

Designing a System Using the Aurora 64B/66B Core (Simplex) on the KC705 Evaluation Kit

Author: Dinesh Kumar, Ramachandra Thupalli and K Krishna Deepak

Summary

This application note explains the steps required to validate the Xilinx LogiCORE[™] Aurora 64B/66B IP core on the Kintex®-7 FPGA KC705 Evaluation Kit. Aurora 64B/66B is a scalable, lightweight, high data rate, link-layer protocol for high-speed serial communication. Aurora is designed to enable easy implementation of Xilinx transceivers using an intuitive wizard interface. The Aurora protocol specification is open and available upon request. The Aurora core is available free of charge in the Vivado® IP catalog and is licensed for use in Xilinx silicon devices.

Aurora is typically used in applications where other industry standard serial interfaces are either too complex or resource intensive. Aurora delivers a low-cost, high data rate, scalable and flexible means to build a serial data channel. Its simple framing structure can be used to encapsulate data from existing protocols, and electrical requirements are compatible with commodity equipment. Aurora can be used to provide increased performance without high FPGA resource costs, software redevelopment, or exotic physical infrastructure.

The reference design is targeted for the Xilinx Kintex-7 FPGA KC705 evaluation board.

Included Systems

The reference design is created and built using the Vivado Design Suite: System Edition 2014.1. The Vivado Design Suite helps simplify the task of instantiating, configuring, and connecting IP blocks to form complex integrated systems. The design also includes VIO and ILA cores to probe the signals.

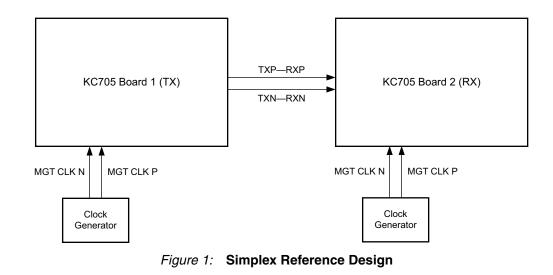
Introduction

This application note details the steps required to configure the Aurora 64B/66B core with Vivado Design Suite and to validate the operation of the core in simplex mode using the VIO and ILA cores to probe various signals.

The example presented is a single-lane simplex configuration using two platforms (see Figure 1). The completed example design can be used to form a building block for more complex systems.

The example test setup uses two clock sources to generate the 156.25 MHz clock signals. *Any suitable conditioned 156.25 MHz clock source can be used to replicate these examples.*

© Copyright 2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.



Hardware Requirements

The single-lane simplex configuration requires these hardware components:

- Two Kintex-7 FPGA KC705 evaluation boards
- Two KC705 Universal 12v power adapters
- Two suitable clock generators to generate 156.25 MHz
- Two JTAG platform USB cables
- Four SMA to SMA connector cables (for reference clock)
- Two SMA to SMA connector cables (for serial data)

Software Requirements

Software requirements for the Aurora 64B/66B simplex example design:

Vivado Design Suite 2014.1

Building Hardware

Simplex Example Design

Customizing the Aurora Core

Follow these steps to customize and generate the Aurora 64B/66B core for the simplex example design:

- 1. Launch Vivado Design Suite.
- 2. Select Create New Project and click Next (Figure 2).

Vivado 2014.1	
ile Fļow <u>T</u> ools <u>W</u> indow <u>H</u> elp	Q- Search commands
VIVADO. Productivity. Multiplied.	
Quick Start	
Create New Project Open Project Open Example Project	
Manage IP Open Hardware Manager Xilinx Td Store	
Information Center	
Documentation and Tutorials Quick Take Videos Release Notes Guide	
) Td Console	
w Project Wizard will guide you through the process of selecting design sources and a target device for a new project.	

Figure 2: Create New Vivado Project

3. Select the project name and path and click **Next** (Figure 3).

🚴 New Project	•	×
Project Name	e	
Enter a nam	ne for your project and specify a directory where the project data files will be stored	
Project name:	xapp 1212	8
Designations	C:/Projects/Vivado	
Project location:	: C:/Projects/vivado	
📝 Create proje	ect subdirectory	
Project will be cr	reated at: C:/Projects/Vivado/xapp1212	
	< Back Next > Einish	Cancel

Figure 3: New Project Name

4. Select **RTL Project** to permit running the example design and check **Do not specify sources at this time** (Figure 4). Click **Next**.

New Project	x
Project Type Specify the type of project to create.	
 <u>BTL Project</u> You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis. <u>Ponot specify sources at this time</u> <u>Post-synthesis Project</u> You will be able to add sources, view device resources, run design analysis, planning and implementation. <u>Do not specify sources at this time</u> <u>J/O Planning Project</u> Do not specify design sources. You will be able to view part/package resources. <u>Imported Project</u> Create a Vivado project from a Synplify, XST or ISE Project File. 	
< Back Next > Einish Cancel	

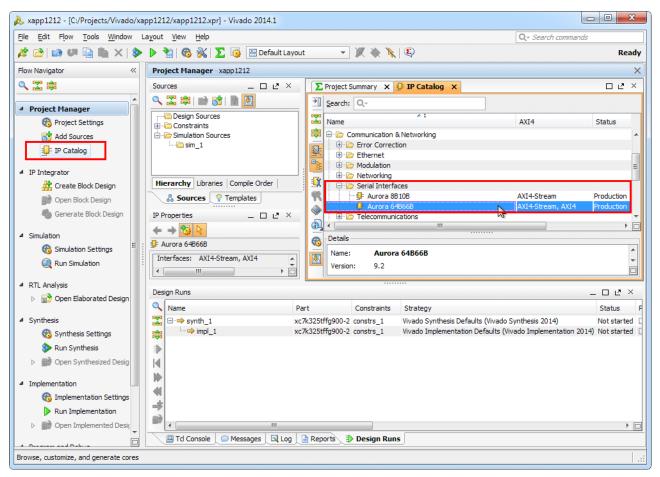
Figure 4: New Project Type

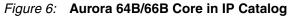
5. Click **xc7k325tffg900-2** or, select the **Boards** option and then click **Kintex-7 KC705 Evaluation platform** (Figure 5).

🚴 New Project						X
Default Part Choose a default Xilinx part or board for your p	project. This can	i be changed la	ater.			~
Specify Filter Parts Boards	Ve <u>n</u> dor Display <u>N</u> ame Board Re <u>v</u>	All	v v			
Search: Q Display Name	Vendor	Board Rev	Part	I/O Pin Count	File Version	Availab IOBs
MicroZed Board	em.avnet.com	e	xc7z010clg400-1	400	1.0	100
ZedBoard Zyng Evaluation and Development Kit	em.avnet.com	d	xc7z020clg484-1	484	1.0	200
Artix-7 AC701 Evaluation Platform	xilinx.com	1.0	xc7a200tfbg676-2	676	1.0	400
Kintex-7 KC705 Evaluation Platform	xilinx.com	1.1	🔷 xc7k325tffg900-2	900	1.0	500
Virtex-7 VC707 Evaluation Platform	xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.0	700
Virtex-7 VC709 Evaluation Platform	xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.0	850
ZYNQ-7 ZC702 Evaluation Board	xilinx.com	1.0	xc7z020clg484-1	484	1.0	200
ZYNQ-7 ZC706 Evaluation Board	xilinx.com	1.1	xc7z045ffg900-2	900	1.0	362
<						•
			< <u>B</u> ack <u>N</u> ext	Eini:	sh C	Cancel

Figure 5: New Project Default Part

- 6. Click Next, then click Finish.
- Under Project Manager in the Flow Navigator panel, select IP catalog and search for Aurora 64B66B. The Aurora cores can be found under Communication & Networking > Serial Interfaces (Figure 6).





8. Right-click Aurora 64B66B and select Customize IP.

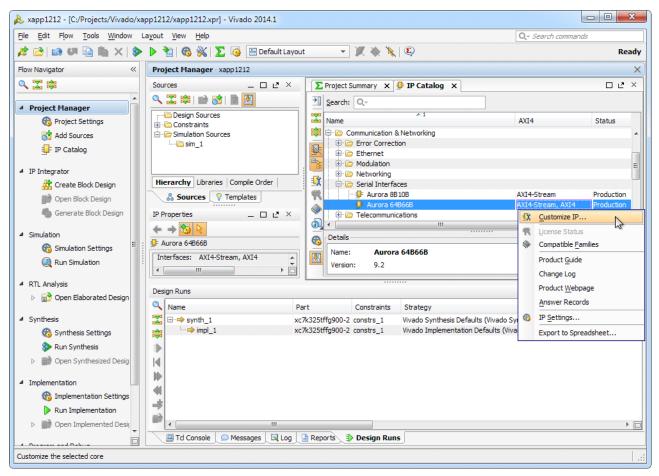


Figure 7: Customize IP

- 9. In the **Core Options** tab of the Customize IP window, set these options (see Figure 8):
 - Set Line Rate (Gbps) to 3.125 and GT Refclk (MHz) to 156.250
 - Set Dataflow Mode to TX-only Simplex or RX-only Simplex, depending on the platform being configured
 - Set Interface to Framing and Flow Control to None.
 - Select the **Vivado Lab Tools** option.

Great Customize IP	••••••••••••••••••••••••••••••••••••••
Documentation in the Location is Switch to Defaults	Component Name aurora_54b66b_0 Core Options GT Selections Shared Logic Line Rate (Gbps) GT Refick (MHz) Dataflow Mode TK-only Simplex Interface Framing Corrol USER K CV Wrado Lab Tools Little Endian Support CRC DRP Mode Additional transceiver control and status ports
	OK Cancel

Figure 8: Aurora 64B/66B Simplex Core Options Settings

- 10. Click the **GT Selections** tab.
- 11. Change the default setting in the lower list box for GTXQ0 from 1 to X.
- 12. Change the lower list box setting for GTXQ2 from **X** to **1** (Figure 9).

Note: The GTXQ2 transceiver is the only transceiver pinned out to SMA connectors on the KC705 board. When placing the cursor over the list box setting, a tooltip appears to verify the location of the selected transceiver.

🖵 Customize IP	-X-
Aurora 64B66B (9.2)	4
Documentation 📄 IP Location 🗔 Switch to Defaults	
Show disabled ports	Component Name Burora_64b66b_0 Core Options GT Selections Shared Logic Columns left Lanes 1 GT Columns GT Type gtx Lane Assignment Note: Lane number selection is for enabling the lane only not for assigning a number to the lane. TrkQ3 X X X X X
۶	Cancel

Figure 9: Aurora 64B/66B Simplex GT Selections

- 13. Options on the **Shared Logic** tab should remain at default values. Click **OK**.
- 14. In the Generate Output Products window, click Generate.

Synthesizing the Example Design

1. When product generation is complete, in the Project Manager section of the Vivado IDE, right-click the core name and select **Open IP Example Design** (see Figure 10).

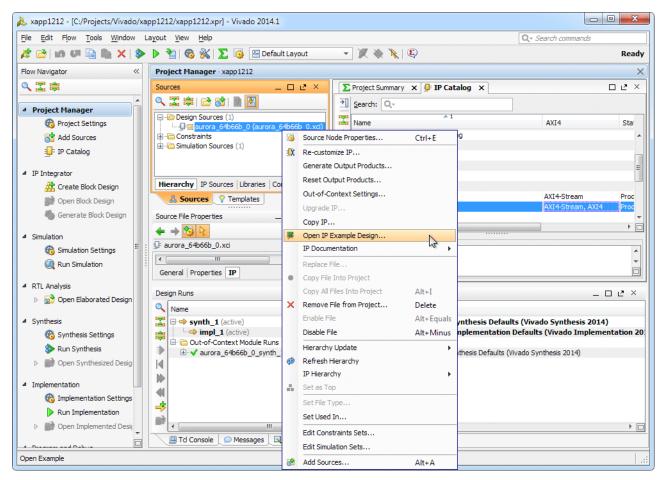


Figure 10: Open IP Example Design

- 2. Click OK to overwrite the existing example design.
- 3. In the newly-opened Vivado IDE window, expand the Constraints entry in the Sources panel of the Project Manager section. Right-click the constraints file (aurora_64b66b_0_exdes.xdc) and select **Open file** (Figure 11).

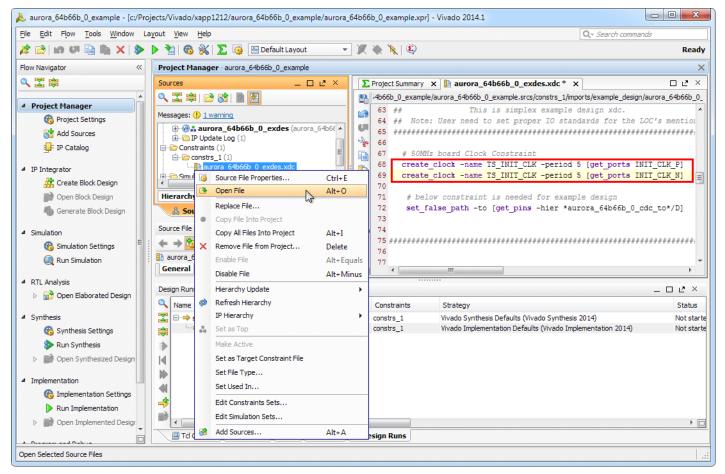


Figure 11: Open Constraints File

- 4. Locate the two 50 MHz board clock constraints (see Figure 11).
- 5. To accommodate the onboard 200 MHz clock, change the clock period from 20 ns to 5 ns. The corrected constraint statements should appear as:

create_clock -name TS_INIT_CLK -period 5 [get_ports INIT_CLK_P]
create_clock -name TS_INIT_CLK -period 5 [get_ports INIT_CLK_N]

6. Assign the pin locations for the Aurora core ports to those shown in Table 1 (see Figure 12).

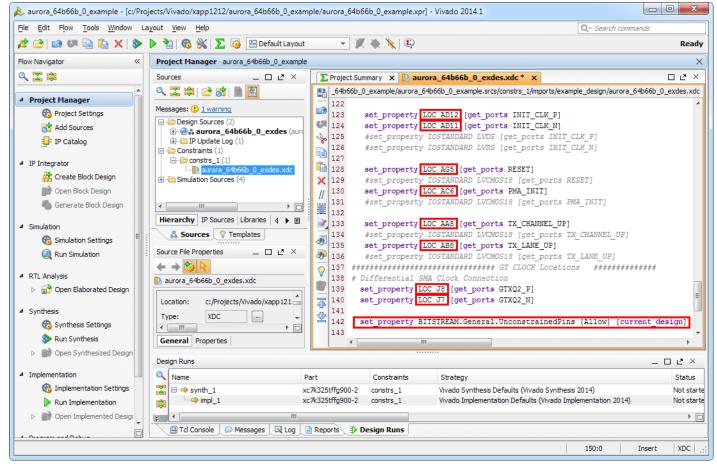


Figure 12: Aurora 64B/66B Simplex LOC Constraints

Table	1:	Aurora	64B/66B	Simplex	Constraints
-------	----	--------	---------	---------	-------------

Pin Name	Loc Value
INIT_CLK_N	AD11
INIT_CLK_P	AD12
RESET	AG5
PMA_INIT	AC6
TX_CHANNEL_UP/RX_CHANNEL_UP	AA8
TX_LANE_UP/RX_LANE_UP	AB8
GTXQ2_N	J7
GTXQ2_P	J8

7. This example contains unconstrained pins. To permit bitsream file generation, add this line to the end of the constraints file (Figure 12):

set_property BITSTREAM.General.UnconstrainedPins {Allow} [current_design]

Caution! Spelling is critical. Double-check changes to the constraints file before proceeding.

- 8. Right-click within the constraints file editor window and select **Save File**. Close the constraints file editor window.
- 9. Select Generate Bitstream from the Flow Navigator panel.

- 10. Click **Yes** to launch Synthesis and Implementation and proceed with bitstream file generation.
- 11. Repeat the steps under Customizing the Aurora Core and Synthesizing the Example Design to generate the bitstream file for the alternate platform:
 - Set Dataflow Mode to TX-only Simplex for the transmit platform
 - Set Dataflow Mode to RX-only Simplex for the receive platform

Executing the Reference Design in Hardware

Setting up the Simplex Example Design

This example illustrates a single-lane Aurora 64B/66B simplex connection between two platforms (see Figure 1, page 2). The platforms consist of two Kintex-7 FPGA KC705 Evaluation Kit boards shown in Figure 13.



Figure 13: KC705 Board Features

In these instructions, numbers in parentheses correspond to callout numbers in Figure 13. Make these connections using the SMA to SMA connector cables.

- Connect TXP from board 1 (4) to RXP of board 2 (5).
- Connect TXN from board 1 (7) to RXN of board 2 (6).
- Connect CLKP from clock source 1 to MGT CLK P of board 1 (2).
- Connect CLKN from clock source 1 to MGT CLK N of board 1 (3).
- Connect CLKP from clock source 2 to MGT CLK P of board 2 (2).
- Connect CLKN from clock source 2 to MGT CLK N of board 2 (3).
- Connect a JTAG platform USB cable from the host PC to the platform cable header of board 1 (1).
- Connect a JTAG platform USB cable from the host PC to the platform cable header of board 2 (1).

- Connect a KC705 Universal 12v power adapter cable to the power connector (9) of both boards.
- Set the power switch (8) of both boards to the ON position.

The completed setup should resemble that shown in Figure 14.

Note: Separate clock sources should be used for each board.

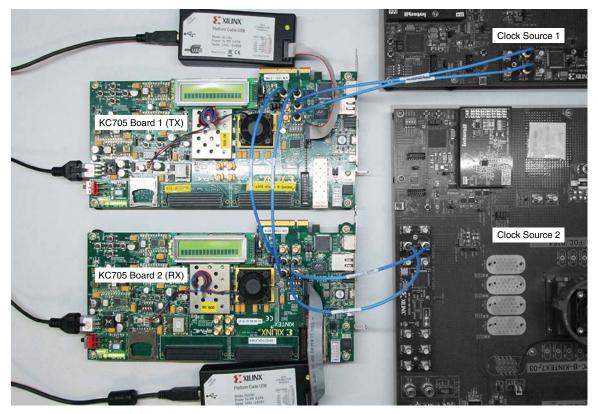
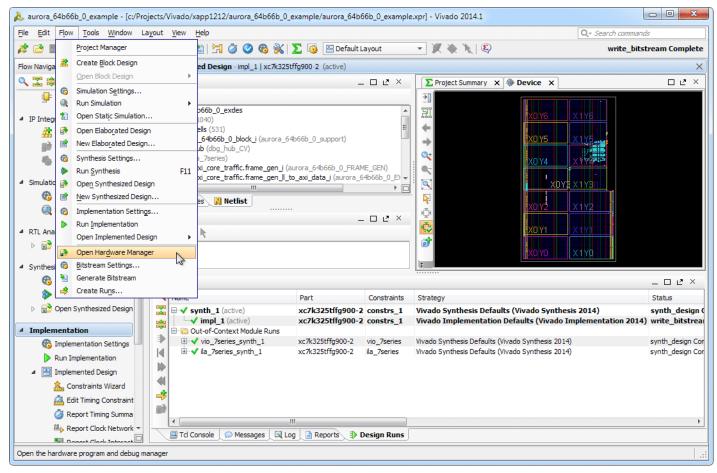


Figure 14: Aurora 64B/66B Simplex Setup

Setting Up the Simplex Example Hardware Session

Programming the Devices

 On completion of bitstream generation, select Flow > Open Hardware Manager (Figure 15).





2. At the top of the Hardware Manager panel (see Figure 16), click **Open a new hardware target** and Click **Next**.

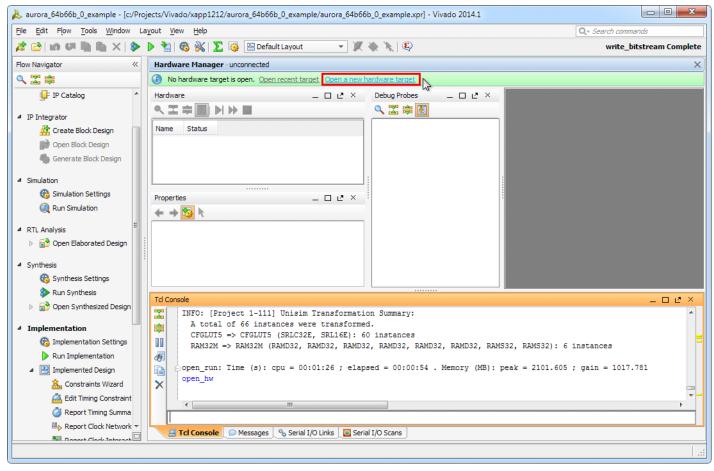


Figure 16: Open a New Hardware Target

3. Select Local server and click Next (Figure 17).

Note: This operation assumes the hardware target is connected to the host PC running Vivado Design Suite. It is possible to connect the hardware target to a second, networked host PC using the Vivado CSE Server application. For details, see the *Vivado Design Suite User Guide: Programming and Debugging* (UG908), [Ref 4].

k	Open New Hardware Target
	erver Settings
	al or remote hardware server, then configure the host name and port settings. Use Local server if the tatched to the local machine; otherwise, use Remote server.
Connect to:	Local server (target is on local machine)
Click Novt to I	launch and/or connect to the vcse_server (port 60001) and hw_server (port 3121) applications on the local
machine.	raunen andzer connectio the wiselserver (port 00001) and hwiserver (port 5121) applications on the local
	< <u>Back</u> <u>Einish</u> Cancel

Figure 17: Hardware Server Settings

4. On the Select Hardware Target page, set the **JTAG Clock Frequency** for both boards to **750000 Hz** (Figure 18).

		—,	oen New Hardware Targ	Jer	
elect Hardw	are Tar <u>o</u>	jet			
				ropriate JTAG clock (TCK) freqi	uency. If 🛛 🧹
you do not	see the	expected devices, decrea	se the frequency or select a d	ifferent target.	
rdware Tar	-				
Туре	Port	Name	JTAG Clock Frequency		
xilinx_tcf			750000 Hz		
xilinx_tcf		Xilinx/00001176367101	750000 Hz		
			1500000 Hz		
			3000000 Hz		
			6000000 Hz		
			12000000 Hz		
rdware Dev	ices (for	unknown devices, specify	the Instruction Register (IR) ler	nath)	
			the Instruction Register (IR) let	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name	ID C	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name 9 xc7k325t_	ID 0 3365	ode IR Length 1093 6	the Instruction Register (IR) ler	ngth)	
Name 9 xc7k325t_	ID 0 3365	ode IR Length	the Instruction Register (IR) ler	ngth)	
Name • xc7k325t_ :SE server:	0 3365	ode IR Length 1093 6	the Instruction Register (IR) ler	ngth)	
Name • xc7k325t_ :SE server:	0 3365	intered IR Length	the Instruction Register (IR) ler	ngth)	

Figure 18: Select Hardware Target

- 5. Highlight the target board to be programmed and click **Next**, then **Finish**.
- 6. In the Hardware panel, click the active device, XC7K325T_0(0) (Active).

7. In the Hardware Device Properties panel, set **Programming file** to the bitstream file name for the receive platform (aurora_64b66b_0_exdes.bit) and set **Probes file** to the appropriate .ltx probes file name (debug_nets.ltx). See Figure 19.

🚴 project_TX_sim	plex_64b66b - [/proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/project_TX_simplex_64b66b.xpr] - Vivado 2014.1
<u>File Edit Flow Tools Window</u>	Layout View Help
🯄 🔂 🕼 🖉 🗎 🐘 🗙 👂 🕨	🔸 🚵 🚳 🛞 🔀 🕼 🔤 Default Layout 💦 👻 🎉 🛞
Flow Navigator	Hardware Manager – localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10)
🔍 🛣 🚔	There are no debug cores. <u>Program device</u> <u>Refresh device</u>
Project Manager G Project Settings C Add Sources IF IP Catalog	Hardware ∠< Debug Probes ∠ Q Image: State of the st
 IP Integrator 	
 In Integration [™] Create Block Design [™] Open Block Design [™] Generate Block Design [™] Simulation [™] Simulation Settings [™] RTL Analysis [™] Open Elaborated Design [™] Synthesis [™] Synthesis [™] Run Synthesis [™] Open Synthesis [™] Open Synthesis [™] Open Synthesis [™] Open Synthesis [™] Open Synthesis 	Anna_Ki/Amin/Jooot IV Golden To populate Probes window, programmable Programmable Programming file: User chain count: 4 Ceneral Properties
 Implementation 	Tcl Console
G Implementation Settings Nun Implementation min Open Implemented Desi Program and Debug	<pre>current_hw_device [lindex [get_hw_devices] 0] Crefresh_hw_device -update_hw_probes false [lindex [get_hw_devices] 0] INFO: [Labtools 27-1432] Device xx7k325t (JTAG device index = 0) is programmed with a design that has 1 ILA core(s) in it. LINFO: [Labtools 27-1832] Device xx7k325t (JTAG device index = 0) is programmed with a design that has 3 VIO core(s) in it. CINFO: [Labtools 27-1832] Device xx7k325t (JTAG device index = 0) is programmed with a design that has 3 VIO core(s) in it. CINFO: [Labtools 27-1832] Device xx7k325t (JTAG device index = 0) is programmed with a design that has 3 VIO core(s) in it. COPE_hw_target [localhost/xilinx_tcf/Xilinx_0000106817c01] INFO: [Labtools 27-147] vccs_server: TCf / Plugin connecting to hw_server</pre>
🚳 Bitstream Settings	INFO: [Labtools 27-147] vcse_server: TCF Plugin connection established with hw_server.
🚵 Generate Bitstream	
🔺 🧱 Hardware Manager 🚽	

Figure 19: Hardware Device Properties

8. Right-click the device in the Hardware list and select **Program Device...** (Figure 20). Ensure that the bitstream file path and name are correct and click **OK**.

🍌 project_TX_simp	lex_64b66b - [/proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/project_TX_simplex_64b66b.xpr] - Vivado 2014.1
<u>Eile Edit Flow Tools Window L</u>	ayout View Help
🯄 😂 🕼 🖉 🗎 🐘 🗙 👂 🕨	1 🚳 🛞 🔽 🎯 🔚 Default Layout 💦 🖉 🌾 🔌 🕼
Flow Navigator	Hardware Manager - localhost/xilinx,tcf/Xilinx/00001008817c01 (xhddplab-10)
Q 🔀 🖨	There are no debug cores. <u>Program device</u> <u>Refresh device</u>
Project Manager	Hardware L × Q 器 等圏 ▶▶■ Q 器 等圏
Received a settings	Name Status
😚 Add Sources	P localhost (xhddplab-10) (2) Connected
💷 IP Catalog	
 IP Integrator 	→ a dware Device Properties Curre
Create Block Design	Program Device
Den Block Design	Ø Refresh Device
Generate Block Design	Add Configuration Memory Device
Generale block besign	Export to Spreadsheet
 Simulation 	To populate There are a set of the set of th
🍪 Simulation Settings	rodes window, program and/or
🔍 Run Simulation	vc7k325t_0
 RTL Analysis 	
🕨 📄 Open Elaborated Design	
 Synthesis 	runs/impl_1/aurora_64b66b_Rx_simplex_exdes.bit
🌀 Synthesis Settings	Trans/mpr_1/debug_nets.nx
📚 Run Synthesis	
👂 📑 Open Synthesized Desigi	General Properties
 Implementation 	Tcl Console
6 Implementation Settings	INFO: [Labtools 27-1432] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 1 ILA core(s) in it.
Run Implementation	E INFO: [Labtools 27-1828] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 3 VIO core(s) in it. closes.hw.target (localhost/xilinx.tcf/xilinx.tcf/xilinx/acd/tilinx/acd/tilin/xilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/tilinx/acd/til
Open Implemented Desi	<pre>pen_hw_target {localhost/xilinx_tcf/xilinx/00001008817c01}</pre>
	INFO: [Labtools 27-147] vcse_server: TCF plugin connecting to hw.server
Program and Debug	set_property PROGRAM.FILE {/proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_64b66b_Rx_simplex_example/aurora_
6 Bitstream Settings	x set_property PROBES.FILE {/proj/1ps/users/thupall1/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_64b66b_Rx_simplex_example/aurora_6
🚵 Generate Bitstream	
4 🧱 Hardware Manager 🛛 🧮	

Figure 20: Program Device

project_TX_simplex_64b66b - [/proj/lps/users/thupalii/XAPP/64b66b_simplex/project_TX_simplex_64b66b/project_TX_simplex_64b66b.xpr] - Vivado 2014. File Edit Flow Tools Window Layout View Help 🯄 😂 🛤 💷 🗎 📉 ≽ 🕨 騺 🚳 % ∑ 🎯 🖭 Default Layout - 🗶 🚸 📜 💽 Flow Navigator « Hardware Manager - localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10) - 🗆 🖻 × ् 🛣 🚔 Hardware 💶 🗆 🖉 🗡 🛛 Debug Probes 🔊 ILA - hw_ila_1 × 🔊 VIO - hw_vio_1 × 🔊 VIO - hw_vio_2 × 🛛 ९ 🛣 🖨 🛃 🕨 🕨 🔳 XIO Probes 🔍 🛣 🖨 🛃 Name Name Value Activity Direction VIO Status Connected **e** 🚯 Project Settings
 Iocalhost (chddplab-10) (2)

 ■ xilinx,tcf/xilinx/00001008817c01 (xx.)

 ● x7A225t,0 (5) (citive)

 ■ wxios,stem Monitor)

 ● bw.vios,1 (Vio)

 ● bw.vios,2 (Vio)

 ● continue,1 (Lik)

 * • 👌 Add Sources G Hardware Target Properties... Ctrl+E \$ 💷 IP Catalog Set as Current Target \times 🧼 Refresh Target ▲ IP Integrator Close Target 👬 Create Block Design 💕 Open Block Design Export to Spreadsheet Drag and drop 🍓 Generate Block Design \rightarrow <const0> 17 VIO probes from Debug Probes Hardware Target Properties - 🗆 🖻 × Simulation **←** → 🗞 🕅 Simulation Settings ∿ <const0>_2 ∿ <const0>_20 localhost/xilinx_tcf/Xilinx/00001008817c01 🔍 Run Simulation - < const0 > _ 21 RTL Analysis Name: localhost/xilinx_tcf/Xilinx/00001 🖻 📑 Open Elaborated Design Type: xilinx tcf 🗹 Is target connection open Synthesis -le <const0>_27 -le <const0>_28 -le <const0>_29 Number of devices: 1 🚳 Synthesis Settings • Run Svnthesis Þ ▼ □ € General Properties Open Synthesized Desig Implementation Tcl Console nsole set_property PROGRAM.FILE {/proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_64b66b_Rx_simplex_example/auror program_hw_devices [lindex [get_hw_devices] 0] INF0: [Labtools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_1 INF0: [Labtools 27-3154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_1 INF0: [Labtools 27-3154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/aurora_1 INF0: [Labtools 27-142] Done pin status: HIGH program_hw_devices [lindex [get_hw_devices] 0] INF0: [Labtools 27-1432] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 1 ILA core(s) in it. INF0: [Labtools 27-1828] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 3 VIO core(s) in it. 🚳 Implementation Settings * Run Implementation Open Implemented De: 10 Program and Debug 🚳 Bitstream Settings × 🚵 Generate Bitstream • 🔺 🧱 Hardware Manager 📸 Open Target -📟 Tcl Console 💭 Messages 🗌 🗞 Serial I/O Links 🗌 🖬 Serial I/O Scans 💣 Program Device

Figure 21: Close Target

10. Right-click the second target platform in the Hardware list and select **Open Target** (Figure 22).

9. When programming completes, right-click the programmed target device in the Hardware list and select **Close Target** (Figure 21).



Run Implementation Run Implementation Info: [Labtools 27-32] Done pin status: High program_hwidevices: Time (s): cpu = 00:00:03; elapsed = 00:03:29 . Memory (HB): peak = 4947.047; gain = 0.000 Program and Debug Program and Debug Bistream Settings Generate Bistream Hardware Manager The Hardware Manager Type a Tcl command here		blex_64b66b - [/proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_64b66b/project_TX_simplex_64b66b.xpr] -
Flow Navgar Forget Manager	<u>File Edit Flow Tools Window I</u>	ayout Yiew Help
	🧦 😂 🕼 💵 🗎 🛍 🗙 🔈 🕨	🛬 🚳 🐝 🔽 🚱 🖽 Default Layout 💦 🗶 🔍
Project Manager Project Manager Project Settings Add Sources Project Settings Add Sources Project Settings Add Sources Project Settings <td>Flow Navigator «</td> <td>Hardware Manager – localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10)</td>	Flow Navigator «	Hardware Manager – localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10)
	🔍 🔄 🖨	3 No hardware target is open. Open closed target Open recent target. Open a new hardware target.
Implementation Implementation Run Implementation Run Implementation Run Implementation More Implementation INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INFO: [Latools 27-1432] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 1 ILA core(s) in it. INFO: [Latools 27-1432] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 3 VID core(s) in it. INFO: [Latools 27-1432] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 3 VID core(s) in it. INFO: [Latools 27-1628] Device xc7k325t (JTAG device index = 0) is programmed with a design that has 3 VID core(s) in it. INFO: Deve Tareat	 Project Manager Project Settings Project Settings Add Sources IP Catalog IP Integrator Create Block Design Open Block Design Generate Block Design Simulation Simulation Settings Rut Simulation RTL Analysis Synthesis Synthesis Settings 	Hardware It X Debug Probes It X Name Status It X It X Icalhost (hddplab-10) (2) Connected It X Icalhost (hddplab-10) (2) Closed It X Icalhost (hddplab-10) (2) Closed It X Icalhost (hddplab-10) (2) Closed It X Icalhost / xilinx_tof/Xilinx/000001068817c01 (0) Close Target Properties Ctrl+E Set as Current Target Refresh Target Close Target Close Target Depen Target Export to Spreadsheet Hardware Target Properties It Icalhost/xilinx_tof/Xilinx/00001176367101 Name: localhost/xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof/Xilinx_tof Is target connection open Number of devices: 0
Gimplementation Settings Run Implementation Debug Program and Debug Generate Bitstream Hardware Manager Type a Tcl command here	Open Synthesized Design	General Properties
Bitstream Settings Concrate Bitstream Hardware Manager Type a Tc1 command here Tc1	Implementation Settings ▶ Run Implementation ▶ min Open Implemented Desi	<pre>program_hw_devices [lindex [get_hw_devices] 0] INF0: [Labtools 27-2154] Reading 11443724 bytes from file /proj/ips/users/thupalli/XAPP/64b66b_simplex/project_TX_simplex_ INF0: [Labtools 27-32] Done pin status: HIGH InF0: [Labtools 27-432] Done pin status: HIGH InF0: [Labtools</pre>
💣 Program Device 📋 🗐 Tcl Console 🔍 Messages 🗞 Serial I/O Links 🔤 Serial I/O Scans	 Generate Bitstream Hardware Manager Open Target 	<pre>Close_hw_target {localhost/xilinx_tcf/xilinx/00001008817c01} Type a Tcl command here </pre>

Figure 22: Open Second Target Platform

- 11. Repeat step 6 and step 7 using the bitstream file name for the transmit platform and the appropriate .ltx probes file name.
- 12. Repeat step 8 to program the device.
- 13. When programming completes, right-click the programmed target device in the Hardware list and select **Refresh Device** (Figure 23).

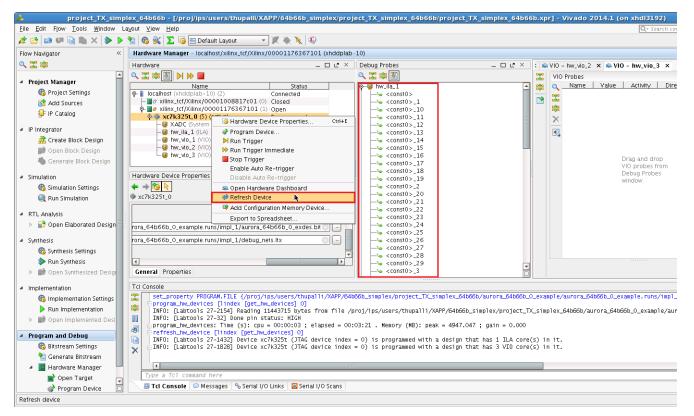


Figure 23: Refresh Device

Executing the Design

1. Right-click the device in the Hardware list and select Run Trigger (Figure 24).

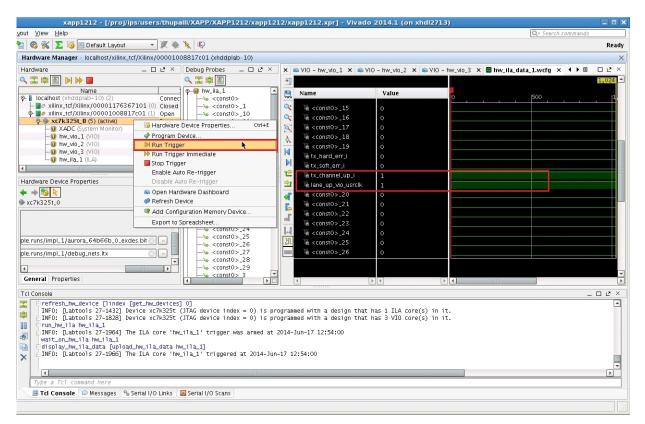


Figure 24: Run Trigger

- 2. In the waveform window that appears, observe a High state on the lane_up_vio_usrclk and tx_channel_up_i signals.
- 3. Control-click to select these signals in the Debug Probes list under hw_vio_1:
 - channel_up_in_initclk
 - lane_up_vio_i
 - gtreset_from_vio_i
 - sysreset_from_vio_i
- 4. Right-click a highlighted signal and select Add Probes to VIO Window (Figure 25).

XILIN>

ut ⊻iew <u>H</u> elp						Q+ Search cor	nmands
🊳 💥 ∑ 🎼 🖭 Default Layout	- X 🚸 🔭 💽						Rea
Hardware Manager – localhost/xilinx_tcf/Xili	inx/00001176367101 (xhddplab-10)						
	_ □ ∟ × Debug Probes _	□ Ľ × VIO	- hw_vio_1 × 🔊 VIO - hw	_vio_2 🗙 🔊 VIO - hw_v	io_3 × 🖀 hw_ila_da	ta_1.wcfg ×	
🔪 🛣 🖨 🛃 🕨 🕪 🔳	a 🔍 🔀 🚔 💽	⇒∎				1,024	
Name	Status	🔺 🖳	Name	Value			
	nnected	cnt 20t			<u> </u>	. <mark>.</mark>	2,000
Q- ■ Ø xilinx_tcf/Xilinx/000010088 Clu Q- ■ Ø xilinx_tcf/Xilinx/000011763 Op	ink_reset_int_in		₩ <const0>_15</const0>	0			
	s anomene al	1.6	🍇 <const0>_16</const0>	0			
- 🔯 XADC (System Monitor)	o-la iteraion_cnt_se		🐚 <const0>_17</const0>	0			
	Idle		\} <const0>_18</const0>	0			
	C – Outputs Res reset test for	from 1	₩ <const0>_19</const0>	0			
	C - Outputs Res		1 tx_hard_err_i	0			
	- Outputs Res	_initclk	The tx_soft_err_i	ŏ			
	Iane_up_vio_i		tx_channel_up_i			_	
Debug Probe Properties	_ □ Ľ × 🕴 🔶 🖕 vio_probe_in2	[7:0]					
► → 🛐 📐		010_1	⅓ lane_up_vio_usrclk	1			
svsreset_from_vio_i	o-te loopback_i[2:0		₩ <const0>_20</const0>	0			
systeset_noni_wo_i	sysreset_from	Néor E	¼ <const0>_21</const0>	0			
Name: svsreset_from_vio_i	● J hw_vio_3	- 🥵 Debug	Probe Properties Ctrl+E	0			
· · · · ·	o-la Inkrst_cnt_20k	nit_ 🔊 🕨 Run Ti	igger	0			
	-ve test_failed_r	🕪 Run Ti	igger Immediate	0			
Probe type: VIO_OUTPUT	-_ test_passed_r		riaaer	0			
Width: 1	—∿a dummy_vio_i_		Auto Re-trigger	ů			
	—Va dummy_vio_i_	3_2 Disabl	e Auto Re-trigger		 		
	-\a dummy_vio_i_	2_2					
General Properties	o-la reset_test_fsm		obes to Waveform				
			Hardware Dashboard				
cl Console			Γ				
	xc7k325t (JTAG device index = 0)						Ŀ
	<pre>xc7k325t (JTAG device index = 0)</pre>						
Elettest_nw_device. time (s). cp	u = 00:00:01 ; elapsed = 00:00:12	. Memory (MB)	: peak = 4947.047 ; gai	n = 0.000			
THEOR Flatterals 37 10643 The TL	A core 'hw_ila_1' trigger was arm	ed at 2014-Mar	-24 12:02:37				
Wait_on_hw_ila_hw_ila_1	and the second sec						
👌 🗄 display_hw_ila_data [upload_hw_							
INFO: [Labtools 27-1966] The IL	A core 'hw_ila_1' triggered at 20.	14-Mar-24 12:0	2:37				1
ं र	1111						Þ
Type a Tcl command here							
Tcl Console Messages Serial	LI(O Links Serial I(O Scans						

Figure 25: Add Probes to VIO Window

5. Toggle the reset signals by clicking the value field for each signal (See Figure 26). Enter 1 or 0 and click **OK**.

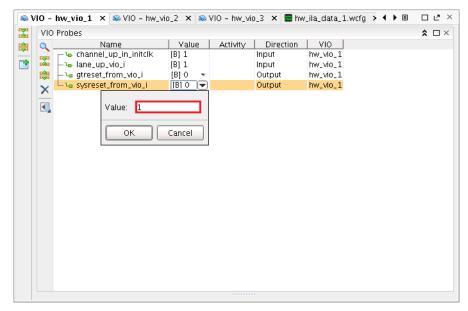


Figure 26: Toggle Reset Signals

6. The channel_up_in_initclk and lane_up_vio_i signals should go Low, then return High after each reset signal is toggled.

Follow these steps to view the results of the reset signals in the waveform display:

- 1. Set one of the reset signals High.
- 2. Right-click the device in the Hardware list and select Run Trigger.
- 3. Click the waveform display tab and observe the results of the reset signal shown in Figure 27.

lex_64b66b - [/proj/ips/u	sers/thupalli/XAPP/64b6	6b_simplex/project_TX_simpl	ex_6	i4b66b/project_TX_sim	plex_64b66b.xpr] - Viva	do 2014.1 (on xho	113192)	
ayout ⊻iew <u>H</u> elp							Q+ Search commai	nds
🚵 🚳 💥 ∑ 🎯 🔚 Defau	ilt Layout 🛛 👻 🕷 🎽	(2) (2)						Ready
Hardware Manager - localh	ost/xilinx_tcf/Xilinx/00001176	367101 (xhddplab-10)						×
Hardware	_ □ Ľ ×	Debug Probes 💷 🗠 🗡	<u>ې</u> ۷	10 - hw_vio_1 🗙 🔊 VIO - hv	w_vio_2 🗙 👒 VIO - hw_vio_3	: 🗙 🔚 hw_ila_data_1	L.wcfg × ◀ ▸	∎ □Ľ×
🔍 🛣 🖨 🛃 🕨 🕨 🔳		🔍 🔀 🖨 🛃	¥				1,024	
Name	Status	● 월 hw_vio_2	8	Name	Value			
P- I localhost (xhddplab-10 - II ocalhost (xhddplab-10 - II ocalhost (xhddplab-10		o-le chan_up_transcnt_20k		10				2,000
P- ■ Ø xilinx_tcf/Xilinx/00		lnk_reset_in_initclk lo soft_err_in_initclk	0-	Va <const0>_15 Va <const0>_16</const0></const0>	0			
∲- § xc7k325t_0 (5) (a	G Hardware Device Prope			Verst0>_16 Verst0>_17	0			
-Standard (System) -Standard (ILA)	Program Device	lity_cntrs	9	Va <const0>_17</const0>	0			
-12 hw_vio_1 (VIO)	Run Trigger	enable_fror fsm_from_v	<u></u>	<pre>\u00e3 < const0 > _10</pre>	ő			
-10 hw_vio_2 (VIO)	▶ Run Trigger Immediate			W tx_hard_err_i	ő			
- ↓ hw_vio_3 (/IO)	📕 Stop Trigger	p_in_initclk		W tx_soft_err_i	ŏ			
	Enable Auto Re-trigger	lo_i _in2[7:0]	12	1∰ tx_channel_up_i	0		1	
Hardware Device Properties	Disable Auto Re-trigger	pm_vio_i	2	𝖓 lane_up_vio_usrclk	0			
← ⇒ 100 k	Se Open Hardware Dashb	1-1-1	4	₩ <const0>_20</const0>	0		-	
xc7k325t_0	🤣 Refresh Device	ten_i rom_vio_i	i,	🍇 <const0>_21</const0>	0			
	Add Configuration Mem	ory Device	-	₩ <const0>_22</const0>	0			
	Export to Spreadsheet.	Property and the second s		₩ <const0>_23</const0>	0			
rora_64b66b_0_example.run	is/impl_1/aurora_64b66b_i	test_failed_r 		\}_24	0			
rora_64b66b_0_example.rur	is/impl_1/debug_nets.ltx	-u dummy_vio_i_3_1	ζη	₩ <const0>_25</const0>	0			
		-u dummy_vio_i_3_2		₩ <const0>_26</const0>	0	ĮĮ		
•		-u dummy_vio_i_3_3 -u dummy_vio_i_3_3						
General Properties				•	• •			× F
Tcl Console								_ D & ×
	-19661 The ILA core 'hw i	la_1' triggered at 2014-Mar-24	12:0	2:37				
🛛 👝 📄 set_property OUTPU		sysreset_from_vio_i -of_object						
commit_hw_vio [get		_vio_i} -of_objects [get_hw_vi	os hv	LV10_1]]				
INFO: [Labtools 27	-1964] The ILA core 'hw_i	la_1' trigger was armed at 2014	4-Mar	-24 12:10:24				
	ila_1 a [upload_hw_ila_data hw_	115 11						
X INFO: [Labtools 27		la_1' triggered at 2014-Mar-24	12:1	LO:24				
								T
•		11111						• •
Type a Tcl command h	ere							
🔚 Tcl Console 🔎 Mes	sages 🛛 🗞 Serial I/O Links 🗌 🖸	Serial I/O Scans						

Figure 27: Reset Signal Results in Waveform

4. Repeat step 2 and step 3 after each change to the reset signals to observe the results.

The preceding steps attempt to demonstrate that when either <code>sysreset_from_vio_i</code> or <code>gtreset_from_vio_i</code> are asserted, both <code>channel_up_in_initclk</code> and <code>lane_up_vio_i</code> go Low as the core (or transceiver) is in reset state. However, when both <code>sysreset_from_vio_i</code> and <code>gtreset_from_vio_i</code> are Low, the core is out of reset state and both <code>channel_up_in_initclk</code> and <code>lane_up_vio_i</code> are High.

Reference Design

Table 2 shows the reference design checklist.

Table 2: Reference Design Checklist

Parameter	Description			
General				
Target devices (stepping level, ES, production, speed grades)	Kintex-7 XC7K325T-2FFG900			
Source code provided	Yes			
Source code format	VHDL/Verilog (some sources encrypted)			
Design uses code/IP from existing Xilinx application note/reference designs, Vivado IP Catalog, or third party	Reference design provided by Aurora core generated from Vivado IP catalog			
Simulation				
Functional simulation performed	No			
Timing simulation performed	No			
Test bench used for functional and timing simulations	N/A			
Test bench format	N/A			
Simulator software/version used	N/A			
SPICE/IBIS simulations	No			
Implementation				
Synthesis software tools/version used	Vivado Design Suite 2014.1			
Implementation software tools/versions used	Vivado Design Suite 2014.1			
Static timing analysis performed	Yes			
Hardware Verification				
Hardware verified	Yes			
Hardware platform used for verification	Kintex-7 FPGA KC705 evaluation kit			

Conclusion

The Kintex-7 FPGA KC705 Evaluation Kit provides an excellent platform to implement and test the LogiCORE IP Aurora 64B/66B core. Following the procedure outlined in this application note, Aurora 64B/66B simplex designs can be verified and extended for specific applications. Various configurations can be quickly evaluated using only the KC705 board, a clock source and the Vivado Design Suite.

References

This application note uses these references:

- 1. LogiCORE IP Aurora 64B/66B Product Guide (PG074)
- 2. Kintex-7 FPGA KC705 Evaluation Kit Getting Started Guide (UG883)
- 3. Vivado Design Suite User Guide: Designing with IP (UG896)
- 4. Vivado Design Suite User Guide: Programming and Debugging (UG908)
- 5. Embedded System Tools Reference Manual (UG111)

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
10/01/2015	1.0.1	Corrected "Aurora 8B/10B" to be "Aurora 64B/66B" in the Conclusion.
01/09/2015	1.0	Initial Xilinx release.

Notice of Disclaimer

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 the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; 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 Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.