Xilinx Answer 72471
UltraScale+ FPGA Gen3 Integrated Block for PCI Express (Vivado 2019.1) - Integrated Debugging Features and Usage Guide

PCIe EoU Integrated Debug Features

Overview

This answer record is an updated version of (Xilinx Answer 68134) in Vivado 2019.1. The target device is a Virtex UltraScale+ VCU118 Evaluation Kit. The documentation also includes a step-by-step tutorial on how to enable and use the following debug features:

- JTAG Debugger
- Enable In-System IBERT
- Descrambler in Gen3 Mode

General Design Steps in Configuring the PCIe Core

Invoke Vivado 2019.1 and configure the PCIe IP core by clicking "Create Project".

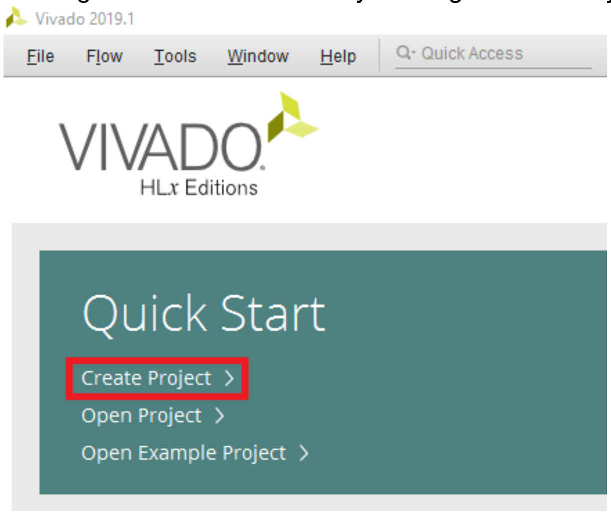
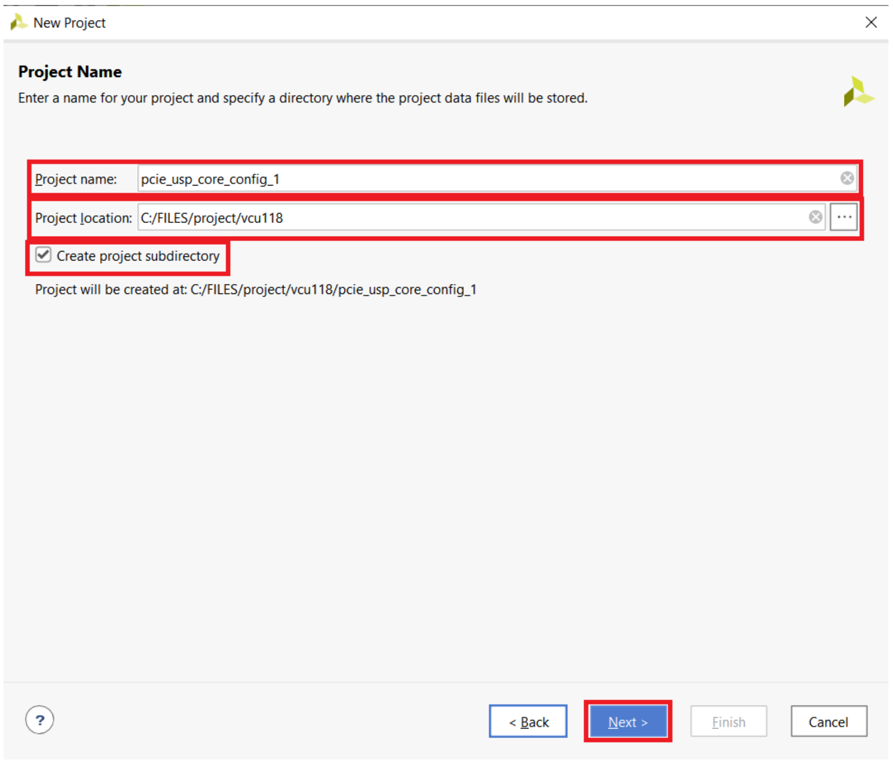


Figure 1 - Create project

Provide a desired project name for each PCIe IP core configuration. Tick the checkbox for "Create project subdirectory". In this documentation, the project names for PCIe IP core configuration correspond to the debug tools as follows:

- pcie_usp_core_config_1 → jtag_debugger_1
- pcie_usp_core_config_2 → in_system_ibert_2
- pcie_usp_core_config_3 → descrambler_3


New Project
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: pcie_usp_core_config_1

Project location: C:/FILES/project/vcu118

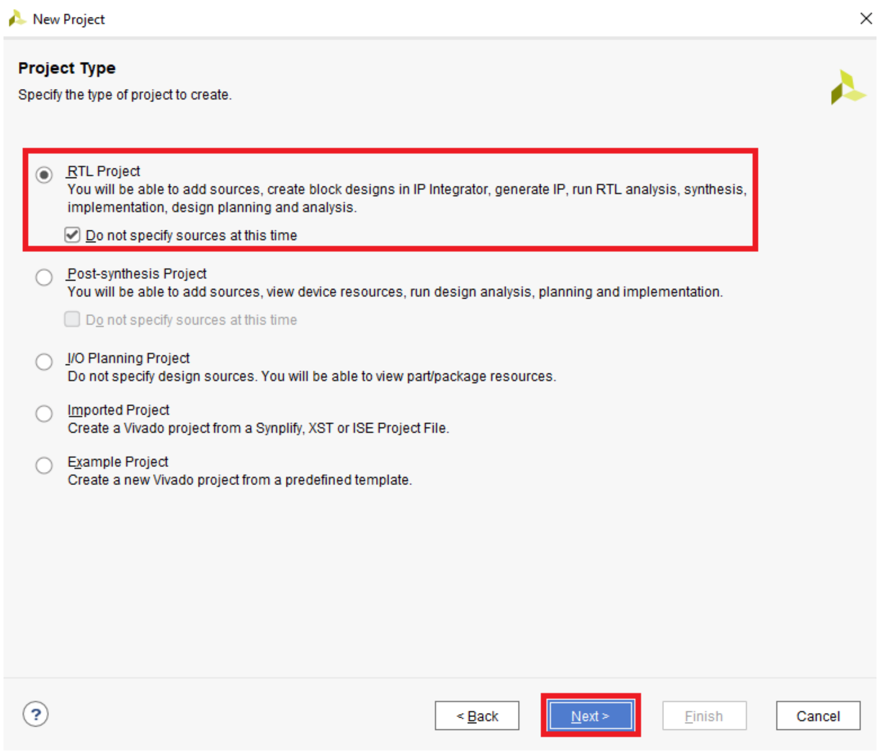
[✓] Create project subdirectory

Project will be created at: C:/FILES/project/vcu118/pcie_usp_core_config_1

< Back
Next >
Finish
Cancel

Figure 2 - Choose the project name

Select **RTL Project** and tick the check box for "Do not specify sources at this time". Click "Next".



● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
[✓] Do not specify sources at this time

○ Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time

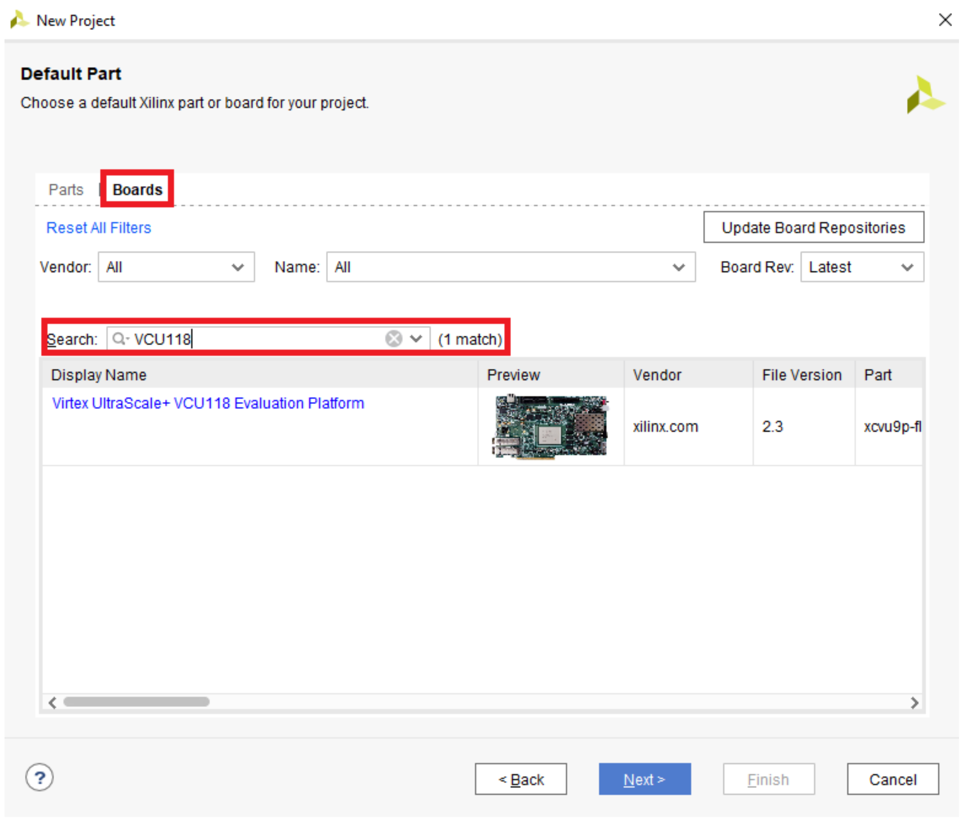
○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

○ Example Project
Create a new Vivado project from a predefined template.

Figure 3 - Select project type

Click **Boards** and enter VCU118 in the search field. Select the **VCU118 Evaluation** board. Click "Next".



New Project

Default Part

Choose a default Xilinx part or board for your project.

Parts | **Boards**

Reset All Filters | Update Board Repositories

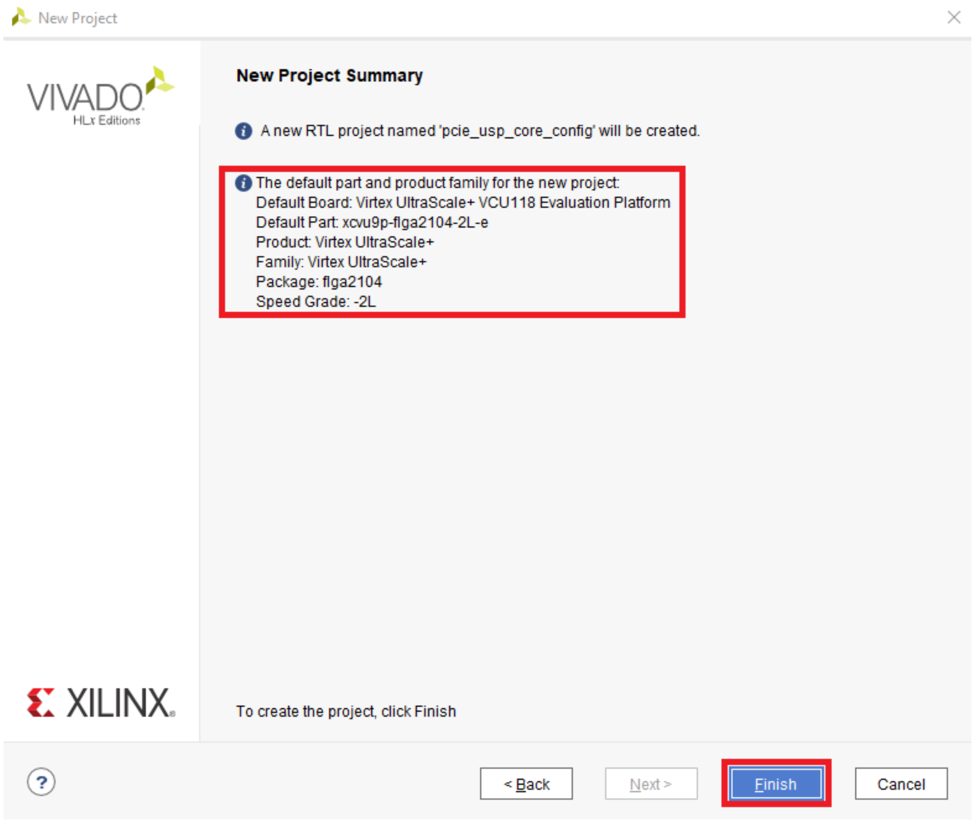
Vendor: All | Name: All | Board Rev: Latest

Search: VCU118 (1 match)

Display Name	Preview	Vendor	File Version	Part
Virtex UltraScale+ VCU118 Evaluation Platform		xilinx.com	2.3	xcvu9p-fl

Figure 4 - Select the board for the project

A message dialog box will show the "New Project Summary". Make sure that the target device is properly selected.



New Project

New Project Summary

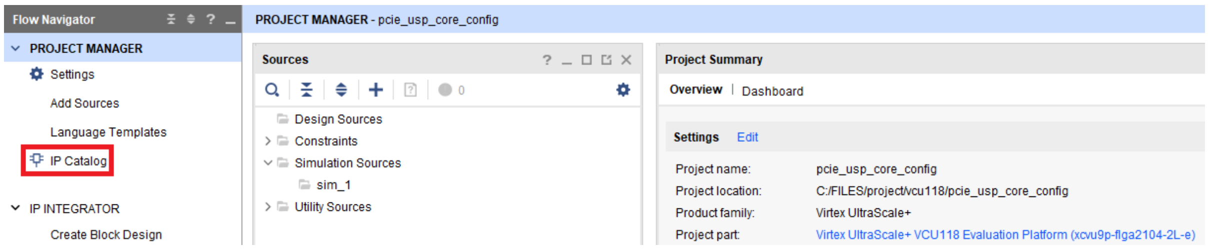
A new RTL project named 'pcie_usp_core_config' will be created.

The default part and product family for the new project:
Default Board: Virtex UltraScale+ VCU118 Evaluation Platform
Default Part: xcvu9p-flga2104-2L-e
Product: Virtex UltraScale+
Family: Virtex UltraScale+
Package: flga2104
Speed Grade: -2L

To create the project, click Finish

Figure 5 - Project summary

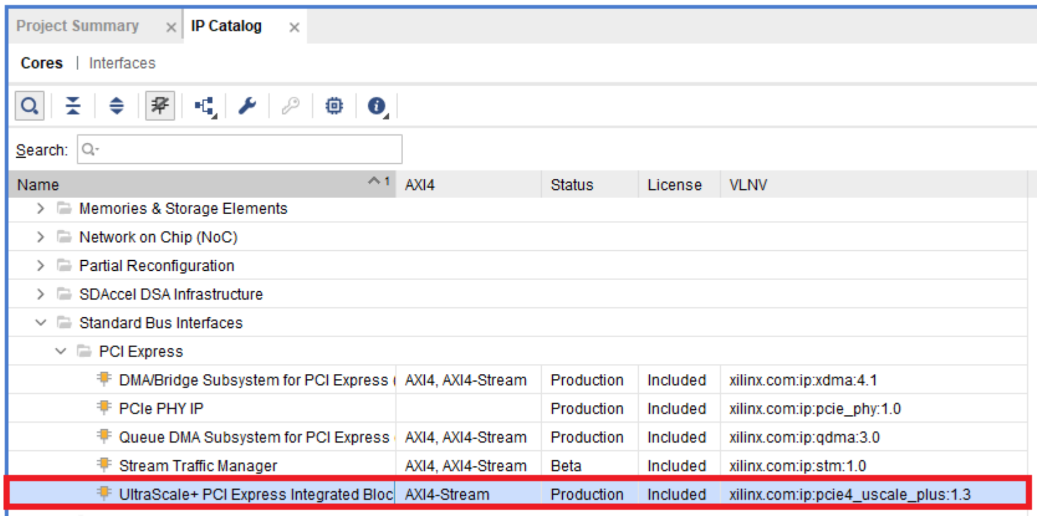
In the Flow Navigator window, click on “IP Catalog”.



Project name: pcie_usp_core_config
Project location: C:/FILES/project/vcu118/pcie_usp_core_config
Product family: Virtex UltraScale+
Project part: Virtex UltraScale+ VCU118 Evaluation Platform (xcvu9p-flga2104-2L-e)

Figure 6 - IP catalog

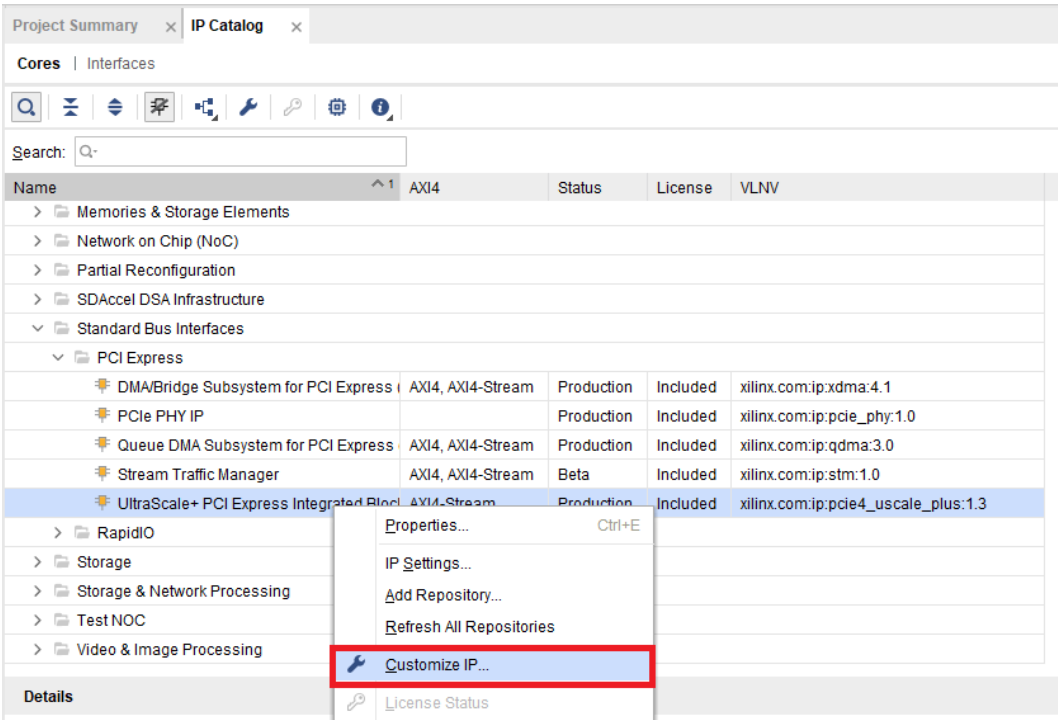
Select “UltraScale+ PCI Express Integrated Block” under PCI Express of Standard Bus Interface.



Name	AXI4	Status	License	VLNV
Memories & Storage Elements				
Network on Chip (NoC)				
Partial Reconfiguration				
SDAccel DSA Infrastructure				
Standard Bus Interfaces				
PCI Express				
DMA/Bridge Subsystem for PCI Express (	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:xdma:4.1
PCIe PHY IP		Production	Included	xilinx.com:ip:pcie_phy:1.0
Queue DMA Subsystem for PCI Express	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:qdma:3.0
Stream Traffic Manager	AXI4, AXI4-Stream	Beta	Included	xilinx.com:ip:stm:1.0
UltraScale+ PCI Express Integrated Bloc	AXI4-Stream	Production	Included	xilinx.com:ip:pcie4_uscale_plus:1.3

Figure 7 - PCIe IP integrated block

Right-click on the selected IP and then click on “Customize IP…” from the drop-down menu.



Name	AXI4	Status	License	VLNV
Memories & Storage Elements				
Network on Chip (NoC)				
Partial Reconfiguration				
SDAccel DSA Infrastructure				
Standard Bus Interfaces				
PCI Express				
DMA/Bridge Subsystem for PCI Express (	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:xdma:4.1
PCIe PHY IP		Production	Included	xilinx.com:ip:pcie_phy:1.0
Queue DMA Subsystem for PCI Express	AXI4, AXI4-Stream	Production	Included	xilinx.com:ip:qdma:3.0
Stream Traffic Manager	AXI4, AXI4-Stream	Beta	Included	xilinx.com:ip:stm:1.0
UltraScale+ PCI Express Integrated Bloc	AXI4-Stream	Production	Included	xilinx.com:ip:pcie4_uscale_plus:1.3
RapidIO				
Storage				
Storage & Network Processing				
Test NOC				
Video & Image Processing				

Figure 8 - Customizing PCIe IP

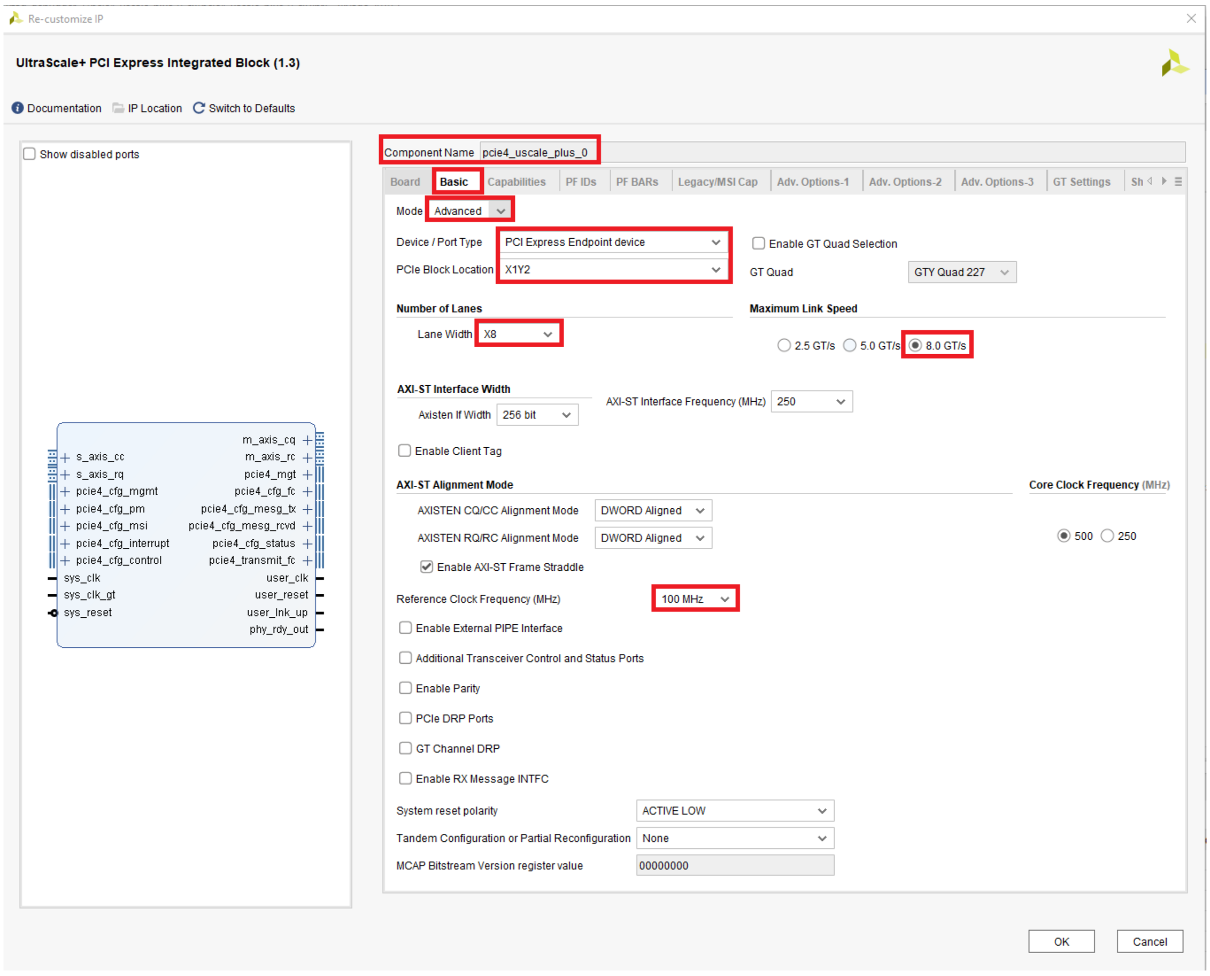
JTAG Debugger

This a debug feature that captures a diagram that shows the Link Training and Status State Machine (LTSSM) which includes the following:

- Link training state diagram
- Reset sequence state diagram
- Receiver detect diagram

Configure the following settings in the “Basic” tab.

- Leave the default “Component Name”.
  Change the “Mode” to **Advanced** to unlock all of the features of the IP.
- Make sure that the “Device/Port Type” is **PCI Express Endpoint device** and the “PCIe Block Location” is at **X1Y2**.
- Change the “Lane Width” to **X8** and the “Maximum Link Speed” to **8.0 GT/s**.
- Check that the “Reference Clock Frequency (MHz)” is set to **100MHz**.



UltraScale+ PCI Express Integrated Block (1.3)

Component Name: pcie4_uscale_plus_0

Mode: Advanced

Device / Port Type: PCI Express Endpoint device

PCIe Block Location: X1Y2

☐ Enable GT Quad Selection

GT Quad: GTY Quad 227

Number of Lanes

Lane Width: X8

Maximum Link Speed

○ 2.5 GT/s ○ 5.0 GT/s [✓ 8.0 GT/s]

AXI-ST Interface Width

Axisten If Width: 256 bit

AXI-ST Interface Frequency (MHz): 250

☐ Enable Client Tag

AXI-ST Alignment Mode

AXISTEN CQ/CC Alignment Mode: DWORD Aligned

AXISTEN RQ/RC Alignment Mode: DWORD Aligned

☑ Enable AXI-ST Frame Straddle

Core Clock Frequency (MHz)

[✓ 500] ○ 250

Reference Clock Frequency (MHz): 100 MHz

☐ Enable External PIPE Interface

☐ Additional Transceiver Control and Status Ports

☐ Enable Parity

☐ PCIe DRP Ports

☐ GT Channel DRP

☐ Enable RX Message INTFC

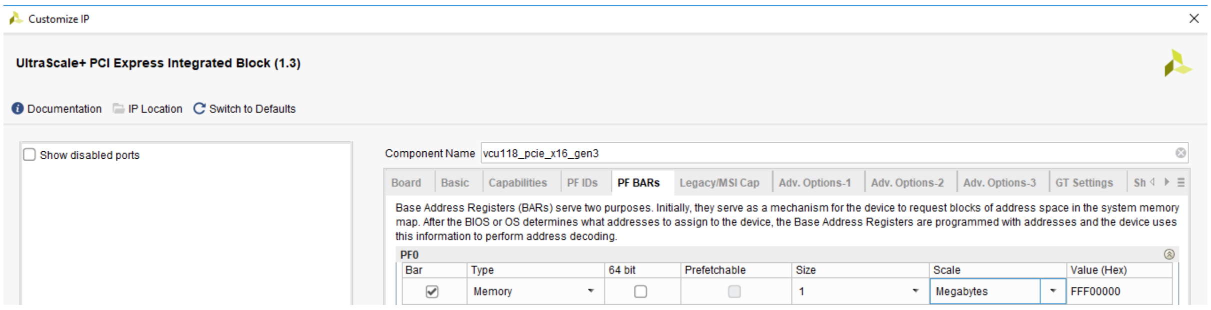
System reset polarity: ACTIVE LOW

Tandem Configuration or Partial Reconfiguration: None

MCAP Bitstream Version register value: 00000000

Figure 9 - Customizing PCIe core

In the “PF BARs” tab, change the “Size” and “Scale” to **1 Megabytes**.


Customize IP

UltraScale+ PCI Express Integrated Block (1.3)

Documentation IP Location Switch to Defaults

Show disabled ports

Component Name vcu118_pcie_x16_gen3

Board | Basic | Capabilities | PF IDs | PF BARs | Legacy/MSI Cap | Adv. Options-1 | Adv. Options-2 | Adv. Options-3 | GT Settings | Sh

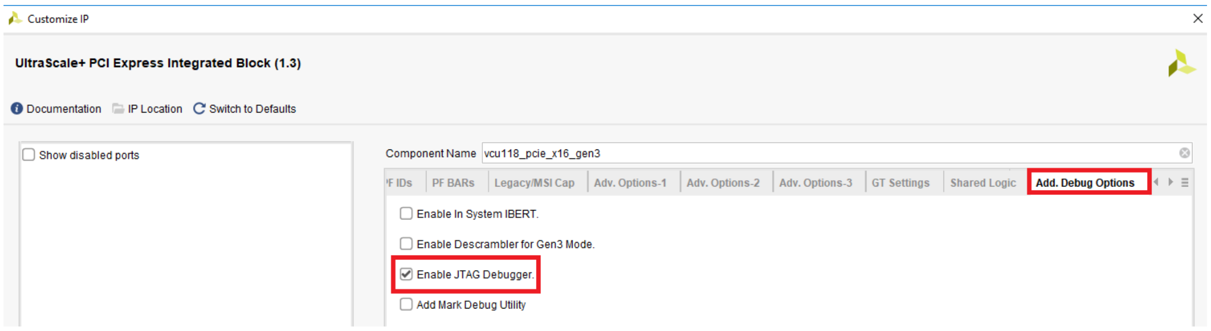
Base Address Registers (BARs) serve two purposes. Initially, they serve as a mechanism for the device to request blocks of address space in the system memory map. After the BIOS or OS determines what addresses to assign to the device, the Base Address Registers are programmed with addresses and the device uses this information to perform address decoding.

PF0

Bar	Type	64 bit	Prefetchable	Size	Scale	Value (Hex)
✓	Memory	☐	☐	1	Megabytes	FFF00000

Figure 10 - Customizing PF BARs

Go to the “Add. Debug Options” tab and tick the checkbox for “Enable JTAG debugger”.


Customize IP

UltraScale+ PCI Express Integrated Block (1.3)

Documentation IP Location Switch to Defaults

Show disabled ports

Component Name vcu118_pcie_x16_gen3

F IDs | PF BARs | Legacy/MSI Cap | Adv. Options-1 | Adv. Options-2 | Adv. Options-3 | GT Settings | Shared Logic | Add. Debug Options

☐ Enable In System IBERT.

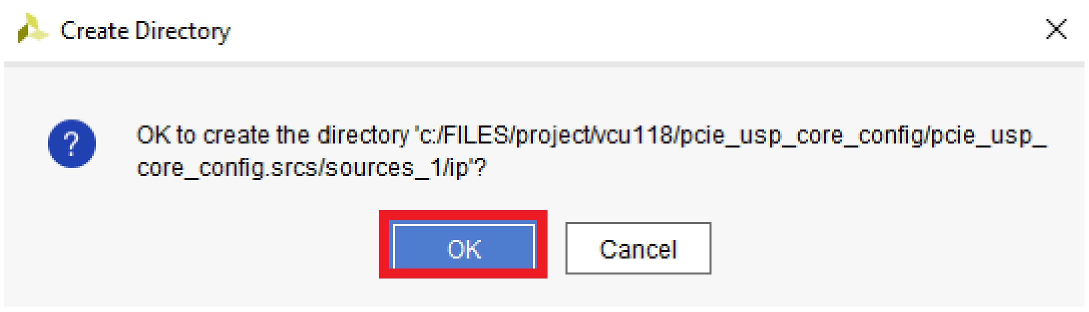
☐ Enable Descrambler for Gen3 Mode.

☑ Enable JTAG Debugger.

☐ Add Mark Debug Utility

Figure 11 - Add debug option

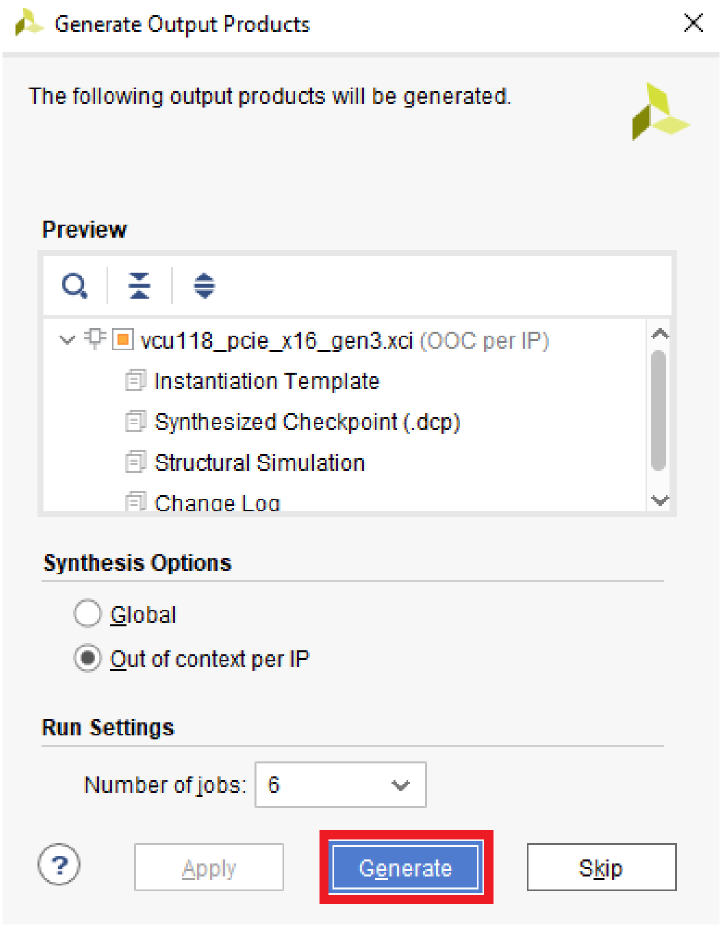
A message dialog box will appear to confirm the IP directory. Click “OK”.


Create Directory

OK to create the directory 'c:/FILES/project/vcu118/pcie_usp_core_config/pcie_usp_core_config.srcs/sources_1/ip'?

Figure 12 - Confirmation of PCIe IP customization

Click “Generate”.


Generate Output Products

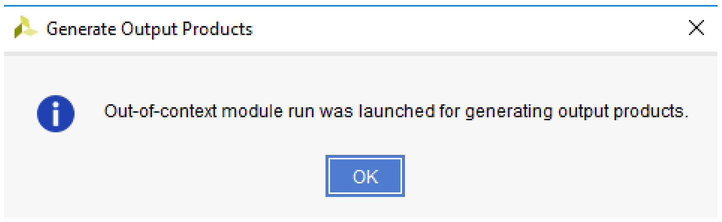
The following output products will be generated.

Synthesis Options: Global [✓ Out of context per IP]

Number of jobs: 6

Figure 13 - Generate output product

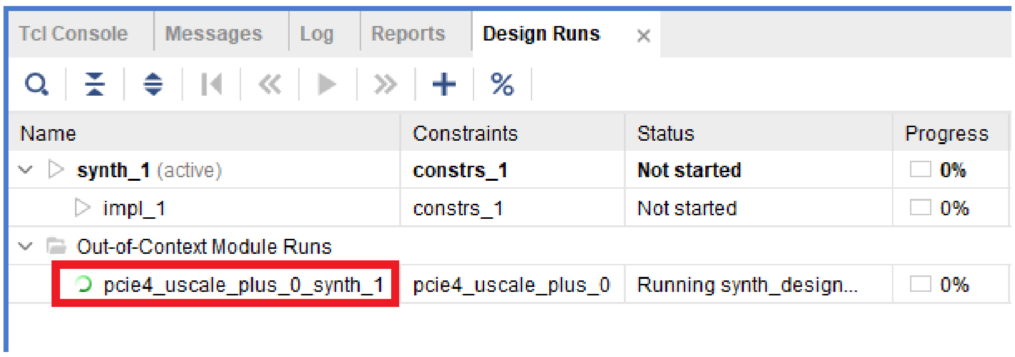
Click “OK”.


Generate Output Products

Out-of-context module run was launched for generating output products.

Figure 14 - Confirmation of generated output product

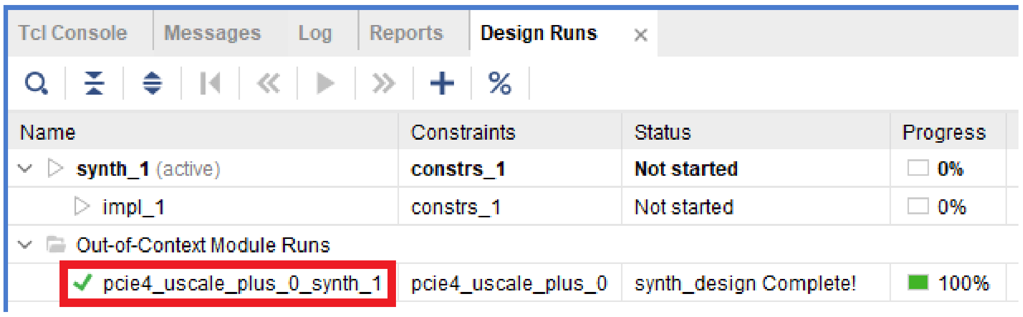
Under the Vivado interface, check the “Design Runs” window that shows the status of synthesizing the **pcie4_uscale_plus_0_synth_1** core configuration.



Name	Constraints	Status	Progress
synth_1 (active)	**constrs_1**	**Not started**	**0%**
impl_1	constrs_1	Not started	0%
Out-of-Context Module Runs			
pcie4_uscale_plus_0_synth_1	pcie4_uscale_plus_0	Running synth_design...	0%

Figure 15 - Synthesizing the generated product

A check mark will appear beside the name of the PCIe core indicating that synthesis is complete.



Name	Constraints	Status	Progress
synth_1 (active)	constrs_1	Not started	0%
impl_1	constrs_1	Not started	0%
Out-of-Context Module Runs			
✓ pcie4_uscale_plus_0_synth_1	pcie4_uscale_plus_0	synth_design Complete!	100%

Figure 16 - Synthesized product

Right-click on “pcie4_uscale_plus_0(pcie4_uscale_plus_0.xci)” in the **Sources** window and click “Open IP Example Design…” from the drop-down menu.

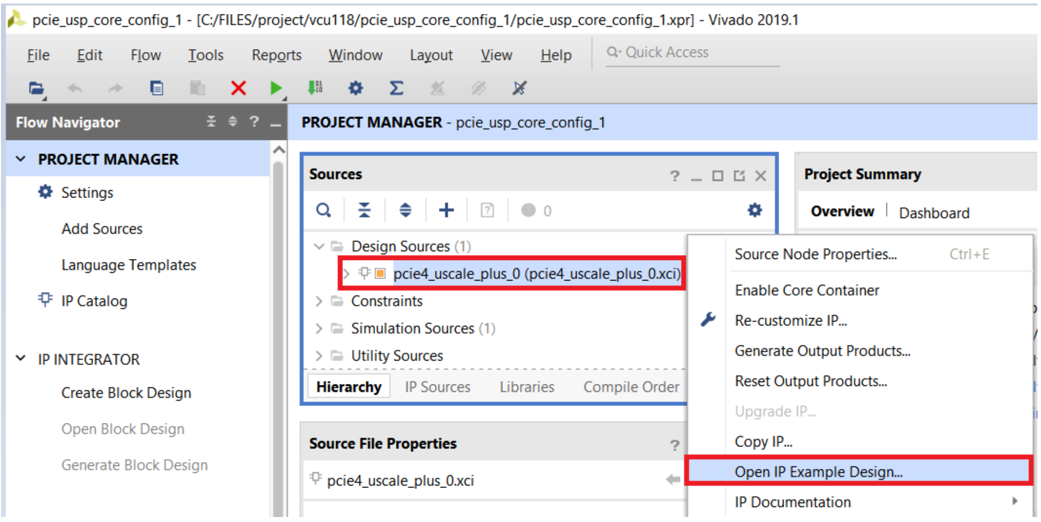
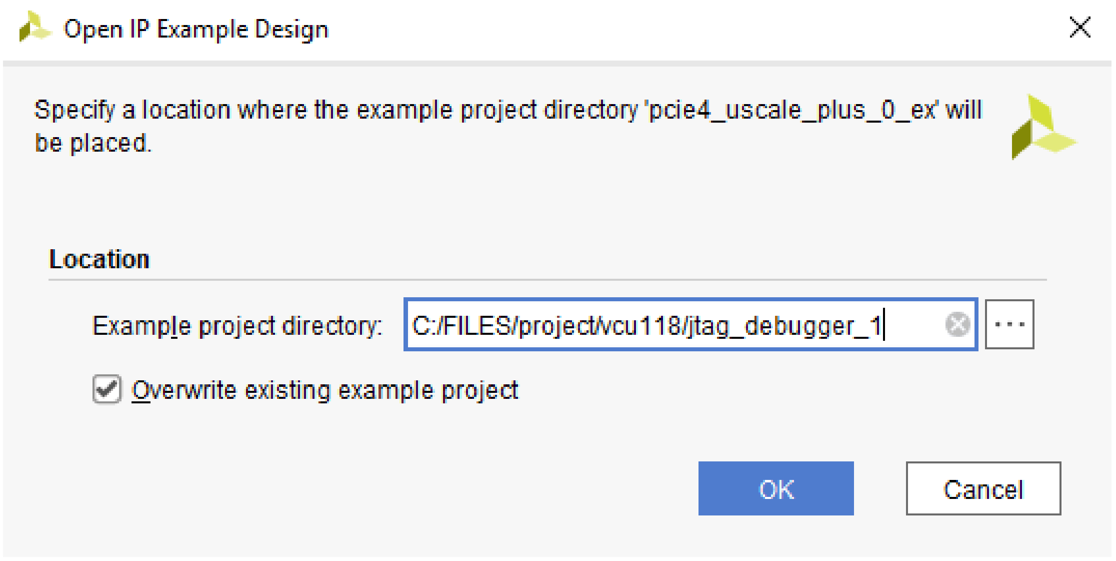

pcie_usp_core_config_1 - [C:/FILES/project/vcu118/pcie_usp_core_config_1/pcie_usp_core_config_1.xpr] - Vivado 2019.1

Figure 17 - Open IP example design

Select a directory path for the example project. Click “OK”.


Open IP Example Design

Specify a location where the example project directory 'pcie4_uscale_plus_0_ex' will be placed.

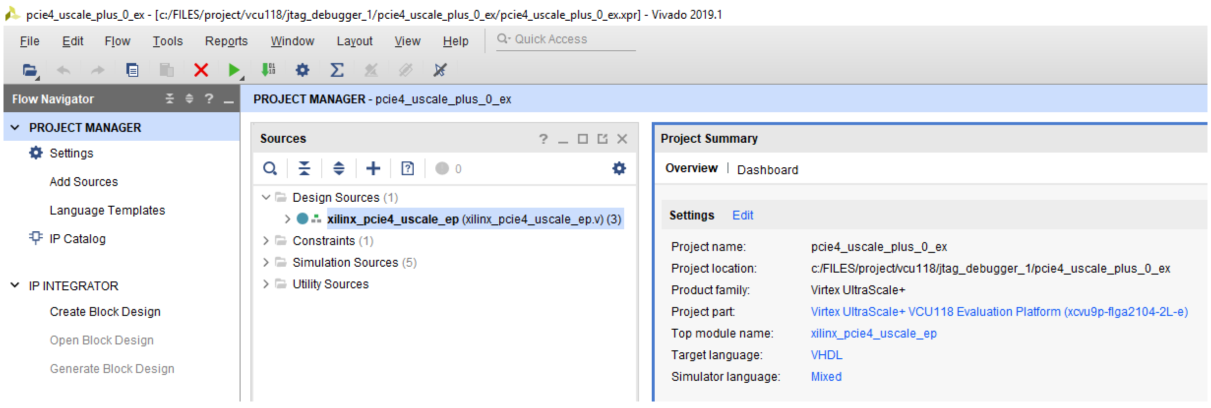
Location

Example project directory: C:/FILES/project/vcu118/jtag_debugger_1

[✓] Overwrite existing example project

Figure 18 - Example project directory

A new Vivado window will open that contains the example reference design.

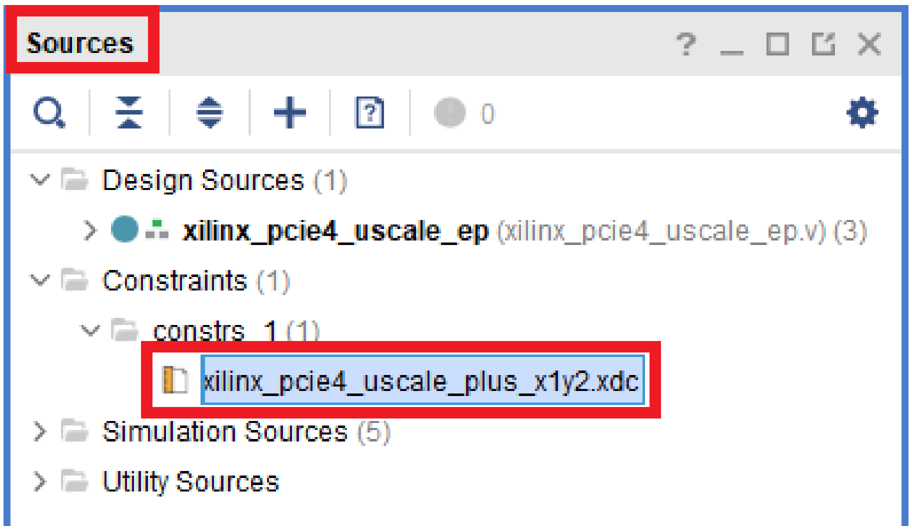

pcie4_uscale_plus_0_ex - [c:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.xpr] - Vivado 2019.1

Design Sources (1)
xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
Constraints (1)
Simulation Sources (5)
Utility Sources

Project name:	pcie4_uscale_plus_0_ex
Project location:	c:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex
Product family:	Virtex UltraScale+
Project part:	Virtex UltraScale+ VCU118 Evaluation Platform (xcvu9p-flga2104-2L-e)
Top module name:	xilinx_pcie4_uscale_ep
Target language:	VHDL
Simulator language:	Mixed

Figure 19 - New Vivado project containing example design

In the **Sources** window, Click and open the constraint file “xilinx_pcie4_uscale_plus_x1y2.xdc”

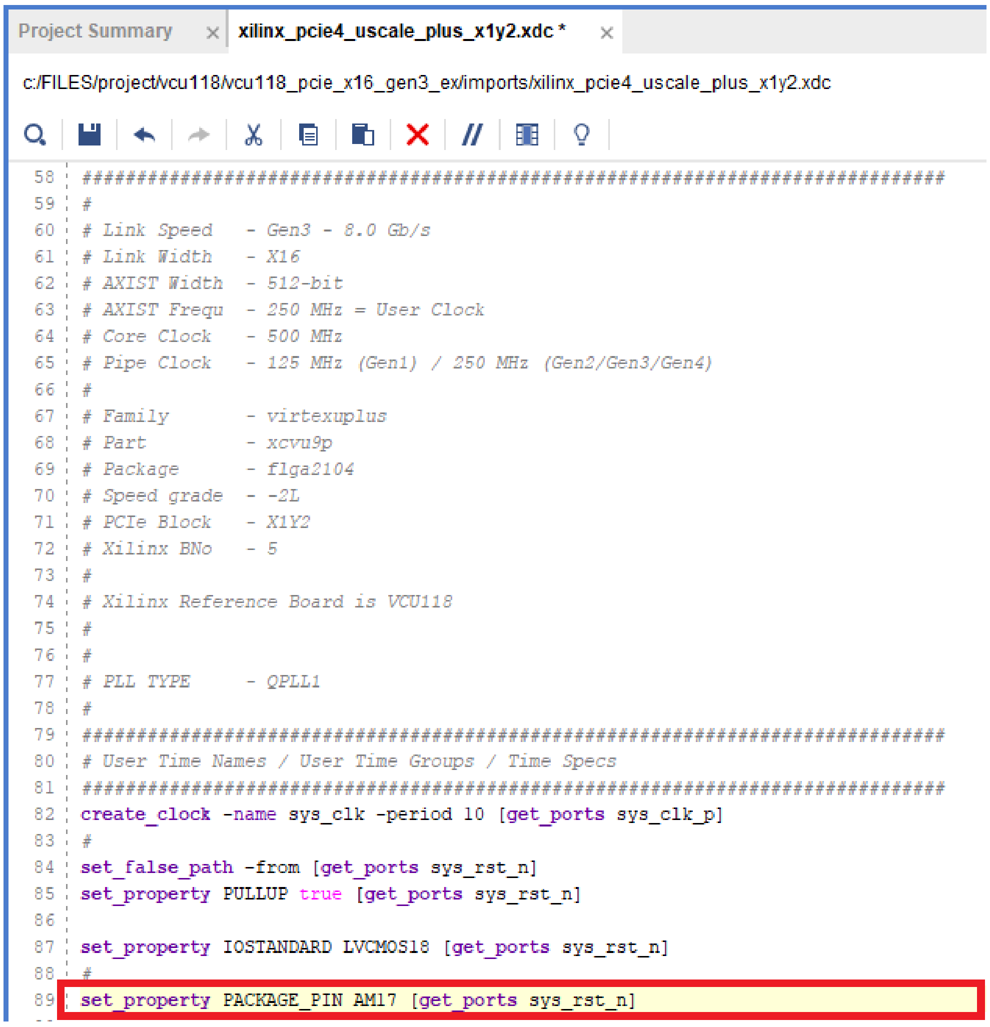


Design Sources (1)
xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
Constraints (1)
constrs_1 (1)
xilinx_pcie4_uscale_plus_x1y2.xdc
Simulation Sources (5)
Utility Sources

Figure 20 - Edit constraint file

Comment out or make the line of code active below.

```
set_property PACKAGE_PIN AM17 [getports sys_rst_n]
```

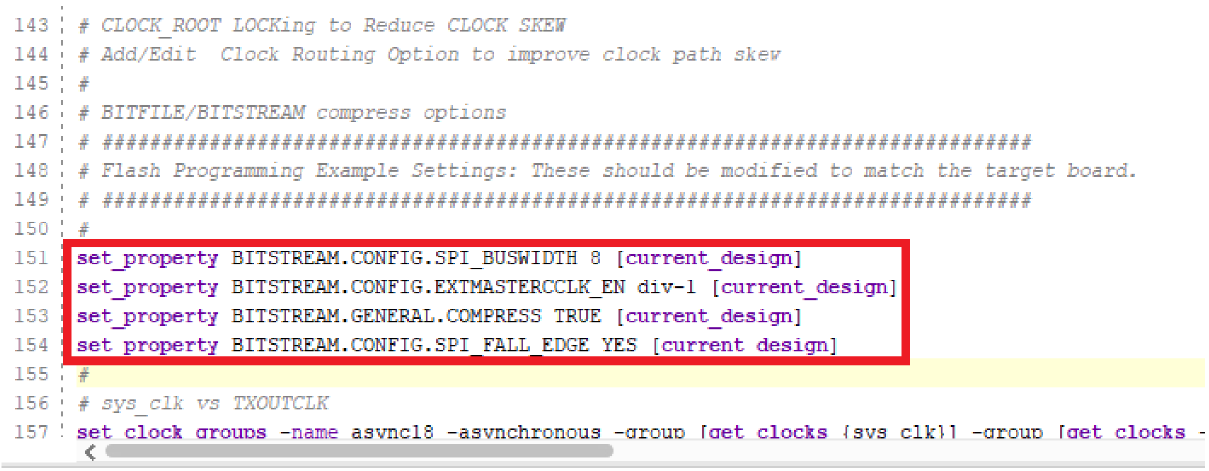

Project Summary
xilinx_pcie4_uscale_plus_x1y2.xdc *
c:/FILES/project/vcu118/vcu118_pcie_x16_gen3_ex/imports/xilinx_pcie4_uscale_plus_x1y2.xdc

```
##############################################################################
#
# Link Speed   - Gen3 - 8.0 Gb/s
# Link Width   - X16
# AXIST Width  - 512-bit
# AXIST Frequ  - 250 MHz = User Clock
# Core Clock   - 500 MHz
# Pipe Clock   - 125 MHz (Gen1) / 250 MHz (Gen2/Gen3/Gen4)
#
# Family       - virtexuplus
# Part         - xcvu9p
# Package      - flga2104
# Speed grade  - -2L
# PCIe Block   - X1Y2
# Xilinx BNo   - 5
#
# Xilinx Reference Board is VCU118
#
#
# PLL TYPE     - QPLL1
#
##############################################################################
# User Time Names / User Time Groups / Time Specs
##############################################################################
create_clock -name sys_clk -period 10 [get_ports sys_clk_p]
#
set_false_path -from [get_ports sys_rst_n]
set_property PULLUP true [get_ports sys_rst_n]

set_property IOSTANDARD LVCMOS18 [get_ports sys_rst_n]
#
set_property PACKAGE_PIN AM17 [get_ports sys_rst_n]
```

Figure 21 – Activate reset pin

Add the following lines of code:

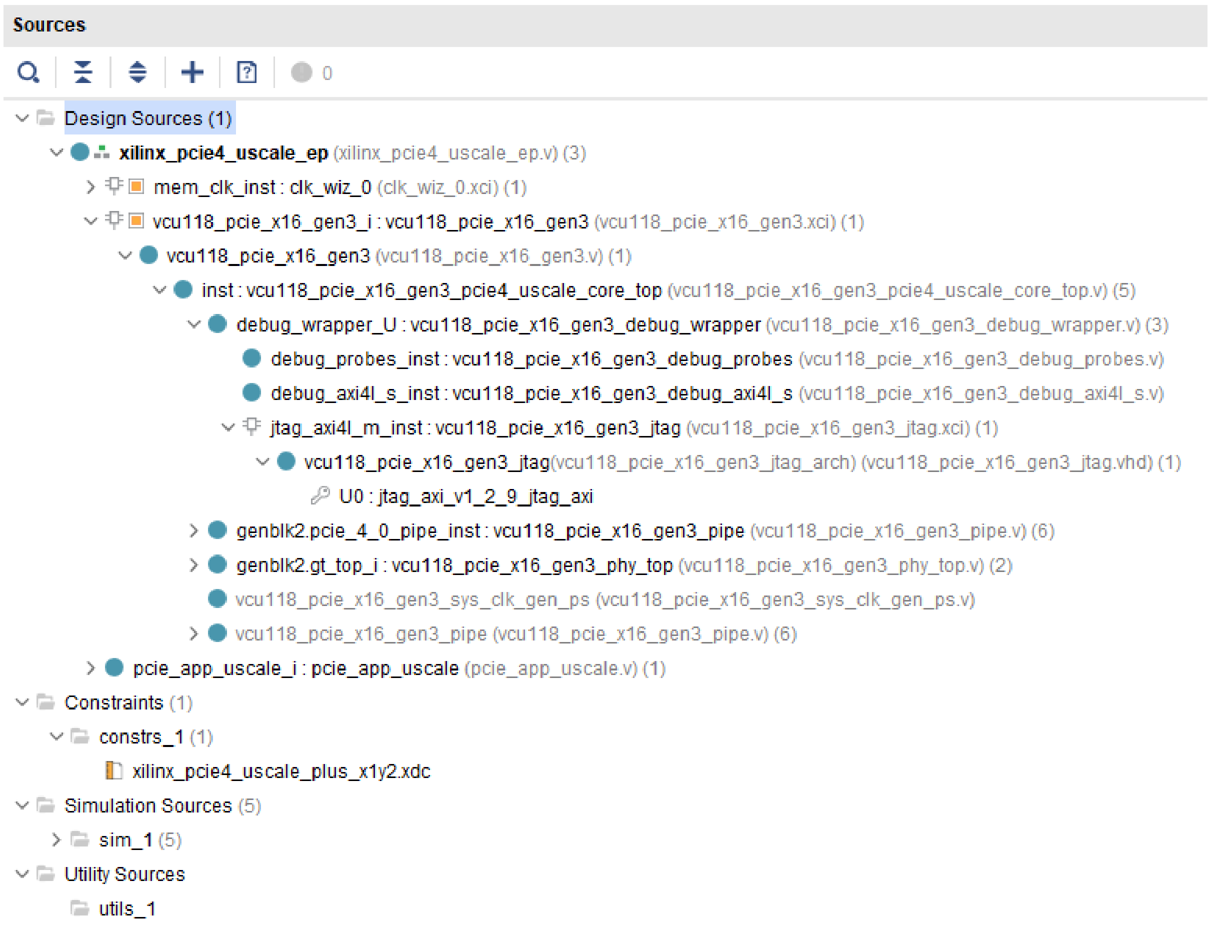
```
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 8 [current_design]
set_property BITSTREAM.CONFIG.EXTMASTERCCLK_EN div-1 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]
```



```
# CLOCK_ROOT LOCKing to Reduce CLOCK SKEW
# Add/Edit  Clock Routing Option to improve clock path skew
#
# BITFILE/BITSTREAM compress options
# ##############################################################################
# Flash Programming Example Settings: These should be modified to match the target board.
# ##############################################################################
#
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 8 [current_design]
set_property BITSTREAM.CONFIG.EXTMASTERCCLK_EN div-1 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]
#
# sys_clk vs TXOUTCLK
set_clock_groups -name async18 -asynchronous -group [get_clocks {sys_clk}] -group [get_clocks -
```

Figure 22 - Adding lines of code

Check the hierarchy of the "Design Sources" to check if it includes a debugger wrapper module.



Sources

Design Sources (1)
- xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
  - mem_clk_inst : clk_wiz_0 (clk_wiz_0.xci) (1)
  - vcu118_pcie_x16_gen3_i : vcu118_pcie_x16_gen3 (vcu118_pcie_x16_gen3.xci) (1)
    - vcu118_pcie_x16_gen3 (vcu118_pcie_x16_gen3.v) (1)
      - inst : vcu118_pcie_x16_gen3_pcie4_uscale_core_top (vcu118_pcie_x16_gen3_pcie4_uscale_core_top.v) (5)
        - debug_wrapper_U : vcu118_pcie_x16_gen3_debug_wrapper (vcu118_pcie_x16_gen3_debug_wrapper.v) (3)
          - debug_probes_inst : vcu118_pcie_x16_gen3_debug_probes (vcu118_pcie_x16_gen3_debug_probes.v)
          - debug_axi4l_s_inst : vcu118_pcie_x16_gen3_debug_axi4l_s (vcu118_pcie_x16_gen3_debug_axi4l_s.v)
          - jtag_axi4l_m_inst : vcu118_pcie_x16_gen3_jtag (vcu118_pcie_x16_gen3_jtag.xci) (1)
            - vcu118_pcie_x16_gen3_jtag(vcu118_pcie_x16_gen3_jtag_arch) (vcu118_pcie_x16_gen3_jtag.vhd) (1)
              - U0 : jtag_axi_v1_2_9_jtag_axi
        - genblk2.pcie_4_0_pipe_inst : vcu118_pcie_x16_gen3_pipe (vcu118_pcie_x16_gen3_pipe.v) (6)
        - genblk2.gt_top_i : vcu118_pcie_x16_gen3_phy_top (vcu118_pcie_x16_gen3_phy_top.v) (2)
        - vcu118_pcie_x16_gen3_sys_clk_gen_ps (vcu118_pcie_x16_gen3_sys_clk_gen_ps.v)
        - vcu118_pcie_x16_gen3_pipe (vcu118_pcie_x16_gen3_pipe.v) (6)
  - pcie_app_uscale_i : pcie_app_uscale (pcie_app_uscale.v) (1)

Constraints (1)
- constrs_1 (1)
  - xilinx_pcie4_uscale_plus_x1y2.xdc

Simulation Sources (5)
- sim_1 (5)

Utility Sources
- utils_1

Figure 23 - Design source hierarchy

Click "Generate Bitstream".

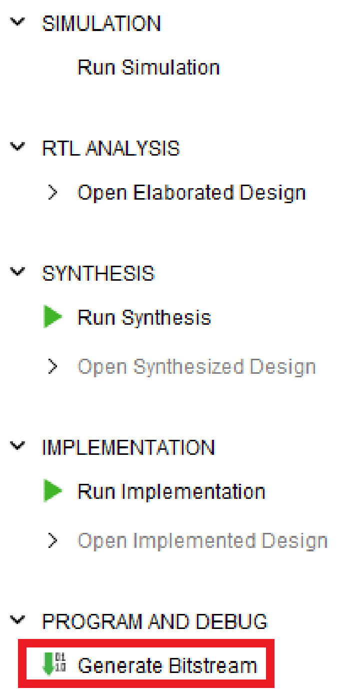
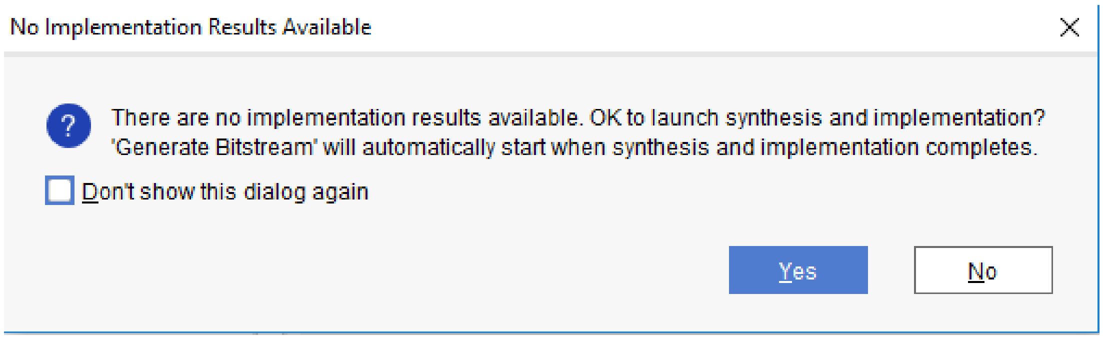


Figure 24 - Generate bitstream

A message dialog box will appear. Click “Yes”.

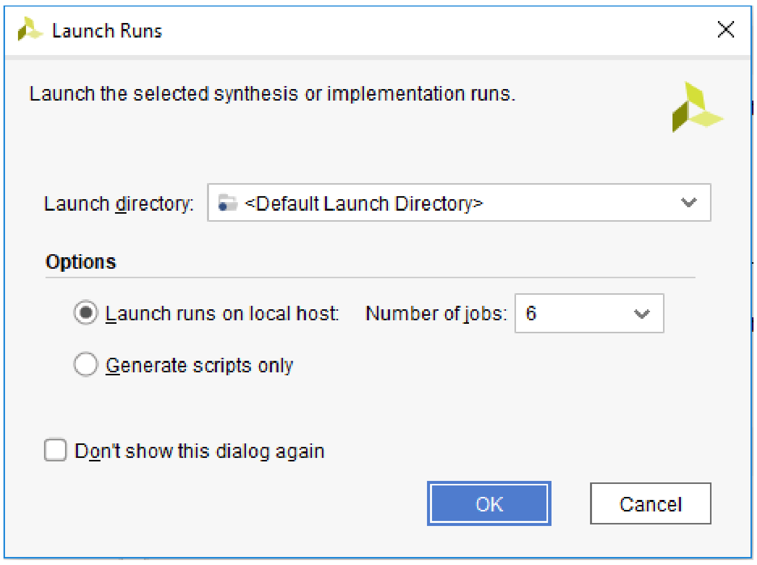

No Implementation Results Available

There are no implementation results available. OK to launch synthesis and implementation? 'Generate Bitstream' will automatically start when synthesis and implementation completes.

☐ Don't show this dialog again

Figure 25 - Launch synthesis and implementation

A message dialog will appear. Click “OK”.


Launch Runs

Launch the selected synthesis or implementation runs.

Launch directory: <Default Launch Directory>

Options

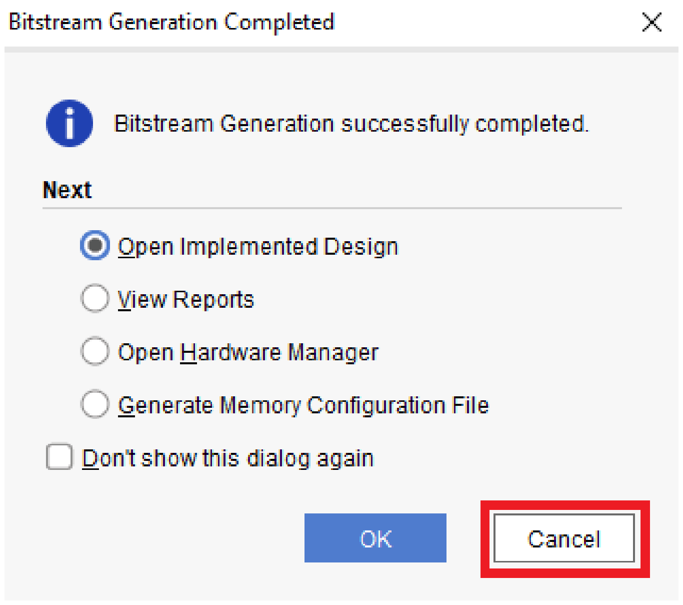
● Launch runs on local host: Number of jobs: 6

○ Generate scripts only

☐ Don't show this dialog again

Figure 26 - Launch Runs

Click “Cancel”.


Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

● Open Implemented Design

○ View Reports

○ Open Hardware Manager

○ Generate Memory Configuration File

☐ Don't show this dialog again

Figure 27 - Bitstream generation completed

Expand “Open Hardware Manager”.

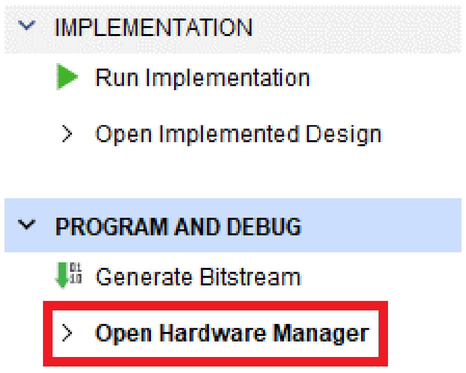


Figure 28 - Open Hardware Manager

Click on “Open Target” and select “Open New Target…” from the drop-down menu.

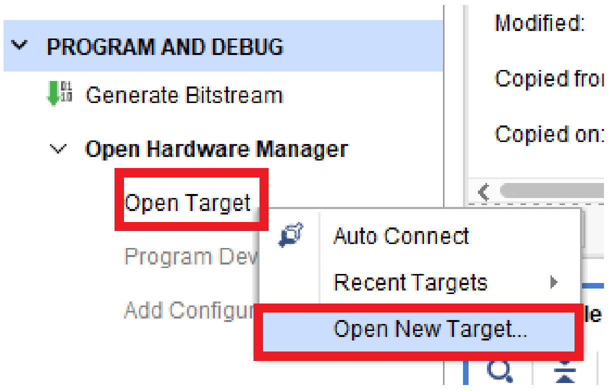
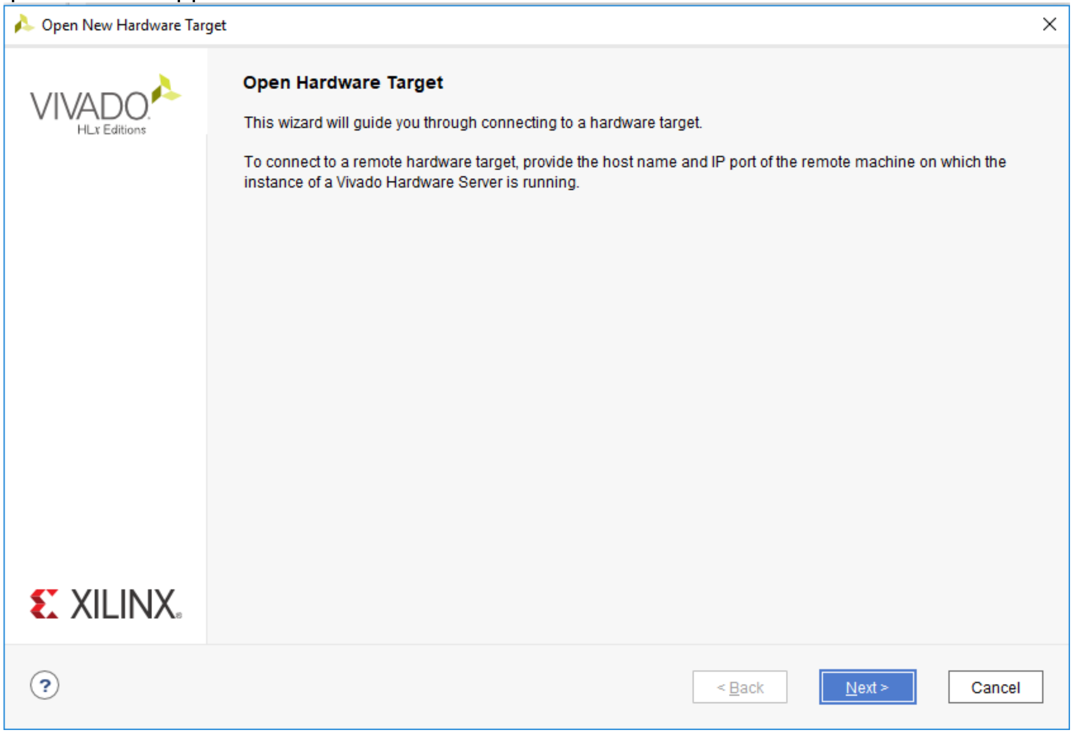


Figure 29 – Open New Target

A setup wizard will appear. Click “Next”.



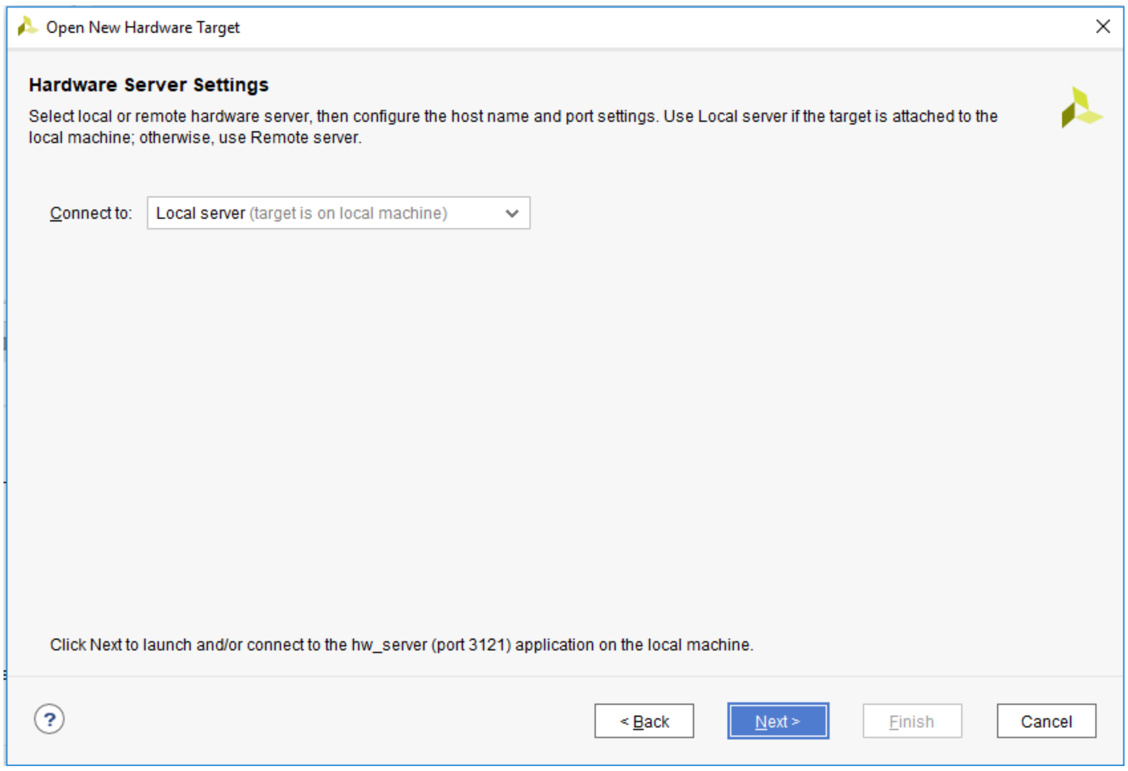
Open Hardware Target

This wizard will guide you through connecting to a hardware target.

To connect to a remote hardware target, provide the host name and IP port of the remote machine on which the instance of a Vivado Hardware Server is running.

Figure 30 - Open Hardware Target

Select “Local server” if the target device is connected to the local machine. Click “Next”.


Open New Hardware Target

Hardware Server Settings

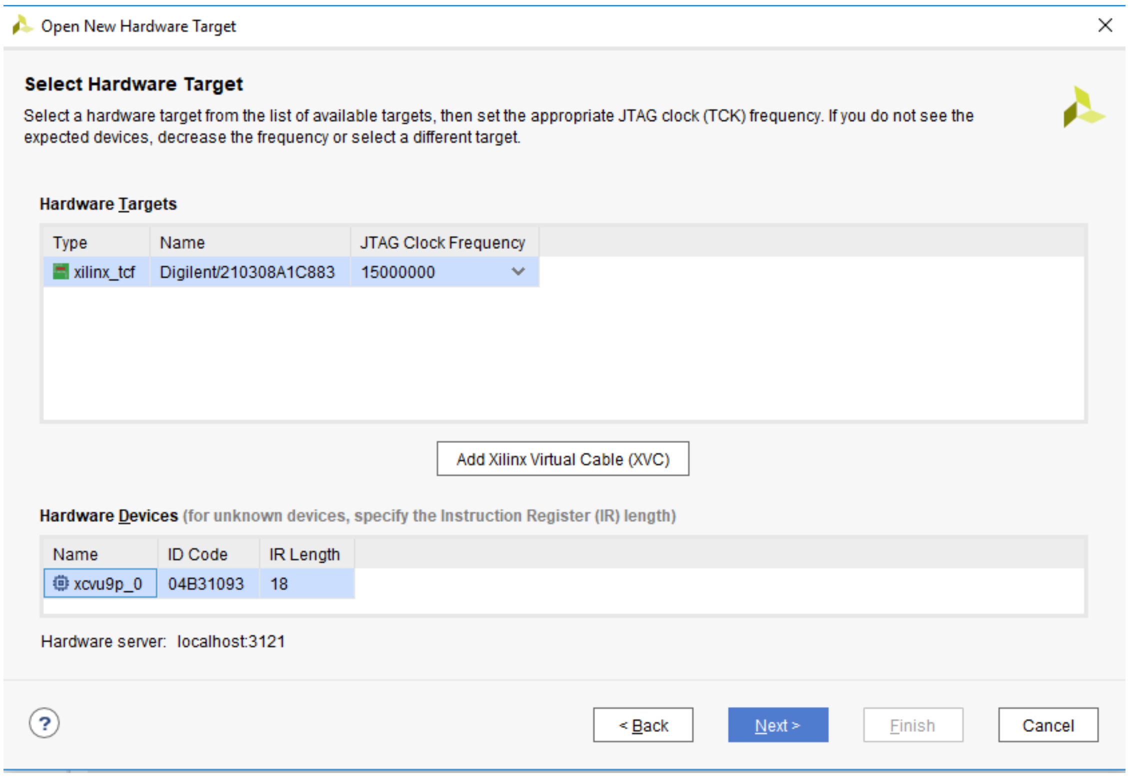
Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local machine; otherwise, use Remote server.

Connect to: Local server (target is on local machine)

Click Next to launch and/or connect to the hw_server (port 3121) application on the local machine.

Figure 31 - Hardware Server Settings

Check if the correct device target is shown under “Hardware Devices”. Click “Next”.


Open New Hardware Target

Select 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.

Hardware Targets

Type	Name	JTAG Clock Frequency
xilinx_tcf	Digilent/210308A1C883	15000000

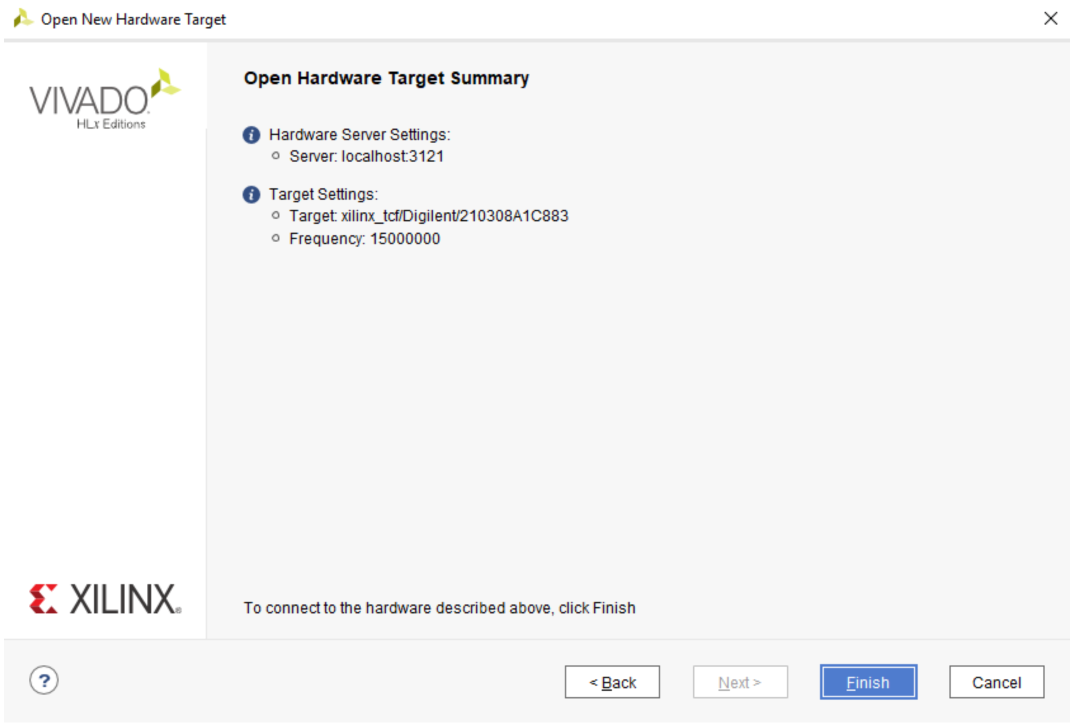
Hardware Devices (for unknown devices, specify the Instruction Register (IR) length)

Name	ID Code	IR Length
xcvu9p_0	04B31093	18

Hardware server: localhost:3121

Figure 32 - Select Hardware Target

A summary page will appear. Click “Finish”.


Open New Hardware Target

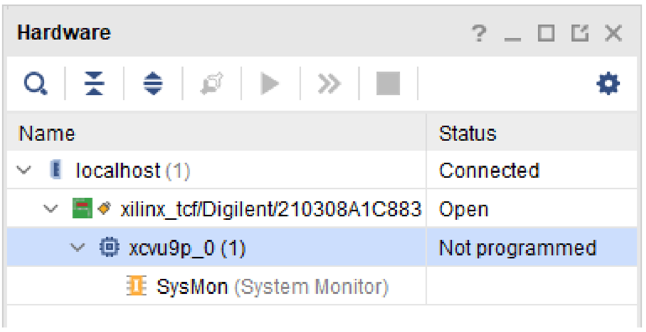
Open Hardware Target Summary

- Hardware Server Settings:
  - Server: localhost:3121
- Target Settings:
  - Target: xilinx_tcf/Digilent/210308A1C883
  - Frequency: 15000000

To connect to the hardware described above, click Finish

Figure 33 - Open hardware target summary

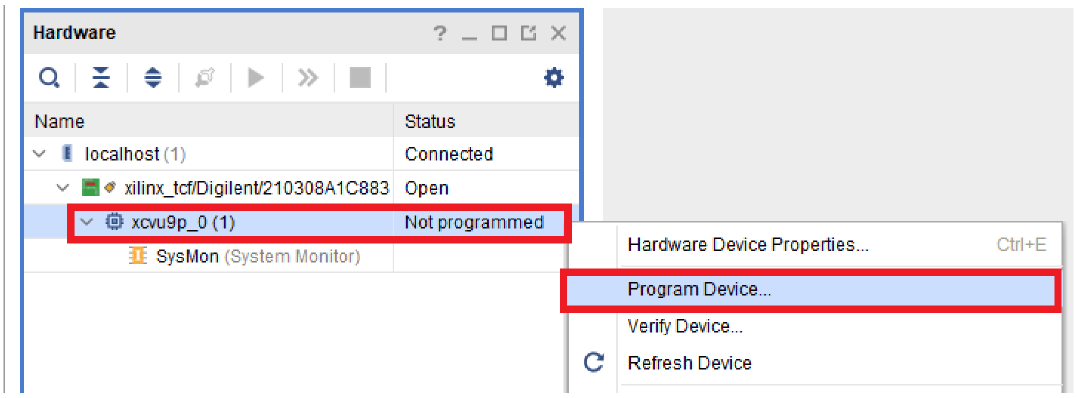
Select “xcu9p_0” in the Hardware window.


Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210308A1C883	Open
xcvu9p_0 (1)	Not programmed
SysMon (System Monitor)	

Figure 34 - Hardware window

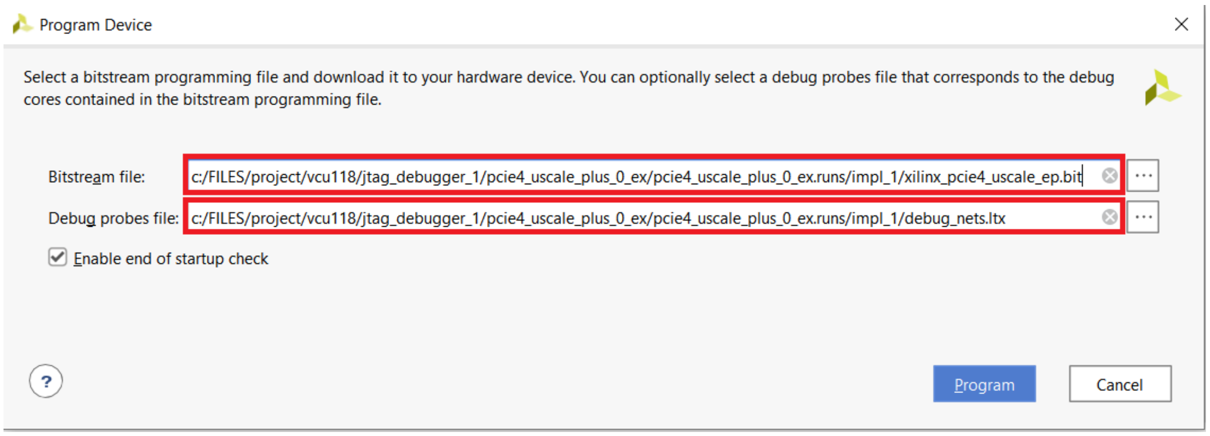
Right-click on it to program the device.


Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210308A1C883	Open
xcvu9p_0 (1)	Not programmed
SysMon (System Monitor)	

Figure 35 - Program device

Make sure the correct “.bit” and “.ltx” files are selected. Click “Program”.


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.
Program
Cancel

Bitstream file: c:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/xilinx_pcie4_uscale_ep.bit

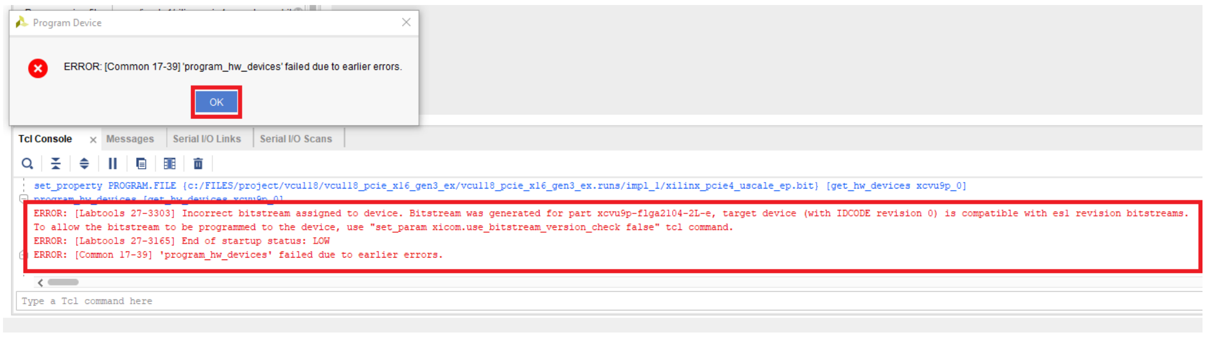
Debug probes file: c:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/debug_nets.ltx

[✓] Enable end of startup check

Figure 36 - Bitstream file and debug probe file

An expected error will be shown in the Tcl Console. Enter the following command to resolve the error:

`set_param xicom.use_bitstream_version_check false`

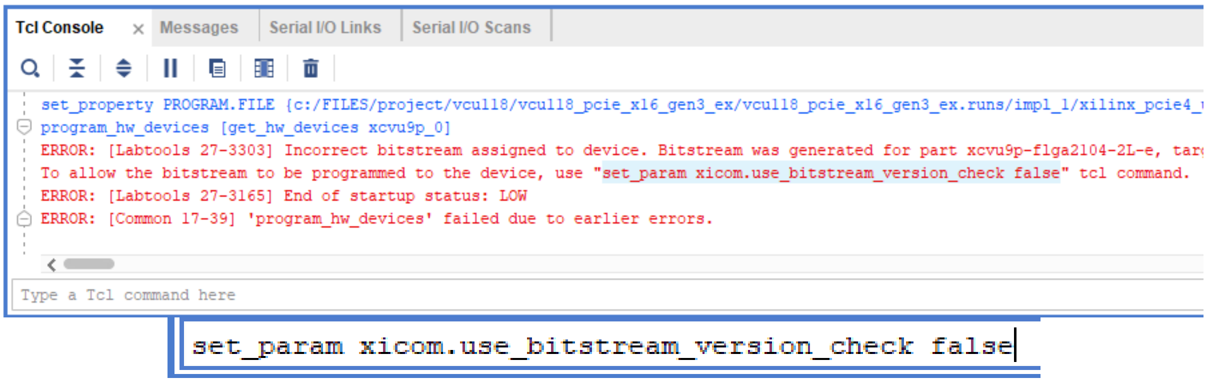


ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.

```
set_property PROGRAM.FILE {c:/FILES/project/vcu118/vcu118_pcie_x16_gen3_ex/vcu118_pcie_x16_gen3_ex.runs/impl_1/xilinx_pcie4_uscale_ep.bit} [get_hw_devices xcvu9p_0]
program_hw_devices [get_hw_devices xcvu9p_0]
ERROR: [Labtools 27-3303] Incorrect bitstream assigned to device. Bitstream was generated for part xcvu9p-flga2104-2L-e, target device (with IDCODE revision 0) is compatible with es1 revision bitstreams.
To allow the bitstream to be programmed to the device, use "set_param xicom.use_bitstream_version_check false" tcl command.
ERROR: [Labtools 27-3165] End of startup status: LOW
ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.
```

Figure 37 - set_param xicom.use_bitstream_version_check false

The Tcl command is shown in the figure below.

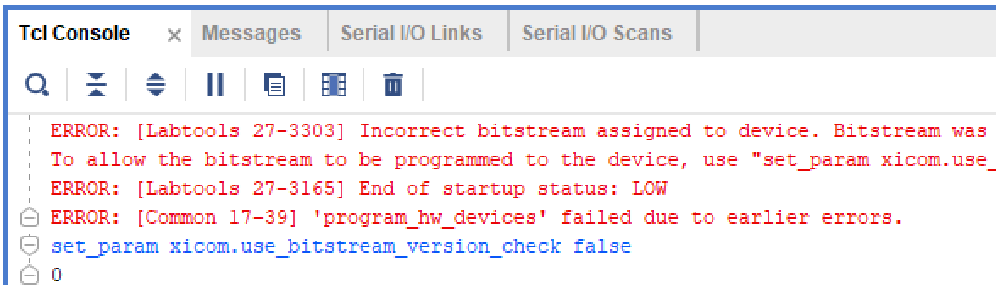


```
set_property PROGRAM.FILE {c:/FILES/project/vcu118/vcu118_pcie_x16_gen3_ex/vcu118_pcie_x16_gen3_ex.runs/impl_1/xilinx_pcie4_
program_hw_devices [get_hw_devices xcvu9p_0]
ERROR: [Labtools 27-3303] Incorrect bitstream assigned to device. Bitstream was generated for part xcvu9p-flga2104-2L-e, tar
To allow the bitstream to be programmed to the device, use "set_param xicom.use_bitstream_version_check false" tcl command.
ERROR: [Labtools 27-3165] End of startup status: LOW
ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.
```

```
set_param xicom.use_bitstream_version_check false
```

Figure 38 - Tcl command

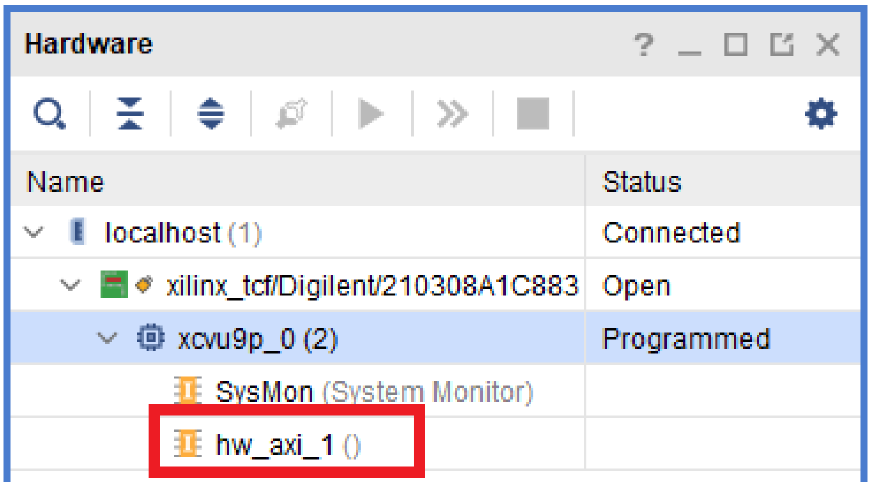
The Tcl Console window will activate the Tcl command as shown below.



```
ERROR: [Labtools 27-3303] Incorrect bitstream assigned to device. Bitstream was
To allow the bitstream to be programmed to the device, use "set_param xicom.use_
ERROR: [Labtools 27-3165] End of startup status: LOW
ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.
set_param xicom.use_bitstream_version_check false
0
```

Figure 39 - Tcl console

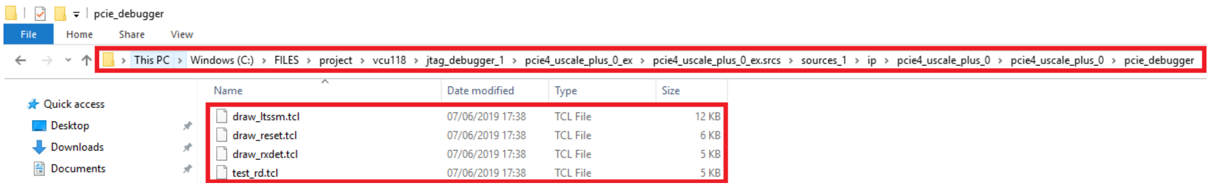
Reprogram the device. After successfully programming the target device, an AXI core "hw_axi_1" should appear in the hardware window.



Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210308A1C883	Open
xcvu9p_0 (2)	Programmed
SysMon (System Monitor)	
hw_axi_1 ()	

Figure 40 - Programmed device with hw_axi_1

Locate the Tcl file "test_rd.tcl" inside the example project. See the example path below.

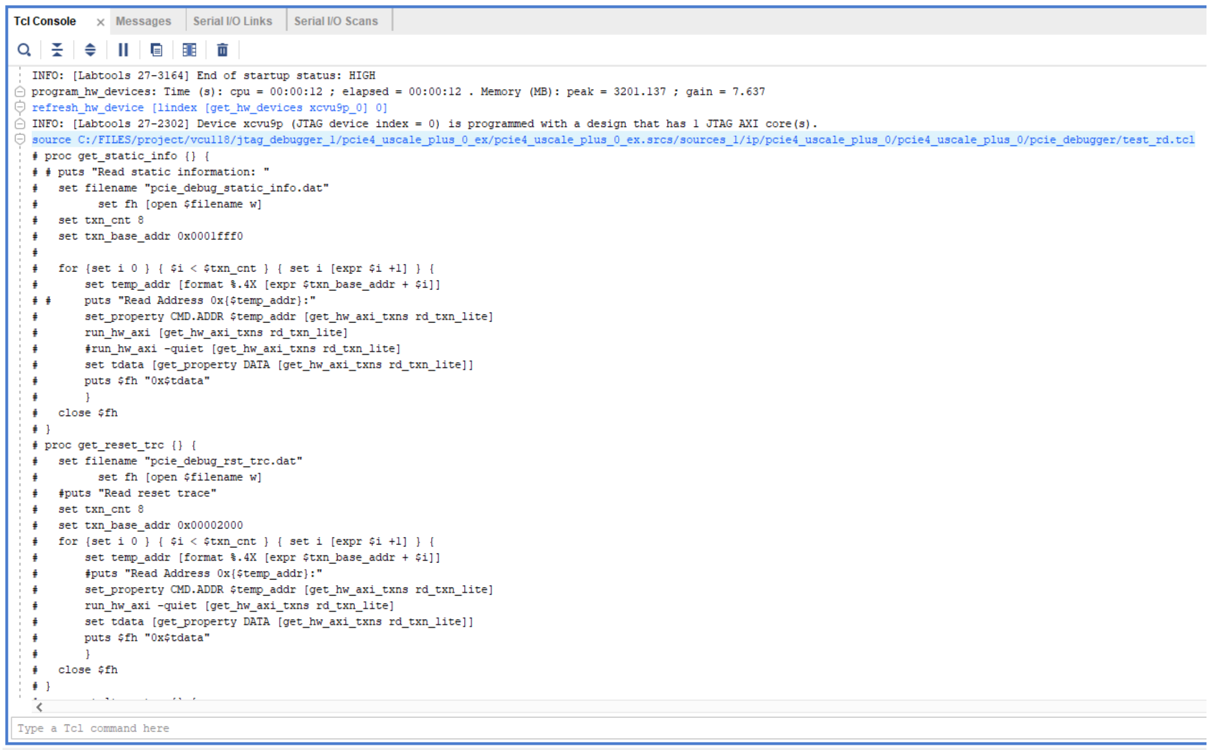

pcie_debugger
File
Home
Share
View
This PC › Windows (C:) › FILES › project › vcu118 › jtag_debugger_1 › pcie4_uscale_plus_0_ex › pcie4_uscale_plus_0_ex.srcs › sources_1 › ip › pcie4_uscale_plus_0 › pcie4_uscale_plus_0 › pcie_debugger
Quick access
Desktop
Downloads
Documents

Name	Date modified	Type	Size
draw_ltssm.tcl	07/06/2019 17:38	TCL File	12 KB
draw_reset.tcl	07/06/2019 17:38	TCL File	6 KB
draw_rxdet.tcl	07/06/2019 17:38	TCL File	5 KB
test_rd.tcl	07/06/2019 17:38	TCL File	5 KB

Figure 41 - Tcl file test_rdl

Source the "test_rd.tcl" file in the Tcl Console. This command is used to read in data stored in BRAM through the Tcl interface. The Tcl command is shown below.

```
Source
C:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.srcs/sou
rces_1/ip/pcie4_uscale_plus_0/pcie4_uscale_plus_0/pcie_debugger/test_rd.tcl
```

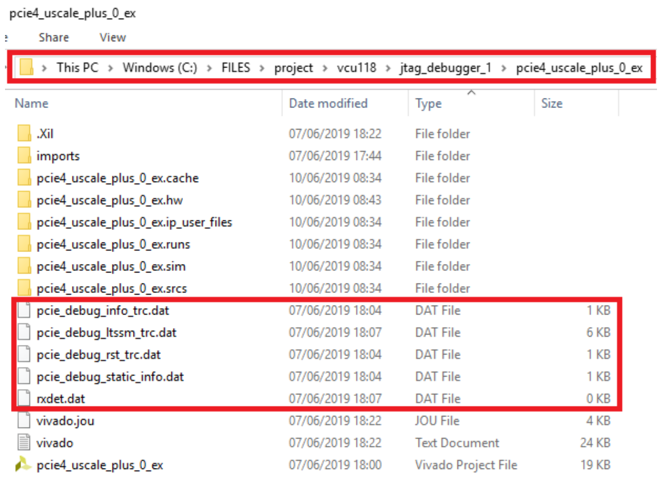


```
INFO: [Labtools 27-3164] End of startup status: HIGH
program_hw_devices: Time (s): cpu = 00:00:12 ; elapsed = 00:00:12 . Memory (MB): peak = 3201.137 ; gain = 7.637
refresh_hw_device [lindex [get_hw_devices xcvu9p_0] 0]
INFO: [Labtools 27-2302] Device xcvu9p (JTAG device index = 0) is programmed with a design that has 1 JTAG AXI core(s).
source C:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.srcs/sources_1/ip/pcie4_uscale_plus_0/pcie4_uscale_plus_0/pcie_debugger/test_rd.tcl
# proc get_static_info {} {
# # puts "Read static information: "
#   set filename "pcie_debug_static_info.dat"
#         set fh [open $filename w]
#   set txn_cnt 8
#   set txn_base_addr 0x0001fff0
#
#   for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#       set temp_addr [format %.4X [expr $txn_base_addr + $i]]
# #     puts "Read Address 0x{$temp_addr}:"
#       set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#       run_hw_axi [get_hw_axi_txns rd_txn_lite]
#       #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"
#       }
#   close $fh
# }
# proc get_reset_trc {} {
#   set filename "pcie_debug_rst_trc.dat"
#         set fh [open $filename w]
#   #puts "Read reset trace"
#   set txn_cnt 8
#   set txn_base_addr 0x00002000
#   for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#       set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#       #puts "Read Address 0x{$temp_addr}:"
#       set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#       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"
#       }
#   close $fh
# }
Type a Tcl command here
```

Figure 42 - Tcl console result of sourcing test_rd.tcl

After sourcing the test_rd.tcl file, it will generate the following set of ".DAT" files. See Appendix A: Tcl Console Result of test_rd.tcl for the complete Tcl console results of a test_rd.tcl file.

- `pcie_debug_static_info.dat`
- `pcie_debug_rst_trc.dat`
- `pcie_debug_ltssm_trc.dat`
- `pcie_debug_rxdet_trc.dat`
- `pcie_debug_info_trc.dat`

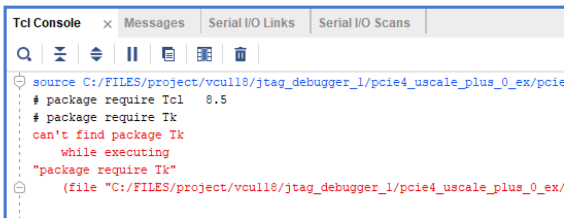


pcie4_uscale_plus_0_ex
Share
View
This PC › Windows (C:) › FILES › project › vcu118 › jtag_debugger_1 › pcie4_uscale_plus_0_ex

Name	Date modified	Type	Size
.Xil	07/06/2019 18:22	File folder	
imports	07/06/2019 17:44	File folder	
pcie4_uscale_plus_0_ex.cache	10/06/2019 08:34	File folder	
pcie4_uscale_plus_0_ex.hw	10/06/2019 08:43	File folder	
pcie4_uscale_plus_0_ex.ip_user_files	10/06/2019 08:34	File folder	
pcie4_uscale_plus_0_ex.runs	10/06/2019 08:34	File folder	
pcie4_uscale_plus_0_ex.sim	10/06/2019 08:34	File folder	
pcie4_uscale_plus_0_ex.srcs	10/06/2019 08:34	File folder	
pcie_debug_info_trc.dat	07/06/2019 18:04	DAT File	1 KB
pcie_debug_ltssm_trc.dat	07/06/2019 18:07	DAT File	6 KB
pcie_debug_rst_trc.dat	07/06/2019 18:04	DAT File	1 KB
pcie_debug_static_info.dat	07/06/2019 18:04	DAT File	1 KB
rxdet.dat	07/06/2019 18:07	DAT File	0 KB
vivado.jou	07/06/2019 18:22	JOU File	4 KB
vivado	07/06/2019 18:22	Text Document	24 KB
pcie4_uscale_plus_0_ex	07/06/2019 18:00	Vivado Project File	19 KB

Figure 43 - Generated DAT files

Source "draw_ltssm.tcl" to capture the LTSSM state diagram. An expected error will occur that requires the user to install Tcl/Tk packages. See Appendix B: Tcl/Tk Package Installation Guide.

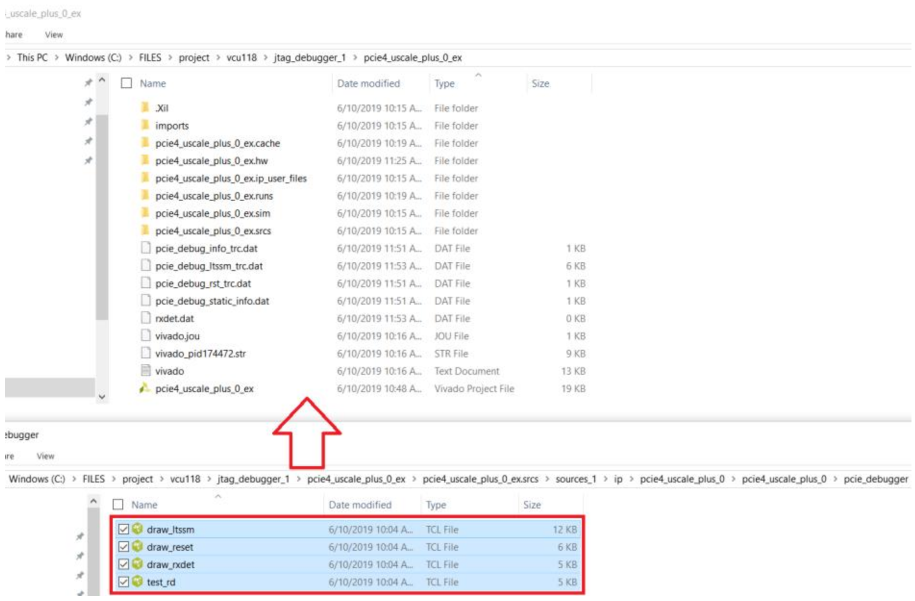


Tcl Console | Messages | Serial I/O Links | Serial I/O Scans

```
source C:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie
# package require Tcl   8.5
# package require Tk
can't find package Tk
    while executing
"package require Tk"
    (file "C:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/
```

Figure 44 - Tcl/Tk package error

Copy the Tcl files from **pcie_debugger folder** into the “pcie_uscale_plus_0_ex” project folder. All the generated **DAT** files and PCIe debug **Tcl** files must be in one location.



This PC > Windows (C:) > FILES > project > vcu118 > jtag_debugger_1 > pcie4_uscale_plus_0_ex

Name	Date modified	Type	Size
.Xil	6/10/2019 10:15 A...	File folder	
imports	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.cache	6/10/2019 10:19 A...	File folder	
pcie4_uscale_plus_0_ex.hw	6/10/2019 11:25 A...	File folder	
pcie4_uscale_plus_0_ex.ip_user_files	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.runs	6/10/2019 10:19 A...	File folder	
pcie4_uscale_plus_0_ex.sim	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.srcs	6/10/2019 10:15 A...	File folder	
pcie_debug_info_trc.dat	6/10/2019 11:51 A...	DAT File	1 KB
pcie_debug_ltssm_trc.dat	6/10/2019 11:53 A...	DAT File	6 KB
pcie_debug_rst_trc.dat	6/10/2019 11:51 A...	DAT File	1 KB
pcie_debug_static_info.dat	6/10/2019 11:51 A...	DAT File	1 KB
rxdet.dat	6/10/2019 11:53 A...	DAT File	0 KB
vivado.jou	6/10/2019 10:16 A...	JOU File	1 KB
vivado_pid174472.str	6/10/2019 10:16 A...	STR File	9 KB
vivado	6/10/2019 10:16 A...	Text Document	13 KB
pcie4_uscale_plus_0_ex	6/10/2019 10:48 A...	Vivado Project File	19 KB

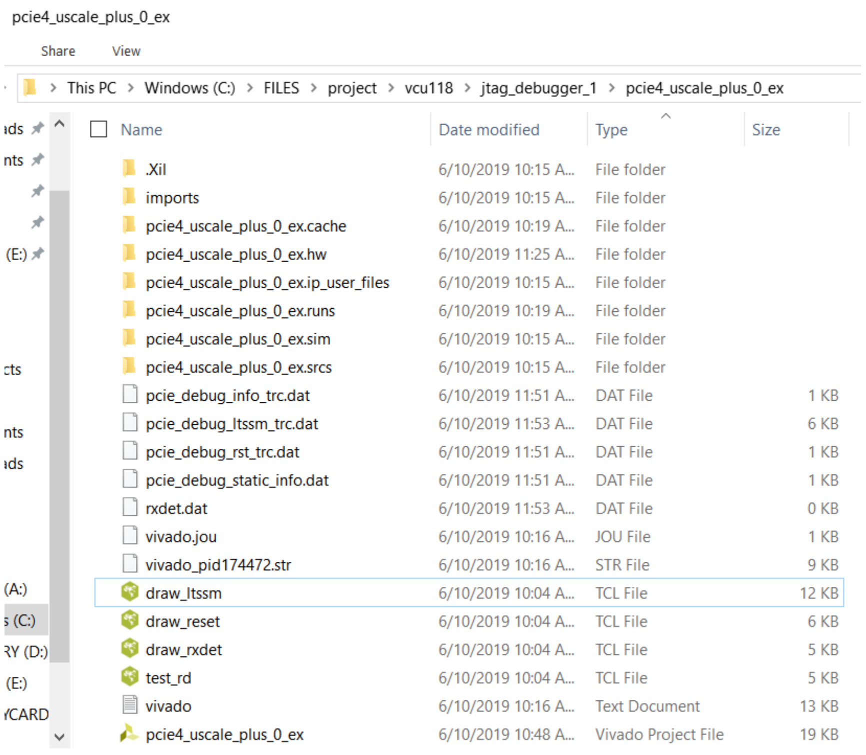
Windows (C:) > FILES > project > vcu118 > jtag_debugger_1 > pcie4_uscale_plus_0_ex > pcie4_uscale_plus_0_ex.srcs > sources_1 > ip > pcie4_uscale_plus_0 > pcie4_uscale_plus_0 > pcie_debugger

Name	Date modified	Type	Size
draw_ltssm	6/10/2019 10:04 A...	TCL File	12 KB
draw_reset	6/10/2019 10:04 A...	TCL File	6 KB
draw_rxdet	6/10/2019 10:04 A...	TCL File	5 KB
test_rd	6/10/2019 10:04 A...	TCL File	5 KB

Figure 45 - Add JTAG debug Tcl files

Double click on each PCIe debugger Tcl files to generate a diagram:

- draw_ltssm.tcl
- draw_reset.tcl
- draw_rxdet.tcl



pcie4_uscale_plus_0_ex

Share View

This PC > Windows (C:) > FILES > project > vcu118 > jtag_debugger_1 > pcie4_uscale_plus_0_ex

Name	Date modified	Type	Size
.Xil	6/10/2019 10:15 A...	File folder	
imports	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.cache	6/10/2019 10:19 A...	File folder	
pcie4_uscale_plus_0_ex.hw	6/10/2019 11:25 A...	File folder	
pcie4_uscale_plus_0_ex.ip_user_files	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.runs	6/10/2019 10:19 A...	File folder	
pcie4_uscale_plus_0_ex.sim	6/10/2019 10:15 A...	File folder	
pcie4_uscale_plus_0_ex.srcs	6/10/2019 10:15 A...	File folder	
pcie_debug_info_trc.dat	6/10/2019 11:51 A...	DAT File	1 KB
pcie_debug_ltssm_trc.dat	6/10/2019 11:53 A...	DAT File	6 KB
pcie_debug_rst_trc.dat	6/10/2019 11:51 A...	DAT File	1 KB
pcie_debug_static_info.dat	6/10/2019 11:51 A...	DAT File	1 KB
rxdet.dat	6/10/2019 11:53 A...	DAT File	0 KB
vivado.jou	6/10/2019 10:16 A...	JOU File	1 KB
vivado_pid174472.str	6/10/2019 10:16 A...	STR File	9 KB
draw_ltssm	6/10/2019 10:04 A...	TCL File	12 KB
draw_reset	6/10/2019 10:04 A...	TCL File	6 KB
draw_rxdet	6/10/2019 10:04 A...	TCL File	5 KB
test_rd	6/10/2019 10:04 A...	TCL File	5 KB
vivado	6/10/2019 10:16 A...	Text Document	13 KB
pcie4_uscale_plus_0_ex	6/10/2019 10:48 A...	Vivado Project File	19 KB

Figure 46 - Debug Tcl files and generated DAT files in one directory

Generated LTSSM diagram from the "draw_ltssm.tcl" file:

- Green color – transitioned state during the capture window
- Orange color – last state
- Red arrow – last transition state
- Numbers beside the arrow – indicates the number of times the transition happened between the two states

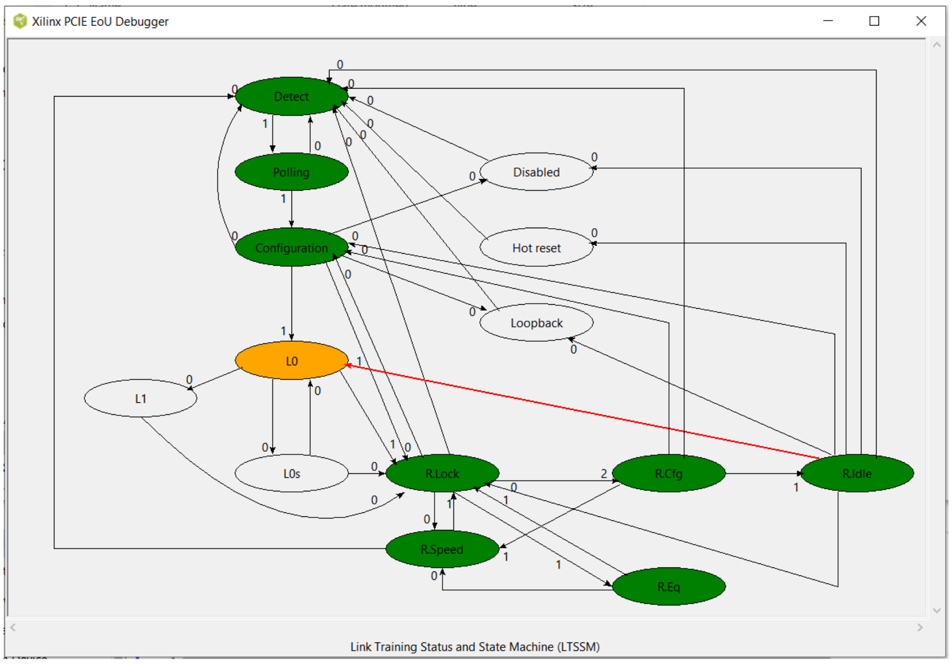


Figure 47 - LTSSM diagram

Xilinx PCIe In-system Debugger for Reset Sequence from the "draw_reset.tcl" file:

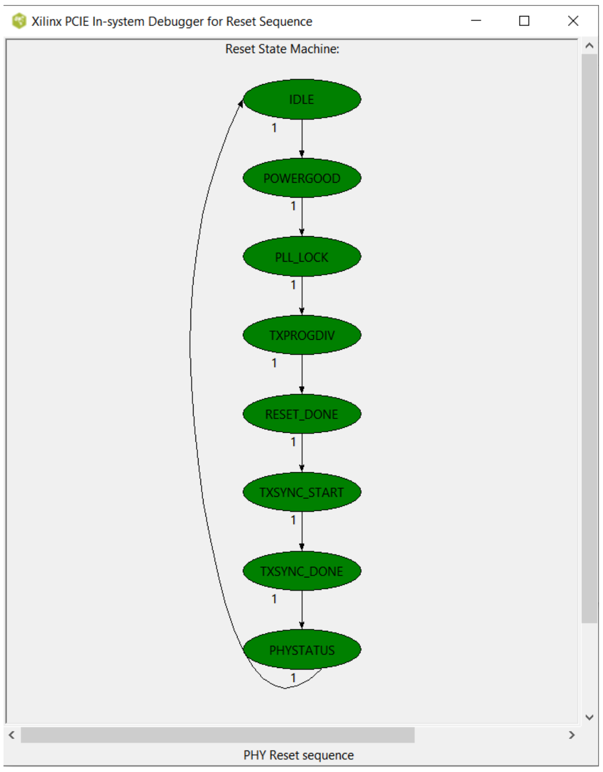
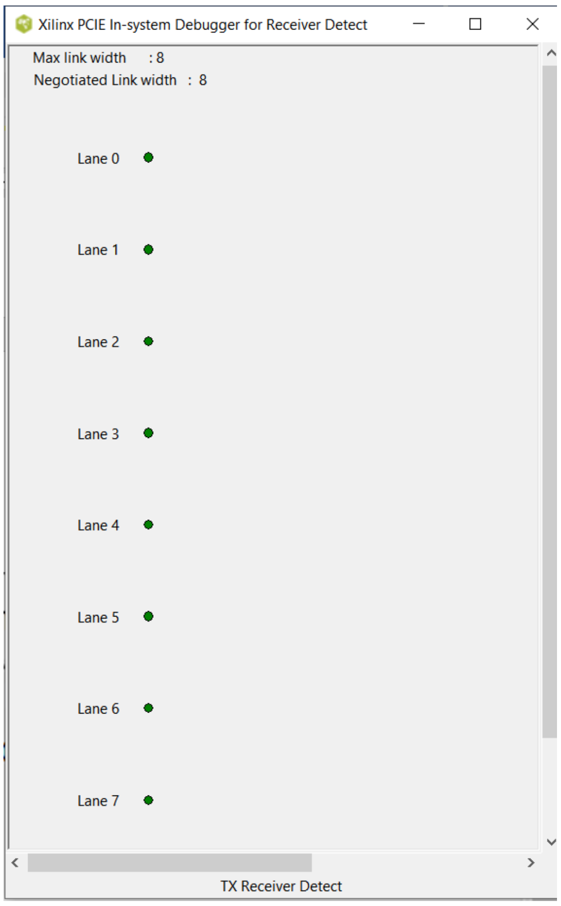


Figure 48 - Reset sequence

Xilinx PCIe In-system Debugger for Receiver Detect from “draw_rxdet.tcl” file:



Max link width : 8
Negotiated Link width : 8

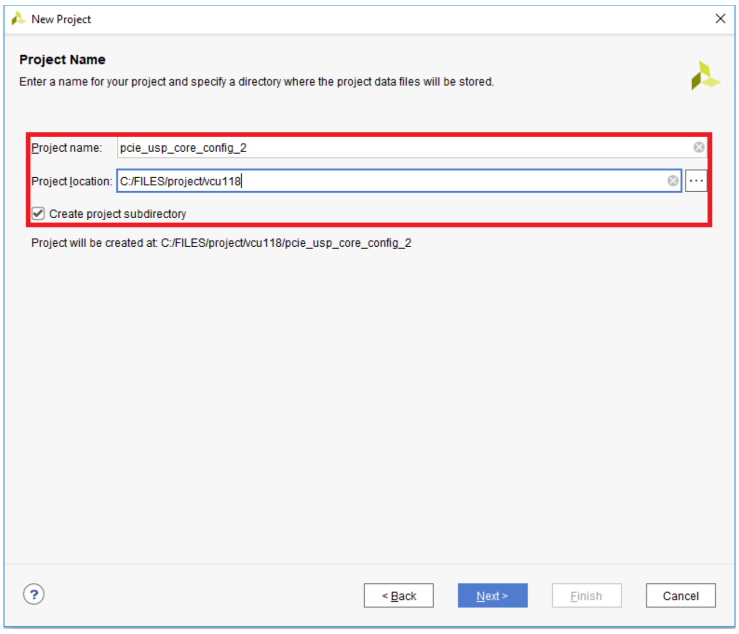
Lane 0 ●
Lane 1 ●
Lane 2 ●
Lane 3 ●
Lane 4 ●
Lane 5 ●
Lane 6 ●
Lane 7 ●

Figure 49 - Receiver detect

In System IBERT

Another debug feature available in UltraScale+ PCIe cores is performing a full 2D Eye Scan limited to user raw data only. To obtain an eye scan, the in-system Integrated Bit Error Ratio Tester (IBERT) must be enabled. The following is a step-by-step guide in obtaining an eye diagram. Please refer to some of the steps in “General Design Steps in Configuring the PCIe Core”.

Create a new project. Provide a project name and location for configuring the PCIe core IP catalog.



New Project

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: pcie_usp_core_config_2

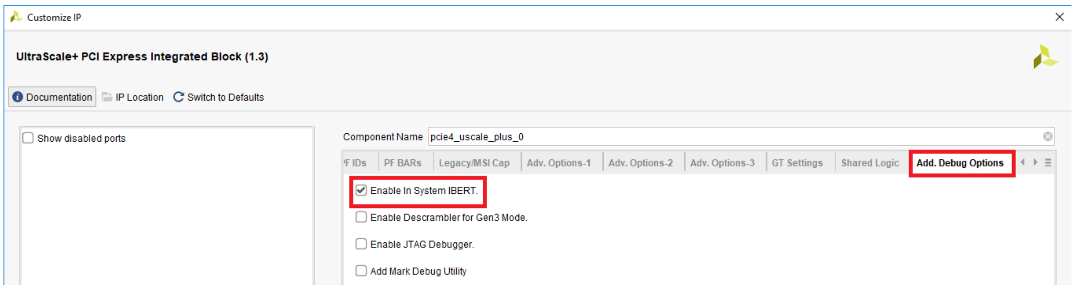
Project location: C:/FILES/project/vcu118

✓ Create project subdirectory

Project will be created at: C:/FILES/project/vcu118/pcie_usp_core_config_2

Figure 50 - Project Name

In the “Add. Debug Options” tick the “Enable In System IBERT” check box.



Customize IP

UltraScale+ PCI Express Integrated Block (1.3)

Component Name: pcie4_uscale_plus_0

✓ Enable In System IBERT.

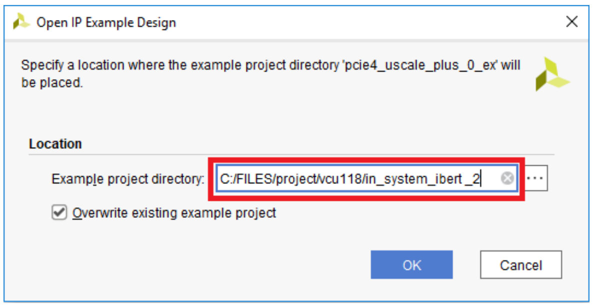
☐ Enable Descrambler for Gen3 Mode.

☐ Enable JTAG Debugger.

☐ Add Mark Debug Utility

Figure 51 - Enable IBERT

Open and provide a location for the IP example design.



Open IP Example Design

Specify a location where the example project directory 'pcie4_uscale_plus_0_ex' will be placed.

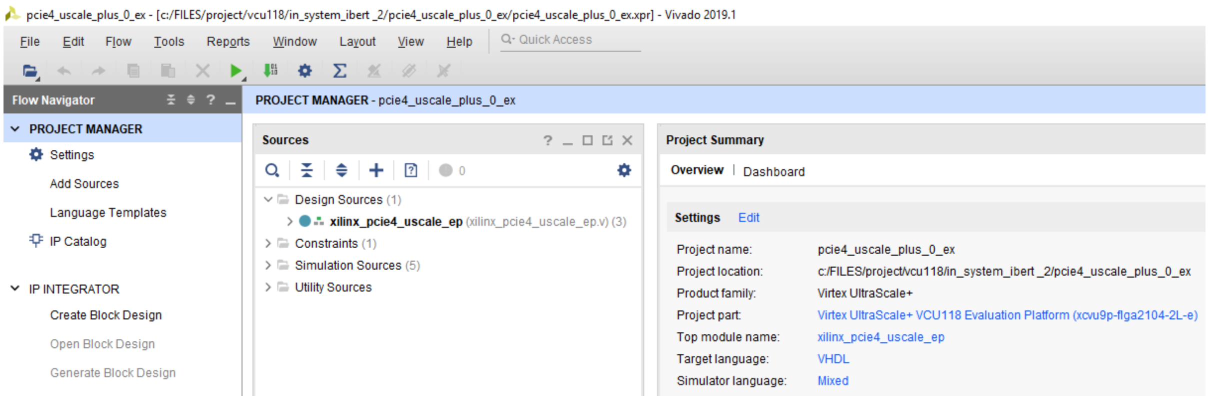
Location

Example project directory: C:/FILES/project/vcu118/in_system_ibert_2

✓ Overwrite existing example project

Figure 52 - IBERT IP example

A new Vivado project containing the IP example design will initialize.

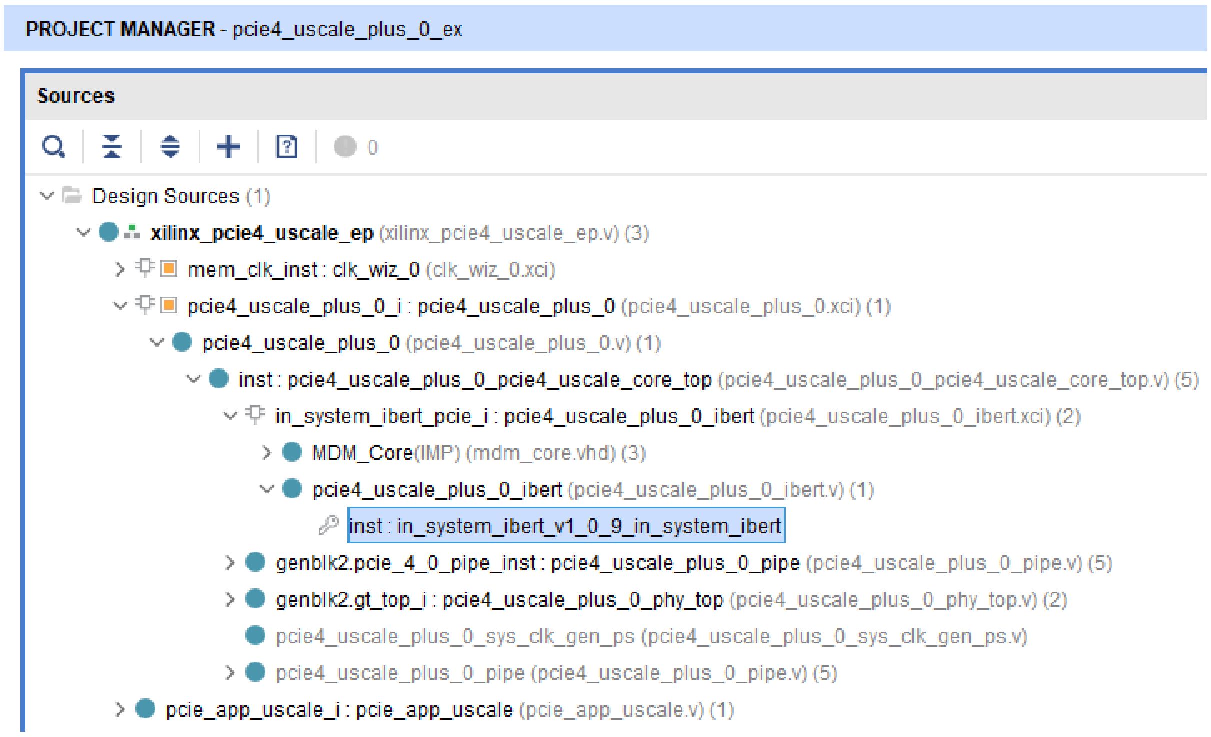

pcie4_uscale_plus_0_ex - [c:/FILES/project/vcu118/in_system_ibert _2/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
PROJECT MANAGER - pcie4_uscale_plus_0_ex
Sources
Project Summary
Overview
Dashboard
Settings
Edit

```
Design Sources (1)
  xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
Constraints (1)
Simulation Sources (5)
Utility Sources
```

Project name:	pcie4_uscale_plus_0_ex
Project location:	c:/FILES/project/vcu118/in_system_ibert _2/pcie4_uscale_plus_0_ex
Product family:	Virtex UltraScale+
Project part:	Virtex UltraScale+ VCU118 Evaluation Platform (xcvu9p-flga2104-2L-e)
Top module name:	xilinx_pcie4_uscale_ep
Target language:	VHDL
Simulator language:	Mixed

Figure 53 - Example IP project

The Design sources hierarchy must include the instantiation of In System IBERT.



```
Design Sources (1)
  xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
    mem_clk_inst : clk_wiz_0 (clk_wiz_0.xci)
    pcie4_uscale_plus_0_i : pcie4_uscale_plus_0 (pcie4_uscale_plus_0.xci) (1)
      pcie4_uscale_plus_0 (pcie4_uscale_plus_0.v) (1)
        inst : pcie4_uscale_plus_0_pcie4_uscale_core_top (pcie4_uscale_plus_0_pcie4_uscale_core_top.v) (5)
          in_system_ibert_pcie_i : pcie4_uscale_plus_0_ibert (pcie4_uscale_plus_0_ibert.xci) (2)
            MDM_Core(IMP) (mdm_core.vhd) (3)
            pcie4_uscale_plus_0_ibert (pcie4_uscale_plus_0_ibert.v) (1)
              inst : in_system_ibert_v1_0_9_in_system_ibert
          genblk2.pcie_4_0_pipe_inst : pcie4_uscale_plus_0_pipe (pcie4_uscale_plus_0_pipe.v) (5)
          genblk2.gt_top_i : pcie4_uscale_plus_0_phy_top (pcie4_uscale_plus_0_phy_top.v) (2)
          pcie4_uscale_plus_0_sys_clk_gen_ps (pcie4_uscale_plus_0_sys_clk_gen_ps.v)
          pcie4_uscale_plus_0_pipe (pcie4_uscale_plus_0_pipe.v) (5)
    pcie_app_uscale_i : pcie_app_uscale (pcie_app_uscale.v) (1)
```

Figure 54 - Design sources hierarchy

In updating the constraint files, see the following figures:

- Figure 20 - Edit constraint file
- Figure 21 – Activate reset pin
- Figure 22 - Adding lines of code

Run synthesis in the flow navigator window.

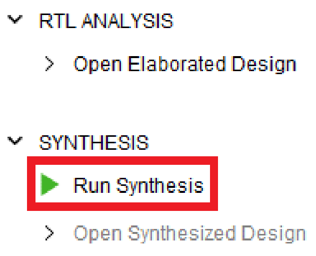


Figure 55 - Synthesis

Generate Bitstream from the flow navigator.

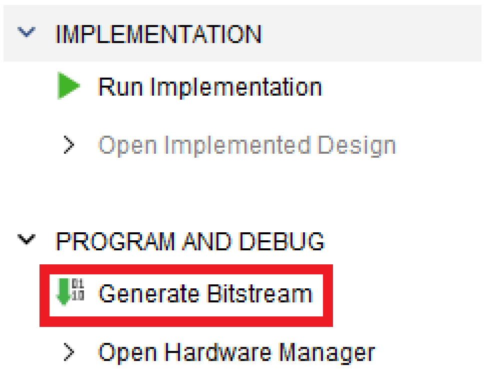
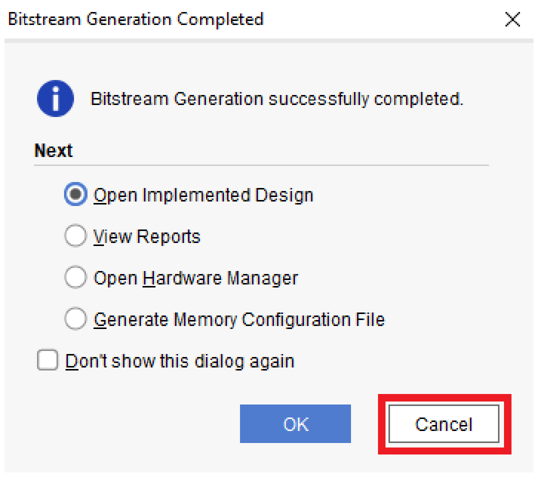


Figure 56 - Generate bitstream

Click "Cancel".



Bitstream Generation Completed

Bitstream Generation successfully completed.

Next

[✓ Open Implemented Design]
View Reports
Open Hardware Manager
Generate Memory Configuration File

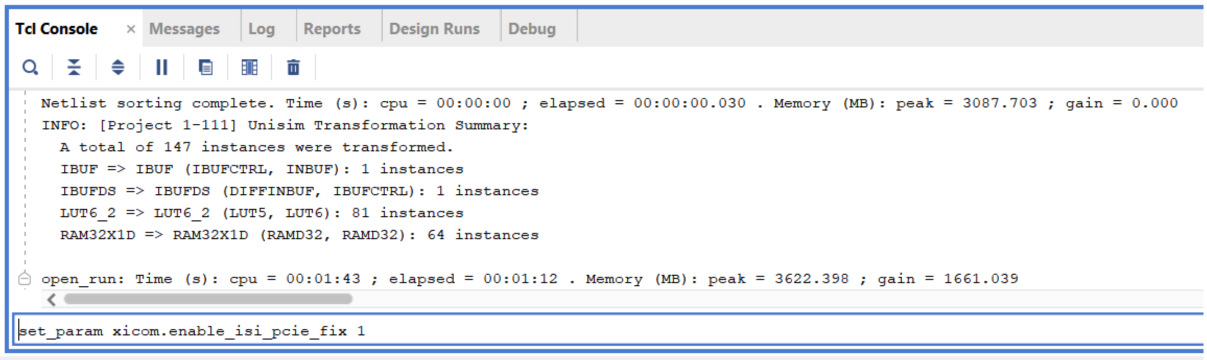
☐ Don't show this dialog again

OK Cancel

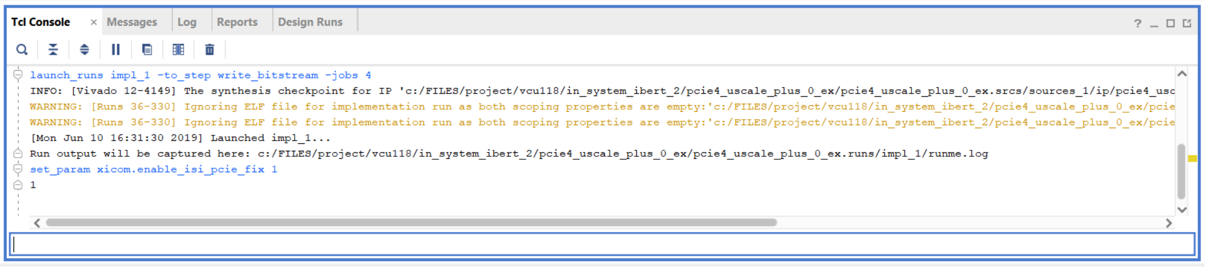
Figure 57 - Bitstream complete

Run the below Tcl command:

```
Set_param xicom.enable_isi_pcie_fix 1
```



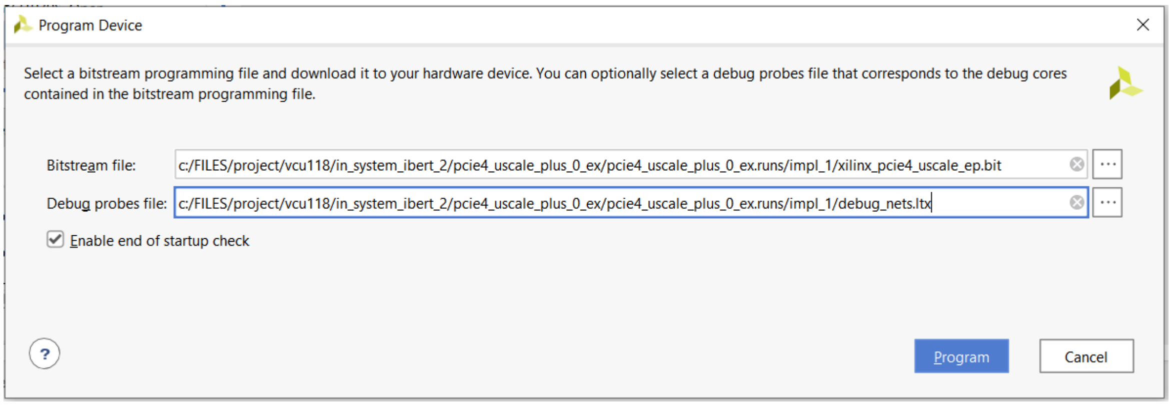
```
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.030 . Memory (MB): peak = 3087.703 ; gain = 0.000
INFO: [Project 1-111] Unisim Transformation Summary:
  A total of 147 instances were transformed.
  IBUF => IBUF (IBUFCTRL, INBUF): 1 instances
  IBUFDS => IBUFDS (DIFFINBUF, IBUFCTRL): 1 instances
  LUT6_2 => LUT6_2 (LUT5, LUT6): 81 instances
  RAM32X1D => RAM32X1D (RAMD32, RAMD32): 64 instances

open_run: Time (s): cpu = 00:01:43 ; elapsed = 00:01:12 . Memory (MB): peak = 3622.398 ; gain = 1661.039

set_param xicom.enable_isi_pcie_fix 1
```



```
launch_runs impl_1 -to_step write_bitstream -jobs 4
INFO: [Vivado 12-4149] The synthesis checkpoint for IP 'c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.srcs/sources_1/ip/pcie4_usc
WARNING: [Runs 36-330] Ignoring ELF file for implementation run as both scoping properties are empty:'c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie
WARNING: [Runs 36-330] Ignoring ELF file for implementation run as both scoping properties are empty:'c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie
[Mon Jun 10 16:31:30 2019] Launched impl_1...
Run output will be captured here: c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/runme.log
set_param xicom.enable_isi_pcie_fix 1
1
```

Figure 58 - Tcl command set_param

Program the device using the correct bitstream file and debug probe file.



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.

Bitstream file: c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/xilinx_pcie4_uscale_ep.bit

Debug probes file: c:/FILES/project/vcu118/in_system_ibert_2/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/debug_nets.ltx

☑ Enable end of startup check

Figure 59 - Program device

If the errors **Xicom 50-38** & **Labtools 27-3176** occur, connect the endpoint device to a host computer to supply a reference clock.

```
WARNING: [Xicom 50-99] Incorrect bitstream assigned to device. Bitstream was generated for part xcvu9p-flga2104-2L-e, target device (with IDCODE revision 0) is compatible
INFO: [Labtools 27-3164] End of startup status: HIGH
program_hw_devices: Time (s): cpu = 00:00:12 ; elapsed = 00:00:12 . Memory (MB): peak = 2101.004 ; gain = 14.258
refresh_hw_device [lindex [get_hw_devices xcvu9p_0] 0]
WARNING: [Xicom 50-38] xicom: No CseXsdb register file specified for CseXsdb slave type: 0, cse driver version: 0. Slave initialization skipped.
WARNING: [Xicom 50-38] xicom: No CseXsdb register file specified for CseXsdb slave type: 0, cse driver version: 0. Slave initialization skipped.
ERROR: [Xicom 50-38] xicom: Device:0, user chain number:1, slave index:2. Reading intermittently wrong data from core. Try slower target speed. Make sure design meets timi
ERROR: [Xicom 50-38] xicom: Device:0, user chain number:1, slave index:2, is not a valid CseXsdb Slave core.
ERROR: [Labtools 27-3176] hw_server failed during internal command.
Resolution: Check that the hw_server is running and the hardware connectivity to the target
```

Figure 60 – Error Xicom 50-38 & Labtools 27-3176

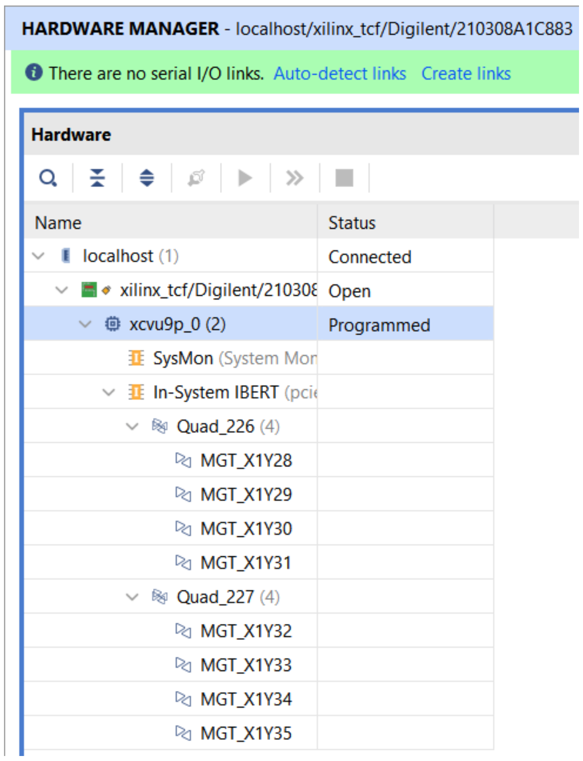
If the errors **Labtools 27-3303** & **Common 17-39** occur, enter the Tcl command below:

```
set_param xicom.use_bitstream_version_check false
```

```
ERROR: [Labtools 27-3303] Incorrect bitstream assigned to device. Bitstream was generated for part xcvu9p-flga2104-2L-e, target device (with IDCODE revision 0) is compatible with es1 revision bitstreams.
To allow the bitstream to be programmed to the device, use "set_param xicom.use_bitstream_version_check false" tcl command.
ERROR: [Labtools 27-3165] End of startup status: LOW
ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.
```

Figure 61 - Error Labtools 27-3303 & Common 17-39

If the program is successful, an “In-System IBERT” should appear in the hardware window as shown below.


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210308A1C883

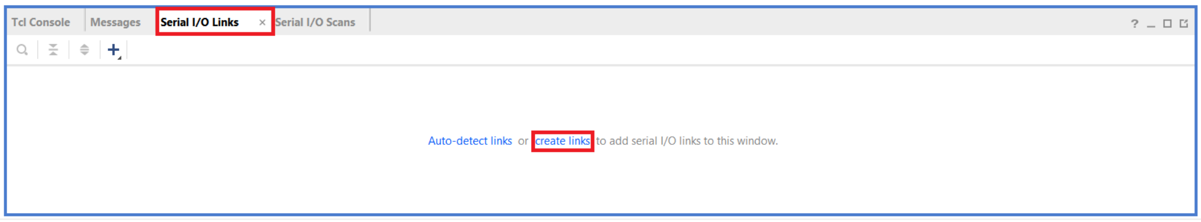
There are no serial I/O links. Auto-detect links Create links

Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210308	Open
xcvu9p_0 (2)	Programmed
SysMon (System Mon	
In-System IBERT (pcie	
Quad_226 (4)	
MGT_X1Y28	
MGT_X1Y29	
MGT_X1Y30	
MGT_X1Y31	
Quad_227 (4)	
MGT_X1Y32	
MGT_X1Y33	
MGT_X1Y34	
MGT_X1Y35	

Figure 62 - Hardware window

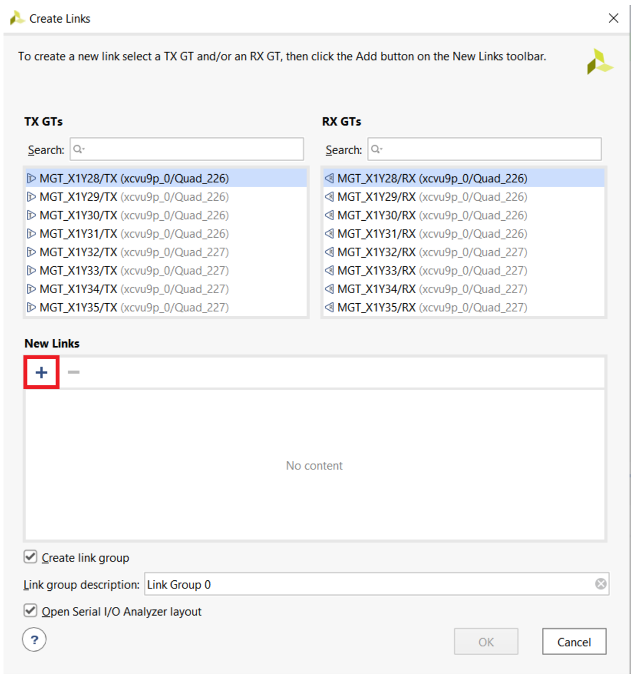
Under the Vivado Interface, open the “Serial I/O Links” tab and select “create links”.



Auto-detect links or create links to add serial I/O links to this window.

Figure 63 - Serial I/O links

Click the “+” sign to select the desired TX GTs and/or RX GTs.



Create Links

To create a new link select a TX GT and/or an RX GT, then click the Add button on the New Links toolbar.

TX GTs

Search:

- MGT_X1Y28/TX (xcvu9p_0/Quad_226)
- MGT_X1Y29/TX (xcvu9p_0/Quad_226)
- MGT_X1Y30/TX (xcvu9p_0/Quad_226)
- MGT_X1Y31/TX (xcvu9p_0/Quad_226)
- MGT_X1Y32/TX (xcvu9p_0/Quad_227)
- MGT_X1Y33/TX (xcvu9p_0/Quad_227)
- MGT_X1Y34/TX (xcvu9p_0/Quad_227)
- MGT_X1Y35/TX (xcvu9p_0/Quad_227)

RX GTs

Search:

- MGT_X1Y28/RX (xcvu9p_0/Quad_226)
- MGT_X1Y29/RX (xcvu9p_0/Quad_226)
- MGT_X1Y30/RX (xcvu9p_0/Quad_226)
- MGT_X1Y31/RX (xcvu9p_0/Quad_226)
- MGT_X1Y32/RX (xcvu9p_0/Quad_227)
- MGT_X1Y33/RX (xcvu9p_0/Quad_227)
- MGT_X1Y34/RX (xcvu9p_0/Quad_227)
- MGT_X1Y35/RX (xcvu9p_0/Quad_227)

New Links

No content

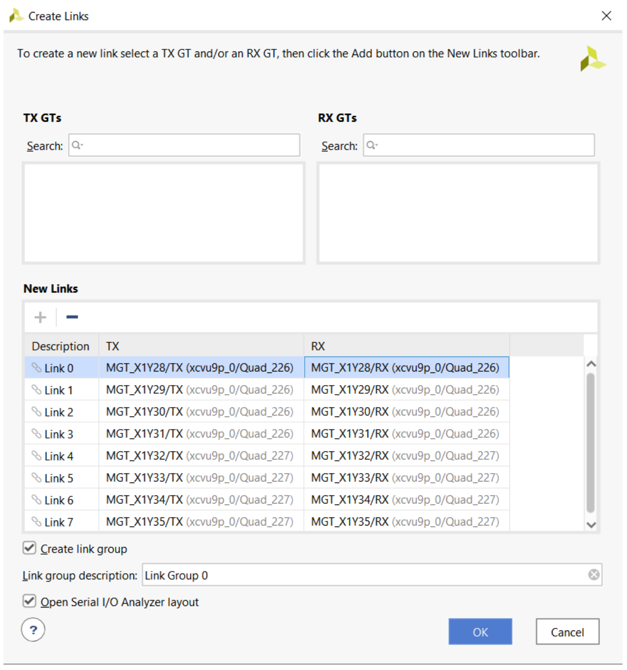
☑ Create link group

Link group description: Link Group 0

☑ Open Serial I/O Analyzer layout

Figure 64 - Create links

The "New Links" field must contain the selected TX GTs and /or RX GTs. Click "OK".



Create Links

To create a new link select a TX GT and/or an RX GT, then click the Add button on the New Links toolbar.

TX GTs

Search:

RX GTs

Search:

New Links

Description	TX	RX
Link 0	MGT_X1Y28/TX (xcvu9p_0/Quad_226)	MGT_X1Y28/RX (xcvu9p_0/Quad_226)
Link 1	MGT_X1Y29/TX (xcvu9p_0/Quad_226)	MGT_X1Y29/RX (xcvu9p_0/Quad_226)
Link 2	MGT_X1Y30/TX (xcvu9p_0/Quad_226)	MGT_X1Y30/RX (xcvu9p_0/Quad_226)
Link 3	MGT_X1Y31/TX (xcvu9p_0/Quad_226)	MGT_X1Y31/RX (xcvu9p_0/Quad_226)
Link 4	MGT_X1Y32/TX (xcvu9p_0/Quad_227)	MGT_X1Y32/RX (xcvu9p_0/Quad_227)
Link 5	MGT_X1Y33/TX (xcvu9p_0/Quad_227)	MGT_X1Y33/RX (xcvu9p_0/Quad_227)
Link 6	MGT_X1Y34/TX (xcvu9p_0/Quad_227)	MGT_X1Y34/RX (xcvu9p_0/Quad_227)
Link 7	MGT_X1Y35/TX (xcvu9p_0/Quad_227)	MGT_X1Y35/RX (xcvu9p_0/Quad_227)

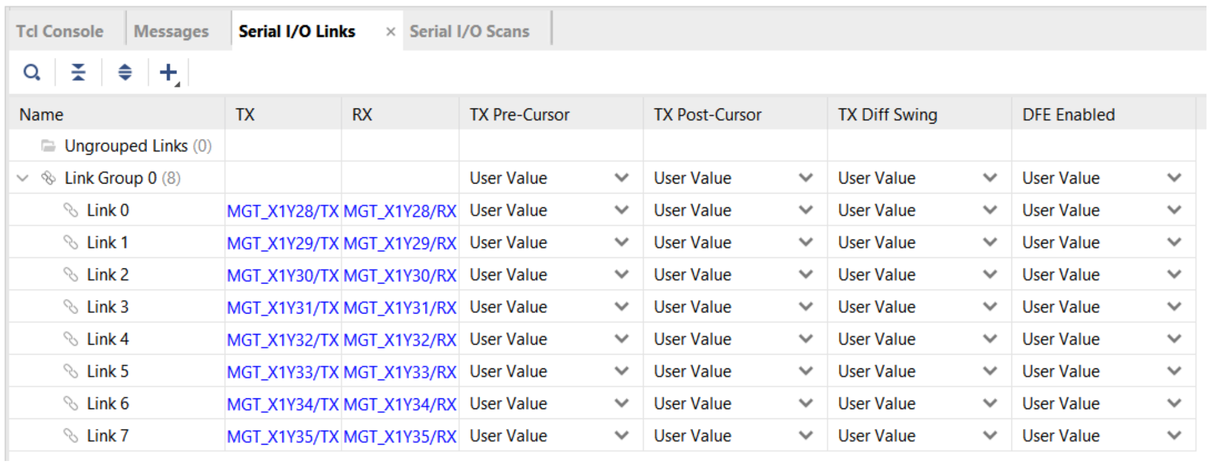
☑ Create link group

Link group description: Link Group 0

☑ Open Serial I/O Analyzer layout

Figure 65 - New links

The "Serial I/O Links" must also contains the links selected from the previous step.



Name	TX	RX	TX Pre-Cursor	TX Post-Cursor	TX Diff Swing	DFE Enabled
Ungrouped Links (0)						
Link Group 0 (8)			User Value	User Value	User Value	User Value
Link 0	MGT_X1Y28/TX	MGT_X1Y28/RX	User Value	User Value	User Value	User Value
Link 1	MGT_X1Y29/TX	MGT_X1Y29/RX	User Value	User Value	User Value	User Value
Link 2	MGT_X1Y30/TX	MGT_X1Y30/RX	User Value	User Value	User Value	User Value
Link 3	MGT_X1Y31/TX	MGT_X1Y31/RX	User Value	User Value	User Value	User Value
Link 4	MGT_X1Y32/TX	MGT_X1Y32/RX	User Value	User Value	User Value	User Value
Link 5	MGT_X1Y33/TX	MGT_X1Y33/RX	User Value	User Value	User Value	User Value
Link 6	MGT_X1Y34/TX	MGT_X1Y34/RX	User Value	User Value	User Value	User Value
Link 7	MGT_X1Y35/TX	MGT_X1Y35/RX	User Value	User Value	User Value	User Value

Figure 66 - Serial I/O links

Right-click on one of the following links (for example, Link 0), then select “Create Scan…” from the drop-down menu.

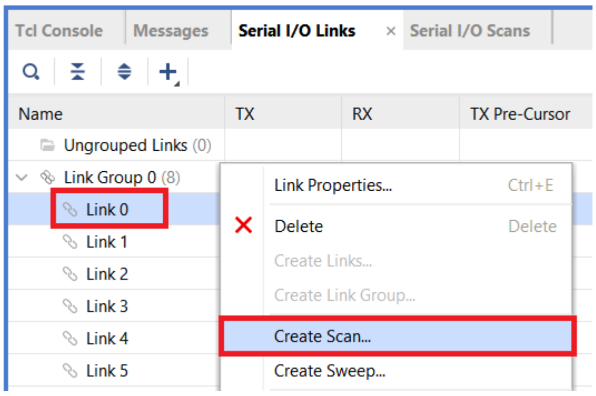
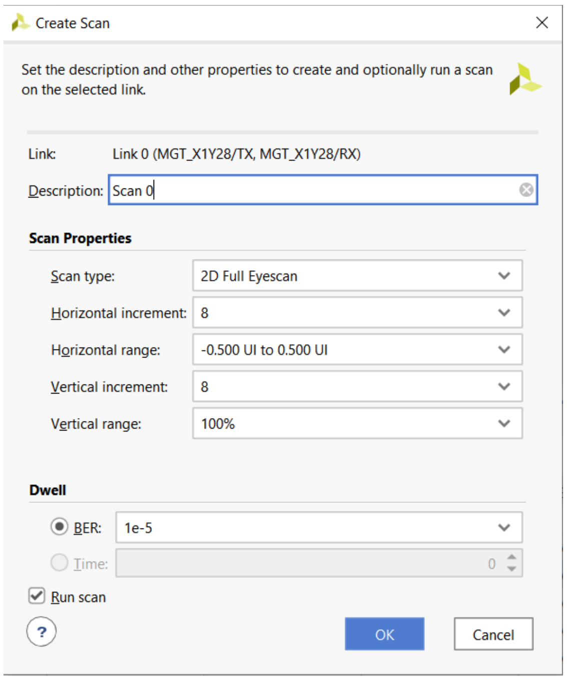


Figure 67 - Create scan

Select “OK”, leaving the default settings to capture a full 2D eye scan.



Create Scan

Set the description and other properties to create and optionally run a scan on the selected link.

Link: Link 0 (MGT_X1Y28/TX, MGT_X1Y28/RX)

Description: Scan 0

Scan Properties

Scan type: 2D Full Eyescan
Horizontal increment: 8
Horizontal range: -0.500 UI to 0.500 UI
Vertical increment: 8
Vertical range: 100%

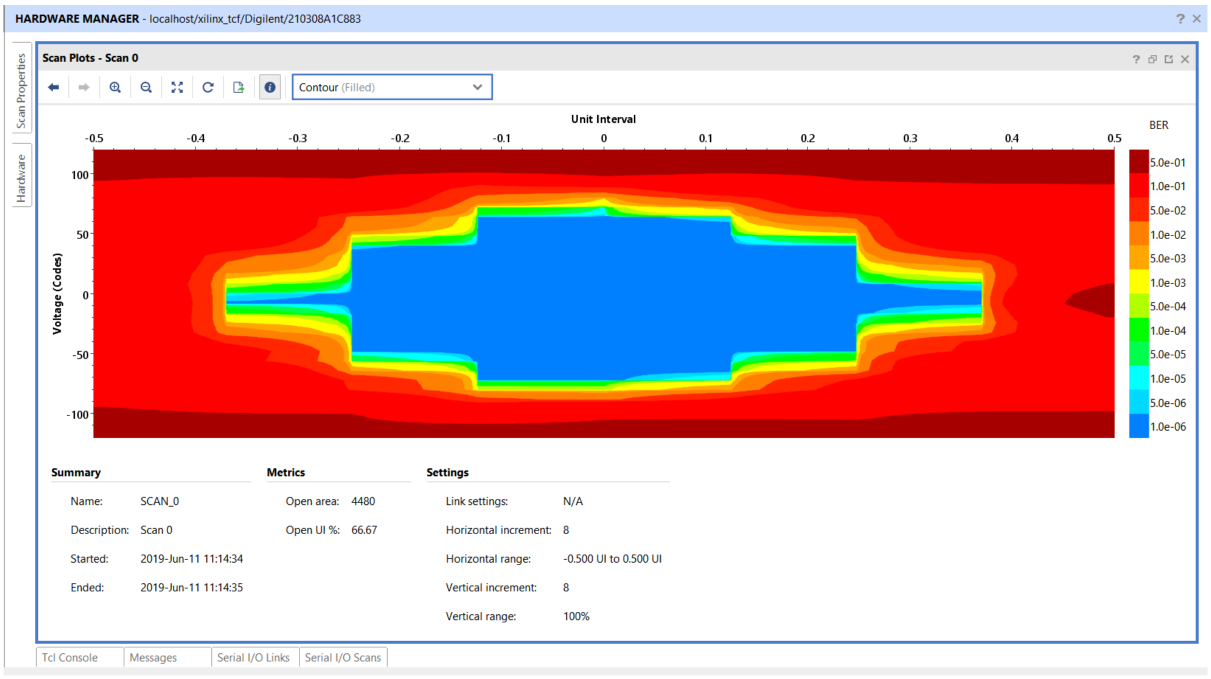
Dwell

[✓ BER:] 1e-5
Time: 0

[✓ Run scan]

Figure 68 - 2D Full Eye scan configuration

Figure 69 shows an example capture of 2D full eye scan in the “Scan Plots” window.



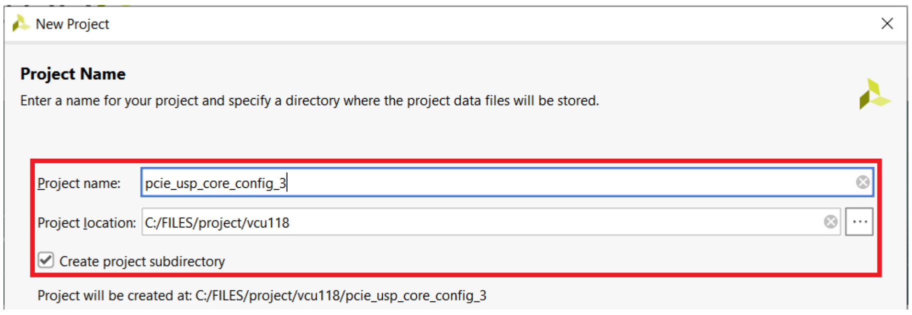
Summary		Metrics		Settings	
Name:	SCAN_0	Open area:	4480	Link settings:	N/A
Description:	Scan 0	Open UI %:	66.67	Horizontal increment:	8
Started:	2019-Jun-11 11:14:34			Horizontal range:	-0.500 UI to 0.500 UI
Ended:	2019-Jun-11 11:14:35			Vertical increment:	8
				Vertical range:	100%

Figure 69 - 2D Full eye scan

Descrambler for Gen3

The data in the PIPE interface during packet transmission is scrambled; analyzing packets in this interface without a descrambler module would be difficult. This section describes how to enable the descrambler module and capture the descrambler packets. Please refer to some of the steps in General Design Steps in Configuring the PCIe Core.

Create a project to configure the PCIe core enabling the descrambler debug tool.



Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: pcie_usp_core_config_3

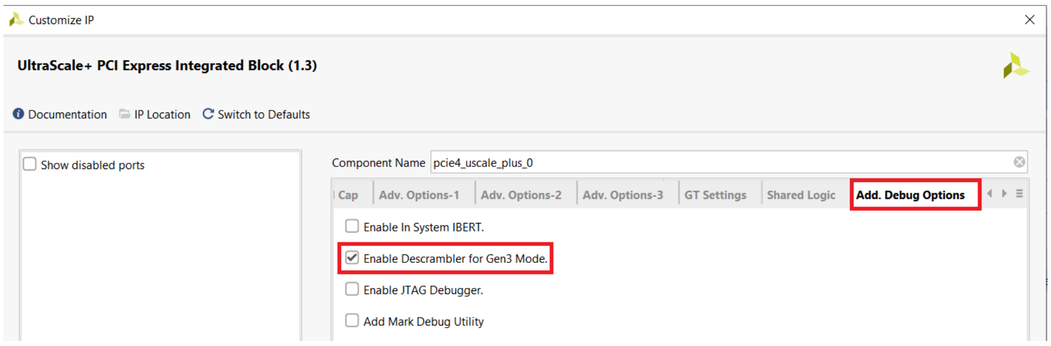
Project location: C:/FILES/project/vcu118

☑ Create project subdirectory

Project will be created at: C:/FILES/project/vcu118/pcie_usp_core_config_3

Figure 70 - Create project

Select the check box “Enable Descrambler for Gen3 Mode”.


Customize IP

UltraScale+ PCI Express Integrated Block (1.3)

Documentation | IP Location | Switch to Defaults

Show disabled ports

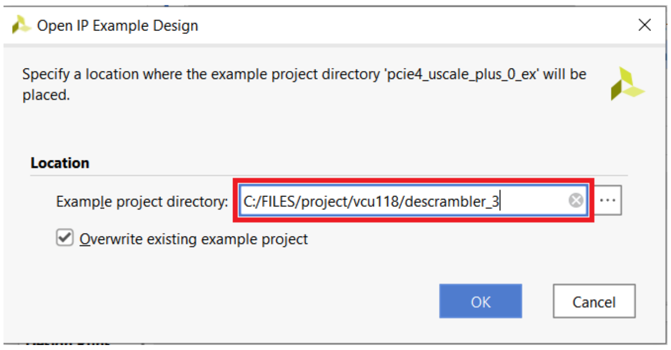
Component Name: pcie4_uscale_plus_0

Cap | Adv. Options-1 | Adv. Options-2 | Adv. Options-3 | GT Settings | Shared Logic | Add. Debug Options

- [ ] Enable In System IBERT.
- [x] Enable Descrambler for Gen3 Mode.
- [ ] Enable JTAG Debugger.
- [ ] Add Mark Debug Utility

Figure 71 - Enable descramble

Open and select the directory for the IP example design.


Open IP Example Design

Specify a location where the example project directory 'pcie4_uscale_plus_0_ex' will be placed.

Location

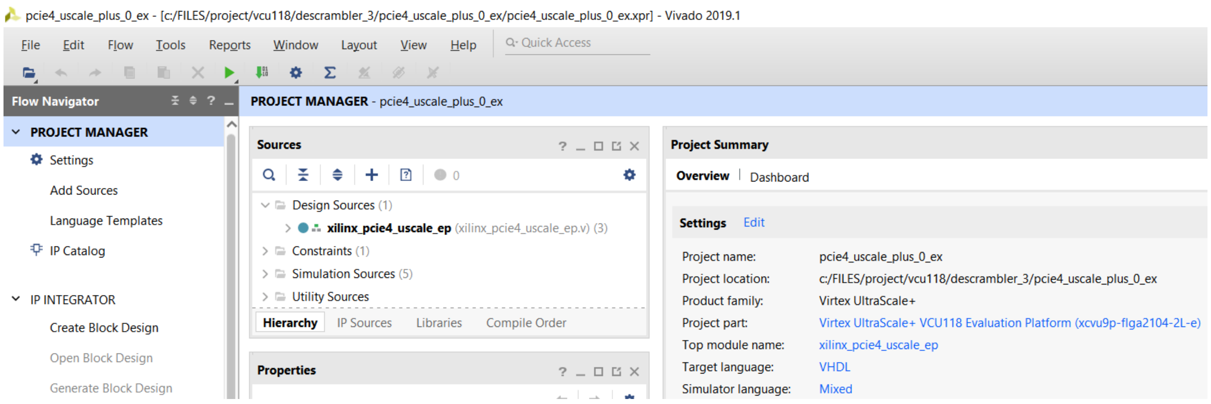
Example project directory: C:/FILES/project/vcu118/descrambler_3

- [x] Overwrite existing example project

OK | Cancel

Figure 72 - Example design

A new Vivado project containing the IP example design will initialize.


pcie4_uscale_plus_0_ex - [c:/FILES/project/vcu118/descrambler_3/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.xpr] - Vivado 2019.1

PROJECT MANAGER - pcie4_uscale_plus_0_ex

Sources

- Design Sources (1)
  - xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
- Constraints (1)
- Simulation Sources (5)
- Utility Sources

Hierarchy | IP Sources | Libraries | Compile Order

Properties

Project Summary

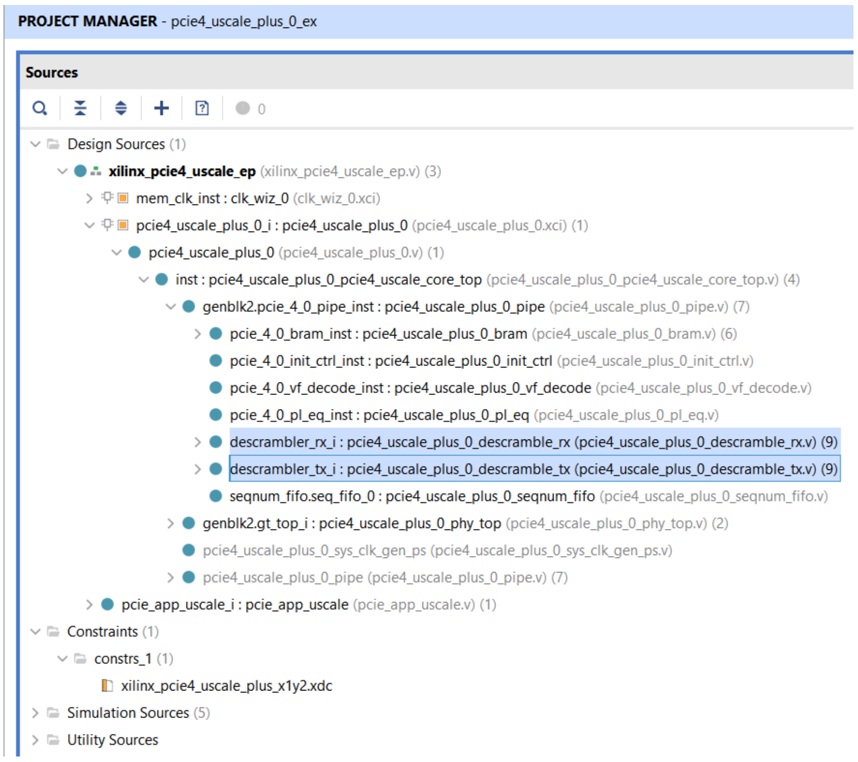
Overview | Dashboard

Settings Edit

Project name:	pcie4_uscale_plus_0_ex
Project location:	c:/FILES/project/vcu118/descrambler_3/pcie4_uscale_plus_0_ex
Product family:	Virtex UltraScale+
Project part:	Virtex UltraScale+ VCU118 Evaluation Platform (xcvu9p-flga2104-2L-e)
Top module name:	xilinx_pcie4_uscale_ep
Target language:	VHDL
Simulator language:	Mixed

Figure 73 - Example design with PCIe descrambler debug enabled

The design source hierarchy must contain the descrambler tool.


PROJECT MANAGER - pcie4_uscale_plus_0_ex

Sources

```
Design Sources (1)
  xilinx_pcie4_uscale_ep (xilinx_pcie4_uscale_ep.v) (3)
    mem_clk_inst : clk_wiz_0 (clk_wiz_0.xci)
    pcie4_uscale_plus_0_i : pcie4_uscale_plus_0 (pcie4_uscale_plus_0.xci) (1)
      pcie4_uscale_plus_0 (pcie4_uscale_plus_0.v) (1)
        inst : pcie4_uscale_plus_0_pcie4_uscale_core_top (pcie4_uscale_plus_0_pcie4_uscale_core_top.v) (4)
          genblk2.pcie_4_0_pipe_inst : pcie4_uscale_plus_0_pipe (pcie4_uscale_plus_0_pipe.v) (7)
            pcie_4_0_bram_inst : pcie4_uscale_plus_0_bram (pcie4_uscale_plus_0_bram.v) (6)
            pcie_4_0_init_ctrl_inst : pcie4_uscale_plus_0_init_ctrl (pcie4_uscale_plus_0_init_ctrl.v)
            pcie_4_0_vf_decode_inst : pcie4_uscale_plus_0_vf_decode (pcie4_uscale_plus_0_vf_decode.v)
            pcie_4_0_pl_eq_inst : pcie4_uscale_plus_0_pl_eq (pcie4_uscale_plus_0_pl_eq.v)
            descrambler_rx_i : pcie4_uscale_plus_0_descramble_rx (pcie4_uscale_plus_0_descramble_rx.v) (9)
            descrambler_tx_i : pcie4_uscale_plus_0_descramble_tx (pcie4_uscale_plus_0_descramble_tx.v) (9)
            seqnum_fifo.seq_fifo_0 : pcie4_uscale_plus_0_seqnum_fifo (pcie4_uscale_plus_0_seqnum_fifo.v)
          genblk2.gt_top_i : pcie4_uscale_plus_0_phy_top (pcie4_uscale_plus_0_phy_top.v) (2)
          pcie4_uscale_plus_0_sys_clk_gen_ps (pcie4_uscale_plus_0_sys_clk_gen_ps.v)
          pcie4_uscale_plus_0_pipe (pcie4_uscale_plus_0_pipe.v) (7)
    pcie_app_uscale_i : pcie_app_uscale (pcie_app_uscale.v) (1)
Constraints (1)
  constrs_1 (1)
    xilinx_pcie4_uscale_plus_x1y2.xdc
Simulation Sources (5)
Utility Sources
```

Figure 74 - Design source hierarchy

For examples of updating the constraint files, see the following figures:

- Figure 20 - Edit constraint file
- Figure 21 – Activate reset pin
- Figure 22 - Adding lines of code

Click on “Run Synthesis”.

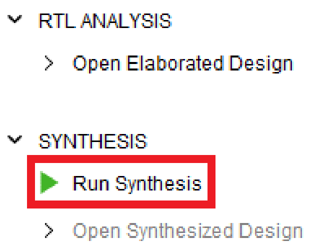
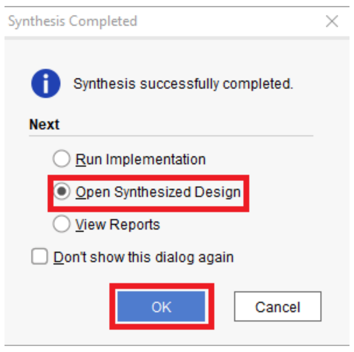


Figure 75 - Synthesis

Select the radio button “Open Synthesized Design”. Click “OK”.


Synthesis Completed

Synthesis successfully completed.

Next

( ) Run Implementation
[✓ Open Synthesized Design]
( ) View Reports

☐ Don't show this dialog again

OK Cancel

Figure 76 - Synthesis complete

Select “Set Up Debug” under “Open Synthesized Design”.

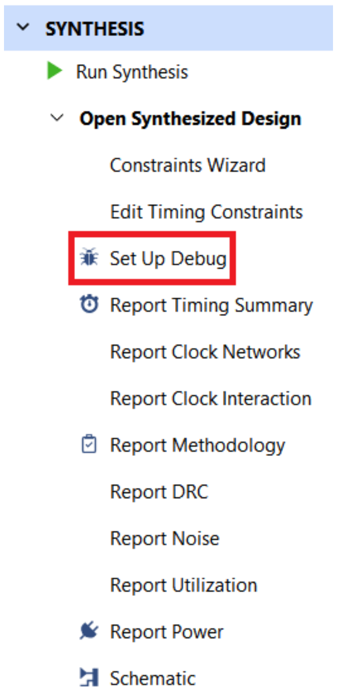
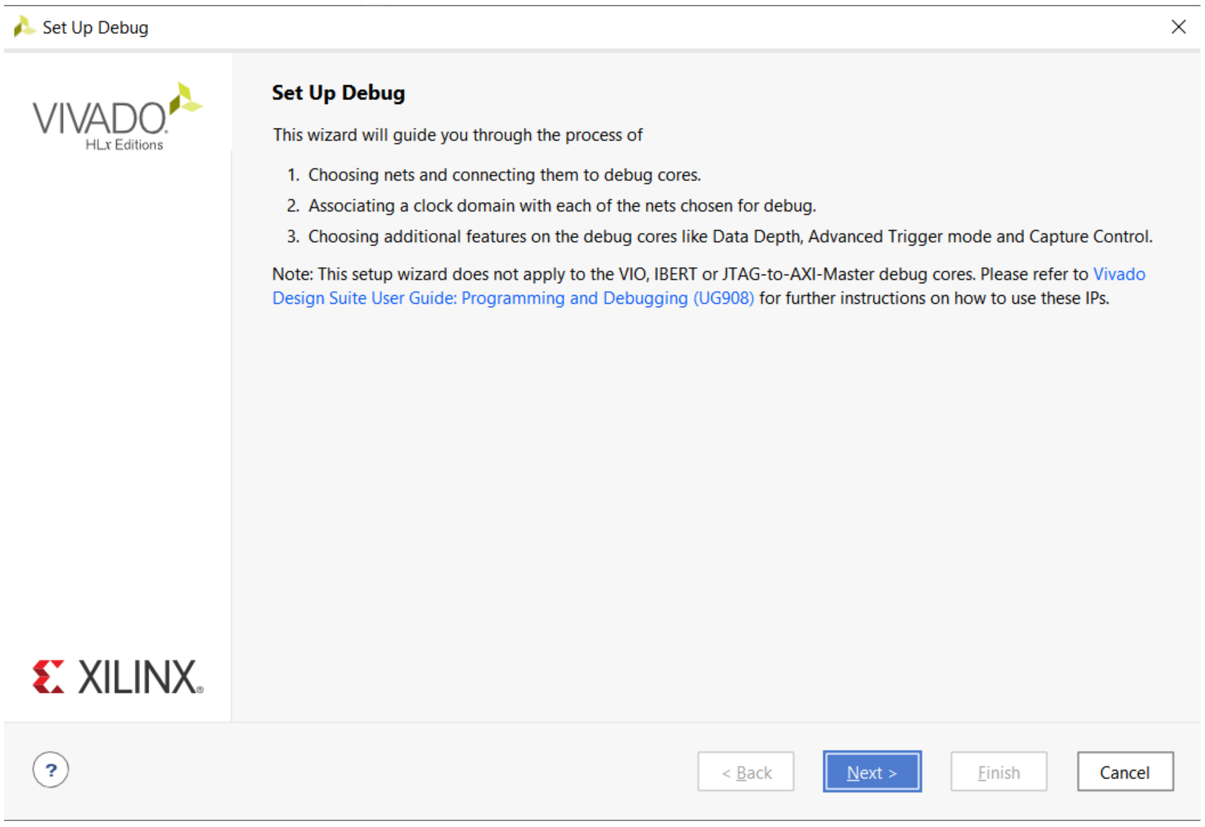


Figure 77 - Set Up debug

The “Set Up Debug” wizard will appear. Click “Next”.


Set Up Debug

Set Up Debug

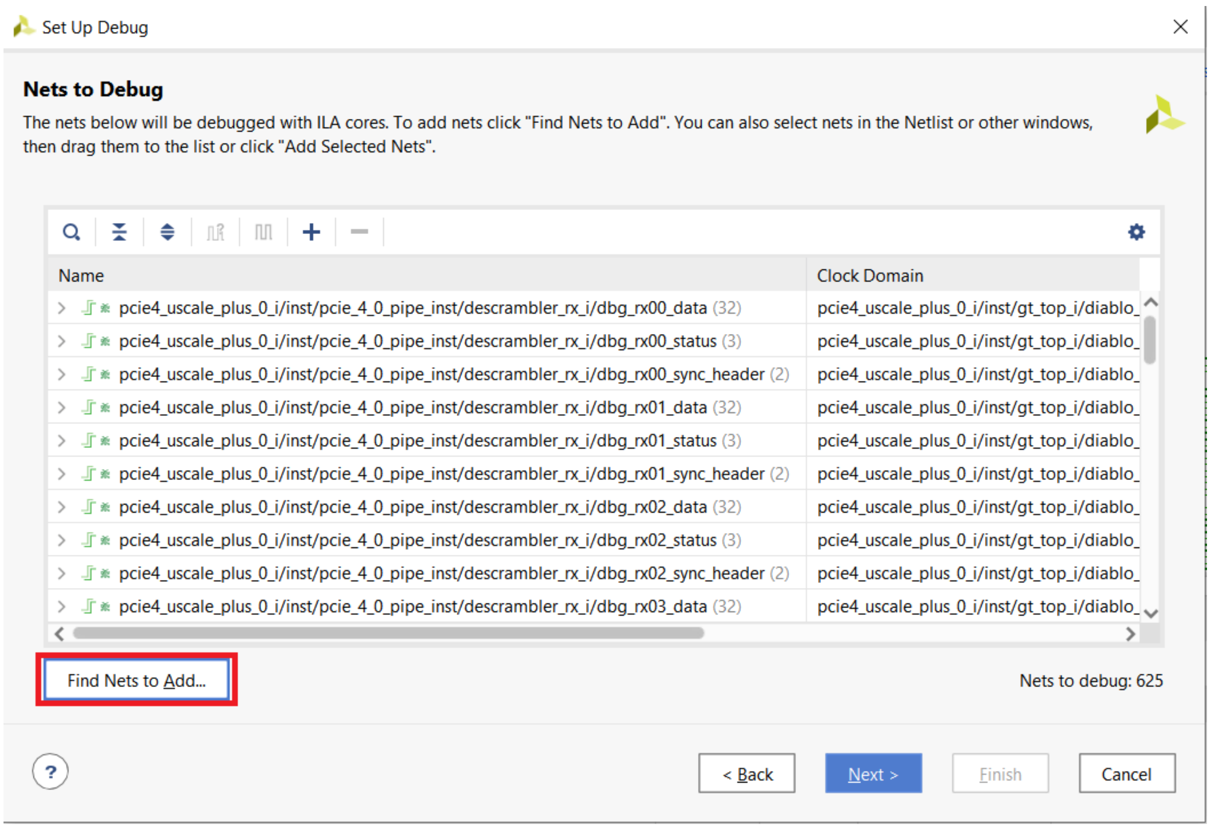
This wizard will guide you through the process of

1. Choosing nets and connecting them to debug cores.
2. Associating a clock domain with each of the nets chosen for debug.
3. Choosing additional features on the debug cores like Data Depth, Advanced Trigger mode and Capture Control.

Note: This setup wizard does not apply to the VIO, IBERT or JTAG-to-AXI-Master debug cores. Please refer to Vivado Design Suite User Guide: Programming and Debugging (UG908) for further instructions on how to use these IPs.

Figure 78 - Set up debug

Click on “Find Nets to Add”. Click “Next”.


Set Up Debug

Nets to Debug

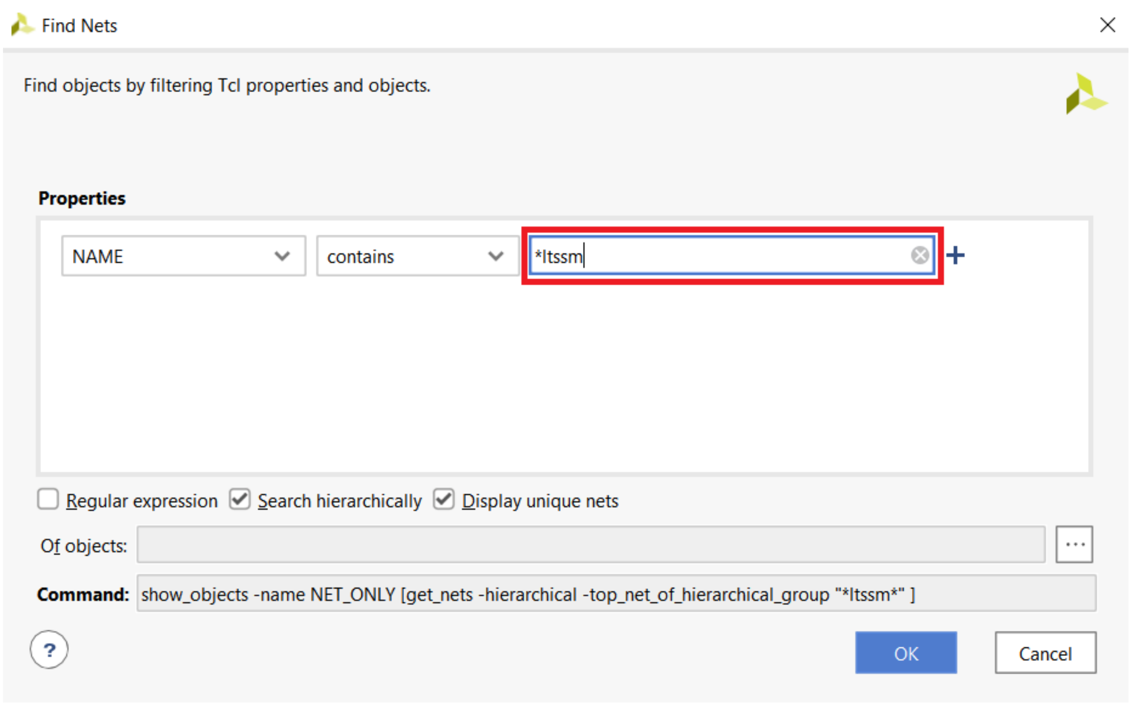
The nets below will be debugged with ILA cores. To add nets click "Find Nets to Add". You can also select nets in the Netlist or other windows, then drag them to the list or click "Add Selected Nets".

Name	Clock Domain
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx01_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx01_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx01_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx02_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx02_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx02_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx03_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_

Nets to debug: 625

Figure 79 - Nets to debug

In the properties below, search for *ltssm. Click “OK”.


Find Nets

Find objects by filtering Tcl properties and objects.

Properties

NAME | contains | *ltssm

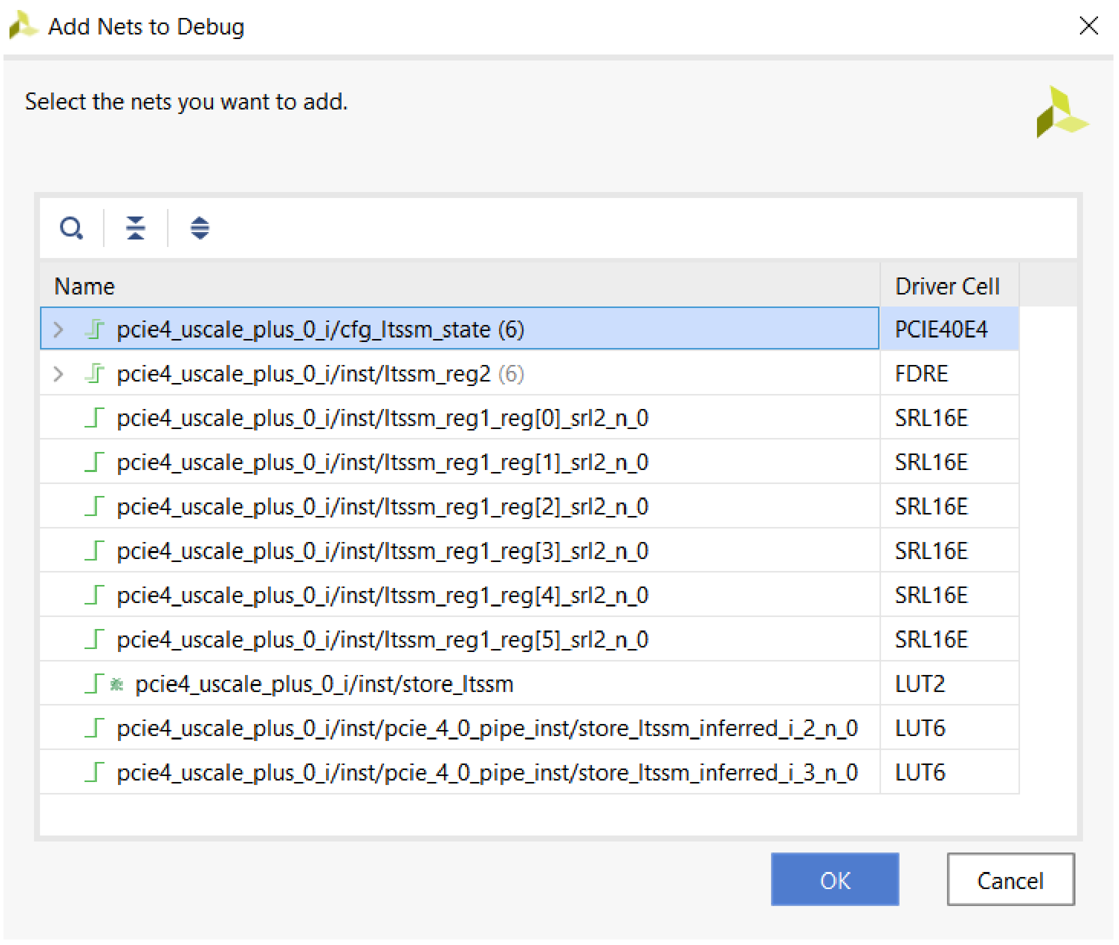
☐ Regular expression ☑ Search hierarchically ☑ Display unique nets

Of objects:

Command: show_objects -name NET_ONLY [get_nets -hierarchical -top_net_of_hierarchical_group "*ltssm*"]

Figure 80 - Find ltssm net

Add **cfg_ltssm_state** net. Click “OK”.

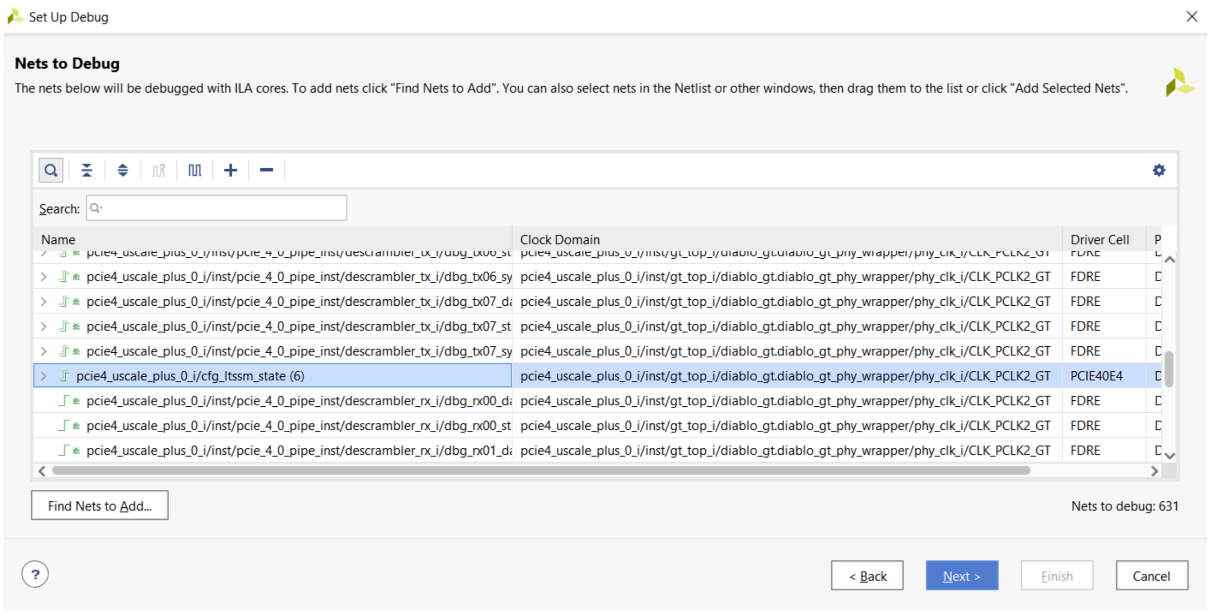


Select the nets you want to add.

Name	Driver Cell
pcie4_uscale_plus_0_i/cfg_ltssm_state (6)	PCIE40E4
pcie4_uscale_plus_0_i/inst/ltssm_reg2 (6)	FDRE
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[0]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[1]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[2]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[3]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[4]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/ltssm_reg1_reg[5]_srl2_n_0	SRL16E
pcie4_uscale_plus_0_i/inst/store_ltssm	LUT2
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/store_ltssm_inferred_i_2_n_0	LUT6
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/store_ltssm_inferred_i_3_n_0	LUT6

Figure 81 - Add nets to debug

Make sure that the “cfg_ltssm_state” net is added.



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".

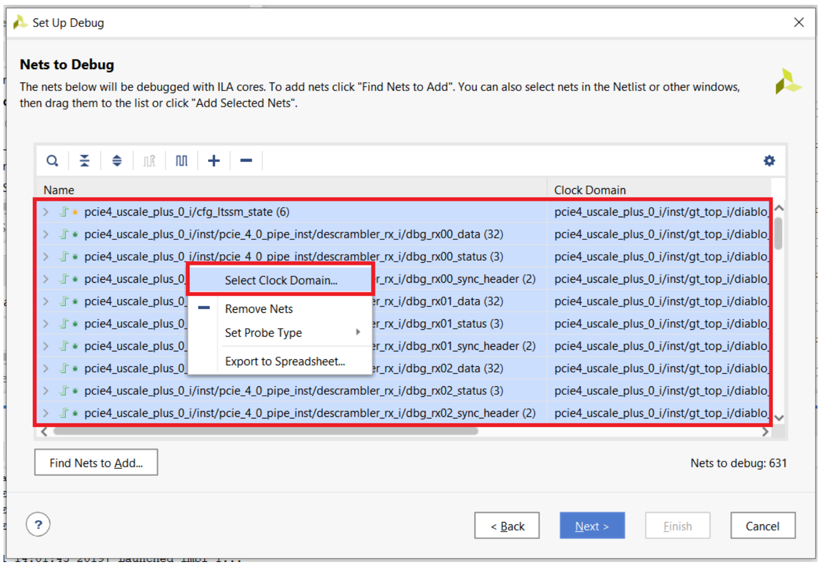
Search:

Name	Clock Domain	Driver Cell	P
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_tx_i/dbg_tx06_st	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_tx_i/dbg_tx06_sy	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_tx_i/dbg_tx07_da	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_tx_i/dbg_tx07_st	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_tx_i/dbg_tx07_sy	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/cfg_ltssm_state (6)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	PCIE40E4	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_da	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_st	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx01_da	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	D

Nets to debug: 631

Figure 82 - Net cfg_ltssm_state

Select all of the nets in the “Nets to Debug” window and right-click on the selected nets. Click on “Select Clock Domain…”.



Set Up Debug

Nets to Debug

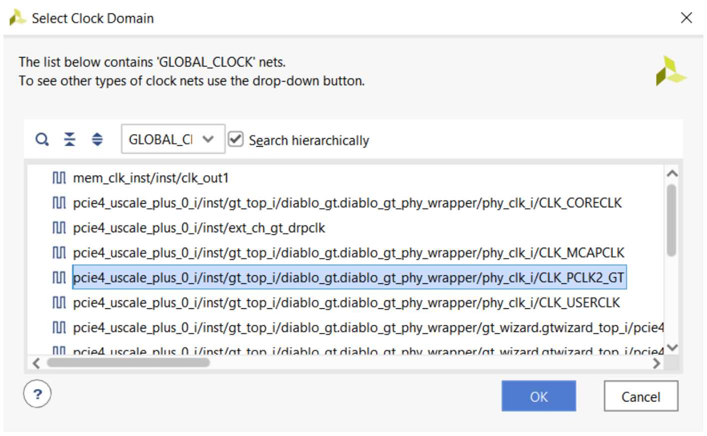
The nets below will be debugged with ILA cores. To add nets click "Find Nets to Add". You can also select nets in the Netlist or other windows, then drag them to the list or click "Add Selected Nets".

Name	Clock Domain
pcie4_uscale_plus_0_i/cfg_ltssm_state (6)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx00_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0 r_rx_i/dbg_rx00_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0 r_rx_i/dbg_rx01_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0 r_rx_i/dbg_rx01_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0 r_rx_i/dbg_rx01_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0 r_rx_i/dbg_rx02_data (32)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx02_status (3)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo
pcie4_uscale_plus_0_i/inst/pcie_4_0_pipe_inst/descrambler_rx_i/dbg_rx02_sync_header (2)	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo

Nets to debug: 631

Figure 83 - Select clock domain

Make sure to select only one clock for all nets to debug. Click "OK".



Select Clock Domain

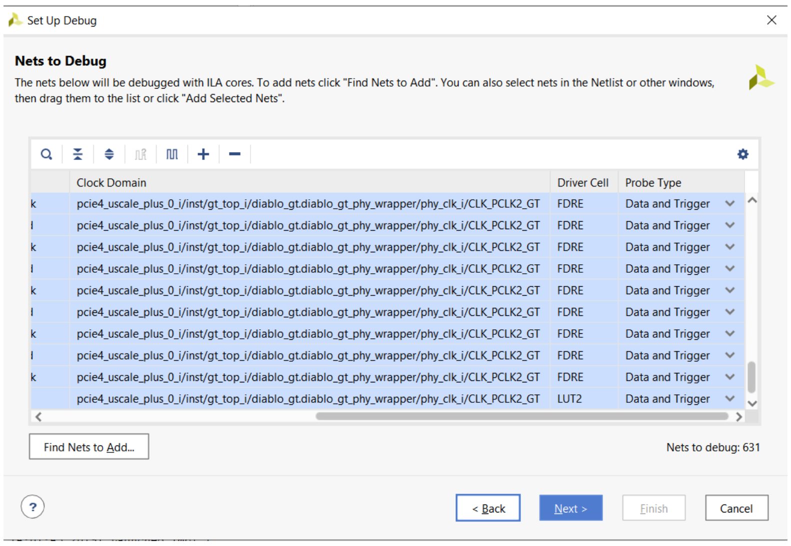
The list below contains 'GLOBAL_CLOCK' nets.
To see other types of clock nets use the drop-down button.

GLOBAL_CI [✓ Search hierarchically]

```
mem_clk_inst/inst/clk_out1
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_CORECLK
pcie4_uscale_plus_0_i/inst/ext_ch_gt_drpclk
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_MCAPCLK
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_USERCLK
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/gt_wizard.gtwizard_top_i/pcie4
pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/gt_wizard.gtwizard_top_i/pcie4
```

Figure 84 - Select clock

The clock domain should display only one clock source.



Set Up Debug

Nets to Debug

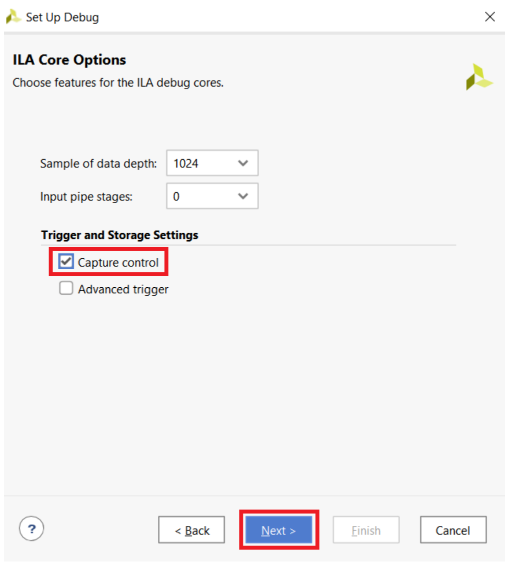
The nets below will be debugged with ILA cores. To add nets click "Find Nets to Add". You can also select nets in the Netlist or other windows, then drag them to the list or click "Add Selected Nets".

	Clock Domain	Driver Cell	Probe Type
k	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
l	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
k	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
l	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
k	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
l	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
k	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
l	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
k	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	FDRE	Data and Trigger
	pcie4_uscale_plus_0_i/inst/gt_top_i/diablo_gt.diablo_gt_phy_wrapper/phy_clk_i/CLK_PCLK2_GT	LUT2	Data and Trigger

Nets to debug: 631

Figure 85 - One clock source

The next part of set up debug wizard contains “Trigger and Storage Settings”. Tick the check box “Capture control”. Click “Next”.



ILA Core Options

Choose features for the ILA debug cores.

Sample of data depth: 1024

Input pipe stages: 0

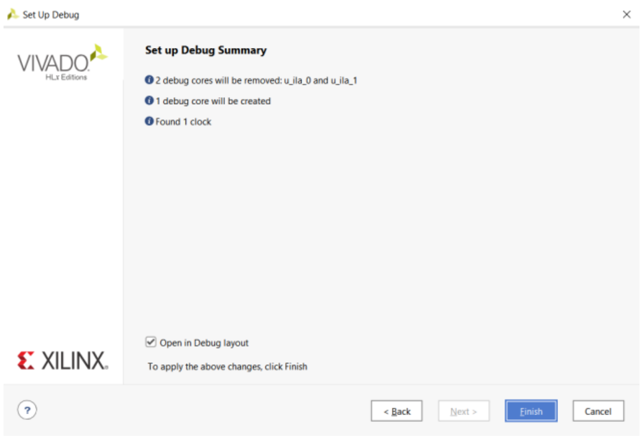
Trigger and Storage Settings

[✓] Capture control

[ ] Advanced trigger

Figure 86 - Capture control

The final phase of the setup wizard shows a debug summary.



Set up Debug Summary

2 debug cores will be removed: u_ila_0 and u_ila_1

1 debug core will be created

Found 1 clock

[✓] Open in Debug layout

To apply the above changes, click Finish

Figure 87 - Set up debug summary

Click on “Generate Bitstream”.

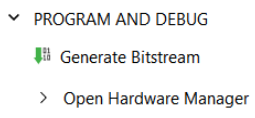
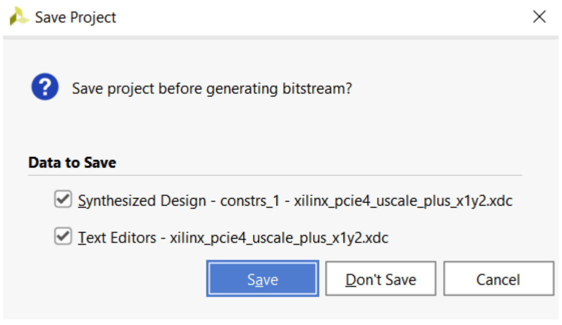


Figure 88 - Generate Bitstream

A message dialog box will appear to save the updated synthesized design constraints. Click “Save”.



Save Project

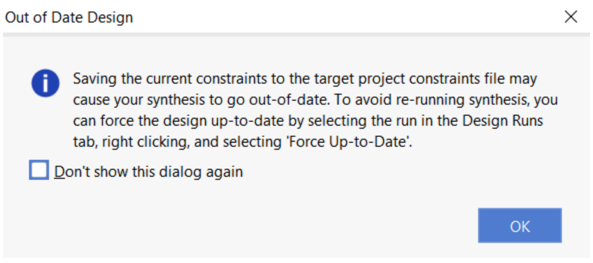
Save project before generating bitstream?

Data to Save

- [✓] Synthesized Design - constrs_1 - xilinx_pcie4_uscale_plus_x1y2.xdc
- [✓] Text Editors - xilinx_pcie4_uscale_plus_x1y2.xdc

Figure 89 - XDC with debug ILA

A message dialog box "Out of Date Design" will appear. Click "OK".



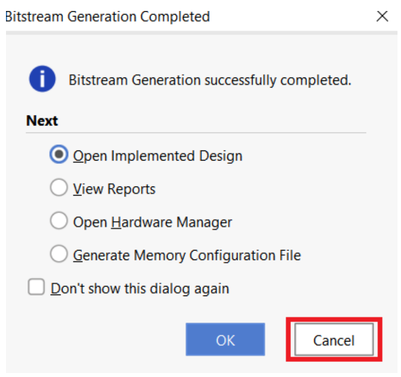
Out of Date Design

Saving the current constraints to the target project constraints file may cause your synthesis to go out-of-date. To avoid re-running synthesis, you can force the design up-to-date by selecting the run in the Design Runs tab, right clicking, and selecting 'Force Up-to-Date'.

- [ ] Don't show this dialog again

Figure 90 - Out of date design

A message dialog box will confirm bitstream generation is completed. Click "Cancel".



Bitstream Generation Completed

Bitstream Generation successfully completed.

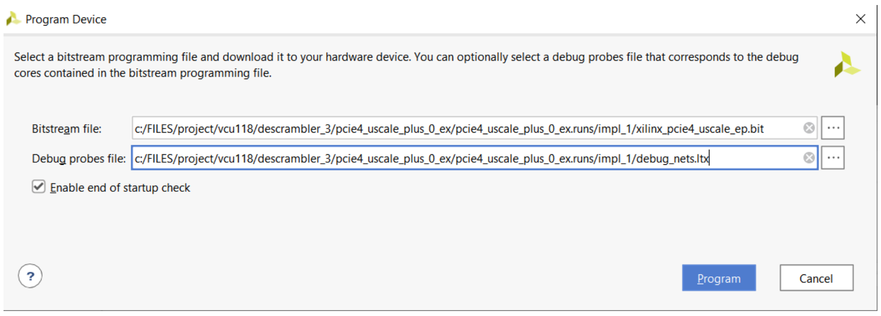
Next

- [✓ Open Implemented Design]
- View Reports
- Open Hardware Manager
- Generate Memory Configuration File

- [ ] Don't show this dialog again

Figure 91 - Bitstream complete

Program the device with the correct bitstream file and debug probe file.



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.

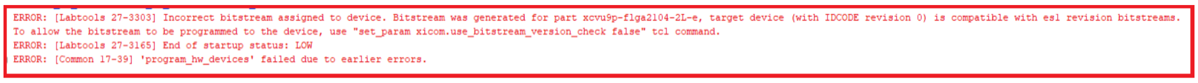
Bitstream file:	c:/FILES/project/vcu118/descrambler_3/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/xilinx_pcie4_uscale_ep.bit
Debug probes file:	c:/FILES/project/vcu118/descrambler_3/pcie4_uscale_plus_0_ex/pcie4_uscale_plus_0_ex.runs/impl_1/debug_nets.ltx

- [✓] Enable end of startup check

Figure 92 - Program device

If the errors **Labtools 27-3303** & **Common 17-39** occur, enter the Tcl command below.

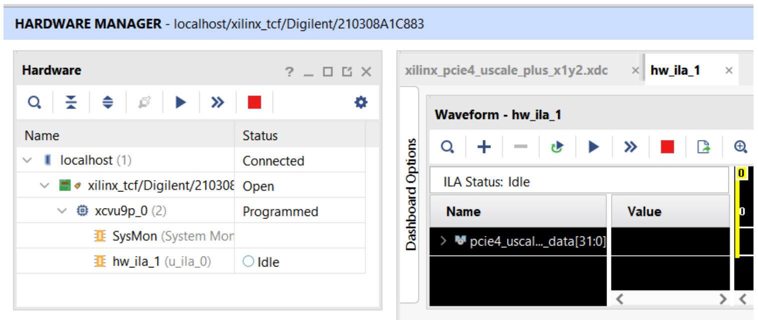
```
set_param xicom.use_bitstream_version_check false
```



```
ERROR: [Labtools 27-3303] Incorrect bitstream assigned to device. Bitstream was generated for part xcvu9p-flga2104-2L-e, target device (with IDCODE revision 0) is compatible with es1 revision bitstreams.
To allow the bitstream to be programmed to the device, use "set_param xicom.use_bitstream_version_check false" tcl command.
ERROR: [Labtools 27-3165] End of startup status: LOW
ERROR: [Common 17-39] 'program_hw_devices' failed due to earlier errors.
```

Figure 93 - Error Labtools 27-3303 & Common 17-39

After successful programming of the target device, a "hw_ila_1" will appear in the **Hardware** window.

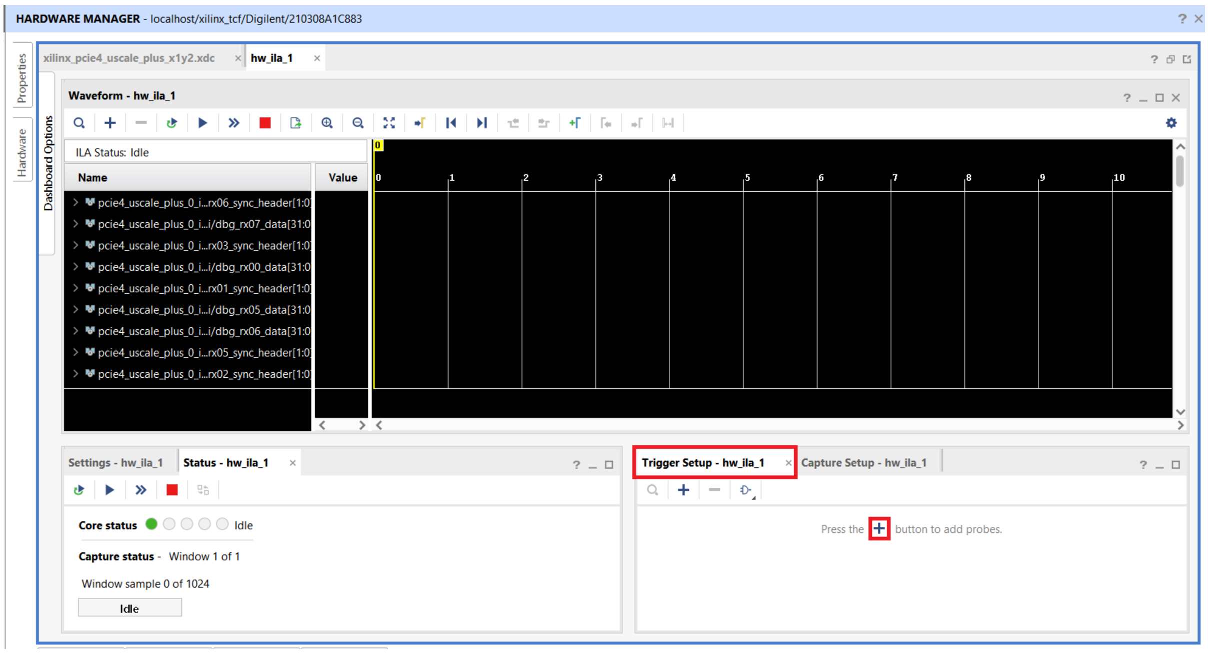


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210308A1C883

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210308	Open
xcvu9p_0 (2)	Programmed
SysMon (System Mon	
hw_ila_1 (u_ila_0)	Idle

Figure 94 - Hardware manager

In the "Trigger Setup" window, add probes by pressing the **+** sign.

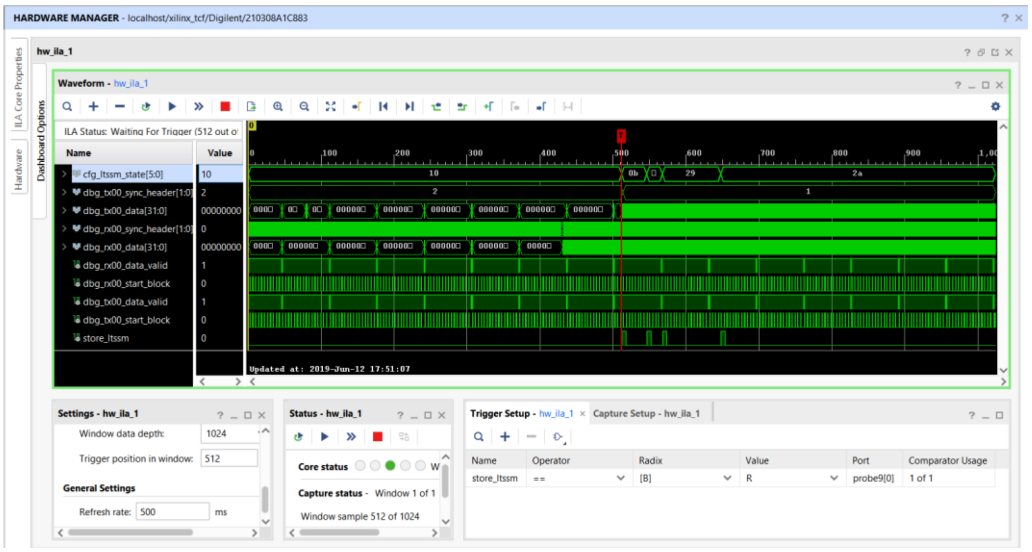


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210308A1C883

Properties
Hardware
Dashboard Options
xilinx_pcie4_uscale_plus_x1y2.xdc
hw_ila_1
Waveform - hw_ila_1
ILA Status: Idle
Name
Value
pcie4_uscale_plus_0_i...rx06_sync_header[1:0]
pcie4_uscale_plus_0_i...i/dbg_rx07_data[31:0]
pcie4_uscale_plus_0_i...rx03_sync_header[1:0]
pcie4_uscale_plus_0_i...i/dbg_rx00_data[31:0]
pcie4_uscale_plus_0_i...rx01_sync_header[1:0]
pcie4_uscale_plus_0_i...i/dbg_rx05_data[31:0]
pcie4_uscale_plus_0_i...i/dbg_rx06_data[31:0]
pcie4_uscale_plus_0_i...rx05_sync_header[1:0]
pcie4_uscale_plus_0_i...rx02_sync_header[1:0]
Settings - hw_ila_1
Status - hw_ila_1
Core status
Idle
Capture status - Window 1 of 1
Window sample 0 of 1024
Idle
Trigger Setup - hw_ila_1
Capture Setup - hw_ila_1
Press the + button to add probes.

Figure 95 - Trigger setup

Click on run trigger icon ▶ to capture on **cfg_ltssm_state**.


HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210308A1C883

Waveform - hw_ila_1

ILA Status: Waiting For Trigger (512 out of

Name	Value
cfg_ltssm_state[5:0]	10
dbg_tx00_sync_header[1:0]	2
dbg_tx00_data[31:0]	00000000
dbg_rx00_sync_header[1:0]	0
dbg_rx00_data[31:0]	00000000
dbg_rx00_data_valid	1
dbg_rx00_start_block	0
dbg_tx00_data_valid	1
dbg_tx00_start_block	0
store_ltssm	0

```
Updated at: 2019-Jun-12 17:51:07
```

Settings - hw_ila_1

Window data depth:	1024
Trigger position in window:	512

General Settings

Refresh rate:	500	ms

Status - hw_ila_1

Core status

Capture status - Window 1 of 1

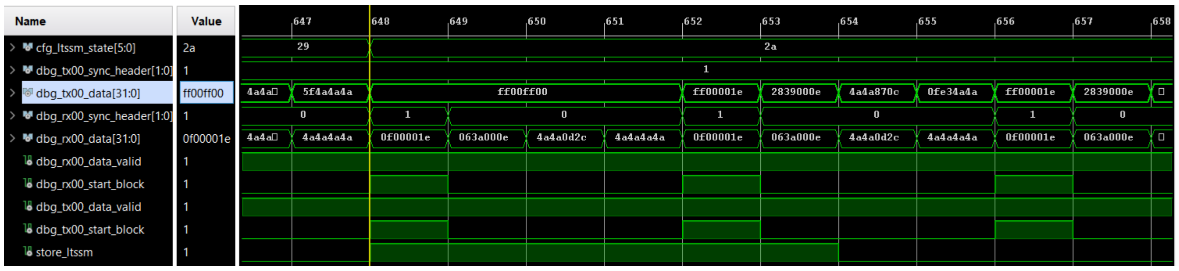
Window sample 512 of 1024

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value	Port	Comparator Usage
store_ltssm	==	[B]	R	probe9[0]	1 of 1

Figure 96 - Run trigger

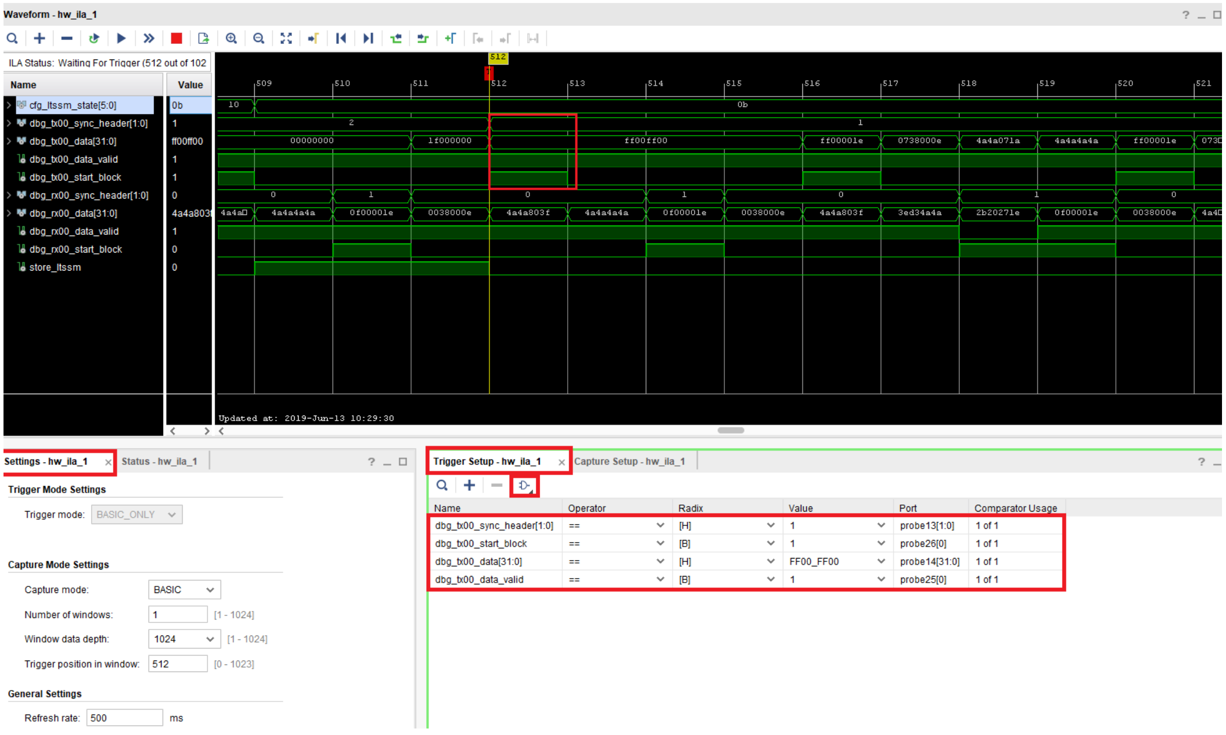
LTSSM state at recovery equalization **[2a]**: (See Appendix C: LTSSM State)



Name	Value
cfg_ltssm_state[5:0]	2a
dbg_tx00_sync_header[1:0]	1
dbg_tx00_data[31:0]	ff00ff00
dbg_rx00_sync_header[1:0]	1
dbg_rx00_data[31:0]	0f00001e
dbg_rx00_data_valid	1
dbg_rx00_start_block	1
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
store_ltssm	1

Figure 97 - Capture of hw_ila_1

LTSSM state at Electrical Idle Exit Ordered Set (EIEOS) with data value of **[FF00_FF00]**:


Waveform - hw_ila_1

ILA Status: Waiting For Trigger (512 out of 102

Name	Value
cfg_ltssm_state[5:0]	0b
dbg_tx00_sync_header[1:0]	1
dbg_tx00_data[31:0]	ff00ff00
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
dbg_rx00_sync_header[1:0]	0
dbg_rx00_data[31:0]	4a4a803f
dbg_rx00_data_valid	1
dbg_rx00_start_block	0
store_ltssm	0

```
Updated at: 2019-Jun-13 10:29:30
```

Settings - hw_ila_1 | Status - hw_ila_1

Trigger Mode Settings

Trigger mode:	BASIC_ONLY

Capture Mode Settings

Capture mode:	BASIC	
Number of windows:	1	[1 - 1024]
Window data depth:	1024	[1 - 1024]
Trigger position in window:	512	[0 - 1023]

General Settings

Refresh rate:	500	ms

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value	Port	Comparator Usage
dbg_tx00_sync_header[1:0]	==	[H]	1	probe13[1:0]	1 of 1
dbg_tx00_start_block	==	[B]	1	probe26[0]	1 of 1
dbg_tx00_data[31:0]	==	[H]	FF00_FF00	probe14[31:0]	1 of 1
dbg_tx00_data_valid	==	[B]	1	probe25[0]	1 of 1

Figure 98 - Capture of hw_ila_1

The figure below shows the Start Data Stream (SDS) with a data value of [**5555_55E1]**.

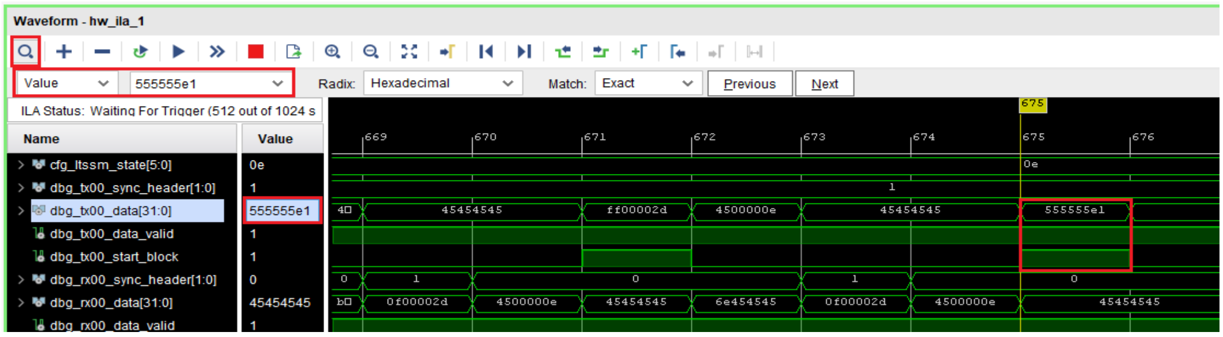


Figure 99 - Start of data stream

The diagram below shows the Start of DLLP Packet (SDP) **[ACF0]** across the multiple lanes.

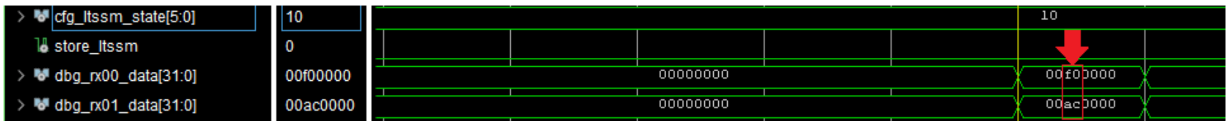
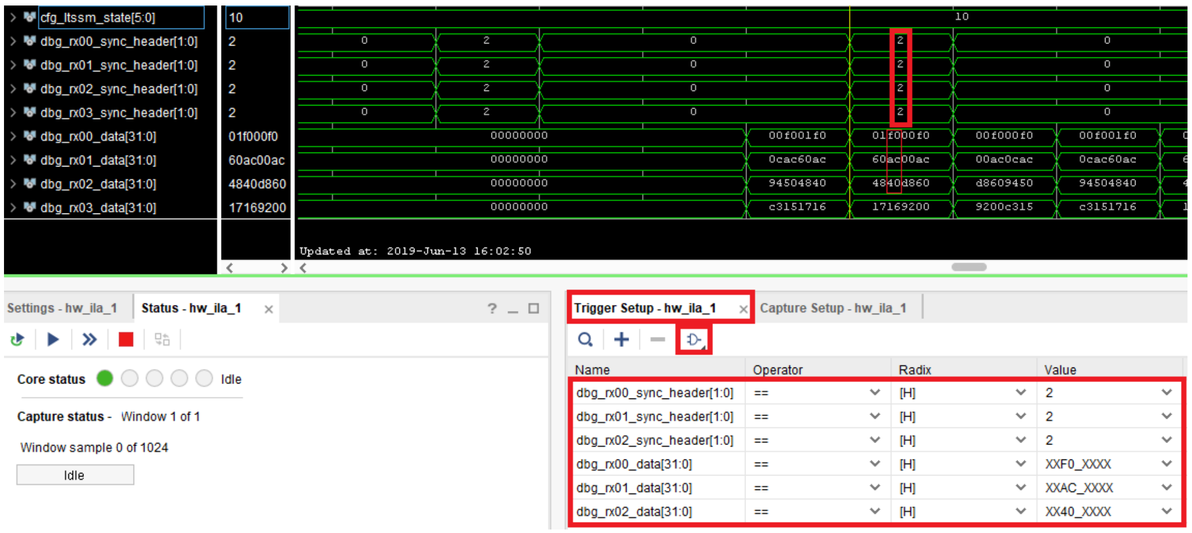


Figure 100 - Start of DLLP Packet (SDP)

Triggering on InitFC1-P (Initial Flow Control Credit for Posted Data) at 40h in **[AC_F0_40]**:

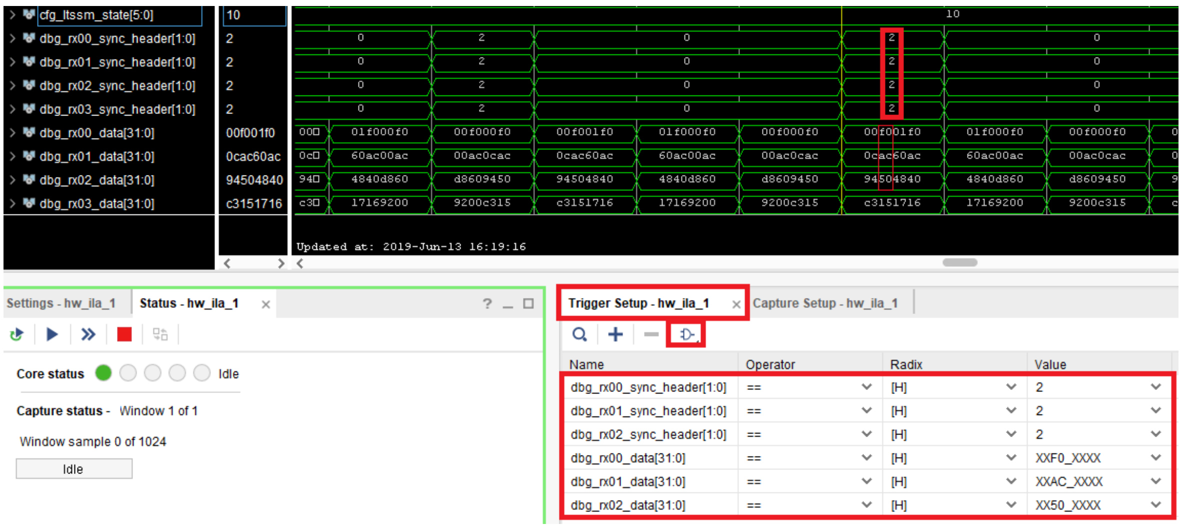


cfg_ltssm_state[5:0] 10
dbg_rx00_sync_header[1:0] 2
dbg_rx01_sync_header[1:0] 2
dbg_rx02_sync_header[1:0] 2
dbg_rx03_sync_header[1:0] 2
dbg_rx00_data[31:0] 01f000f0
dbg_rx01_data[31:0] 60ac00ac
dbg_rx02_data[31:0] 4840d860
dbg_rx03_data[31:0] 17169200
Updated at: 2019-Jun-13 16:02:50
Settings - hw_ila_1
Status - hw_ila_1
Core status Idle
Capture status - Window 1 of 1
Window sample 0 of 1024
Idle
Trigger Setup - hw_ila_1
Capture Setup - hw_ila_1

Name	Operator	Radix	Value
dbg_rx00_sync_header[1:0]	==	[H]	2
dbg_rx01_sync_header[1:0]	==	[H]	2
dbg_rx02_sync_header[1:0]	==	[H]	2
dbg_rx00_data[31:0]	==	[H]	XXF0_XXXX
dbg_rx01_data[31:0]	==	[H]	XXAC_XXXX
dbg_rx02_data[31:0]	==	[H]	XX40_XXXX

Figure 101 - InitFC1-P at 40h

Triggering on InitFC1-NP (Initial Flow Control Credit for Non-Posted Data) at 50h in **[AC_F0_50]**:



Name	Value
cfg_ltssm_state[5:0]	10
dbg_rx00_sync_header[1:0]	2
dbg_rx01_sync_header[1:0]	2
dbg_rx02_sync_header[1:0]	2
dbg_rx03_sync_header[1:0]	2
dbg_rx00_data[31:0]	00f001f0
dbg_rx01_data[31:0]	0cac60ac
dbg_rx02_data[31:0]	94504840
dbg_rx03_data[31:0]	c3151716

```
Updated at: 2019-Jun-13 16:19:16
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status ● Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

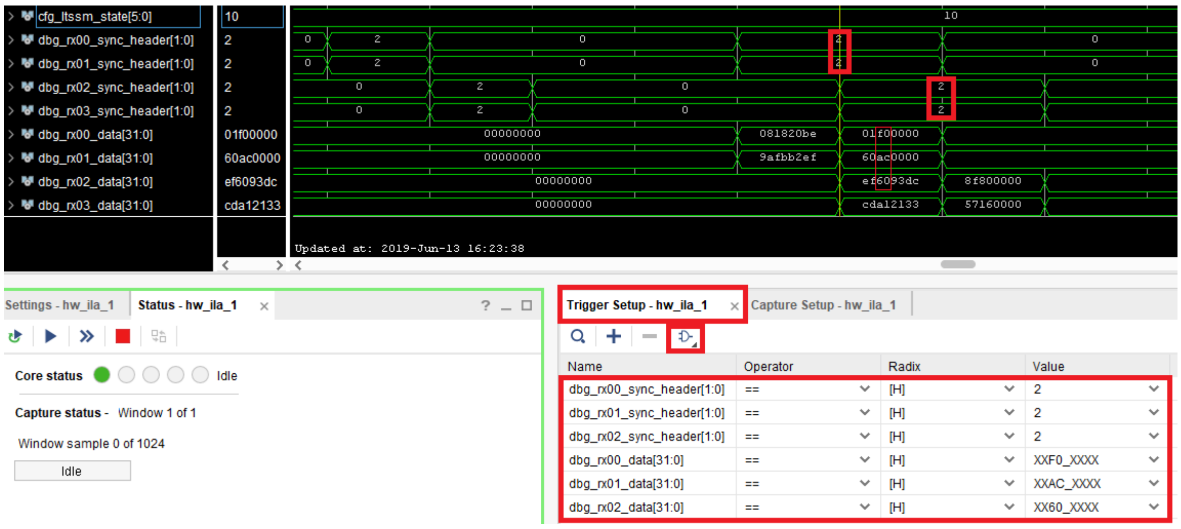
Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value
dbg_rx00_sync_header[1:0]	==	[H]	2
dbg_rx01_sync_header[1:0]	==	[H]	2
dbg_rx02_sync_header[1:0]	==	[H]	2
dbg_rx00_data[31:0]	==	[H]	XXF0_XXXX
dbg_rx01_data[31:0]	==	[H]	XXAC_XXXX
dbg_rx02_data[31:0]	==	[H]	XX50_XXXX

Figure 102 – InitFC1-NP at 50h

Triggering on InitFC1-Cpl (Initial Flow Control Credit for Completion) at 60h in **[AC_F0_60]**:



Name	Value
cfg_ltssm_state[5:0]	10
dbg_rx00_sync_header[1:0]	2
dbg_rx01_sync_header[1:0]	2
dbg_rx02_sync_header[1:0]	2
dbg_rx03_sync_header[1:0]	2
dbg_rx00_data[31:0]	01f00000
dbg_rx01_data[31:0]	60ac0000
dbg_rx02_data[31:0]	ef6093dc
dbg_rx03_data[31:0]	cda12133

```
Updated at: 2019-Jun-13 16:23:38
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status ● Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

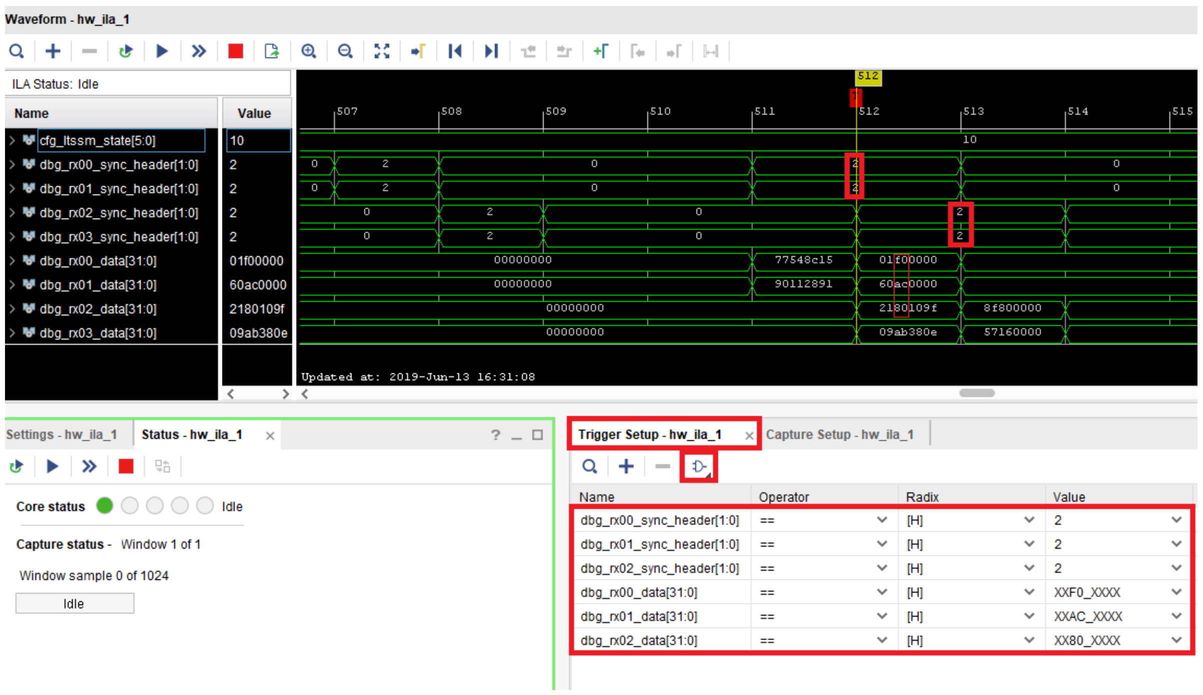
Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value
dbg_rx00_sync_header[1:0]	==	[H]	2
dbg_rx01_sync_header[1:0]	==	[H]	2
dbg_rx02_sync_header[1:0]	==	[H]	2
dbg_rx00_data[31:0]	==	[H]	XXF0_XXXX
dbg_rx01_data[31:0]	==	[H]	XXAC_XXXX
dbg_rx02_data[31:0]	==	[H]	XX60_XXXX

Figure 103 – InitFC1-Cpl at 60h

Triggering on UpdateFC-P at 80h in **[AC_F0_80]**:



Waveform - hw_ila_1

ILA Status: Idle

Name	Value
cfg_ltssm_state[5:0]	10
dbg_rx00_sync_header[1:0]	2
dbg_rx01_sync_header[1:0]	2
dbg_rx02_sync_header[1:0]	2
dbg_rx03_sync_header[1:0]	2
dbg_rx00_data[31:0]	01f00000
dbg_rx01_data[31:0]	60ac0000
dbg_rx02_data[31:0]	2180109f
dbg_rx03_data[31:0]	09ab380e

```
Updated at: 2019-Jun-13 16:31:08
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status ● Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

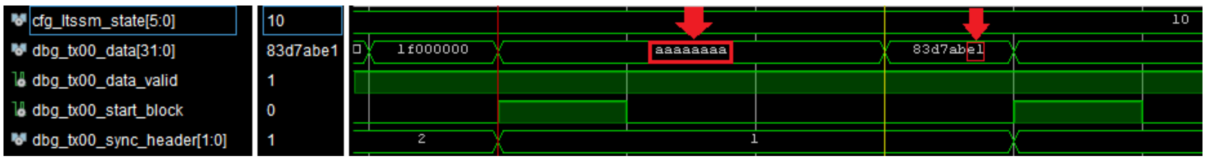
Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value
dbg_rx00_sync_header[1:0]	==	[H]	2
dbg_rx01_sync_header[1:0]	==	[H]	2
dbg_rx02_sync_header[1:0]	==	[H]	2
dbg_rx00_data[31:0]	==	[H]	XXF0_XXXX
dbg_rx01_data[31:0]	==	[H]	XXAC_XXXX
dbg_rx02_data[31:0]	==	[H]	XX80_XXXX

Figure 104 - Update FC-P at 80h

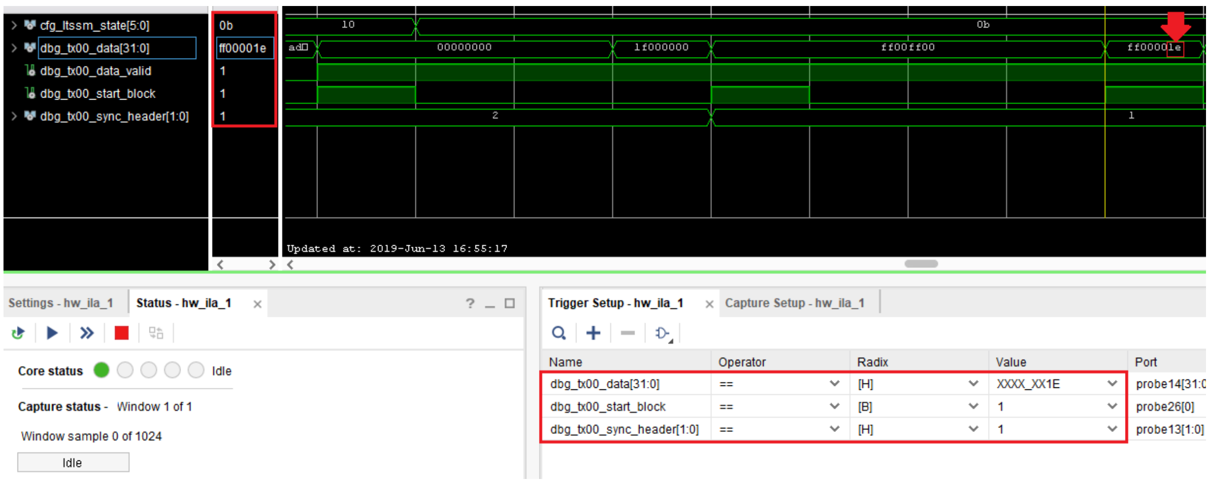
The figure below shows the **SKP** ordered set [**AAAA_AAAA**]and the **SKP_END** symbol [XXXX_XX**E1**].



Name	Value
cfg_ltssm_state[5:0]	10
dbg_tx00_data[31:0]	83d7abe1
dbg_tx00_data_valid	1
dbg_tx00_start_block	0
dbg_tx00_sync_header[1:0]	1

Figure 105 - SKP & SKP_END

The diagram shows the TS1 ordered set on a Gen3 link [XXXX_XX**1E**].



Name	Value
cfg_ltssm_state[5:0]	0b
dbg_tx00_data[31:0]	ff00001e
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
dbg_tx00_sync_header[1:0]	1

```
Updated at: 2019-Jun-13 16:55:17
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status ● Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value	Port
dbg_tx00_data[31:0]	==	[H]	XXXX_XX1E	probe14[31:0
dbg_tx00_start_block	==	[B]	1	probe26[0]
dbg_tx00_sync_header[1:0]	==	[H]	1	probe13[1:0]

Figure 106 - TS1 of Gen3 at 1Eh

The figure below shows the Symbol-4 which contains 0E in [XXXX_XX**0E**].

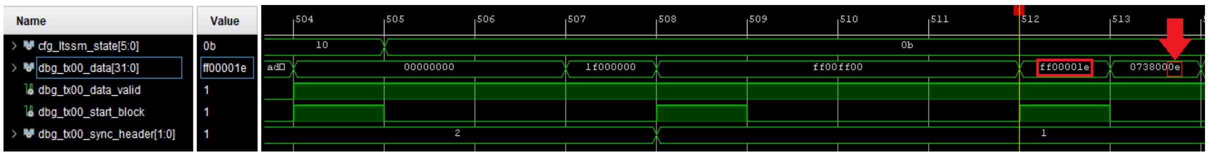
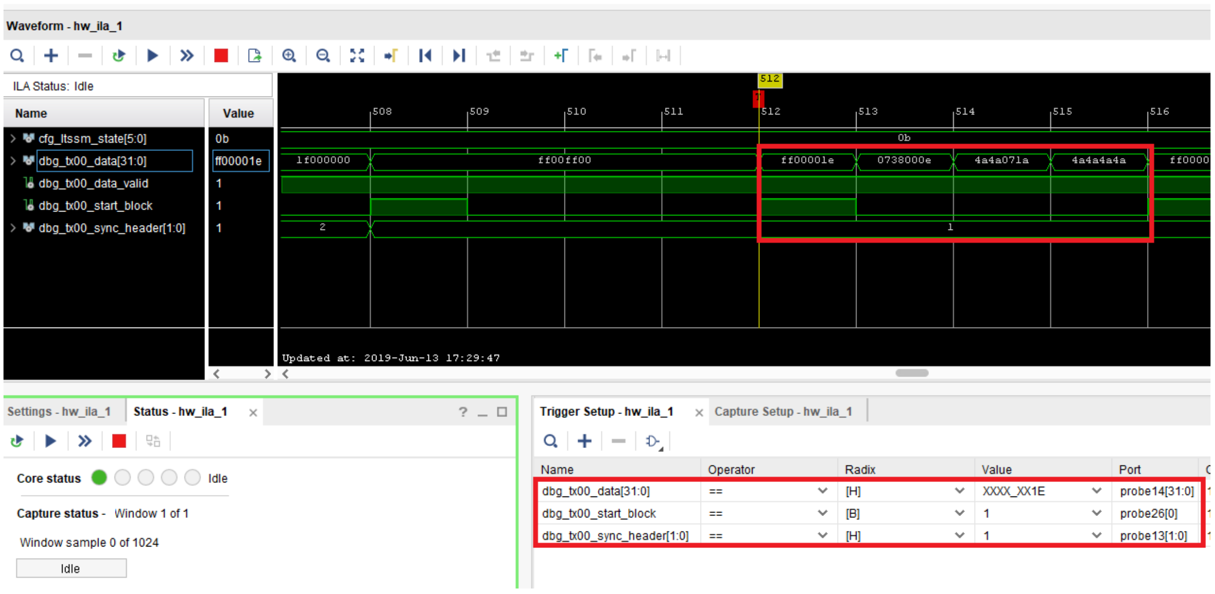

Name	Value
cfg_ltssm_state[5:0]	0b
dbg_tx00_data[31:0]	ff00001e
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
dbg_tx00_sync_header[1:0]	1

Figure 107- Symbol-4 at 0Eh

The waveform below shows Gen3 TS1 ordered set [XXXX_XX**1E**].


Waveform - hw_ila_1

ILA Status: Idle

Name	Value
cfg_ltssm_state[5:0]	0b
dbg_tx00_data[31:0]	ff00001e
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
dbg_tx00_sync_header[1:0]	1

```
Updated at: 2019-Jun-13 17:29:47
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

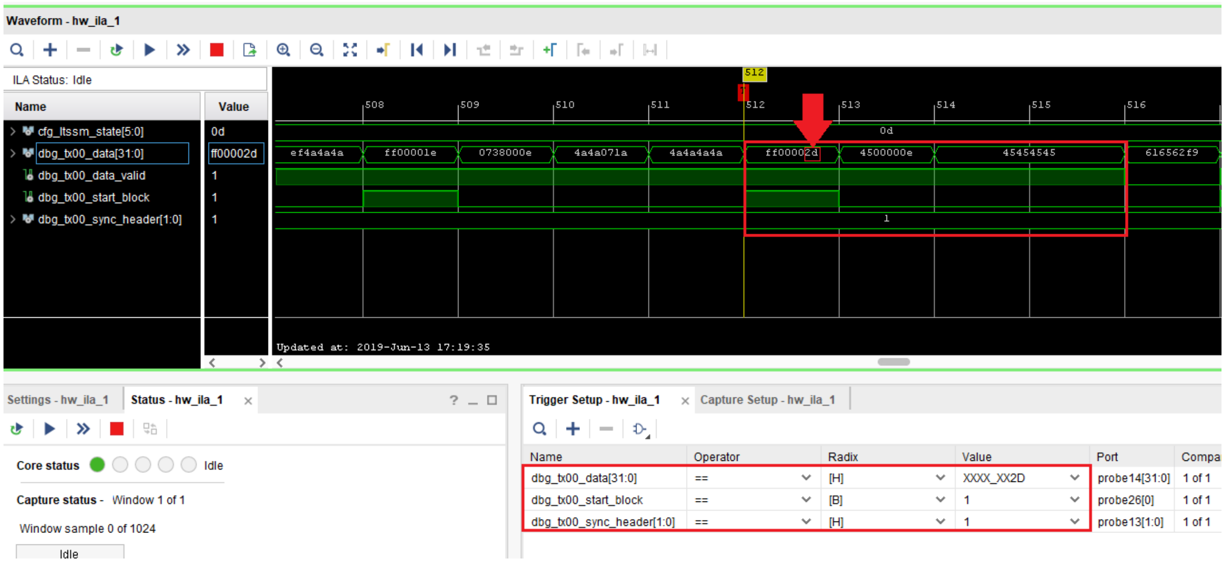
Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

Name	Operator	Radix	Value	Port
dbg_tx00_data[31:0]	==	[H]	XXXX_XX1E	probe14[31:0]
dbg_tx00_start_block	==	[B]	1	probe26[0]
dbg_tx00_sync_header[1:0]	==	[H]	1	probe13[1:0]

Figure 108 - Gen3 TS1 ordered set

The diagram below shows Gen3 TS2 ordered set [XXXX_XX**2D**].


Waveform - hw_ila_1

ILA Status: Idle

Name	Value
cfg_ltssm_state[5:0]	0d
dbg_tx00_data[31:0]	ff00002d
dbg_tx00_data_valid	1
dbg_tx00_start_block	1
dbg_tx00_sync_header[1:0]	1

```
Updated at: 2019-Jun-13 17:19:35
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

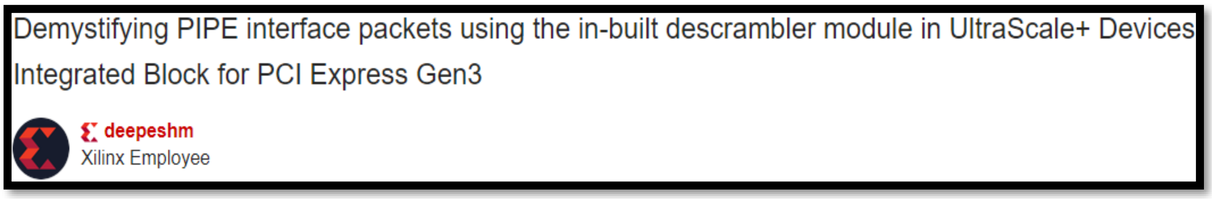
Name	Operator	Radix	Value	Port	Compa
dbg_tx00_data[31:0]	==	[H]	XXXX_XX2D	probe14[31:0]	1 of 1
dbg_tx00_start_block	==	[B]	1	probe26[0]	1 of 1
dbg_tx00_sync_header[1:0]	==	[H]	1	probe13[1:0]	1 of 1

Figure 109 - TS2 Gen3 ordered set

Please refer to the link below entitled “Demystifying PIPE interface packets using the in-built descrambler module in UltraScale+ Devices Integrated Block for PCI Express Gen3” for further analysis.

https://forums.xilinx.com/t5/Design-and-Debug-Techniques-Blog/Demystifying-PIPE-interface-packets-using-the-in-built/ba-

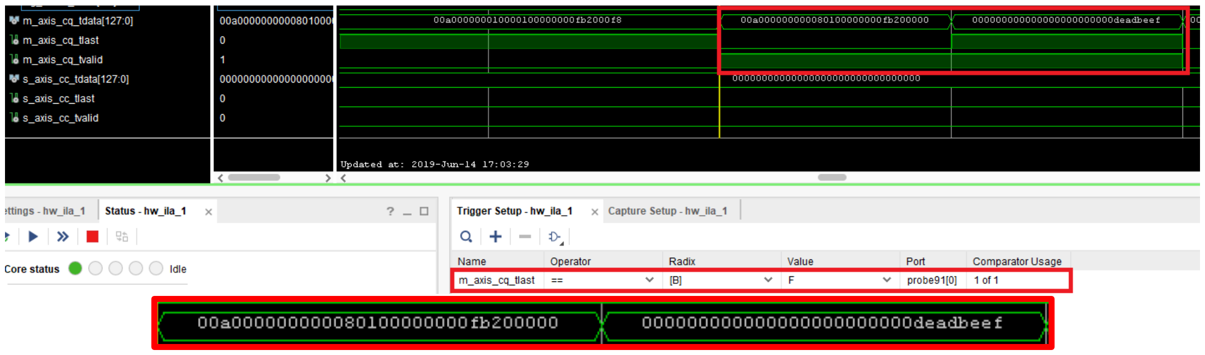
p/980246?fbclid=IwAR1tWreaT71aq_gePCfohJY2Dpe4_EfdIBzt3yHqidY-Tzsue9S1QJYurDc


Demystifying PIPE interface packets using the in-built descrambler module in UltraScale+ Devices Integrated Block for PCI Express Gen3

deepeshm
Xilinx Employee

Figure 110 - Demystifying PIPE

The waveform below shows that the Endpoint receives a memory write request from **m_axi_cq_tdata**.


```
m_axis_cq_tdata[127:0]   00a0000000000801000
m_axis_cq_tlast          0
m_axis_cq_tvalid         1
s_axis_cc_tdata[127:0]   0000000000000000000
s_axis_cc_tlast          0
s_axis_cc_tvalid         0

00a00000000100001000000000fb2000f8
00a0000000000801000000000fb200000
000000000000000000000000deadbeef
00000000000000000000000000000000

Updated at: 2019-Jun-14 17:03:29
```

Core status Idle

Name	Operator	Radix	Value	Port	Comparator Usage
m_axis_cq_tlast	==	[B]	F	probe91[0]	1 of 1

Figure 111 - Write request to endpoint device (CQ - Completer reQuest)

[127:0] 00a0_0001_0000_0801_0000_0000_fb20_0000

[1:0]	00b → Address Type
[63:2]	0000_0000_fb20_0000h → Address
[74:64]	000_0000_0001b →DWORD Count
[78:75]	000_1b → Reg Type
[87:80]	0000_0000b → Device/Function (Requester ID)
[95:88]	0000_0000b → Bus (Requester ID)
[103:96]	0000_0001b → Tag
[111:104]	0000_0000b → Target Function
[114:112]	000b → BAR ID
[120:115]	0_1010_0000b → BAR Aperture

[127:0] 0000_0000_0000_0000_0000_0000_dead_beef → This is the data.

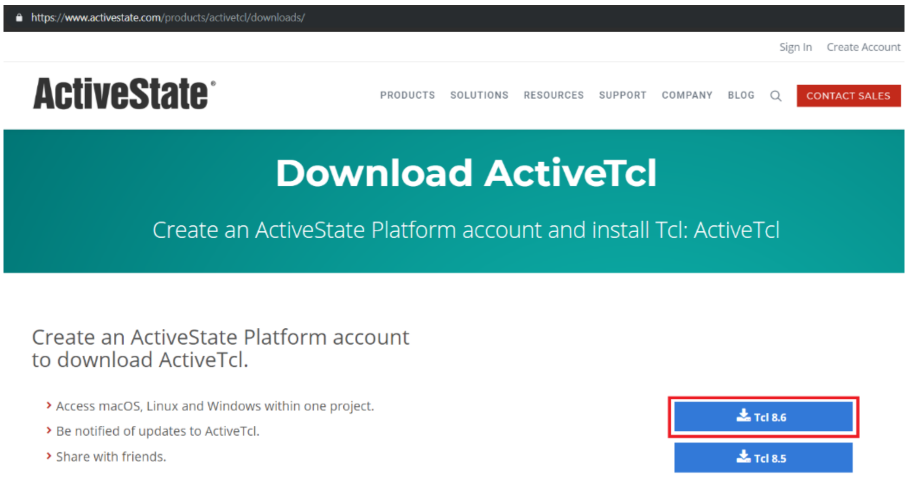
Appendix A: Tcl Console Result of test_rd.tcl

```
source
C:/FILES/project/vcu118/jtag_debugger_1/pcie4_uscale_plus_0_ex/pcie4_uscale
_plus_0_ex.srcs/sources_1/ip/pcie4_uscale_plus_0/pcie4_uscale_plus_0/pcie_d
ebugger/test_rd.tcl

# proc get_static_info {} {
# #   puts "Read static information: "
#     set filename "pcie_debug_static_info.dat"
#         set fh [open $filename w]
#     set txn_cnt 8
#     set txn_base_addr 0x0001fff0
#
#     for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
# #         puts "Read Address 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           run_hw_axi [get_hw_axi_txns rd_txn_lite]
#           #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"
#           }
#     close $fh
# }
# proc get_reset_trc {} {
#     set filename "pcie_debug_rst_trc.dat"
#         set fh [open $filename w]
#     #puts "Read reset trace"
#     set txn_cnt 8
#     set txn_base_addr 0x00002000
#     for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#           #puts "Read Address 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           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"
#           }
#     close $fh
# }
# proc get_ltssm_trc {} {
#     set filename "pcie_debug_ltssm_trc.dat"
#         set fh [open $filename w]
#     #puts "Read ltssm trace"
#     set txn_cnt 512
#     set txn_base_addr 0x00001000
#     for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#           #puts "Read LTSSM TRACE 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           run_hw_axi -quiet [get_hw_axi_txns rd_txn_lite]
#           set trans [get_property DATA [get_hw_axi_txns rd_txn_lite]]
#
#             #set temp_addr [format %.4X [expr $txn_base_addr + $i +1]]
#           #set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           #run_hw_axi -quiet [get_hw_axi_txns rd_txn_lite]
#           #set trans_dur [get_property DATA [get_hw_axi_txns rd_txn_lite]]
#
#           puts $fh "0x$trans"
#           #0x$trans_dur"
#           }
#     close $fh
# }
# proc get_rxdet_trc { lane_index } {
# #   puts "Read RX detection trace"
#     set filename "pcie_debug_rxdet_trc.dat"
#         set fh [open $filename w]
#     set txn_cnt 4
#     set txn_base_addr [expr 0x00003000 + (${lane_index} <<4)]
#
#     for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#           #puts "Read RX Detect trace 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           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"
#           }
#     close $fh
# }
# proc get_other_info {} {
# #   puts "Read RX detection trace"
#     set filename "pcie_debug_info_trc.dat"
#         set fh [open $filename w]
#     set txn_cnt 3
#     set txn_base_addr 0x00004000
#
#     for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#           #puts "Read RX Detect trace 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           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"
#           }
#     close $fh
# }
WARNING: [Labtoolstcl 44-227] No matching hw_axi_txns were found
# set myread [llength [get_hw_axi_txns rd_txn_lite] ]
# if { $myread == 0 } {
#     create_hw_axi_txn rd_txn_lite [get_hw_axis hw_axi_1] -address 0001fff0
-type read
#     }
# get_static_info
INFO: [Labtoolstcl 44-481] READ DATA is: 041150ae
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
INFO: [Labtoolstcl 44-481] READ DATA is: 00000000
# get_reset_trc
# get_other_info
# get_ltssm_trc
# set fp [open "pcie_debug_info_trc.dat" r]
# set count 0
# while {[gets $fp line]!=-1} {
#     incr count
#         if {$count==1} {
#           set temp0 [expr $line & 0xFF]
#           set phy_lane [expr int($temp0)]
#           #[format "%02x" $temp0]
#           puts "phy_lane : $phy_lane"
#            }
#         if {$count==2} {
#           set temp1 [expr $line & 0xff]
#           set width [format "%02x" $temp1]
#           puts "width : $width"
#         }
#         if {$count==3} {
#           set temp2 [expr $line & 0xFF]
#           set speed [format "%02x" $temp2]
#           puts "speed : $speed"
#         }
#      }
phy_lane : 0
width : 00
speed : 00
```

```
# close $fp
# set filename "rxdet.dat"
# set fh [open $filename w]
# set txn_cnt 4
# set j 0
# while { $j < $phy_lane } {
#   for {set i 0 } { $i < $txn_cnt } { set i [expr $i +1] } {
#     set txn_base_addr [expr 0x00003000 + ($j <<4)]
#           set temp_addr [format %.4X [expr $txn_base_addr + $i]]
#           #puts "Read RX Detect trace 0x{$temp_addr}:"
#           set_property CMD.ADDR $temp_addr [get_hw_axi_txns rd_txn_lite]
#           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
# }
# close $fh
```

Appendix B: Tcl/Tk Package Installation Guide

Download the appropriate platform of ActiveStateTcl package from the link below
https://www.activestate.com/products/activetcl/downloads/



Download ActiveTcl

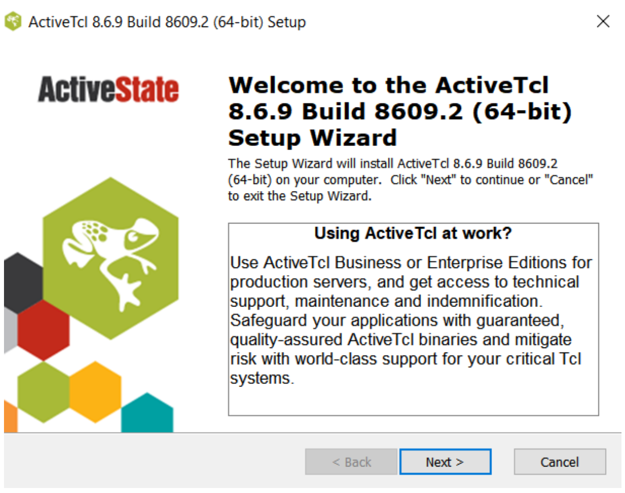
Create an ActiveState Platform account and install Tcl: ActiveTcl

Create an ActiveState Platform account to download ActiveTcl.

- Access macOS, Linux and Windows within one project.
- Be notified of updates to ActiveTcl.
- Share with friends.

Figure 112 - ActiveStateTcl package download site

In this guide, Windows platform is used. Install the application file. Click "Next".



Welcome to the ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup Wizard

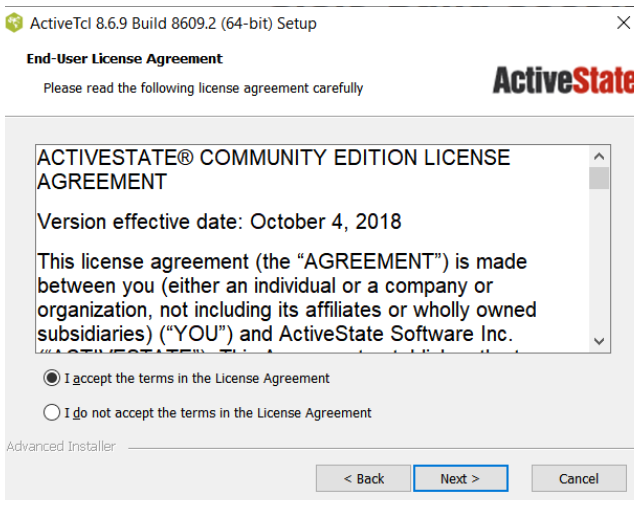
The Setup Wizard will install ActiveTcl 8.6.9 Build 8609.2 (64-bit) on your computer. Click "Next" to continue or "Cancel" to exit the Setup Wizard.

Using ActiveTcl at work?

Use ActiveTcl Business or Enterprise Editions for production servers, and get access to technical support, maintenance and indemnification. Safeguard your applications with guaranteed, quality-assured ActiveTcl binaries and mitigate risk with world-class support for your critical Tcl systems.

Figure 113 - Setup wizard of ActiveTcl

Select the radio button for “I accept the terms in the license agreement”. Click “Next”.



ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup

End-User License Agreement

Please read the following license agreement carefully

ACTIVESTATE® COMMUNITY EDITION LICENSE AGREEMENT

Version effective date: October 4, 2018

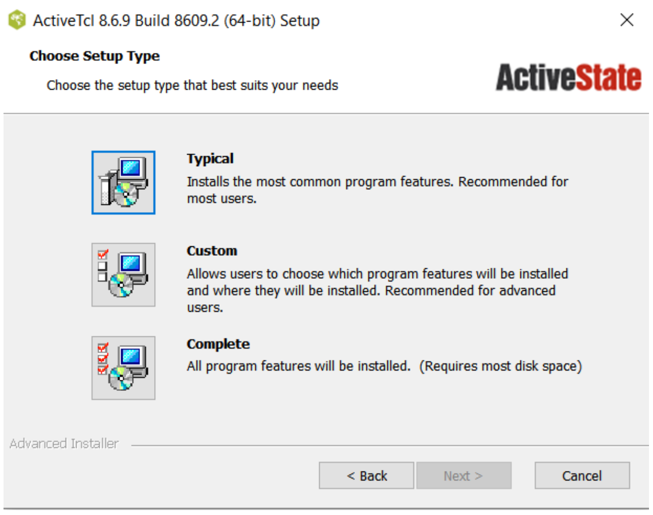
This license agreement (the “AGREEMENT”) is made between you (either an individual or a company or organization, not including its affiliates or wholly owned subsidiaries) (“YOU”) and ActiveState Software Inc.

(●) I accept the terms in the License Agreement

( ) I do not accept the terms in the License Agreement

Figure 114 - ActiveTcl license agreement

Choose a setup type. Click “Next”.



ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup

Choose Setup Type

Choose the setup type that best suits your needs

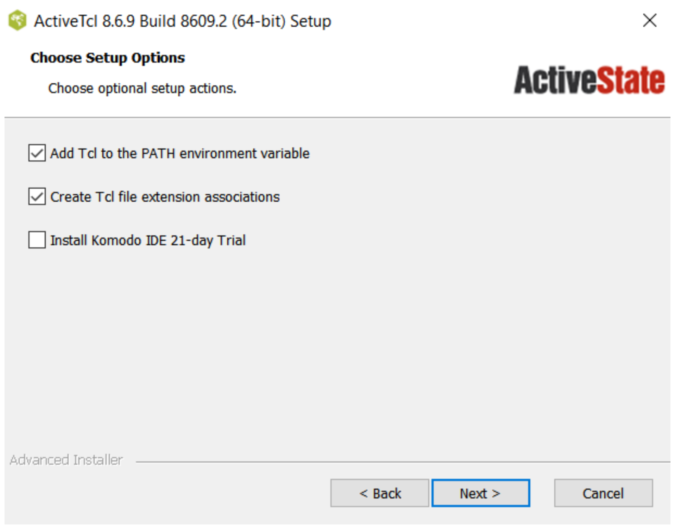
Typical
Installs the most common program features. Recommended for most users.

Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

Complete
All program features will be installed. (Requires most disk space)

Figure 115 - Setup type

Choose additional setup options. Click “Next”.



ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup

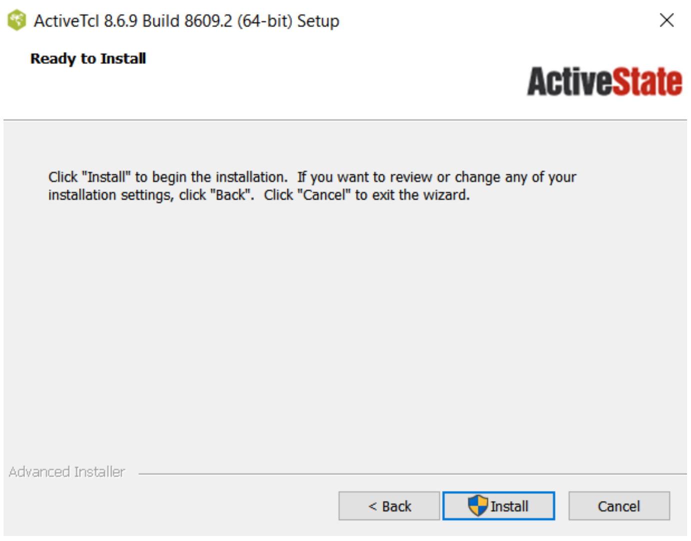
Choose Setup Options

Choose optional setup actions.

- [✓] Add Tcl to the PATH environment variable
- [✓] Create Tcl file extension associations
- [ ] Install Komodo IDE 21-day Trial

Figure 116 - Additional options

Click “Install”.



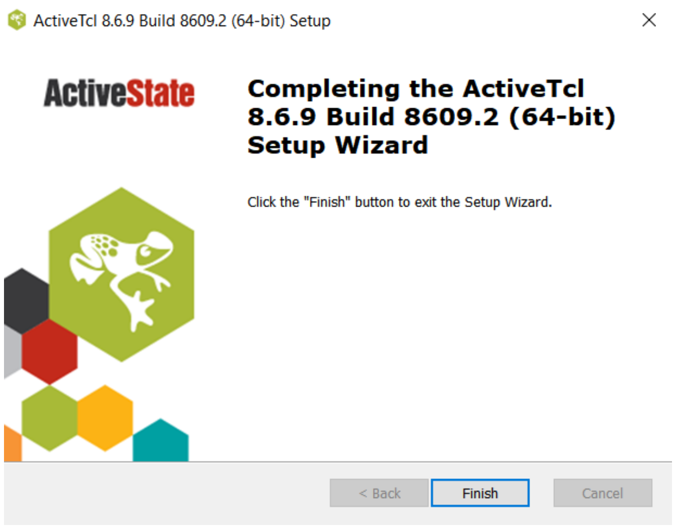
ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup

Ready to Install

Click "Install" to begin the installation. If you want to review or change any of your installation settings, click "Back". Click "Cancel" to exit the wizard.

Figure 117 - Install ActiveTcl

Click “Finish”.



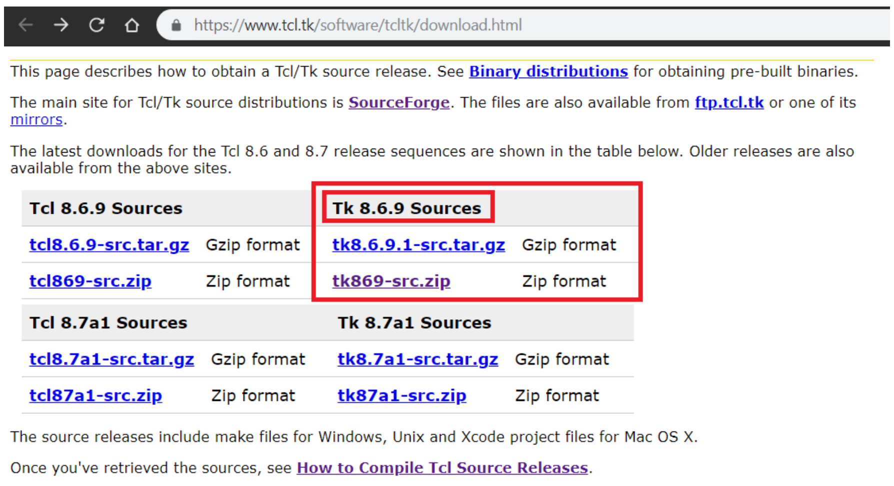
ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup

Completing the ActiveTcl 8.6.9 Build 8609.2 (64-bit) Setup Wizard

Click the "Finish" button to exit the Setup Wizard.

Figure 118 - Installation completed

Download the Tk sources from the link below:
https://www.tcl.tk/software/tcltk/download.html


https://www.tcl.tk/software/tcltk/download.html

This page describes how to obtain a Tcl/Tk source release. See Binary distributions for obtaining pre-built binaries.

The main site for Tcl/Tk source distributions is SourceForge. The files are also available from ftp.tcl.tk or one of its mirrors.

The latest downloads for the Tcl 8.6 and 8.7 release sequences are shown in the table below. Older releases are also available from the above sites.

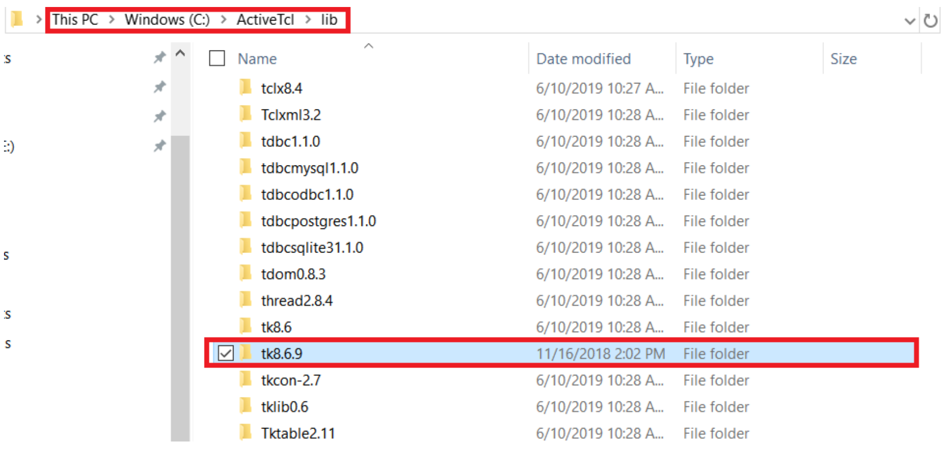
Tcl 8.6.9 Sources		Tk 8.6.9 Sources	
tcl8.6.9-src.tar.gz	Gzip format	tk8.6.9.1-src.tar.gz	Gzip format
tcl869-src.zip	Zip format	tk869-src.zip	Zip format
Tcl 8.7a1 Sources		**Tk 8.7a1 Sources**	
tcl8.7a1-src.tar.gz	Gzip format	tk8.7a1-src.tar.gz	Gzip format
tcl87a1-src.zip	Zip format	tk87a1-src.zip	Zip format

The source releases include make files for Windows, Unix and Xcode project files for Mac OS X.

Once you've retrieved the sources, see How to Compile Tcl Source Releases.

Figure 119 - Tk source

Extract the Tk source (tk8.6.9) folder from the downloaded zip file into the ActiveTcl lib folder.

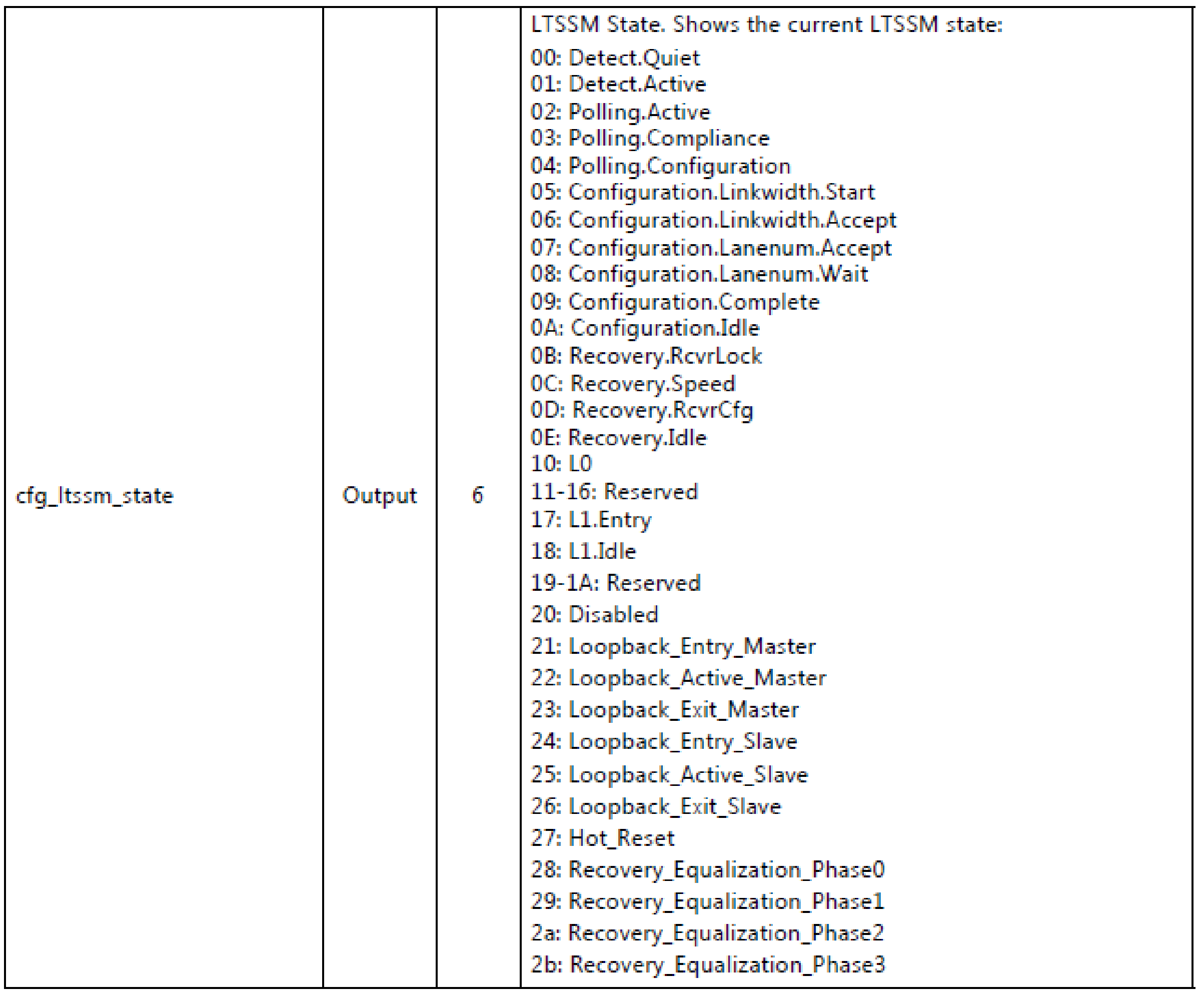

This PC > Windows (C:) > ActiveTcl > lib

Name	Date modified	Type	Size
tclx8.4	6/10/2019 10:27 A...	File folder	
Tclxml3.2	6/10/2019 10:28 A...	File folder	
tdbc1.1.0	6/10/2019 10:28 A...	File folder	
tdbcmysql1.1.0	6/10/2019 10:28 A...	File folder	
tdbcodbc1.1.0	6/10/2019 10:28 A...	File folder	
tdbcpostgres1.1.0	6/10/2019 10:28 A...	File folder	
tdbcsqlite31.1.0	6/10/2019 10:28 A...	File folder	
tdom0.8.3	6/10/2019 10:28 A...	File folder	
thread2.8.4	6/10/2019 10:28 A...	File folder	
tk8.6	6/10/2019 10:28 A...	File folder	
tk8.6.9	11/16/2018 2:02 PM	File folder	
tkcon-2.7	6/10/2019 10:28 A...	File folder	
tklib0.6	6/10/2019 10:28 A...	File folder	
Tktable2.11	6/10/2019 10:28 A...	File folder	

Figure 120 - Tk source

Appendix C: LTSSM State

The figure below shows the different link training state (LTSSM).



cfg_ltssm_state	Output	6	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 17: L1.Entry 18: L1.Idle 19-1A: Reserved 20: Disabled 21: Loopback_Entry_Master 22: Loopback_Active_Master 23: Loopback_Exit_Master 24: Loopback_Entry_Slave 25: Loopback_Active_Slave 26: Loopback_Exit_Slave 27: Hot_Reset 28: Recovery_Equalization_Phase0 29: Recovery_Equalization_Phase1 2a: Recovery_Equalization_Phase2 2b: Recovery_Equalization_Phase3

Figure 121 - LTSSM state from page 49 of Xilinx (PG213) UltraScale+ Devices Block for PCIe v1.1

References

1. (Xilinx Answer 68134) UltraScale and UltraScale+ FPGA Gen3 Integrated Block for PCI Express - Integrated Debugging Features and Usage Guide
2. (Xilinx Answer 71355) Vivado ILA Usage Guide for UltraScale FPGA Gen3 Integrated Block for PCI Express
3. (PG213) UltraScale+ Devices Integrated Block for PCI Express v1.1 LogiCORE IP Product Guide
4. Demystifying PIPE interface packets using the in-built descrambler module in UltraScale+ Devices Integrated Block for PCI Express Gen3 https://forums.xilinx.com/t5/Design-and-Debug-Techniques-Blog/Demystifying-PIPE-interface-packets-using-the-in-built/ba-p/980246?fbclid=IwAR1tWreaT71aq_gePCfohJY2Dpe4_EfdIBzt3yHqidY-Tzsue9S1QJYurDc
5. (UG908) Vivado Design Suite User Guide Programming and Debugging
6. (UG936) Vivado Design Suite Tutorial Programming and Debugging
7. Virtex UltraScale+ VCU118 Evaluation Kit https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0049-vcu118-evaluation-kit-hub.html
8. PCI Express® Base Specification Revision 3.0 November 10, 2010

Revision History

06/25/2019 - Initial release