ECP5 and ECP5-5G PCI Express Soft IP Ease of Use Guidelines

Technical Note

FPGA-TN-02045 1.2

January 2022

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
LPC	Lattice Parameter Configuration
LPF	Lattice Preference File
PCI	Peripheral Component Interconnect

1. Introduction

The ECP5™ and ECP5-5G™ PCI (Peripheral Component Interconnect) Express Endpoint Soft IP Ease of Use Guidelines Technical Note provides instructions on implementing the PCI Express® Endpoint IP Core for ECP5/ECP5-5G devices. It focuses on the entire process, from installing the soft IP to implementing the design and meeting timing closure. Information on how to resolve issues that are commonly encountered by designers in specific steps are provided.

For a complete and detailed guide on the PCI Express Endpoint IP Core, refer to PCI Express x1/x2/x4 Endpoint IP Core User Guide (FPGA-IPUG-02009).

2. Requirements

To use the PCI Express Endpoint IP Core, the following items are required:

- Lattice Diamond® 3.8 – Software to implement PCI Express Endpoint IP Core
- Synopsys® Synplify Pro® for Lattice I-2013.09L-SP1 – Synthesis tool

  Note: LSE currently not supported.
- Lattice Diamond® IP License for PCI Express Endpoint
- PC with a PCI Express 1.0/2.0 slot

  Note: Gen 1(ECP5) supports x1/x2/x4 while Gen 2(ECP5-5G) supports x1/x2

Other useful tools include:

- Aldec® Active HDL 10.3 – Simulation support

  Note: For Windows Only
- Mentor Graphics® ModelSim 10.2 – Simulation support

  Note: For Verilog Only
- ECP5/ECP5-5G Versa Development Kit – Contains three demos and an ECP5-5G Versa Board that use the PCI Express Endpoint IP Core

3. PCI Express Endpoint IP Core License

An IP license is required to fully access all the features of the PCI Express IP Core. Without a license, you may be able to download and generate the PCI Express IP Core. To access timing simulation, open the design in the Diamond tool, and generate the bitstreams, a license is required.

To request for a license, refer to the Design Software and IP page in the Lattice website.

4. Generating the PCI Express Endpoint IP Core using Clarity Designer

Detailed instructions on how to generate the PCI Express IP Core using Clarity Designer is provided in PCI Express x1/x2/x4 Endpoint IP Core User Guide (FPGA-IPUG-02009).

The following is an express installation guide.

4.1. Creating a New Project

To create a new project:

1. Open the Diamond tool.
2. Choose **File > New > Project**.
3. Name your project and add your design source files.
4. When selecting the device, make sure you select the family ECP5/ECP5UM5G with the corresponding device, performance grade and package type of your ECP5/ECP5-5G FPGA.
5. Select Synplify Pro as the synthesis tool.
   Note: PCIe 5G Endpoint Core currently does NOT support LSE.
6. Complete the creation of the new project.

4.2. Installing the PCI Express Endpoint IP Core

To install the PCI Express Endpoint IP Core:

1. Click the Clarity Designer button.
   Note: If you do not see the Clarity Designer button, it can also be opened from **Tools > Clarity Designer**. You can also right-click below the File tab and check **Tools** to bring up a Clarity Designer shortcut.
2. Browse to the desired location and enter a design name. The Clarity Designer screen is then displayed.
   Note: The design name is the name of the IP module that you will use later.
3. In the Lattice IP Server tab under the Catalog tab, choose **PCI Express 5G Endpoint** as shown in Figure 4.1.
4. Click the **Install PCI Express 5G Endpoint** button below the Clarity Designer tab. When done, the IP will be ready for use under the Catalog > Lattice IP tab as shown in Figure 4.2.

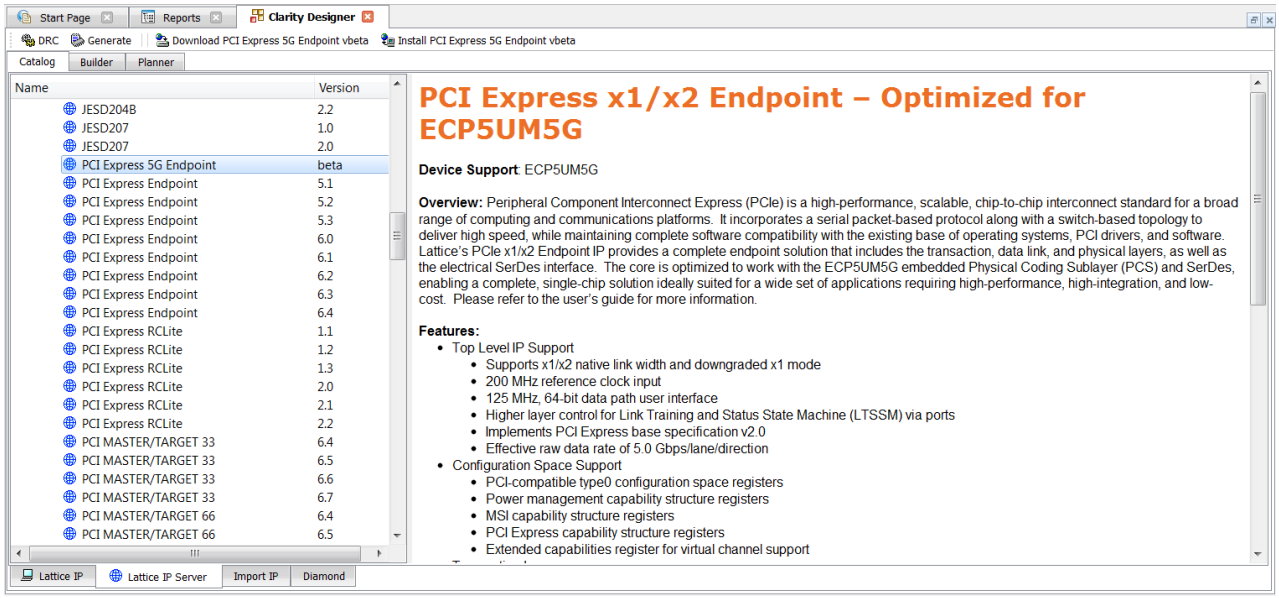

Figure 4.1. Location of the PCI Express 5G Endpoint Installer

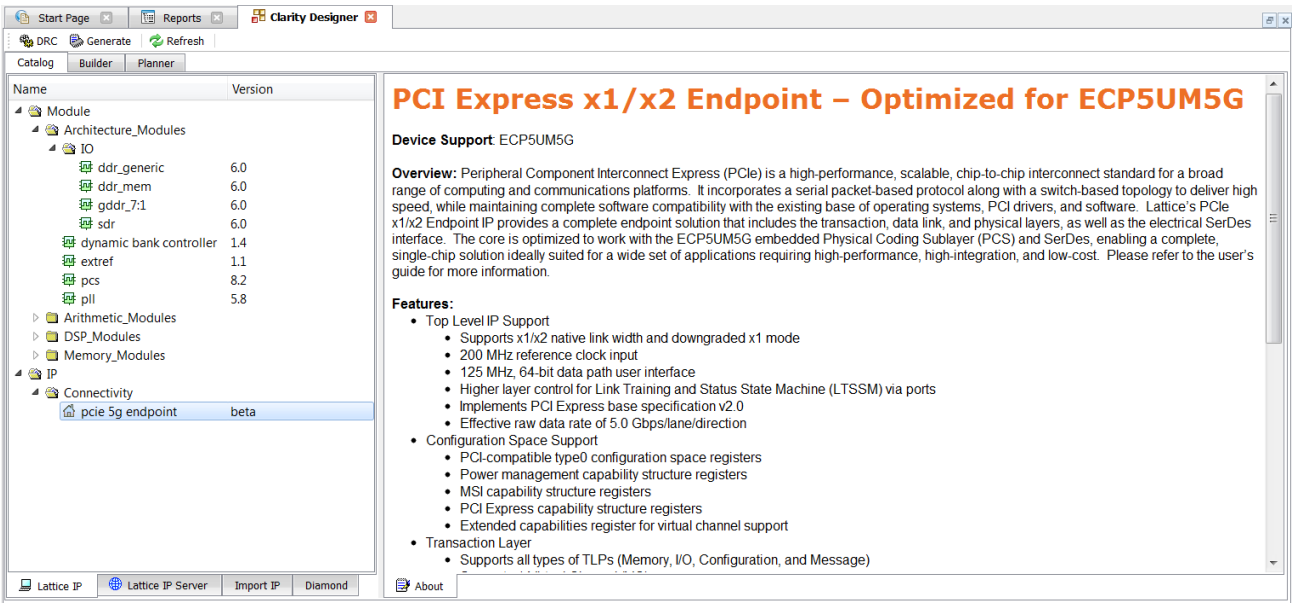

Figure 4.2. PCIe 5G Endpoint Successfully Installed

4.3. Creating and Generating PCIe IP in Clarity Designer

To create and generate the PCIe IP in Clarity Designer:

1. In the Catalog tab under Connectivity, double-click **pcie 5g endpoint** version beta for 5G devices.
   Note: If you do not see the IP, please refer to the Troubleshooting section.
2. Create an instance name and click **Customize**.
   Note: The instance name should be short as it will be added to the front of all the signals called in the "design_name.v" file. For example, "reset_n" will become "pcie_reset_n" if your instance name is "pcie".
3. A settings interface for the PCIe Endpoint IP is displayed. For a detailed description of the settings, refer to FPGA-IPUG-02009, PCI Express x1/x2/x4 Endpoint IP Core User Guide.
4. Apply the settings and click **Configure**. Check the generate log to confirm that there are no errors. Once you close the window, the IP instance is generated and is displayed in the Planner tab of Clarity Designer.
   Note: This may take a couple of minutes depending on your system performance. If you are uncertain of your PCIe Core settings, refer to the Lattice Parameter Configuration (LPC) file. It provides the values to be entered for Clarity Designer.
5. PCI Express also needs an EXTREF module to be connected for reference clocks so it can be shared across multiple IPs. In Clarity Designer, under Catalog > Module > Architecture_Modules, double click on **extref** and generate the instance.
6. After generating the two instances, click the Planner tab. The interface resulting interface is shown in Figure 4.3.
   Note: The PCI Express settings in the figure is for an x1 configuration.

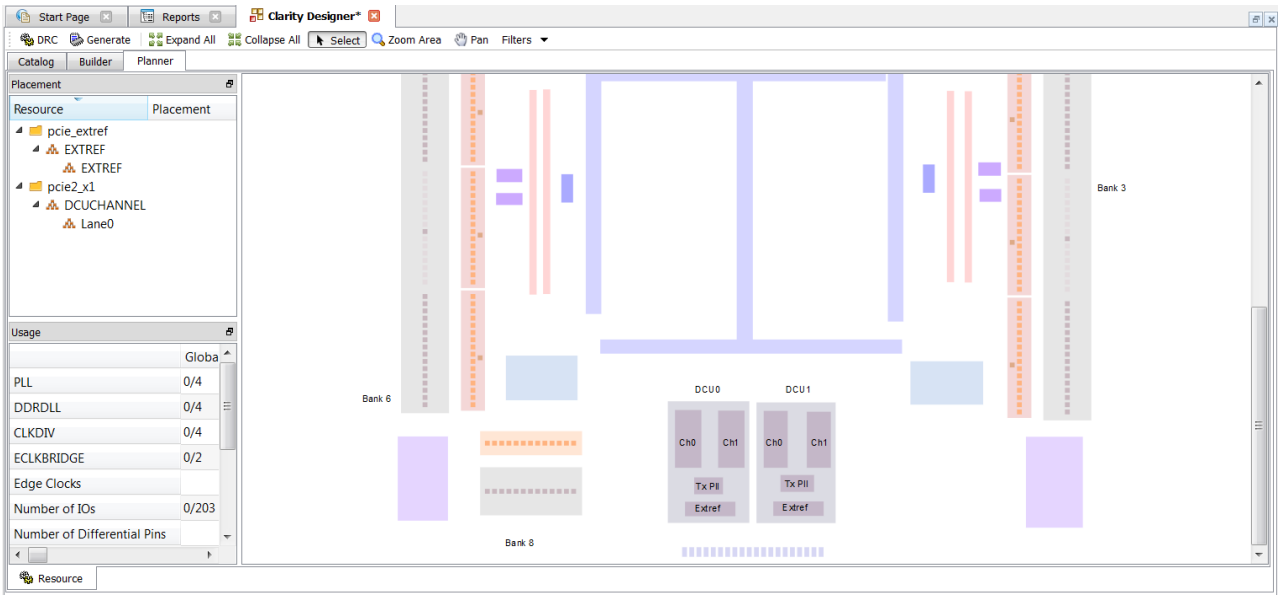

Figure 4.3. Clarity Designer Interface with PCI Express Endpoint and EXTREF IP Instances

7. Under DCUCHANNEL, drag **Lane0** to either Ch0 or Ch1 and to either DCU0 or DCU1 depending on your SerDes channel. If you selected x2 configuration, place lane0 and lane1 in the same DCU.
8. Drag the **EXTREF** instance to the EXTREF box. Your Clarity Designer interface should be similar to Figure 4.4. If you are not able to place a part, right-click on the part under the Planner tab and click **Reset**. You will be able to place the part again.

   Note: PCI Express Endpoint IP needs the ExtRef module to be connected for reference clocks which can be shared across multiple IPs. So it has to be generated outside the PCI Express endpoint IP.

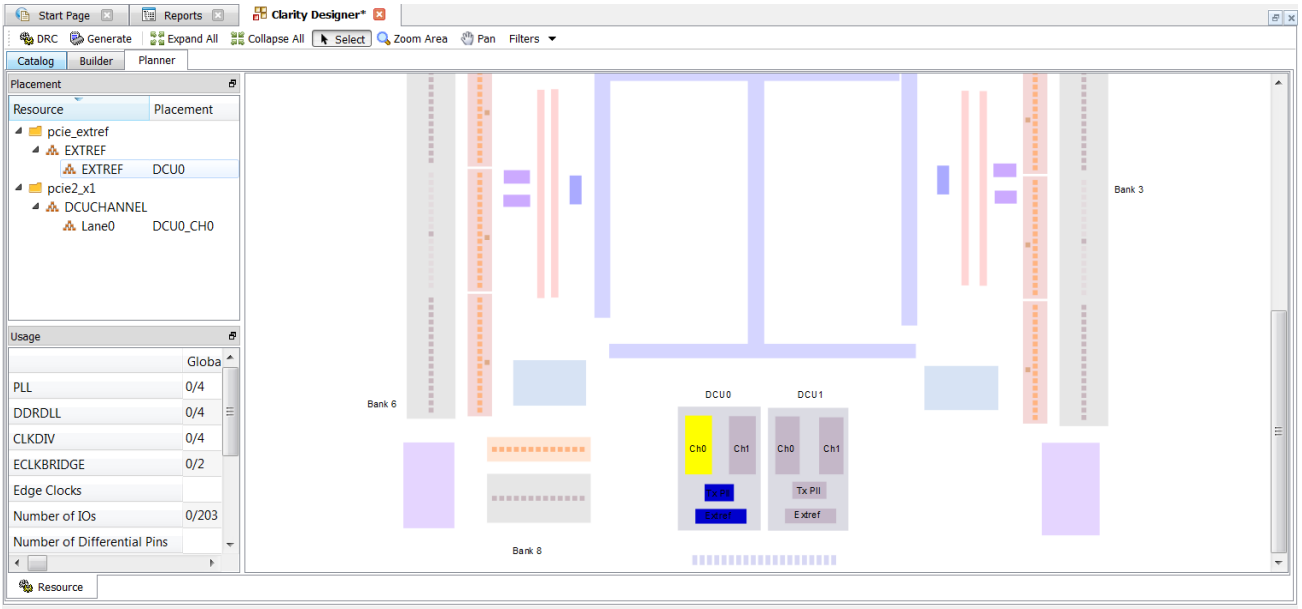

Figure 4.4. Clarity Designer Interface with PCI Express Endpoint and EXTREF Placed

9. After placing both instances, double-click on the channel and make sure the TX Setting and RX Setting are changed to **DCU0_EXTREF/DCU1_EXTREF** This is to link the ExtRef clock to PCI Express Endpoint IP instead of the Primary clock.

10. Click the **Generate** button. This generates the IP core as an .sbx file and places it in File List under Input Files. This sbx module is used to bring in the encrypted IP Core for the PCI Express Endpoint Core.
11. After generating the IP Core, Clarity Designer creates several files that are used throughout the design cycle. Table 4.1 shows some of the key files created by Clarity Designer and their use. Details on other specific files are described in FPGA-IPUG-02009, PCI Express x1/x2/x4 Endpoint IP Core User Guide.

Table 4.1. List of Important Files for the PCI Express Endpoint IP Core

File	Sim	Synthesis	Description
<username>.v	Yes		This file provides the PCI Express core for simulation. This file provides a module which instantiates the PCI Express core and the PIPE interface.
<username>_core_bb.v		Yes	This file provides the synthesis black box for the PCI express core.
<username>_beh.v	Yes		This file provides the front-end simulation library for the PCI Express core. This file is located in the pcie_eval/<user_name>/src/top directory.
pci_exp_params.v	Yes		This file provides the user options of the IP for the simulation model.
pci_exp_ddefines.v	Yes		This file provides parameters necessary for the simulation.
<username>_core/phy.ngo		Yes	This file provides the synthesized IP core used by the Diamond software. This file needs to be pointed to by the Build step by using the search path property.
<username>.lpc			This file contains the configuration options used to recreate or modify the core in the IPexpress tool.
pmi_*.ngo			These files contains the memories used by the IP core. These files need to be pointed to by the Build step by using the search path property.

You have successfully generated your PCI Express Endpoint IP Core.

5. Frequently Asked Questions

The following section provides information on how to resolve issues that you may encounter while creating the PCI Express Endpoint IP Core.

Q: After installing the IP, I do not see it listed under Connectivity.

A: Check to see if your project is for the corresponding IP and that it is using Synplify for synthesis. The PCI Express Endpoint IP Core is only compatible with ECP3/ECP5UM/ECP5UM5G devices and the PCI Express 5G Endpoint is only compatible with ECP5UM5G.

If this does not solve the problem, check your LatticeCore directory to see whether the IP was installed correctly. Check also your ipsetting.lst file under the lsc_env directory to see if it was correctly called. It should be named "pcie_5g_vbeta=C:\LatticeCore" for 5G devices or "pci_express_endpoint_v6.4=C:\LatticeCore" for non 5G devices.

Q: I want to edit my PCI Express Endpoint IP Core settings. Do I need to recreate a new IP Core from scratch?

A: No. Click **Clarity Designer > Planner** tab. Right-click on the soft IP core to display the option for configuring the PCI Express settings and apply the changes. Regenerate the core to implement the changes in the project design files. There is also an option to delete the soft IP core if required.

Q: Why do I see IPexpress instead of Clarity Designer?

A: IPexpress currently supports LatticeECP3 devices. If your are targeting LatticeECP3 devices, then please refer to the IPexpress Flow for LatticeECP3 Devices section of the PCI Express x1/x2/x4 Endpoint IP Core User Guide. If you are targeting ECP5/ECP5-5G devices, go to **Project > Device** and select **ECP5/ECP5-5G**.

Q: Workaround for Lattice ECP5 (LFE5UM) Known Issue with SerDes Interface Connections Due to Unstable Reset Soft Logic.

A: The ECP5 SerDes reset logic may encounter instability issue during application. Lattice provides a workaround to resolve this issue, which can be found in the Workaround for Lattice ECP5 (LFE5UM) Known Issue with SerDes Interface Connections Due to Unstable Reset Soft Logic (FPGA-PB-02001) document.

6. Timing Closure

Timing closures can be frustrating and hard to close. Below is a list of preferences specific to PCI Express that may help you achieve timing closure. The three different constraints used are UGROUP, BLOCK and MULTICYCLE.

6.1. Default Constraints

When generating a PCI Express Endpoint IP Core, there will be default constraints added to your Lattice Preference File (LPF). The BLOCK RESETPATHS and BLOCK ASYNCPATHS constraints are included by default when creating a new project. After generating the PCI Express Endpoint IP Core, other default constraints are added as shown in Figure 6.1. If the design changes the name of these signals or their hierarchy, change the signal names as well.

Start Page | Reports | pci_exp_top.lpf

```
# Frequency Declerations
FREQUENCY NET "pclk" 250.000000 MHz PAR_ADJ 30.000000 ;
FREQUENCY NET "sys_clk_125_c" 125.000000 MHz PAR_ADJ 30.000000 ;
FREQUENCY NET "u1_pcs_pipe/ff_rx_fclk_0" 250.000000 MHz PAR_ADJ 30.000000 ;
FREQUENCY NET "rxrefclk_c" 200.000000 MHz ;
FREQUENCY NET "pll_refclki_c" 200.000000 MHz ;
USE PRIMARY NET "pclk" ;
USE PRIMARY NET "sys_clk_125_c" ;
USE PRIMARY NET "u1_pcs_pipe/ff_rx_fclk_0" ;
# Block Paths
BLOCK RESETPATHS ;
BLOCK ASYNCPATHS ;
BLOCK NET "rst_n_c*" ;
BLOCK NET "*RESET_n*" ;
# Assign input setup constraints
# INPUT_SETUP ALLPORTS 7.5 ns HOLD 1ns CLKNET "sys_clk_125_c" ;
# Assign clock to out constraints
# CLOCK_TO_OUT ALLPORTS 7.5 ns CLKNET "sys_clk_125_c";
BLOCK PATH FROM CLKNET "pclk" TO CLKNET "pll_refclki_c" ;
BLOCK PATH FROM CLKNET "pll_refclki_c" TO CLKNET "pclk" ;
BLOCK PATH FROM CLKNET "pll_refclki_c" TO CLKNET "sys_clk_125_c" ;
BLOCK PATH FROM CLKNET "sys_clk_125_c" TO CLKNET "pll_refclki_c" ;
```

Figure 6.1. Default Constraint File

6.2. UGROUP

For designs with identified critical paths, the UGROUP constraint can group components closer to the critical paths in order to shorten routing distances along the path. The following is a constraint option that was found useful for PCIe timing closure.

```
GROUP "PCS_PIPE" BBOX 10 20
BLKNAME pcie/pcie2_x1_inst/u1_dut/u1_dut/u1_dut/u1_pipe;
```

6.3. BLOCK

Many designs include paths that are asynchronous relative to the clocks of the design but never propagate a signal state because of logic encoding. If these parts of the design are slow, the user can use the BLOCK constraint to prevent the engine and trace from analyzing that part of the design. This releases the engine from spending time on irrelevant paths to meet timing requirements. BLOCK constraints specifically for the PCI Express Endpoint IP Core are listed below. These are helpful in meeting timing closure.

```
BLOCK NET "*ffs_pcie_con*" ;
BLOCK NET "*chx_RESET_n_i*" ;
BLOCK PATH FROM PORT "rstn";
BLOCK PATH FROM CELL "*ctc_reset_chx*" ;
BLOCK PATH FROM CLKNET "pcie/pcie2_x1_inst/pclk" TO CLKNET "refclk" ;
BLOCK PATH FROM CLKNET "refclk" TO CLKNET "pcie/pcie2_x1_inst/pclk" ;
```

```
BLOCK PATH FROM CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/tx_pclk" TO CLKNET "refclk" ;
BLOCK PATH FROM CLKNET "refclk" TO CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/tx_pclk" ;
BLOCK PATH FROM CLKNET "clk_125" TO CLKNET "refclk" ;
BLOCK PATH FROM CLKNET "refclk" TO CLKNET "clk_125" ;
BLOCK PATH FROM CLKNET "clk_125" TO CLKNET "pcie/pcie2_x1_inst/pclk" ;
BLOCK PATH FROM CLKNET "pcie/pcie2_x1_inst/pclk" TO CLKNET "clk_125" ;
BLOCK PATH FROM CLKNET "clk_125" TO CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/tx_pclk" ;
BLOCK PATH FROM CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/tx_pclk" TO CLKNET "clk_125" ;
BLOCK PATH FROM CLKNET "clk_125" TO CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/ff_rx_fclk_0" ;
BLOCK PATH FROM CLKNET "pcie/pcie2_x1_inst/u1_pcs_pipe/ff_rx_fclk_0" TO CLKNET "clk_125" ;
```

6.4. MULTICYCLE

A MULTICYCLE constraint allows you to specify a timing requirement that is different than the default case. This is usually used to relax clock period analysis. These three MULTICYCLE constraints also help with timing closure.

```
MULTICYCLE FROM CELL "*nfts_rx_skp_cnt*" TO CELL "*cnt_done_nfts_rx*" 2.000000 X ;
MULTICYCLE FROM CELL "*nfts_rx_skp_cnt*" TO CELL "*ltssm_nfts_rx_skp*" 2.000000 X ;
MULTICYCLE FROM CELL "*power_down*" TO CELL "*sll_inst*" 2.000000 X ;
```

References

For more information, refer to the following documents:

- PCI Express x1/x2/x4 Endpoint IP Core User Guide (FPGA-IPUG-02009) – An in depth user guide of the PCI Express Endpoint IP Core. This document provides details about its functionalities, parameter settings, IP core generation and evaluation, how to use the IP Core, and core verification.
- PCI Express Demos for the ECP5 and ECP5-5G Versa Development Board User Guide (FPGA-UG-02006) – A user guide on how to use the PCI Express soft IP Core with the ECP5/ECP5-5G Versa Board. Three different demos have been provided to help the user become familiar with Lattice PCI Express solutions and software development tools.
- Workaround for Lattice ECP5 (LFE5UM) Known Issue with SerDes Interface Connections Due to Unstable Reset Soft Logic (FPGA-PB-02001)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.2, January 2022

Section	Change Summary
All	Minor adjustments in formatting across the document.
Frequently Asked Questions	Updated content to add FAQ for ECP5 SerDes workaround.
References	Added Workaround for Lattice ECP5 (LFE5UM) Known Issue with SerDes Interface Connections Due to Unstable Reset Soft Logic document reference.

Revision 1.1, November 2019

Section	Change Summary
Disclaimers	Added this section.
Frequently Asked Questions	Indicated that PCI Express Endpoint IP Core is only compatible with ECP3/ECP5UM/ECP5UM5G devices.
References	Updated document number of PCI Express Demos for the ECP5 and ECP5-5G Versa Development Board User Guide to FPGA-UG-02006.
All	Minor formatting and style changes.

Revision 1.0, December 2016

Section	Change Summary
All	Initial release.

www.latticesemi.com