to Slave device. Differential clocking is valid when C_USE_DIFF_CLK is set to 1.
axi_c2c_selio_rx_diff_data_in_p[m-1:0] axi_c2c_selio_rx_diff_data_in_n[m-1:0]	Input	SelectIO differential data signals from Master to Slave device. 'm' is number of Input I/Os required for Master to Slave device interfacing. For details, see User Tab in Chapter 4. Differential data is valid when C_USE_DIFF_IO is set to 1.
Slave FPGA Aurora Interface Signals		
axi_c2c_aurora_channel_up	Input	Asserted (active-High) when Aurora channel initialization is complete, and channel is ready to send data.
axi_c2c_aurora_tx_tready	Input	Asserted (active-High) during clock edges when signals from the AXI Chip2Chip core are accepted (if axi_c2c_aurora_tx_tvalid is also asserted). De-asserted (Low) on clock edges when signals from the AXI Chip2Chip core are ignored.
axi_c2c_aurora_tx_tdata	Output	Streaming Data output from AXI Chip2Chip core.
axi_c2c_aurora_tx_tvalid	Output	Asserted (active-High) when AXI4-Stream signals from the AXI Chip2Chip core are valid. Deasserted (Low) when AXI4-Stream control signals and/or data from the AXI Chip2Chip core should be ignored.
axi_c2c_aurora_rx_tdata	Input	Streaming Data input to AXI Chip2Chip core.
axi_c2c_aurora_rx_tvalid	Input	Asserted (active-High) when data and control signals from an Aurora 64B/66B core are valid. De-asserted (active-Low) when data and/or control signals from an Aurora 64B/66B core should be ignored.

Status Signals

Table 2-16 describes the status signals for the AXI Chip2Chip Slave core.

Table 2-16: **Status Signals**

Name	Direction	Description
axi_c2c_link_status_out	Output	Link Status: Link Status from AXI Chip2Chip Slave core. Asserted when the Link Detect FSM is in the SYNC state. Deasserted when either the Master or Slave AXI Chip2Chip core is under reset or when Link Detect FSM is not in the SYNC state.
axi_c2c_multi_bit_error_out	Output	Multibit Error: Multibit Error Status from AXI Chip2Chip Slave core. Indicates multiple bits are received with error. For the SelectIO interface, a multibit error is determined during deskew operations and indicates failure of those operations.
axi_c2c_config_error_out	Output	Configuration Error Interrupt: Applicable only when Aurora is selected as the FPGA interface for the core. When asserted, this interrupt indicates that Link Detect FSM failed due to configuration mismatch of Master and Slave AXI Chip2Chip cores.

Designing with the Core

This chapter includes guidelines and additional information to make designing with the core easier.

General Design Guidelines

The customizable AXI Chip2Chip core provides multiple clocking and I/O interface options to the user. Users can determine the frequency at which the interface needs to be operated. Based on the interface frequency, the user can select the I/O type by providing the appropriate constraints in the Xilinx constraints file (XDC). Selecting the SelectIO™ interface DDR option doubles the I/O speed without impacting the latency or performance. Based on the selection in the User Tab, additional internal width conversion stages can be enabled. Each 2:1 stage of width conversion can increase bridging latencies and can also impact performance.

In addition, common clock and independent clock operations can be selected by the user. The common clock mode of operation reduces clock domain crossing latencies, and the independent clock mode provides additional clock conversion functionality. Both AXI Chip2Chip Master and AXI Chip2Chip Slave cores can be independently selected for either Common Clock or Independent clock operation with the exception of an AXI Chip2Chip Master core with Aurora interface. Operating the AXI Chip2Chip core at frequencies greater than AXI interface frequencies (Independent clock operation) reduces the bridging latencies and can improve overall performance of the AXI Chip2Chip bridging function.

Clocking

Figure 3-1 provides the clocking requirement for the SelectIO interface. In addition to AXI clocks, the deskew function, when enabled, requires an additional 200 MHz or 300 MHz (± 10 MHz) reference clock. Both AXI Chip2Chip Master and AXI Chip2Chip Slave cores can be independently selected for either Common Clock or Independent Clock operations. When the AXI Chip2Chip Slave core is selected for Common Clock operation, the core provides clock and reset (Link Status) to the interfacing slave device.

When the AXI4-Lite interface is enabled, it always operates on an independent `axi_lite` clock input. The AXI4-Lite Master core operations are synchronous to `s_axi_lite_aclk`, and the AXI Lite Slave core operations are synchronous to `m_axi_lite_aclk`.

IMPORTANT: *All input clocks to the Master or Slave Chip2Chip cores must be stable when Reset input to the core is deasserted.*

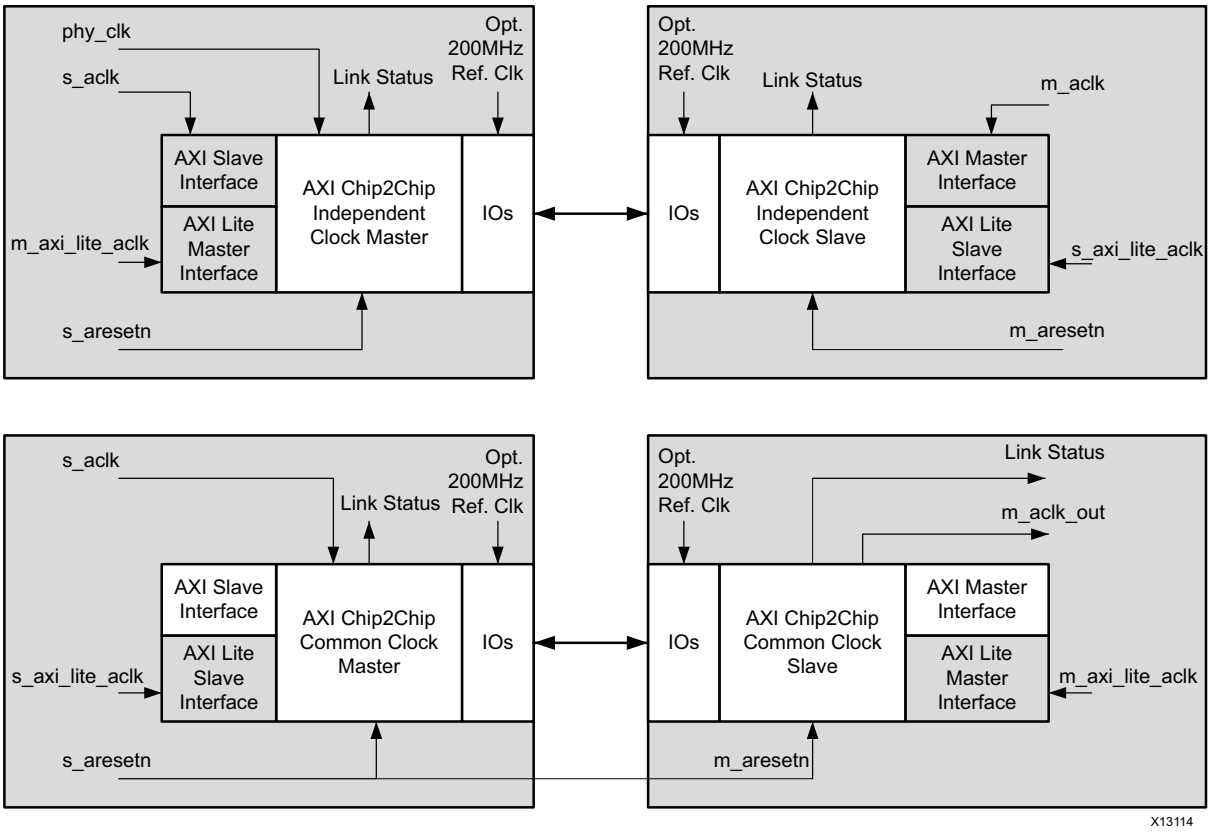

Figure 3-1: **Clocking with the SelectIO Interface**

Resets

The AXI Chip2Chip core allows both Master and Slave cores to have independent reset mapping. The link detect FSM ensures the transactions from the Master device (AXI4 and AXI4-Lite) get initiated only when both Master and Slave AXI Chip2Chip cores are out of reset and ready to accept transactions. Reset can also be propagated from Master device to Slave device. In this case, the user needs to map the reset from Master device to Slave device.

There is no separate reset for the AXI4-Lite interface. The AXI4-Lite interface is brought out of reset when the link detects FSM is in LINKUP state (when the link status output of the

core is asserted). All input clocks to the core, including AXI4-Lite clock, must be stable when the core is brought out of reset (when `aresetn` core input is deasserted).

It is not recommended to reset either Master or Slave AXI Chip2Chip core during normal operation or when Link Status is asserted. AXI Chip2Chip cores maintain a dedicated channel to communicate the Reset condition during normal operations. When the Slave device is reset during normal operations, the Link Status is deasserted and a link error interrupt is asserted.

Calibration and Link Error Detection

Figure 3-2 shows the calibration and link error detection sequence in the AXI Chip2Chip core.

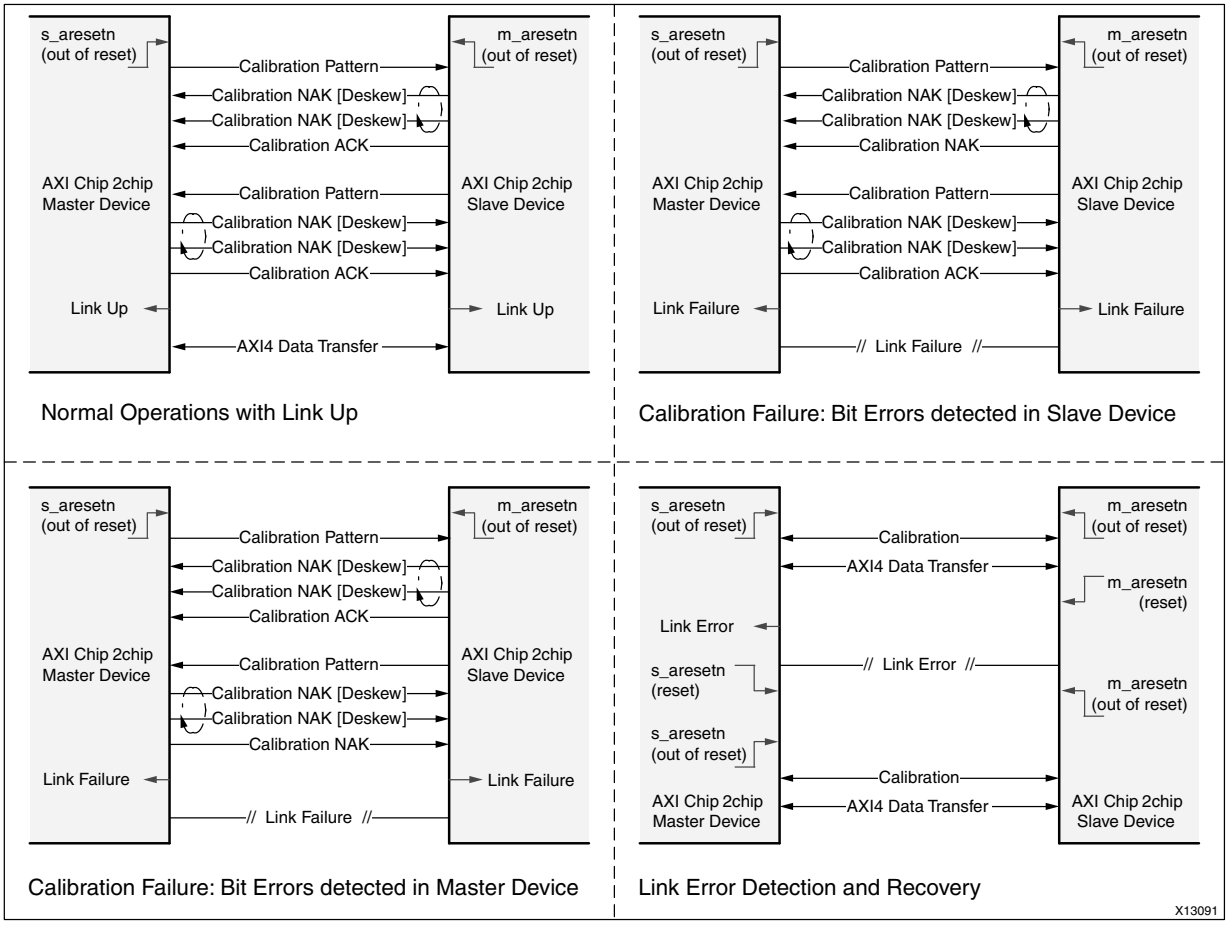

Figure 3-2: **Calibration and Link Error Detection**

The AXI Chip2Chip core implements Link Detect FSM for device detection and calibration functions for the SelectIO PHY interface. The calibration process is initiated when either the Master or Slave AXI Chip2Chip is brought out of reset. During the calibration process, a

fixed set of patterns are exchanged between the Master and Slave devices. The receiving device responds with a NAK when the patterns received do not match the fixed patterns sent by the transmitting device. Deskew operations align the data until an optimized sampling point is determined. Once the patterns are determined to match for the greatest number of the sampling points, the receiving device responds with an ACK. This operation is performed at nibble level for 32 sampling points. The best sampling point is determined for each nibble in the data. Link status is asserted once both Master and Slave devices respond with an ACK. The Link Failure (`axi_c2c_multi_bit_error_out`) signal is asserted when a multi-bit error is determined during deskew operations and indicates the failure of those operations. In this case, either the interface rate can be reduced or IO Type can be appropriately selected to achieve the required interface rate.

When the Link Status signal is asserted, the AXI Chip2Chip core transparently bridges transactions in compliance with AXI protocol specifications. It is not recommended to reset or disconnect either the Master or Slave AXI Chip2Chip core during normal operation or when the Link Status signal is asserted. When the Slave device is reset or if the cable is disconnected during normal operations, the Link Status signal will be deasserted and a link error interrupt will be asserted in the Master device. After being asserted, a link error interrupt can be cleared only with a reset. The AXI Chip2Chip core operations are re-initiated when the Master and Slave AXI Chip2Chip devices are brought out of reset.

Auto-Negotiation

The AXI Chip2Chip core implements Link Detect FSM for auto-negotiation operations when using an Aurora PHY interface. During auto-negotiation, the capabilities of the AXI Chip2Chip Master and Slave cores are matched before the link layer is enabled. After the Aurora channel is up, the Master and Slave cores exchange patterns (based on the respective configurations) with each other. The Link status signal is asserted when the configurations of the Master and Slave cores match. Otherwise, the Configuration Error status signal is asserted.

Customizing and Generating the Core

This chapter contains information and instructions for using the Vivado™ Design Suite to customize the LogiCORE IP AXI Chip2Chip core.

Vivado Integrated Design Environment

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or popup menu.

For details, see the sections, "Working with IP" and "Customizing IP for the Design" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 4] and the "Working with the Vivado IDE" section in the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7].

Note: Figures in this chapter are illustrations of the Vivado Integrated Design Environment (IDE). This layout might vary from the current version.

Figure 4-1 shows the Vivado IDE for the AXI Chip2Chip core. The options are described following the figure.

User Tab

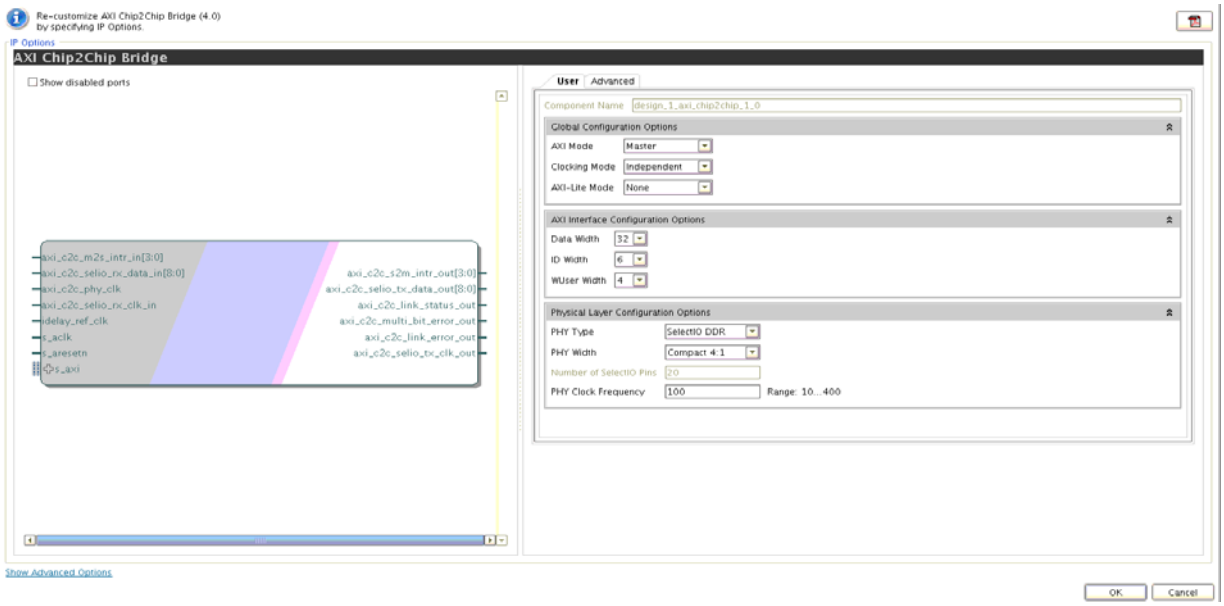

Figure 4-1: **Customization Vivado IDE for the AXI Chip2Chip Core**

- **Chip2Chip AXI Mode**: The Chip2Chip AXI Mode configuration option determines AXI Chip2Chip Master or Slave mode of operation.
- **AXI Clocking Mode**: The AXI Chip2Chip core can be configured with either Independent or Common Clock domains.

  The Independent Clock configuration allows you to implement unique clock domains on the AXI interface and FPGA I/Os. The AXI Chip2Chip core handles the synchronization between clock domains. Both the AXI interface and FPGA I/Os can also be maintained in a single clock domain. The AXI Chip2Chip core can be used to generate a core optimized for a single clock by selecting the Common Clock option.

- **Chip2Chip AXI4-Lite Mode**: The Chip2Chip AXI4-Lite Mode configuration option determines AXI4-Lite Master or Slave mode of operation, as shown in Table 4-1. When AXI4-Lite interfacing is not required, this configuration option should be set to "None."

Table 4-1: **AXI4-Lite Configuration Options**

Chip2Chip Mode		Chip2Chip AXI4-Lite Options	
Mode	**AXI4 Interface**	**Mode**	**AXI4-Lite Interface**
Master FPGA	Slave	Master	Slave
		Slave	Master
		None	None

Table 4-1: **AXI4-Lite Configuration Options *(Cont'd)***

Chip2Chip Mode		Chip2Chip AXI4-Lite Options	
Mode	**AXI4 Interface**	**Mode**	**AXI4-Lite Interface**
Slave FPGA	Master	Master	Slave
		Slave	Master
		None	None

- **AXI Data Width**: The AXI Data Width user option allows the width of AXI data to be configured. Valid settings for the AXI Data Width are 32 and 64. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.
- **AXI ID Width**: The AXI ID provides an identification tag for the group of signals in the channel. AXI ID is supported for all write and read channels. ID width can be configured from 1 to 6 bits. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.
- **AXI WUSER Width**: AXI WUSER defines sideband information that can be transmitted with the write data channel. The valid range for WUSER width is from 1 to 4 bits. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.

TIP: *The AXI ID Width of the AXI Chip2Chip Slave core should match the AXI ID Width of the AXI Chip2Chip Master core.*

IMPORTANT: In IP Integrator, the AXI ID and WUSER Width of the interconnect are automatically propagated to the AXI Chip2Chip Master core. However for the AXI Chip2Chip Slave core, you have to override the AXI ID Width and WUSER Width so that it matches the parameters of the Master AXI Chip2Chip core.

- **Chip2Chip PHY Type**: The Chip2Chip PHY type can be set to either "SelectIO SDR", "SelectIO DDR", or "Aurora64B66B". This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.
- **Chip2Chip PHY Width**: The Chip2Chip PHY Width configuration determines I/Os used for device-to-device SelectIO interfacing. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores. Table 4-2 provides the mapping between Chip2Chip PHY width and the number of input and output I/Os utilized with the selected option.

Table 4-2: **FPGA SelectIO Utilization**

AXI Data Width	Chip2Chip PHY Type[1]	Chip2Chip PHY Width	Number of Output I/Os	Number of Input I/Os
32	SelectIO SDR	Compact 4:1[2]	19	19
		Compact 2:1	31	31
	SelectIO DDR	Compact 4:1[2]	10	10
		Compact 2:1	16	16
		Compact 1:1	29	29
64	SelectIO SDR	Compact 4:1[2]	26	26
		Compact 2:1	45	45
	SelectIO DDR	Compact 4:1[2]	14	14
		Compact 2:1	23	23
		Compact 1:1	42	42

1. SelectIO PHY interface routes the clock with the data pins.
2. Compact 4:1 is not supported when the AXI4-Lite Interface is enabled for the core.

- **Chip2Chip PHY Frequency**: When using the SelectIO FPGA interface, the Chip2Chip PHY implements the mixed-mode clock manager (MMCM) on the PHY input clocks. MMCMs are used for clock phase alignment, clock slew reduction, and for compensating clock buffer delays. For common clock AXI Chip2Chip Slave operations, the `m_aclk_out` output is generated from the MMCM. The Chip2Chip PHY Frequency provides the clock frequency parameter to the MMCM.

  For Common clock, C_SELECTIO_PHY_CLK must be set to the `s_aclk` frequency. For Independent clock, C_SELECTIO_PHY_CLK must be to set to the `axi_c2c_phy_clk` frequency. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.

IMPORTANT: *In IP Integrator, the PHY Frequency parameter is automatically computed based on the clock frequency of the port connected to `axi_c2c_phy_clk` (Master Independent clocking configuration) or `axi_c2c_selio_rx_*_clk_in*` port(s) (Slave configuration). In Master Common clocking configuration, the frequency of the connected AXI clock is propagated to the PHY Frequency parameter.*

- **Aurora Width**: When the interface is Aurora 64B/66B, the Aurora width parameter shows the size of the streaming data signal. This parameter is calculated with the AXI data width and Chip2Chip PHY width.

Advanced Tab

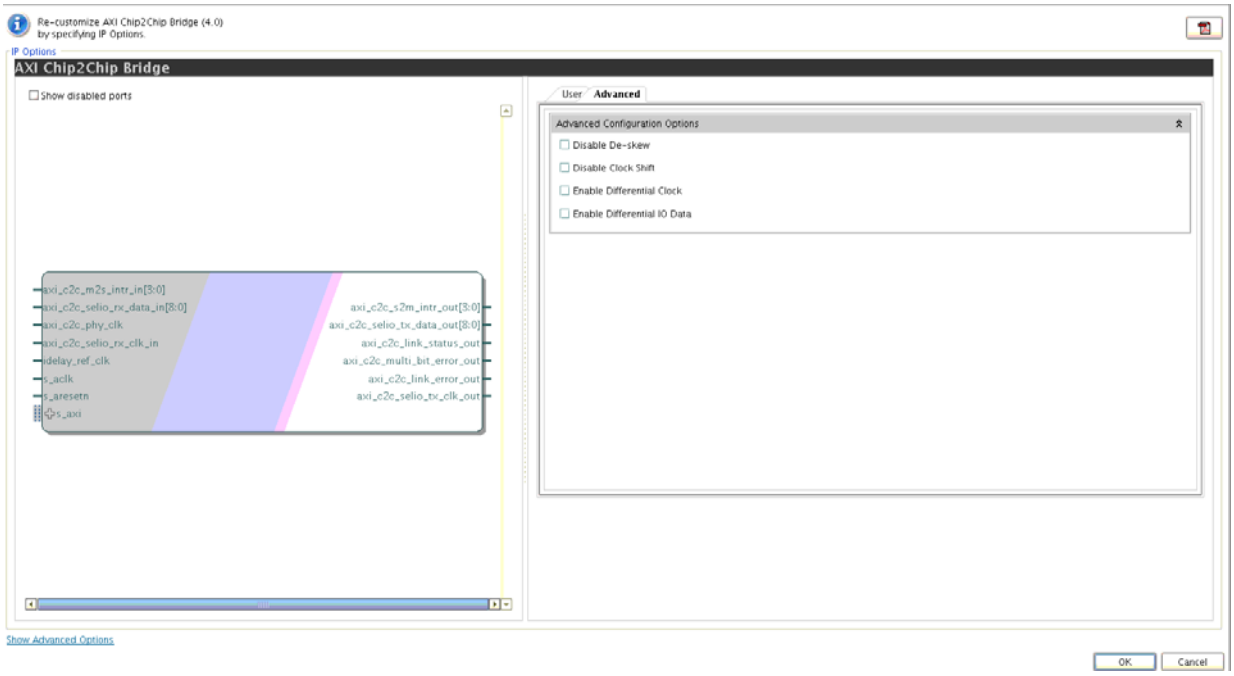

Figure 4-2: **Vivado IDE for Advanced AXI Chip2Chip Core Parameters**

Figure 4-2 shows the Vivado IDE for advanced AXI Chip2Chip core parameters. This tab includes the following options for the SelectIO FPGA interface:

- **Disable De-Skew**: When set to 1, disables the deskew function in the Master or Slave core. The deskew function can be disabled for low frequency I/O operations. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.
- **Disable Clock Shift**: When set to 1, disables 90-degree phase shift for DDR I/O and 180-degree phase shift for SDR I/O in the MMCM. Clock shifting can be disabled for high-frequency DDR I/O operations when the clock half cycle period is less than the maximum deskew range of 2.5 ns. This setting must be maintained the same in both Master and Slave, AXI Chip2Chip cores.
- **Enable Differential Clock**: When set to 1, implements differential IO buffer on the two clocks I/Os used for device interfacing. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.
- **Enable Differential IO Data**: When set to 1, implements differential IO buffer on the data I/Os used for device interfacing. This setting must be maintained the same in both Master and Slave AXI Chip2Chip cores.

Output Generation

For specifics about files created when the core is generated, see "Generating IP Output Products" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

This chapter contains details about constraining the core.

Required Constraints

The physical layer is a set of SelectIO™ interface pins that carry source synchronous clock with the data pins.

These I/O pins need I/O Location and I/O Standard constraints. These constraints are board specific and needs to be specified accordingly in the top-level XDC.

Clock Frequencies

The recommended frequency for the AXI interface is up to 200 MHz. For the maximum frequency numbers achieved on the SelectIO PHY interface, see Table A-1 in Appendix A, Verification, Compliance, and Interoperability. The clocking mode for the AXI Chip2Chip core needs to be set based on the AXI Interface Frequency and the required SelectIO interface PHY frequency. The required clocking constraints for the AXI Chip2Chip core are listed below:

- **s_aclk**. The AXI interface of the AXI Chip2Chip Master core operates in the `s_aclk` clock domain.
- **axi_c2c_phy_clk**. `axi_c2c_phy_clk` is the SelectIO interface PHY clock and is applicable when the AXI Chip2Chip Master core is configured in Independent Clock mode. For Common Clock mode, this clock constraint is not required because the PHY clock is the same as `s_aclk`. When using the Aurora interface, the `axi_c2c_phy_clk` of the AXI Chip2Chip Master and Slave cores should be constrained with the frequency of the `USER_CLK` output of the Aurora 64B66B core.
- **m_aclk**. The AXI interface of the AXI Chip2Chip Slave core operates in the `m_aclk` clock domain.
- **s_axi_lite_aclk**. AXI4-Lite Master Mode operates in the `s_axi_lite_aclk` clock domain.

- **m_axi_lite_aclk**. AXI4-Lite Slave Mode operates in the `m_axi_lite_aclk` clock domain.
- **idelay_ref_clk**. Both the master and slave AXI Chip2Chip cores utilize the IDELAY_CTRL block for SelectIO PHY calibration. The `idelay_ref_clk` input is the reference clock to the IDELAY_CTRL block. This clock is 200 MHz or 300 MHz (± 10MHz) based on the selected device. This constraint is not required when deskew operation is disabled for the cores.
- **axi_c2c_selio_rx_clk_in**. `axi_c2c_selio_rx_clk_in` is the source synchronous clock of the SelectIO physical layer. This clock pin must be constrained with the PHY clock frequency. When Common Clocking mode is used, this clock runs at the same frequency as `s_aclk`.

Clock Management

The AXI Chip2Chip core utilizes the MMCM module to recover the SelectIO PHY clock. The frequency of the PHY clock is specified by setting the C_SELECTIO_PHY_CLK parameter.

Clock Placement

The clock input pins on the physical layer must be placed on clock-capable I/Os only.

Banking

Device-specific banking rules for placement of PHY I/O pins need to be considered when specifying the top-level XDC.

I/O Standard and Placement

The I/O pins of the AXI Chip2Chip core need I/O Location and I/O Standard constraints. These constraints need to be specified in the top-level XDC.

Synthesis and Implementation

This chapter contains information about synthesis and implementation in the Vivado® Design Suite environment.

For details about synthesis and implementation, see “Synthesizing IP” and “Implementing IP” in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Example Design

This chapter contains information about the provided example design in the Vivado® Design Suite environment.

Overview

Figure 7-1 shows the configuration of the example design with a SelectIO interface. For an Aurora interface, the streaming interface of the AXI Chip2Chip core is connected to the Aurora 64B66B core, as shown in Figure 7-2.

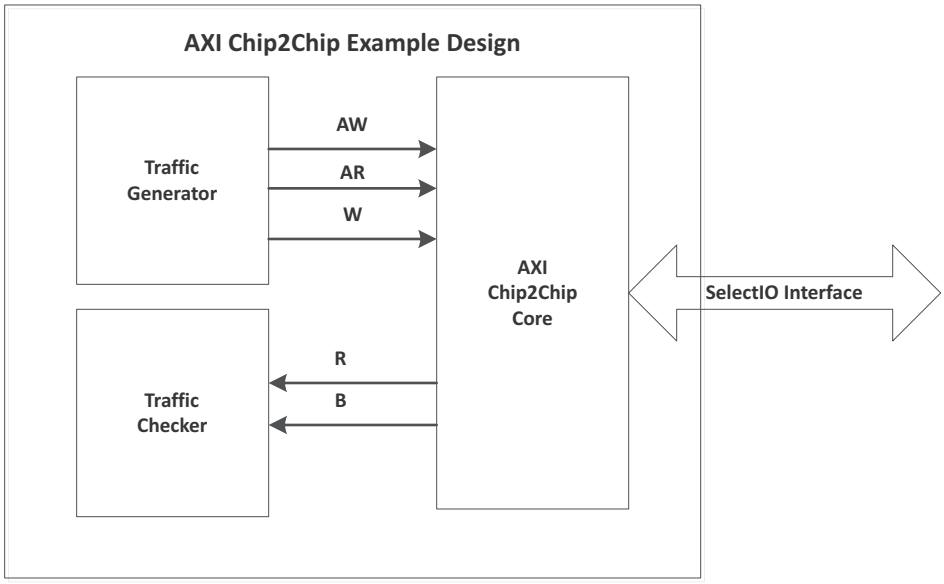

Figure 7-1: **Example Design Block Diagram**

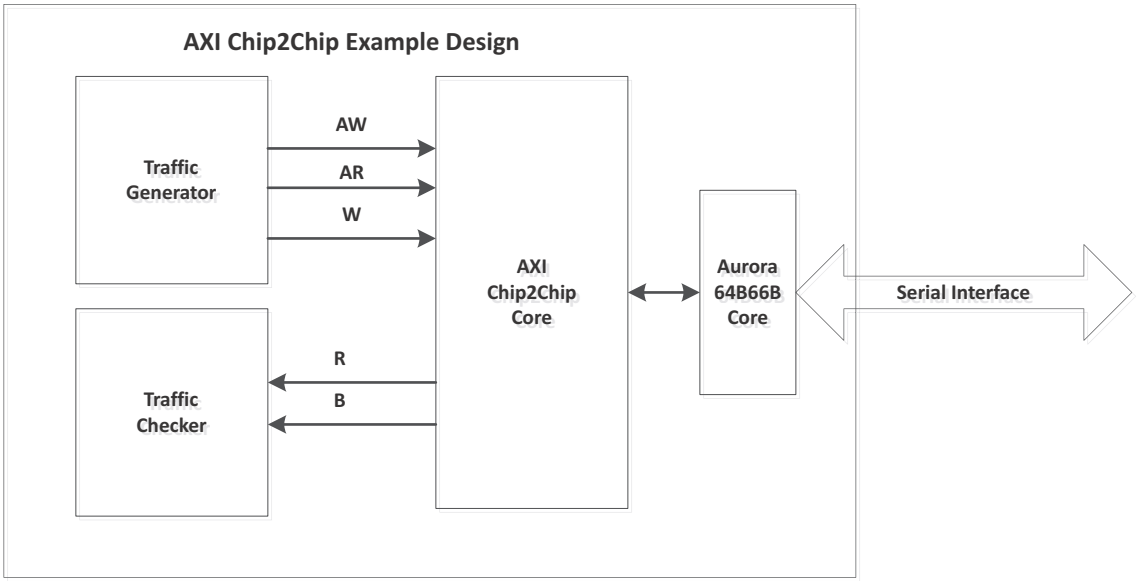

Figure 7-2: **Example Design Block Diagram with Aurora Interface**

The example design contains the following:

- An instance of the AXI Chip2Chip core
- Clocking wizard to generate clock signals for the example design
- Traffic generator for AXI4 and AXI4-Lite interfaces
- Traffic checker for AXI4 and AXI4-Lite interfaces
- When Aurora 64B/66B interface is chosen, the example design includes an instance of the Aurora 64B66B core in duplex configuration.

Implementing the Example Design

Depending on the board selected, provide XDC constraints for the system clock pins and SelectIO pins of AXI Chip2Chip core, or set the location constraints of the Aurora interface pins and transceivers. The status signals (Link Status, Multi-Bit Error, Configuration Error and Link Error) can be mapped to LEDs to show the status of the AXI Chip2Chip cores.

See the *AXI Chip2Chip Reference Design for Real-Time Video Application* (XAPP1160) [Ref 1] to set the SelectIO pin constraints for AXI Chip2Chip core on KC705 board.

Test Bench

This chapter contains information about the provided test bench in the Vivado® Design Suite environment.

Figure 8-1 and Figure 8-2 show the demonstration test bench with a SelectIO interface and an Aurora interface, respectively.

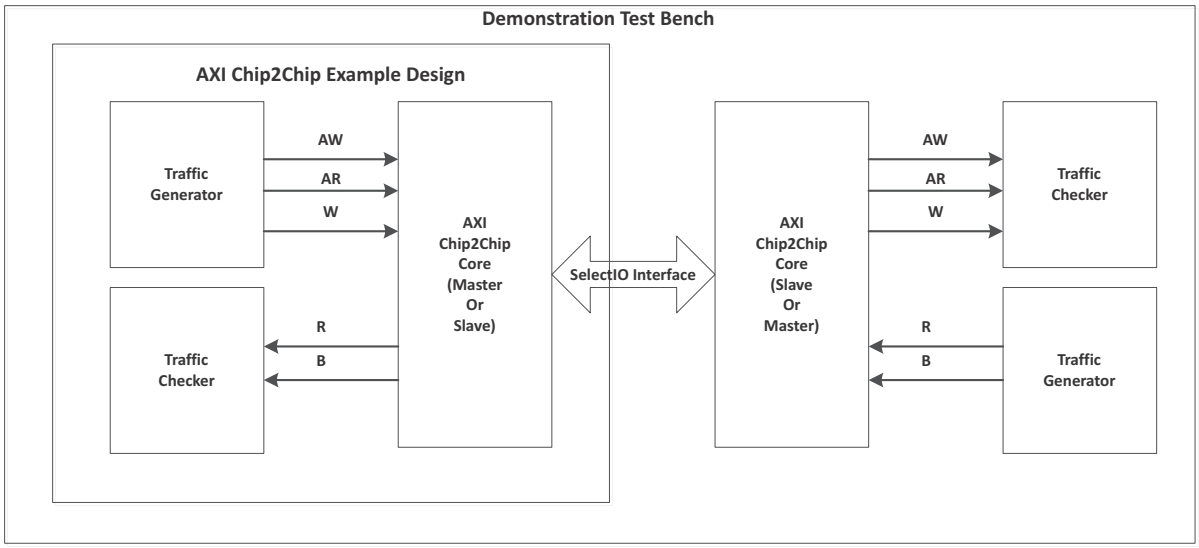

Figure 8-1: **Demonstration Test Bench Block Diagram**

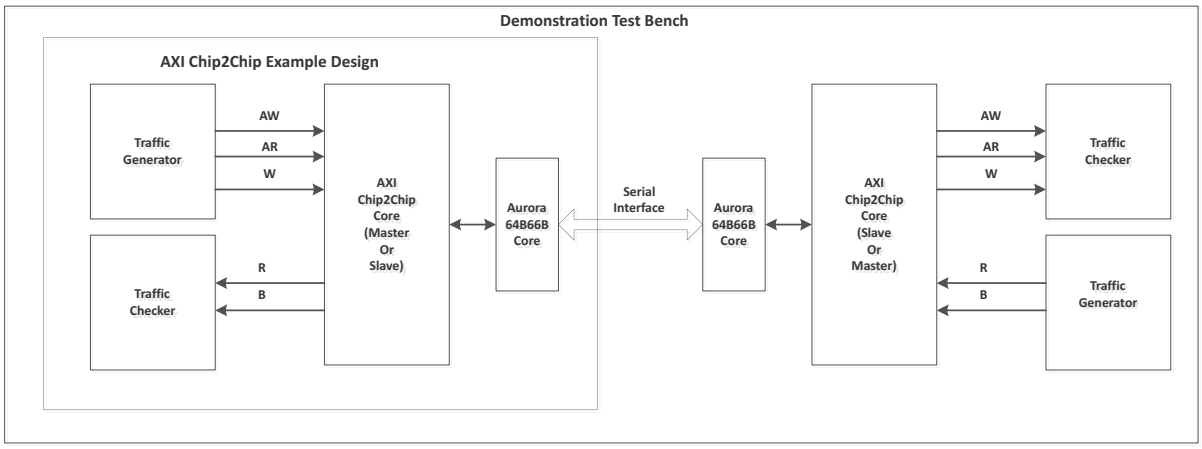

Figure 8-2: **Demonstration Test Bench with Aurora Interface**

To demonstrate the AXI Chip2Chip core, an instance of AXI Chip2Chip core in complementary mode is connected to the AXI Chip2Chip core in the example design.

The demonstration test bench performs the following tasks:

- Generates input clock signals.
- Applies a reset to the example design.
- Waits for one of the interrupt signals (Link Status, Configuration Error (Aurora PHY) and Multi-Bit Error) to be asserted. If Link status is asserted, a stable link is established between the Master and Slave AXI Chip2Chip cores. If Configuration Error or Multi-Bit Error is asserted, the test bench fails with `Error: Link Not Detected`.
- If a link is successfully established, `Link detected` is displayed in the console.
- The traffic generator starts generating fixed traffic patterns at the inputs of the AXI Chip2Chip cores.
- The traffic checker checks the output signals of the AXI Chip2Chip cores against expected patterns. If the received data has an error, then error messages are issued at the console with the name, expected value and actual value of the signal in error condition.
- The transactions are shown for a time interval of 10000 ns and the test bench finishes with the `Test Completed Successfully` in the console.

Appendix A

Verification, Compliance, and Interoperability

This appendix includes information about how the IP was tested for compliance with the protocol to which it was designed.

Simulation

AXI Chip2Chip cores have been tested with Xilinx Vivado Design Suite and the Mentor Graphics Questa SIM simulator. For the supported versions of these tools, see the Xilinx Design Tools: Release Notes Guide.

For more details about simulating your design, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 2].

The IP is tested using Xilinx proprietary standard AXI Memory Mapped OVM Verification Components (OVCs).

Hardware Testing

Figure A-1 shows the hardware testing setup for the AXI Chip2Chip core.

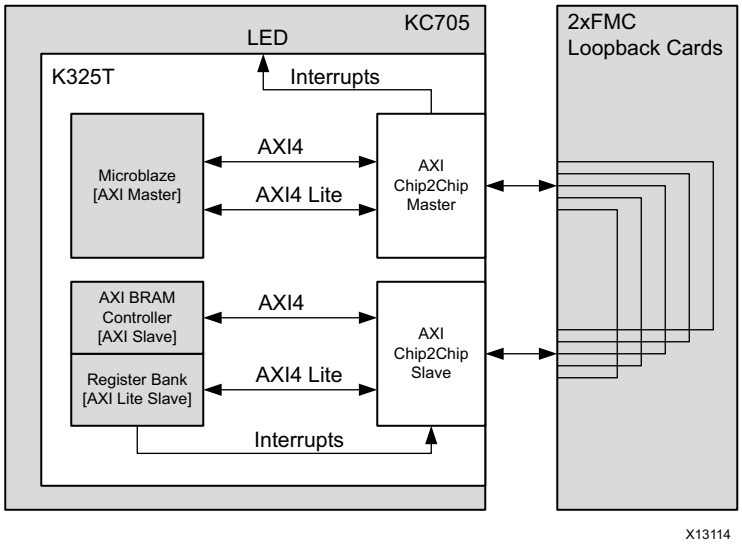

Figure A-1: **AXI Chip2Chip Hardware Testing Setup**

The AXI Chip2Chip core with a SelectIO FPGA interface has been hardware validated on a KC705 board using a Kintex™-7 FPGA with -1 speed grade (325T). The setup uses two additional FMC loopback cards. Table A-1 provides configuration details for the AXI Chip2Chip core and the frequency achieved by utilizing this setup with the SelectIO interface.

Table A-1: **Hardware Testing Configuration with a SelectIO FPGA interface**

Features			I/Os Utilized	PHY Clock (MHz)
AXI Data Width	**Chip2Chip PHY Type**	**Chip2Chip PHY Width**	**Single Ended [HR I/O Banks]**	**LVCMOS_25 IO [Unterminated]**
32-bit	SelectIO SDR	Compact 4:1	38	200
	SelectIO DDR	Compact 1:1	58	100
	SelectIO DDR	Compact 2:1	32	150
	SelectIO DDR	Compact 4:1	20	150
64-bit	SelectIO DDR	Compact 2:1	46	100
	SelectIO DDR	Compact 4:1	28	150

Notes:

1. The AXI (system) clock frequency was set to 100 MHz, and the Common Clock mode of operation was selected for configurations having the same PHY clock and AXI clock frequencies (100 MHz).

In addition, XAPP1160 provides a setup demonstrating real-time video traffic across Kintex-7 FPGA boards (KC705) and Zynq-7000 devices [Ref 1]. This setup uses the AXI Chip2Chip core for connectivity across the FPGA using LPC/HPC connector cables.

The AXI Chip2Chip core with an Aurora FPGA interface has been validated on two KC705 boards. Table A-2 and Table A-3 provide the configuration details of AXI Chip2Chip and Aurora 64B/66B cores.

Table A-2: **Hardware Testing Configuration with an Aurora Interface**

AXI Data Width	Chip2Chip PHY Type	Chip2Chip PHY Width	Aurora Data Width	Line rate
32-bit	Aurora PHY	Compact 1:1	64	3.125 Gbps
64-bit	Aurora PHY	Compact 2:1	64	3.125 Gbps

Notes:

1. The AXI clock frequency was set to 100 MHz and Independent Clock mode of operation was selected for Slave AXI Chip2Chip core.

Table A-3: **Hardware Testing Configuration for Aurora Interface**

Parameter	Value
GT Refclk	125 MHz
Line rate	3.125 Gbps
Dataflow Mode	Duplex
Interface	Streaming
Flow control	None
DRP Mode	Native
User K	Unchecked
Vivado Lab Tools	Unchecked
Lanes	1
GT Type	gtx

Migrating and Upgrading

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 5].

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Parameter Changes

When upgrading from v4.0 to v4.1, C_AXI_SIZE_WIDTH is set to value 3 from 2.

Port Changes

When upgrading from v4.0 to v4.1, `s_axi_awsize` and `s_axi_arsize` is set to width 3 from 2.

Debugging

This appendix provides information for using the resources available on the Xilinx Support website, debug tools, and other step-by-step processes for debugging designs that use the AXI Chip2Chip core.

Finding Help on Xilinx.com

To help in the design and debug process when using the AXI Chip2Chip core, the Xilinx Support web page (www.xilinx.com/support) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the AXI Chip2Chip core. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page (www.xilinx.com/support) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Design Tools tab on the Downloads page (www.xilinx.com/download). For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can also be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as:

- Product name
- Tool message(s)

- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the AXI Chip2Chip

AR: 54806

Contacting Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

Xilinx provides premier technical support for customers encountering issues that require additional assistance. To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Additional Resources.

When opening a WebCase, include:

- Target FPGA including package and speed grade.
- All applicable Xilinx Design Tools and simulator software versions.
- Additional files based on the specific issue might also be required. See the relevant sections in this debug guide for guidelines about which file(s) to include with the WebCase.

Note: Access to WebCase is not available in all cases. Login to the WebCase tool to see your specific support options.

Debug Tools

There are many tools available to address AXI Chip2Chip core design issues. It is important to know which tools are useful for debugging various situations.

Vivado Lab Tools

Vivado® lab tools insert logic analyzer and virtual I/O cores directly into your design. Vivado lab tools allow you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature

represents the functionality in the Vivado IDE that is used for logic debugging and validation of a design running in Xilinx devices in hardware.

The Vivado logic analyzer is used to interact with the logic debug LogiCORE IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 6].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB or connector cable issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `locked` port.

Core-Specific Checks

The following checks can further the debugging process:

- Check that the `axi_c2c_multi_bit_error_out` signals of both the Master and Slave cores are not asserted. The `axi_c2c_link_status_out` signal should be asserted High once the cores are calibrated.
- If the Slave is reset during normal operation (`axi_c2c_link_error_out`), reset the entire Master-Slave system once.
- After downloading the software in any of the boards, reset the entire system. If there is no reset propagation, the Slave needs to be reset first, followed by the Master.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. Output `s_axi_arready` asserts when the read address is valid, and output `s_axi_rvalid` asserts when the read data/response is valid. If the interface is unresponsive, ensure that the following conditions are met:

- The interface is enabled, and `s_axi_lite_aclk`/`m_axi_lite_aclk` are stable when the core is brought out of reset.
- The interface is not being held in reset, and the link status output of the core is asserted.
- The other interface inputs and outputs are connected and toggling.
- The main core clocks are toggling and the link error or multi-bit error interrupt outputs of the core are not asserted.
- If the simulation has been run, verify in simulation that the waveform is correct for accessing the AXI4-Lite interface.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

This section provides supplemental material useful with this product guide:

1. *AXI Chip2Chip Reference Design for Real-Time Video Application* (XAPP1160)
2. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
3. *Vivado Design Suite User Guide: Implementation* (UG904)
4. *Vivado Design Suite User Guide: Designing with IP* (UG896)
5. *ISE to Vivado Design Suite Migration Methodology Guide* (UG911)
6. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
9. AMBA® AXI4 specification

   http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0022d/index.html

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/25/12	1.0	Xilinx Beta release.
10/16/12	2.0	Xilinx initial release. Updated core to v2.00a and ISE Embedded Development Kit (EDK) to v14.3.
10/25/12	2.1	Corrected typo in Figure 1-1.
12/18/12	3.0	• Updated core to v3.00a and ISE Embedded Development Kit (EDK) to v14.4. • Added support for AXI4-Lite. • Added Appendix C, Debugging.
03/30/13	4.0	• Updated core to v4.0. • Added support for Vivado Design Suite. • Removed support for ISE Embedded Development Kit (EDK).
10/02/13	4.1	• Updated core to v4.1. • Added Example Design and Test Bench chapters. • Added the Migrating and Updating appendix. • Added support for IP Integrator. • Changed all signals and ports to lowercase.
12/18/13	4.1	• Added support for Aurora interface.

Notice of Disclaimer