

Introduction

Using an AXI-to-AXI Connector (axi2axi_connector), the slave interface of one AXI Interconnect module can be connected to the master interface of another AXI Interconnect with no intervening logic. The axi2axi_connector IP provides the port connection points necessary to represent the connectivity in the system, plus a set of parameters used to configure the respective interfaces of the AXI Interconnect modules being connected.

Features

- Connects the master interface of one AXI Interconnect module to a slave interface of another AXI Interconnect module.
- Directly connects all master interface signals to all slave interface signals; contains no logic or storage.
- Modeled as a bus bridge in EDK.

Functional Description

The AXI slave interface of the axi2axi_connector (“connector”) module always connects to one attachment point (slot) of the master interface of one AXI Interconnect module (the “upstream interconnect”). The AXI master interface of the connector always connects to one slave interface slot of a different AXI Interconnect module (the “downstream interconnect.”) [Figure 1, page 2](#).

LogiCORE IP Facts Table					
Core Specifics					
Core Name	axi2axi_connector				
Supported Device Family ⁽¹⁾	Virtex®-6, Spartan®-6				
Supported User Interfaces	AXI3, AXI4-Lite, AXI4				
	Resources				Frequency
Configuration	LUTs	FFs	DSP Slices	Block RAMs	Max. Freq.
Config1	0	0	0	0	N/A
Provided with Core					
Documentation	This document				
Design Files	Verilog, VHDL				
Example Design	Figure 1, page 2				
Test Bench	N/A				
Constraints File	N/A				
Simulation Model	Verilog, VHDLI				
Tested Design Tools					
Design Entry Tools	XPS				
Simulation	N/A				
Synthesis Tools	XST				
Support					
Provided by Xilinx, Inc.					

1. For a complete listing of supported devices, see the release notes for this core and www.xilinx.com/ise/embedded/ddsupport.htm for derivative device support.

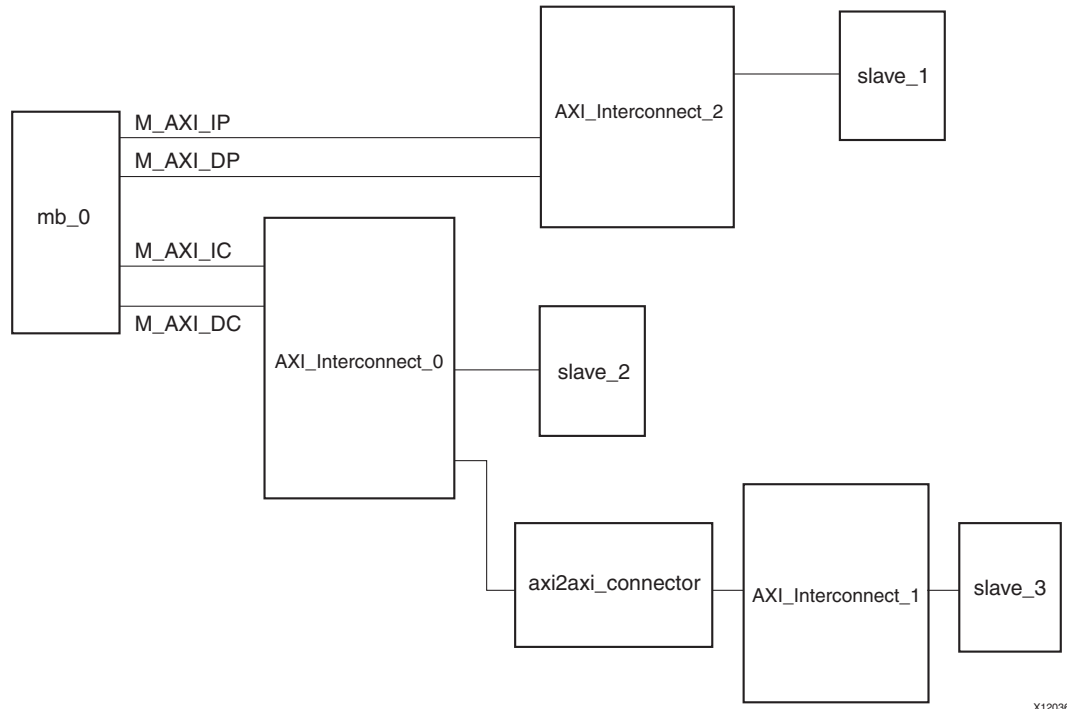


Figure 1: System using axi2axi_connector to Cascade Two AXI Interconnects

Application Details

When using an axi2axi_connector (to cascade two AXI Interconnects), the EDK tools set the data width and clock frequency parameters on the axi2axi_connector IP so that the characteristics of the master and slave interfaces match.

Also, the EDK tools auto-connect the clock port of the axi2axi_connector so that the interfaces of the connected interconnect modules are synchronized by the same clock source. Assuming the first interconnect (AXI_Interconnect0) is connected to the second interconnect (AXI_Interconnect1) using an axi2axi_connector, this is done based on the following rules:

1. If the native internal data width (C_INTERCONNECT_DATA_WIDTH) of AXI_Interconnect0 is equal to the native data width of AXI_Interconnect1, then:
 - a. Set the data width of the axi2axi_connector module (both master and slave interfaces) to the native data width of the two interconnects.
 - b. Connect the clock port of the axi2axi_connector to the native clock port (INTERCONNECT_ACLK) of AXI_Interconnect0 or AXI_Interconnect1, whichever has the lower frequency.
2. If the native internal data widths of AXI_Interconnect0 and AXI_Interconnect1 are not equal, then:
 - a. Set the data width of the axi2axi_connector module (both interfaces) to match the interconnect which has smaller native data width.
 - b. Connect the clock port of the axi2axi_connector to the native clock port of AXI_Interconnect0 or AXI_Interconnect1, whichever has the smaller native data width. This ensures that clock conversion is always performed in the wider interconnect so that data bandwidth is never reduced needlessly. The axi2axi_connector contains slave and master interface I/O signals.

I/O Signals

The axi2axi_connector contains one AXI slave interface and one AXI master interface. Each AXI interface contains a super-set of signals supporting AXI4, AXI3, and AXI4-Lite protocols.

Global I/O Signals

Table 1: Global I/O Signals

Signal Name	Interface	Signal Type	Description
Global Signals			
ACLK	Global	I	AXI bus clock. Used only to establish clock connectivity to connected AXI Interconnect interfaces.
ARESETN	Global	I	AXI active-Low reset. (Not used by axi2axi_connector.)

Slave I/O Signals

The following table provides the axi2axi_connector I/O slave signals.

Table 2: I/O Slave Signals

Signal Name	Interface	Signal Type	Description
AXI Write Address Channel Signals (AW)			
S_AXI_AWID [C_S_AXI_ID_WIDTH-1:0]	AW	I	AXI address Write ID.
S_AXI_AWADDR [C_S_AXI_ADDR_WIDTH-1:0]	AW	I	AXI Write address.
S_AXI_AWLEN [7:0]	AW	I	AXI address write burst length.
S_AXI_AWSIZE [2:0]	AW	I	AXI address write burst size.
S_AXI_AWBURST [1:0]	AW	I	AXI address write burst type.
S_AXI_AWLOCK	AW	I	AXI write address lock signal.
S_AXI_AWCACHE [3:0]	AW	I	AXI write address cache control signal.
S_AXI_AWPROT [2:0]	AW	I	AXI write address protection signal.
S_AXI_AWREGION [3:0]	AW	I	Channel address region index
S_AXI_AWQOS [3:0]	AW	I	Channel Quality of Service (QoS).
S_AXI_AWUSER [C_S_AXI_AWUSER_WIDTH-1:0]	AW	I	User-defined AW Channel signals.
S_AXI_AWVALID	AW	I	AXI write address valid.
S_AXI_AWREADY	AW	O	AXI write address ready.
AXI Write Data Channel Signals (W)			
S_AXI_WID [C_S_AXI_ID_WIDTH-1:0]	W	I	AXI3 Write ID.
S_AXI_WDATA [C_S_AXI_DATA_WIDTH-1:0]	W	I	AXI write data.
S_AXI_WSTRB [C_S_AXI_DATA_WIDTH/8-1:0]	W	I	AXI write data strobes.
S_AXI_WLAST	W	I	AXI write data last signal. Indicates the last transfer in a write burst.
S_AXI_WUSER [C_S_AXI_WUSER_WIDTH-1:0]	W	I	User-defined W Channel signals.
S_AXI_WVALID	W	I	AXI write data valid.

Table 2: I/O Slave Signals (Cont'd)

Signal Name	Interface	Signal Type	Description
S_AXI_WREADY	W	O	AXI write data ready.
AXI Write Response Channel Signals (B)			
S_AXI_BID [C_S_AXI_ID_WIDTH-1:0]	B	O	AXI write response ID.
S_AXI_BRESP [1:0]	B	O	AXI write response code.
S_AXI_BUSER			
S_AXI_BVALID	B	O	AXI write response valid.
S_AXI_BREADY	B	I	Write response ready.
AXI Read Address Channel Signals (AR)			
S_AXI_ARID [C_S_AXI_ID_WIDTH-1:0]	AR	I	AXI address read ID.
S_AXI_ARADDR [C_S_AXI_ADDR_WIDTH-1:0]	AR	I	AXI read address.
S_AXI_ARLEN [7:0]	AR	I	AXI address read burst length.
S_AXI_ARSIZE [2:0]	AR	I	AXI address read burst size.
S_AXI_ARBURST [1:0]	AR	I	AXI address read burst type.
S_AXI_ARLOCK	AR	I	AXI read address lock signal.
S_AXI_ARCACHE [3:0]	AR	I	AXI read address cache control signal.
S_AXI_ARPROT [2:0]	AR	I	AXI read address protection signal.
S_AXI_ARREGION [3:0]	AR	I	Channel address region index.
S_AXI_ARQOS [3:0]	AR	I	Channel Quality of Service.
S_AXI_ARUSER [C_S_AXI_ARUSER_WIDTH-1:0]	AR	I	User-defined AR Channel signals.
S_AXI_ARVALID	AR	I	AXI read address valid.
S_AXI_ARREADY	AR	O	AXI read address ready.
AXI Read Data Channel Signals (B)			
S_AXI_RID [C_S_AXI_ID_WIDTH-1:0]	R	O	AXI read data response ID.
S_AXI_RDATA [C_S_AXI_DATA_WIDTH-1:0]	R	O	AXI read data.
S_AXI_RRESP [1:0]	R	O	AXI read response code.
S_AXI_RLAST	R	O	AXI read data last signal.
S_AXI_RUSER [C_S_AXI_RUSER_WIDTH-1:0]	R	O	User-defined R Channel signals.
S_AXI_RVALID	R	O	AXI read valid.
S_AXI_RREADY	R	I	Read ready.

Master I/O Signals

The following table contains the axi2axi_connector master I/O signals.

Table 3: I/O Master Signals

Signal Name	Interface	Signal Type	Description
AXI Write Address Channel Signals (AW)			
M_AXI_AWID [C_S_AXI_ID_WIDTH-1:0]	AW	O	AXI address write ID.
M_AXI_AWADDR [C_S_AXI_ADDR_WIDTH-1:0]	AW	O	AXI Write address.
M_AXI_AWLEN [7:0]	AW	O	AXI address write burst length.
M_AXI_AWSIZE [2:0]	AW	O	AXI address write burst size.
M_AXI_AWBURST [1:0]	AW	O	AXI address write burst type.
M_AXI_AWLOCK	AW	O	AXI write address lock signal.
M_AXI_AWCACHE [3:0]	AW	O	AXI write address cache control signal.
M_AXI_AWPROT [2:0]	AW	O	AXI write address protection signal.
M_AXI_AWREGION [3:0]	AW	O	Write Address Channel address region index.
M_AXI_AWQOS [3:0]	AW	O	Write Address Channel Quality of Service (QoS).
M_AXI_AWUSER [C_S_AXI_AWUSER_WIDTH-1:0]	AW	O	User-defined AW Channel signals.
M_AXI_AWVALID	AW	O	AXI write address valid.
M_AXI_AWREADY	AW	I	AXI write address ready.
AXI Write Data Channel Signals (W)			
M_AXI_WID [C_S_AXI_ID_WIDTH-1:0]	W	O	AXI3 write ID.
M_AXI_WUSER [C_S_AXI_WUSER_WIDTH-1:0]	W	O	User-defined W Channel signals.
M_AXI_WDATA [C_S_AXI_DATA_WIDTH-1:0]	W	O	AXI write data.
M_AXI_WSTRB [C_S_AXI_DATA_WIDTH/8-1:0]	W	O	AXI write data strobes.
M_AXI_WLAST	W	O	AXI write data last signal.
M_AXI_WVALID	W	O	AXI write data valid.
M_AXI_WREADY	W	I	AXI write data ready.
AXI Write Response Channel Signals (B)			
M_AXI_BID [C_S_AXI_ID_WIDTH-1:0]	B	I	AXI write data response ID.
M_AXI_BRESP [1:0]	B	I	AXI write response code.
M_AXI_BUSER [C_S_AXI_BUSER_WIDTH-1:0]	B	I	User-defined B Channel signals.
M_AXI_BVALID	B	I	AXI write response valid.
M_AXI_BREADY	B	O	Write response ready.
AXI Read Address Channel Signals (AR)			
M_AXI_ARID [C_S_AXI_ID_WIDTH-1:0]	AR	O	AXI address read ID.
M_AXI_ARADDR [C_S_AXI_ADDR_WIDTH-1:0]	AR	O	AXI read address.
M_AXI_ARLEN [7:0]	AR	O	AXI address read burst length.
M_AXI_ARSIZE [2:0]	AR	O	AXI address read burst size.
M_AXI_ARBURST [1:0]	AR	O	AXI address read burst type.

Table 3: I/O Master Signals (Cont'd)

Signal Name	Interface	Signal Type	Description
M_AXI_ARLOCK	AR	O	AXI read address lock signal.
M_AXI_ARCACHE [3:0]	AR	O	AXI read address cache control signal.
M_AXI_ARPROT [2:0]	AR	O	AXI read address protection signal.
M_AXI_ARREGION [3:0]	AR	O	Channel address region index.
M_AXI_ARQOS [3:0]	AR	O	AR Channel Quality of Service (QoS).
M_AXI_ARUSER [C_S_AXI_ARUSER_WIDTH-1:0]	AR	O	User-defined AR Channel signals.
M_AXI_ARVALID	AR	O	AXI read address valid.
M_AXI_ARREADY	AR	I	AXI read address ready. I
AXI Read Data Channel Signals (B)			
M_AXI_RID [C_S_AXI_ID_WIDTH-1:0]	R	I	AXI read data response ID.
M_AXI_RDATA [C_S_AXI_DATA_WIDTH-1:0]	R	I	AXI Read data.
M_AXI_RRESP [1:0]	R	I	AXI Read response code.
M_AXI_RLAST	R	I	AXI read data last signal.
M_AXI_RUSER [C_S_AXI_RUSER_WIDTH-1:0]	R	I	User-defined Channel Signals.
M_AXI_RVALID	R	I	AXI read valid.
M_AXI_RREADY	R	O	Read ready.

Design Parameters

The following tables contain the user-visible design parameters for the axi2axi_connector.

Table 4: axi2axi_Connector Design Parameters

Parameter Name	Default Value	Allowable Values	Description
C_S_AXI_PROTOCOL	AXI4	String (AXI3, AXI4, AXI4LITE)	Protocol conversion to be performed in MI of upstream interconnect. (No conversion if "AXI4".) Same value is observed by SI of downstream interconnect.
C_S_AXI_ADDR_WIDTH	32	constant (32)	Width of ADDR signals (both S and M interfaces).
C_S_AXI_DATA_WIDTH	N/A	Integer (32, 64, 128, 256)	Specifies data-width conversions to be performed in MI of upstream interconnect or SI of downstream interconnect, if different than C_INTERCONNECT_DATA_WIDTH for each interconnect. Set by tools to match C_INTERCONNECT_DATA_WIDTH of either the upstream or downstream interconnect. (See "Application Details, page 2.") User cannot override. (Same value is observed by both interconnects.)
C_S_AXI_ID_WIDTH	N/A	Integer	Width of all ID signals. Set by tools based on ID width of upstream interconnect. (User cannot override.)

Table 4: axi2axi_Connector Design Parameters (Cont'd)

Parameter Name	Default Value	Allowable Values	Description
C_S_AXI_SUPPORTS_READ	1	Integer (0, 1)	Indicates to both interconnects whether read transactions need to be propagated. Default value set by tools based on connectivity map of upstream interconnect; user may override. (Same value is observed by both interconnects.)
C_S_AXI_SUPPORTS_WRITE	1	Integer (0, 1)	Indicates to both interconnects whether write transactions need to be propagated. Default value set by tools based on connectivity map of upstream interconnect; user may override. (Same value is observed by both interconnects.)
C_S_AXI_SUPPORTS_USER_SIGNALS	0	Integer (0, 1)	Indicates whether USER signals need to be propagated. (Same value is observed by both interconnects.)
C_S_AXI_AWUSER_WIDTH	1	Integer (1-2147483647)	Width of AWUSER signals (both S and M interfaces; valid only when C_S_AXI_SUPPORTS_USER_SIGNALS = 1.)
C_S_AXI_ARUSER_WIDTH	1	Integer (1-2147483647)	Width of ARUSER signals (both S and M interfaces; valid only when C_S_AXI_SUPPORTS_USER_SIGNALS = 1)
C_S_AXI_BUSER_WIDTH	1	Integer (1-2147483647)	Width of BUSER signals (both S and M interfaces; valid only when C_S_AXI_SUPPORTS_USER_SIGNALS = 1)
C_S_AXI_RUSER_WIDTH	1	Integer (1-2147483647)	Width of RUSER signals (both S and M interfaces; valid only when C_S_AXI_SUPPORTS_USER_SIGNALS = 1)
C_S_AXI_WUSER_WIDTH	1	Integer (1-2147483647)	Width of WUSER signals (both S and M interfaces; valid only when C_S_AXI_SUPPORTS_USER_SIGNALS=1)
C_S_AXI_SUPPORTS_NARROW_BURST	1	Integer (0, 1)	Indicates to both interconnects whether narrow burst transactions are expected to be propagated. Default value set by tools based on connectivity map of upstream interconnect; user may override. (Same value is observed by both interconnects.)
C_S_AXI_NUM_ADDR_RANGES	1	Integer (1-16)	Specifies the number of address ranges.
C_S_AXI_RNG0-15_BASEADDR	0xFFFFFFFF	0x1000	Base Address 0-15.
C_S_AXI_RNG0-15_HIGHADDR	0x00000000	0x1000	High Address 0-15.

In addition to the parameters listed in this table, there are also inferred parameters for each AXI interface in the EDK tools. Through the design, these inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to each AXI interface, see the [AXI Interconnect IP Data Sheet](#) (DS768).

The tools set the default values of inferred parameters C_INTERCONNECT_S_AXI_WRITE_ACCEPTANCE and C_INTERCONNECT_S_AXI_READ_ACCEPTANCE:

- To 8, when C_S_AXI_PROTOCOL = "AXI4" or "AXI3"
- To 1, when C_S_AXI_PROTOCOL = "AXI4LITE"

The user can override each of these. The same values are then observed as inferred parameters C_INTERCONNECT_M_AXI_WRITE_ISSUING and C_INTERCONNECT_M_AXI_READ_ISSUING, respectively, by the SI of the downstream interconnect.

References

- <http://www.xilinx.com/ipcenter/axi4.htm>
- [AXI Interconnect IP Data Sheet](#) (DS768)
- [AXI Reference Guide](#) (UG761)

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
09/21/2010	1.0	Initial Xilinx release.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.