LogiCORE IP ChipScope AXI Monitor (v1.01a)

DS810 December 14, 2010 **Product Specification**

Introduction

The ChipScope™ AXI Monitor core was designed to monitor and debug AXI interfaces. The core allows probing any signals going from a peripheral to the AXI interconnect. For instance, the user can instantiate a monitor on a MicroBlaze™ instruction or data interface to observe all memory transactions going in and out of the processor.

Each monitor core works independently which allows the changing of trigger outputs to enable taking system-level measurements. By using the auxiliary trigger input and trigger output ports of a monitor core, the user can create multi-level triggering situations to simplify complex system level measurements. For instance, if a system consists of a master device operating at 100 MHz and a slave device operating at 50 MHz, a user can analyze the transfer of data going from one time domain to the next with the multi-tiered triggering functionality of monitor cores.

Moreover, with this system level measurement, the user is able to not only debug complex multi-time domain system level issues, but is able to analyze latency restrictions in a system as well.

Features

- Selectable data samples
- Generic Trigger/Data Unit with selectable width
- Auto-generated CDC file
- Multiple monitor support in single system through the use of trigger in and trigger out ports

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Virtex®-6(2), Spartan®-6(3)
Supported User Interfaces	AXI4, AXI4-Lite, AXI4-Stream
Resources Used	
N/A.	
Provided with Core	
Documentation	Product Specification, UG029 ChipScope Pro Software and Cores User Guide
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	N/A
Tested Design Tools	
Design Entry Tools	12.4 EDK
Simulation	N/A
Synthesis Tools	XST
Support	
Provided by Xilinx, Inc.	

Notes:

1. For a listing of supported devices, see the release notes for this core.
2. For more information on the Virtex-6 devices, see the DS150 Virtex-6 Family Overview.
3. For more information on the Spartan-6 devices, see the DS160 Spartan-6 Family Overview.

Functional Description

The AXI Monitor can be used to debug top-level signals in a system that uses AXI4 protocol specifications by connecting it to an AXI Core in Xilinx Platform Studio. When placed in an AXI system, the connection of the AXI Monitor probes between the AXI Interconnect and the AXI Core. The signal list is shown in Table 1. The user can monitor either AXI Memory Map signals through the MON_AXI bus interface or AXI Streaming signals through the MON_AXI_S bus interface (but not both) with a single AXI Monitor core. Communication with the ILA core is conducted using a connection to the JTAG port through the ICON core. See Figure 1.

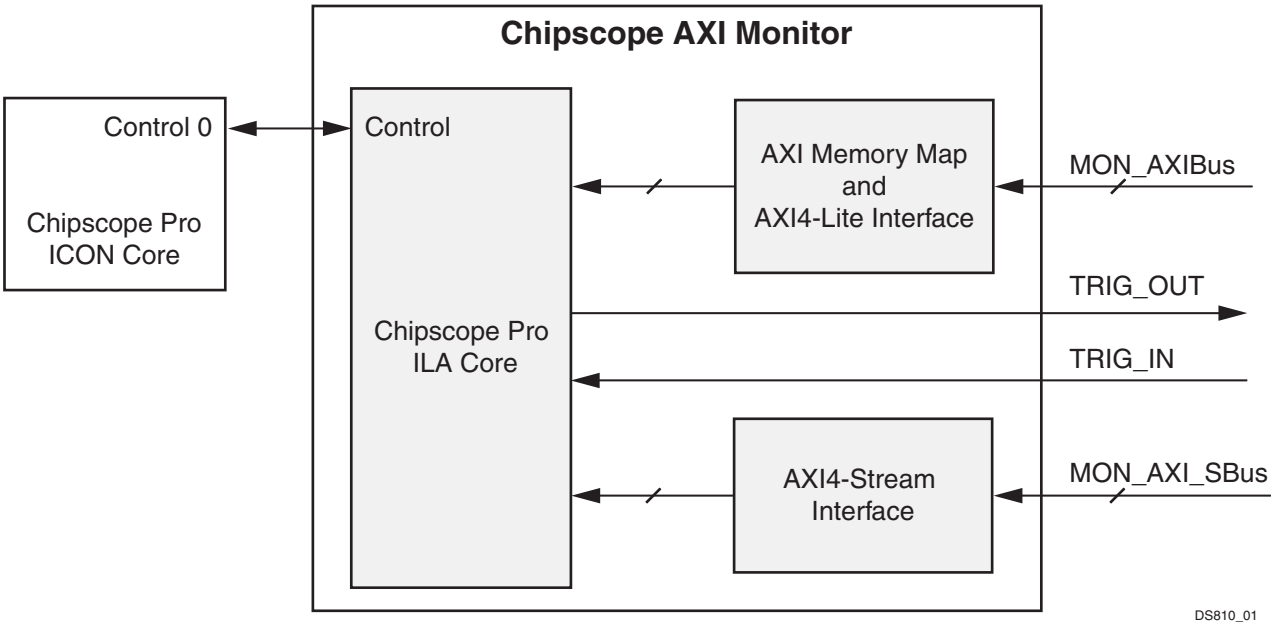

Figure 1: **ChipScope AXI Monitor Block Diagram**

The AXI Monitor is essentially a wrapper for the ChipScope ILA core. It functions the same way as the ChipScope ILA, except the wrapper creates a specific ILA for monitoring AXI signals by creating trigger groups designed to be useful for debugging purposes. When connected to a core in an AXI system, the user will specify which bus interface is to be connected (Memory Map or Streaming) and then supply values for parameters such as sample size and enable trigger out. These parameters are listed in Table 2.

After downloading the bitstream from the design to the FPGA, the ChipScope Analyzer tool is used to set up triggering and also view waveforms from the system. The core generates a CDC file which is used by ChipScope Analyzer to label the AXI signals with the appropriate header information and defines trigger groups. More information on the ChipScope ILA or the ChipScope Analyzer tool can be found in the DS299 ChipScope ILA document and the UG029 ChipScope Analyzer document.

Another feature of the AXI Monitor core is multiple AXI Monitors use in a single AXI system to monitor multiple cores or can be used together by creating a trigger condition for the TRIG_OUT port from one monitor then connecting it to the TRIG_IN port of another monitor. In Xilinx Platform Studio, when instantiating the AXI Monitor, the user simply needs to connect the trigger out and trigger in pins on separate monitors, then use ChipScope Analyzer to create the trigger condition on which one monitor will trigger the other to begin capturing data.

I/O Signals

Table 1: **I/O Signal Description**

Signal Name	Interface	Signal Type	Init Status	Description
AXI4 Memory Map Signals				
CHIPSCOPE_ICON_CONTROL(35:0)	N/A	I/O		Control bus connection to the ICON core. Mandatory. ***Note:*** For XPS designs, the direction of this port is IN.
MON_AXI_TRIG_OUT	N/A	O		Trigger output port. (Optional)
MON_AXI_TRIG_IN	N/A	I		Trigger input port. (Optional)
MON_AXI_ACLK	AXI4	I		Clock
MON_AXI_ARESETN	AXI4	I		Reset (active low)
MON_AXI_AWID(C_MON_AXI_ID_WIDTH-1:0)	AXI4	I		Write address channel transaction ID
MON_AXI_AWADDR(C_MON_AXI_ADDR_WIDTH-1:0)	AXI4	I		Write address channel address
MON_AXI_AWLEN(7:0)	AXI4	I		Write address burst length: Gives the exact number of transfers in a burst
MON_AXI_AWSIZE(2:0)	AXI4	I		Write address burst size: Indicates the size of each transfer in the burst
MON_AXI_AWBURST(1:0)	AXI4	I		Write address burst type
MON_AXI_AWLOCK	AXI4	I		Write address lock type
MON_AXI_AWCACHE(3:0)	AXI4	I		Write address cache type
MON_AXI_AWPROT(2:0)	AXI4	I		Write address protection type
MON_AXI_AWQOS(3:0)	AXI4	I		Write address channel quality of service
MON_AXI_AWREGION(3:0)	AXI4	I		Selects address range within multirange slave
MON_AXI_AWVALID	AXI4	I		Write address valid: Indicates a valid write address and control information is available
MON_AXI_AWREADY	AXI4	I		Write address ready: slave is ready to accept address and control information
MON_AXI_WID(C_MON_AXI_ID_WIDTH)-1:0)	AXI4	I		Write data channel transaction ID
MON_AXI_WDATA(C_MON_AXI_DATA_WIDTH-1:0)	AXI4	I		Write data bus
MON_AXI_WSTRB(C_MON_AXI_DATA_WIDTH/8)-1:0)	AXI4	I		Write strobes: Indicates which byte lanes have valid data. MON_AXI_WSTRB[n] corresponds to MON_AXI_WDATA[(8xn)]+7:(8xn)]
MON_AXI_WLAST	AXI4	I		Indicates last write data word
MON_AXI_WVALID	AXI4	I		Write valid: Indicated valid write data and strobes are available. 1 = write data and strobes available 0 = write data and strobes not available

Table 1: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
MON_AXI_WREADY	AXI4	I		Write ready: Indicates the slave can accept the write data 1= slave ready 0 = slave not ready
MON_AXI_BID(C_MON_AXI_ID_WIDTH-1:0)	AXI4	I		Write response channel ID
MON_AXI_BRESP(1:0)	AXI4	I		Write response: Indicates the status of the write transaction
MON_AXI_BVALID	AXI4	I		Write response valid: Indicates a valid write response is available 1= write response available 0 = write response not available
MON_AXI_BREADY	AXI4	I		Write response ready: Indicates the master can accept the response information 1 = master ready 0 = master not ready
MON_AXI_ARID(C_MON_AXI_ID_WIDTH-1:0)	AXI4	I		Read address ID
MON_AXI_ARADDR(C_MON_AXI_ADDR_WIDTH-1:0)	AXI4	I		Read address bus
MON_AXI_ARLEN(7:0)	AXI4	I		Read address burst length: Gives the exact number of transfers in a burst
MON_AXI_ARSIZE(2:0)	AXI4	I		Read address burst size: Indicates the size of each transfer in the burst
MON_AXI_ARBURST(1:0)	AXI4	I		Read address burst type
MON_AXI_ARLOCK	AXI4	I		Read address lock type
MON_AXI_ARCACHE(3:0)	AXI4	I		Read address cache type
MON_AXI_ARPROT(2:0)	AXI4	I		Read address protection type
MON_AXI_ARQOS(3:0)	AXI4	I		Read address channel quality of service
MON_AXI_ARREGION(3:0)	AXI4	I		Selects address range within multirange slave
MON_AXI_ARVALID	AXI4	I		Read address valid: When HIGH this signal indicates the read address and control information is valid and will remain valid until MON_AXI_ARREADY is HIGH 1 = Address and control information valid 0= Address and control information not valid
MON_AXI_ARREADY	AXI4	I		Address ready: Indicates the slave is ready to accept an address and associated control signals
MON_AXI_RID(C_MON_AXI_ID_WIDTH-1:0)	AXI4	I		Read data ID
MON_AXI_RDATA(C_MON_AXI_DATA_WIDTH-1:0)	AXI4	I		Read data bus
MON_AXI_RRESP(1:0)	AXI4	I		Read response: Indicates the status of the read transaction.
MON_AXI_RLAST	AXI4	I		Read last: Indicates last read data word

Table 1: **I/O Signal Description *(Cont'd)***

Signal Name	Interface	Signal Type	Init Status	Description
MON_AXI_RVALID	AXI4	I		Read data valid: Indicates the read data is available and the read transfer can complete 1 = read data available 0 = read data not available
MON_AXI_RREADY	AXI4	I		Read ready: Indicates the master can accept the read data and response information 1 = master ready 0 = master not ready
AXI4-Stream Signals				
MON_AXI_S_TVALID	AXI4-Stream	I		AXI4-Stream valid
MON_AXI_S_TREADY	AXI4-Stream	I		AXI4-Stream ready
MON_AXI_S_TDATA(C_MON_AXI_S_TDATA_WIDTH-1:0)	AXI4-Stream	I		AXI4-Stream data bus
MON_AXI_S_TSTRB(C_MON_AXI_S_TDATA_WIDTH/8-1:0)	AXI4-Stream	I		AXI4-Stream byte qualifier
MON_AXI_S_TLAST	AXI4-Stream	I		AXI4-Stream last word
MON_AXI_S_TID(C_MON_AXI_S_TID_WIDTH-1:0)	AXI4-Stream	I		AXI4-Stream ID
MON_AXI_S_TDEST(C_MON_AXI_S_TDEST_WIDTH-1:0)	AXI4-Stream	I		AXI-Stream destination
MON_AXI_S_TUSER(C_MON_AXI_S_TUSER_WIDTH-1:0)	AXI4-Stream	I		AXI-Stream user data

Design Parameters

Table 2: **Design Parameters**

Feature Description	Parameter Name	Allowable Values	Default Values	Type
User Specified AXI Implemented Parameters				
Active bus interface type	C_USE_INTERFACE	(0: AXI4/AXI4-Lite, 1: AXI4-Stream)	0	integer
Sets number of data samples	C_NUM_DATA_SAMPLES	(1024,2048,4096,8192,16384,32768,65536,131072)	1048	integer
Maximum number of sequencer levels	C_MAX_SEQUENCER_LEVELS	2	(1:16)	integer
Enable trigger in	C_USE_TRIG_IN	(0,1)	0	integer
Trigger input width	C_TRIG_IN_WIDTH	(1:255)	1	integer
User Specified AXI Implemented Parameters				
Device family	C_FAMILY	virtex6, spartan6	spartan6	string
Device/part	C_DEVICE		xc6slx45t	string
Device package	C_PACKAGE		fgg484	string
Device speed grade	C_SPEEDGRADE	(-1, -2, -3)	-2	string
AXI memory map BUSIF name	C_MON_AXI_PROTOCOL	axi4, axi4-lite	axi4	string

Table 2: **Design Parameters *(Cont'd)***

Feature Description	Parameter Name	Allowable Values	Default Values	Type
AXI stream BUSIF name	C_MON_AXI_S_PROTOCOL		generic	string
System supports threads	C_MON_AXI_SUPPORTS_THREADS	(0,1)	0	integer
AXI memory map ID width	C_MON_AXI_ID_WIDTH		1	integer
AXI stream ID width	C_MON_AXI_S_TID_WIDTH		1	integer
AXI memory map address width	C_MON_AXI_ADDR_WIDTH	32	32	integer
AXI stream TDEST width	C_MON_AXI_S_TDEST_WIDTH	(0:32)	32	integer
AXI4 memory map data width	C_MON_AXI_DATA_WIDTH	(32, 64, 128, 256)	32	integer
AXI stream data width	C_MON_AXI_S_TDATA_ WIDTH	(32, 64, 128, 256)	32	integer
Supports streaming TUSER bus	C_MON_AXI_S_TUSER_WIDTH	(0,1)	1	integer
Supports memory map USER buses (Note: this is left disabled)	C_MON_AXI_SUPPORTS_ USER_SIGNALS	(0,1)	0	integer
System supports read operations	C_MON_AXI_SUPPORTS_READ	(0,1)	1	integer
System supports write operations	C_MON_AXI_SUPPORTS_WRITE	(0,1)	1	integer

Device Utilization and Performance Benchmarks

The device utilization will vary based on the parameter combinations set by the system and user.

References

- More information on the ChipScope Pro software and cores is available in the *Software and Cores User Guide*, located at http://www.xilinx.com/documentation.
- Information about hardware debugging using ChipScope Pro in EDK is available in the Platform Studio 12 online help, located at http://www.xilinx.com/documentation.
- Information about hardware debugging using ChipScope Pro in System Generator for DSP is available in the *Xilinx System Generator for DSP User Guide*, located at http://www.xilinx.com/documentation.

Support

Xilinx provides technical support for this LogiCORE 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*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

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 modules and software, please contact your local Xilinx sales representative.

Revision History

Date	Version	Description of Revisions
9/21/10	1.0	Initial Xilinx Release
12/14/10	1.1	Added functionality to support AXI4-Lite and AXI4-Stream signals

Notice of Disclaimer