System Integrated Logic Analyzer v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG261 June 7, 2017

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

The customizable System Integrated Logic Analyzer (System ILA) IP core is a logic analyzer which can be used to monitor the internal signals and interfaces of a design. The System ILA core includes many advanced features of modern logic analyzers, including boolean trigger equations and edge transition triggers. The core also offers interface debug and monitoring capability along with AXI4-MM and AXI4-Stream protocol checking[1]. Because the System 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 System ILA core.

The IP is functionally equivalent to an ILA but offers additional benefits in debugging interfaces in both IPI and the Hardware Manager. It is recommended to use the System ILA IP in IPI for debugging interfaces and nets. For more information on how to view AXI interface transactions and channel events in the Vivado Hardware Manager waveform viewer, see [Ref 6].

[1] AXI4/AXI4-Stream protocol violations can be displayed in waveform view by enabling protocol checker. The `pc_asserted` signal assertion indicates protocol violation while the `pc_status` signal indicates the nature of protocol violation. For more details on the description of `pc_status` signal, see [Ref 9] and [Ref 10].

Features

- User-selectable number of probe ports and probe width.
- Multiple probe ports, which can be combined into a single trigger condition.
- Debugging of any debuggable interface including AXI4-MM and Stream in a system created in IP Integrator.
- User-selectable AXI4-MM channel debug and AXI Data/Address width selection.
- Data and Trigger probe and interface type selection.
- BRAM estimation.
- AXI4-MM and AXI4-Stream Protocol Checking.
- IPI Support.

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale+™, UltraScale™, Zynq®-7000 All Programmable SoC, 7 Series
Supported User Interfaces	IEEE Standard 1149.1 - JTAG
Resources	
Provided with Core	
Design Files	N/A
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	Not Provided
Supported S/W Driver[2]	N/A
Tested Design Flows[3]	
Design Entry	Vivado® Design Suite
Simulation	Not Provided
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. Standalone driver details can be found in the software development kit (SDK) directory (<install_directory>/SDK/<release>/data/embeddedsw/doc/xilinx_drivers.htm). Linux OS and driver support information is available from the Xilinx Wiki page.
3. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

Feature Summary

Signals and interfaces in the FPGA design are connected to the System ILA probe and slot inputs (see Figure 1-1). These signals and interfaces, attached to the probe and slot inputs respectively, are sampled at design speeds and stored using on-chip block RAM (BRAM).

The core parameters specify:

- The number of probes and interface slots.
- Interface types, trace sample depth.
- Data and trigger property of probes and interfaces.
- Number of comparators and the width for each probe and individual ports within interfaces.

Communication with the System ILA core is conducted using an auto-instantiated debug core hub which connects to the JTAG interface of the FPGA.

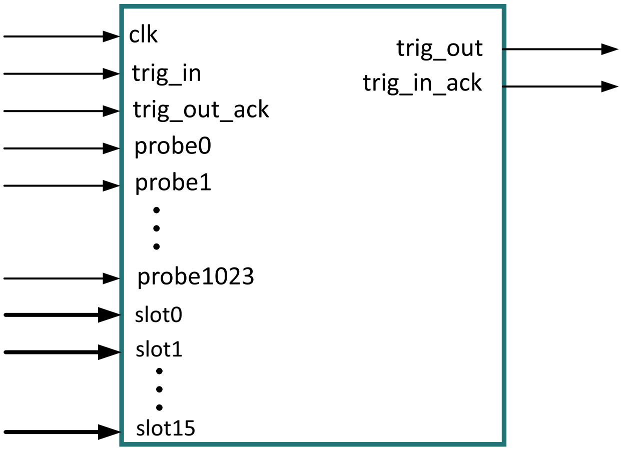


Figure 1-1: **System ILA Symbol**

After the design is loaded into the FPGA, you can use the Vivado® logic analyzer software to set up a trigger event for the System 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 System ILA core.

As System ILA is capable of monitoring interface level signals, it can convey transaction level information such as outstanding transaction for AXI4 interfaces.

System ILA Probe Trigger Comparator

Each probe input is connected to trigger comparators which can perform various operations. After the design has been programmed into the device, in the Vivado Hardware Manager 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 also perform more complex comparisons, including >, <, >=, <=.

IMPORTANT: The number of comparators used by the System ILA is set during customization of the System ILA IP.

System ILA Trigger Condition

The trigger condition is the result of a Boolean *AND* or *OR* calculation of each of the System ILA probe trigger comparator results. Using the Vivado logic analyzer, you can select whether to *AND* or *OR* probe trigger comparators.

The *AND* setting causes a trigger event when all of the System ILA probe comparisons are satisfied.

The *OR* setting causes a trigger event when any of the System ILA probe comparisons are satisfied. The trigger condition is the trigger event used for the System ILA trace measurement.

Applications

The System ILA IP core is designed to be used in any application which requires verification or debugging using the Vivado logic analyzer. Figure 1-2 shows MicroBlaze™ processor writes and read from AXI BRAM controller through M_AXI_DP interface. The System ILA core is connected to the interface net between the MicroBlaze™ processor and AXI Bram controller to monitor the AXI4 transaction between MicroBlaze™ processor and AXI BRAM controller in hardware manager.

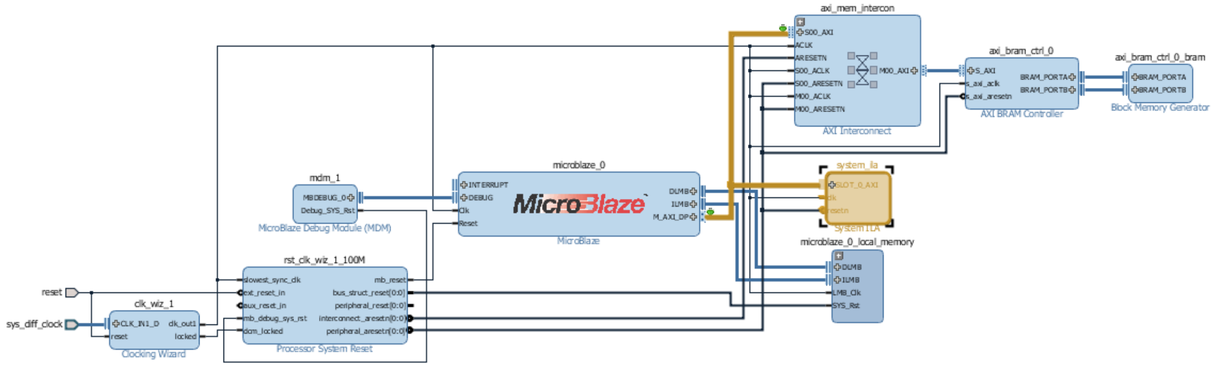


Figure 1-2: **AXI Interface Debugging use case**

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

The customizable System Integrated Logic Analyzer (System ILA) IP core is a logic analyzer which can be used to monitor the internal signals and interfaces of a design. The System ILA core includes many advanced features of modern logic analyzers, including boolean trigger equations and edge transition triggers. The core also offers interface debug and monitoring capability along with AXI4-MM and AXI4-Stream protocol checking. Because the System 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 System ILA core.

Port Descriptions

The port descriptions for the System ILA IP Core Ports and Parameters are described in the following sections.

Table 2-1: **ILA Ports**

Port Name	I/O	Description
clk	I	Design clock that clocks all trigger and storage logic.
Probe<n>[<m> - 1:0]	I	Probe port input. The probe port <**n**> number 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. You must declare this port as a vector. For a 1-bit port, use probe<n>[0:0]
Slot_<p>_<intf_name>	O	Slot interface. The type of the interface **<intf_name>** is created dynamically based on the **slot_<p>_<intf_name>** interface type parameter. The individual ports within the interfaces are available for monitoring in the hardware manager.
trig_out	O	The **trig_out** can be generated either from trigger condition or from an external **trig_in** port. There is a Vivado Hardware Manager control from the Logic Analyzer to switch between trigger condition and **trig_in** to drive **trig_out**. See Figure 1-1.
trig_in	I	Input trigger port used in process based system such as Zynq-7000 AP SoC for Embedded Cross Trigger. This port can be connected to another ILA to create a cascading Trigger.

Table 2-1: **ILA Ports**

Port Name	I/O	Description
trig_out_ack	I	An acknowledgment to **trig_out**.
trig_in_ack	O	An acknowledgment to **trig_in**.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
Component_Name	String with A-Z, 0-9, and '_'	System_ila_0	Name of instantiated component.
C_MON_TYPE	NATIVE INTERFACE MIX	INTERFACE	System ILA monitor type: **NATIVE**: Monitor Individual probes **INTERFACE**: Monitor interfaces **MIX**: Monitor Individual probes and interfaces
C_NUM_OF_PROBES	1-1,024	1	Number of System ILA Probe ports.
C_NUM_MONITOR_SLOTS	1-16	1	Number of Interface Slots.
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 for each probe input.
C_PROBE<n>_WIDTH	1-4,096	1	Width of probe port **<n>**. Where **<n>** has a value of 0 to 1,023.
C_PROBE_WIDTH_PROPAGATION	AUTO, MANUAL	AUTO	The width of individual native probes can only be set when the native probe width propagation is set to manual. Otherwise, the individual probe width parameters are not editable. The value is automatically set post propagation in IPI design.
C_INPUT_PIPE_STAGES	0-6	0	Add extra flops to the probe ports. One parameter applies to all of the probe ports.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
C_EN_STRG_QUAL	0,1	0	Enables the Capture (Storage) Qualifier. By enabling this you can specify the capture condition in Vivado Logic Analyzer thus capture the probes selectively. 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 from four to three.
C_ADV_TRIGGER	True/False	False	Enables the advance trigger option. This enables trigger state machine so you can write your own trigger sequence in Vivado Logic Analyzer.
ALL_PROBE_SAME_MU	True/False	True	Forces the same compare value units (match units) to all of the probes.
C_PROBE<n>_MU_CNT	1-16	1	Number of Compare Value (Match) units per probe. This is valid only if ALL_PROBE_SAME_MU is FALSE.
C_PROBE<n>_TYPE	Data, Trigger, Data & Trigger	Data & Trigger	Data and/or Trigger property of probe port <n>. Where **<n>** is the probe port with a value of 0 to 1,023
C_SLOT_<p>_TXN_CNTR_EN	0,1	1	Enable the AXI transaction tracking capability for slot <**p**> (when the slot is configured as an AXI interface).
C_SLOT_<p>_MAX_WR_BURSTS	2,4,8,16,32,64	2	Configure the number of outstanding *write* transactions.
C_SLOT_<p>_MAX_RD_BURSTS	2,4,8,16,32,64	2	Configure the number of outstanding *read* transactions.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
C_SLOT_<p>_INTF_TYPE	All VLNV[Ref 11] of debuggable interfaces available in Vivado.	xilinx.com:interface:aximm_rtl:1.0	Interface type is the VLNV of interface connected to **slot_<p>_<intf_name>**
C_SLOT_<p>_APC_EN	0,1	0	Enable AXI protocol checker IP for slot **<p>** (when the slot type is configured as an AXI interface).
C_SLOT_<p>_TYPE	Data, Trigger, Data & Trigger	Data & Trigger	To choose a selected **slot_<p>** for specifying trigger conditions, data storage, or both.
C_SLOT_<p>_AXI_AW_SEL_DATA	0,1	0	Configure Write Address channel signals as Data when **slot_<p>** is configured as AXI interface and **slot_<p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_W_SEL_DATA	0,1	0	Configure Write Data channel signals as Data when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_B_SEL_DATA	0,1	0	Configure Write Response channel signals as Data when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_AR_SEL_DATA	0,1	0	Configure Read Address channel signals as Data when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_R_SEL_DATA	0,1	0	Configure Read Data channel signals as Data when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_AW_SEL_TRIG	0,1	0	Configure Write Address channel signals as Trigger when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
C_SLOT_<p>_AXI_W_SEL_ TRIG	0,1	0	Configure Write Data channel signals as Trigger when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_B_SEL_TRIG	0,1	0	Configure Write Response channel signals as Trigger when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_AR_SEL_TRIG	0,1	0	Configure Read Address channel signals as Trigger when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_R_SEL_TRIG	0,1	0	Configure Read Data channel signals as Trigger when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_DATA_SEL	0,1	0	Configure AXI4-Stream signals as Data when **slot_<p>** is configured as AXI-Stream interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_TRIG_SEL	0,1	0	Configure AXI4-Stream signals as Trigger when **slot_<p>** is configured as AXI-Stream interface where **slot <p>** has a value of 0 to 15.
C_SLOT_<p>_AXI_PROTOCOL	AXI3, AXI4LITE, AXI4	AXI4	AXI4 protocol when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_DATA_WIDTH	32, 64, 128, 256, 512, 1024	32	AXI-MM data width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
C_SLOT_<p>_AXI_ADDR_WIDTH	1-32	32	AXI-MM address width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_ID_WIDTH	0-32	0	AXI-MM ID width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_AWUSER_WIDTH	0-1,024	0	AXI-MM AWUSER width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_WUSER_WIDTH	0-1,024	0	AXI-MM WUSER width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_BUSER_WIDTH	0-1,024	0	AXI-MM BUSER width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXI_ARUSER_WIDTH	0-1,024	0	AXI-MM ARUSER width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.

Table 2-2: **ILA Parameters**

Parameter Name	Allowable Values	Default	Description
C_SLOT_<p>_AXI_RUSER_WIDTH	0-1,024	0	AXI-MM RUSER width when **slot_<p>** is configured as AXI interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXIS_TDATA_WIDTH	8, 16, 24, 32, 64, 128, 256, 512, 1024	32	AXI-Stream tdata width when **slot_<p>** is configured as AXIS interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXIS_TID_WIDTH	0-32	0	AXI-Stream tid width when **slot_<p>** is configured as AXIS interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXIS_TUSER_WIDTH	0-32	0	AXI-Stream TUSER width when **slot_<p>** is configured as AXIS interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
C_SLOT_<p>_AXIS_TDEST_WIDTH	0-32	0	AXI-Stream TDEST width when **slot_<p>** is configured as AXIS interface where **slot <p>** has a value of 0 to 15. The value is automatically set post propagation in IPI design.
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.

Chapter 3

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 System 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 System ILA core.

Resets

System 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 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 1]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 3]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [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.

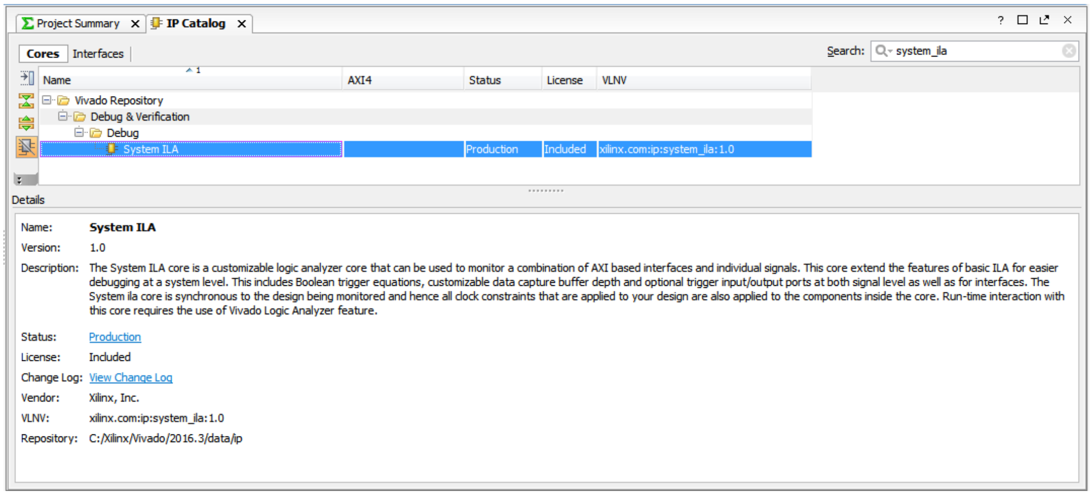
If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 1] for detailed information. IP integrator may auto-compute certain configuration values when validating or generating the design. To check whether the values do 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 **Add IP** option from the Vivado IPI right click menu.
2. Add the **System ILA** IP from Vivado IPI catalog. (See Figure 4-1.)
3. Double-click the selected IP or select the **Customize Block** 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 3].

Note: Figures in this chapter are illustrations of the Vivado Integrated Design Environment (IDE). The layout depicted here might vary from the current version.



Name	AXI4	Status	License	VLNV
Vivado Repository				
Debug & Verification				
Debug				
System ILA		Production	Included	xilinx.com:ip:system_ila:1.0

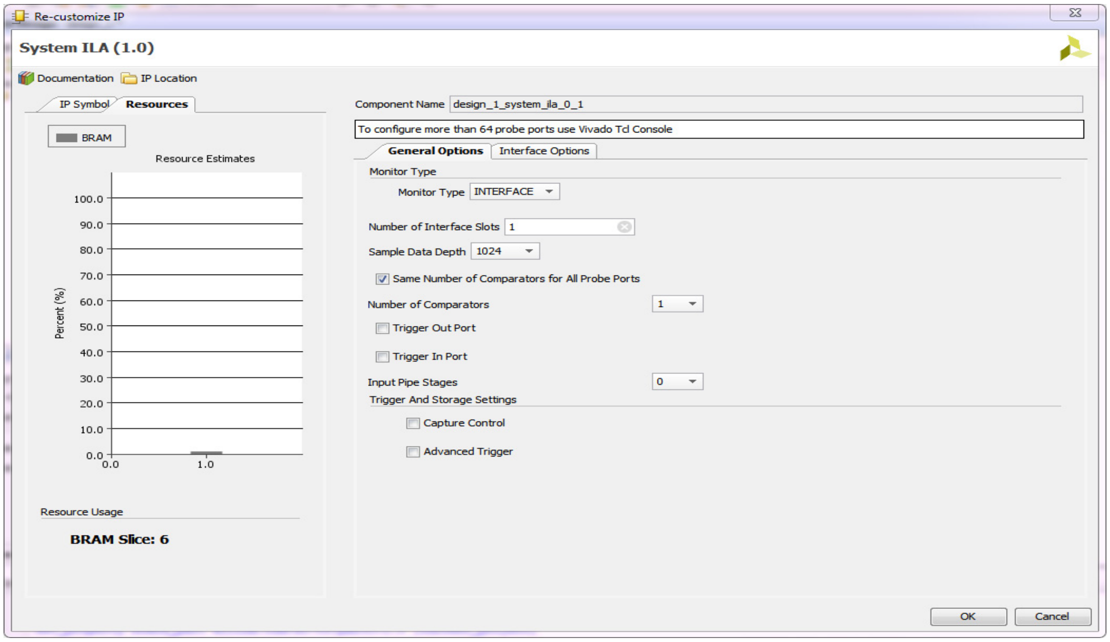
Details

Name: **System ILA**
Version: 1.0
Description: The System ILA core is a customizable logic analyzer core that can be used to monitor a combination of AXI based interfaces and individual signals. This core extend the features of basic ILA for easier debugging at a system level. This includes Boolean trigger equations, customizable data capture buffer depth and optional trigger input/output ports at both signal level as well as for interfaces. The System ila core is synchronous to the design being monitored and hence all clock constraints that are applied to your design are also applied to the components inside the core. Run-time interaction with this core requires the use of Vivado Logic Analyzer feature.
Status: Production
License: Included
Change Log: View Change Log
Vendor: Xilinx, Inc.
VLNV: xilinx.com:ip:system_ila:1.0
Repository: C:/Xilinx/Vivado/2016.3/data/ip

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

General Options

Figure 4-2 shows the **General Options** tab (*Interface* Monitor Type selected). This allows you to specify general configuration options.



BRAM Slice: 6

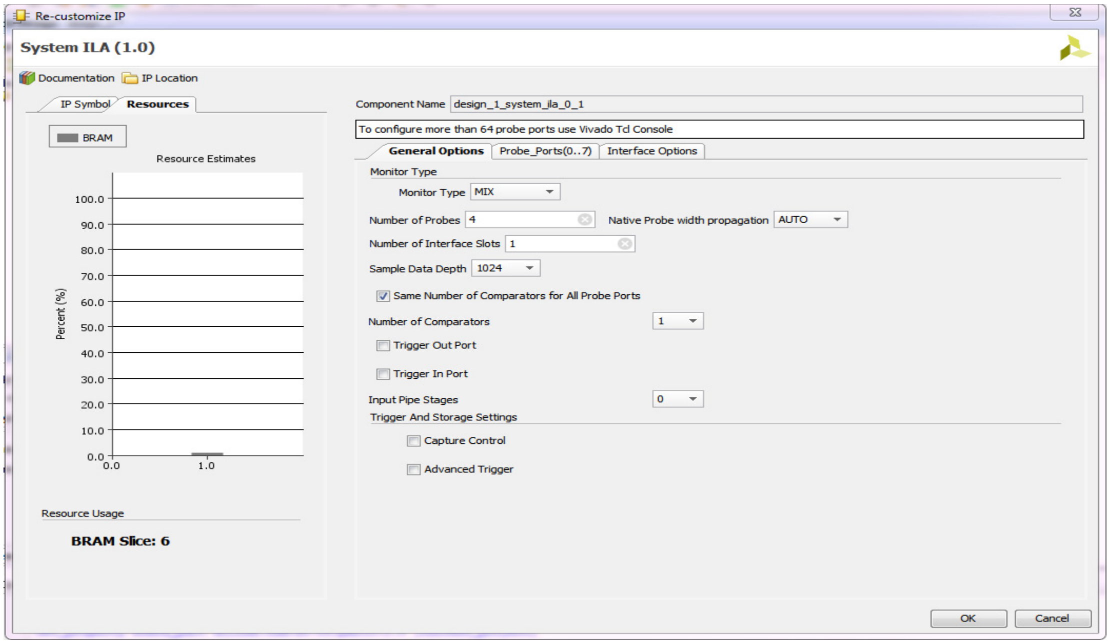
Component Name: design_1_system_ila_0_1

To configure more than 64 probe ports use Vivado Tcl Console

Monitor Type: INTERFACE
Number of Interface Slots: 1
Sample Data Depth: 1024
☑ Same Number of Comparators for All Probe Ports
Number of Comparators: 1
☐ Trigger Out Port
☐ Trigger In Port
Input Pipe Stages: 0
☐ Capture Control
☐ Advanced Trigger

Figure 4-2: **General Options Tab (interface Monitor Type)**

Figure 4-3 shows the **General Options** tab (*Mix* Monitor Type selected). This allows you to specify general configuration options.



BRAM Slice: 6

Component Name: design_1_system_ila_0_1

To configure more than 64 probe ports use Vivado Tcl Console

Monitor Type: MIX

Number of Probes: 4

Native Probe width propagation: AUTO

Number of Interface Slots: 1

Sample Data Depth: 1024

☑ Same Number of Comparators for All Probe Ports

Number of Comparators: 1

☐ Trigger Out Port

☐ Trigger In Port

Input Pipe Stages: 0

☐ Capture Control

☐ Advanced Trigger

Figure 4-3: **General Options Tab (Mix Monitor Type)**

Component Name - Use this field to provide a unique module name for the System ILA core.

Monitor Type - This option specifies which type of interface that System ILA should be debugging. Values for this parameter are **INTERFACE**, **NATIVE** and **MIX**.

- Select **INTERFACE** to monitor interface level signals.
- Select **NATIVE** to monitor scaler signals.
- Select **MIX** to monitor both signal types.

Number of Probes - Use this field to configure the number of probe ports required on the System ILA core. The valid range used in the Vivado IDE is 1 to 64. If more than 64 probe ports are required, use the Tcl command flow to generate the System ILA core.

Native Probe Width propagation - Update the Native probe width propagation to **MANUAL** in order set the individual probe width parameters.

Number of Interface Slots - Select the number of interface slots on the System ILA core. The valid range in the Vivado IDE is 1-16.

Sample Data Depth - Select the required sample depth from the drop-down menu.

Same Number of Comparators for all Probes - Check to enable the same number of comparators for all the enabled ports and interfaces. This option is available in both Basic and Advanced Trigger modes.

Number of Comparators - Select to enable the number of comparators that applies to all enabled probes. The maximum number of comparators is 16 per probe. The number of comparators can be set from 1 to 16 in both Basic and Advanced Trigger modes.

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.

Capture Control - Check to enable the qualifier for the trace capture.

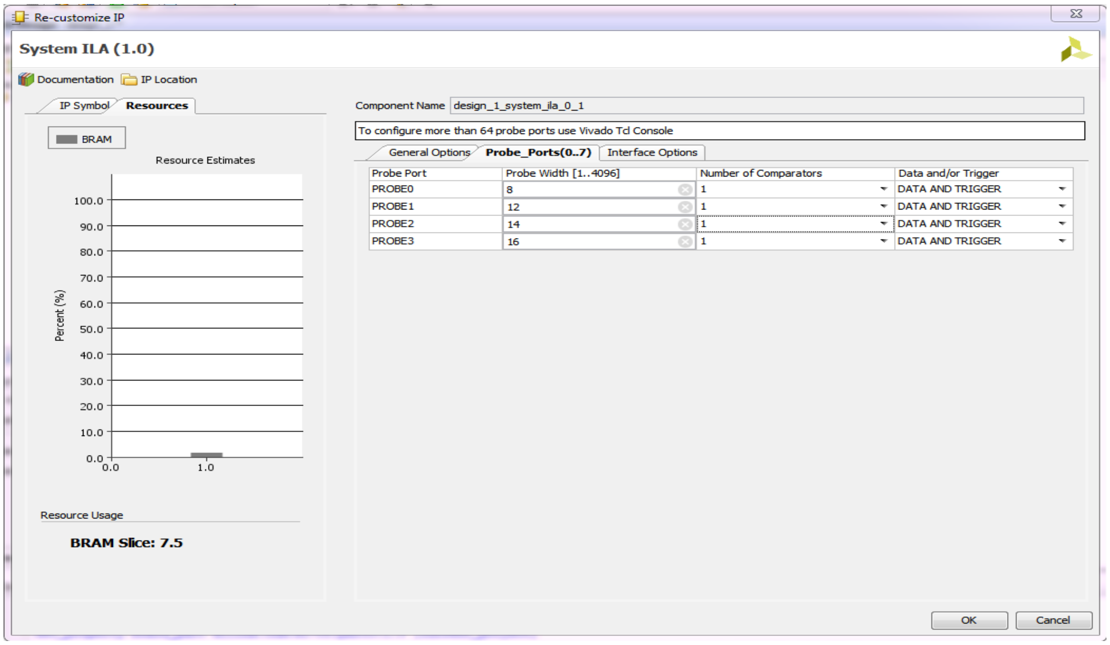
Advanced Trigger - Check to enable the state machine-based trigger sequencing.

Probe Ports

Figure 4-4 shows the **Probe Ports** tab in the Native/MIX monitor type. This tab allows you to configure Probe Port options.

Probe Port Width can be configured in this tab. The width of the individual probes are editable only when the **Native Probe Width Propagation** parameter value is set to **MANUAL** (see Figure 4-3). Otherwise, the width of each individual probe is set automatically based on IP Integrator parameter propagation. Each Probe Port Panel has up to eight ports.

The number of comparators per probe can be configured on this tab. This option is available only when the **Advanced Trigger** option is selected and the **Same Number of Comparator for all Probes** option is cleared on the first page of Vivado IDE (see Figure 4-3).



Component Name design_1_system_ila_0_1

To configure more than 64 probe ports use Vivado Tcl Console

Probe Port	Probe Width [1..4096]	Number of Comparators	Data and/or Trigger
PROBE0	8	1	DATA AND TRIGGER
PROBE1	12	1	DATA AND TRIGGER
PROBE2	14	1	DATA AND TRIGGER
PROBE3	16	1	DATA AND TRIGGER

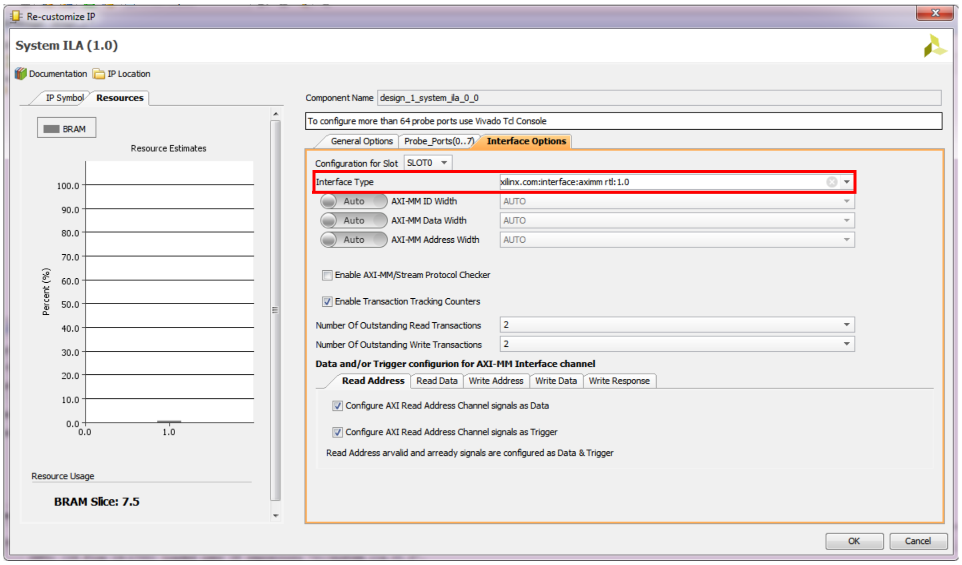
Resource Usage

BRAM Slice: 7.5

Figure 4-4: **Probe Ports Tab**

Interface Options

Figure 4-5 shows the **Interface Options** tab in the Interface/MIX monitor type with the AXI4-MM Interface type selected (highlighted).



BRAM Slice: 7.5

Component Name design_1_system_ila_0_0

To configure more than 64 probe ports use Vivado Tcl Console

Configuration for Slot SLOT0

Interface Type xilinx.com:interface:aximm rtl:1.0

Auto AXI-MM ID Width AUTO

Auto AXI-MM Data Width AUTO

Auto AXI-MM Address Width AUTO

☐ Enable AXI-MM/Stream Protocol Checker

☑ Enable Transaction Tracking Counters

Number Of Outstanding Read Transactions 2

Number Of Outstanding Write Transactions 2

Data and/or Trigger configurion for AXI-MM Interface channel

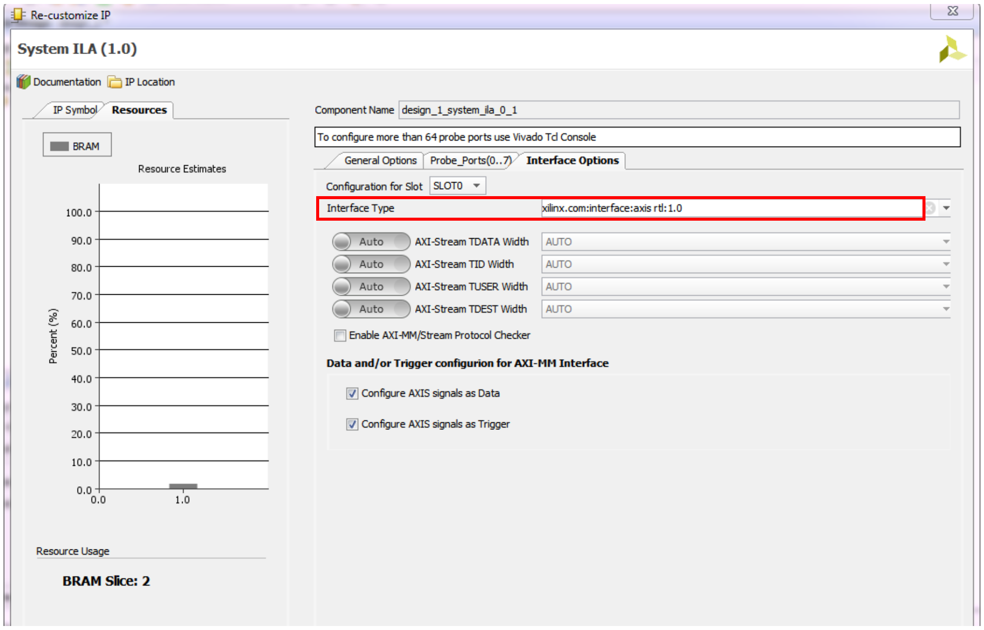
☑ Configure AXI Read Address Channel signals as Data

☑ Configure AXI Read Address Channel signals as Trigger

Read Address arvalid and arready signals are configured as Data & Trigger

Figure 4-5: **Interface Options Tab (AXI4-MM)**

Figure 4-6 shows the **Interface Options** tab in the Interface/MIX monitor type with the AXI4-Stream Interface type selected (highlighted).



BRAM Slice: 2

Component Name design_1_system_ila_0_1

To configure more than 64 probe ports use Vivado Tcl Console

Configuration for Slot SLOT0

Interface Type xilinx.com:interface:axis rtl:1.0

Auto AXI-Stream TDATA Width AUTO

Auto AXI-Stream TID Width AUTO

Auto AXI-Stream TUSER Width AUTO

Auto AXI-Stream TDEST Width AUTO

☐ Enable AXI-MM/Stream Protocol Checker

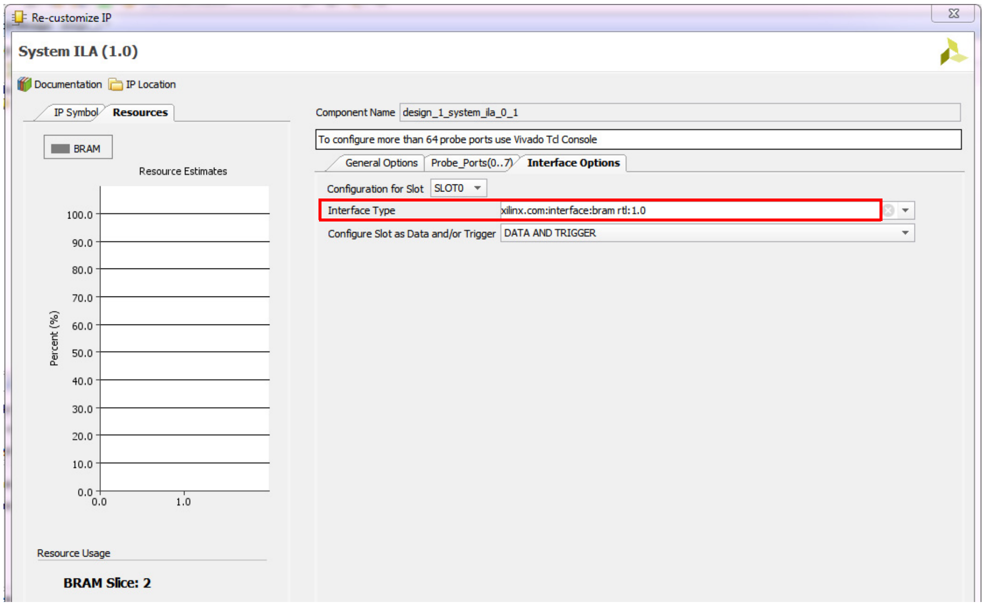
Data and/or Trigger configurion for AXI-MM Interface

☑ Configure AXIS signals as Data

☑ Configure AXIS signals as Trigger

Figure 4-6: **Interface Options Tab (AXI4-Stream)**

Figure 4-7 shows the **Interface Options** tab in the Interface/MIX monitor type with a Non AXI Interface type selected (highlighted).



Re-customize IP

System ILA (1.0)

Component Name: design_1_system_ila_0_1

To configure more than 64 probe ports use Vivado Tcl Console

Configuration for Slot: SLOT0

Interface Type: xilinx.com:interface:bram rtl:1.0

Configure Slot as Data and/or Trigger: DATA AND TRIGGER

Resource Usage

BRAM Slice: 2

Figure 4-7: **Interface Options Tab (Non AXI)**

Configuration for Slot - Selects the parameters corresponding to **slot_<p>** (where <p> is the slot number).

Interface Type - VLNV [Ref 11] of the interface to be monitored by the System ILA core.

AXI-MM Protocol - Selects the protocol of AXI interface when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

AXI-MM ID Width - Selects the ID width of the AXI interface when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

AXI-MM Data Width - Selects the Data width of the AXI interface when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

AXI-MM Address Width - Selects the Address width of the AXI interface when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Enable AXI-MM/Stream Protocol Checker - Enables AXI4-MM or AXI4-Stream protocol checker for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Enable Transaction Tracking Counters - Enables AXI4-MM transaction tracking capability.

Number of Outstanding Read Transactions - Specifies the number of outstanding Read transactions per ID. The value should be equal to or greater than the number of outstanding Read transactions for that connection.

Number of Outstanding Write Transactions - Specifies the number of outstanding Write transactions per ID. The value should be equal to or greater than the number of outstanding Write transactions for that connection.

Monitor APC Status signals - Enable monitoring of APC status signals for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI read address channel as Data - Select read address channel signals for data storage purpose for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI read address channel as Trigger - Select read address channel signals for specifying trigger condition for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI read data channel as Data - Select read data channel signals for data storage purposes for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI read data channel as Trigger - Select read data channel signals for specifying trigger conditions for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write address channel as Data - Select write address channel signals for data storage purpose for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write address channel as Trigger - Select write address channel signals for specifying trigger conditions for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write data channel as Data - Select write data channel signals for data storage purpose for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write data channel as Trigger - Select write data channel signals for specifying trigger condition for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write response channel as Data - Select write response channel signals for data storage purposes for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

Configure AXI write response channel as Trigger - Select write response channel signals for specifying trigger condition for slot <p> when the slot_<p> interface type is configured as AXI-MM, where <p> is the slot number.

AXI-Stream Tdata Width - Selects the Tdata width of the AXI-Stream interface when the slot_<p> interface type is configured as AXI-Stream>, where <p> is the slot number.

AXI-Stream TID Width - Selects the TID width of the AXI-Stream interface when the slot_<p> interface type is configured as AXI-Stream>, where <p> is the slot number.

AXI-Stream TUSER Width - Selects the TUSER width of the AXI-Stream interface when the slot_<p> interface type is configured as AXI-Stream>, where <p> is the slot number.

AXI-Stream TDEST Width - Selects the TDEST width of the AXI-Stream interface when the slot_<p> interface type is configured as AXI-Stream>, where <p> is the slot number.

Configure AXIS signals as Data - Select AXI4-Stream signals for data storage purpose for slot <p> when the slot_<p> interface type is configured as AXI-Stream where <p> is the slot number.

Configure AXIS signals as Trigger - Select AXI4-Stream signals for specifying trigger condition for slot <p> when the slot_<p> interface type is configured as AXI-Stream, where <p> is the slot number.

Configure Slot as Data and/or Trigger - Selects non-AXI slot signals for specifying trigger condition or for data storage purpose or for both for slot <p> when the slot_<p> interface type is configured as non-AXI, where <p> is the slot number.

BRAM Resources estimation graph - The bar graph gives an estimates percentage consumption of BRAM slices for the device part being used.

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

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.

Appendix A

Upgrading

This appendix is not applicable for the first release of the core.

Appendix B

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 System ILA, the Xilinx Support web page 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 System 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 or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. 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.

Technical Support

Xilinx provides technical support at the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Debug Tools

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

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also 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 devices.

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 6].

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 6].

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 debug feature is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debug feature 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.

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: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. *ISE to Vivado Design Suite Migration Guide* (UG911)
6. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
7. *Vivado Design Suite User Guide: Implementation* (UG904)
8. *LogiCORE IP AXI Interconnect Product Guide* (PG059)
9. *LogiCORE IP AXI4-Stream Protocol Checker* (PG145)
10. *LogiCORE IP AXI Protocol Checker* (PG101)
11. VLNV (AR# 50478)

Training Resources

1. Vivado Design Suite Hands-on Introductory Workshop
2. Vivado Design Suite Tool Flow

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/07/2017	1.0	Updated IP Facts Introduction text.
04/05/2017	1.0	Three interface parameters added.
11/30/2016	1.0	Debug chapter updated for core name.
10/05/2016	1.0	Initial Xilinx release.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.