ChipScope Pro VIO (v. 1.00a, 1.01a, 1.02a, 1.03a)

DS284 September 16, 2009 **Product Specification**

Introduction

The Virtual Input/Output (VIO) core is a customizable core that can both monitor and drive internal FPGA signals in real time. Two different kinds of inputs and two different kinds of outputs are available, both of which are customizable in size to interface with the FPGA design.

Features

- Provides virtual LEDs and other status indicators through asynchronous and synchronous input ports
- Has activity detectors on input ports to detect rising and falling transitions between samples
- Provides virtual buttons and other controls through asynchronous and synchronous output ports
- For synchronous outputs, provides ability to define a pulse train, which is a 16-cycle train of ones and zeros that run at design speed.

For more information about the VIO core, refer to the *ChipScope Pro Software and Cores User Guide*.

LogiCORE IP Facts				
Core Specifics				
Supported Device Family(1)	Spartan®-3, Spartan-3E, Spartan-3A, Spartan-3A DSP, Spartan-6, Virtex®-4, Virtex-5, Virtex-6			
Resources Used(2)	I/O	LUTs	FFs	Block RAMs
	0	124	264	0
Special Features	N/A			
Provided with Core				
Documentation	Product Specification			
Design File Formats	N/A			
Constraints File	N/A			
Verification	N/A			
Instantiation Template	Verilog and VHDL Wrapper			
Reference Designs /Application Notes	None			
Additional Items	Signal Description File (.cdc)			
Design Tool Requirements				
Xilinx® Implementation Tools	ISE® 11.1			
Verification	ChipScope™ Pro 11.1			
Simulation	Not supported in simulation			
Synthesis	Netlist is pre-synthesized by XST			
Support				
Provided by Xilinx, Inc.				

1. Including the variants of these FPGA device families.
2. These estimates assume Virtex-4 device family with one 8-bit wide bank.

Applications

The VIO core is designed to be used in any application that requires verification or debugging using the ChipScope Pro software and cores.

Functional Description

The VIO core is a customizable core that can both monitor and drive internal FPGA signals in real time. Unlike the ILA and IBA cores, no on- or off-chip RAM is required. Communication with the VIO core is conducted using a connection to the JTAG port via the ICON core, as shown in Figure 1.

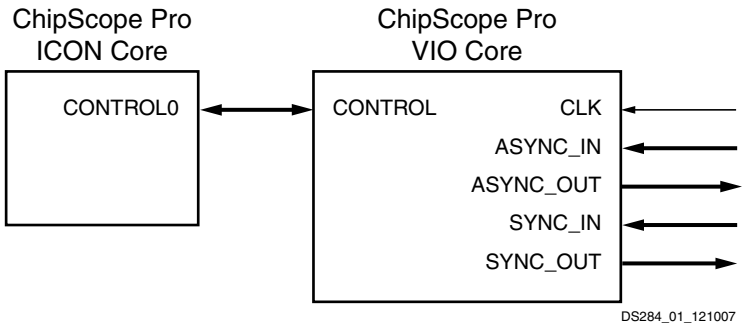

Figure 1: **VIO Core Connection to ICON Core**

Four types of signals are available in the VIO core:

- Asynchronous inputs:

  These are sampled using the JTAG clock signal that is driven from the JTAG cable. The input values are read back periodically and displayed in the Analyzer.
- Synchronous inputs:

  These are sampled using the design clock. The input values are read back periodically and displayed in the Analyzer.
- Asynchronous outputs:

  These are user-defined in the Analyzer and driven out of the core to the surrounding design. A logical 1 or 0 value can be defined for individual asynchronous outputs.
- Synchronous outputs:

  These are user-defined in the Analyzer, synchronized to the design clock and driven out of the core to the surrounding design. A logical 1 or 0 can be defined for individual synchronous outputs. Pulse trains of 16 clock cycles worth of ones and zeros can also be defined for synchronous outputs.

Activity Detectors

Every VIO core input has additional cells to capture the presence of transitions on the input. Since the design clock will most likely be much faster than the sample period of the Analyzer, it's possible for the signal being monitored to transition many times between successive samples. The activity detectors capture this behavior and the results are displayed along with the value in the Analyzer.

In the case of a synchronous input, activity cells capable of monitoring for asynchronous and synchronous events are used. This feature can detect glitches as well as synchronous transitions on the synchronous input signal.

Pulse Trains

Every VIO synchronous output has the ability to output a static 1, a static 0, or a pulse train of successive values. A pulse train is a 16-clock cycle sequence of 1s and 0s that drive out of the core on successive design clock cycles. The pulse train sequence is defined in the Analyzer and is executed only one time after it is loaded into the core.

VIO Interface Ports

The I/O signals of the VIO core consist of the control bus to ICON, as well as the four interface ports and the design clock. None of the ports are required, but at least one port must be enabled.

Table 1: **VIO Interface Ports**

Port Name	Direction	Description
ASYNC_IN[<*m*>-1:0]	IN	Asynchronous input port of width <*m*>. Optional (depends on enable_asynchronous_input_port). You must declare this port as a vector. For a one-bit port, use ASYNC_IN[0:0].
ASYNC_OUT[<*m*>-1:0]	OUT	Asynchronous output port of width <*m*>. Optional (depends on enable_asynchronous_output_port). You must declare this port as a vector. For a one-bit port, use ASYNC_OUT[0:0].
CLK	IN	Design clock used to register synchronous input or output ports. Optional (depends on enable_synchronous_input_port and/or enable_synchronous_output_port)
CONTROL[35:0]	INOUT	Control bus to ICON core. Mandatory.
SYNC_IN[<*m*>-1:0]	IN	Synchronous input port of width <*m*>. Optional (depends on enable_synchronous_input_port). You must declare this port as a vector. For a one-bit port, use SYNC_IN[0:0].
SYNC_OUT[<*m*>-1:0]	OUT	Synchronous output port of width <*m*>. Optional (depends on enable_synchronous_output_port). You must declare this port as a vector. For a one-bit port, use SYNC_OUT[0:0].

VIO XCO Parameters

Table 2: **VIO XCO Parameters**

Parameter Name	Allowable Values	Default Value	Description
component_name	String with A-z, 0-9, and _ (underscore)	vio	Name of instantiated component
enable_asynchronous_input_port	true, false	false	Enables the ASYNC_IN port
enable_asynchronous_output_port	true, false	false	Enables the ASYNC_OUT port
enable_synchronous_input_port	true, false	false	Enables the SYNC_IN port
enable_synchronous_output_port	true, false	false	Enables the SYNC_OUT port
asynchronous_input_port_width	1-256	8	Width of the ASYNC_IN port
asynchronous_output_port_width	1-256	8	Width of the ASYNC_OUT port
synchronous_input_port_width	1-256	8	Width of the SYNC_IN port
synchronous_output_port_width	1-256	8	Width of the SYNC_OUT port
invert_clock_input	true, false	false	Invert the design clock input

Restrictions

A maximum of 15 VIO cores can be used in a single design.

Using the VIO Core in EDK

The VIO core can be inserted into an embedded processor design using the EDK. In this case, the VIO core depends on ICON and OPB_MDM component instances already being in the design, as shown in Figure 2.

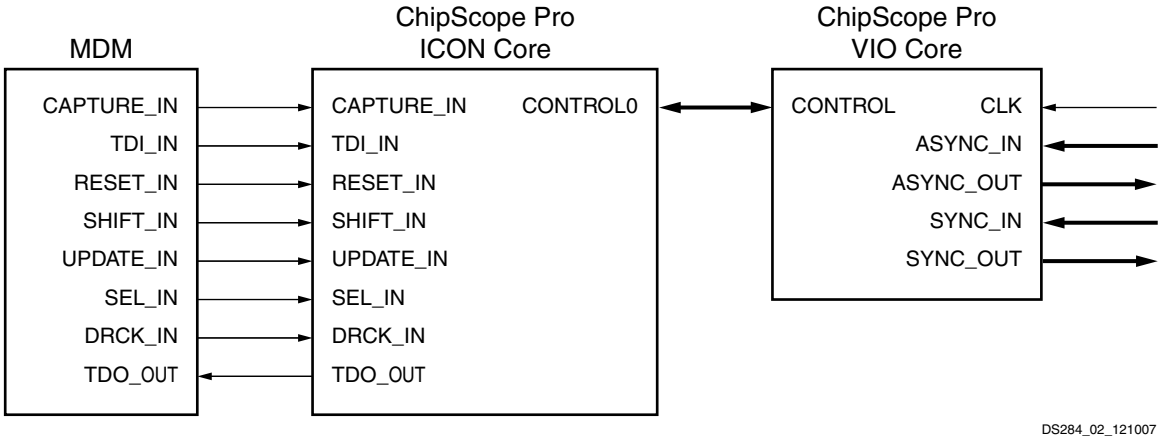

Figure 2: **VIO Core Component in EDK Design**

In EDK, the VIO core is integrated into the tool using a Tcl script. When the EDK platgen tool is run, the Tcl script is called and the script internally calls CORE Generator in command line mode. The Tcl script provides CORE Generator an arguments file (.xco) to generate the VIO core netlist. The Tcl script also generates an HDL wrapper to match the VIO ports based on the core parameters found in Table 3.

Table 3: **VIO EDK-specific Parameters**

Parameter Name	Allowable Values	Default Value	Description
c_async_input_enable	0, 1	0	0 = disable port, 1 = enable port
c_async_input_width	1-256	8	Asynchronous input port width, if used
c_async_output_enable	0, 1	0	0 = disable port, 1 = enable port
c_async_output_width	1-256	8	Asynchronous output port width, if used
c_family	virtex4, virtex5, virtex6, virtex6l, spartan3, spartan3E, spartan3A, spartan3Adsp, spartan6	N/A	Device family to use
c_rising_clock_edge	0, 1	1	Edge of input clock to use. 0 = falling edge, 1 = rising edge
c_sync_input_enable	0, 1	0	0 = disable port, 1 = enable port
c_sync_input_width	1-256	8	Synchronous input port width, if used
c_sync_output_enable	0, 1	0	0 = disable port, 1 = enable port
c_sync_output_width	1-256	8	Synchronous output port width, if used
c_use_srl16s	0, 1	1	0 = do no use SRL16s, 1 = use SRL16s

The XST synthesis tool is used for synthesizing the wrapper HDL generated for the VIO core. The NGC netlist outputs from XST and ChipScope Pro Core Generator are subsequently incorporated into the Xilinx ISE tool suite for actual device implementation.

Verification

Xilinx has verified the VIO core in a proprietary test environment, using an internally developed bus functional model.

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 Studdio 11.1 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

The VIO core is provided under the ISE Design Suite End-User License Agreement and can be generated using the Xilinx CORE Generator system 11.1 or higher. The CORE Generator system is shipped with Xilinx ISE Design Suite development software.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
03/24/2008	1.0	Release 10.1 (Initial Xilinx release).
09/19/2008	2.0	Release 10.1, Service Pack 3.
04/07/2009	3.0	Release 11.1.
06/24/2009	3.1	Release 11.2.
09/16/2009	3.1.1	Corrections to document. Published with 11.3 software release.

Notice of Disclaimer