LogiCORE IP Integrated Logic Analyzer (ILA) v2.1

Product Guide for Vivado Design Suite

PG172 June 19, 2013

XILINX®

XILINX

Table of Contents

Appendix E: Debugging

Appendix F: Additional Resources

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 trigger width, data width, and data depth
- Multiple probe ports, which can be combined into a single trigger condition

For more information about the ILA core, see the *Vivado Design Suite User Guide: Programming and Debugging* [Ref 6].

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

Notes:

1. For a complete list of supported devices, see Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

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.

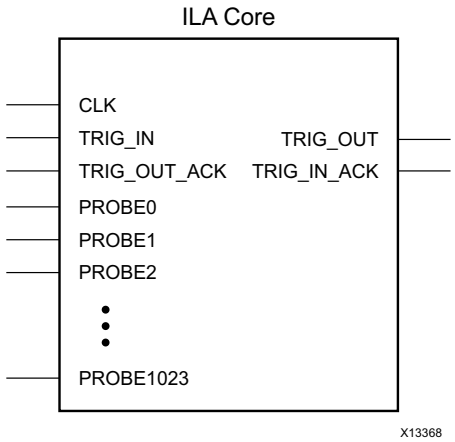

Figure 1-1: **ILA Core Symbol**

Note: The numerical range from Probe 3 to Probe 1023 is indicated by ellipses (...) in Figure 1-1.

After the design is loaded into the FPGA device on the board, 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 runtime 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.

Product Specification

Performance

The ILA core can be configured to Select 1024 probes each of width ranging from 1 to 4096. 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 Kintex®-7 devices. FPGAs (Table 2-1). These values were generated using Vivado® IP Catalog. They are derived from post-synthesis reports, and might change during implementation.

Kintex-7 FPGAs

Table 2-1 provides approximate resource counts for the various core options on Kintex-7 FPGAs.

Table 2-1: **Device Utilization – Kintex-7 FPGAs**

Configuration	Slices	LUTs	FFs	BRAMs
64 probe 8192 data depth (Same data width - 1)	2965	465	3994	15
64 probe 8192 data depth (Data width - 94)	5117	3708	15174	512
64 probe 8192 data depth (Variable data width)	8759	9233	33767	1338

Port Descriptions

ILA Ports and Parameters

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

Table 2-2: **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> will be in the range from 0 to 1023. The probe port width (denoted by <m>) is in the range of 1 to 4096. ***Note:*** You must declare this port as a vector. For a one-bit port, use PROBE<n>[0:0].
TRIG_OUT	OUT	Goes HIGH when the trigger condition occurs. Mainly used in process based system such as Zynq-7000 for Embedded Cross Trigger. Can be connected to another ILA and create cascading of Trigger. Refer to Figure 1-1.
TRIG_IN	IN	Input trigger port used in process based system such as Zynq-7000 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.

ILA Parameters

Table 2-3: **ILA Parameters**

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	1–1024	1	Number of ILA probe ports
C_DATA_DEPTH	1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072	1024	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	1–4096	1	Width of probe port <n>. Where <n> is the probe port having a value from 0 to 1023.

Table 2-3: **ILA Parameters *(Cont'd)***

Parameter Name	Allowable Values	Default Value	Description
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

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.

Customizing and Generating the Core

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

Vivado Integrated Design Environment

The ILA core can be found in `/Debug & Verification/Debug/` in the Vivado IP Catalog (Figure 4-1).

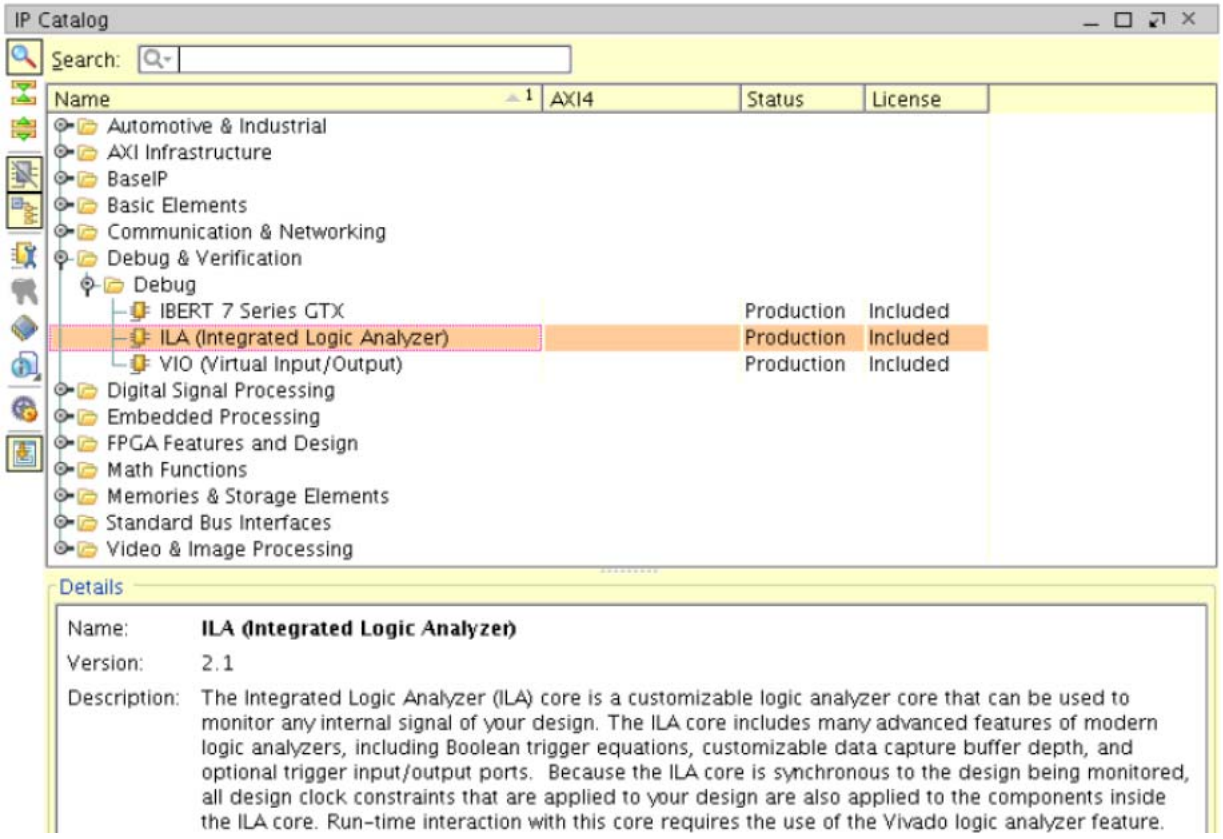

Figure 4-1: **ILA Core in Vivado IP Catalog**

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 on **ILA** to bring up the core name GUI.

General Options Panel

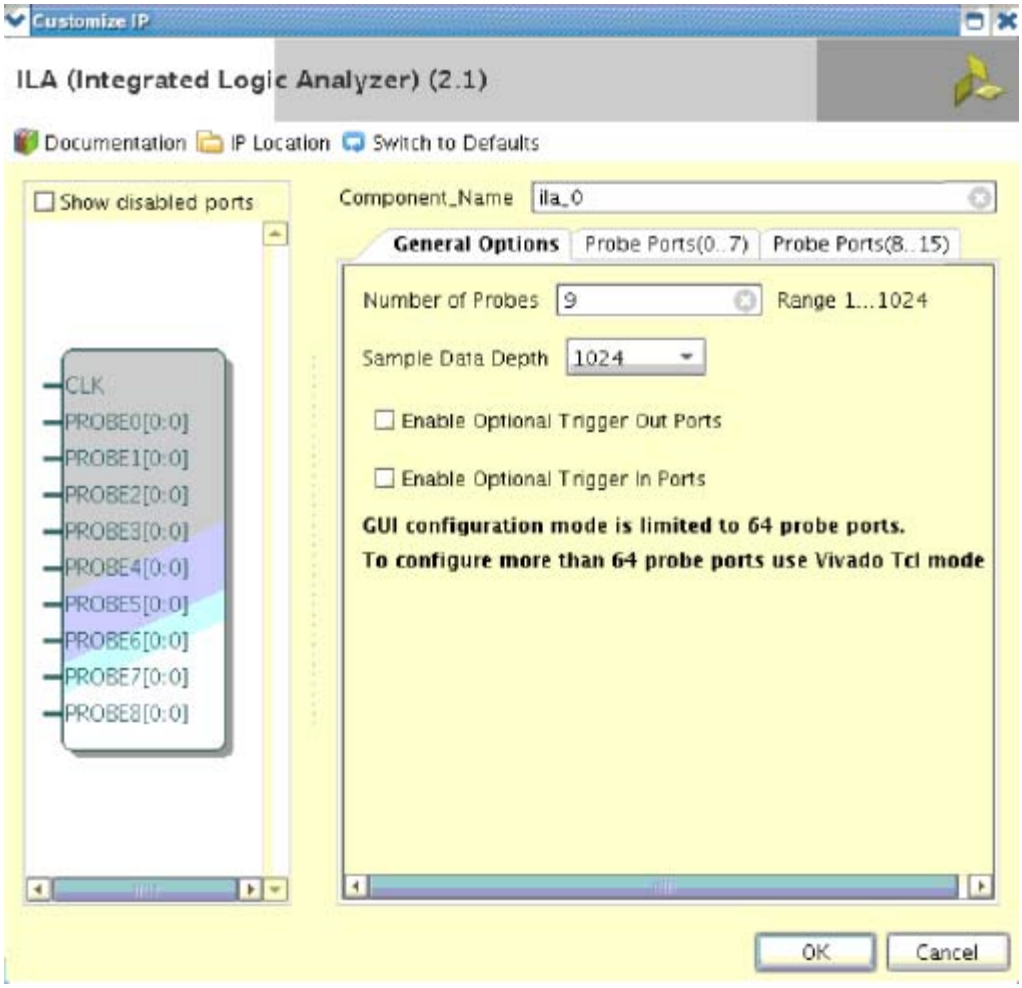

Figure 4-2: **General Options Panel**

- **Component Name** – Use this text field to provide a unique module name for the ILA core.
- **Number of Probes** – Use this text field to select the number of probe ports on the ILA core. The valid range used in the GUI is 0 to 64. If you need more than 64 probe ports, you need to use the Tcl command flow to generate the ILA core.

  Note: At least one input or output probe port needs to be specified.

- **Sample Data Depth** - Select the suitable sample depth from the drop-down menu.
- **Enable Trigger Out Port** - Check the radio button to enable the optional trigger out port.
- **Enable Trigger In Port** - Check the radio button to enable the optional trigger in port.

Probe Port Panels

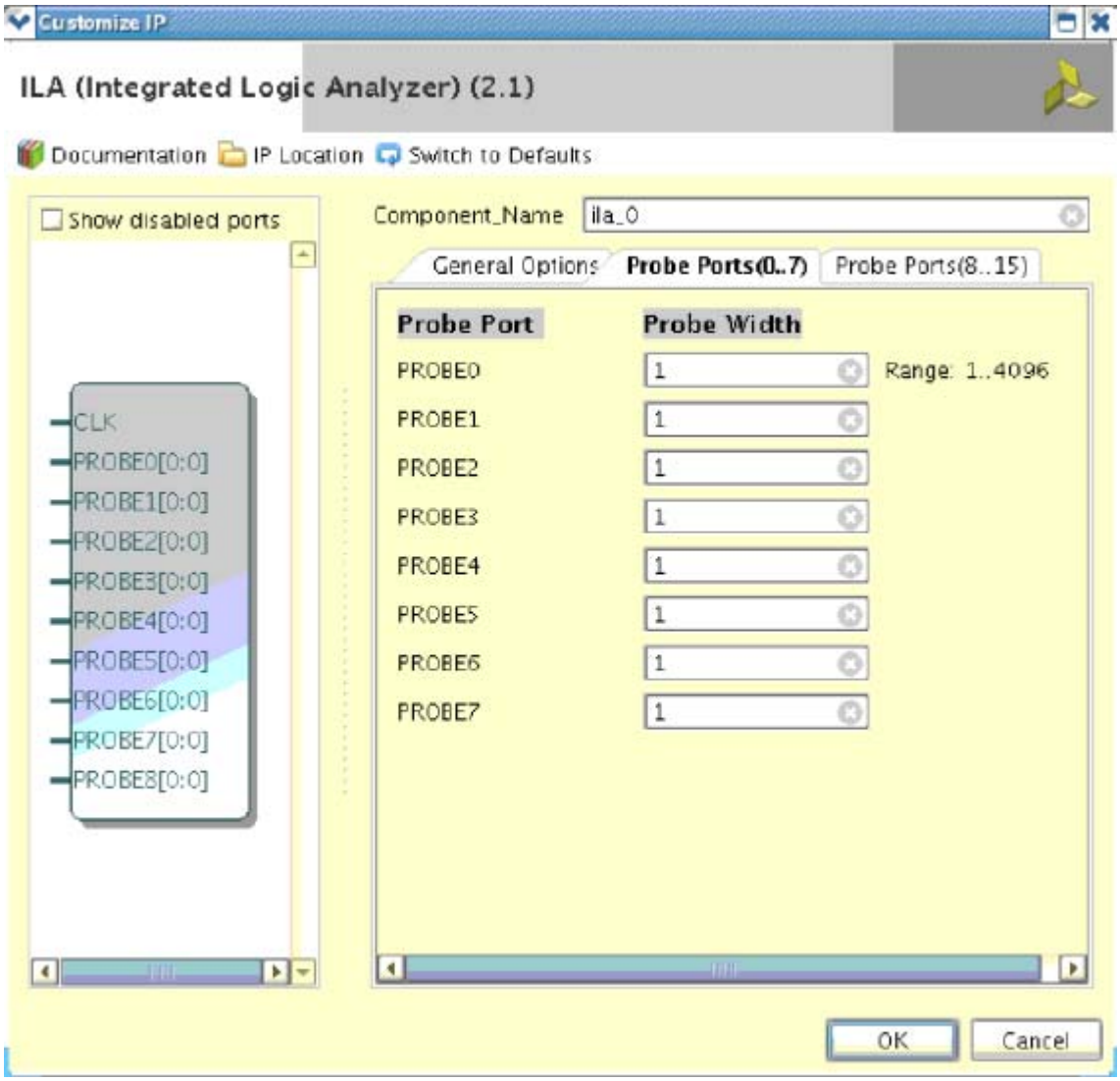

Figure 4-3: **Probe Ports Panel**

- **Probe Port Panels** - Width of each Probe Port can be configured in Probe Port Panels. Each Probe Port Panel has up to 7 ports.

Output Generation

This section describes the files and directory structure generated by the IP. For the purposes of this document, assume the name of the project is the default "project_1."

- **project_1/project_1.srcs**
  Top-level project directory; name is user-defined
  - sources_1/ip/<component name>
    - constraints
      - ila.xdc
    - labtools_general_components_lib_v2_0/hdl/verilog
    - labtools_xsdb_slave_lib_v2_1/hdl/verilog
    - synth
      - <component name>.v/.vhd
    - ila_v2_a/hdl/verilog
      - ila_v2_1_ila
      - ila_v2_1_ila_cap_addrgen
      - ila_v2_1_ila_cap_ctrl_legacy
      - ila_v2_1_ila_cap_sample_counter
      - ila_v2_1_ila_cap_window_counter
      - ila_v2_1_ila_core
      - ila_v2_1_ila_lib_function
      - ila_v2_1_ila_register
      - ila_v2_1_ila_reset_ctrl
      - ila_v2_1_ila_trace_memory
      - ila_v2_1_ila_trig_match
      - ila_v2_1_ila_trigger
      - ila_v2_1_ila_ver_inc
    - <component name>.veo/.vho
    - <component name>.xci
    - <component name>.xml

project_1/project_1.srcs/sources_1/ip

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

Table 4-1: **ILA Source Files**

Name	Description
constraints/ila.xdc	Constraints file for the ILA core.
labtools_general_components_lib_v2_0/*	Common components used by all lab tools cores.
labtools_xsdb_slave_lib_v2_1/*	Common interface library used by all lab tools to connect to the dbg_hub core.
synth/<component name>.v/.vhd	Verilog (.v) or VHDL (.vhd) file used by synthesis.
ila_v2_1/hdl/verilog/*.v	Verilog source files used to describe the ILA v2.1 core.
<component name>.veo/.vho	Verilog (.veo) or VHDL (.vho) file used to describe the instantiation template for the ILA core.
<component name>.xci	Core description file for the ILA core.
<component name>.xml	Component XML file for the ILA core.

Appendix A

Verification, Compliance, and Interoperability

Xilinx has verified the ILA v2.1 core in a proprietary test environment, using an internally developed bus functional model.

Constraining the Core

This chapter contains information about constraining the core in the Vivado® Design Suite environment.

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 constraints.

Detailed Example Design

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

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>
    - See Output Generation, page 12 section for details on the files in this directory.

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 C-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 4].

Migrating

For information on migrating to the Vivado® Design Suite, see *Vivado Design Suite Migration Methodology Guide* [Ref 1].

Port Changes

There are no Trigger and Data ports. The Probe port can be used as both Trigger and Data.

The maximum number of Probe ports is 1024 and the maximum width of each Probe port can be up to 4096. However, the total number of bits (sum of all probe ports) can not exceed 65536 bits.

Functionality Changes

The ILA v2.1 core no longer has separate Data and Trigger ports. The Probe port is used for both Data and Trigger.

IMPORTANT: *The ILA v2.1 core is not compatible with the legacy ChipScope™ Pro Analyzer tool. The ILA v2.1 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. In addition, this appendix provides a step-by-step debugging process to guide you through debugging the ILA core.

The following topics are included in this appendix:

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 opening a Technical Support WebCase.

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.

Known Issues

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 are listed below, and can also 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 Support Quick Links.

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.

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.

Example Design

The ILA is delivered with an example design that can be synthesized, complete with functional test benches. Information about the example design can be found in *Chapter 6, Detailed Example Design*.

Vivado Lab Tools

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

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

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

License Checkers

If the IP requires a license key, the key must be verified. The Vivado design tools have several license check points 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 Implementation
- Vivado Logic Analyzer

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

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. The Vivado logic analyzer tool is a valuable resource to use in hardware debug.

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.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `LOCKED` port.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado Design Suite User Guide - Logic Simulation* (UG900)
2. *Vivado Design Suite User Guide - Implementation* (UG904)
3. Vivado design tools user documentation
4. *Vivado Design Suite User Guide: Designing with IP* (UG896)
5. *Vivado Design Suite Migration Methodology Guide* (UG911)
6. Xilinx Vivado Design Suite User Guide: Programming and Debugging (UG908).

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/19/2013	2.1	Initial Xilinx release of Product Guide, derived from DS875.

Notice of Disclaimer