

Introduction

The Chipscope™ Integrated Logic Analyzer (ILA) core is a customizable core that can be used to monitor internal FPGA signals in real time. The Chipscope ILA core in EDK is useful to monitor individual signals in an MHS-based embedded design. This core is based on Tcl script that generates a HDL wrapper for the ILA core and calls the Chipscope core generator to generate the netlist based on user parameters.

Features

- Multiple configurable ILA trigger units
 - Configurable Trigger input widths
 - Configurable Chipscope match types for use with different input signals types
- Separate Data and Trigger inputs
- Trigger out signal for cross-probing between Chipscope analyzer and processor debug tools, such as XMD/GDB

For more information, see the *Chipscope Pro Software and Cores User Guide* in the Chipscope Installation.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan™-II, Spartan-II-E, Spartan-3, Virtex™, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-E	
Version of Core	chipscope_ila	v1.00a
Resources Used		
	Min.	Max
Slices	150	2964
LUTs	70	131
FFs	146	4553
Block RAMs	1	17
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL/EDIF	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 6.3i or later	
Verification	Chipscope Pro 6.3i or later	
Simulation	Not Supported in Simulation	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

© 2005 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice. NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, 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.

Functional Description

The Chipscope Integrated Logic Analyzer (ILA) core is shown in the block diagram in [Figure 1](#).

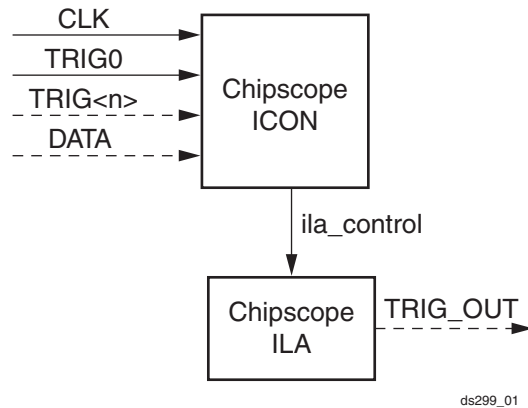


Figure 1: Chipscope ILA block diagram

ILA Parameters

To obtain a Chipscope ILA uniquely tailored for a system, certain features can be parameterized on the ILA. This feature allows the user to configure a design that utilizes only the resources required by the system, which results in the best possible performance. The Chipscope ILA features that can be parameterized are described in [Table 1](#).

Chipscope ILA peripheral can support as much as 4 individual trigger inputs that can be parameterizable separately. In the following table, C_TRIG<n>_UNIT refers to each of those trigger units, where <n> is 0 to 3.

Allowable Parameter Combinations

The parameter C_DATA_IN_WIDTH is valid only when the separate data input (not the same as trigger input) is enabled on the Chipscope ILA by specifying C_DATA_SAME_AS_TRIGGER = 0.

Parameters C_TRIG<n>_TRIGGER_IN_WIDTH, C_TRIG<n>_UNIT_COUNTER_WIDTH and C_TRIG<n>_UNIT_MATCH_TYPE are valid only when the corresponding trigger unit is enabled by setting C_TRIG<n>_UNITS to 1 or more.

For more information, refer the *Chipscope Pro Software and Cores User Guide*, in the Chipscope installation.

Table 1: Chipscope ILA Parameters

Parameter Name	Allowable Values	Default Value	VHDL Type	Feature / Description
C_NUM_DATA_SAMPLES	Integer (512, 1024, 2048, 4096, 8192, 16384)	512	integer	Number of Data Samples captured for every trigger match
C_ENABLE_TRIGGER_OUT	Integer 1 = Enable Trigger out 0 = Disable Trigger out	0	integer	Enable the Trigger out signal trig_out which will be asserted when ILA gets triggered
C_FAMILY	Xilinx FPGA families	virtex2	strings	Target Family
C_DATA_SAME_AS_TRIGGER	Integer 1 = Data same as Trigger 0 = Separate Data signal to the ILA	0	integer	Use the ILA Trigger signals as the ILA Data signals
C_DATA_IN_WIDTH	Integer (1-256)	32	integer	Width of the ILA Data input signal
C_DISABLE_RPM	Integer 1 = RPM disable 0 = RPM enabled in netlist	0	integer	Disable RPM placement information in netlist
C_TRIG<n>_UNITS Ex : C_TRIG0_UNITS	Integer (0-16) 0 = Disable Unit 1-16 = Number of Match Units	0	integer	Number of Match Units enabled for nth Trigger Unit. Each trigger unit has a separate trigger input.
C_TRIG<n>_TRIGGER_IN_WIDTH Ex : C_TRIG0_TRIGGER_IN_WIDTH	Integer (1-256)	8	integer	Width of trigger input for nth Trigger Unit
C_TRIG<n>_UNIT_COUNTER_WIDTH Ex : C_TRIG0_UNIT_COUNTER_WIDTH	Integer (0-32) 0 - Disable Match Counter 1-32 - Match Counter Width (Refer Chipscope User Guide)	0	integer	Counter Width for Match Unit <n>
C_TRIG<n>_UNIT_MATCH_TYPE Ex : C_TRIG<n>_UNIT_MATCH_TYPE	"basic", "basic with edges", "extended", "extended with edges", "range", "range with edges" (See the <i>Chipscope Pro Software and Cores User Guide</i>)	"basic"	string	Match Type for Match Unit <n>

Parameter - Port Dependencies

Table 2: Chipscope ILA Parameter - Port dependencies

Port Name	Parameter Dependency	Description
data	C_DATA_IN_WIDTH	Width of the data input port varies with this parameter when C_DATA_SAME_AS_TRIGGER is set to 0 (disabled)
trig<n>	C_TRIG<n>_IN_WIDTH	Width of the trigger(/data) input port varies with this parameter when the corresponding trigger unit is enabled by setting the C_TRIG<n>_UNITS parameter to 1 or more.
trig_out	C_ENABLE_TRIGGER_OUT	The trig_out port is enabled when this parameter is set to 1

Chipscope ILA I/O Signals

The I/O signals for the Chipscope ILA are shown in [Figure 1](#) and listed in [Table 3](#)

Table 3: Chipscope ILA I/O Signals

Signal Name	I/O	Description
chipscope_ila_control	I[0:35]	ILA control signals to be connected to the Chipscope ICON
clk	I	Clock for Trigger signals
data	I	Separate Chipscope ILA data input. This signal is valid only when C_DATA_SAME_AS_TRIGGER is disabled (= 0).
trig<n>	I	Trigger input to Nth Trigger Unit
trig_out	O	Trigger output from Chipscope ILA for cross-triggering processor debug tools like XMD/GDB by connecting this to the processor stop signal.

Chipscope ILA Module Register Descriptions

Not Applicable.

Chipscope ILA Module Interrupt Descriptions

Not Applicable.

Design Implementation

Design Tools

The **Chipscope ILA** design consists mainly of a Tcl script. When the EDK platgen tool is run, this Tcl script gets called and the script internally calls the Chipscope Pro Core Generator tool in commandline mode and provides it an arguments file (.arg) to generate the Chipscope ILA netlist. The Tcl script also generates a HDL wrapper to match the ILA ports based on the core parameters.

XST is the synthesis tool used for synthesizing the wrapper HDL generated for the Chipscope ILA. The EDIF netlist outputs from XST and Chipscope Core Generator are then input to the Xilinx Foundation tool suite for actual device implementation.

Target Technology

The intended target technology is all Xilinx FPGAs.

Device Utilization and Performance Benchmarks

The device utilization varies widely based on the parameter combinations set by the user.

Restrictions

Currently, only one Chipscope ILA peripheral can be instantiated in a single EDK design (MHS file). But, multiple trigger(/data) ports can be used to monitor many signals.

Using the Chipscope ILA wrapper in EDK, only signals at the MHS level can be monitored. But, the fpga_editor and fpga_edline tools can be used to monitor signals at any level of hierarchy in the design.

Reference Documents

Chipscope Pro Software and Cores Users Guide.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/16/04	1.0	Initial release.
8/17/04	1.1	Updated for Gmm; Assigned new document number, updated trademarks and supported device family listing, added version in title, made minor edits in content.
8/30/04	1.2	Added Block diagram, added parameter-port dependency
4/1/05	1.3	Updated for EDK 7.1.1 SP1; updated supported device family listing.
7/15/05	1.4	Converted to new DS template.