

Designing a System Using the Aurora 8B/10B 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 8B/10B IP core on the Kintex®-7 FPGA KC705 Evaluation Kit. Aurora 8B/10B 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 with 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 8B/10B 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)
 - HPC-HPC Samtec cable (for back channel testing with sideband mode)

Software requirements for the Aurora 8B/10B simplex example design:

Software Requirements

• Vivado Design Suite 2014.1

Building Hardware

Simplex Example Design

Customizing the Aurora Core

Follow these steps to customize and generate the Aurora 8B/10B core for the simplex example design:

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

🚴 Vivado 2014.1	
<u>File</u> Flow <u>T</u> ools <u>Window</u> <u>H</u> elp	Q- Search commands
VIVADO. Productivity. Multiplied.	E XILINX ALL PROGRAMMABLE.
Quick Start	
Create New Project Open Project Tasks	
Manage IP Open Hardware Manager Xilmx Td Store	
Information Center	
Documentation and Tutorials Quick Take Videos Release Notes Guide	
I Td Console	
New 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 Enter a nam	e ne for your project and specify a directory where the project data files will be stored	
Project name:	xapp1211	8
Project location:	: C:/Projects/Vivado	
🔽 Create proje	ect subdirectory	
Project will be cro	reated at: C:/Projects/Vivado/xapp1211	
	Z Back Maxt Na Elizich	ancel
		ancer

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
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>Po not specify sources at this time</u> <u>Bost-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>JO 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.
< <u>Back</u> 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	iter.			
Specify Filter	Ve <u>n</u> dor Display <u>N</u> ame Board Re <u>v</u>	All All Latest	v v Filtere			
Search: Q-	Vendor	Board Rev	Part	I/O Pin Count	File Version	Availab IOBs
MicroZed Board	em.avnet.com	e	xc7z010dg400-1	400	1.0	100
ZedBoard Zyng Evaluation and Development Kit	em.avnet.com	d	xc7z020clq484-1	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> </u>						
			< <u>B</u> ack <u>N</u> ext	Eini:	sh C	ancel

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 8B10B. The Aurora cores can be found under Communication & Networking > Serial Interfaces (Figure 6).

\lambda xapp1211 - [C:/Projects/Vivado/xa	app1211/xapp1211.xpr] - Vivado 2014.1	
<u>File E</u> dit F <u>l</u> ow <u>T</u> ools <u>W</u> indow	Layout View Help Qr Search commands	
🯄 😂 🕼 💷 🗎 🐂 🗙 🔌	🔸 ▶ 📸 🚳 🐒 🔽 🚳 😬 Default Layout 💦 🖉 🔆 🔍	Ready
Flow Navigator 🛛 🔍	Project Manager - xapp1211	×
🔍 🔀 🖨	Sources _ 🗆 🗠 × 🛛 😰 Project Summary 🗴 🔮 IP Catalog ×	
Project Manager	조 음। 값	
🏀 Project Settings	AXI4	
Add Sources	🖨 🗁 Simulation Sources	•
=== IP Catalog	□ sim_1 ⊕	
	🖳 🖳 🗁 Modulation	-
4 IP Integrator	Hierzychy Librariae Compile Order	
Create Block Design	AX14-Stream	
Dpen Block Design	AXI4-Stream	, AXI4
Senerate Block Design	IP Properties _ C X Y Brie Telecommunications	-
▲ Simulation		
🏀 Simulation Settings 📱	Aurora 88108	
📖 Run Simulation	Interfaces: AXI4-Stream	-
A RTL Analysis	Design Runs	D @ X
Open Elaborated Design	Name Part Constraints Strategy	Status
4 Synthesis	🔀 ⊡-⇒ synth_1 xc7k325tffg900-2 constrs_1 Vivado Synthesis Defaults (Vivado Synthesis 2014)	Not started
🚳 Synthesis Settings	impl_1 xc7k325tffg900-2 constrs_1 Vivado Implementation Defaults (Vivado Implementation 2014)) Not started
📚 Run Synthesis	•	
Open Synthesized Desig		
4 Implementation		
Implementation Setting		
Run Implementation	*	
Open Implemented Des		
▼	Td Console Messages Log Reports Design Runs	
browse, customize, and generate cores		



8. Right-click Aurora 8B10B and select Customize IP (Figure 7).



🚴 xapp1211 - [C:/Projects/Vivado/x	app1211/xapp1211.xpr] - Vivado 2014.1	
<u>File Edit Flow Tools Window</u>	Layout View Help C	2 → Search commands
🧦 😂 🐼 💵 🗎 🖿 🔌	🕨 🚵 🚳 🐝 ∑ 🧐 😬 Default Layout 💿 🗶 🔖 🍾 🖏	Ready
Flow Navigator	Project Manager - xapp1211	×
Q 🔀 🖨	Sources _ 🗆 🗠 × 🛛 💽 Project Summary 🗙 💷 IP Catalog 🗴	□ Ŀ×
Project Manager	Q Z ⇒ B E 21 Search: Q-	
🍈 Project Settings	B-Constraints	AXI4
👌 Add Sources	Simulation Sources	*
📲 IP Catalog	In Internet	
	Modulation	=
IP Integrator Grapta Black Design	Hierarchy Libraries Compile Order	
Create block Design	& Sources 9 Templates	AXI4-Stream
Caperate Plack Design	Aurora 648668	Customize IP
Generate block Design		🖷 License Status
4 Simulation		Compatible Families
🚳 Simulation Settings 🗏	JF Aurora 88108	Product <u>G</u> uide
🔍 Run Simulation	Interfaces: AXI4-Stream	Change Log
4 PTI Apalycis		Product <u>W</u> ebpage
Open Elaborated Design	Design Runs	Answer Records
	Name Part Constraints Strategy	IP Settings
4 Synthesis	□ → synth_1 xc7k325tffg900-2 constrs_1 Vivado Synthesis Defaults (Vivado	Export to Spreadsheet
🍪 Synthesis Settings		ado Implementation 2014) Not started
🔊 Run Synthesis	•	
Open Synthesized Desig		
Implementation		
nplementation Setting:		
Run Implementation		
Open Implemented Des		
	🔚 Td Console 🗋 🗩 Messages 🛛 🔀 Log 🗋 Reports 🗨 Design Runs	
Customize the selected core		

Figure 7: Customize IP

- 9. In the Core Options tab of the Customize IP window, set these options (see Figure 8):
 - Set GT Refclk (MHz) to 156.25
 - Set **Dataflow Mode** to **TX-only Simplex** or **RX-only Simplex**, depending on the platform being configured
 - Set Back Channel to Timer or Sideband depending on the desired testing mode.
 - Select the Vivado Lab Tools option.

ocumentation 🚞 IP Location 🇔 Switch to Defa	ts
Show disabled ports	Component Name aurora_8b10b_0
·	Core Options GT Selections Shared Logic
	Physical Layer
	Lane Width (Bytes) 2
	Line Rate (Gbps) 3.125 [0.56.6]
	GT Refdk (MHz) 156.250 🔻
₩ ⊕GT0_DRP_IF	Link Layer
-wam_cc	Dataflow Mode TX-only Simplex 👻
-tx_system_reset tx_hard_err- -gt_reset tx_channel_up -	Interface Framing 👻
<pre>-doopback[2:0] tx_lane_up[0:0]drpclk in txp[0:0] -</pre>	Flow Control None 👻
-power_down tm[0:0] -	Back Channel Timer 💌
-gt0_qplicet_in tx_resetdone_out - -gt_qplick_quad1_in gt0_qpliceset_out -	Scrambler/Descrambler
-gt_qpinercik_quad1_in tx_out_cik - -init_clk_in sys_reset_out -	Error Detection
—pll_not_locked —user_clk —sync_clk	CRC CRC
_gt_refclk1	Debug and Control
	Vivado Lab Tools
	C Additional transceiver control and status ports

Figure 8: Aurora 8B/10B 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 Aurora 8B10B (10.2)		×
💕 Documentation 늡 IP Location 🗔 Switch to Default	s	
Show disabled ports	Component Name aurora_8b10b_0 Core Options GT Selections Shared Logic Lanes 1 v	0
	Lane Assignment Note: Lane number selection is for enabling the lane only not for assigning a number to the lane.	•
duSer_DATA_S_AXI_TX ∥deTu_DRP_IF −do_cc −wam_cc −tu_system_reset to_hard_err−	GTXQ2 K V V GTXQ2 K V V K V V K V V	* * *
_gt_reset tr_channel_up - =loopback[2:0] tr_lane_up(0:0] = -drpck_in trp(0:0] = -prover_down trn(0:0) = -gr0.gplllock_in tr_lock =	GTXQ1 K + K GTXQ0 K + K K + K	* * *
-gr0_gpllmfcHdost_in tx_resetdone_out -gr_gplldk_gudd3_in gr0_gplreset_out -gr_gpllefcH_gudd3_in tx_ou_clk = -init_clk_in sys_reset_out -gl_not_locked -urer_clk	GT Refdk1 GTXQ2 V GT Refdk2 None V	
-syn_cik -g_refdk1		
K F V		Cancel

Figure 9: Aurora 8B/10B 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_8b10b_0_exdes.xdc) and select Open file (Figure 11).



Figure 11: Open Constraints File

5. Locate these two LOC property statements in the constraints file:

set_property LOC G7 [get_ports GTXQ2_N]
set_property LOC G8 [get_ports GTXQ2_P]

 Referring to Table 1, replace the two LOC property statements with the following nine statements for the respective platform (see Figure 12):

Transmit Platform:

set_property LOC J7 [get_ports GTXQ2_N] set_property LOC J8 [get_ports GTXQ2_P] set_property PACKAGE_PIN AD11 [get_ports INIT_CLK_N] set_property PACKAGE_PIN AD12 [get_ports INIT_CLK_P] set_property LOC AG5 [get_ports RESET] set_property LOC AC6 [get_ports GT_RESET_IN] set_property LOC AA8 [get_ports TX_CHANNEL_UP] set_property LOC AB8 [get_ports TX_LANE_UP] set_property LOC B17 [get_ports TX_HARD_ERR]

Receive Platform:

set_property LOC J7 [get_ports GTXQ2_N] set_property LOC J8 [get_ports GTXQ2_P] set_property PACKAGE_PIN AD11 [get_ports INIT_CLK_N] set_property PACKAGE_PIN AD12 [get_ports INIT_CLK_P] set_property LOC AG5 [get_ports RESET] set_property LOC AC6 [get_ports GT_RESET_IN] set_property LOC AA8 [get_ports RX_CHANNEL_UP] set_property LOC AB8 [get_ports RX_LANE_UP] set_property LOC B17 [get_ports RX_HARD_ERR]



Figure 12: Simplex LOC Constraints

Table 1: Aurora 8B/10B Simplex Constraints

Pin Name	Loc Value
INIT_CLK_N	AD11
INIT_CLK_P	AD12
RESET	AG5
GT_RESET_IN	AC6
TX_CHANNEL_UP/RX_CHANNEL_UP	AA8
TX_LANE_UP/RX_LANE_UP	AB8
GTXQ2_N	J7
GTXQ2_P	J8
TX_HARD_ERR/RX_HARD_ERR	B17
TX_ALIGNED/RX_ALIGNED ⁽¹⁾	C12
TX_VERIFY/RX_VERIFY ⁽¹⁾	B12
TX_RESET/RX_RESET ⁽¹⁾	C11
Notes:	

1. These pins are for back channel testing with sideband mode only.

7. If **Sideband** was selected for **Back Channel** in step 9, page 6, add these three lines to the constraints file for the respective platform:

Transmit platform:

set_property LOC C12 [get_ports TX_ALIGNED]
set_property LOC B12 [get_ports TX_VERIFY]
set_property LOC C11 [get_ports TX_RESET]

Receive platform:

set_property LOC C12 [get_ports RX_ALIGNED]
set_property LOC B12 [get_ports RX_VERIFY]
set_property LOC C11 [get_ports RX_RESET]

- 8. Locate the init_clk_i setting in the constraints file.
- 9. Change the comment from 50 MHz to 200 MHz.
- 10. Change the period from 20.000 to 5.000 (see Figure 13).





11. This example contains unconstrained pins. To permit bitsream file generation, add this line to the end of the constraints file (Figure 14):

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

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



Figure 14: Aurora 8B/10B Simplex Unconstrained Pins Property

- 12. Right-click within the constraints file editor window and select **Save File**. Close the constraints file editor window.
- 13. Select Generate Bitstream from the Flow Navigator panel.
- 14. Click **Yes** to launch Synthesis and Implementation and proceed with bitstream file generation.
- 15. 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
HardwareSetting up the Simplex Example DesignThis example illustrates a single-lane Aurora 8B/10B 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 15.

www.xilinx.com



Figure 15: KC705 Board Features

In these instructions, numbers in parentheses correspond to callout numbers in Figure 15. 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 (10) of both boards.
- If sideband mode is selected, connect a Samtec HPC-to-HPC cable between the HPC connectors (8) of both boards.
- Set the power switch (9) of both boards to the ON position.

The completed setup should resemble that shown in Figure 16. Figure 17 shows the completed setup with sideband mode.

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



Figure 16: Aurora 8B/10B Simplex Setup





Figure 17: Aurora 8B/10B Simplex Setup with Sideband Mode

Setting Up the Simplex Example Hardware Session

For back channel testing in sideband mode, skip to Programming the Devices for Back Channel Testing in Sideband Mode, page 25

Programming the Devices for Back Channel with Timer Selection

 On completion of bitstream generation, select Flow > Open Hardware Manager (Figure 18).





2. At the top of the Hardware Manager panel (see Figure 19), click **Open a new hardware target** and Click **Next**.

www.xilinx.com



Figure 19: Open a New Hardware Target

3. Select Local server and click Next (Figure 20).

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].

www.xilinx.com

1	Open New Hardware Target
Hardware Se	erver Settings
Select loca target is a	al or remote hardware server, then configure the host name and port settings. Use Local server if the 👘 💦 🛝
Connect to:	Local server (target is on local machine)
	<u>Back</u> <u>Next</u> > <u>Einish</u> Cancel

Figure 20: Hardware Server Settings

4. On the Select Hardware Target page, set the **JTAG Clock Frequency** for both boards to **750000 Hz** (Figure 21).

lect Hardware Target Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) frequency. If you do not see the expected devices, decrease the frequency or select a different target. rdware Targets Type Port Name JTAG Clock Frequency Xilinx,tcf Xilinx/00001176367101 Xilinx_tcf Xilinx/00001108817c01 Xilinx_tcf Xilinx/00001008817c01 Xilinx/00000 Hz 300000 Hz 300000 Hz Xilinx_tcf Xilinx/00001008817c01 Xilinx/00000 Hz 3000000 Hz 3000000 Hz 3000000 Hz Xilinx/00000 Hz Xilinx/0000 Hz <th></th> <th></th> <th>0</th> <th>pen New Hardware Ta</th> <th>irget</th> <th></th> <th></th> <th></th>			0	pen New Hardware Ta	irget			
Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) frequency. If you do not see the expected devices, decrease the frequency or select a different target. Type Port Name JTAG Clock Frequency XIIInx_tcf XIIInx/00001176367101 750000 Hz 3000000 Hz 3000000 Hz 3000000 Hz 12000000 Hz 1200000 Hz 100 Later Hz H	elect Hardw	are Tar	jet .		-			
you do not see the expected devices, decrease the frequency or select a different target.	Select a ha	ardware	- target from the list of avail:	able targets, then set the ap	opropriate ITAG	clock (TCK) f	reauencv. If	
rdware Targets Type Port Name JTAC Clock Frequency TS0000 Hz TS0000 Hz TS0000 Hz TS0000 Hz TS00000 Hz TS0000 H	you do not	t see the	expected devices, decreas	se the frequency or select a	different target.		. ,	
rdware Targets Type Port Name ITAG Clock Frequency xilinx_tcf Xilinx/0000176367101 750000 Hz 1500000 Hz 1500000 Hz 6000000 Hz 12000000 Hz 1200000 Hz 120000 Hz 1200 Hz 1200 Hz 120000 Hz 1200 Hz 12000 Hz 1200 Hz 1200 Hz 1200 Hz								
Type Port Name LTAG Clock Frequency xilinx_tcf Xilinx/00001176367101 75.0000 Hz 75.0000 Hz xilinx_tcf Xilinx/00001008817c01 75.0000 Hz 3000000 Hz 3000000 Hz 3000000 Hz 3000000 Hz 12000000 Hz rdware Devices (for unknown devices, specify the Instruction Register (IR) length) Name ID Code IR Length xc7k325t_0 43651093 6 6 6	irdware Tar	gets						
XIIInx_ttr XIIInx/000011/636/101 ZG000 Hz 750000 Hz 1500000 Hz 3000000 Hz 3000000 Hz 12000000 Hz 12000000 Hz	Туре	Port	Name	JTAG Clock Frequency				
xilinx_tti xilinx/00001008817401 1500000 Hz 3000000 Hz 6000000 Hz 12000000 Hz 1200000 Hz	xilinx_tcf		Xilinx/00001176367101	750000 Hz 💌				
rdware Server: xhddplab-10:3121	xilinx_ter		XIIInx/0000100881/c01	750000 Hz				
rdware server: xhddplab-10:3121				1500000 Hz				
rdware server: xhddplab-10:3121				3000000 HZ				
rdware Devices (for unknown devices, specify the Instruction Register (IR) length) Name ID Code IR Length xc7k325t_0 43651093 6				12000000 HZ				
rdware Devices (for unknown devices, specify the Instruction Register (IR) length) Name ID Code IR Length xc7k325t_0 43651093 6				1200000 Hz	1			
rdware Devices (for unknown devices, specify the Instruction Register (IR) length) Name ID Code IR Length xc7k325t_0 43651093 6								
rdware Devices (for unknown devices, specify the Instruction Register (IR) length) Name ID Code IR Length xc7k325t_0 43651093 6								
rdware server: xhddplab-10:3121								
Name ID Code IR Length xc7k325t_0 43651093 6 rdware server: xhddplab-10:3121								
xc7k325t_0 43651093 6 rdware server: xhddplab-10:3121	rdware Dev	rices (for	unknown devices, specify	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	ardware Dev Name	rices (for	unknown devices, specify Tode IR Length	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	ardware Dev Name • xc7k325t_	/ices (for ID (0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	ardware Dev Name • xc7k325t_	rices (for ID (0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	ndware Dev Name ≥xc7k325t_	ices (for ID (0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	rdware Dev Name ≻xc7k325t_	vices (for ID (0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	rdware Dev Name > xc7k325t_	ices (for ID (0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	rdware Dev Name xc7k325t_	vices (for ID (0 4365	unknown devices, specify Dode R Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	rdware Dev Name • xc7k325t_	vices (for D (0 0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	rdware Dev Name xc7k325t_	ices (for 10 0 0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
rdware server: xhddplab-10:3121	Name Name xc7k325t	ices (for 10 0 0 4365	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
ruware server. xnouplao-10.3121	ndware Dev Name ≻xc7k325t_	vices (for 0 4365	unknown devices, specify ode IR Length 1093 6	the Instruction Register (IR)	length)			
	ndware Dev Name > xc7k325t_	ices (for	unknown devices, specify Tode IR Length 1093 6	the Instruction Register (IR)	length)			
	ardware Dev Name > xc7k325t_	ver: xhd	unknown devices, specify Tode IR Length 1093 6 dplab-10:3121	the Instruction Register (IR)	length)			
<u>A Back</u> <u>Next</u> Einish Can	urdware Dew Name > xc7k325t_ urdware sen	ver: xhd	dplab-10:3121	the Instruction Register (IR)	length)			

Figure 21: 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(3) (Active).

 In the Hardware Device Properties panel, set Programming file to the bitstream file name for the receive platform and set Probes file to the appropriate .ltx probes file name (Figure 22).



Figure 22: Hardware Device Properties

8. Right-click the device in the Hardware list and select **Program Device...** (Figure 23). Ensure that the bitstream file path and name are correct and click **OK**.

2		Vivado 2014	4.1 (o	on xl	ndl-xhdc	omp1787)		
<u>File Edit Flow Tools Window Layout V</u>	jew <u>H</u> elp								
🏄 📸 🕼 🕼 🗎 🐂 🗙 😽 📴 Defa	ult Layout 🛛 👻 🔌 🔌 😣								
Hardware Manager - localhost/xilinx_tcf/Xilin	x/00001176367101 (xhddplab-10)								
Hardware	_ □ 🗠 × 🛛 Debug Probes	_ 🗆 🖻 ×		ILA -	hw_ila_1	× 🔊 VIO -	hw_vio_1	×	
🔍 🛣 🖨 🛃 🕨 🕪 🔳	🔍 🔀 🖨 🛃			VIC	Probes				
Name	Status			0	Name	Value	Activity	Direction V	10
P- localhost (xhddplab-10) (2)	Connected		-	3					
0-@ xc7k325t 0 (3) (active)	u open	1		(
- 4 XADC (System Monitor)	G Hardware Device Properties Ctri+E			×					
- 2 hw_ila_1 (ILA)	Program Device R								
willinx tcf/Xilinx/00001008817c01	Run Trigger			ĽЦ					
	M Run Trigger Immediate								
	Enable Auto Politriggor								
Hardware Device Properties	Disable Auto Re-trigger								Drag and drap
$\leftarrow \rightarrow \bigotimes \land$	Open Hardware Dackboard	loto							VIO probes from
xc7k325t_0	Refresh Device	vindow,							Debug Probes
	Add Configuration Memory Device	and/or							window.
	Export to Spreadsheet	uevice.							
		1							
014_example.runs/impl_1/aurora_8b10b_RX_10032014_exdes.bit									
014_example.runs/impl_1/debug_nets.ltx									
4									
General Properties									

Figure 23: Program Device

9. When programming completes, right-click the programmed target device in the Hardware list and select **Close Target** (Figure 24).



Figure 24: Close Target

10. Right-click the second target platform in the Hardware list and select **Open Target** (Figure 25).

<u>}</u>		Vivado 2014	1.1 (on xhdl-xhdcomp1787)
<u>File Edit Flow Tools Window Layout Y</u>	/iew Help		
🯄 🚵 🖾 💷 💼 🐂 🗙 😽 📴 Defi	ault Layout 🛛 👻 🕷 🦎	E	
Hardware Manager - localhost/xilinx_tcf/Xilin	x/00001176367101 (xhddplab-	10)	
Open closed is open. Open closed	I target Open recent target Open	n a new hardware target	
Hardware	_ 🗆 🖻 ×	Debug Probes 💷 🗆 🛃 🗡]
역 🔀 🖨 💽 🕨 🕨 🔳		🔍 🛣 🌞 🛃	
Name	Status		
P- localhost (xhddplab-10) (2)	Connected		
- 30 XIIIIX_(CI/XIIIIX/00001178387101	(0) Closed		
	Hardware Target Properties	s Ctrl+E	
	Set as Current Target		
	🗭 Refresh Target		
	Close Target		
	Open Target 🛛 💦		
Hardware Target Properties	Export to Spreadsheet		
← → ¹ / ₁			
localhost/xilinx_tcf/Xilinx/00001008817c01			
Name: localhost/xilinx	_tcf/Xilinx/0000100881/c01		
Type: xilinx_tcf			
□ Is target connection open			
Number of devices: 0			
General Properties			

Figure 25: 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 26).

		Vivado 2014.1 (on xhdl-xhdcomp1787)
<u>F</u> ile <u>E</u> dit F <u>l</u> ow <u>T</u> ools <u>W</u> ind	ow Layout View Help	
🧦 📸 🕼 🖉 🗎 🖌 💈	🐒 🔯 🔚 Default Layout 🛛 🔻 🔌 🌾	Q
Hardware Manager - localhos	t/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-	10)
Hardware	_ 🗆 🖻 ×	Debug Probes _ 🗆 🗠 × 🛛 📾 ILA - hw_ila_1 × 📾 VIO - hw_vio_1 ×
으୍ 🔀 🖨 🛃 🕨 🕨 🔳		🔍 🛣 🚔 🛃 🛛 🔀 VIO Probes
Name Name Ilocalhost (xhddplab-10)	Status Connected Status Status Connected Status Status Connected Status Status Connected Status	
Tcl Console		
INFO: [Labtools 27-3 program_Mw.devices; Frefersh.wd.device [INFO: [Labtools 27-1 INFO: [Lab	22] Done pin status: HIGH Time (5): cpu = 00:00:03 ; elapsed = 00: index [get_www.devices] 0] 432] Device xc7k325t (JTAG device index 828] Device xc7k325t (JTAG device index e e e e e e e e contact (2) Contact (2) Contact (2) Contact (2) Contact (2) Contact (2) Contact (2) Contact (2) Contact (2) Conta	01:41 . Memory (MB): peak = 4797.527 ; gain = 0.000 = 0) is programmed with a design that has 1 ILA core(s) in it. = 0) is programmed with a design that has 1 VIO core(s) in it.
🔜 I Ci Console 💭 Messa	ges 🚬 🗞 Serial I/O Links 🗋 🖬 Serial I/O Scans	
Refresh device		

Figure 26: Refresh Device

14. Observe that the CHANNEL_UP and LANE_UP LEDs are lit on both platform boards (see Table 2).

Pin Name	LOC Value	Switches/LEDs
RESET	AG5	SW6
GT_RESET_IN	AC6	SW3
LANE_UP	AB8	GPIO_LED_0
CHANNEL_UP	AA8	GPIO_LED_1

Table 2: Push Button Switches and LED Locations

Executing the Design

1. Right-click the device in the Hardware list and select Run Trigger (Figure 27).

A	Vivado 2014	1.1 (on xhdl-xhdcomp178	7)		
<u>Eile Edit Flow Tools Window Layout View H</u> elp					Q+ Search commands
🥕 😂 📾 💵 🗎 🛍 🗙 🐝 🎯 🖭 Default Layout 🛛 👻 🔌 🆎	٩				
Hardware Manager - localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-	10)				×
Hardware _ 🗆 🗷 🗵	Debug Probes 💷 🗗 🗶	👒 ILA - hw_ila_1 🗙 👒 VIO	- hw_vio_1 🗙 📟 hw_ila_	data_1.wcfg ×	
🔍 🛣 🖨 🛃 🕨 🍽 📕	🔍 🛣 🚔 🛃	→]			1,024
Name Status	🌳 🙀 hw_ila_1 🔄	Name Name	Value		1500
Incalhost (chddplab-10) (2) Connected Image: State of the state of t		With the second secon	b39 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
aurora_8b10b_0_example.runs/impl_1/aurora_8b10b_0_exdes.bit	-le <const0>_28 -le <const0>_29</const0></const0>	16 <const0>_15</const0>	0		
aurora_8b10b_0_example.runs/impl_1/debug_nets.tx	-te <const0>_3</const0>		0		
	- < const0>_30				.
General Properties		4			•
Tcl Console					_ 🗆 🖻 ×
Fun_hw_ila_hw_ila_1 INF0: [Labtools 27-1956] The ILA core 'hw_ila_1' trigger was wait_on_hw_ila_thw_ila_1 display_hw_ila_data [upload_hw_ila_data hw_ila_1] display_hw_ila_data [upload_hw_ila_data hw_ila_1] fixer of the ILA core 'hw_ila_1' triggered at { /// Prove a Tcl command here Tcl Console @ Messages % Serial I/O Links @ Serial I/O Scans	armed at 2014-Mar-12 17:05:54	4			4
Run trigger					

Figure 27: Run Trigger

- 2. In the waveform window that appears, observe a High state on the <code>lane_up</code> and <code>channel_up</code> signals.
- 3. Control-click to select these signals in the Debug Probes list under hw_vio_1:
 - tx_channel_up_r
 - lane_up_i_i
 - gtreset_vio_i
 - sysreset_vio_i
- 4. Right-click a highlighted signal and select Add Probes to VIO Window (Figure 28).

🔥 Vivado 201	14.1 (on xhdl-xhdcomp1787)	
Eile Edit Flow Tools Window Layout View Help	Qv Search command:	15
🟄 📸 💷 🖶 📉 🐝 🥵 😬 Default Layout 💿 🗶 🗞 🎉		
Hardware Manager - localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10)		×
Hardware _ C × Debug Probes _ C ×	< 🕼 ILA - hw_ila_1 × 📾 VIO - hw_vio_1 × 📓 hw_ila_data_1.wcfg ×	
	Z VIO Probes	☆ □×
Name Status	🔺 😂 🔍 Name Value Activity Direction VIO	
Connected Connected Const0>_36	Giffeset vio.i [B] 0 * Output hwwwo_l	
	🚔 – 🖕 lane_up_i_i [B] 1 Input hw_vio_1	
	x w tx_channel_up_r [B] 1 Input hw_vio_1	
- VADC (System Monitor) - Verse the value of the second s		
willia_1 (ILA)		
-le <const0>_6</const0>		
Debug Probe Properties < const0 >_7		
lane_up_i_i_1		
Name: sysreset_vio_i		
Source: NETLIST		
Probe type: vI0_OUTPUT		
Width: 1	-	
P-₩ hw_vio_1		
-lo lane_up_i_i		
o-la chipscope1.i_vio/prol		
General Properties	2bug Probe Properties Ctrl+E	
Tel Cancelo	un Trigger	
PRUT	in Trigger Immediate	
INFO: [Labtools 27-1964] The ILA core 'hw_ila_1' trigger was armed at 2014-Mar-	op ingger veble Auto Re-trigger	
wait_on_hw_ila_hw_ila_1	sable Auto Re-trigger	_
And Antipication in the second	in Probes to Waveform	_
891 (D)	nen Hardware Dashboard	v
Add	td Probes to VIO Window	
Type a Tc1 command here		
🔤 TEL CURSUR 💭 messages 🔊 seriai I/O Links 🔛 seriai I/O scans		
Add selected probes to VIO window		

Figure 28: Add Probes to VIO Window

- 5. Toggle the reset signals by clicking the value field for each signal. Enter 1 or 0 and click OK.
- $\label{eq:linear} 6. \ \mbox{The } \mbox{tx_channel_up_r} \ \mbox{and } \ \mbox{lane_up_i_i} \ \mbox{signals should go Low, then return High after } each \ \mbox{reset signal is toggled.}$

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 29.

🔥 Vivado 2014	4.1 (on xhdl-xhdcomp1787)	
Eile Edit Flow Tools Window Layout Yiew Help		Q - Search commands
🟄 📸 💷 🗎 🐘 🗙 😽 📴 Default Layout 💿 🗶 🔖 🔖 🔍		
Hardware Manager - localhost/xilinx_tcf/Xilinx/00001008817c01 (xhddplab-10)		×
Hardware _ Debug Probes _ D C ×	🔊 ILA - hw_ila_1 🗙 🔊 VIO - hw_vio_1 🗙 🖀 hw_ila_data_1.wcfg 🗙	
	20	1,024
Name Status P Iocalhost (hddplab-10) (2) Connected Connected Program Device Program Device Program Device Stop Trigger Connected Disable Auto Re-trigger Connected Stop Trigger Connected Stop Trigger Connected Constol - 41 Constol - 41 Constol - 41 Constol - 41 Constol - 40 Constol - 40 Constol - 40 Constol - 41 Constol - 41 Constol - 41 Constol - 40 Constol - 40 Constol - 40 Constol - 40 </td <td>Name Value s Name Value s s Name Value s s s Name Value s</td> <td></td>	Name Value s Name Value s s Name Value s s s Name Value s	
sysreset_vio_i		
General Properties		
Tcl Console		- 🗆 🖻 ×
Frun_Jwcila_hwclla_1 INF0: [Labtools 27-1964] The ILA core 'hw_ila_1' trigger was armed at 2014-Mar-12 17:28:3 wait_on_Jwcila_thw_ila_1 display_mw_ila_data [upload_hw_ila_data hw_ila_1] display_mw_ila_data [upload_hw_ila_data hw_ila_1] INF0: [Labtools 27-1966] The ILA core 'hw_ila_1' triggered at 2014-Mar-12 17:28:38 // /// June Arctice Armonia Ar	8	۹ ۲ ۲
Run trigger		

Figure 29: 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_vio_i</code> or <code>gtreset_vio_i</code> are asserted, both <code>tx_channel_up_r</code> and <code>lane_up_i_i</code> go Low as the core (or transceiver) is in reset state. However, when both <code>sysreset_vio_i</code> and <code>gtreset_vio_i</code> are Low, the core is out of reset state and both <code>tx_channel_up_r</code> and <code>lane_up_i_i</code> are High.

Programming the Devices for Back Channel Testing in Sideband Mode

Ensure that hardware connections are established as shown in Figure 17, page 16 before proceeding with sideband testing.

1. Disconnect the JTAG cable to the transmit platform board. Optionally, the cable can be disabled as shown in Figure 30.



Figure 30: Disable USB JTAG Cable

2. In a Windows command prompt or terminal window, enter this command to change directories to the receive platform bitstream file location:

```
% cd <Rx project directory>/aurora_8b10b_0_example/
aurora_8b10b_0_example.runs/impl_1
```

 Enter these commands to invoke the Xilinx Microprocessor Debugger tool (XMD) and download the bitstream file:

```
% xmd
XMD% fpga -f aurora_8b10b_0_exdes.bit
XMD% exit
```

- 4. Disconnect the JTAG cable from the receive platform board and connect the cable to the transmit platform board. Optionally, enable the transmit platform cable and disable the receive platform cable as shown in Figure 30.
- 5. In a Windows command prompt or terminal window, enter this command to change directories to the receive platform bitstream file location:

```
% cd <Tx project directory>/aurora_8b10b_0_example/
aurora_8b10b_0_example.runs/impl_1
```

6. Enter these commands to invoke XMD and download the bitstream file:

```
% xmd
XMD% fpga -f aurora_8b10b_0_exdes.bit
XMD% exit
```

7. Observe that the CHANNEL_UP and LANE_UP LEDs are lit on both platform boards (see Table 2, page 22).

Back Channel Testing in Sideband Mode

Refer to Table 2, page 22 for location of buttons and LEDs used in these steps.

- Press RESET (SW6) on the receive platform board and observe that the CHANNEL_UP and LANE_UP LEDs on both boards are not lit. After releasing RESET, the CHANNEL_UP and LANE_UP LEDs on both boards should be lit. Similar behavior should be observed when pressing and releasing GT_RESET_IN (SW3).
- Press RESET (SW6) on the transmit platform board and observe that the LANE_UP LED remains lit while the CHANNEL_UP LED is not lit. After releasing RESET, the CHANNEL_UP and LANE_UP LEDs should be lit. Similar behavior should be observed when pressing and releasing GT_RESET_IN (SW3).

Reference Design

Table 3 shows the reference design checklist.

Table 3: 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	Verilog (VHDL indirect support through vho/netlist)		
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 8B/10B core. Following the procedure outlined in this application note, Aurora 8B/10B 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 8B/10B Product Guide (PG046)
- 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
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, STATUTORY, INCLUDING BUT LIMITED TO WARRANTIES OF IMPLIED, OR NOT 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.

www.xilinx.com