Integrated Logic Analyzer v5.1

LogiCORE IP Product Guide

Vivado Design Suite

PG172 April 1, 2015

XILINX
ALL PROGRAMMABLE™

Table of Contents

Appendix C: Debugging

Appendix D: Additional Resources and Legal Notices

Introduction

The customizable Integrated Logic Analyzer (ILA) IP core is a logic analyzer that can be used to monitor the internal signals of a design. The ILA core includes many advanced features of modern logic analyzers, including boolean trigger equations and edge transition triggers. Because the ILA core is synchronous to the design being monitored, all design clock constraints that are applied to your design are also applied to the components of the ILA core.

Features

- User-selectable number of probe ports and probe_width
- Multiple probe ports, which can be combined into a single trigger condition
- AXI interface on ILA IP core to debug AXI IP cores in a system

For more information about the ILA core, see the *Vivado® Design Suite User Guide: Programming and Debugging* (UG908) [Ref 1].

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	UltraScale™ Architecture, Zynq®-7000 All Programmable SoC, 7 Series
Supported User Interfaces	IEEE Standard 1149.1 – JTAG
Resources	See Table 2-1 and Table 2-2
Provided with Core	
Design Files	N/A
Example Design	Verilog
Test Bench	VHDL and Verilog
Constraints File	XDC
Simulation Model	Not Provided
Supported S/W Driver	Not Provided
Tested Design Flows(2)	
Design Entry	Vivado® Design Suite Vivado
Simulation	Not Provided
Synthesis(3)	Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.
3. The standard synthesis flow for Synplify is not supported for the core.

Overview

Feature Summary

Signals in the FPGA design are connected to ILA core clock and probe inputs (Figure 1-1). These signals, attached to the probe inputs, are sampled at design speeds and stored using on-chip block RAM (BRAM). The core parameters specify the number of probes, trace sample depth, and the width for each probe input. Communication with the ILA core is conducted using an auto-instantiated debug core hub that connects to the JTAG interface of the FPGA.

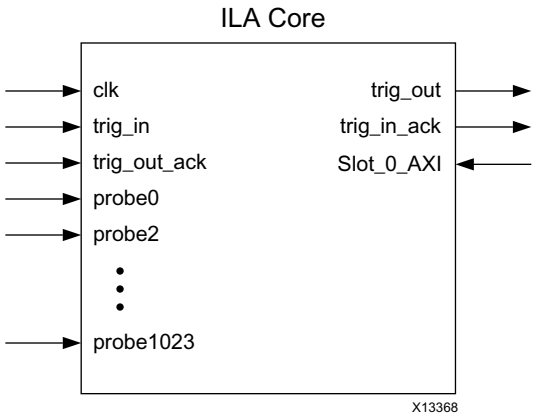

Figure 1-1: **ILA Core Symbol**

Note: The numerical range from probe3 to probe1022 is indicated by ellipses (...) in Figure 1-1.

After the design is loaded into the FPGA, use the Vivado® logic analyzer software to set up a trigger event for the ILA measurement. After the trigger occurs, the sample buffer is filled and uploaded into the Vivado logic analyzer. You can view this data using the waveform window.

Regular FPGA logic is used to implement the probe sample and trigger functionality. On-chip block RAM memory stores the data until it is uploaded by the software. No user input or output is required to trigger events, capture data, or to communicate with the ILA core.

ILA Probe Trigger Comparator

Each probe input is connected to a trigger comparator that is capable of performing various operations. At run time the comparator can be set to perform = or != comparisons. This includes matching level patterns, such as X0XX101. It also includes detecting edge transitions such as rising edge (R), falling edge (F), either edge (B), or no transition (N). The trigger comparator can perform more complex comparisons, including >, <, ≥, and ≤.

IMPORTANT: *Note that the comparator is set at run time through the Vivado logic analyzer.*

ILA Trigger Condition

The trigger condition is the result of a Boolean "AND" or "OR" calculation of each of the ILA probe trigger comparator result. Using the Vivado logic analyzer, you select whether to "AND" probe trigger comparators probes or "OR" them. The "AND" setting causes a trigger event when all of the ILA probe comparisons are satisfied. The "OR" setting causes a trigger event when any of the ILA probe comparisons are satisfied. The trigger condition is the trigger event used for the ILA trace measurement.

Applications

The ILA core is designed to be used in any application that requires verification or debugging using the Vivado logic analyzer.

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 about pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

License Checkers

If the IP requires a license key, the key must be verified. The Vivado design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following tools:

- Vivado design tools: Vivado Synthesis
- Vivado Implementation
- Bitstream Generation

IMPORTANT: *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

Product Specification

Performance

The ILA core can be configured to Select 1,024 probes each of width ranging from 1 to 4,096. This probe ports should be connected to user design signals which needs to be monitored in Vivado® logic analyzer during the run time.

Resource Utilization

Resources required for the ILA core have been estimated for UltraScale™ architecture and Kintex®-7 FPGAs (Table 2-1 and Table 2-2). These values were generated using Vivado IP catalog. They are derived from post-synthesis reports, and might change during implementation.

UltraScale FPGAs

Table 2-1 provides approximate resource counts for the various core options using UltraScale FPGAs.

Table 2-1: **Device Utilization – UltraScale FPGAs (XCU040-FFVA1156-3-E-ES1)**

Configuration	Slices	LUTs	Flip-Flops	Block RAMs
64 probe 8,192 data depth (Same data width – 1)	5,222	3,939	5,210	15
64 probe 8,192 data depth (Data width – 94)	5,358	4,078	5,346	22
64 probe 8,192 data depth (Variable data width)	7,891	5,760	7,879	162

Kintex-7 FPGAs

Table 2-2 provides approximate resource counts for the various core options using Kintex-7 FPGAs.

Table 2-2: **Device Utilization – Kintex-7 FPGAs**

Configuration	Slices	LUTs	Flip-Flops	Block RAMs
64 probe 8,192 data depth (Same data width – 1)	5,760	4,139	5,750	15
64 probe 8,192 data depth (Data width – 94)	5,872	4,221	5,870	22
64 probe 8,192 data depth (Variable data width)	7,605	4,811	9,101	201

Port Descriptions

ILA Ports and Parameters

Table 2-3 and Table 2-4 provide the details about the ILA ports and parameters.

Table 2-3: **ILA Ports**

Port Name	Direction	Description
clk	In	Design clock that clocks all trigger and storage logic.
probe<n>[<*m*> – 1:0]	In	Probe port input. The probe port number <n> is in the range from 0 to 1,023. The probe port width (denoted by <m>) is in the range of 1 to 4,096. ***Note:*** You must declare this port as a vector. For a 1-bit port, use probe<n>[0:0].
trig_out	Out	The trig_out can be generated either from trigger condition or from an external trig_in port. There is a run time control from the Logic Analyzer to switch between trigger condition and trig_in to drive trig_out. See Figure 1-1.
trig_in	In	Input trigger port used in process based system such as Zynq-7000 AP SoC for Embedded Cross Trigger. Can be connected to another ILA to create cascading Trigger.
trig_out_ack	In	An acknowledgment to trig_out.
trig_in_ack	Out	An acknowledgment to trig_in.

Table 2-4: **ILA Parameters**[(1)]

Parameter Name	Allowable Values	Default Value	Description
Component_Name	String with A–Z, 0–9, and _ (underscore)	ila_0	Name of instantiated component.
C_NUM_OF_PROBES	1–1,024	1	Number of ILA probe ports.
C_DATA_DEPTH	1,024, 2,048, 4,096, 8,192, 16,384, 32,768, 65,536, 131,072	1,024	Probe storage buffer depth. This number represents the maximum number of samples that can be stored at run time for each probe input.
C_PROBE<n>_WIDTH	1–4,096	1	Width of probe port <n>. Where <n> is the probe port having a value from 0 to 1,023.
C_TRIGOUT_EN	True/False	FALSE	Enables the trig out functionality. Ports trig_out and trig_out_ack are used.
C_TRIGIN_EN	True/False	FALSE	Enables the trig in functionality. Ports trig_in and trig_in_ack are used
C_INPUT_PIPE_STAGES	0–6	0	Add extra flops to the probe ports. One parameter applies to all of the probe ports.
C_EN_STRG_QUAL	0, 1	0	Enable the Capture (Storage) Qualifier. By enabling this you can specify the capture condition in Vivado Logic Analyzer thus capture the probes selectively. ***Note:*** This takes one extra compare values (match) unit. This means if advance trigger (C_ADV_TRIGGER) option is enabled, the maximum number of match units per probes reduces to three from four.
C_ADV_TRIGGER	True/False	FALSE	Enables the advance trigger option. This enables trigger state machine and you can write your own trigger sequence in Vivado Logic Analyzer.
ALL_PROBE_SAME_MU	True/False	TRUE	This forces the same compare value units (match units) to all of the probes.
C_PROBE<n>_MU_CNT	1–4	1	Number of Compare Value (Match) units per probe. This is valid only if ALL_PROBE_SAME_MU is FALSE.

Notes:

1. The maximum number of compare value (match) units are limited to 1,024. For the basic trigger (C_ADV_TRIGGER = FALSE), each probe has one compare value unit (as in the earlier version). But for the advance trigger option (C_ADV_TRIGGER = FALSE), this means the individual probes can still have possible selection of number of compare values units from one to four. But all of the compare value units cannot exceed more than 1,024. This also means if you need four compare units per probe then you are allowed to use only 256 probes.

Designing with the Core

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

Clocking

The `clk` input port is the clock used by the ILA core to register the probe values. For best results, it should be the same clock signal that is synchronous to the design logic that is attached to the probe ports of the ILA core.

Resets

ILA can only be reset using the Vivado® logic analyzer.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the Vivado IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4]

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado Design Suite.

If you are customizing and generating the core in the IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

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 right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4].

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

To access the core name, perform the following:

1. Open a project by selecting **File** then **Open Project** or create a new project by selecting **File** then **New Project** in Vivado.
2. Open the IP catalog and navigate to any of the taxonomies.
3. Double-click **ILA** to bring up the core name Vivado IDE.

General Options Panel

Figure 4-1 shows the General Options tab in the Native setting that allows you to specify the options.

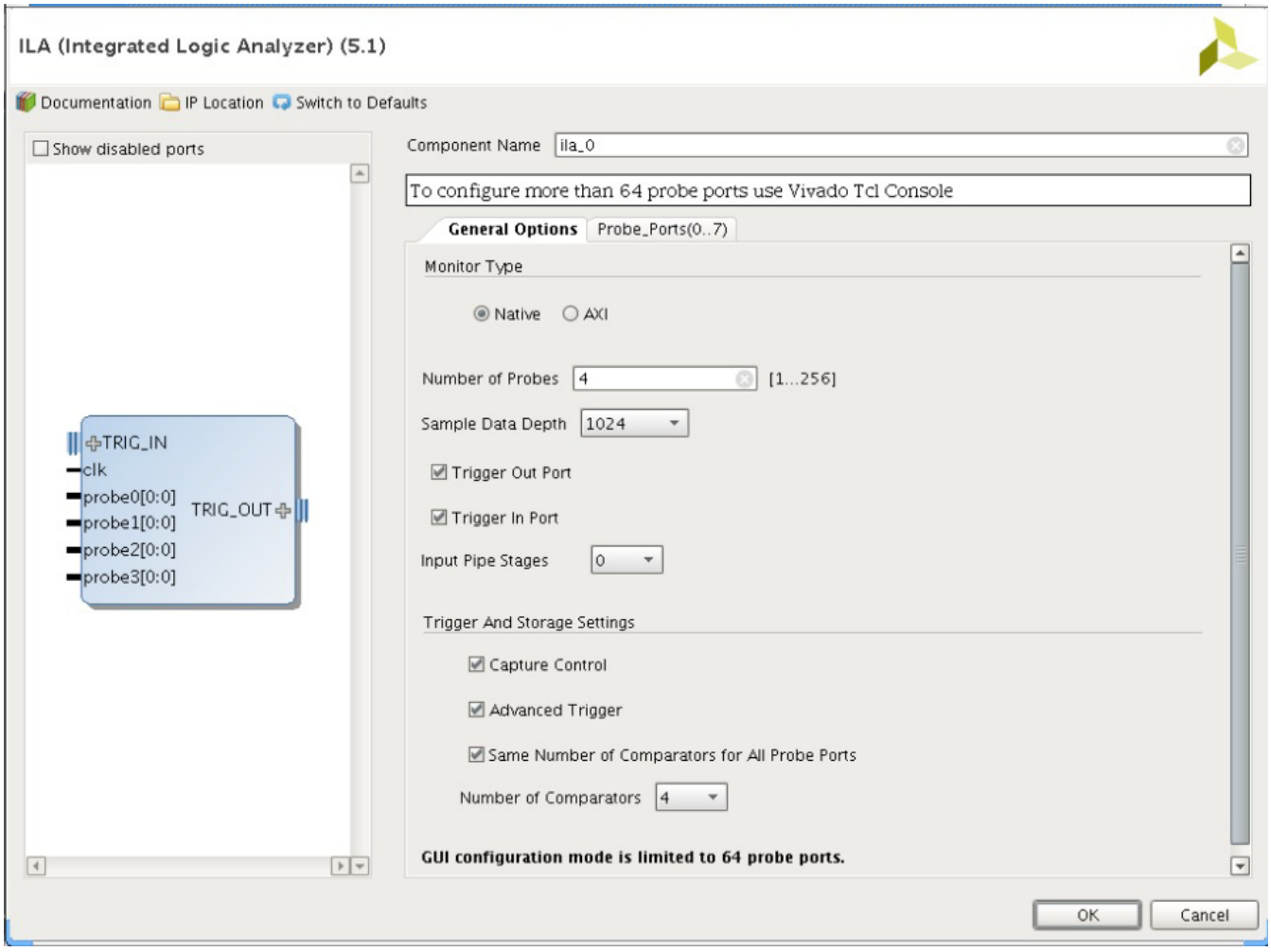

Figure 4-1: **General Options Panel – Native Monitor Type**

Figure 4-2 shows the General Options tab in the AXI setting that allows you to specify the options.

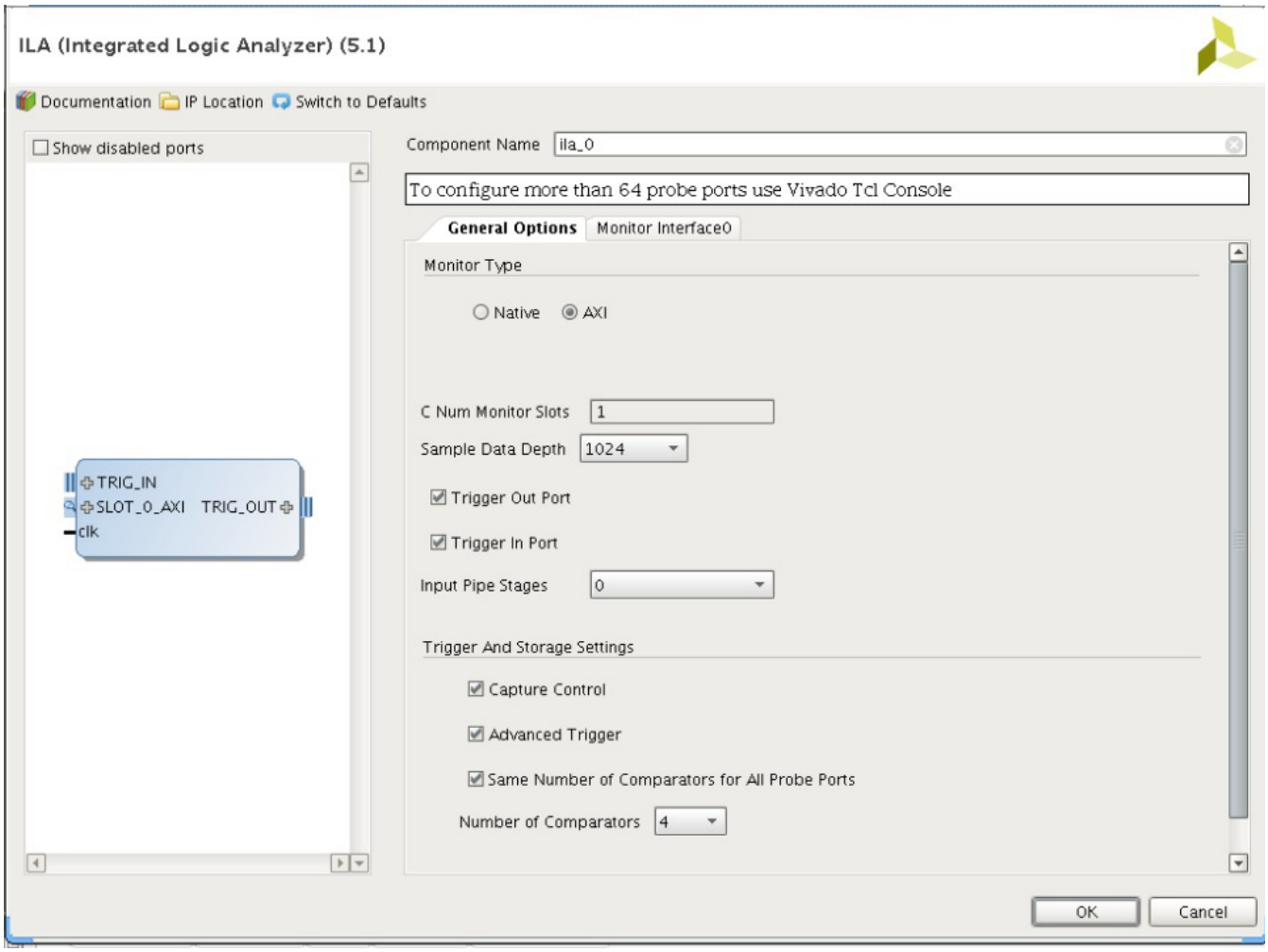

Figure 4-2: **General Options Panel – AXI Monitor Type**

- **Component Name** – Use this text field to provide a unique module name for the ILA core.
- **Monitor Type** – This option specifies which type of interface ILA should be debugging. Currently the values for this parameter are "native" and "AXI."
- **C_NUM_MONITOR_SLOTS** (Only available in AXI type) – This option allows you to select the number of AXI interface slots that needs to be connected to the ILA. For 2014.1 to currently, only one slot is supported for the ILA.
- **Number of Probes** – Use this text field to select the number of probe ports on the ILA core. The valid range used in the Vivado IDE is 1 to 64. If you need more than 64 probe ports, you need to use the Tcl command flow to generate the ILA core.

- **Sample Data Depth** – Select the suitable sample depth from the drop-down menu.
  - **Trigger Out Port** – Check to enable the optional trigger out port.
  - **Trigger In Port** – Check to enable the optional trigger in port.
- **Input Pipe Stages** – Select the number of registers you want to add for the probe. This parameter applies to all of the probes.
- **Storage Qualification** – Check to enable the qualifier for the trace capture.
- **Advanced Trigger** – Check to enable the state machine-based trigger sequencing.
- **Same No. of Comparators for all Probes** – Check to enable the same number of comparators for all the enabled probes.
- **No. of Comparators** – Select to enable the number of comparators that applies to all enabled probes.

Probe Port Panels

Figure 4-3 shows the Probe Ports tab that allows you to specify the settings.

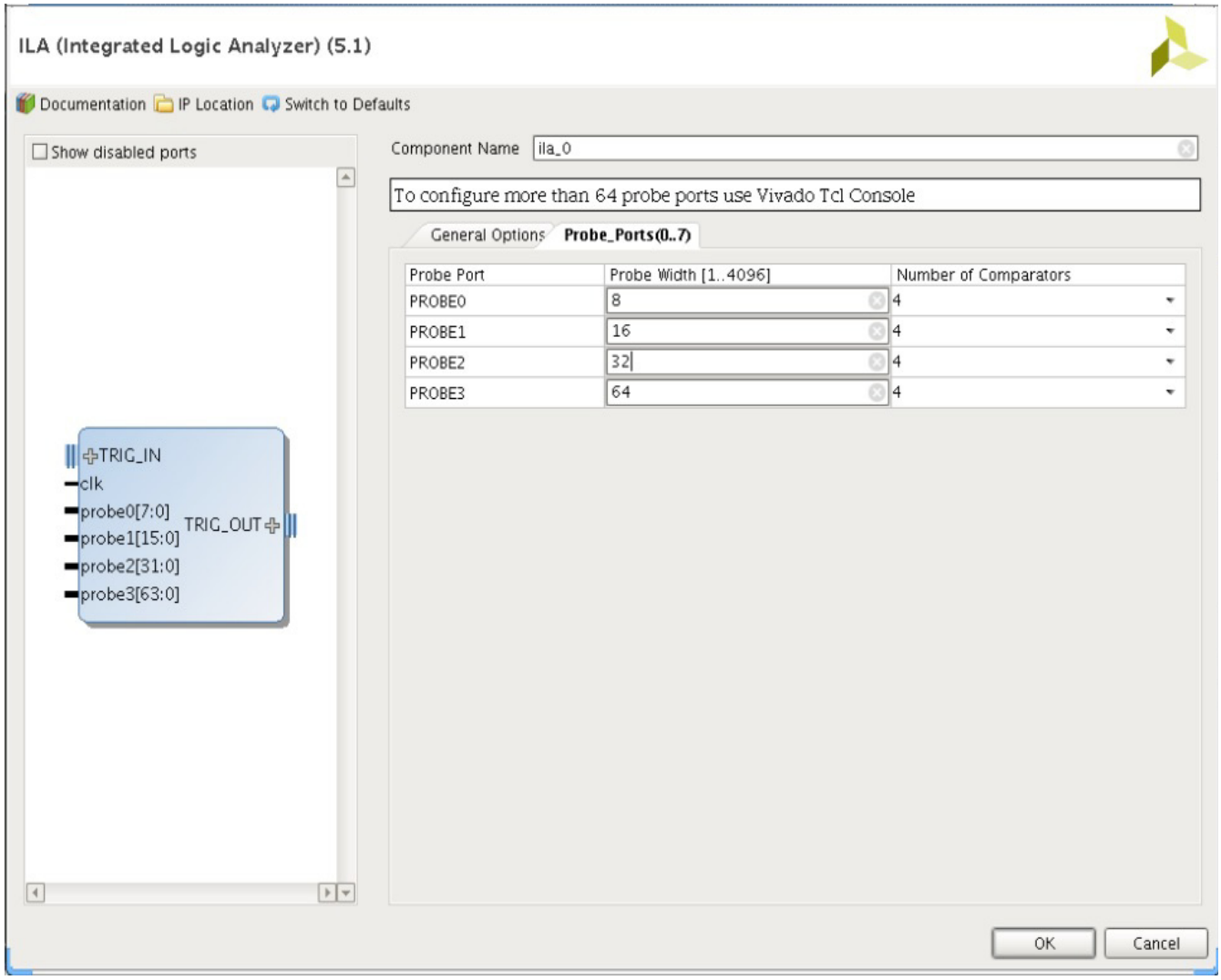

Figure 4-3: **Probe Ports Panel**

- **Probe Port Panels** – Width of each Probe Port can be configured in Probe Port Panels. Each Probe Port Panel has up to seven ports.

  Also, number of comparator per probe can be configured on this panel. This option appears only when **Advanced Trigger** option is selected and the **Same No. of Comparator for all Probes** is disabled on the first page of Vivado IDE.

Monitor Interface0 Panel

Figure 4-4 shows the Monitor Interface0 tab that allows you to specify the settings.

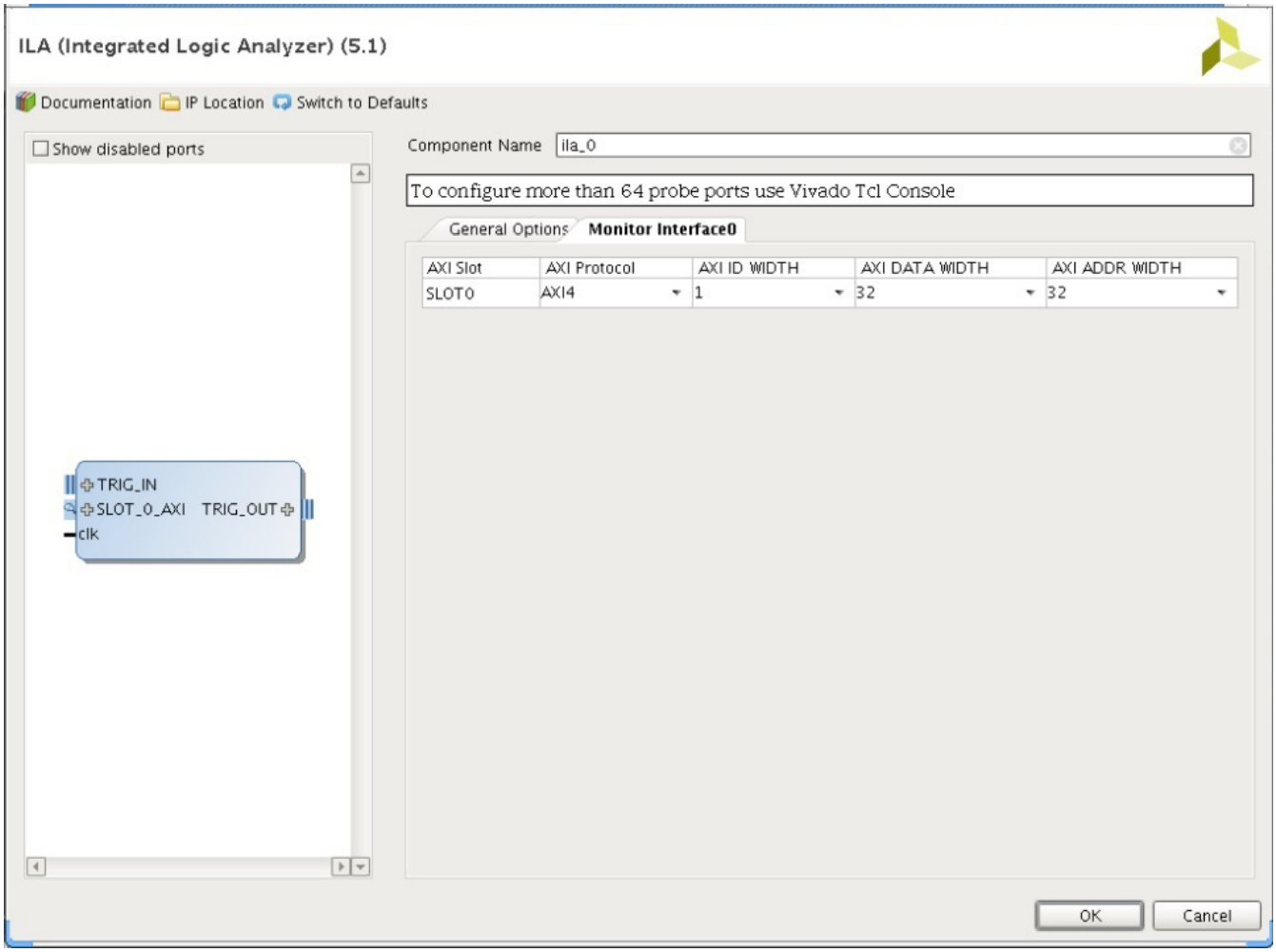

Figure 4-4: **Probe Ports Panel**

- **C_ENABLE_AXI_MON** – Enables AXI monitor in the ILA IP core. This option allows you to debug designs with AXI interface at interface level.
- **C_SLOT_0_AXI_ARUSER_WIDTH** – Default value is always 1 (Not shown in the AXI Read Address Channel User Width of the Vivado IDE).
- **C_SLOT_0_AXI_RUSER_WIDTH** – Default value is always 1 (Not shown in the AXI Read Channel User Width of the Vivado IDE).
- **C_SLOT_0_AXI_AWUSER_WIDTH** – Default value is always 1 (Not shown in the AXI Write Address Channel User Width of the Vivado IDE).
- **C_SLOT_0_AXI_WUSER_WIDTH** – Default value is always 1 (Not shown in the AXI Write Channel User Width of the Vivado IDE).

- **C_SLOT_0_AXI_BUSER_WIDTH** – Default value is always 1 (Not shown in the AXI Write Response Channel User Width of the Vivado IDE).
- **C_SLOT_0_AXI_ID_WIDTH** – AXI ID Width. Valid range is from 1 to 32 (Shown in the Vivado IDE).
- **C_SLOT_0_AXI_DATA_WIDTH** – AXI data width. Valid values are 32, 64, 128, 256, 512, and 1,024 (Shown in the Vivado IDE).
- **C_SLOT_0_AXI_ADDR_WIDTH** – AXI address width. Valid range is from 1 to 32 (Shown in the Vivado IDE).
- **C_SLOT_0_AXI_PROTOCOL** – AXI interface protocol. AXI protocols supported are AXI3, AXI4, AXI4-Lite, and AXI-Stream (AXIS).
- **C_SLOT_0_AXIS_TDATA_WIDTH** – AXIS data width. Valid values are 8, 16, 32, 64, 128, 256, 512, and 1,024.
- **C_SLOT_0_AXIS_TID_WIDTH** – AXIS ID width. Valid range is from 1 to 32.
- **C_SLOT_0_AXIS_TUSER_WIDTH** – AXIS user width. Valid range is from 0 to 1,024.
- **C_SLOT_0_AXIS_TDEST_WIDTH** – AXIS destination width. Valid range is from 1 to 32.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

The ILA core includes an XDC file that contains appropriate false path constraints to prevent the over-constraining of clock domain crossing synchronization paths. It is also expected that the clock signal connected to the `clk` input port of the ILA core is properly constrained in your design.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

This core does not support simulation.

Synthesis and Implementation

This section contains information about synthesis and implementation in the Vivado Design Suite. For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

IMPORTANT: *Synthesis with Synopsys Synplify is not supported for the core.*

Example Design

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

Directory and File Contents

- **<component name>_example/<component name>_example.srcs/**
  Top-level project directory; name is user-defined
  - constrs_1/imports/<component name>/
    - example_<component name>.xdc
  - sources_1/imports/<component name>/
    - example_<component name>.v
  - sources_1/ip/<component name>

Note: Only Verilog is supported.

<component name>_example/<component name>_example.srcs/

This directory contains the source files needed to synthesize the ILA core whose name is `<component name>`.

Table 5-1: **ILA Example Design Source Files**

Name	Description
constrs_1/imports/<component name>/	
example_<component name>.xdc	Constraints file for the example design
sources_1/imports/<component name>/	
example_<component name>.v	Verilog (.v) source file for the example design

Implementation

To implement the example design, select **Run Implementation** in the **Vivado Project Manager** window. For further details on setting up the implementation, see the *Vivado Design Suite User Guide: Implementation* (UG904) [Ref 6].

Test Bench

There is no test bench for this IP core release.

Appendix A

Verification, Compliance, and Interoperability

Xilinx® has verified the ILA v5.1 core in a proprietary test environment, using an internally developed bus functional model.

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 on migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 7].

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.

Port Changes

- The maximum number of Probe ports is 1,024 and the maximum width of each Probe port can be up to 4,096. However, the total number of bits (sum of all probe ports) cannot exceed 65,536 bits.
- The port names are changed from uppercase to lowercase and you have to take care of updating this in your design.
- There are other new parameters added to support the new features. For example,
  - Storage Qualifier
  - Advance Trigger
- The new features are default disable to have backward compatibility.
- To upgrade the IP to the latest version use the upgrade IP service in Vivado.

Functionality Changes

The ILA v5.1 core no longer has separate Data and Trigger ports. The Probe port is used for both Data and Trigger.

IMPORTANT: *The ILA v5.1 core is not compatible with the legacy ChipScope™ Pro Analyzer tool. The ILA v5.1 core requires the Vivado logic analyzer feature for run time interaction.*

Debugging

This appendix includes details about resources available on the Xilinx® Support website and debugging tools.

TIP: *If the IP generation halts with an error, there might be a license issue. See License Checkers in Chapter 1 for more details.*

Finding Help on Xilinx.com

To help in the design and debug process when using the ILA, 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 ILA. 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 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 ILA

AR: 54606

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.

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. Log in to the WebCase tool to see your specific support options.

Debug Tools

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

Vivado Lab Edition

Vivado® Lab Edition inserts logic analyzer and virtual I/O cores directly into your design. Vivado Lab Edition allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

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

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

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 1].

For more information on the ability to interact with the ILA core using Tcl Console commands, see Chapter 5 in the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 1].

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. Vivado Lab Edition is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using Vivado Lab Edition for debugging the specific problems.

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. If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `locked` port.

Appendix D

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado® Design Suite User Guide: Programming and Debugging* (UG908)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994)
4. *Vivado Design Suite User Guide: Getting Started* (UG910)
5. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
6. *Vivado Design Suite User Guide: Implementation* (UG904)
7. *ISE® to Vivado Design Suite Migration Guide* (UG911)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/01/2015	5.1	• Updated to latest GUIs in Figs. 4-1 to 4-4.
10/01/2014	5.0	• Updated Figs. 4-1 to 4-4. • Updated C_SLOT_0_AXI_PROTOCOL description. • Added C_SLOT_0_AXIS_TDATA_WIDTH to C_SLOT_0_AXIS_TDEST_WIDTH.
04/02/2014	4.0	• Added AXI interface on ILA IP core to debug AXI to Features in IP Facts. • Updated Fig. 1-1 in Overview chapter. • Added UltraScale Device Utilization table in Resource Utilization section. • Updated trig_out description in Table 2-2: ILA Ports. • Added Design Flow Steps chapter. • Updated descriptions in Vivado Lab Tools section in Debug Appendix.
12/18/2013	3.0	Added UltraScale support.
10/02/2013	3.0	Revision number advanced to 3.0 to align with core version number 3.0. • Updated Fig. 1-1. • Updated Table 2-1: Device Utilization – Kintex-7 FPGAs • Updated Table 2-3: ILA Parameters. • Updated Figs. 4-1 to 4-3 in Customizing and Generating the Core. • Added descriptions in General Options Panel. and Probe Port Panels. • Updated Migrating Appendix.
06/19/2013	2.1	Initial Xilinx release of Product Guide, derived from DS875.

Please Read: Important Legal Notices