

Xilinx Answer 68134 UltraScale and UltraScale+ FPGA Gen3 Integrated Block for PCI Express - Integrated Debugging Features and Usage Guide

Important Note: This downloadable PDF of an Answer Record is provided to enhance its usability and readability. It is important to note that Answer Records are Web-based content that are frequently updated as new information becomes available. You are reminded to visit the Xilinx Technical Support Website and review (Xilinx Answer 68134) for the latest version of this Answer.

Introduction

Prior to Vivado 2016.3 release, a manual insertion of ILA core was required to probe signals and find out the LTSSM transitions during the link training process. To do an eye scan of a PCI Express link, users had to opt for a manual approach such as the reference design provided in XAPP1198. Another major issue in debugging PCI express issues in UltraScale devices was interpreting the scrambled data on a PIPE interface. All of these difficulties have been addressed in the Vivado 2016.3 release of UltraScale and UltraScale+ PCI Express cores. The core configuration now comes with the following three integrated debug options.

- Enable JTAG Debugger
- Enable In system IBERT
- Enable Descrambler of Gen3 Mode

This document describes all of these debug features in detail with screenshots to make it easier for users to understand its implementation and usage.

JTAG Debugger

The JTAG Debugger provides users with a visual representation of the Itssm state transitions during the link training, PHY reset FSM transitions and the receiver detect status on each lane of a PCI Express link. PHY reset FSM is an internal state machine that is used by the PCIe core.

Figure 1 shows the architecture of the JTAG debugger implemented in the PCIe core when this option is enabled in the core configuration GUI. The Itssm transitions, receiver detect status and PHY reset FSM status are stored in the block rams. The stored data is read through AXI JTAG Debugger via Tcl interface. A tcl script, *test_rd.tcl*, is provided with the core generation which is executed in the Vivado Tcl console. The script reads the data stored in the memory and outputs the following set of files:

- pcie_debug_info_trc.dat
- pcie_debug_ltssm_trc.dat
- pcie_debug_rst_trc.dat
- pcie_debug_static_info.dat
- rxdet.dat

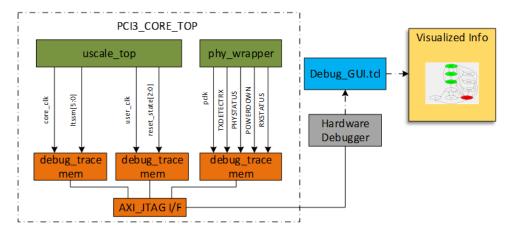
The following Tcl scripts are generated along with the generation of the PCI Express core.

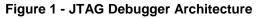
- draw_ltssm.tcl
- draw_reset.tcl



- draw_rxdet.tcl
- test_rd.tcl

These scripts take in the *.dat files and plot graphs for LTSSM transitions, PHY reset FSM transitions and receiver detect status.





The screenshots below show the step-by-step instruction for using JTAG Debugger in the PCI Express example design on a KCU105 development board.

Choose a defau	t Xilinx part or board for your p	roject. This can	be changed la	ater.			
elect: 🛞 Parts	Boards						
Filter/Preview	Boards						
Filler/Freview							
Ve <u>n</u> dor:	All	-			min		
Display Name:	All	-		Professional Contraction			
	l shash			Electron from the second			
Board Re <u>v</u> :	Latest	-					
	Reset All Filters			The second second second	ale and and	u	
		_					
earch: Q-	•	-					
Display Name		Vendor	Board Rev	Part	I/O Pin Count	File Version	Bloci RAM
ZedBoard Zyng E	valuation and Development Kit	em.avnet.com	d	xc7z020clg484-1	484	1.3	140
Artix-7 AC701 E	valuation Platform	xilinx.com	1.1	xc7a200tfbg676-2	676	1.3	365
	Evaluation Platform	xilinx.com	1.1	xc7k325tffg900-2	900	1.4	445
Kintex-7 KC705 E	KCU105 Evaluation Platform	xilinx.com	1.0	🔷 🕸 xcku040-ffva1156-2-e	1,156	1.1	600
	. I =1	xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.3	1030
	valuation Platform						1470
Kintex-UltraScale Virtex-7 VC707 E	valuation Platform	xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.8	
Kintex-UltraScale Virtex-7 VC707 E Virtex-7 VC709 E		xilinx.com xilinx.com	1.0 1.0	xc7vx690tffg1761-2 xcvu095-ffva2104-2-e	-	1.8	1728
Kintex-UltraScale Virtex-7 VC707 E Virtex-7 VC709 E Virtex-UltraScale	valuation Platform				2,104		
Kintex-UltraScale Virtex-7 VC707 E Virtex-7 VC709 E Virtex-UltraScale	valuation Platform VCU108 Evaluation Platform VCU110 Evaluation Platform	xilinx.com	1.0	xcvu095-ffva2104-2-e	2,104	1.2	1728 3780 140
Kintex-UltraScale Virtex-7 VC707 E Virtex-7 VC709 E Virtex-UltraScale Virtex-UltraScale	valuation Platform VCU108 Evaluation Platform VCU110 Evaluation Platform valuation Board	xilinx.com xilinx.com	1.0 1.0	 xcvu095-ffva2104-2-e xcvu190-flgc2104-2-e 	2,104 2,104	1.2 1.1	3780

Figure 2 – Create a Vivado project targeting KCU105 Development Board

© Copyright 2016 Xilinx

🗜 Customize IP		
UltraScale FPGA Gen3 Integrated Block for PCI Expr	ess (4.2)	4
🍘 Documentation 📋 IP Location 🧔 Switch to Defaults		
Show deabled ports m_axis_cq + m_axis_rc + pcie_7x_mqt + # +s_axis_rc pcie_fg_ft + # +pcie_cfg_mgnt pcie3_cfg_mesg_tt + # +pcie_cfg_mgnt pcie3_cfg_mesg_rt + # +pcie3_cfg_ms pcie3_cfg_mesg_rt + # +pcie3_cfg_ms pcie3_cfg_mesg_rt + # +pcie3_cfg_ms # +pcie3_cfg_ms pcie3_reg_mt # +pcie3_cfg_ms	Component Name pde3_ultrascale_0 Basic Capabilities PF0 IDs PF0 BAR Legacy/MSI Cap MSI-X Cap Power Management Extd. Capabilities Mode Advanced Device / Port Type PCI Express Endpoint device PCIe Block Location N070 Cft Quad Cft Quad Selection PCIe Block Location N070 Cft Quad Cft Quad Cft Quad 225 Number of Lanes Lane Width X8 AXI-ST Interface Width AXI-ST Interface Width 256 bit AXI-ST Interface Width 256 bit OWORD Aligned Address Aligned	_
Implies in the second secon	Reference Clock Frequency (MHz) 100 MHz Xilinx Development Board KCU105 Enable Parity PCte DRP Ports G T Channel DRP Enable RX Message INTFC V Use the dedicated PERST routing resources System reset polarity ACTIVE LOW Tandem Configuration or Partial Reconfiguration None Tandem Configuration or Partial Reconfiguration	Enable AXI-ST Frame Straddle Enable Client Tag Enable External PIPE Interface Additional Transceiver Control and Status Ports

Figure 3 – Select Gen3 link speed, x8 lane width and KCU105 board. Only Gen3 is currently supported; there is no restriction in lane width

Note: Gen1 and Gen2 link speed support will be enabled in a future release.

🖵 Customize IP						
JItraScale FPGA Gen3 Integrated Block for PCI Express (4.2)						
💕 Documentation 📄 IP Location 🧔 Switch to Defaults						
Show disabled ports	Component Name pcie3_ultrascale_0					
	PF0 IDs PF0 BAR Legacy/MSI Cap MSI-X Cap Power Management Extd. Capabilities-1 Extd. Capabilities-2 Shared Logic GT Set ngs Add. Debug Options					
m_axis_cq ⊕ ⊞ m_axis_rc ⊕ ⊞ pcie 7x mat ⊕ III	Enable In System IBERT Enable Descrambler for Gen3 mode Figure 17AG Debugger					

Figure 4 – Enable JTAG Debugger

Note: All three options can be selected in the same design. The options are selected one at a time in this document for illustration purpose only.



Project Manager - pcie3_ultrascale_0_ex	
Sources	? _ 🗆 🖻 ×
 < 🔀 🖨 📷 🔂 2 	
En Contraction Sources (5)	
weilinx_pcie3_uscale_ep (xilinx_pcie_uscale_ep.v) (3)	
pcie3_ultrascale_0_i - pcie3_ultrascale_0 (pcie3_ultrascale_0.xci)	
Description: De	
🚊 😳 debug_wrapper_U - debug_wrapper (debug_wrapper.v) (3)	
📄 🖵 jtag_axi4l_m_inst - pcie4_usp_jtag (pcie4_usp_jtag.xci) (1)	
⊡ · 😡 pcie4_usp_jtag (pcie4_usp_jtag.v) (1)	
····· 💡 inst - jtag_axi_v1_2_0_jtag_axi	
⊡- 1 TCL (4)	
Est_rd.td	
Constraints (1)	
□ · 🗁 constrs_1 (1)	
xilinx_pcie3_uscale_ep_x8g3.xdc	
Hierarchy IP Sources Libraries Compile Order	

Figure 5 – Generate the core and open the example design. The source hierarchy should show debug_wrapper and axi_jtag instantiation and list of all generated Tcl files.

bug			
Name	Driver Cell	Driver Pin	Probe Type
🗣 🦉 dbg_hub (labtools_xsdbm_v2)			
└፼ jtag_axi4l_m_inst (xilinx_jtag-axi_v1)			
🗣 🗁 Unassigned Debug Nets (347)			
 ・「並 debug_wrapper_U/AXI_araddr (32) ・「並 debug_wrapper_U/AXI_arprot (3) ・「並 debug_wrapper_U/AXI_awaddr (32) ・「並 debug_wrapper_U/AXI_awprot (3) ・「並 debug_wrapper_U/AXI_awprot (3) ・ ・ ・	FDRE	Q	
∲-∰¤ debug_wrapper_U/AXI_arprot (3)	GND	G	
💁 🖣 🕱 debug_wrapper_U/AXI_awaddr (32)	FDRE	Q	
∲-∰¤ debug_wrapper_U/AXI_awprot (3)	GND	G	
∲-乐☆ debug_wrapper_U/AXI_bresp (2)	GND	G	
∲-∰¤ debug_wrapper_U/AXI_rdata (32)	Multiple	Multiple	
	GND	G	
💁 🖟 🛱 debug_wrapper_U/AXI_wdata (32)	FDRE	Q	
💁 🖵 🛱 debug_wrapper_U/AXI_wstrb (4)	VCC	Р	
🖕 🖟 🛱 debug_wrapper_U/debug_probes_inst/ltssm_mem_raddr (9)	FDRE	Q	
🗣 🖅 🕱 debug_wrapper_U/debug_probes_inst/Itssm_mem_rdata (16)	RAMB18E2	Multiple	
∮- fræ debug_wrapper_U/debug_probes_inst/itssm_mem_waddr (9)	FDCE	Q	

Figure 6 - The debug window view after synthesis



(model) (7) + 1 + 1 = 2 + 1	and a state of the		
tirnxds1) (Z:) ▶ h ▶ gen3debu	igcheck2 ► pcie3_ultr	ascale_0_ex ♦1imports	
notepad (23)	TCL File (4)	VH File (4)	XDC File (1)
🖬 board	draw_ltssm.tcl	board_common.vh	xilinx_pcie3_uscale_ep_x8g3.xc
📽 debug_axi4I_s	draw_reset.tcl	pci_exp_expect_tasks.vh	
📝 debug_probes	draw_rxdet.tcl	sample_tests.vh	
📝 debug_wrapper	test_rd.tcl	tests.vh	
📝 ep_mem			
📝 pci_exp_usrapp_cfg			
😰 pci_exp_usrapp_com			
📝 pci_exp_usrapp_rx			
📝 pci_exp_usrapp_tx			
📝 pcie_app_uscale			
📝 pcie3_uscale_rp_core_top			
📝 pcie3_uscale_rp_top			
🛒 pio			
🛒 pio_ep			
📝 pio_ep_mem_access			
📝 pio_intr_ctrl			
📝 pio_rx_engine			
😰 pio_to_ctrl			
📝 pio_tx_engine			
📽 sys_clk_gen			
📽 sys_clk_gen_ds			
😰 xilinx_pcie_uscale_ep			
🖹 xilinx_pcie_uscale_rp			

Figure 7 – The Tcl files are located in the 'imports' folder inside the example design project directory

Hardware Manager - localhost/xilinx_tcf/Digilent/210308956781			
There are no debug cores. Program device Refresh device			
Hardware ?	×		
역 🔀 😂 📳 🕨 🕨 🖿			
Name	Status		
⊡ Iocalhost (1)	Connected		
⊡ 📓 🏕 xilinx_tcf/Digilent/210308956781 (1)	Open		
Ė 🚸 xcku040_0 (1)	Not programmed		
🔤 🐼 SysMon (System Monitor)			

Figure 8 - Open Hardware Manager and connect to KCU105 board

🔥 Program Device		×				
	Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.					
Bitstre <u>a</u> m file:	Z:/h/gen3debugcheck2/pcie3_ultrascale_0_ex/pcie3_ultrascale_0_ex.runs/impl_1/xilinx_pcie3_uscale_ep.bit	8 -				
Debug probes file:	Z:/h/gen3debugcheck2/pcie3_ultrascale_0_ex/pcie3_ultrascale_0_ex.runs/impl_1/debug_nets.ltx	8				
✓ Enable end of s	startup check					
?	Program	Cancel				

Figure 9 - Program the device with the generated bit and Itx files

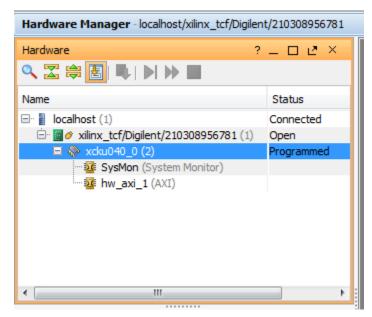


Figure 10 – After the bit file has been programmed, the Hardware window should list hw_axi_1 © Copyright 2016 Xilinx



Td (Console			
\mathbf{Z}	25/10/20	16 10:57	15,756	pio_ep.v
e	25/10/20	16 10:57	25,553	pcie_app_uscale.v
	25/10/20	16 10:57	68,212	ep_mem.v
	25/10/20	16 10:57	20,786	debug_probes.v
4	25/10/20	16 10:57	8,026	debug_axi41_s.v
	25/10/20	16 10:57	14,100	debug_wrapper.v
	25/10/20	16 10:57	11,604	draw_ltssm.tcl
U	25/10/20	16 10:57	5,251	draw_reset.tcl
	25/10/20	16 10:57	4,581	draw_rxdet.tcl
	25/10/20	16 10:57	4,594	test_rd.tcl
	25/10/20	16 10:57	5,910	<pre>xilinx_pcie3_uscale_ep_x8g3.xdc</pre>
	25/10/20	16 10:57	18,426	pci_exp_usrapp_cfg.v
	•			
	source ./tes	t_rd.tcl		
	🛄 Tcl Console	🔎 Messages	J	

Figure 11 - Source test_rd.tcl in Vivado Tcl Console, located in 'imports' folder

m (\\xirnxds1) (Z:) ▶ h ▶ gen3deb	ugcheck2 > pcie3_ultrascale_0	ex imports		
		5 1			
_					
•	DAT File (5)	notepad (23)	TCL File (4)	VH File (4)	XDC File (1)
	pcie_debug_info_trc	🖬 board	draw_ltssm.tcl	board_common.vh	xilinx_pcie3_uscale_ep_x8g3.xdc
	pcie_debug_ltssm_trc	📝 debug_axi4l_s	draw_reset.tcl	pci_exp_expect_tasks.vh	
	pcie_debug_rst_trc	📓 debug_probes	draw_rxdet.tcl	sample_tests.vh	
	🗖 pcie_debug_static_info	📓 debug_wrapper	test_rd.tcl	tests.vh	
	🗖 rxdet	😰 ep_mem			
		😰 pci_exp_usrapp_cfg			
		pci_exp_usrapp_com			
		📝 pci_exp_usrapp_rx			
		🖉 pci_exp_usrapp_tx			
		🖉 pcie_app_uscale			
		pcie3_uscale_rp_core_top			
η.		pcie3_uscale_rp_top			
		📝 pio			
		📓 pio_ep			
		igno_ep_mem_access			
		☑ pio_intr_ctrl ☑ pio_rx_engine			
		pio_rx_engine			
		pio_to_ctil			
		sys_clk_gen			
		sys_clk_gen_ds			
		xilinx_pcie_uscale_ep			
		xilinx_pcie_uscale_rp			
		P			

Figure 12 - *dat files are generated after running test_rd.tcl

© Copyright 2016 Xilinx



<pre># run_hw_axi -quiet [get_hw_axi_txns rd_txn_lite] # set tdata [get_property DATA [get_hw_axi_txns rd_txn_lite] # puts \$fh "0x\$tdata" # } # incr j # } # incr j # } # close \$fh # close \$fh # source ./draw_ltssm.tcl # package require Tcl 8.5 # package require Tk can't find package Tk while executing "package require Tk" (file "./draw_ltssm.tcl" line 9)</pre>

Figure 13 – Error if draw_ltssm.tcl, draw_reset.tcl and draw_rxdet.tcl are sourced in the Vivado Tcl Console

Note: draw_ltssm.tcl, draw_reset.tcl and draw_rxdet.tcl are separate from Vivado.

After the *data files have been generated, double click on the Tcl files in the 'imports' directory to generate the respective graphs. To generate the graph, users should make sure Tcl/Tk 8.5 package have been installed. Go to the links below to download the Tcl/Tk packages for the platform that is being used.

- http://www.activestate.com/activetcl/downloads
- <u>https://www.tcl.tk/software/tcltk/download.html</u>

DOWNLOAD TCL: OTHER PLATFORMS AND VERSIONS

Version	Windows (x86)	Windows (64-bit, x64)	Mac OS X (10.5+, x86_64/x86)	Linux (x86)	Linux (x86_64)
8.6.4.1	Windows Installer (EXE)	Windows Installer (EXE)	Mac Disk Image (DMG)	AS Package	AS Package
8.5.18.0	Windows Installer (EXE)	Windows Installer (EXE)	Mac Disk Image (DMG)	AS Package	AS Package
					÷.

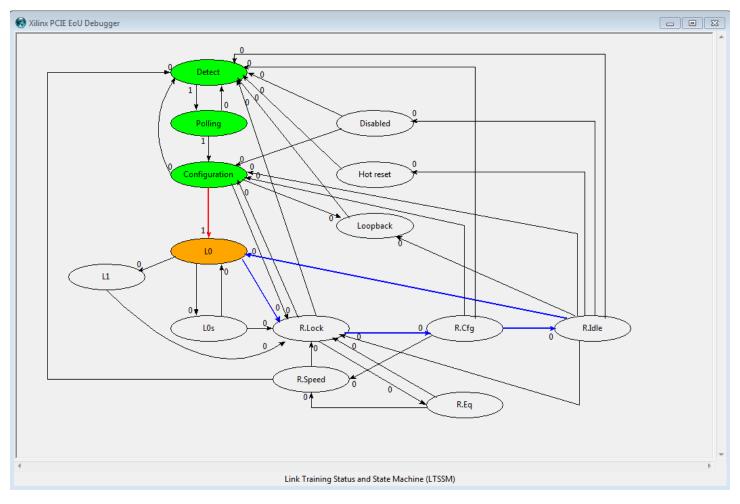


T File (5)	notepad (23)	TCL File (4)	VH File (4)	XDC File (1)
pcie_debug_info_trc	🖬 board	draw_ltssm.tcl	board_common.vh	📄 xilinx_pcie3_uscale_er
pcie_debug_ltssm_trc	🖬 debug_axi4I_s	draw_reset.tcl	pci_exp_expect_tasks.vh	
<pre> pcie_debug_rst_trc </pre>	🖬 debug_probes	draw_rxdet.tcl	sample_tests.vh	
pcie_debug_static_info	🖬 debug_wrapper	test_rd.tcl	tests.vh	
👔 rxdet	📝 ep_mem			
	📝 pci_exp_usrapp_cfg			
	📓 pci_exp_usrapp_com			
	🖻 nci ovn usrann n			

Figure 14 – Double click on draw_ltssm.tcl to generate LTSSM graph

Note: Use the following commands in Linux

- wish draw_ltssm.tcl &
- wish draw_reset.tcl &
- wish draw_rxdet.tcl &



© Copyright 2016 Xilinx



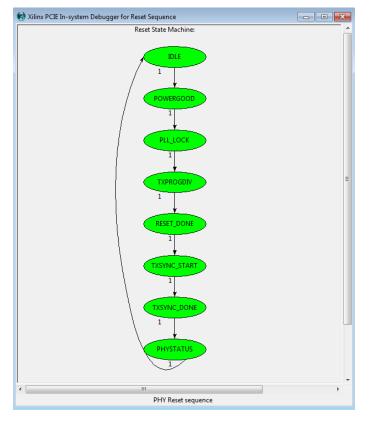
Figure 15 – Generated LTSSM graph

Note on generated LTSSM graph:

- The orange color indicates the last Itssm state of the capture window of the Itssm signal.
- The red arrow indicates the last transition of Itssm state in the capture window.
- The green color indicates the states that Itssm transitioned to during the capture window.
- The number on the arrow between the Itssm states shows the number of times the transition took place between the two Itssm states in the direction pointed to by the arrow.
- The blue color in the arrows in Figure 15 can be ignored. It should be black; it will be corrected in a future release of the IP.

DAT File (5)	notepad (23)		VH File (4)	XDC File (1) —
pcie_debug_info_trc	📝 board	draw_ltssm.tcl	board_common.vh	xilinx_pcie3_u
pcie_debug_ltssm_trc	📓 debug_axi4l_s	draw_reset.tcl	pci_exp_expect_tasks.vh	
pcie_debug_rst_trc	🖬 debug_probes	draw_rxdet.tcl	sample_tests.vh	
🔯 pcie_debug_static_info	📝 debug_wrapper	test_rd.tcl	tests.vh	
🖸 rxdet	📝 ep_mem			
	📓 pci_exp_usrapp_cfg			
	📓 pci_exp_usrapp_com			
	🗹 nci evn usrann rv			

Figure 16 – Double click on draw_reset.tcl to generate PHY reset FSM graph



© Copyright 2016 Xilinx





DAT File (5)	notepad (23)		VH File (4)	XDC File (1)
pcie_debug_info_trc	📓 board	draw_ltssm.tcl	board_common.vh	xilinx_pcie3_uscale_ep_x8g
🙍 pcie_debug_ltssm_trc	📝 debug_axi4l_s	draw_reset.tcl	pci_exp_expect_tasks.vh	
🙍 pcie_debug_rst_trc	📝 debug_probes	draw_rxdet.tcl	sample_tests.vh	
🙍 pcie_debug_static_info	📝 debug_wrapper	test_rd.tcl	tests.vh	
🗖 rxdet	📝 ep_mem			
	📝 pci_exp_usrapp_cfg			
	🖬 nci exn usrann com			

Figure 18 - Double Click on draw_rxdet.tcl to check the status of receiver detect on each lane

😪 Xilinx PCIE In-system Debugger for Rec 👝 💷 🛋
Max link width : 8
Negotiated Link width : 8
Lane 0 🗢
Lane 1 🔍
Lane 2 🔹
Lane 3 🔍
Lane 4 🔹
Lane 5 🗢
Lane 6 💿
Lunco
Lane 7 🗢
· · · · · · · · · · · · · · · · · · ·
< III ► TX Receiver Detect
TA Receiver Detect

Figure 19 – Receiver successfully detected on all 8 lanes



In-System IBERT

In-System IBERT is a powerful feature, integrated into the Vivado 2016.3 core. This allows users to capture an eye diagram in real-time without any additional effort. An in-system eye scan is valuable in PCI express applications because placing a PCI Express link in loopback is not practical and generally not possible.

The screenshots below show the step-by-step instruction for using In-System IBERT in PCI Express Example design on a KCU105 development board.

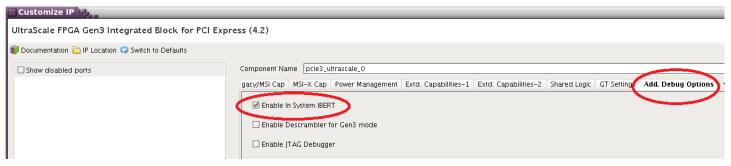


Figure 20 - Enable In-System IBERT

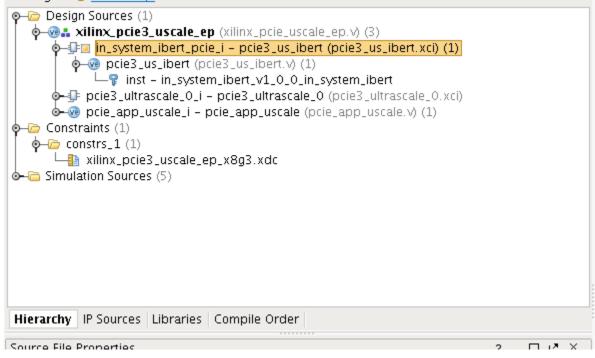


Figure 21 – IBERT instantiation in the generated example design

XILINX.

Synthesized Design - synth_1 xcku040-ffva	1156–2–e (active)	
Netlist	? _ 🗆 🖻 ×	🔄 Schematio
🔀 🔄 🖪		-
🕅 xilinx_pcie3_uscale_ep		4
🔶 🫅 Nets (2244)		
🗭 🫅 Leaf Cells (15)		-
🗕 🧧 dbg_hub (dbg_hub_CV)		Q+
🖗 虄 in_system_ibert_pcie_i (pcie3_us_ibert)		
🖕 🦳 Nets (1106)		Q-
🖕 🍋 Leaf Cells (1)		0
inst (pcie3_us_ibert_in_system_ibert_v	1_0_0_in_system_ibert)	<u> </u>
pcie3_ultrascale_0_i (pcie3_ultrascale_0)		
🖢 📵 pcie_app_uscale_i (pcie_app_uscale)		ı. ۲
		. .
		N_₩

Figure 22 - IBERT instantiation in the synthesized design

d Console
<pre>start_gui open_hw INFO: [IP_Flow 19-234] Refreshing IP repositories INFO: [IP_Flow 19-1704] No user IP repositories specified INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2016.3/data/ip'. set_param xicom.enable_isi_pcie_fix 1 1</pre>
Type a Tcl command here
Tcl Console OMessages

Figure 23 - Before programming the device execute set_param xicom.enable_isi_pcie fix 1

Hardware Manager - localhost/xilinx_tcf/Digilen	t/210308956781	
(i) There are no serial I/O links. Auto-detect link	s <u>Create links</u>	
Hardware ?	_ 🗆 🖻 ×	
역 🔀 🖨 🖪 🚱 📭 🔛 🕨 🔳		
Name	Status	
🖃 🚪 localhost (1)	Connected	
⊡ · 📓 🖉 xilinx_tcf/Digilent/210308956781 (1)	Open	
🖻 🚸 xcku040_0 (2)	Programmed	
🖃 🦉 In-System IBERT (IBERT)		
🖃 🖏 Quad_224 (4)		
MGT_X0Y0		
MGT_X0Y1		
MGT_X0Y2		
MGT_X0Y3		
⊡-1\$ Quad_225 (4)		
MGT_X0Y4		
MGT_X0Y5		
MGT_X0Y6		
NGT_X0Y7		
· ·		
Hardware Device Properties ?	_ D & ×	
xcku040_0		

Figure 24 – In-System IBERT instantiation in Hardware Manager after programming the device

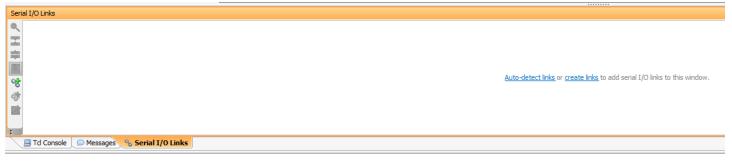


Figure 25 – 'Auto-detect links' and 'Create links' in 'Serial I/O Links' tab of the Hardware Manager

To create a new link select a TX GT and/or an RX GT, then dick the	Add button on the New Links toolbar.
X GTs	RX GTs
Search: Q,-	Search: Q-
MGT_X0Y0/TX (xcku040_0/Quad_224)	MGT_X0Y0/RX (xdku040_0/Quad_224)
MGT_X0Y1/TX (xcku040_0/Quad_224)	MGT_X0Y1/RX (xcku040_0/Quad_224)
> MGT_X0Y2/TX (xcku040_0/Quad_224)	MGT_X0Y2/RX (xcku040_0/Quad_224)
> MGT_X0Y3/TX (xcku040_0/Quad_224)	MGT_X0Y3/RX (xcku040_0/Quad_224)
> MGT_X0Y4/TX (xcku040_0/Quad_225)	MGT_X0Y4/RX (xdku040_0/Quad_225)
> MGT_X0Y5/TX (xcku040_0/Quad_225) > MGT_X0Y6/TX (xcku040_0/Quad_225)	MGT_X0Y5/RX (xcku040_0/Quad_225) MGT_X0Y6/RX (xcku040_0/Quad_225)
MGT_X0Y7/TX (xcku040_0/Quad_225)	MGT_X0YD/KX (Xcku040_0/Quad_225) MGT_X0Y7/KX (xcku040_0/Quad_225)
ew Links	
+	
	Press the 🕂 button to Add Link
<u>C</u> reate link group	
Create link group Ink Group 0	Press the 🕂 button to Add Link
<u>C</u> reate link group	

Figure 26 - Create link by clicking on 'Create links' in 'Serial I/O Links'

					•
GTs			RX GTs		
earch: Q			Search:	Qv	
w Links					
-					
		RX			
% Link (MGT_X0Y0/TX (xcku040	_0/Quad_224) MGT_X0Y2/RX (:			
S Link (MGT_X0Y0/TX (xcku040 MGT_X0Y1/TX (xcku040	_0/Quad_224) MGT_X0Y2/RX (: _0/Quad_224) MGT_X0Y0/RX (:	xcku040_0/Quad_224)		
ි Link (ඉ Link : ඉ Link :	MGT_X0Y0/TX (xcku040 MGT_X0Y1/TX (xcku040 MGT_X0Y2/TX (xcku040	_0/Quad_224) MGT_X0Y2/RX (: _0/Quad_224) MGT_X0Y0/RX (: _0/Quad_224) MGT_X0Y1/RX (:	xcku040_0/Quad_224) xcku040_0/Quad_224)		
ବ୍ତ Link (ବ୍ୟୁ Link) ବ୍ୟୁ Link 3 ବ୍ୟୁ Link 3	MGT_X0Y0/TX (xcku040) MGT_X0Y1/TX (xcku040) MGT_X0Y2/TX (xcku040) MGT_X0Y2/TX (xcku040)	_0/Quad_224) MGT_X0Y2/RX (: _0/Quad_224) MGT_X0Y0/RX (: _0/Quad_224) MGT_X0Y1/RX (: _0/Quad_224) MGT_X0Y3/RX (:	xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_224)		
S Link C S Link 2 S Link 2 S Link 3 S Link 3	MGT_X0Y0/TX (xdau040 MGT_X0Y1/TX (xdau040 MGT_X0Y2/TX (xdau040 MGT_X0Y2/TX (xdau040 MGT_X0Y3/TX (xdau040 MGT_X0Y4/TX (xdau040)	_0/Quad_224) MGT_X0Y2/RX (: _0/Quad_224) MGT_X0Y0/RX (: _0/Quad_224) MGT_X0Y1/RX (: _0/Quad_224) MGT_X0Y1/RX (: _0/Quad_225) MGT_X0Y4/RX (:	xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_225)		
S Link C S Link 2 S Link 2 S Link 3 S Link 4 S Link 4	MGT_X0Y0/TX (xdxu040 MGT_X0Y1/TX (xdxu040 MGT_X0Y2/TX (xdxu040 MGT_X0Y3/TX (xdxu040 MGT_X0Y3/TX (xdxu040 MGT_X0Y4/TX (xdxu040 MGT_X0Y5/TX (xdxu040)	0/Quad_224) MGT_X0Y2/RX (0/Quad_224) MGT_X0Y0/RX (0/Quad_224) MGT_X0Y1/RX (0/Quad_224) MGT_X0Y1/RX (0/Quad_225) MGT_X0Y3/RX (0/Quad_225) MGT_X0Y5/RX (xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_225) xcku040_0/Quad_225)		
Solution Control Contr	MGT_X0Y0/TX (xcku040 MGT_X0Y1/TX (xcku040 MGT_X0Y2/TX (xcku040 MGT_X0Y3/TX (xcku040 MGT_X0Y4/TX (xcku040 MGT_X0Y4/TX (xcku040 MGT_X0Y6/TX (xcku040	_0/Quad_224) MGT_X0Y2/RX (: _0/Quad_224) MGT_X0Y0/RX (: _0/Quad_224) MGT_X0Y1/RX (: _0/Quad_224) MGT_X0Y1/RX (: _0/Quad_225) MGT_X0Y4/RX (:	xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_225) xcku040_0/Quad_225) xcku040_0/Quad_225)		
ی Link و کی Link و کی Link و کی Link و کی Link و کی Link و کی Link و	MGT_X0Y0/TX (xcku040 MGT_X0Y1/TX (xcku040 MGT_X0Y2/TX (xcku040 MGT_X0Y3/TX (xcku040 MGT_X0Y4/TX (xcku040 MGT_X0Y5/TX (xcku040 MGT_X0Y6/TX (xcku040 MGT_X0Y7/TX (xcku040	0/Quad_224) MGT_X0Y2/RX (0/Quad_224) MGT_X0Y0/RX (0/Quad_224) MGT_X0Y1/RX (0/Quad_224) MGT_X0Y1/RX (0/Quad_225) MGT_X0Y4/RX (0/Quad_225) MGT_X0Y4/RX (0/Quad_225) MGT_X0Y6/RX (xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_224) xcku040_0/Quad_225) xcku040_0/Quad_225) xcku040_0/Quad_225)		

Figure 27 – In System IBERT x8 link

Xilinx Answer 68134 – UltraScale and UltraScale+ PCIe Integrated Debugging Features and Usage Guide 15

5	Name	тх	RX	TX Pre-Cursor		TX Post-Curso	r	TX Diff Swing		DFE Enable	d
2	: 🗁 Ungrouped Links (0)										
2	E Stink Group 0 (8)			User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	
_	S Link 0	MGT_X0Y0/TX	MGT_X0Y2/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	
	🗞 Link 1	MGT_X0Y1/TX	MGT_X0Y0/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	,
5	🗞 Link 2	MGT_X0Y2/TX	MGT_X0Y1/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	,
	🗞 Link 3	MGT_X0Y3/TX	MGT_X0Y3/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	,
ŀ,	% Link 4	MGT_X0Y4/TX	MGT_X0Y4/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	,
ł.	% Link 5	MGT_X0Y5/TX	MGT_X0Y5/RX	User Value	Ŧ	User Value	w	User Value	Ŧ	User Value	•
	- % Link 6	MGT_X0Y6/TX	MGT_X0Y6/RX	User Value	Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	,
þ	% Link 7	MGT_X0Y7/TX	MGT_X0Y7/RX	User Value	Ŧ	User Value	w	User Value	Ŧ	User Value	•
3	-										

Figure 28 – Serial I/O Links

Jngrouped Linl 👩 .ink Group 0 (8 Link 0	TX Link Prope	RX	TX Pre-Curs	or	TX Post-Curs	or	TX Diff Swin	q	DFE Enable	be
ink Group 0 (8	Link Prope							-		
· · ·		rties	Ctrl+E							
Link 0 🗛	Delete				User Value		User Value	Ŧ	ober value	
-			Delete		User Value	- *	User Value		Jser Value	
sLink 1 역	Create Lin	ks			User Value	-	User Value	Ŧ	User Value	•
🖢 Link 2 🧃	Create Lin	k Group		Ŧ	User Value	Ŧ	User Value	Ŧ	User Value	
🖢 Link 3 📃					User Value	÷	User Value	Ŧ	User Value	
🗞 Link 4 🛛 🗳	Create Sc	an		Ŧ	User Value	-	User Value	Ŧ	User Value	
🗞 Link 5 🛛 🍯	Create Sw	eep		Ŧ	User Value		User Value	Ŧ	User Value	
🗞 Link 6	Commit Dr	operties			User Value	÷	User Value	Ŧ	User Value	•
		operues		÷	User Value		User Value	Ŧ	User Value	•
¢	Refresh S	erial I/O Objects								
onsole 🔍 N	Select		•	Sca	ine					_
	Link 2	Link 2 Link 3 Link 4 Link 5 Link 5 Link 6 Link 7 Link 7 Link 7 Link 7 Link 7 Link 7 Link 8 Link 8 Link 8 Link 6 Link 8 Link 7 Link 8 Link 8 Li	Link 2 Image: Create Link Group Link 3 Image: Create Scan Link 4 Image: Create Scan Link 5 Image: Create Sweep Link 6 Commit Properties Link 7 Refresh Serial I/O Objects Select Select	Link 2 Image: Create Link Group Link 3 Create Scan Link 4 Create Sweep Link 5 Create Sweep Link 6 Commit Properties Link 7 Refresh Serial I/O Objects Select Felex	Link 2 Image: Create Link Group Link 3 Image: Create Scan Link 4 Image: Create Scan Link 5 Image: Create Sweep Link 6 Create Sweep Link 7 Refresh Serial I/O Objects Select Sca	Link 2 Image: Create Link Group User Value Link 3 Create Scan User Value Link 4 Create Scan User Value Link 5 Create Sweep User Value Link 6 Commit Properties User Value Console Refresh Serial I/O Objects Scans	Link 2 Image: Create Link Group User Value Select Scans 	Link 2 Image: Create Link Group User Value User Value Link 3 Create Scan User Value User Value Link 4 Create Scan User Value User Value Link 5 Create Sweep User Value User Value Link 6 Commit Properties User Value User Value Link 7 Refresh Serial I/O Objects Scans	Link 2 Image: Create Link Group User Value 	Link 2 Image: Create Link Group User Value User Value User Value User Value Link 3 Create Scan User Value User Value User Value User Value Link 4 Create Scan User Value User Value User Value User Value Link 5 Create Sweep Commit Properties User Value User Value User Value Link 7 Refresh Serial I/O Objects Select Select User Value User Value

Figure 29 – Scan link

🔶 Create S	can		×
Set the des selected lin		her properties to create and optionally run a scan on the	4
Link:	Link 0 (MGT_X	0Y0/TX, MGT_X0Y2/RX)	
Description:	Scan 0		8
Scan Proper	ties		
<u>S</u> can ty	pe:	2D Full Eyescan	*
<u>H</u> orizon	tal increment:	8	-
H <u>o</u> rizon	tal range:	-0.500 UI to 0.500 UI	-
<u>V</u> ertical	increment:	8	-
V <u>e</u> rtical	range:	100%	-
Dwell			
<u>B</u> EF	R: 1e-5		Ŧ
⊙ <u>T</u> im	e:	(
<mark>√</mark> <u>R</u> un sca	n		
		ОК	Cancel

Figure 30 – Default scan options

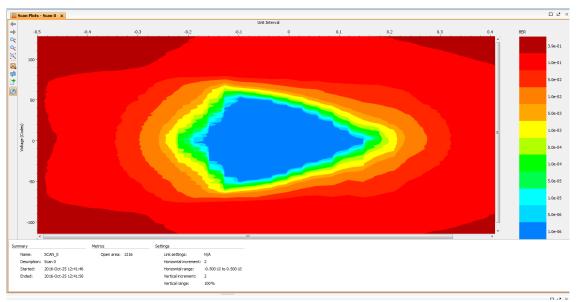


🔥 Create S	can		×				
Set the des selected lin		her properties to create and optionally run a scan on the	4				
Link:	Link 0 (MGT_X	DY0/TX, MGT_X0Y2/RX)					
Description:	Scan 0	n 0 😒					
Scan Proper	ties						
<u>S</u> can ty	pe:	2D Full Eyescan	•				
<u>H</u> orizon	tal increment:	2	-				
H <u>o</u> rizon	tal range:	-0.500 UI to 0.500 UI	•				
<u>V</u> ertical	increment:	2	*				
V <u>e</u> rtical	range:	100%	•				
Dwell							
BEF	R: 1e-5		•				
⊙ <u>T</u> im	ie:	0	A V				
<mark>∢ R</mark> un sca	n	Dwell time value in seconds (0 to 1	000000)				
		ОКС	ancel				

Figure 31 - For better results, try Horizontal and Vertical increment by 2 instead of the default value



Figure 32 - Scan in progress



© Copyright 2016 Xilinx



Figure 33 - Generated Eye Diagram

Note: 'Enable In-System IBERT' should not be used with the 'Falling Edge Receiver Detect' option in the GT Settings tab.

Descrambler Module

The data on the PIPE interface is scrambled and hence the incoming and outgoing data cannot be read on this interface. The ability to interpret the data on this interface is especially useful where packets presented by the endpoint user application do not show up at the host or vice versa. Being able to identify the corresponding packets on the PIPE interface confirms whether the ingress packets definitely made it into the FPGA and whether the egress packets were definitely presented to the transceiver (PHY) by the PCIe MAC Hard IP.

In Vivado 2016.3, a new feature has been added where the user has an option to enable descrambler module to descramble the PIPE data. Figure 34 shows where the descrambler module sits. The descrambled data is read through an ILA. The instantiated descrambler module is encrypted and only provides a way for hardware-only support to debug Gen3 designs on the board; simulation with the descrambler module is not supported.

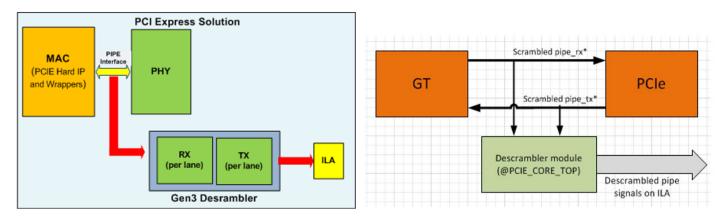


Figure 34 – Descrambler Module

The screenshots below show the step-by-step instruction for enabling the Descrambler Module and viewing the descrambled data on the ILA waveform in the PCI Express Example design on a KCU105 development board.

xpress (4.2)
Component Name pcie3_ultrascale_0
gacy/MSI Cap MSI-X Cap Power Management Extd. Capabilities-1 Extd. Capabilities-2 Shared Logic GT Settings Add. Debug Options
Enable In System IBERT
Enable Descrambler for Gen3 mode
Enable JTAG Debugger

Figure 35 - Enable Descrambler for Gen3 Mode



Project Manager – pcie3_ultrascale_0_ex		
Sources ?	2 2 3	×
a 🔀 🚔 📑 🔁 🖪		
Design Sources (1) Design Sources (2)	e_top.v) (4)	
Hierarchy IP Sources Libraries Compile Order		

Figure 36 - TX Descrambler and RX Descrambler instantiation

	pcie3_	ultra	scale_	0_ex - [,	/home/	/deep	eshm/h	/gen2	debugcheck/pcie3_ultra
<u>F</u> il	e <u>E</u> dit	F <u>l</u> ow	<u>T</u> ools	<u>W</u> indow	Layout	<u>V</u> iew	<u>H</u> elp		
2		ы сл [E	loorplanni	ng				🕨 🔀 🍕 🔚 Debug
	-)w Navig	ator	<u> </u>	/O Plannin	g				• u040-ffva1156-2-e (active
	. 🔀 🖨		I	iming					▶ ? □ L ² ×
				Edit Ti <u>m</u> ing					
⊿	Project	Manag		ower Con	straints A	dvisor.			
	6	Project	1	ch <u>e</u> matic				F4	cale_ep (xilinx_pcie_uscale)
		Add So	S	how <u>C</u> onn	· ·			Ctrl+T F6	le_0_i - pcie3_ultrascale_0
		Langua		i <u>h</u> ow Hiera	rcny			10	scale_0 (pcie3_ultrascale_0
		IP Cata		Re <u>p</u> ort	Dura a set				 cie3_ultrascale_0_pcie3_usi gen_500 - pcie3_ultrascale
				dit Device					gen_250 – pcie3_ultrascale
4	IP Integ	rator		Ireate and Ireate Inte					gen_125 - pcie3_ultrascale
	#	Create	-	Enable Pari					gen_62_5 - pcie3_ultrascal 3_uscale_top_inst - pcie3_u
		Open B		Run Tel Ser		inigurat	1011		Irhold_i - pcie3_ultrascale_
	- 45	Genera		Property Ed	•			Ctrl+J	pp_i - pcie3_ultrascale_0_p
	Simulati	ion	Á	Associate E	L <u>F</u> Files				3_uscale_top_inst - pcie3_t crambler_rx_i - descramble
1				Generate M	lemory C	lonfi <u>gu</u> r	ation File.		crambler_tx_i - descramble
	600 (m)	Simulat		Compile Sir	nulation	Librarie	·S		le_i - pcie_app_uscale (pci(
	<u>uu</u>	Run Sir	🔍 🍭 S	iet Up Deb	ug				•
⊿	RTL An	alysis		(ilinx <u>T</u> cl Si					raries Compile Order
	6	Elabor	0	<u>u</u> stomize	Commar	nds			•
	Þ 🔂	Open E	1	Project <u>S</u> ett	-				? _ 🗆 🖻 🗶
		·	🛛 🖓 L	anguage]	emplate	!S			
4	Synthe			options			_pues_a	scale_ek	
		Conthe-	nia Cattie						

Figure 37 – After synthesizing the design, run 'Set Up Debug'

Find Nets
Find objects in the current design or device by filtering Tcl properties and objects.
Properties
NAME Contains Titssm*
🗌 <u>R</u> egular expression 🖉 Search hierarchically 🗹 <u>D</u> isplay unique nets
Of objects:
Command: show_objects -name NET_ONLY [get_nets -hierarchical -top_net_of_hierarchical_group "*"]
? OK Cancel

Figure 38 – Search cfg_ltssm_state signal



	dd Nets to Debug	×
0	Name	Driver Cell
	♀-{r cfg_ltssm_state (6)	PCIE_3_1
<u> </u>	pcie3_ultrascale_0_i/inst/ltssm_reg2 (6)	FDRE
1	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[0]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[1]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[2]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[3]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[4]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/ltssm_reg1_reg[5]_srl2_n_0	SRL16E
	pcie3_ultrascale_0_i/inst/pcie3_uscale_top_inst/pci	LUT6
	└ pcie3_ultrascale_0_i/inst/pcie3_uscale_top_inst/pci	LUT6
		OK Cancel

Figure 39 - Add cfg_ltssm_state to Debug

Nets to Debug

The nets below will be debugged with ILA cores. To add nets click "Find Nets to Add". You can also select nets in the Netlist or other windows, then drag them to the list or click "Add Selected Nets".

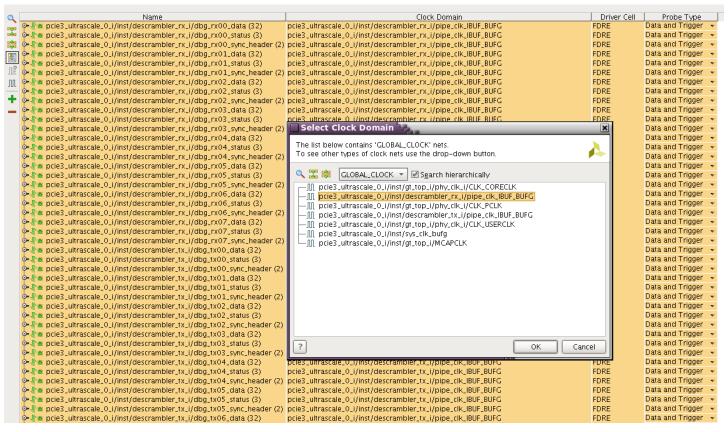


Figure 40 - Select the same clock domain for all signals

© Copyright 2016 Xilinx



<pre>pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx06_status (3) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_data (32) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_status (3) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_status (3) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_sync_header g_tssm_state (6) pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block</pre>	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE</pre>	BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG		FDRE FDRE FDRE FDRE FDRE FDRE FDRE
<pre>pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_data (32) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_status (3) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_sync_header g_ltssm_state (6) pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block</pre>	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6 pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_l6</pre>	BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG		FDRE FDRE FDRE PCIE_ FDRE
pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_status (3) pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_sync_header g_tssm_state (6) pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE	8UF_BUFG 8UF_BUFG 8 <mark>UF_BUFG</mark> 8UF_BUFG 8UF_BUFG		FDRE FDRE PCIE_ FDRE
pcie3_ultrascale_0_i/inst/descrambler_tx_i/dbg_tx07_sync_header g_tssm_state(6) pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	<pre>r (2) pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE</pre>	BUF_BUFG BUF_BUFG BUF_BUFG BUF_BUFG		FDRE PCIE_ FDRE
g_ltssm_state (6) pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE	BUF_BUFG BUF_BUFG BUF_BUFG		PCIE_ FDRE
ocie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_data_valid ocie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block ocie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid ocie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE	BUF_BUFG BUF_BUFG		FDRE
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE	BUF_BUFG		
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE	-		
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block				FDRE
				FDRE
	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_cik_ie pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_cik_ie			FDRE
<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx02_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx02_start_block</pre>				FDRE
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx03_data_valid	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_cik_i6			FDRE
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx03_data_valu		-		FDRE
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx03_statc_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx04_data_valid	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_le			FDRE
				FDRE
				FDRE
				FDRE
peles_ultrascale_o_t/hist/descrambler_tx_t/ubg_txos_start_block	peles_un ascale_o_mismuesci ampier_rx_mpipe_cik_ie	JOF_BOFG		FURE
🕱 pcie3_ultrascale_0_i/inst/store_ltssm 👘 👘 pcie3_ultrascale_0_i/in	nst/descrambler_rx_i/pipe_clk_IBUF_BUFG	LUT2	Data and Trigger	* -
nd Nets to Add			Nets to debug:	625
			Neto to debug.	~2.5
	< Back	Next >	Finish Can	icel
pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx04_start_block pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx05_data_valid pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx05_start_block	pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE pcie3_ultrascale_0_i/inst/descrambler_rx_i/pipe_clk_lE nst/descrambler_rx_i/pipe_clk_lBUF_BUFG	BUF_BUFG BUF_BUFG BUF_BUFG	Data and Trigger Nets to debug: Einish Can	FD FD FD

Figure 41 – In addition to signals related to descrambler, there will be two additional signals: store_ltssm and cfg_ltssm_state in the debug nets

Note: store_Itssm is used to capture data on every transition of the Itssm states. To do that, enable 'Capture Control' in 'Set up Debug' as shown in Figure 42.

Set Up Debug
ILA Core Options
Choose features for the ILA debug cores.
Sample of data depth: 1024 🔹
Input pipe stages: 0 🔹
Trigger and Storage Settings
Capture control
Advanced trigger

Figure 42 - Enable 'Capture Control'

Synthesized Design - xcku040-ffva1156-2-e (active)	
Sources ? = 🗆 🖻 🗶	🔄 Schematic 🗙 🚯 xilinx_pcie3_uscale_ep_x8g3.xdc 🔀
오 🔀 🚔 📑 🔮 🛃	/home/deepeshm/h/gen2debugcheck/pcie3_ultrascale_0_ex/imports/xilinx_pcie3_uscale_ep_x8g3.xdc
Consisting Sources (1) Consisting Sourc	<pre>//Inite/deepsimi/i/genciecougcieck/pices_undascale_0_ex/imputes/imputes_unda_pices_undascale_0_i/inst/descrambler. 333 cet_property port_width 2 [get_debug_ports u_ila_0/probe45] 334 cennect_debug_port u_ila_0 probe 335 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe46] 336 connect_debug_port u_ila_0 probe 337 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe47] 338 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe47] 339 set_property port_width 32 [get_debug_ports u_ila_0/probe47] 340 connect_debug_port u_ila_0/probe47 [get_nets [list {prie3_ultrascale_0_i/inst/descrambler. 341 create_debug_port u_ila_0/probe47 [get_nets [list {prie3_ultrascale_0_i/inst/descrambler. 343 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe48] 344 connect_debug_port u_ila_0/probe48 [get_nets [list {cfg_ltssm_state[0]} {cfg_ltssm_state[]} 346 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe49] 347 set_property port_width 6 [get_debug_ports u_ila_0/probe49] 346 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe49] 347 set_property port_width 1 [get_debug_ports u_ila_0/probe49] 348 connect_debug_port u_ila_0/probe49 [get_nets [list prie3_ultrascale_0_i/inst/descrambler_ 349 create_debug_port u_ila_0/probe50] 352 connect_debug_port u_ila_0/probe50 [get_nets [list prie3_ultrascale_0_i/inst/descrambler_ 353 create_debug_port u_ila_0/probe51 [get_nets [list prie3_ultrascale_0_i/inst/descrambler_ 353 create_debug_port u_ila_0/probe51 [get_nets [list prie3_ultrascale_0_i/inst/descrambler_ 355 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe51] 355 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe51] 356 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe52] 350 connect_debug_port u_ila_0 probe 358 set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe52] 350 connect_debug_port u_ila_0/probe51 [get_nets [list prie3_ultrascal</pre>

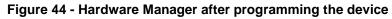
Figure 43 – After the 'Set up Debug' is complete, save the project. The XDC file should be updated with the ILA constraints

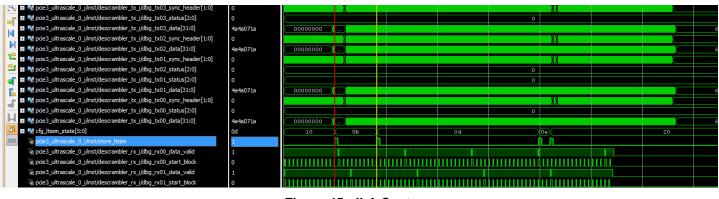
Select 'store_Itssm' in the Trigger Setup.

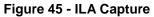
Hardware Manager - localhost/xilinx_tcf/Digilent/210308956781			
Hardware ? = 🗆 🖻 ×	<u>s</u> hw_ila_1 ×		
으, 🔀 🖨 📕 🕨 🕨 🔳	Settings - hw ila 1 ? _ 🗆 ×	Status - hw ila 1	
Name Status □ is localhost (1) Connected □ is 0 xlinx_tcf/Diglent/210308955781 (1) Open □ is xdcu040,0 (2) Programmed □ is ysvMon (System Monitor) Programmed	Trigger Mode Settings	Core status Ide Pre-Trigger Waiting for Trigger Post-Trigger Full Capture status <	Ī
wysładni (systelni monius) i hw.jla_1 (u_jla_0) ○ Idle	Capture Mode Settings Capture Mode Settings Capture mode: BASIC • Number of windows: 1 [1 - 1024] Window data deept: 1024 • [1 - 1024]	Vindow 1 of 1 Window sample 0 of 1024 Tidle Idle Idle Idle	
	Index position in vindow: 50 [0 - 1023] General Settings Refresh rate: 500 ms	Trigger Setup - <u>Inv. ila.1</u> ? X Name Operator Radix Value Port + pole3_ultrascale_0_ul/not/store_itsom == * [8] * [8. * proble proble	Capture Setup - <u>two & a 1</u>
Contraction of the second sec		Waveform - <u>twy fin 1</u>	
Type: ILA Probe type: Data and Trigger Width: 1 Display Name © Long name: pole3_ultrascale_0_i/inst/store_itsss © Short name: store_itssm © Custom name:		Image: the probability of th	
		The set of the se	

© Copyright 2016 Xilinx

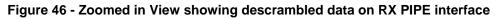




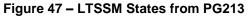




			<u> </u>	n		T	-		<u> </u>	n		n <u> </u>	<u> </u>	í	
	pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx06_status[2:0]	0											0		
_	<pre>multi-scale_0_i/inst/descrambler_rx_i/dbg_rx07_data[31:0]</pre>	0038000e	c	0£07	0038	4a4a	4a4a	0£07	0038	4a4a	8e4a	0f07	0038	4a4a	4a4
Q+	pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx07_sync_header[1:0]	3	0				0	1	<u>(3</u>)	<u> </u>	X		Х З		0
Q-	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx02_status[2:0]</pre>	0											0		
٩	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx00_status[2:0]</pre>	0											0		
	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data[31:0]</pre>	0038000e	d.,	0f01	0038	4a4a	4a4a	0f01	0038	4a4a	4a4a	0f01	0038	4a4a	4 44
	U pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_data_valid	1													
	U pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx01_start_block	0													
	💀 📲 pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx02_data[31:0]	0038000e	4	0f02	0038	(4a4a)	9443	(Of02	0038	4a4a	<u>a592</u>	0f02	X 0038	4a4a	X 049
*	pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx02_sync_header[1:0]	3	2			3			1		2	1	<u>)</u> 2	0	X
2	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx03_status[2:0]</pre>	0											0		
a r	<pre>pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx04_data[31:0]</pre>	0038000e	0	0f04	0038	4a4a	4a4a	0f04	0038	4a4a	514a	0f04	0038	4a4a	114
R	pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx04_sync_header[1:0]	3	0		3		2	1	X		Х З			1	
	pcie3_ultrascale_0_i/inst/descrambler_rx_i/dbg_rx05_status[2:0]	0											0		
-										,			~	/	~ <u> </u>



IN_LUS_IIS = 3 LTSSM State. Shows the current LTSSM state: 00: Detect.Quiet 01: Detect.Active 02: Polling.Active 03: Polling.Compliance 04: Polling.Configuration 05: Configuration.Linkwidth.Start 06: Configuration.Linkwidth.Accept 07: Configuration.Lanenum.Accept 08: Configuration.Lanenum.Wait 09: Configuration.Complete 0A: Configuration.Idle 0B: Recovery.RcvrLock 0C: Recovery.Speed 0D: Recovery.RcvrCfg 0E: Recovery.Idle 10: L0 11-16: Reserved cfg_ltssm_state Output 6 17: L1.Entry 18[.] I 1 Idle



Capture status					
Window 1 of 1 Idle	Window sample 0 of Idle		ble 0 of 1024 idle		
Trigger Setup - <u>hw ila 1</u>				? _ 🗆 ×	Capture Setup - <u>hw ila 1</u>
Name	Opera	tor Radix Va	alue	Port	
cfg_ltssm_state[5:0]	==	▼ [H] ▼ 05		- prob	+
-					
•				*	
Waveform - <u>hw_ila_1</u>					
177 B					67
ILA Status:Idle					

Figure 48 - Trigger Setup to trigger when cfg_ltssm_state is '05' (Configuration.Linkwidth.Start)

0			U	
0			0	
0000000			0000000	
0			0	
0			0	
0000000			0000000	
08	04	<u>05</u>	06	
0		1		
0				
0				
0				
0				
0				
	00000000 0 0 00000000 0 0 0 0 0 0 0 0	00000000 0 0 0 0 0 0 0 0 0 0	00000000	00000000 0 00000000 0



Revision History

11/20/2016 - Initial release