



PCIe Basic Demo for Lattice Nexus-based FPGAs

User Guide

FPGA-UG-02145-1.2

August 2025

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

Contents	3
Acronyms in This Document	6
1. Introduction	7
1.1. Learning Objectives	7
2. Hardware and Software Requirements	8
2.1. Hardware Requirements	8
2.2. Software Requirements	8
3. Demo Design Overview	9
3.1. Theory of Operation	9
3.2. FPGA Design	10
3.2.1. PCIe Hard IP	11
3.2.2. BAR RAM	11
3.2.3. Display Controller	11
4. Importing and Building the FPGA Demonstration	12
4.1. Demo Package Directory Structure	12
4.1.1. Demonstration Directory Structure	12
4.1.2. Hardware Directory Structure	12
4.2. Building Lattice Radiant Project	12
5. Setting Up the Demo	13
5.1. Hardware Setup	13
5.1.1. Jumper Configuration	13
5.1.2. Programming the FPGA	17
5.1.3. Status LED	20
5.2. Software Setup	21
5.2.1. Software Setup and Installation for Windows	21
5.2.2. Software Setup for Linux	36
6. Application Overview	39
6.1. Running the PCI Express Test Application	39
6.2. Using the PCI Express Test Application User Interface	40
6.3. Running PCIe Basic Demo	41
6.4. Running PCIe Memory Access Demo	43
7. Customizing the Design for CertusPro-NX	46
7.1. Changing the PCIe Lane Width	46
7.2. Changing the PCIe Link Speed	50
8. Troubleshooting	52
8.1. SPI Flash Update	52
8.2. Installing the Driver and User Interface Launch for Windows	53
8.2.1. Problem with Installing the Driver	53
8.2.2. Problem with Launching User Interface	53
8.3. Installing the Driver User Interface Launch for Linux	54
8.3.1. Problem with Installing the Driver	54
8.3.2. Problem in Loading the Driver	54
8.3.3. Problem with User Interface Launching	56
References	57
Technical Support Assistance	58
Revision History	59

Figures

Figure 3.1. Relationship of the Hardware and Software Components	9
Figure 3.2. PCIe Basic Demo SW Design	10
Figure 3.3. PCIe Basic Demo FPGA Design	10
Figure 5.1. CrossLink-NX PCIe Bridge Board Jumper Location	13
Figure 5.2. Certus-NX Versa Evaluation Board Jumper Location	14
Figure 5.3. CertusPro-NX Versa Evaluation Board Jumper Location	15
Figure 5.4. CrossLink-NX PCIe Bridge Board Connection	15
Figure 5.5. Certus-NX Versa Evaluation Board Connection	16
Figure 5.6. CertusPro-NX Versa Evaluation Board Connection	16
Figure 5.7. Creating a New Project from a Scan	17
Figure 5.8. Radiant Programmer Window	17
Figure 5.9. CrossLink-NX FPGA Device Settings	18
Figure 5.10. Certus-NX FPGA Device Settings	18
Figure 5.11. CertusPro-NX FPGA Device Settings	18
Figure 5.12. Device Properties Window for CrossLink-NX SPI Flash Programming	18
Figure 5.13. Device Properties Window for Certus-NX SPI Flash Programming	19
Figure 5.14. Device Properties Window for CertusPro-NX SPI Flash Programming	19
Figure 5.15. Programmer Menu Bar	20
Figure 5.16. Programmer Output Window	20
Figure 5.17. CrossLink-NX Status LED	20
Figure 5.18. Certus-NX Programming Done LED	21
Figure 5.19. CertusPro-NX Programming Done LED	21
Figure 5.20. Running Disable Integrity Checks Command	22
Figure 5.21. Running Test Sign On Command	22
Figure 5.22. Troubleshoot Option	23
Figure 5.23. Advanced Options	23
Figure 5.24. Select Startup Settings	23
Figure 5.25. Restarting Windows	24
Figure 5.26. Windows Installer: Preparing to Install Page	24
Figure 5.27. Windows Installer: Welcome Page	25
Figure 5.28. Windows Installer: Destination Folder Page	25
Figure 5.29. Windows Installer: Summary Page	26
Figure 5.30. Windows Installer: Application Installed	26
Figure 5.31. Device Configuration Prompt	27
Figure 5.32. Device Driver Installation Wizard	27
Figure 5.33. Windows Security in Driver Installation	27
Figure 5.34. Device Driver Installation Completed	28
Figure 5.35. Device Manager	28
Figure 5.36. Showing Device Properties	29
Figure 5.37. Hardware IDs of Basic Demo Device for CrossLink-NX	29
Figure 5.38. Hardware IDs of Basic Demo Device for Certus-NX	30
Figure 5.39. Hardware IDs of Basic Demo Device for CertusPro-NX	30
Figure 5.40. Update Driver Menu in Device Manager	31
Figure 5.41. Update Driver Options	31
Figure 5.42. Browse the Driver for Device	32
Figure 5.43. Windows Security in Device Manager	32
Figure 5.44. Driver Installation Status Message for CrossLink-NX	33
Figure 5.45. Driver Installation Status Message for Certus-NX	33
Figure 5.46. Driver Installation Status Message for CertusPro-NX	34
Figure 5.47. Basic Demo Device Name Displayed in Device Manager for CrossLink-NX	34
Figure 5.48. Basic Demo Device Name Displayed in Device Manager for Certus-NX	34
Figure 5.49. Basic Demo Device Name Displayed in Device Manager for CertusPro-NX	35

Figure 5.50. Uninstall the Device in Device Manager	35
Figure 5.51. Uninstall Device Dialog Box	36
Figure 5.52. Display Make Utility in Linux Terminal	36
Figure 5.53. Display Compiler Version in Linux Terminal	37
Figure 5.54. Update Compiler Version in Linux Terminal	37
Figure 6.1. PCIe Test Application Device Info Tab for CrossLink-NX.....	39
Figure 6.2. PCIe Test Application Config Space Tab.....	40
Figure 6.3. Three 7-Segment Displays on Board for CrossLink-NX and CertusPro-NX.....	41
Figure 6.4. Single 7-Segment Display on Board for Certus-NX	41
Figure 6.5. Interactive 7 Seg Display Tab for CrossLink-NX and CertusPro-NX.....	42
Figure 6.6. Interactive 7 Seg Display Tab for Certus-NX	42
Figure 6.7. PCIe Test Application Memory Tab.....	44
Figure 7.1. Save New Project	46
Figure 7.2. Open PCIe4 IP.....	46
Figure 7.3. PCIe IP Configuration to Select Lane Width.....	47
Figure 7.4. IP Generation	47
Figure 7.5. Selecting Input and Output Ports	48
Figure 7.6. Update PCIe Instance Module	48
Figure 7.7. Data Width Selection	49
Figure 7.8. Example of PDC File	49
Figure 7.9. Save and Recompilation of the Design	49
Figure 7.10. PCIe IP Configuration to Link Speed	50
Figure 7.11. Open PLL IP	50
Figure 7.12. PLL IP Configuration.....	51
Figure 8.1. TCK Frequency Setting	52
Figure 8.2. Port Selection.....	52
Figure 8.3. User Interface with No Device Driver	53
Figure 8.4. Error Message.....	54
Figure 8.5. lspci -vnm Output Image for CrossLink-NX	54
Figure 8.6. lspci -vnm Output Image for Certus-NX.....	55
Figure 8.7. lspci -vnm Output Image for CertusPro-NX	55
Figure 8.8. Content List of Demonstration/Linux Directory.	56
Figure 8.9. Content List of Software/Linux Directory	56

Tables

Table 1.1. Target Boards and Devices.....	7
Table 5.1. Jumper Configuration for the CrossLink-NX PCIe Bridge Board.....	13
Table 5.2. Jumper Configuration for the Certus-NX Versa Evaluation Board	14
Table 5.3. Jumper Configuration for the CertusPro-NX Versa Evaluation Board.....	14
Table 5.4. CrossLink-NX Status LED Description	20
Table 6.1. Device Info Tab	40
Table 6.2. Configuration Space Tab	40
Table 6.3. Memory Data and Corresponding 7-Segment Display Values	43
Table 6.4. Memory Tab Description	44

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
API	Application Programming Interface
BAR	Base Address Register
FDSOI	Fully Depleted Silicon on Insulator
FPGA	Field-Programmable Gate Array
LED	Light-emitting diode
MIPI	Mobile Industry Processor Interface
PCIe	PCI Express
PHY	Physical Layer
SPI	Serial Peripheral Interface
USB	Universal Serial Bus

1. Introduction

This user guide describes how to setup and run the PCIe Basic Demo using devices built on the Lattice Nexus™ platform, specifically CrossLink™-NX, Certus™-NX, and CertusPro™-NX. The Lattice Nexus platform features low power 28 nm FDSOI technology.

Table 1.1. Target Boards and Devices

Device Family	Board	Device
CrossLink-NX	CrossLink-NX PCIe Bridge Board	400-ball caBGA package (LIFCL-40-8BG400C)
Certus-NX	Certus-NX Versa Evaluation Board	256-ball caBGA package (LFD2NX-40-8BG256C)
CertusPro-NX	CertusPro-NX Versa Board	672-ball LFG package (LFCPNX-100-9LFG672)

This user guide familiarizes you with the process of setting up the PCI Express development environment. It is assumed that no associated tools are installed on the system.

The demo discussed in this document includes the PCI Express Basic Demo.

1.1. Learning Objectives

After completing the steps in this guide, you can perform the following:

- Set up and install all applicable development tools and PCI Express demos;
- Establish communication between the FPGA and the system through the PCI Express link;
- Run the PCI Express Basic Demo that allows you to control 7-segment LEDs on the board, as well as manipulate the onboard memory of the FPGA through the PCIe interface.
- Modify and rebuild the PCI Express Basic Demo.
- Become familiar with the software development tools and major design flow steps employed in this kit.
- Use other existing documentation in conjunction with this guide.

This document assumes that the Lattice Radiant™ design software is already installed. This document covers some of the basic of function of Lattice Radiant software. To learn more about Lattice Radiant software, refer to the Lattice Radiant software Help system.

2. Hardware and Software Requirements

2.1. Hardware Requirements

To install the kit design and run the demo software, a computer with a PCI Express ×16, ×8, ×4, or ×1 slot is required. The computer must also have a USB port and be able to run the Lattice Radiant Software. All other hardware and drivers are included in the kit.

- Mini-USB to USB-A cable for programming the bitstream
- 12 V Power Adapter
- Evaluation Board
- CrossLink-NX PCIe Bridge Board for CrossLink-NX
- Certus-NX Versa Board for Certus-NX
- CertusPro-NX Versa Board for CertusPro-NX

2.2. Software Requirements

The following software is required to obtain the expected results for the procedures described in this guide:

- Lattice Radiant Software 2.2 or later (available at the Lattice website [Design Software and IP](#) page).
- PCIe Basic Demo for Windows Operating System or Linux Operating System
- Windows or Ubuntu 24.04.2
- Bit file for the SPI Flash (included in the demo package)
- LIFCL-40_PCl_e_Basic_Demo.bit for CrossLink-NX
- LFD2NX40_PCl_e_Basic_Demo_impl_1.bit for Certus-NX
- BasicDemo_implementation.bit for CertusPro-NX

3. Demo Design Overview

3.1. Theory of Operation

The demo runs on a standard x64 PC and accesses the FPGA Board installed in a PCIe slot. [Figure 3.1](#) shows the relationship of the hardware and software components of the demo. The PCIe IP present in the Lattice FPGA acts as a PCIe endpoint occupying certain ranges of PCI memory space. When the PC boots, the BIOS and OS probe the PCI Express and PCI buses and detect the devices present on the buses and assign them ranges in the PCI memory space. The PCI memory space is mapped into the PC's memory space by the BIOS. Once the device driver is installed, the application software can read/write from/to the PCIe device memory (Embedded Block RAM - EBR). The application software can also read from the PCIe configuration space registers.

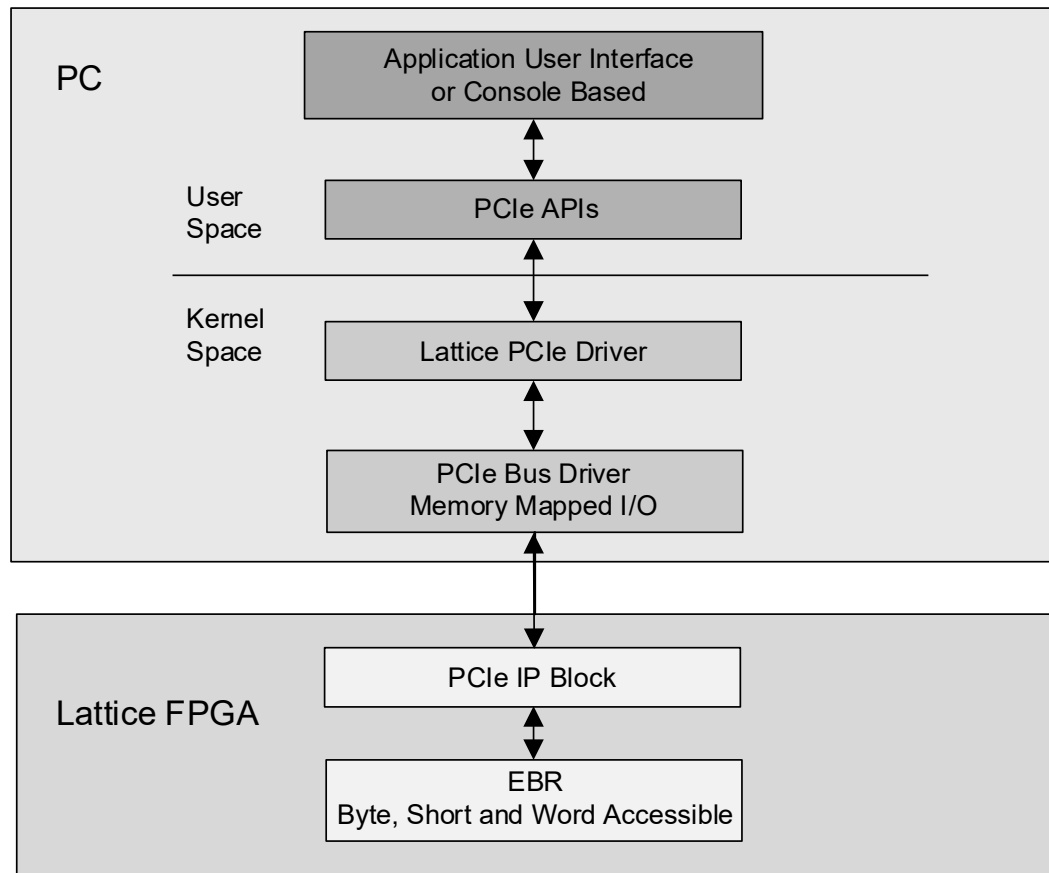


Figure 3.1. Relationship of the Hardware and Software Components

3.2. FPGA Design

A user interface application is provided for demonstrating the PCIe Basic demo. Application software is developed using a layered architecture consisting of the following layers:

- User interface application
- Protocol library
- Driver
 - PCIe Driver for BAR memory read write
 - PCIe Bus driver for Lattice Endpoint

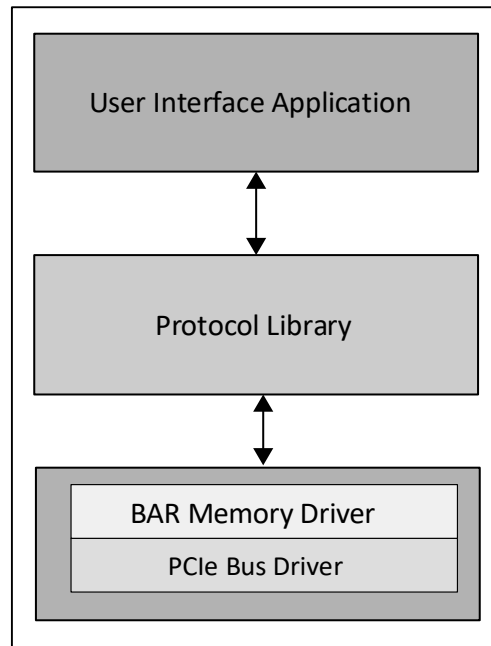


Figure 3.2. PCIe Basic Demo SW Design

PCIe hard IP is used on the FPGA side to implement the PCIe endpoint. At the transaction layer, logic implemented in the `pcie_tx_engine` and `pcie_rx_engine` modules are used to handle communication to/from the PCIe hard IP transaction layer, which consists primarily of encoding TLPs for transmission, and decoding received TLPs and storing this data in local memory. An application, 4 kB of Bar memory and 7-Segment displays are used with single PCIe Function. Apart from that, the LMMI interface is used for initial reconfiguration of the PCIe IP.

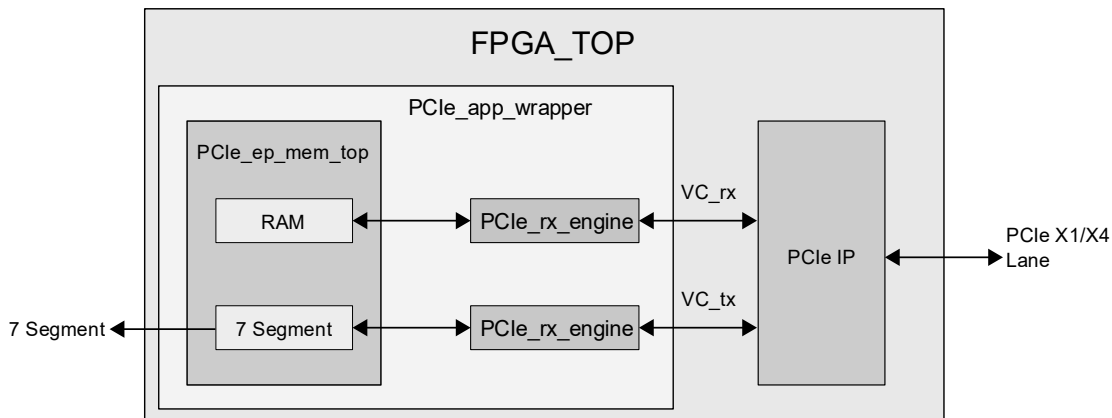


Figure 3.3. PCIe Basic Demo FPGA Design

3.2.1. PCIe Hard IP

The PCIe hard IP has three layers: physical layer, data link layer and transaction layer. On power up, the physical layer is responsible for doing link training. Once the training is completed, a `pl_link_up` signal is asserted, indicating that the physical layer is up. The Physical layer is also responsible for the rate negotiation, 8b/10b encoding/decoding (Gen1 and Gen2) and 128b/130b encoding (Gen3), byte striping, scrambling/descrambling, and packet serialization/deserialization.

The Data Link Layer is responsible for the creation and transmission of DLLPs (Data Link Layer Packet) and decoding these received packets. This layer is also responsible for error detection and retransmission (ACK/NAK protocol).

The Transaction Layer is responsible for transmitting and receiving TLPs (Transaction Layer Packet). These packets can be memory read/write, I/O read/write, messages and others

The PCIe hard IP is hardened into the FPGA. It does not consume fabric resources when used.

The PCIe hard IP has three signals `u_pl_link_up`, `u_dl_link_up`, `u_tl_link_up` that indicate the link-up status of the three previously mentioned layers.

3.2.2. BAR RAM

A 4 kB BAR memory is implemented on the application side to store the data. Through software, you can read or write data from/to this RAM.

3.2.3. Display Controller

The BAR offset address 0x0 is mapped to the 7-segment display(s) on the board. Refer to the [Running PCIe Basic Demo](#) section to learn about controlling the 7-segment display(s) through the application user interface.

4. Importing and Building the FPGA Demonstration

4.1. Demo Package Directory Structure

The basic demo package consists of *Demonstration* and *Hardware* folders.

4.1.1. Demonstration Directory Structure

The *Demonstration* folder contains the following directories:

- *Bitstream* – Contains the bitstream file of the project. You may program this bitstream file to the board directly.
- *Windows10/Linux* – Contains the device drivers, API library and user interface related files for the application.

4.1.2. Hardware Directory Structure

The *Hardware* folder inside the package contains the following directories.

- *CL_NX_BridgeBoard_PClEBasicDemo*, *CertusNXBoard_PClEBasicDemo*, or *CPNX_PClE_BasicDemo*
 - *IP* – Contains the pre-generated IPs used in the design. These IPs can be configured by clicking the .ipx file after opening the project in Radiant.
 - *Implementation* – Contains the Lattice Radiant project (.rdf) file, constraints file (.pdc), implemented design and bit files.
 - *Source* – Contains RTL files required for the design.

4.2. Building Lattice Radiant Project

Note: The bitstream file of the project is included in the *Demonstration/Bitstream* folder. This section may be skipped if there is no plan to modify or recompile the project.

To open, recompile the project, and regenerate the bitstream file:

1. Open the Lattice Radiant software.
2. Click **Open project** and browse to the .rdf file
 - For CrossLink-NX, the file is LIFCL-40_PClE_Basic_Demo.rdf, which is located in the *Hardware/CLNX_BasicDemo/Implementation/BasicDemo/LIFCL-40_PClE_Basic_Demo/* folder.
 - For Certus-NX, the file is LFD2NX40_PClE_Basic_Demo.rdf, which is located in the *Hardware/PClEBasicDemo/Implementation/BasicDemo/* folder.
 - For CertusPro-NX, the file is *BasicDemo.rdf*, which is located in the *Hardware/VersaBoard/BasicDemo/* folder.
3. Once the project loads, click **Task Detail View**. This shows a list of actions that Lattice Radiant performs to build the .bit file.
4. Select the files and reports to generate.

Note: The options needed to regenerate the .bit file are selected by default.

After selecting the preferred reports, click **Run All**. This creates a .bit file with the project's current name in the *imp1* folder.

5. Setting Up the Demo

5.1. Hardware Setup

This section covers the steps in programming the demo to the SPI memory of the Board.

5.1.1. Jumper Configuration

For CrossLink-NX, install the jumpers listed in [Table 5.1](#). If this is the first time the board is used, the jumpers must be in the indicated configuration.

Table 5.1. Jumper Configuration for the CrossLink-NX PCIe Bridge Board

Jumper Checklist	Configuration
J11, J18	Connect pins 1 and 2
J7, J6, J13, J14	Connect pins 1 and 2
J31, J32, J33	Connect pins 2 and 3
J8, J9, J10, J15, J16, J19, J20, J21, J22, J23, J25, J26, J29, J30, J34, J35, J36, J37, J38, J39, J42	<i>These are current measurement headers and must be kept Open.</i>
J51	Connect pins 2 and 4
J48	Connect pins 2 and 3
J27	Connect pins 1 and 2
J45, J46	Connect pins 1 and 2
J17	Open

[Figure 5.1](#) shows the location of the jumpers for the CrossLink-NX PCIe Bridge Board.

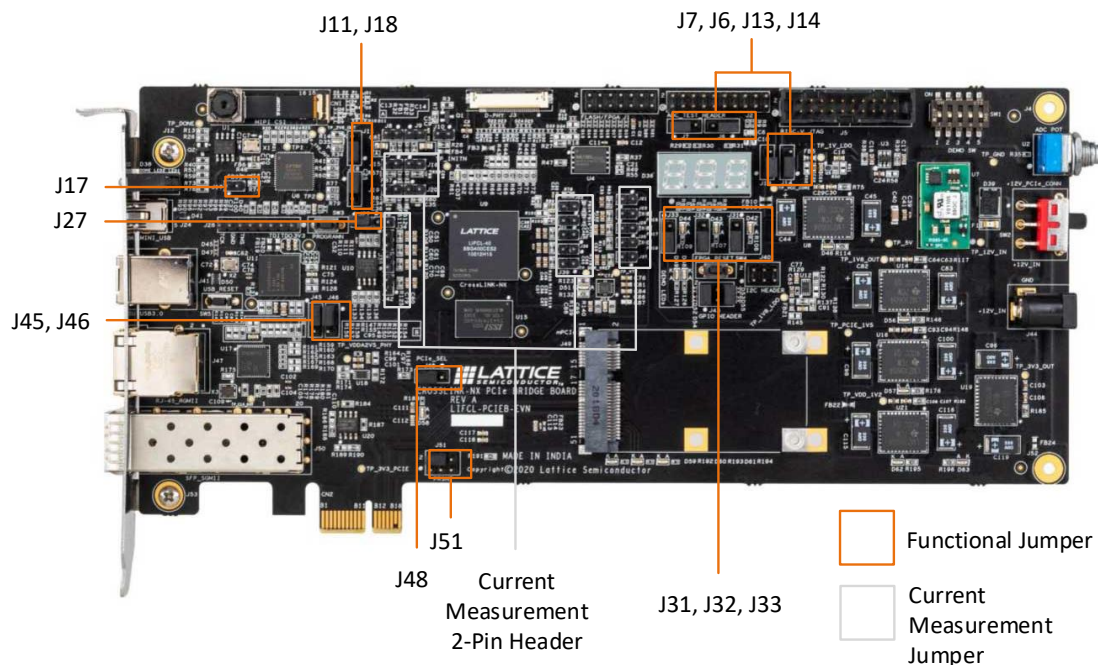


Figure 5.1. CrossLink-NX PCIe Bridge Board Jumper Location

For Certus-NX, install the jumpers listed in [Table 5.2](#). Note the difference between J and JP jumpers. If this is the first time utilizing the Certus-NX Versa Evaluation Board, the jumpers are already in this configuration. Refer to [Certus-NX Versa Evaluation Board User Guide \(FPGA-EB-02032\)](#) for a detailed list of jumpers on the board.

Table 5.2. Jumper Configuration for the Certus-NX Versa Evaluation Board

Jumper Checklist	Configuration (Note Pin 1 is denoted with a small triangle)
J16	Connect Pins 1 and 2
J47	Connect Pins 2 and 3
J48	Connect Pins 1 and 2
JP23	Short

[Figure 5.2](#) shows the location of the jumpers for the Certus-NX Versa Evaluation Board.

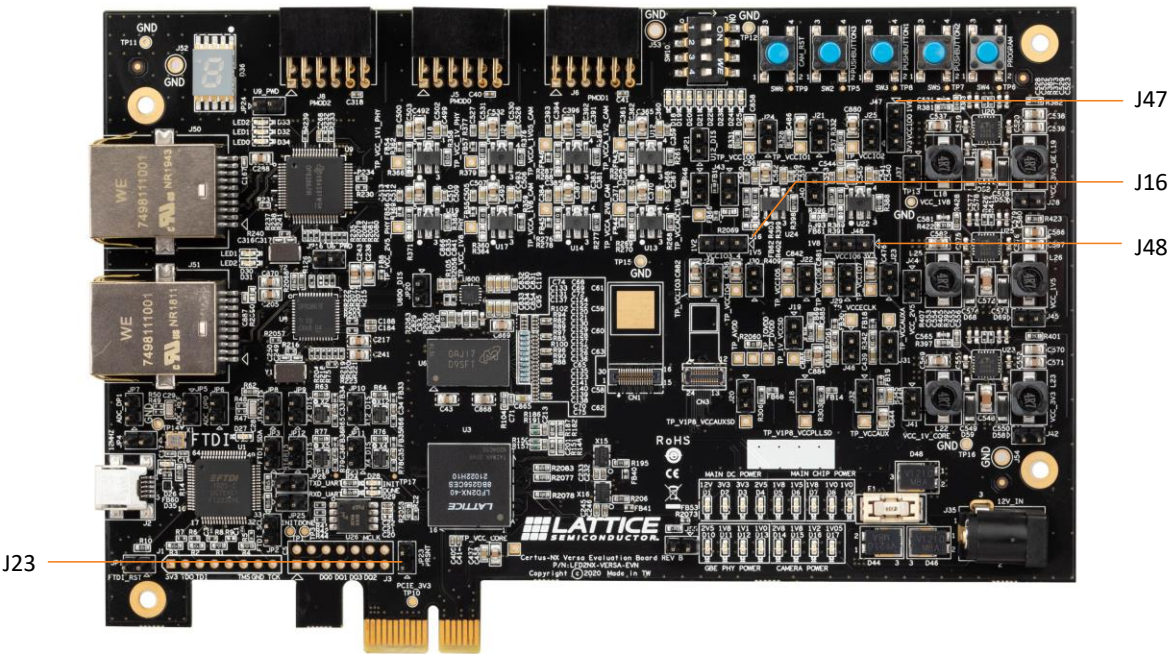


Figure 5.2. Certus-NX Versa Evaluation Board Jumper Location

For CertusPro-NX, take note on the PCIe related jumper in [Table 5.3](#). If this is the first time utilizing the board, the jumpers are already in this configuration. Refer to the [CertusPro-NX Versa Board User Guide \(FPGA-EB-02053\)](#) for a detailed list of jumpers on the board.

Table 5.3. Jumper Configuration for the CertusPro-NX Versa Evaluation Board

Jumper Checklist	Configuration
J1	Default Short 2-4 for PCIe X4

[Figure 5.3](#) shows the location of the jumpers for the CertusPro-NX Versa Evaluation Board.

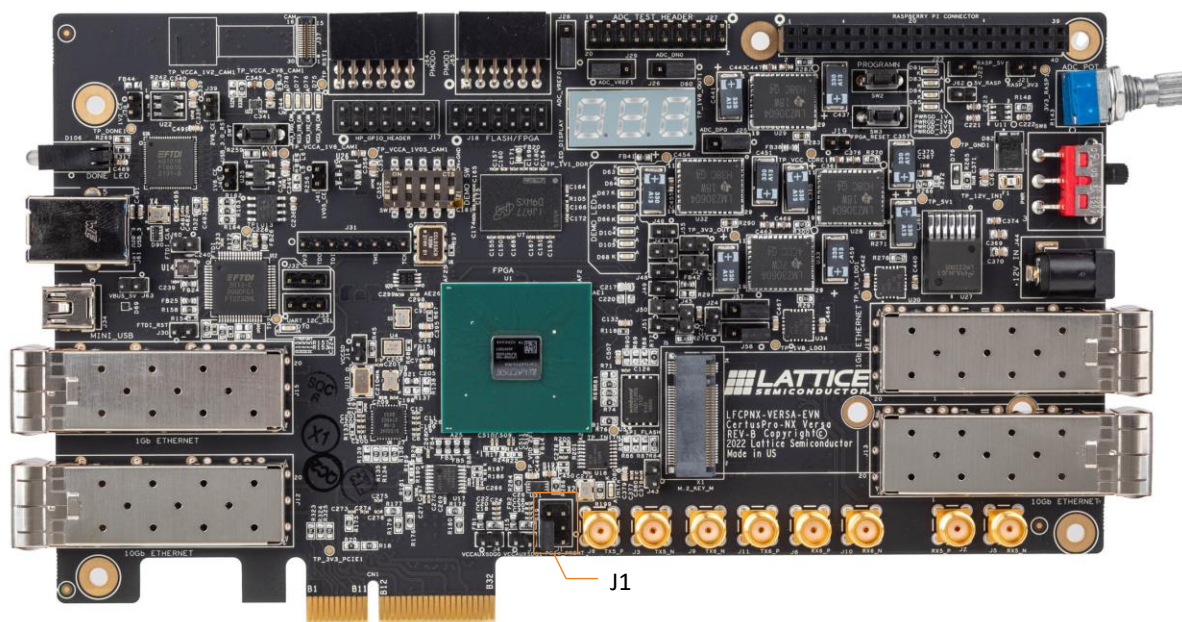


Figure 5.3. CertusPro-NX Versa Evaluation Board Jumper Location

After the board jumpers have been configured appropriately, make sure the board is plugged into a PCIe slot or connected to the 12 V input supply.

For CrossLink-NX, if the board is plugged into the PCIe slot, external power is provided by the system, and SW2 must be in the *up* position to receive power from the PCIe slot. If external 12 V power is provided, then SW2 must be in the *down* position to receive power from the external 12 V connection. Connect the board to the PC running the Lattice Radiant software with the Mini USB Type A Cable as shown in Figure 5.4.

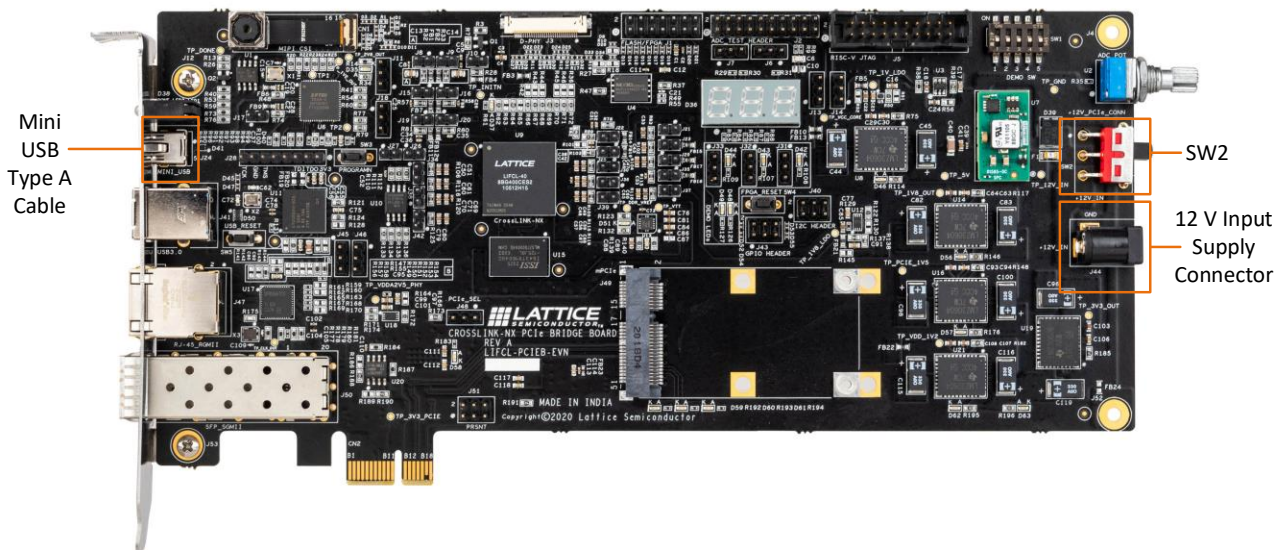


Figure 5.4. CrossLink-NX PCIe Bridge Board Connection

For Certus-NX, there is no toggle switch to select the power source, and the board can be powered through the PCIe slot or from an external 12 V input supply. If the board is powered from the PCIe slot, the external 12 V input supply connector does not need to be connected. Connect the board to the PC running the Lattice Radiant software with the Mini USB Type A Cable as shown in [Figure 5.5](#).

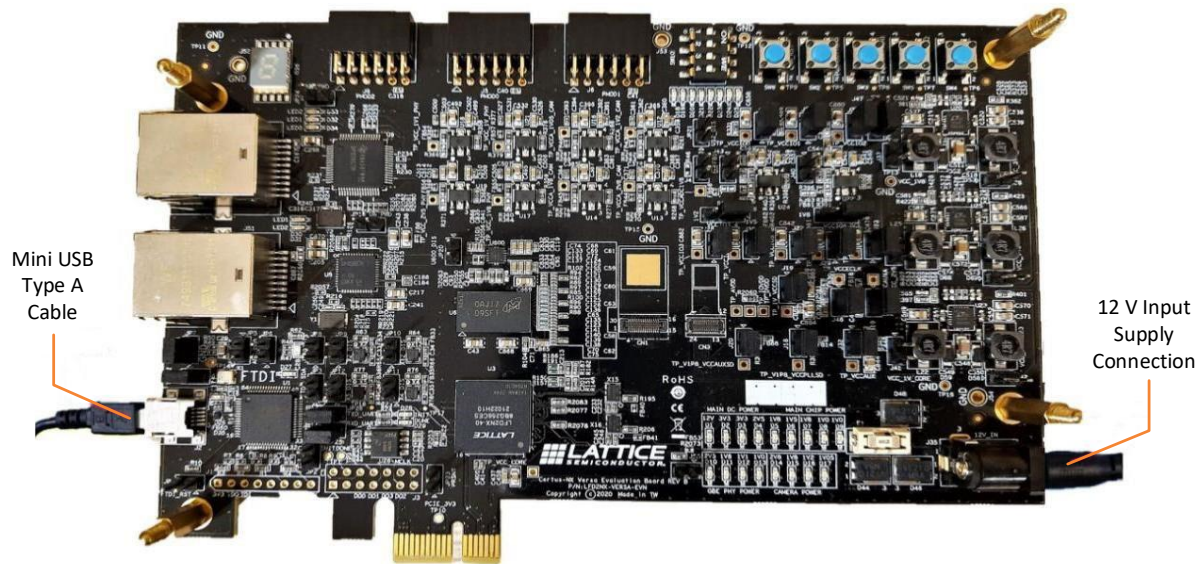


Figure 5.5. Certus-NX Versa Evaluation Board Connection

For CertusPro-NX, when the board is plugged into the PCIe slot, external power is provided by the system, and SW6 must be in the *up* position to receive power from the PCIe slot. Connect the board to the PC running the Lattice Radiant software with the Mini USB Type A Cable as shown in [Figure 5.6](#).

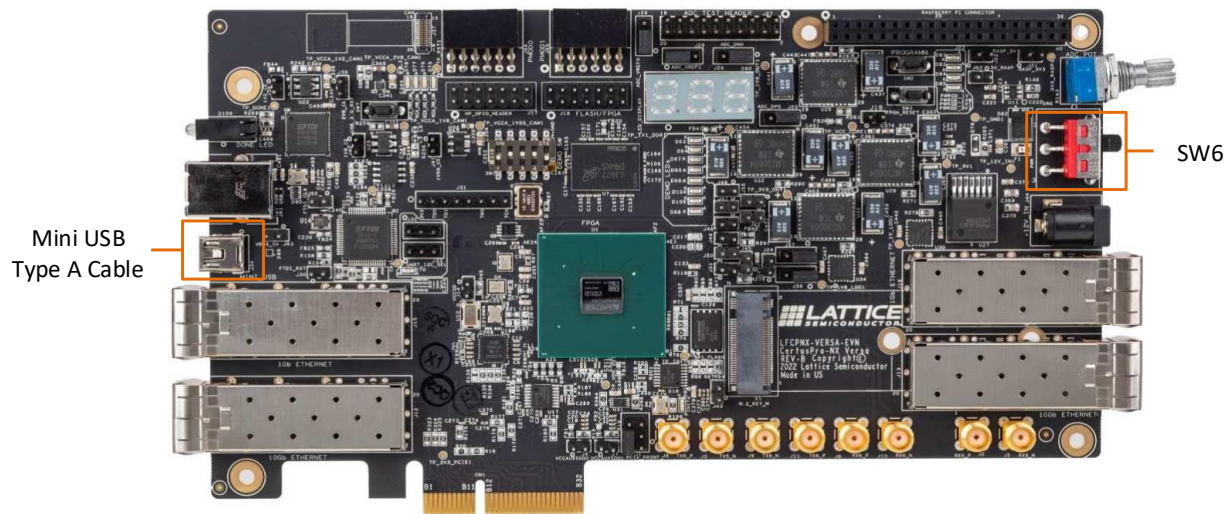


Figure 5.6. CertusPro-NX Versa Evaluation Board Connection

5.1.2. Programming the FPGA

To program the FPGA device with the bitstream file:

1. Create a new project using the Lattice Radiant Programmer software. In the **Getting Started** dialog box, indicate **Project Name** and **Location** as shown in [Figure 5.7](#).
2. Select Create a new project from a scan. Values are indicated in the Cable, Port, and TCK Divider Setting (0-30x) fields.
3. Click **OK**.

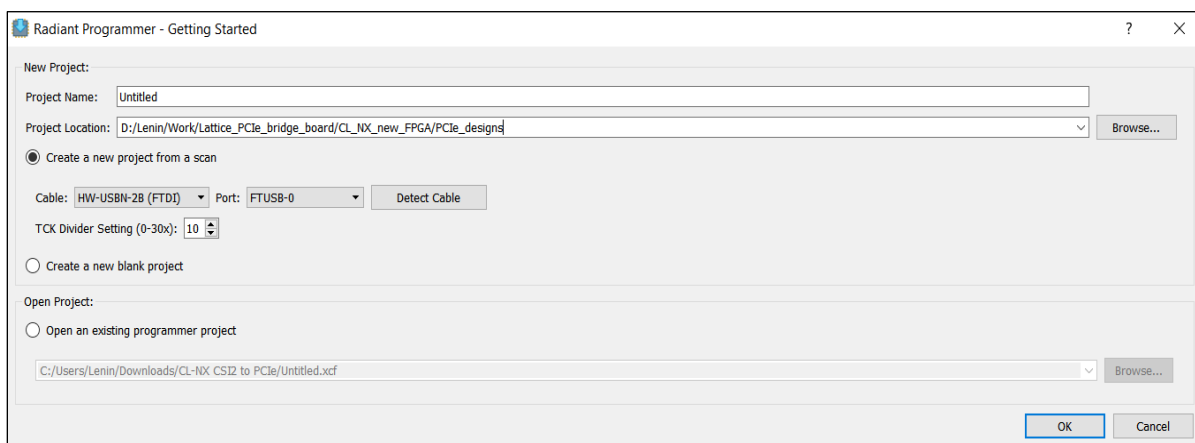


Figure 5.7. Creating a New Project from a Scan

4. The main interface opens as shown in [Figure 5.8](#).

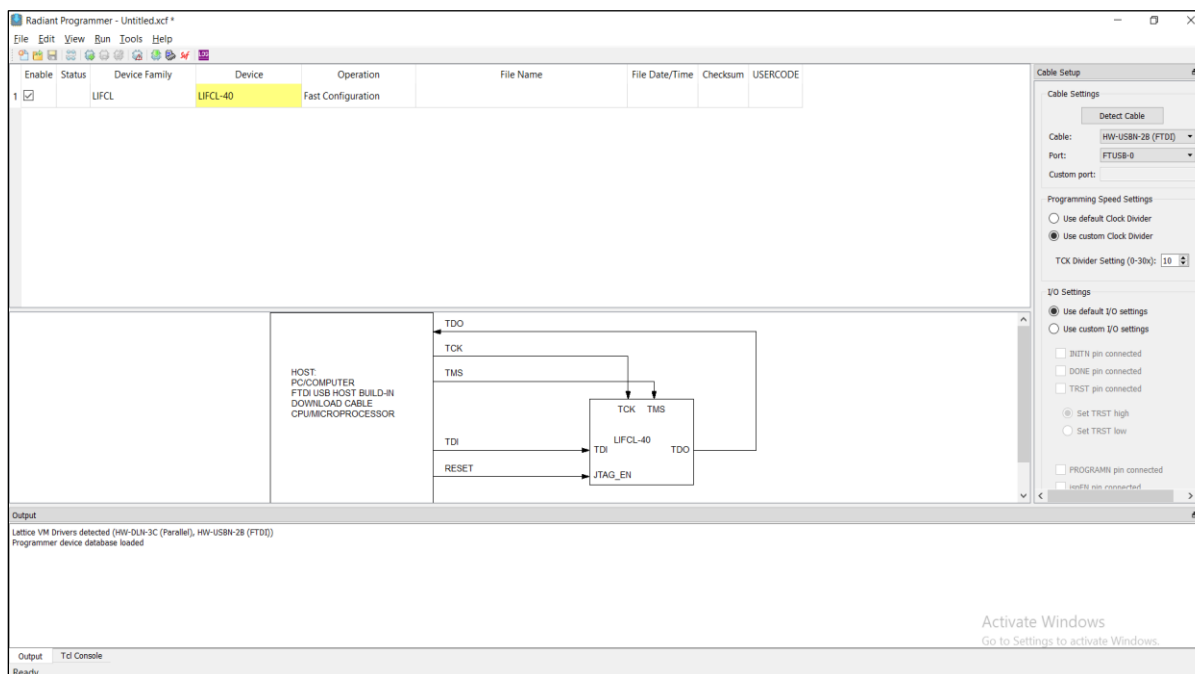


Figure 5.8. Radiant Programmer Window

- If the Programmer settings do not match the settings shown in [Figure 5.9](#) for CrossLink-NX, [Figure 5.10](#) for Certus-NX, or in [Figure 5.11](#) for CertusPro-NX select these settings manually from drop-down menu.

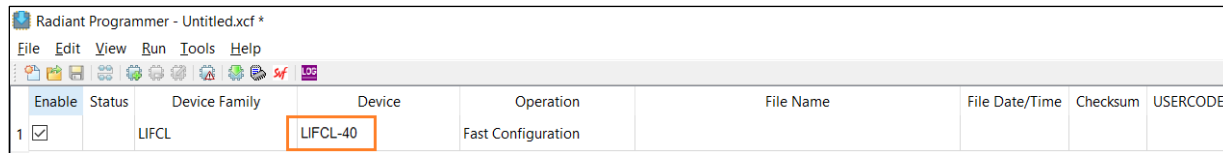


Figure 5.9. CrossLink-NX FPGA Device Settings

Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☒	LFD2NX	LFD2NX-40	Fast ...				

Figure 5.10. Certus-NX FPGA Device Settings

Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☒	LFCPNX	LFCPNX-100	Fast ...				

Figure 5.11. CertusPro-NX FPGA Device Settings

To select programming settings:

- Browse and select the Programming file *Demonstration/Bitstream*.
 - For CrossLink-NX, select *LIFCL-40_PCIe_Basic_Demo.bit* from *Demonstration/Bitstream*.
 - For Certus-NX, select *LFD2NX40_PCIe_Basic_Demo_impl_1.bit* from *Demonstration/Bitstream*.
 - For CertusPro-NX, select *BasicDemo_Implementation.bit* from *Demonstration/Bitstream*.
- Click **OK**.
- Double-click under **Operation** to open the **Device Properties** dialog box.
- For CrossLink-NX, select the settings as shown in [Figure 5.12](#).

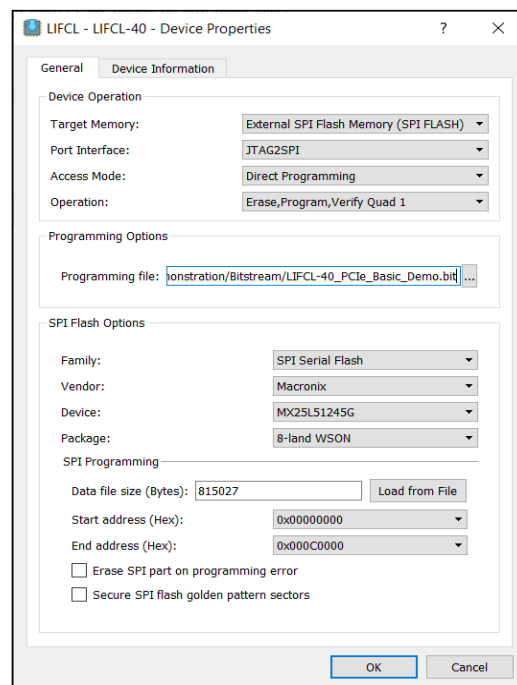


Figure 5.12. Device Properties Window for CrossLink-NX SPI Flash Programming

- For Certus-NX, select the settings as shown in [Figure 5.13](#).

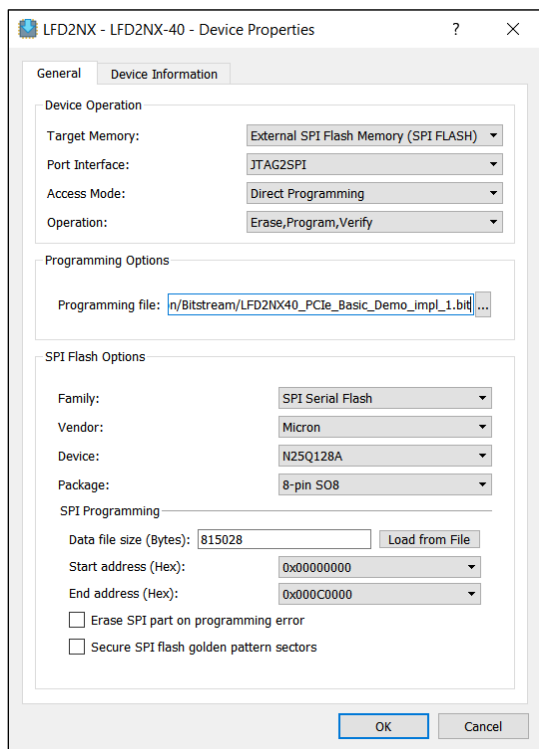


Figure 5.13. Device Properties Window for Certus-NX SPI Flash Programming

- For CertusPro-NX Versa Board, select the settings as shown in [Figure 5.14](#).

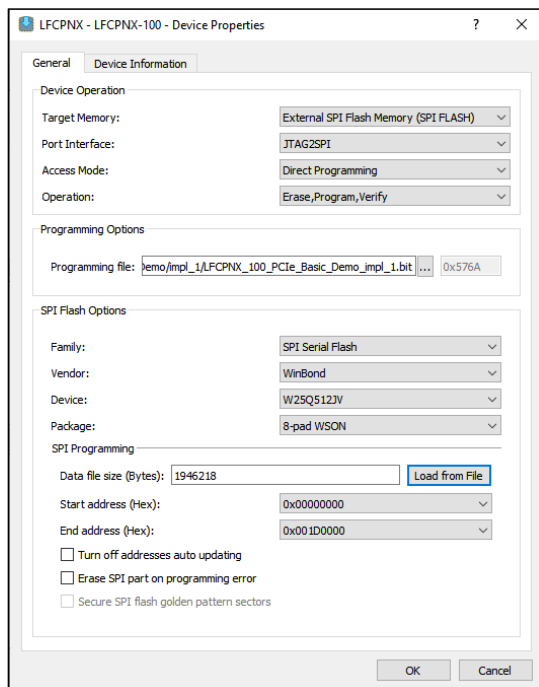


Figure 5.14. Device Properties Window for CertusPro-NX SPI Flash Programming

- Click the **Programming** button from the menu bar shown in [Figure 5.15](#) to start programming.



Figure 5.15. Programmer Menu Bar

When the FPGA programming is successful, the output console shows an Operation Successful message as shown in [Figure 5.16](#).

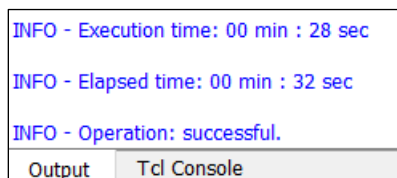


Figure 5.16. Programmer Output Window

If the programming operation is not successful, refer to the troubleshooting section of this document.

After programming, power cycle the board and check the status LEDs on the board. The LED status is discussed in the next section.

5.1.3. Status LED

For CrossLink-NX, the three status LEDs are shown in [Figure 5.17](#).

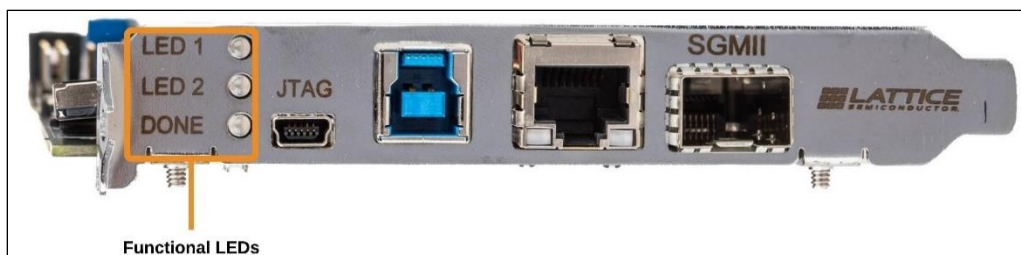


Figure 5.17. CrossLink-NX Status LED

The description of each status LED is provided in [Table 5.4](#).

Table 5.4. CrossLink-NX Status LED Description

Sl. No	Name	Description
1	125 MHz Clock present	Green – Blinking when clock present (125 MHz Clock)
2	PCIe Link Up	Green – PCIe link up is successful Red – PCIe link up is not successful
3	DONE	Green – Configuration is successful Red – Board is not programmed

For Certus-NX, the programming done LED lights up in green if configuration is successful. The LED is located at D29 on the board, as indicated in [Figure 5.18](#).

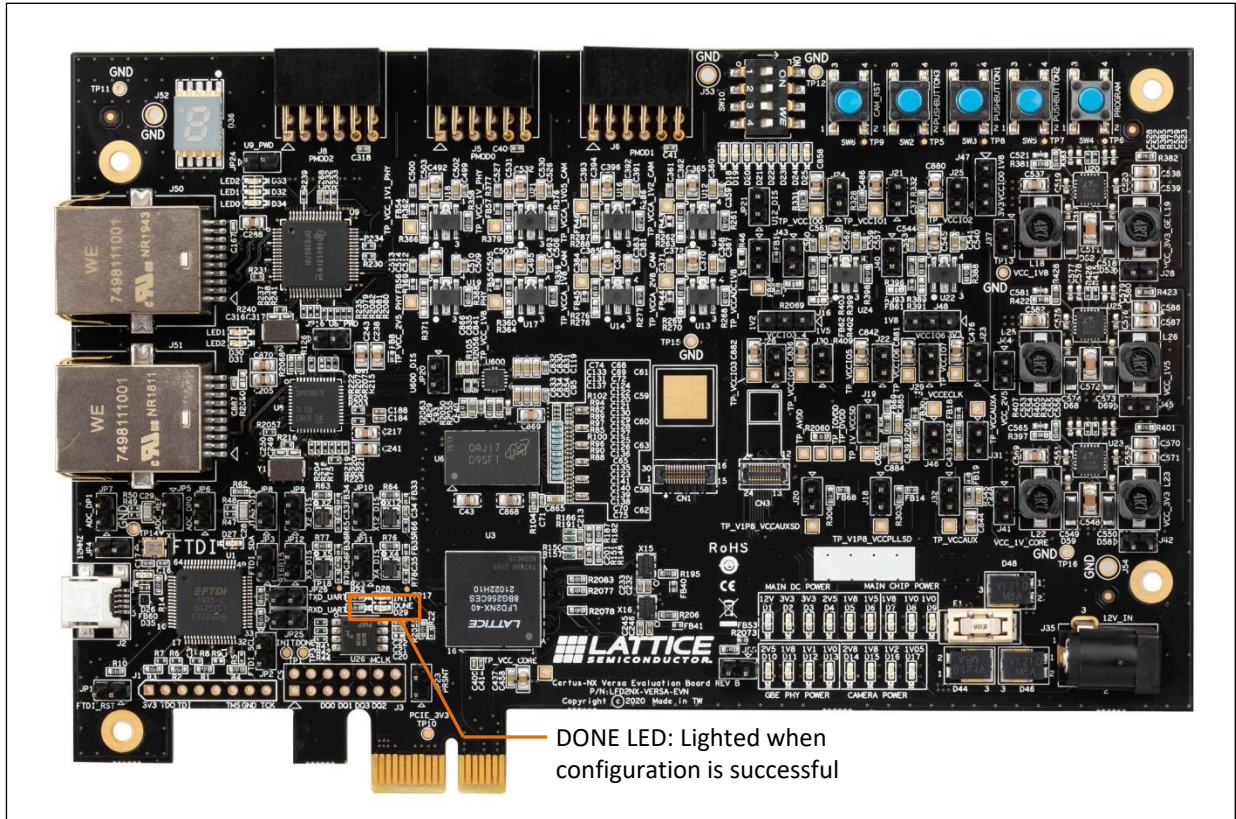


Figure 5.18. Certus-NX Programming Done LED

For CertusPro-NX, the programming done LED lights up in green if configuration is successful. The LED is located as shown in Figure 5.19.

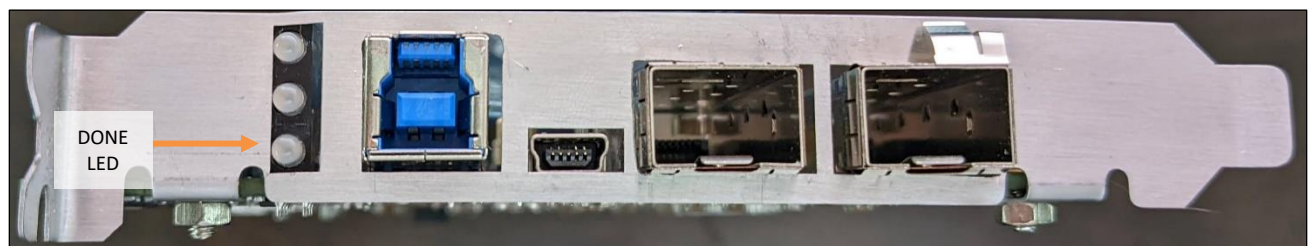


Figure 5.19. CertusPro-NX Programming Done LED

5.2. Software Setup

This section provides the procedure for installing software onto the host machine.

5.2.1. Software Setup and Installation for Windows

Before installing the driver, driver signature enforcement must be disabled.

5.2.1.1. Disabling Driver Signature Enforcement Permanently

To permanently disable driver signature enforcement:

1. Start the Command Prompt as administrator.
2. Enter the following lines and press Enter.

```
bcdedit.exe -set loadoptions DISABLE_INTEGRITY_CHECKS
```

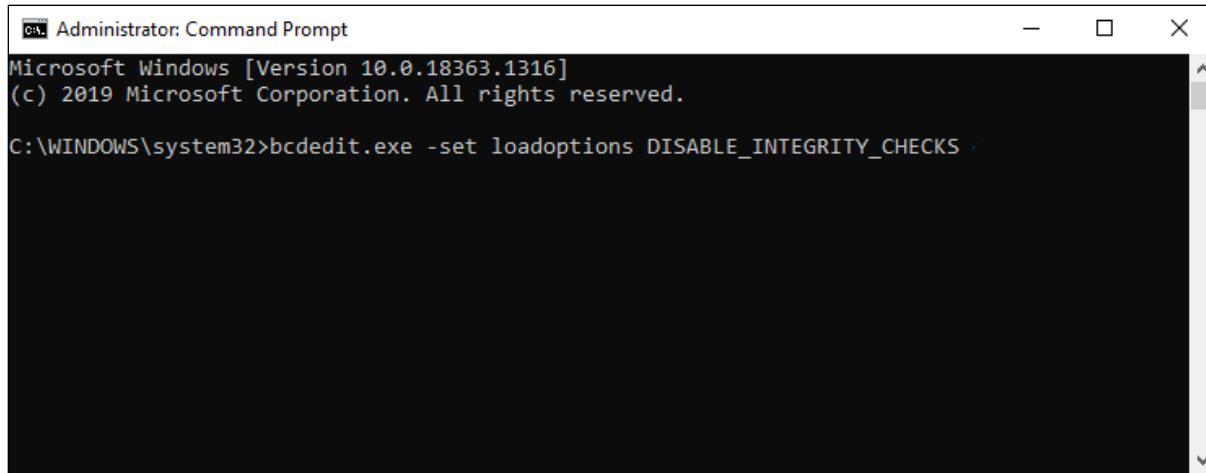


Figure 5.20. Running Disable Integrity Checks Command

```
bcdedit.exe -set TESTSIGNING ON
```

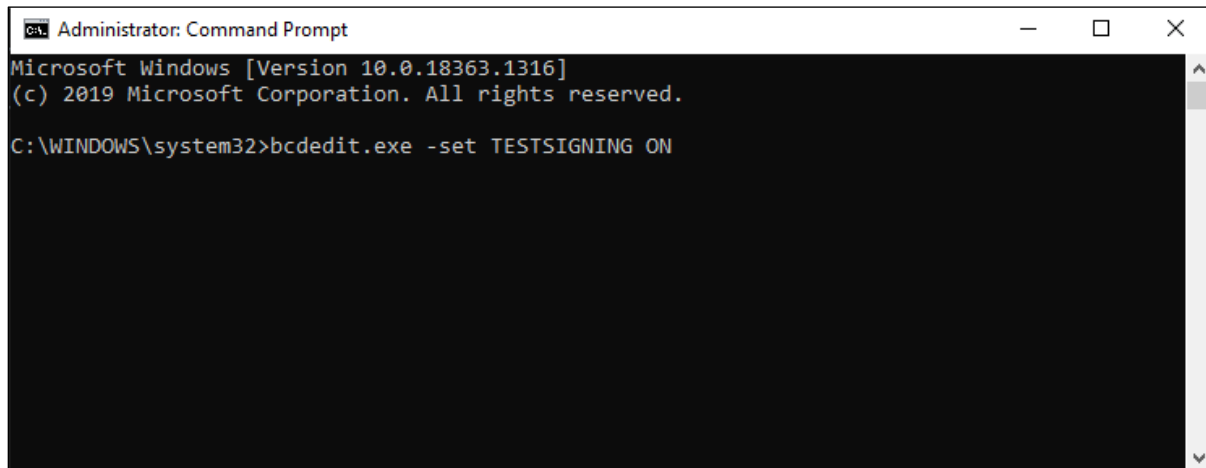


Figure 5.21. Running Test Sign On Command

3. Close the Command Prompt and restart the PC.

5.2.1.2. Disabling Driver Signature Enforcement Temporarily

Note: If Driver Signature Enforcement is already disabled, skip this section and proceed to the [Installing the Driver](#) section.

To disable the driver signature enforcement temporarily on Windows operating system:

1. Press the Windows key and click the power button.
2. Press and hold the Shift key and click **Restart**.
3. When the **Choose an option** screen appears as shown in [Figure 5.22](#), release the Shift key.
4. Click **Troubleshoot**.

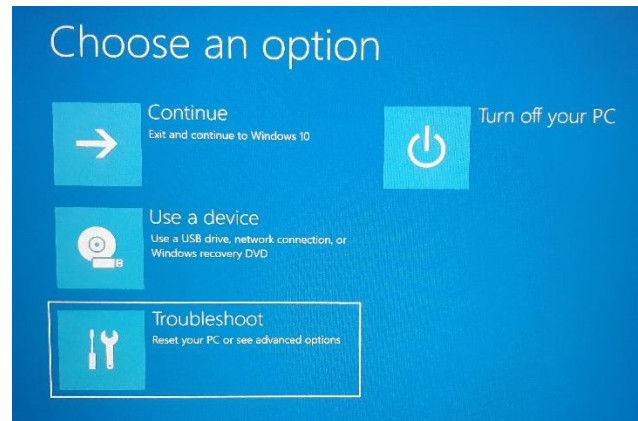


Figure 5.22. Troubleshoot Option

5. Select **Advanced options** and press Enter.



Figure 5.23. Advanced Options

6. Select **Startup Settings** and press Enter.



Figure 5.24. Select Startup Settings

7. Press Enter to restart.

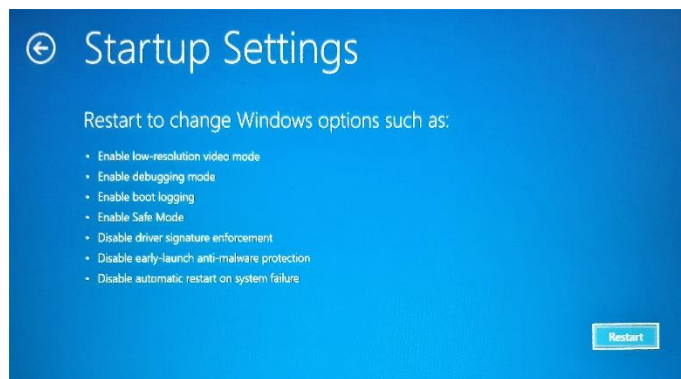


Figure 5.25. Restarting Windows

8. After restarting, select Option 7 to disable driver signature verification.

5.2.1.3. Installing the Driver

Choose either one of the two methods below to install the device driver:

- Install through a user interface installer (.exe) that is included in the demo package
- Install through Device Manager manually

Installing Basic Demo Device Driver through the User Interface Installer

The Basic Demo device driver is installed during the installation of the user interface as described in the following section.

The installer provides a standard packaging format for applications and a standard method for customizing the applications. The installer helps to install the PCIe Basic Demo application in the system.

To install the Basic Demo device driver through user interface:

1. In the Demonstration\Windows\Application folder, double click setup.exe.
2. Click **Next** when the software installation wizard opens.

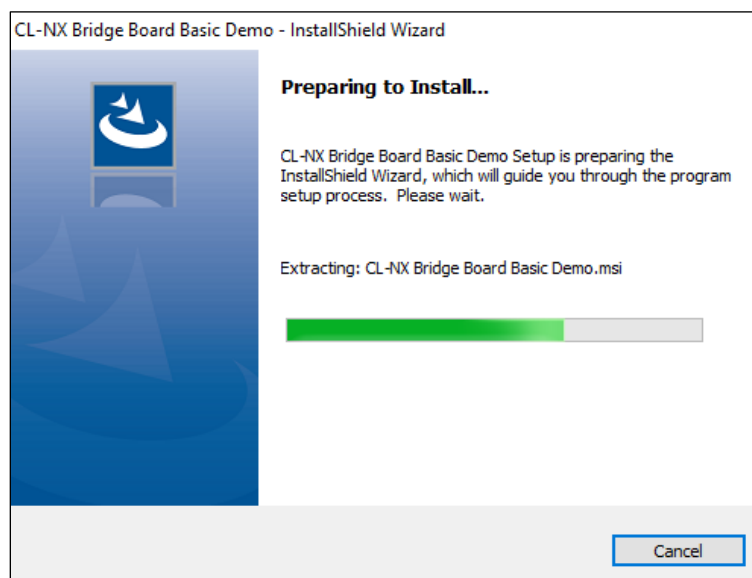


Figure 5.26. Windows Installer: Preparing to Install Page

3. The welcome page appears. Click **Next**.

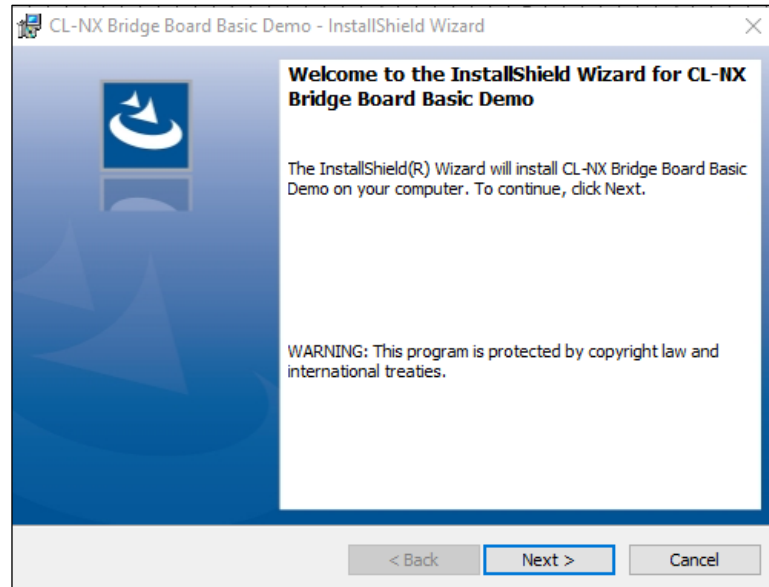


Figure 5.27. Windows Installer: Welcome Page

4. Provide the location where to install the application. Click **Next**.

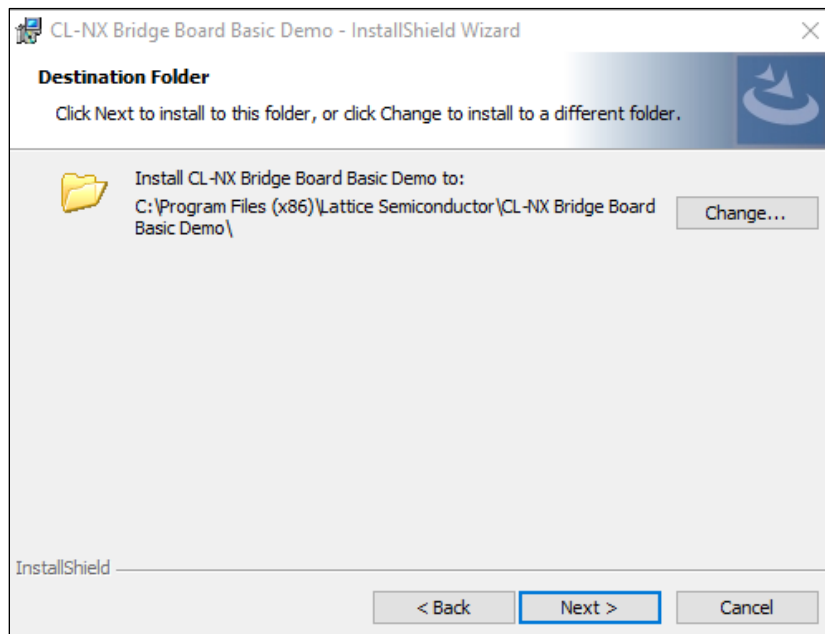


Figure 5.28. Windows Installer: Destination Folder Page

5. The installation summary page is shown. Click **Install**.

Note: Administrative access is required to run this command.

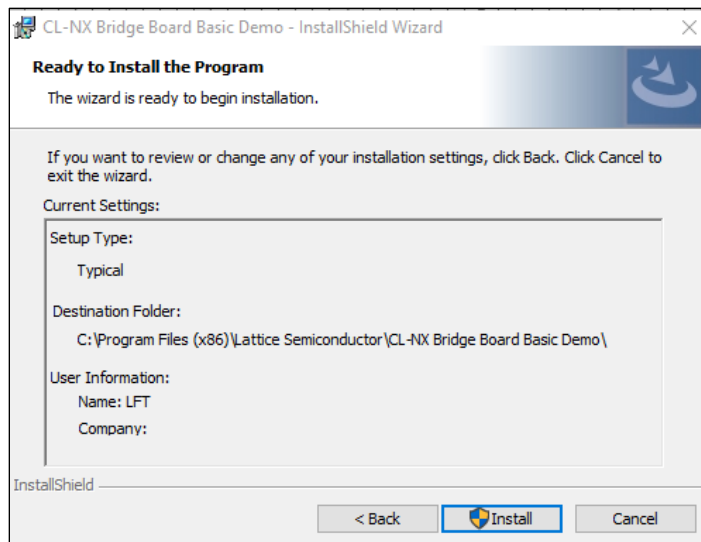


Figure 5.29. Windows Installer: Summary Page

6. Installation of the Basic Demo application starts. [Figure 5.30](#) shows the installation is completed.

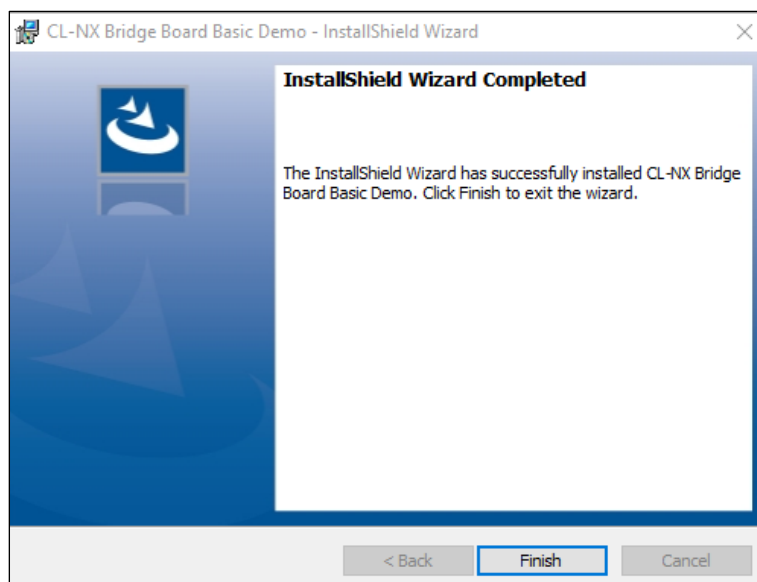


Figure 5.30. Windows Installer: Application Installed

7. When a message box pops up, click **Yes**.

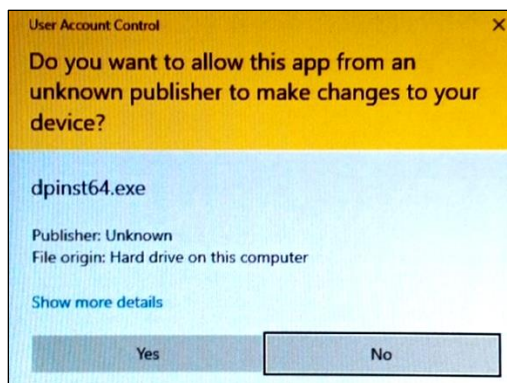


Figure 5.31. Device Configuration Prompt

8. The device driver installation wizard opens. Click **Next**.

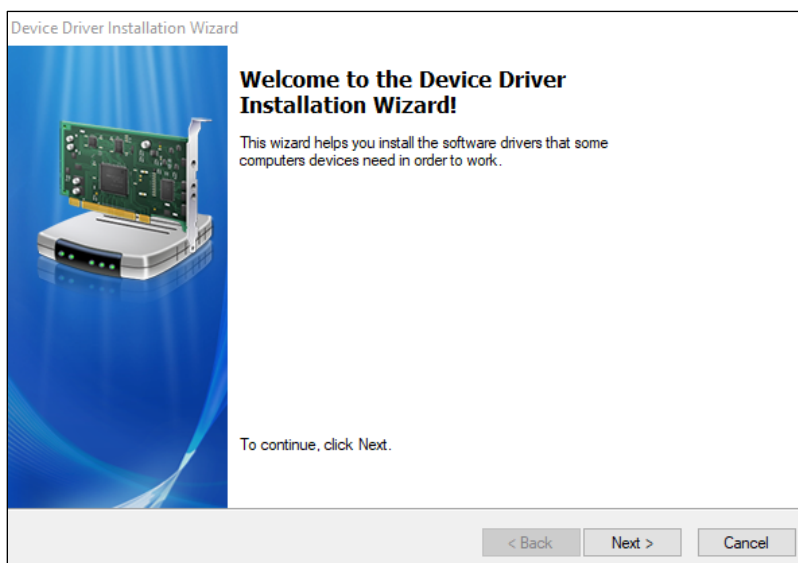


Figure 5.32. Device Driver Installation Wizard

9. If a Windows Security prompt is received, select **Install this driver software anyway** as shown in [Figure 5.33](#).

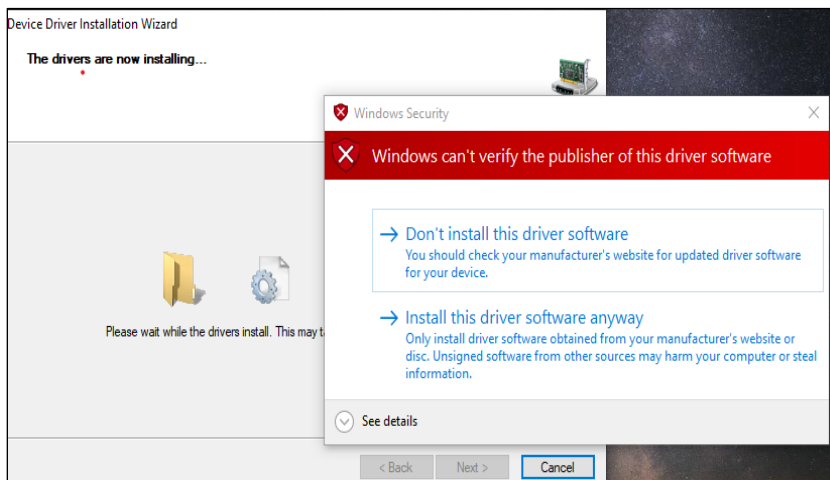


Figure 5.33. Windows Security in Driver Installation

10. If the driver is installed successfully, a message is displayed as shown in [Figure 5.34](#).

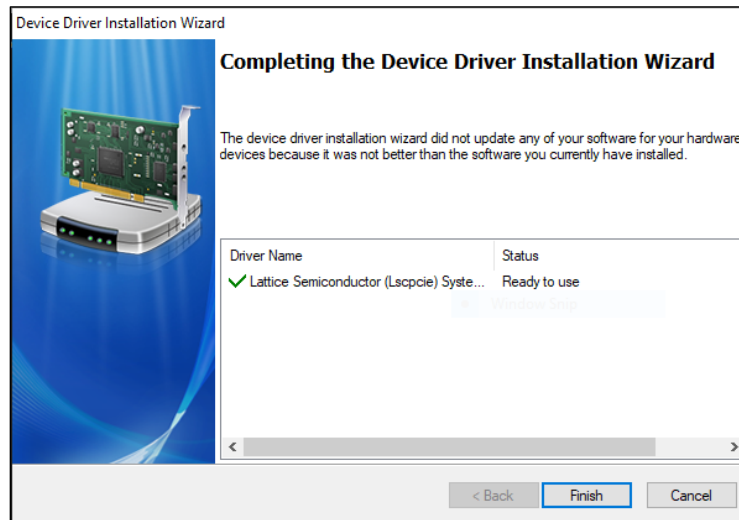


Figure 5.34. Device Driver Installation Completed

Installing Basic Demo Device Driver Manually

The drivers for the Basic Demo can be found in the *Demonstration\Windows\Driver* folder.

To install the Basic Demo device driver manually:

1. Open **Device Manager**, which shows the connected PCIe devices, as shown in [Figure 5.35](#).

Note: If the Device Manager does not show the connected PCIe device, see [Troubleshooting](#) section.

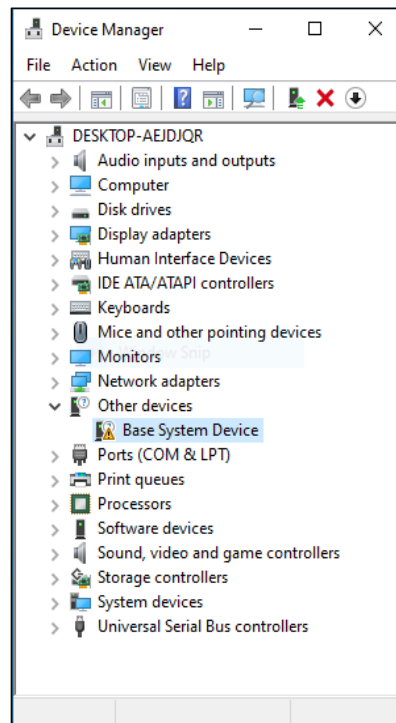


Figure 5.35. Device Manager

2. Right-click the device and select **Properties** as shown in [Figure 5.36](#).

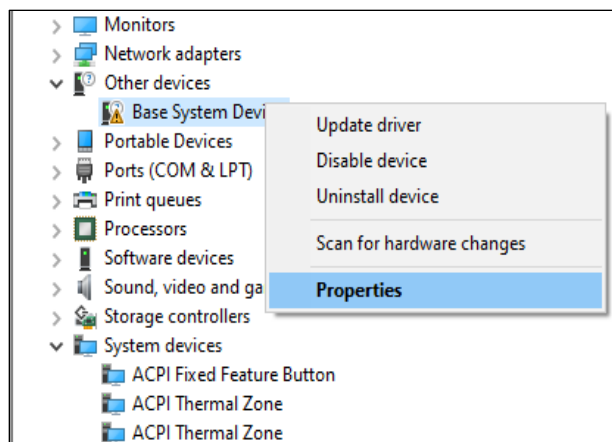


Figure 5.36. Showing Device Properties

3. The information in **Hardware IDs** is needed to install the correct driver for the corresponding device. For CrossLink-NX, the Hardware IDs of the Base System Device for the Basic Demo is `PCI\VEN_1204&DEV_9C1D&SUBSYS_E00419AA` as shown in Figure 5.37.

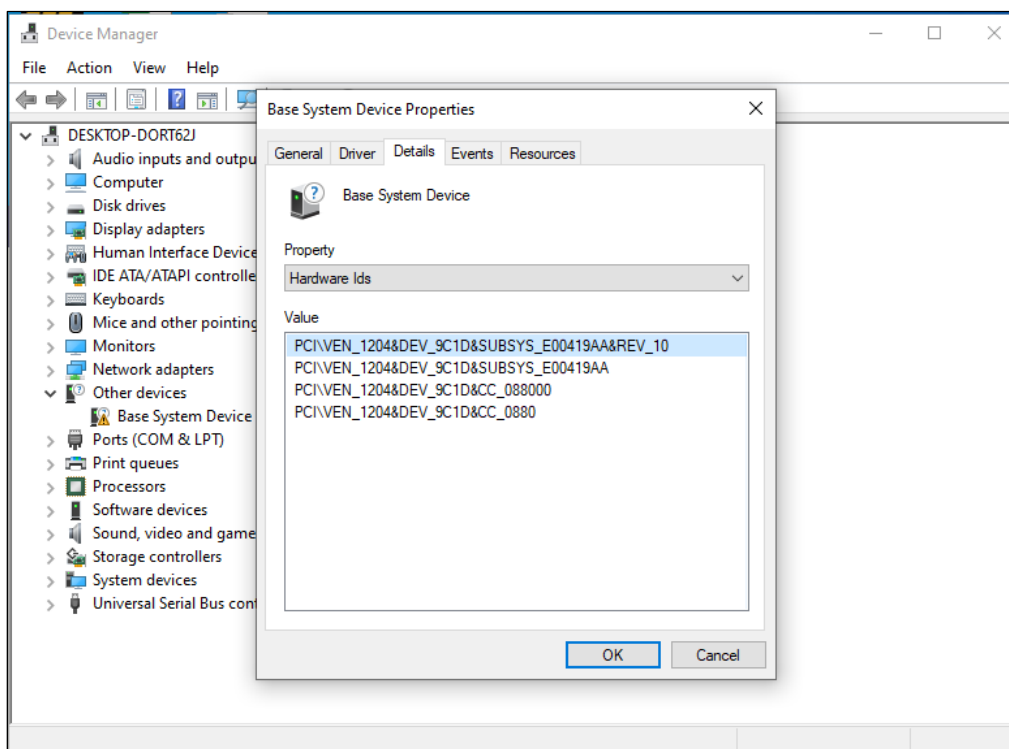


Figure 5.37. Hardware IDs of Basic Demo Device for CrossLink-NX

For Certus-NX, the Hardware IDs of the Base System Device for the Basic Demo is `PCI\VEN_1204&DEV_9C1D&SUBSYS_E00419AA` as shown in Figure 5.38.

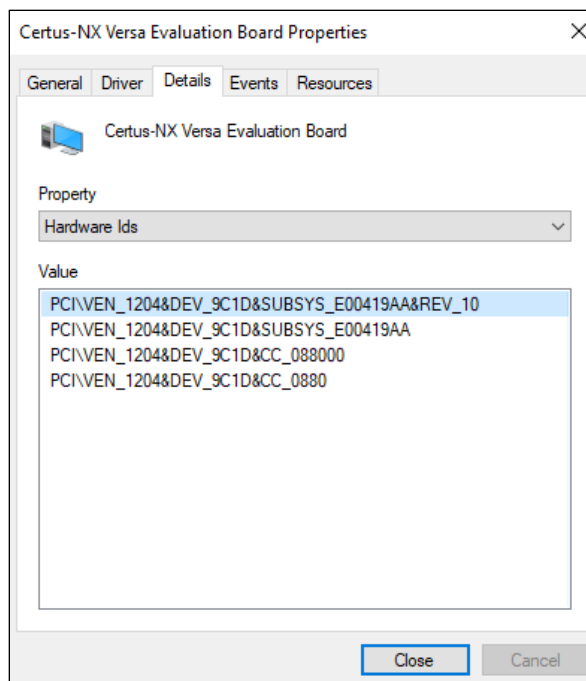


Figure 5.38. Hardware IDs of Basic Demo Device for Certus-NX

For CertusPro-NX, the Hardware IDs of the Base System Device for the Basic Demo is `PCI\VEN_1204&DEV_9C25&SUBSYS_E00419AA` as shown in [Figure 5.39](#).

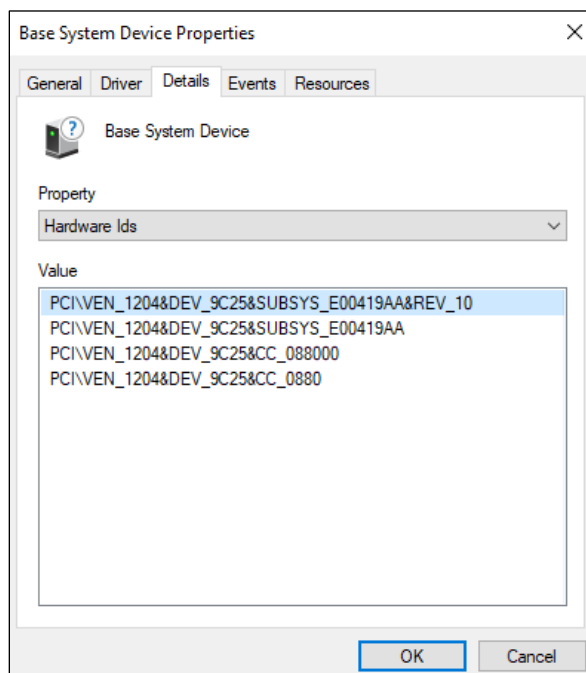


Figure 5.39. Hardware IDs of Basic Demo Device for CertusPro-NX

4. Right-click on the device and select **Update driver**.

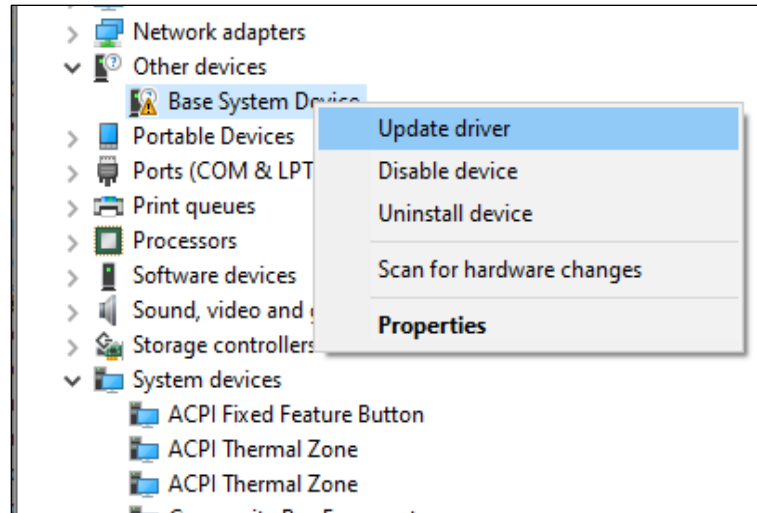


Figure 5.40. Update Driver Menu in Device Manager

5. Select the **Browse my computer for driver software** option as shown in [Figure 5.41](#).

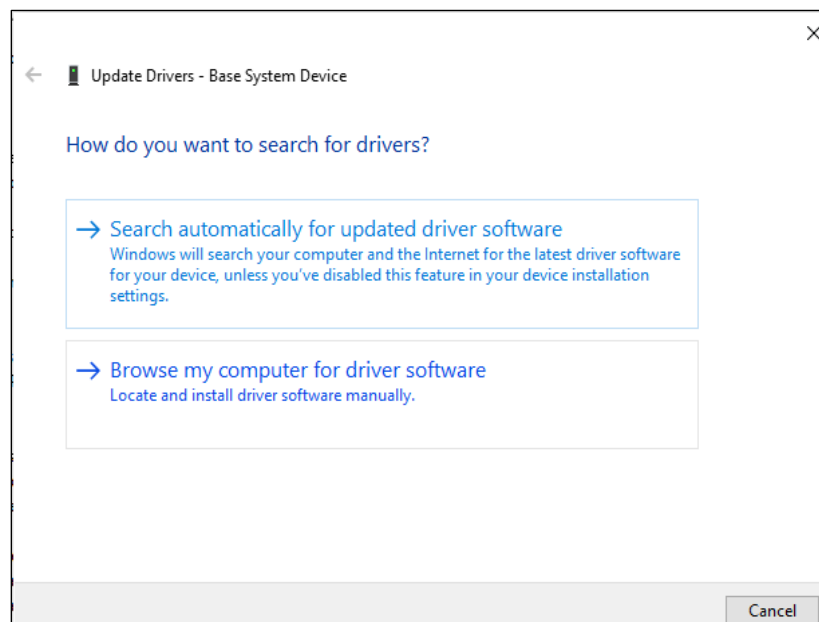


Figure 5.41. Update Driver Options

6. Browse for the Basic Demo device driver as shown in [Figure 5.42](#).

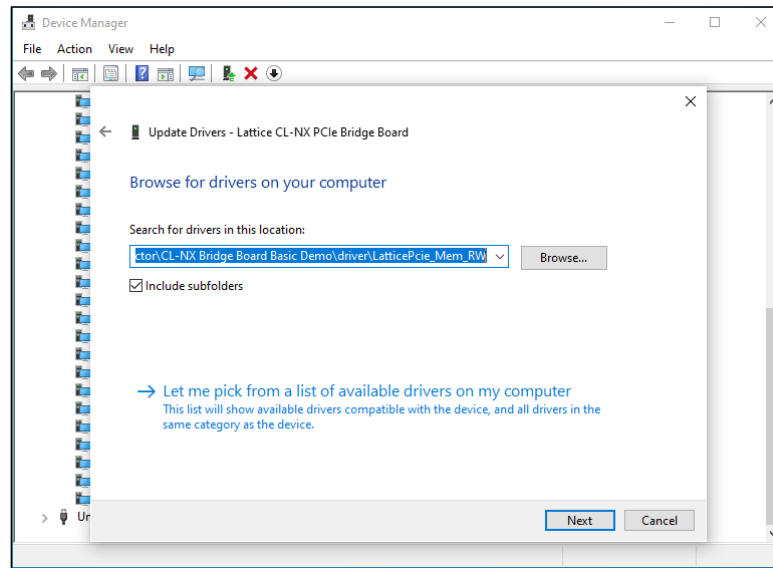


Figure 5.42. Browse the Driver for Device

7. If a Windows Security prompt is received, select **Install this driver software anyway** as shown in [Figure 5.43](#).

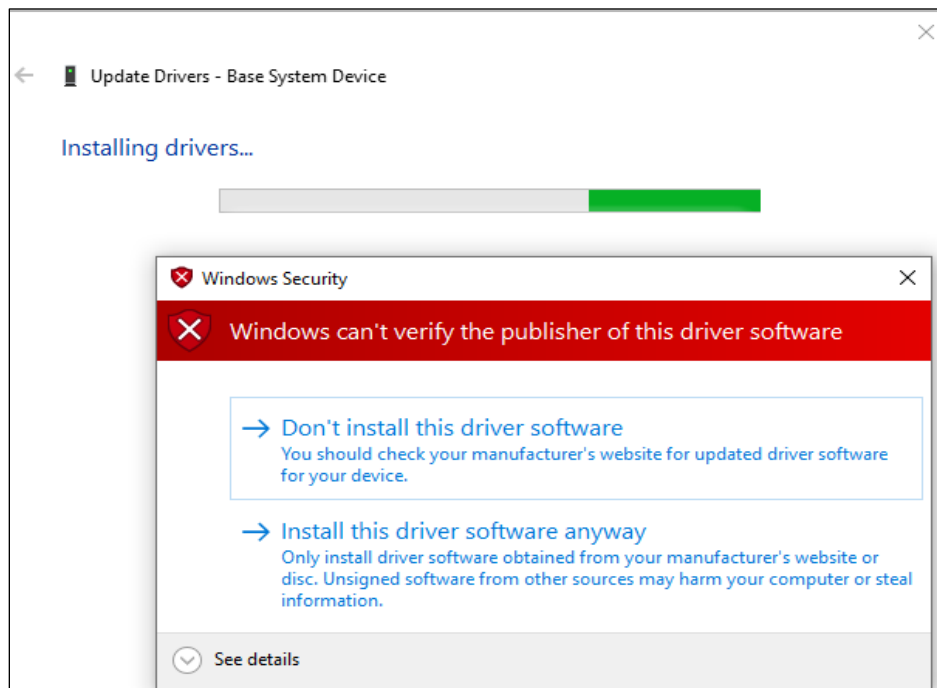


Figure 5.43. Windows Security in Device Manager

8. If the driver is installed successfully, a message is displayed as shown in [Figure 5.44](#) for CrossLink-NX, [Figure 5.45](#) for Certus-NX, and [Figure 5.46](#) for CertusPro-NX.

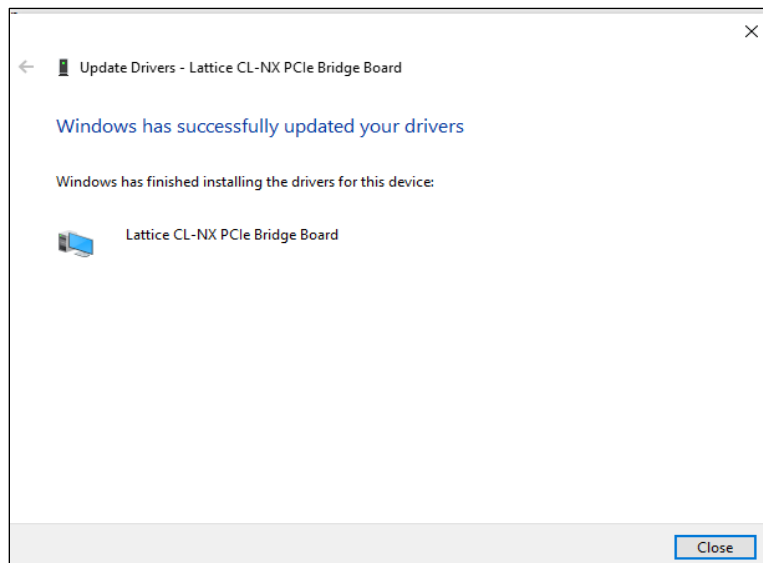


Figure 5.44. Driver Installation Status Message for CrossLink-NX

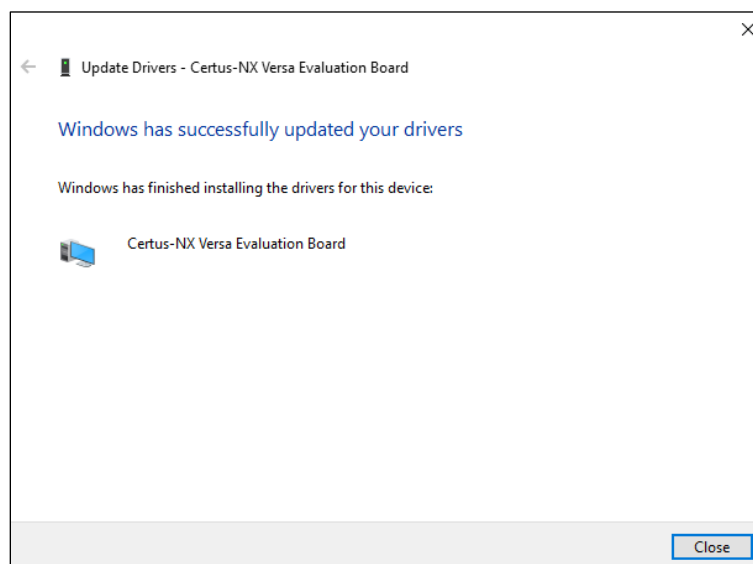


Figure 5.45. Driver Installation Status Message for Certus-NX

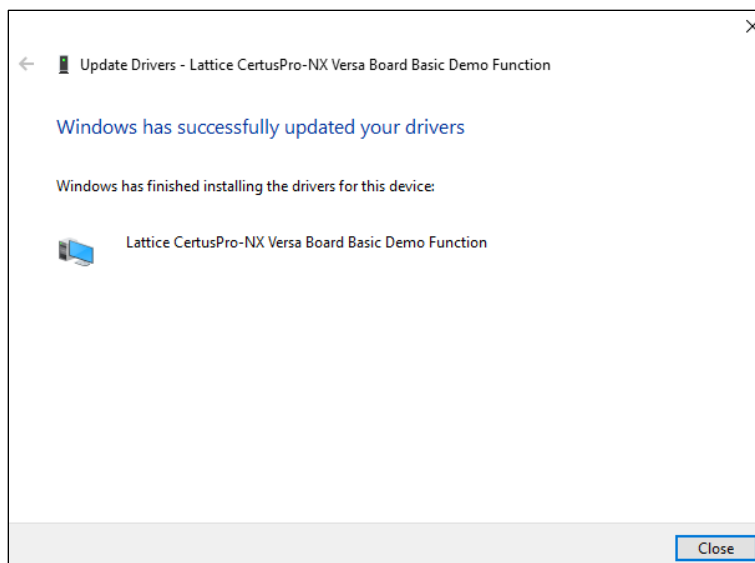


Figure 5.46. Driver Installation Status Message for CertusPro-NX

- After the driver is installed, the device driver name is displayed in **Device Manager**, as shown in Figure 5.47 for CrossLink-NX, Figure 5.48 for Certus-NX, and Figure 5.49 for CertusPro-NX.

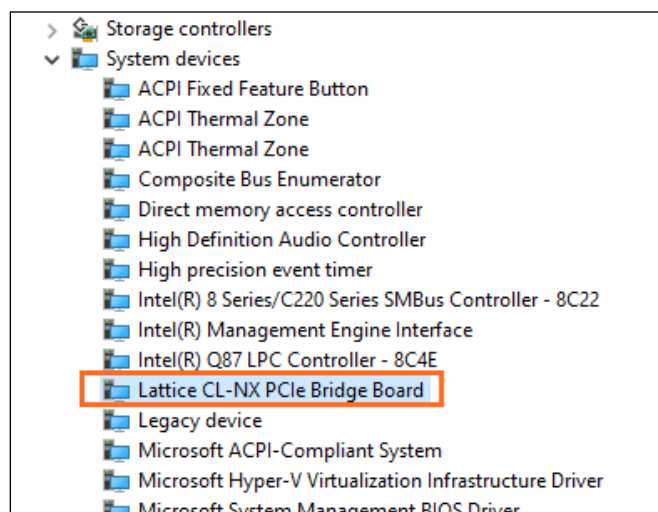


Figure 5.47. Basic Demo Device Name Displayed in Device Manager for CrossLink-NX

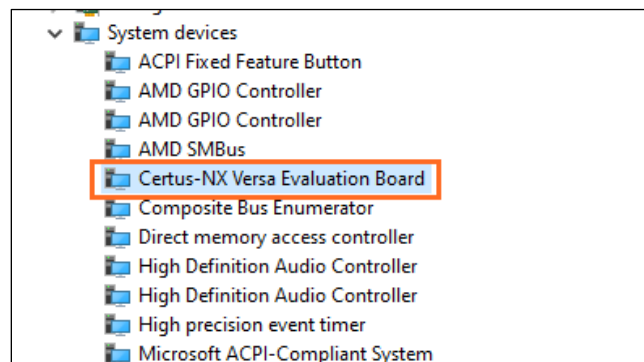


Figure 5.48. Basic Demo Device Name Displayed in Device Manager for Certus-NX

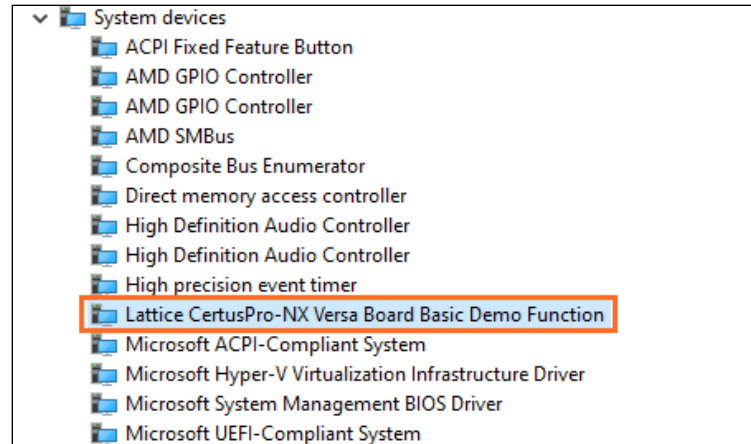


Figure 5.49. Basic Demo Device Name Displayed in Device Manager for CertusPro-NX

5.2.1.4. Driver Uninstallation

Follow the steps below to uninstall the device driver.

1. Open **Device Manager**. Identify the targeted device driver.
2. Right-click on the device driver and select **Uninstall device** as shown in [Figure 5.50](#).

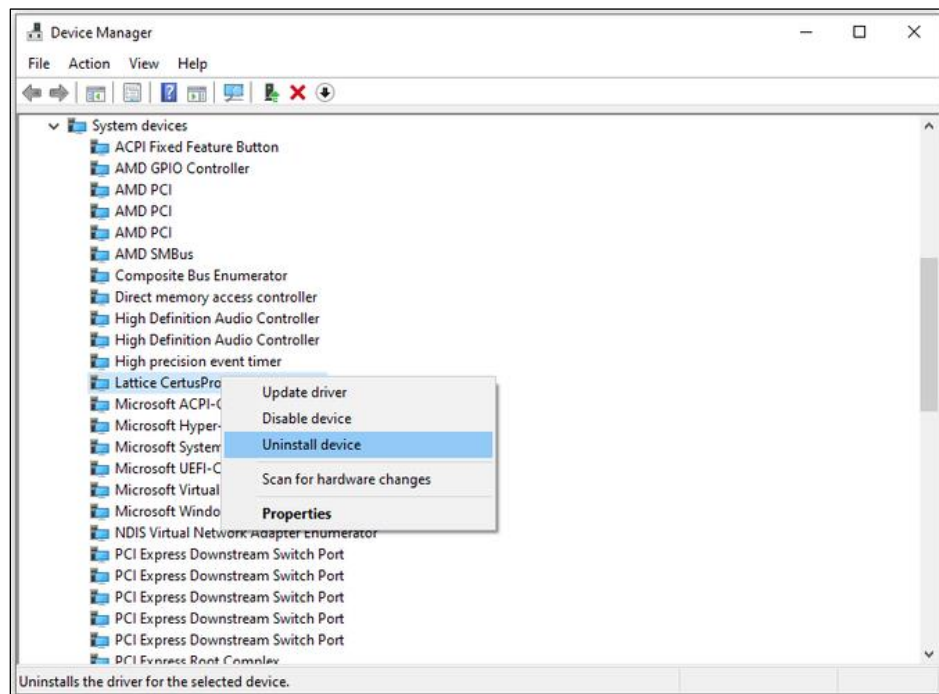


Figure 5.50. Uninstall the Device in Device Manager

3. Checked the option **Delete the driver software for this device** and click on **Uninstall** as shown in [Figure 5.51](#).

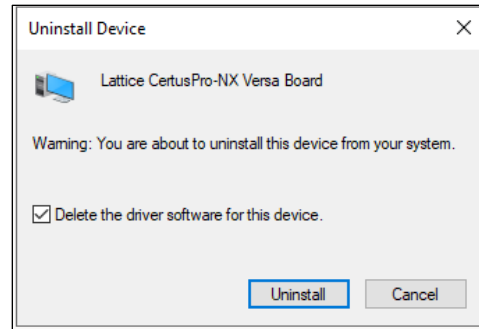


Figure 5.51. Uninstall Device Dialog Box

4. In the **Device Manager**, click the **Action** menu and select **Scan for hardware changes**. Ensure that the uninstalled device is not listed in the **Device Manager**.

5.2.2. Software Setup for Linux

5.2.2.1. Supported Operating System

- Distribution: Ubuntu
- Description: Ubuntu 24.04.2 LTS
- Release: 24.04.2

5.2.2.2. Required Packages

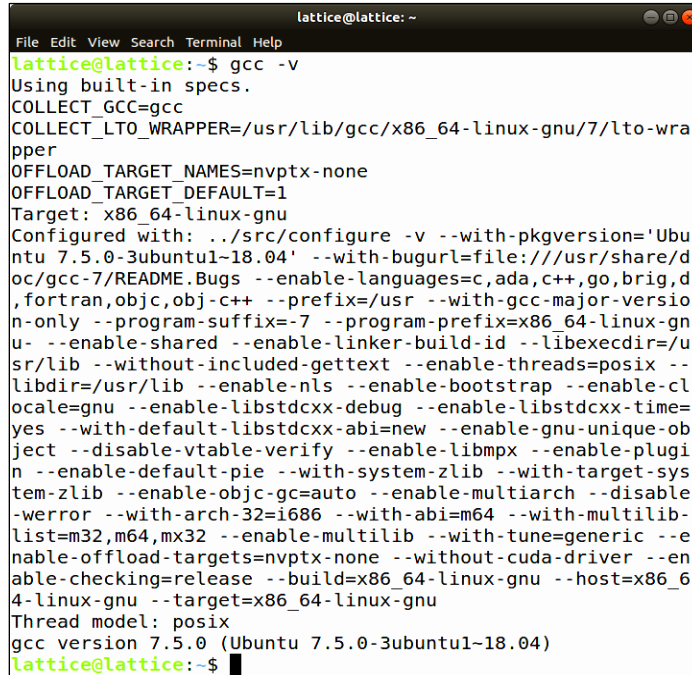
To check whether the required packages are installed or not, run the following commands on a terminal as shown below.

```
make -v
```

```
lattice@lattice: ~  
File Edit View Search Terminal Help  
lattice@lattice:~$ make -v  
GNU Make 4.1  
Built for x86_64-pc-linux-gnu  
Copyright (C) 1988-2014 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
lattice@lattice:~$
```

Figure 5.52. Display Make Utility in Linux Terminal

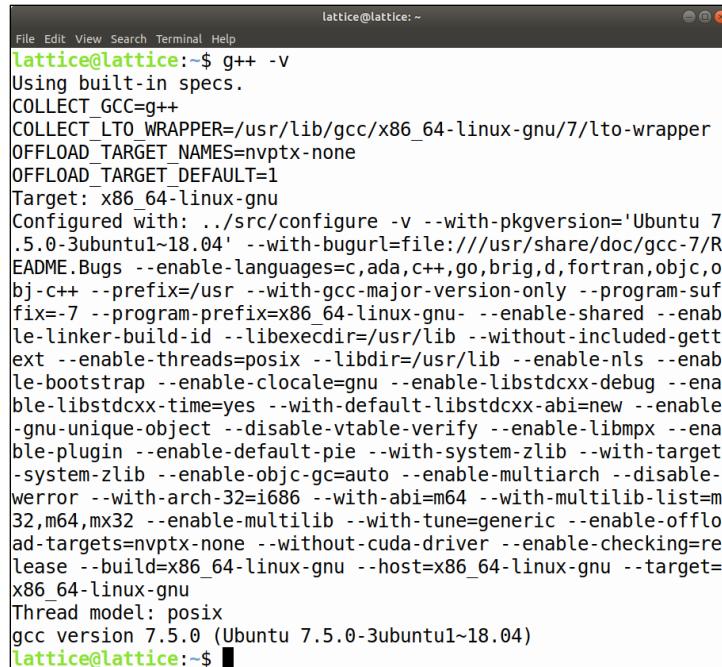
```
gcc -v
```



```
lattice@lattice: ~
File Edit View Search Terminal Help
lattice@lattice:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-3ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
lattice@lattice:~$
```

Figure 5.53. Display Compiler Version in Linux Terminal

```
g++ -v
```



```
lattice@lattice: ~
File Edit View Search Terminal Help
lattice@lattice:~$ g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.5.0-3ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bootstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
lattice@lattice:~$
```

Figure 5.54. Update Compiler Version in Linux Terminal

5.2.2.3. Packages Installation Steps

Run the following commands on a terminal to install the required packages

```
sudo apt update
sudo apt install build-essential
sudo apt install flex
```

5.2.2.4. Automatic Setup and Installation

To setup the demo in an automatic mode:

1. Go to the *Demonstration/Linux* directory.
2. Change the permission of the *install.sh* file by running *chmod* command.

```
sudo chmod 777 install.sh
```
3. Run the command below, which builds the driver and API library, install the driver, and launch the application user interface.

```
sudo ./install.sh
```
4. To uninstall the driver, go to the *Demonstration/Linux* directory and run the command below.

```
sudo ./uninstall.sh
```

5.2.2.5. Manual Setup and Installation

Note: This section may be skipped if the driver is installed.

Before installing the driver, the driver must be built.

To build the driver:

1. Go to the *Demonstration* directory.
2. Run the following command on terminal.

```
sudo chmod 777 -R Linux
```
3. Go to the *Linux* directory.
4. Go to in *Source_Code* directory.
5. Run the following commands.

```
sudo make clean
sudo make
```

This builds the driver and API library.
6. Make sure that the driver is not installed.
7. To remove an existing driver, run the following command in a terminal window:

```
sudo rmmod lattice_main.ko
```

To install the driver:

1. Go to *drv_src/drvr/* directory.
2. Run the command below.

```
sudo insmod lattice_main.ko
```
3. To launch the user interface application, go to the *app_src/gui/deploy/* directory and run the command below.

```
sudo ./PCIe_Test_App.sh
```

6. Application Overview

The PCIe Basic demo shows the capabilities of the Lattice FPGA and the PCI Express Endpoint IP core functionality in a PCI Express slot in a Linux (Ubuntu)/Windows PC.

This demo software allows you to access memory and registers on the board and provides real time interaction with the hardware to demonstrate a functional PCI Express communications path between the application, driver software and the FPGA IP.

6.1. Running the PCI Express Test Application

This section describes how to run PCIe Test Application software after installation.

On Linux, this application can be launched by running the `sudo ./PCle_Test_App.sh`, which is located in the `Demonstration/Linux/Source_Code/app_src/gui/deploy` directory.

On Windows, click the Basic Demo shortcut icon on the desktop. If the shortcut icon is accidentally deleted, the application user interface can still be launched by double clicking on `lattice_bd.exe` file located in `<INSTALLATION FOLDER>\Lattice Semiconductor\<CL-NX Bridge Board Basic Demo/Certus-NX Basic Demo/CertusPro-NX Basic Demo>\bin` folder.

The graphical user interface opens the PCI Express Test Application software with the **Device Info** tab selected as shown in [Figure 6.1](#).

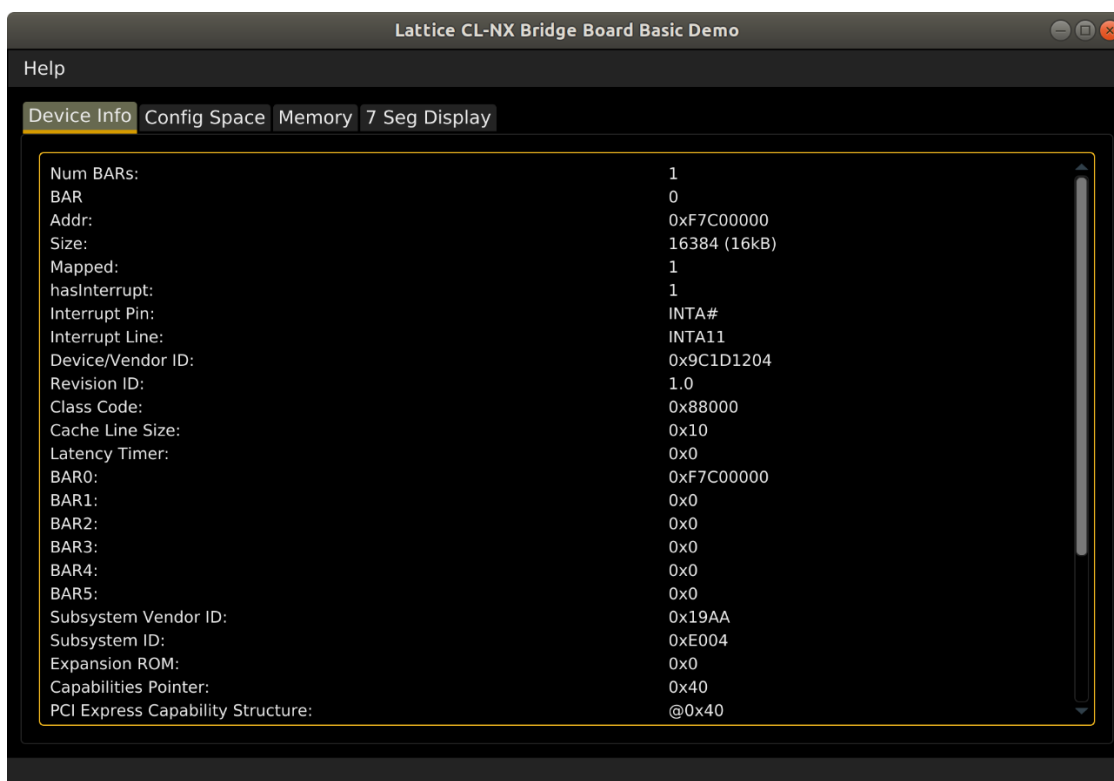


Figure 6.1. PCIe Test Application Device Info Tab for CrossLink-NX

6.2. Using the PCI Express Test Application User Interface

This section describes the pages and features of the PCI Express Test Application software user interface.

The PCI Express **Device Info** page displays information about the device driver and device's PCI configuration registers. The data displayed is for information purpose only and cannot be edited. If there is a problem in accessing the inserted device or the device is not inserted or not properly inserted, an error message appears in this page.

Table 6.1 describes the **Device Info** tab.

Table 6.1. Device Info Tab

Feature	Description
Lattice Device Info page	Displays windows resource assigned to the device driver to access the board. Obtained from the board PCI Config space registers by the Lattice PCI Express driver when the application is started.

Clicking on the **Config Space** tab displays the PCI Configuration registers in 8-bit, 16-bit, and 32-bit mode. When the application is started, 32-bit mode is selected and PCI Configuration Space is displayed. Selecting the 8-bit, 16-bit, or 32-bit radio button displays the corresponding configuration space registers. The data displayed is for information purpose only and cannot be edited.

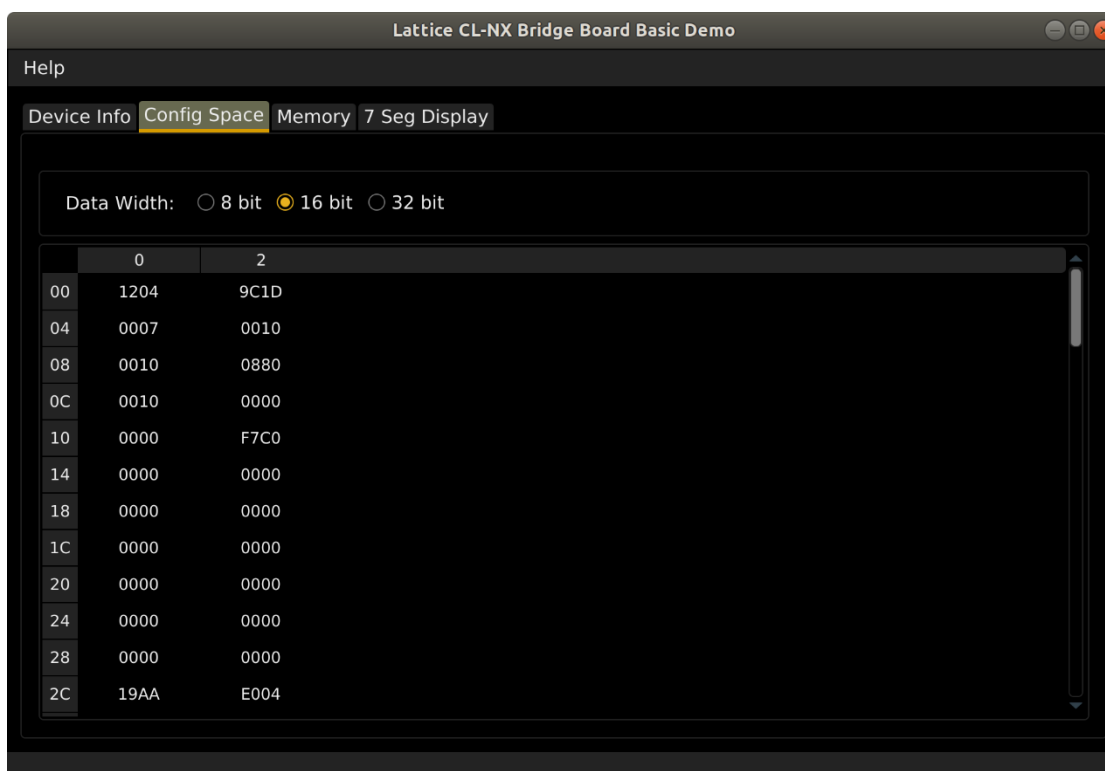


Figure 6.2. PCIe Test Application Config Space Tab

Table 6.2 provides descriptions of the **Config Space** tab.

Table 6.2. Configuration Space Tab

Feature	Description
Data Width radio buttons	8-bit, 16-bit and 32-bit radio buttons allow to select one of the modes
Configuration Space Info area	Displays the standard PCI Configuration Space

6.3. Running PCIe Basic Demo

The 7 Seg Display tab controls the 7-segment LED. In this tab, you can run a demonstration LED sequence and controls the display on the board from the application. The 7 Seg Display tab provides a way to interactively toggle segments on the display.

The states of the LED segments are converted to an 8-bit word value (each segment is controlled by a bit) and written to the LED control register in the FPGA. This demonstrates a memory write across the PCI Express bus.

For CrossLink-NX and CertusPro-NX, there are three digits of 7-segment displays. For Certus-NX, there is only one digit of 7-segment display.



Figure 6.3. Three 7-Segment Displays on Board for CrossLink-NX and CertusPro-NX



Figure 6.4. Single 7-Segment Display on Board for Certus-NX

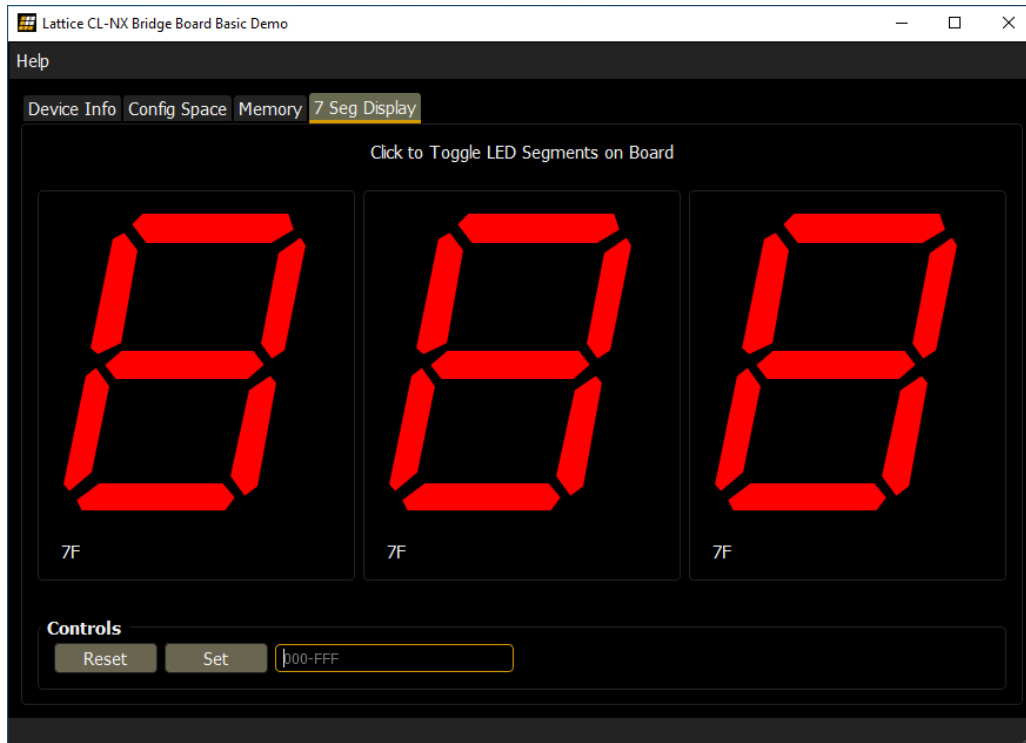


Figure 6.5. Interactive 7 Seg Display Tab for CrossLink-NX and CertusPro-NX

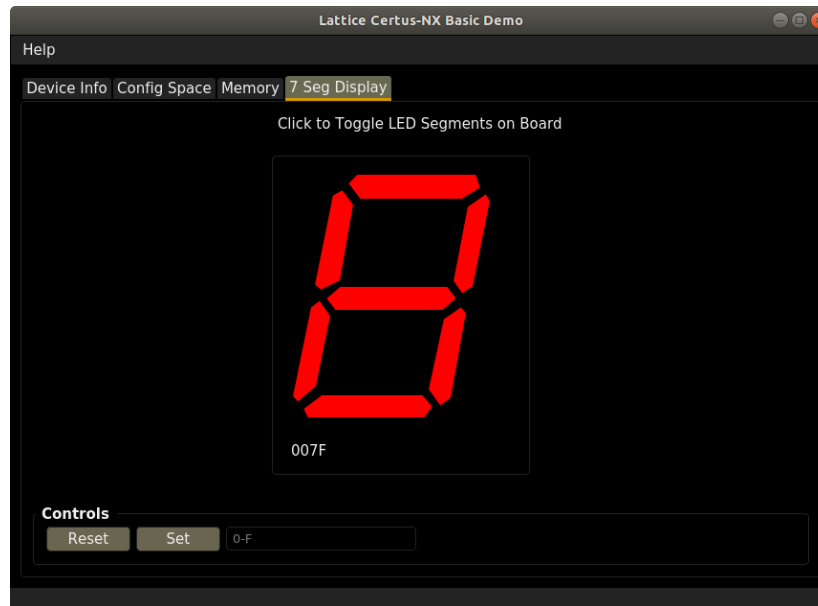


Figure 6.6. Interactive 7 Seg Display Tab for Certus-NX

Click on any segment in the interactive segment display in the same tab. Notice that any selection immediately causes the corresponding segment on the LED to light on the board's LED display. Clicking on a segment turns it on or off.

In the Controls panel, click the Reset button turns off all segments of the LED display. Enter a character between 0 and F and click on the Set button. This toggles on the respective segments both on the board and the LED in this tab.

The interactive **7 Seg Display** tab demonstration illustrates that a functional PCI Express communications path exists between the application and driver software on the CPU and the FPGA IP.

Alternatively, the 7-segment display(s) may be controlled through the memory write data. Go to the **Memory** tab, select either the 8-bit, 16-bit, or 32-bit radio button. In the Write Offset field, enter 0x0. In the Data field, enter the value as stated in [Table 6.3](#) and click the Write button. The first byte controls the first digit of the display, the second byte controls the second digit of the display, and the third byte controls the third digit of the display. After each write operation, you may verify that the correct segments are turned on in the *7 Segment Display* tab and in the 7-segment display(s) of the board.

Table 6.3. Memory Data and Corresponding 7-Segment Display Values

Hex value to be entered in the Data field at offset 0x0	Corresponding 7-Segment Display Value
3F	0
06	1
5B	2
4F	3
66	4
6D	5
7D	6
07	7
7F	8
6F	9
77	A
7C	B
39	C
5E	D
7B	E
71	F

6.4. Running PCIe Memory Access Demo

The **Memory** tab provides several elements to control the read-write process. [Table 6.4](#) provides descriptions of the Memory tab.

When this tab is clicked the whole contents of the EBR memory is displayed in Memory Contents table, based on the radio button selected. Data access can be specified as 8-bit, 16-bit, or 32-bit operations by selecting Data Size. Access is to the selected Base Address Register (BAR).

Data can be written to the registers by clicking the Write button. Specify the BAR offset to start writing at and the hex data in the Data field. The data size must match the data width selected at the top to the page.

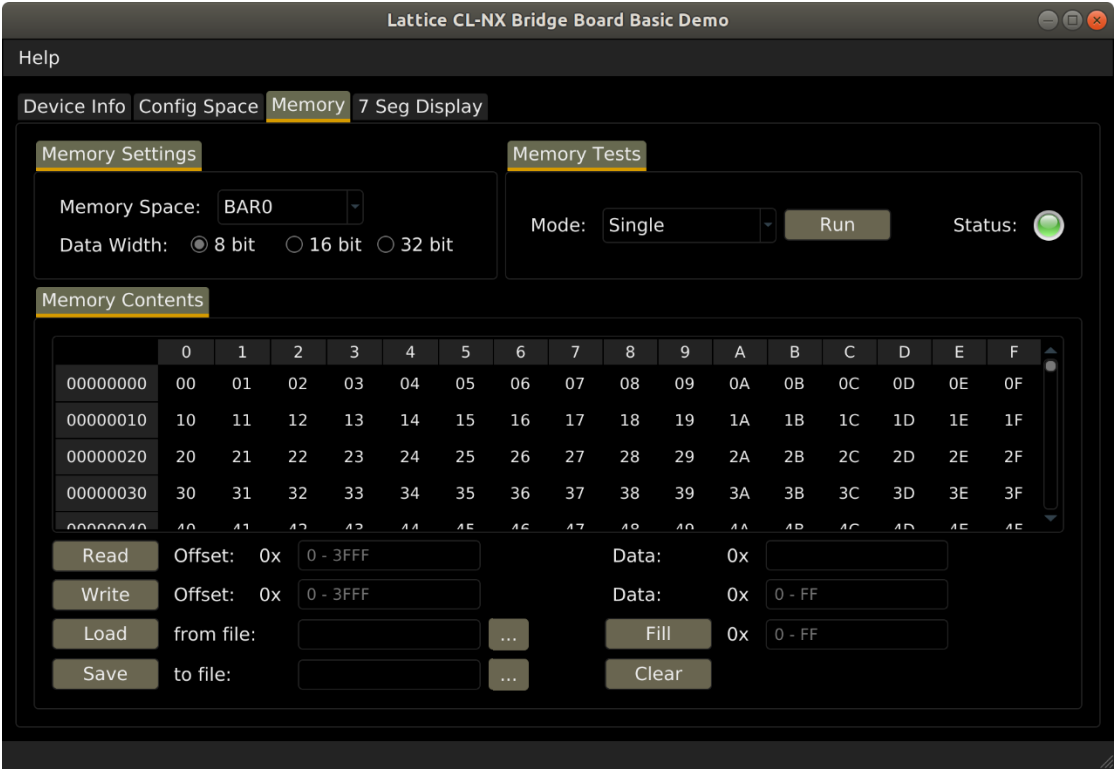


Figure 6.7. PCIe Test Application Memory Tab

Table 6.4. Memory Tab Description

Feature	Description
Memory Space	Indicates the base address register (BAR) memory space to access.
Data Width	Contains radio buttons that control the size of data that is being written or read.
Memory Tests	In Memory Test mode, sequential data is written and read back to check that all locations of EBR properly accessed.
Test mode dropdown	There are two test modes: Single and Continuous . In Single mode, the write read operation is done only once for all memory locations; while in Continuous mode, the write read operation is done in a loop until you choose to stop.
Run	Runs the selected test in background thread.
Status LED	Shows the result of running test.
Memory Contents	Displays the 4KB memory contents.

Feature	Description
Read Offset	The offset from where data is to be read. Range of address depends on the bar size. In 16-bit mode, this address must be a multiple of 2. In 32-bit mode, the address must be a multiple of 4.
Read Data	The data read from the BAR address is displayed here. In 8-bit mode, maximum value of data can be 0xFF; in 16-bit mode, it can be 0xFFFF and in 32-bit mode, 0xFFFFFFFF is the maximum value that can be entered.
Write Offset	The offset where data is to be written. Range of address depends on the bar size. In 16-bit mode this address must be a multiple of 2 and in 32-bit mode address must be a multiple of 4.
Write Data	The data to be written is entered here. The maximum value that can be written depends on the data width. The maximum value in 8-bit mode is 0xFF, in 16-bit mode it is 0xFFFF, and in 32-bit mode it is 0xFFFFFFFF.
Load	Loads 4 kB of binary data from the file specified (or as much data as in the file) into EBR memory starting at location 0. This can be used to load a known pattern into the EBR memory by using a file.
Save	Saves the entire 4 kB contents of the EBR memory into the specified file. This can be used to save the contents of EBR for off-line processing. The file is saved in binary format (.bin).
Fill	Writes the value entered in the field to all 4KB locations.
Clear	Sets all 4 kB to 0.

When using the file saving operations, ensure write permission for the location where the file is being saved. Similarly, for file reading operations, the directory from where the file is being loaded must have read permission.

7. Customizing the Design for CertusPro-NX

The Basic Demo Design for Certus-Pro NX can be customized to the following:

- Change PCIe lane width from x4 to x2 or x1
- Change PCIe link speed from Gen3 to Gen2 or Gen1

7.1. Changing the PCIe Lane Width

To customize the lane width manually, perform the following:

1. Open the Lattice Radiant software.
2. Click **Open project** and browse to the .rdf file.
 - For CertusPro-NX, the file is *LFCPNX100_PCIe_BasicDemo.rdf*, which is in the *Hardware/BasicDemo/* folder. It is recommended to save as a new project for any design customization. Click **Save Project As** to save the project with a new name and open the new project.

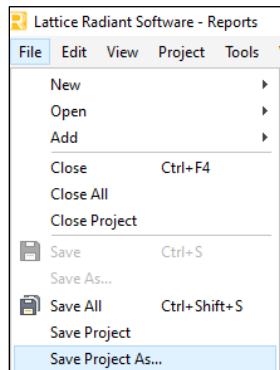


Figure 7.1. Save New Project

3. Click **Open** the PCIe4 IP.

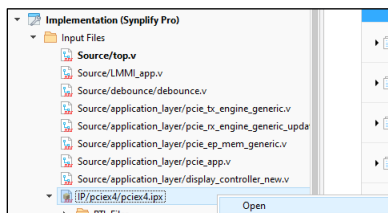


Figure 7.2. Open PCIe4 IP

4. Select the Lane Width from the *Bifurcation Select* tab as following:
 - a. Select 1x1 value for x1 lane width.
 - b. Select 1x2 value for x2 lane width.
 - c. Select 1x4 value for x1 lane width.

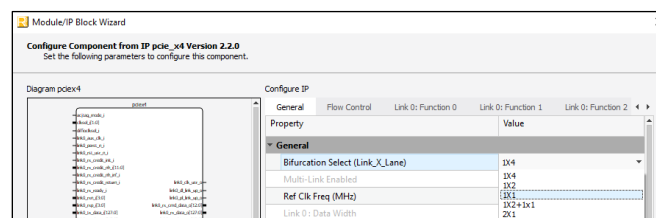


Figure 7.3. PCIe IP Configuration to Select Lane Width

5. Click **Generate** to re-generate the PCIe_x4 IP instance.

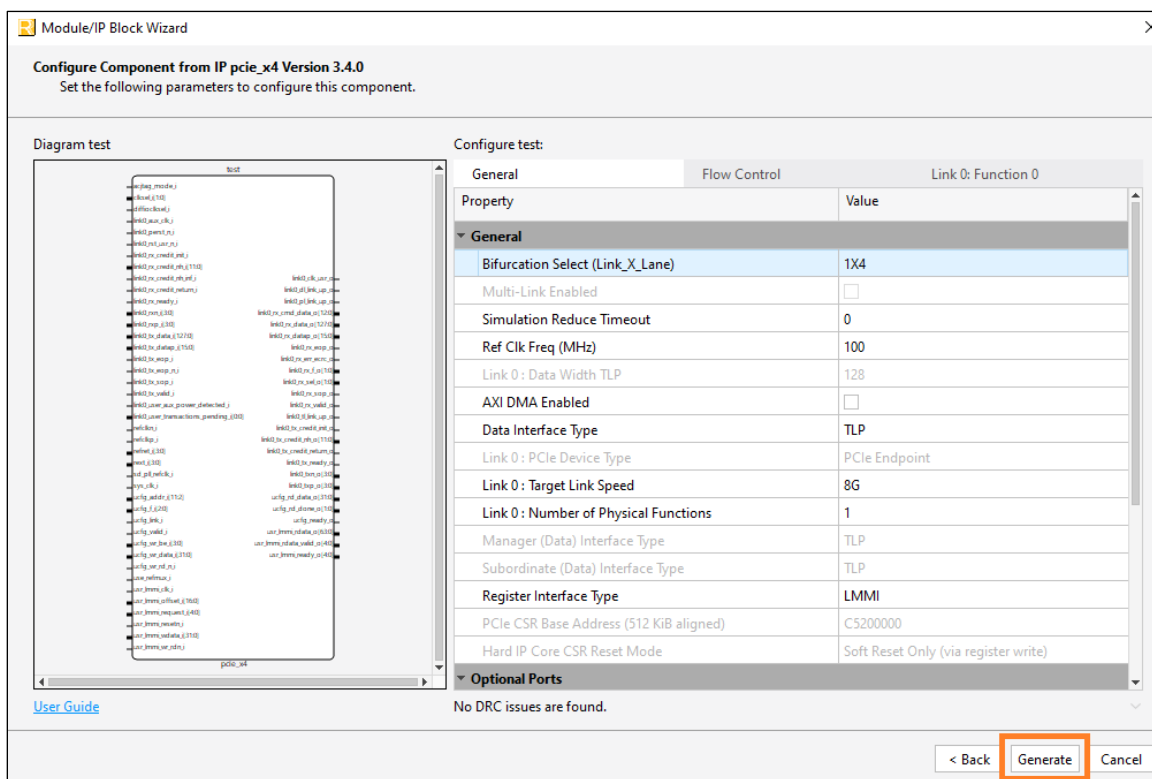


Figure 7.4. IP Generation

6. Open the top level design file located in `Hardware/BasicDemi/Source/top.v` in the Radiant tool or in the notepad.
 - a. Specify appropriate lane width for x2 and x1 configuration by inserting single line comments in the top-level design file as shown in [Figure 7.5](#). This reduces the input and output ports required for x2 and x1 configuration.

```

4 module fpga_top #(
5     parameter SIM = 0
6 )
7 (
8     input  sdq0_refclkp,
9     input  sdq0_refclkn,
10    input  rext_0,
11    input  rext_1,          //Comment for x1
12    input  rext_2,          //Comment for x1 and x2
13    input  rext_3,          //Comment for x1 and x2
14    input  refret_0,
15    input  refret_1,        //Comment for x1
16    input  refret_2,        //Comment for x1 and x2
17    input  refret_3,        //Comment for x1 and x2
18    input  ch0_sdrxp,
19    input  ch1_sdrxp,        //Comment for x1
20    input  ch2_sdrxp,        //Comment for x1 and x2
21    input  ch3_sdrxp,        //Comment for x1 and x2
22    input  ch0_sdrxn,
23    input  ch1_sdrxn,        //Comment for x1
24    input  ch2_sdrxn,        //Comment for x1 and x2
25    input  ch3_sdrxn,        //Comment for x1 and x2
26    output ch0_sdtxp,
27    output ch1_sdtxp,        //Comment for x1
28    output ch2_sdtxp,        //Comment for x1 and x2
29    output ch3_sdtxp,        //Comment for x1 and x2
30    output ch0_sdtxn,
31    output ch1_sdtxn,        //Comment for x1
32    output ch2_sdtxn,        //Comment for x1 and x2
33    output ch3_sdtxn,        //Comment for x1 and x2
34    input  perst_n_i,

```

Figure 7.5. Selecting Input and Output Ports

- b. Add or remove comment lines in the PCIe Instance module to select the appropriate lane width as described in the design file.

```

204 pciex4 u_PCIe_gen3
205 (
206     .link0_rxp_i({ch3_sdrxp,ch2_sdrxp,ch1_sdrxp,ch0_sdrxp}), //Selection for x4
207     .link0_rxp_i({ch1_sdrxp,ch0_sdrxp}), //Selection for x2
208     .link0_rxp_i(ch0_sdrxp), //Selection for x1
209
210     .link0_rxn_i({ch3_sdrxn,ch2_sdrxn,ch1_sdrxn,ch0_sdrxn}), //Selection for x4
211     .link0_rxn_i({ch1_sdrxn,ch0_sdrxn}), //Selection for x2
212     .link0_rxn_i(ch0_sdrxn), //Selection for x1
213
214     .refclkp_i(sdq0_refclkp),
215     .refclkn_i(sdq0_refclkn),
216
217     .link0_txp_o({ch3_sdtxp,ch2_sdtxp,ch1_sdtxp,ch0_sdtxp}), //Selection for x4
218     .link0_txp_o({ch1_sdtxp,ch0_sdtxp}), //Selection for x2
219     .link0_txp_o(ch0_sdtxp), //Selection for x1
220
221     .link0_txn_o({ch3_sdtxn,ch2_sdtxn,ch1_sdtxn,ch0_sdtxn}), //Selection for x4
222     .link0_txn_o({ch1_sdtxn,ch0_sdtxn}), //Selection for x2
223     .link0_txn_o(ch0_sdtxn), //Selection for x1
224
225     .refret_i({refret_3,refret_2,refret_1,refret_0}), //Selection for x4
226     .refret_i({2'b00,refret_1,refret_0}), //Selection for x2
227     .refret_i({3'b000,refret_0}), //Selection for x1
228
229     .rext_i({rext_3,rext_2,rext_1,rext_0}), //Selection for x4
230     .rext_i({2'b00,rext_1,rext_0}), //Selection for x2
231     .rext_i({3'b000,rext_0}), //Selection for x1

```

Figure 7.6. Update PCIe Instance Module

- c. Change the PCIe IP interface data width by modifying the `DATA_WIDTH` parameter as shown in Figure 7.7.
 - Select 32 bus width for x1 lane width.
 - Select 64 bus width for x2 lane width.
 - Select 128 bus width for x4 lane width.

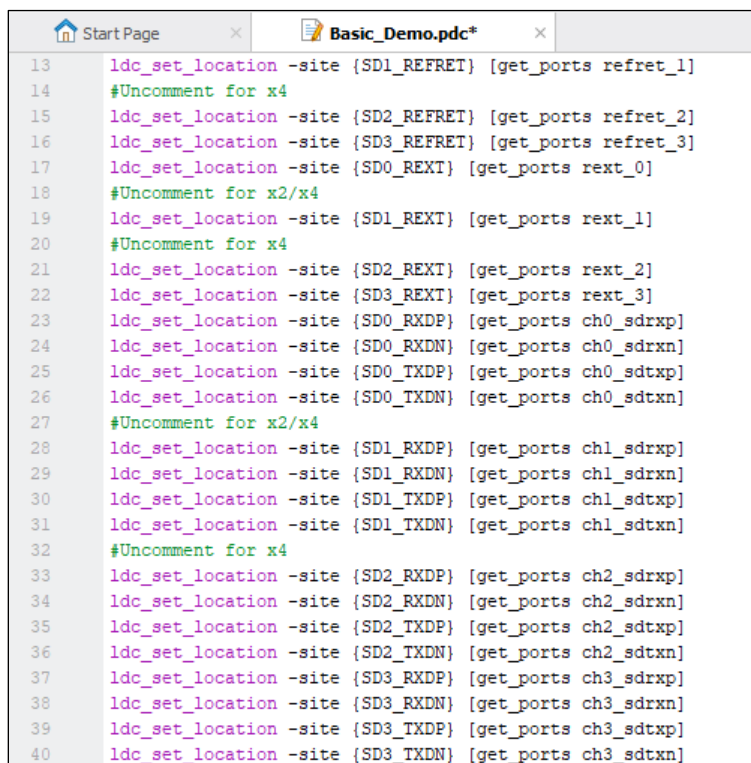
```

47 parameter DATA_WIDTH = 128;
48 //Select 32 for x1
49 //Select 64 for x2
50 //Select 128 for x4

```

Figure 7.7. Data Width Selection

7. Save the changes done in the top-level design file.
8. Open the PDC file located in `Hardware/BasicDemo/Source/constraint/Basic_Demo.pdc` using Radiant tool or notepad. Update the PDC file according to the lane width selection. Add comment lines using “#” on the `ldc_set_location` command according to the lane width selection. Save the changes.



```

13 ldc_set_location -site {SD1_REFRET} [get_ports refret_1]
14 #Uncomment for x4
15 ldc_set_location -site {SD2_REFRET} [get_ports refret_2]
16 ldc_set_location -site {SD3_REFRET} [get_ports refret_3]
17 ldc_set_location -site {SD0_REXT} [get_ports rext_0]
18 #Uncomment for x2/x4
19 ldc_set_location -site {SD1_REXT} [get_ports rext_1]
20 #Uncomment for x4
21 ldc_set_location -site {SD2_REXT} [get_ports rext_2]
22 ldc_set_location -site {SD3_REXT} [get_ports rext_3]
23 ldc_set_location -site {SD0_RXDP} [get_ports ch0_sdrxp]
24 ldc_set_location -site {SD0_RXDN} [get_ports ch0_sdrxn]
25 ldc_set_location -site {SD0_TXDP} [get_ports ch0_sdtxp]
26 ldc_set_location -site {SD0_TXDN} [get_ports ch0_sdtxn]
27 #Uncomment for x2/x4
28 ldc_set_location -site {SD1_RXDP} [get_ports ch1_sdrxp]
29 ldc_set_location -site {SD1_RXDN} [get_ports ch1_sdrxn]
30 ldc_set_location -site {SD1_TXDP} [get_ports ch1_sdtxp]
31 ldc_set_location -site {SD1_TXDN} [get_ports ch1_sdtxn]
32 #Uncomment for x4
33 ldc_set_location -site {SD2_RXDP} [get_ports ch2_sdrxp]
34 ldc_set_location -site {SD2_RXDN} [get_ports ch2_sdrxn]
35 ldc_set_location -site {SD2_TXDP} [get_ports ch2_sdtxp]
36 ldc_set_location -site {SD2_TXDN} [get_ports ch2_sdtxn]
37 ldc_set_location -site {SD3_RXDP} [get_ports ch3_sdrxp]
38 ldc_set_location -site {SD3_RXDN} [get_ports ch3_sdrxn]
39 ldc_set_location -site {SD3_TXDP} [get_ports ch3_sdtxp]
40 ldc_set_location -site {SD3_TXDN} [get_ports ch3_sdtxn]

```

Figure 7.8. Example of PDC File

9. Save the changes and recompile the design. Use the configuration bit file located in the new design. The bit file in the original design is available in `Hardware\BasicDemo\Implementation\BasicDemo_Implementation.bit`.

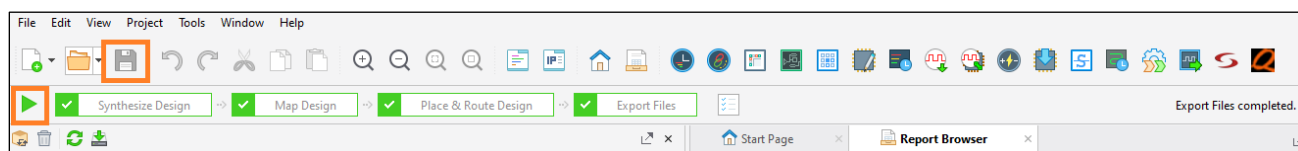


Figure 7.9. Save and Recomilation of the Design

7.2. Changing the PCIe Link Speed

To customize the link speed from Gen3 to Gen2 and Gen1 manually, perform the following steps:

1. Open the Lattice Radiant software.
2. Click **Open project** and browse to the .rdf file
 - For CertusPro-NX, the file is LFCPNX100_PCIe_BasicDemo.rdf, which is in the *Hardware/BasicDemo/* folder. It is recommended to save as a new project for any design customization. Click **Save Project As** to save the project with a new name and open the new project as shown in [Figure 7.1](#).
3. Click **Open** the PClex4 IP as shown in [Figure 7.2](#).
4. Select the value from the *Link 0: Target Link Speed* tab as following:
 - a. Select 8G value for Gen3 speed.
 - b. Select 5G value for Gen2 speed.
 - c. Select 2P5G value for Gen1 speed.

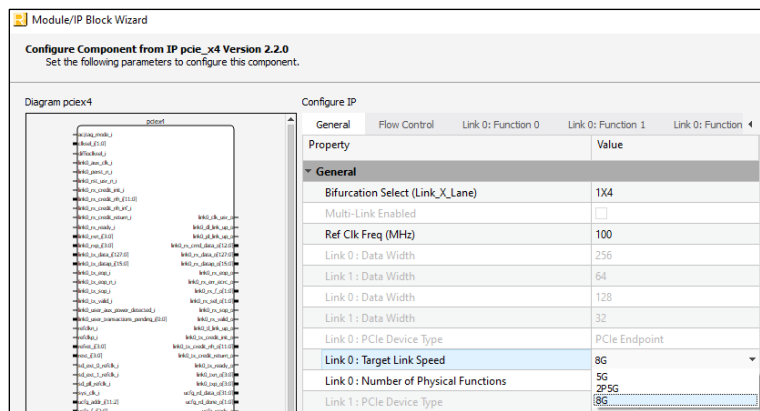


Figure 7.10. PCIe IP Configuration to Link Speed

5. Click **Generate** to re-generate the PClex4 IP instance as shown in [Figure 7.4](#).
6. Click **Open** the PLL IP instance as shown in [Figure 7.11](#).

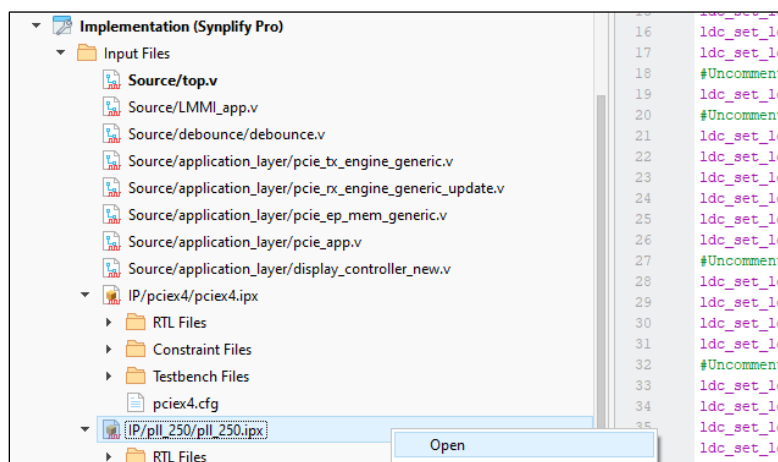


Figure 7.11. Open PLL IP

7. Change the Primary Clock Output value as following and click **Generate**.
 - 250MHz for Gen3
 - 125MHz for Gen2
 - 62.5MHz for Gen1

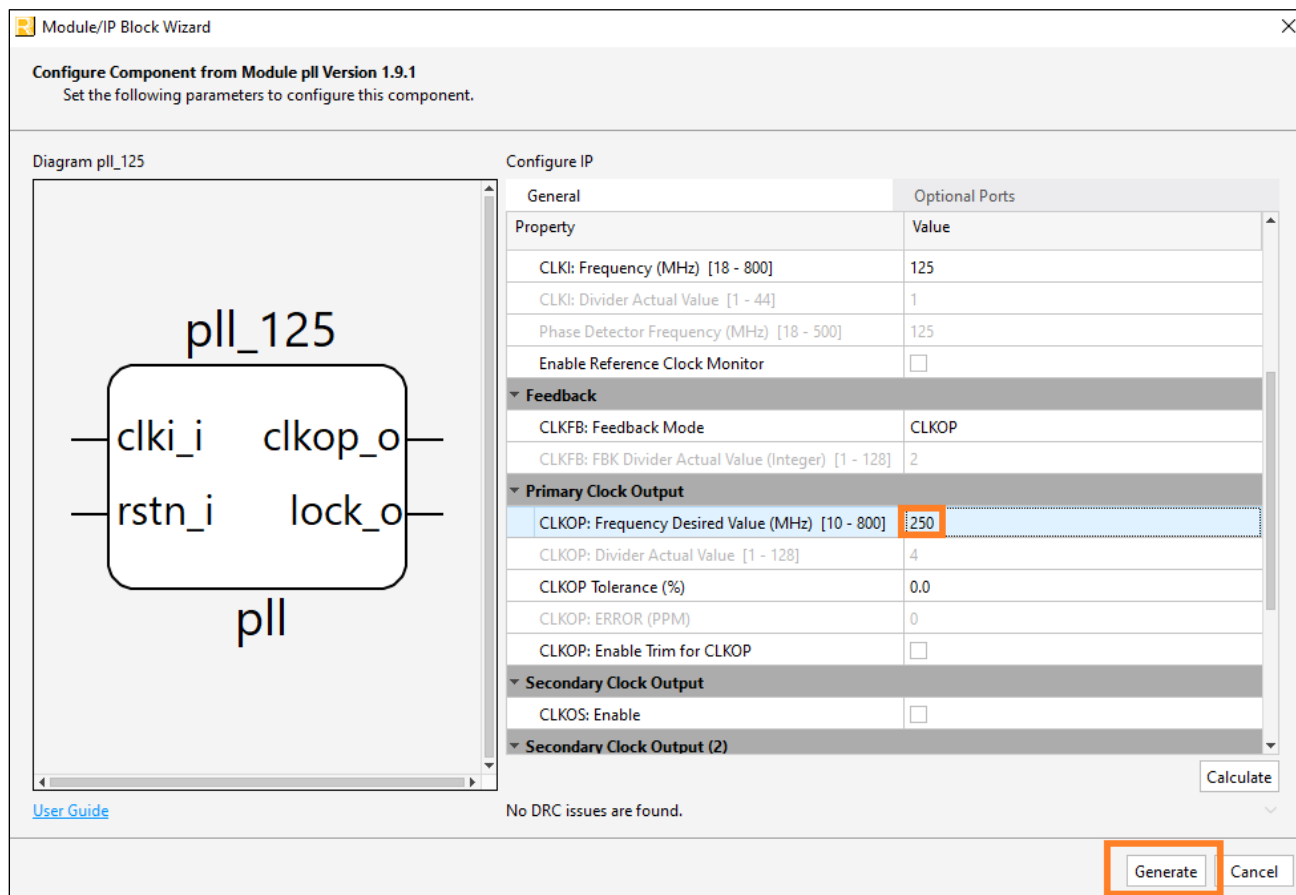


Figure 7.12. PLL IP Configuration

8. Save the design changes and recompile the design as shown in Figure 7.9. Use the configuration bit file located in the new design. The bit file in the original design is available in *Hardware\BasicDemo\Implementation\BasicDemo_Implementation.bit*.

8. Troubleshooting

8.1. SPI Flash Update

- If you are getting a verification error while programming the .bit file, try changing the TCK frequency to a value greater than 4. The TCK Divider Setting option is in the Cable Setup dialog box of the Radiant Programmer window, as shown in [Figure 8.1](#). Restart programming by clicking the Program button.

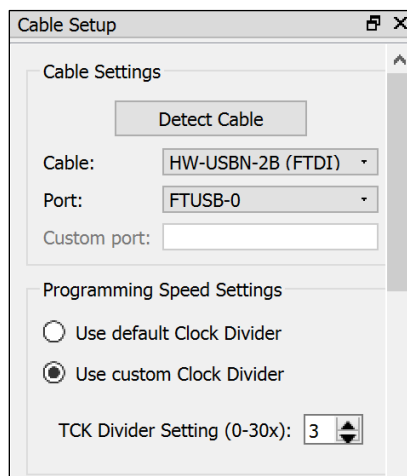


Figure 8.1. TCK Frequency Setting

- If the verification error problem persists, press and hold the PROGRAMN push button on the board before clicking the Program button.
- If the device is not getting detected on port FTUSB-0. Click Detect cable and select the port FTUSB-1.

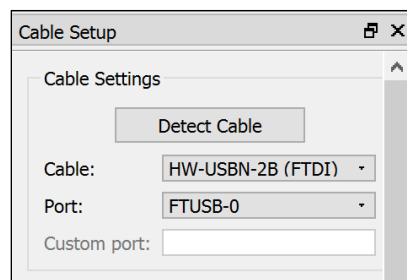


Figure 8.2. Port Selection

8.2. Installing the Driver and User Interface Launch for Windows

8.2.1. Problem with Installing the Driver

- If the hardware IDs are not found in Device Manager, ensure that the board is properly installed in a PCIe slot.
- Ensure that the board has a valid PCIe bitstream file loaded in the SPI Flash. Shut down the system and try another slot. If the board is present, check the Properties and the Resource tab to verify if memory is assigned to it. Also, verify if the Vendor ID and Device ID are valid, as seen by Windows Plug-n-Play. If the values are invalid, perhaps the bitstream file is corrupt and needs to be reloaded into SPI flash. If the PCIe board is shown in the list of Device Manager, install the driver manually.
- If the driver is not being installed even though the device is present in the Device Manager, make sure that the driver signature enforcement is disabled permanently by following the steps described in the [Disabling Driver Signature Enforcement Permanently](#) section.

8.2.2. Problem with Launching User Interface

The Device information is present in the Device Info tab if the drivers are installed properly. The *Driver Open [FAILED]* message as shown in [Figure 8.3](#) is displayed if the driver is not installed properly.

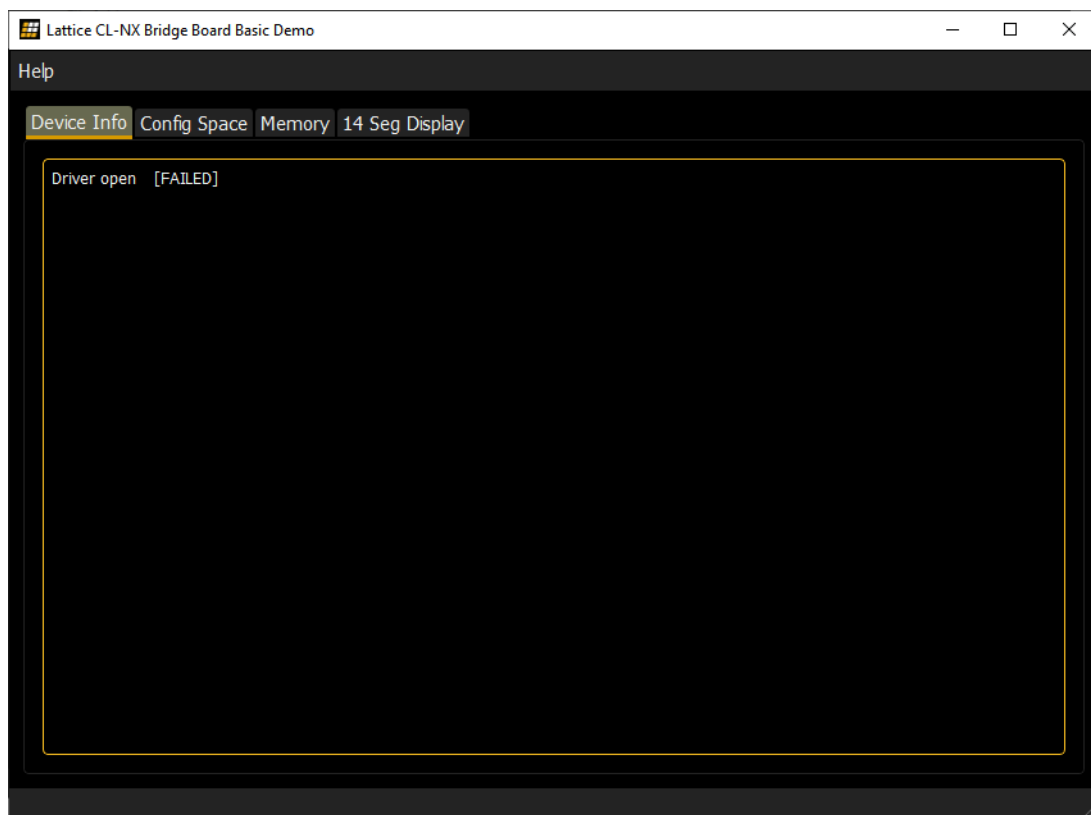


Figure 8.3. User Interface with No Device Driver

If the device is not inserted in the system or the device driver is not installed, any read or write operation is not executed. A sample error message is shown in [Figure 8.4](#).

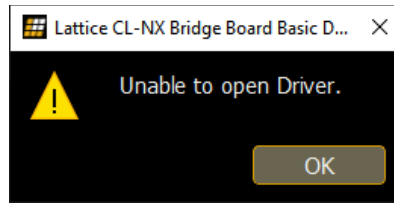


Figure 8.4. Error Message

If any such error message occurs, ensure that the device is properly inserted in the PCI Express slot. After that, verify that drivers are installed. It is also possible that the incorrect driver has been installed. In this case, uninstall the currently installed driver and manually reinstall the appropriate driver. When all the issues are resolved, restart the user application.

8.3. Installing the Driver User Interface Launch for Linux

8.3.1. Problem with Installing the Driver

The following error may occur during software driver installation.

Issue: insmod: ERROR: could not insert module *lattice_main.ko*: Operation not permitted.

Resolution: Ensure that the Secure Boot option is disabled in the BIOS of the system. Refer to the BIOS user manual to learn about the steps for turning off the Secure Boot option.

Issue: insmod: ERROR: could not insert module *drv_src/drvr/lattice_main.ko*: Invalid module format.

Resolution: Run the following commands sequentially in the terminal.

```
sudo apt update && sudo apt upgrade
sudo apt remove --purge linux-headers-*
sudo apt autoremove && sudo apt autoclean
sudo apt install linux-headers-generic
sudo apt-get install build-essential linux-headers
sudo apt-get install build-essential linux-headers-`uname -r`
```

8.3.2. Problem in Loading the Driver

After launching the user interface, if the driver is not loaded properly, a *Driver open[Failed]* message is displayed in the Device Info section.

- Make sure that the board is properly connected to the PC.
- Run the command below in a terminal window to show the list of PCIe devices connected to PC.

```
sudo lspci -vnm
```

Ensure that the Lattice device is present by checking the device and vendor IDs, as shown in [Figure 8.5](#) for CrossLink-NX, [Figure 8.6](#) for Certus-NX, and [Figure 8.7](#) for CertusPro-NX.

```
Device: 01:00.0
Class: 0880
Vendor: 1204
Device: 9c1d
SVendor:      19aa
SDevice:      e004
Rev: 10
```

Figure 8.5. lspci -vnm Output Image for CrossLink-NX

```
Device: 01:00.0
Class: 0880
Vendor: 1204
Device: 9c1d
SVendor:      19aa
SDevice:      e004
Rev: 10
```

Figure 8.6. lspci -vnm Output Image for Certus-NX

```
Device: 01:00.0
Class: 0880
Vendor: 1204
Device: 9c25
SVendor:      19aa
SDevice:      e004
Rev: 10
```

Figure 8.7. lspci -vnm Output Image for CertusPro-NX

- If the device is present, perform the [Manual Setup and Installation](#) steps one by one.
- If the driver does not build properly, check for any software or kernel dependencies.

8.3.3. Problem with User Interface Launching

- Go to the *Demonstration/Linux* directory. Check all content of this directory. Default content is shown in [Figure 8.8](#). Check the permissions of these files.

```
total 16
-rw-r--r-- 1 root root 175 Feb 16 18:27 install.sh
-rw-r--r-- 1 root root 237 Feb 16 18:27 Readme.txt
drwxr-xr-x 6 root root 4096 Feb 16 18:27 Source_Code
-rw-r--r-- 1 root root 47 Feb 16 18:27 uninstall.sh
```

Figure 8.8. Content List of Demonstration/Linux Directory.

- The script files should have *execute* permission – to set permissions on the file, run the command below in the terminal and try to run the application again.
`sudo chmod 777 filename`
- If the user interface still does not launch, change directory into the *Source_Code* directory and verify that all files are present in this directory, according to [Figure 8.9](#).

```
total 36
drwxr-xr-x 5 root root 4096 Feb 16 18:27 app_src
-rw-r--r-- 1 root root 27 Feb 16 18:27 compile.sh
drwxr-xr-x 5 root root 4096 Feb 16 18:27 drv_src
drwxr-xr-x 2 root root 4096 Feb 16 18:27 include
-rw-r--r-- 1 root root 137 Feb 16 18:27 install_drv.sh
-rw-r--r-- 1 root root 52 Feb 16 18:27 launch_gui.sh
drwxr-xr-x 2 root root 4096 Feb 16 18:27 lib
-rw-r--r-- 1 root root 566 Feb 16 18:27 Makefile
-rw-r--r-- 1 root root 156 Feb 16 18:27 Readme.txt
```

Figure 8.9. Content List of Software/Linux Directory

- If all files are present in this directory, change directory to *app_src/gui/deploy* and try to run the command below directly.

```
sudo ./PCIe_Test_App.sh
```

References

- [Certus-NX FPGA](#) web page
- [CertusPro-NX FPGA](#) web page
- [CrossLink-NX FPGA](#) web page
- [Lattice Radiant Software FPGA](#) web page
- [Lattice Insights](#) for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.2, August 2025

Section	Change Summary
All	- Made editorial fixes across the document, including section names in Troubleshooting. - Changed Windows 10 to <i>Windows Operating System</i> across the document.
Hardware and Software Requirements	Updated bullet items for PCIe Basic Demo, Windows/Ubuntu, and CertusPro-NX .bit in Software Requirements .
Importing and Building the FPGA Demonstration	Updated the directory for CrossLink-NX, Certus-NX, and CertusPro-NX for the .rdf file step in Building Lattice Radiant Project .
Setting Up the Demo	- Updated section content to change Windows10 to <i>Windows</i> in folder names. - Updated Programming the FPGA section to change CertusPro-NX bit to <i>BasicDemo_Implementation.bit</i>. - Updated Description and Release versions and removed OS Type and Codename bullet points in Supported Operating System and added figure captions for images in Required Packages.
Customizing the Design for CertusPro-NX	Added this section.
References	Added this section.

Revision 1.1, May 2023

Section	Change Summary
All	- Updated document title to PCIe Basic Demo for Lattice Nexus-based FPGAs. - Added support for Certus-NX and CertusPro-NX.
Introduction	- Added Certus-NX and CertusPro-NX information. - Modified list in Learning Objectives.
Hardware and Software Requirements	Added requirements for Certus-NX and CertusPro-NX.
Demo Design Overview	Moved this section. Modified contents to apply to supported devices. Updated Figure 3.1. Relationship of the Hardware and Software Components and Figure 3.3. PCIe Basic Demo FPGA Design.
Importing and Building the FPGA Demonstration	Moved this section and restructured the contents. Added the Demo Package Directory Structure as a new subsection.
Setting Up the Demo	In the Jumper Configuration section, added information for Certus-NX and CertusPro-NX, including the following tables and figures: - Table 5.2. Jumper Configuration for the Certus-NX Versa Evaluation Board - Table 5.3. Jumper Configuration for the CertusPro-NX Versa Evaluation Board - Figure 5.2. Certus-NX Versa Evaluation Board Jumper Location - Figure 5.3. CertusPro-NX Versa Evaluation Board Jumper Location - Figure 5.5. Certus-NX Versa Evaluation Board Connection - Figure 5.6. CertusPro-NX Versa Evaluation Board Connection In the Programming the FPGA section, added information for Certus-NX and CertusPro-NX, including the following figures: - Figure 5.11. CertusPro-NX FPGA Device Settings - Figure 5.13. Device Properties Window for Certus-NX SPI Flash Programming - Figure 5.14. Device Properties Window for CertusPro-NX SPI Flash Programming - Figure 5.18. Certus-NX Programming Done LED - Figure 5.19. CertusPro-NX Programming Done LED - Figure 5.17. CrossLink-NX Status LED was also updated.

Section	Change Summary
Application Overview	Added information for Certus-NX and CertusPro-NX, including the following figures: - Figure 6.4. Single 7-Segment Display on Board for Certus-NX - Figure 6.5. Interactive 7 Seg Display Tab for CrossLink-NX and CertusPro-NX - Figure 6.6. Interactive 7 Seg Display Tab for Certus-NX
Troubleshooting	- Updated the Driver Installation and User Interface Launch for Windows section. Modified the Problem with Launching User Interface subsection. - Updated the Driver Installation User Interface Launch for Linux section. Added the Problem in Driver Installation subsection and added contents for Certus-NX and CertusPro-NX. - Renamed Figure 7.5. lspci -vnm Output Image for CrossLink-NX and added Figure 7.6. lspci -vnm Output Image for Certus-NX and Figure 7.7. lspci -vnm Output Image for CertusPro-NX.
Technical Support Assistance	Added FAQ website link.
All	Minor adjustments in formatting and style.

Revision 1.0, November 2022

Section	Change Summary
All	Initial release.



www.latticesemi.com