Mixed Mode PCIe and TSEMAC for CertusPro-NX

Reference Design

FPGA-RD-02313-1.0

July 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 all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
AHB	Advanced High-Performance Bus
APB	Advanced Peripheral Bus
API	Application Programming Interface
CRC	Cyclic Redundancy Check
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
SGMII	Serial Gigabit Media Independent Interface
TSEMAC	Tri-Speed Ethernet Media Access Controller
USB	Universal Serial Bus

1. Introduction

This guide describes how to set up and run the Mixed Mode PCI Express (PCIe) basic demo and Tri-Speed Ethernet (TSE) in loopback mode using devices built on the CertusPro™-NX FPGA.

The demo is targeted for CertusPro-NX Versa Evaluation Board, which features the CertusPro-NX FPGA in the LFG672 package. The above-mentioned FPGAs are built on the Nexus™ FPGA platform using low-power 28 nm FD-SOI technology.

This guide describes the process of setting up and running a design that incorporates both PCI Express and TSE protocol and assumes that you do not have any associated tools installed on your system.

1.1. Quick Facts

Download the reference design files from the Lattice reference design web page: PCIe Basic Demo for Lattice Nexus FPGAs.

Table 1.1. Summary of the Reference Design

General	Target devices	LFCPNX-100
	Source code format	Verilog
Simulation	Functional simulation	Performed
	Timing simulation	Not performed
	Testbench	Available
	Testbench format	Verilog
Software Requirements	Software tool and version	Lattice Radiant™ software 2022 or later (CertusPro-NX)
	IP version (if applicable)	PCIe IP – PCIe x4 IP core v2.2 or later TSE MAC IP – TSE MAC IP core v1.6.0 or later
Hardware Requirements	Board	CertusPro-NX Versa Board
	Cable	• Mini-USB to USB-A cable for programming the bitstream • 12 V power adapter

1.2. Learning Objectives

After completing the steps in this guide, you will be able to perform the following:

- Set up and install all applicable development tools and PCIe demos.
- Establish communication between the FPGA and the system through the PCIe link.
- Run the mixed-mode demo that implements two separate PCIe functions on a single endpoint device and one TSE protocol interface. Each PCIe function allows you to control a different aspect of the FPGA board while the TSE protocol incorporates a random pattern generator and checker.
- Use what the demo teaches you about designing Lattice PCIe and TSE protocol solutions.
- 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 you have already installed the Lattice Radiant software. This document covers some of the basic functions of the Lattice Radiant software. If you would like to learn more about the Lattice Radiant software, refer to the Lattice Radiant software Help system.

2. Directory Structure and Files

The hardware folder (*./CertusPro_NX_Mixed_Mode*) inside the package contains the following subfolders:

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

3. Functional Description

3.1. Theory of Operation

The demo runs on a standard ×64 PC and accesses the FPGA board installed in a PCIe slot. Figure 3.1 shows the relationship between the hardware and software components of the demo. The PCIe IP present in the Lattice FPGA occupying the Quad 0 Channel 0 acts as a PCIe endpoint occupying certain ranges of PCI memory space. The TSE protocol occupies Quad 0 Channel 2 and currently consist of a random pattern generator and checker. The Quad 0 Channel 2 used in the CertusPro-NX Versa Evaluation Board is connected to the PCIe Goldfinger. Therefore, a serial loopback from TX to RX is used in this demo to showcase the data transmission for TSE protocol. When the PC boots, the BIOS and operating system (OS) probe the PCIe 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 memory space by the BIOS. After the device driver is installed, the application software can read/write from/to the PCIe device memory (Embedded block RAM (EBR)). Currently the design allows the GPIO function within the PCIe to trigger the start of random pattern generator and checker within the TSE protocol. Therefore, the application software can write to PCIe device memory, which triggers the start of the random pattern generator and checker within the TSE protocol. The application software can also read from the PCIe configuration space registers. In the Mixed Mode demo, two separate PCIe functions are implemented within a single endpoint.

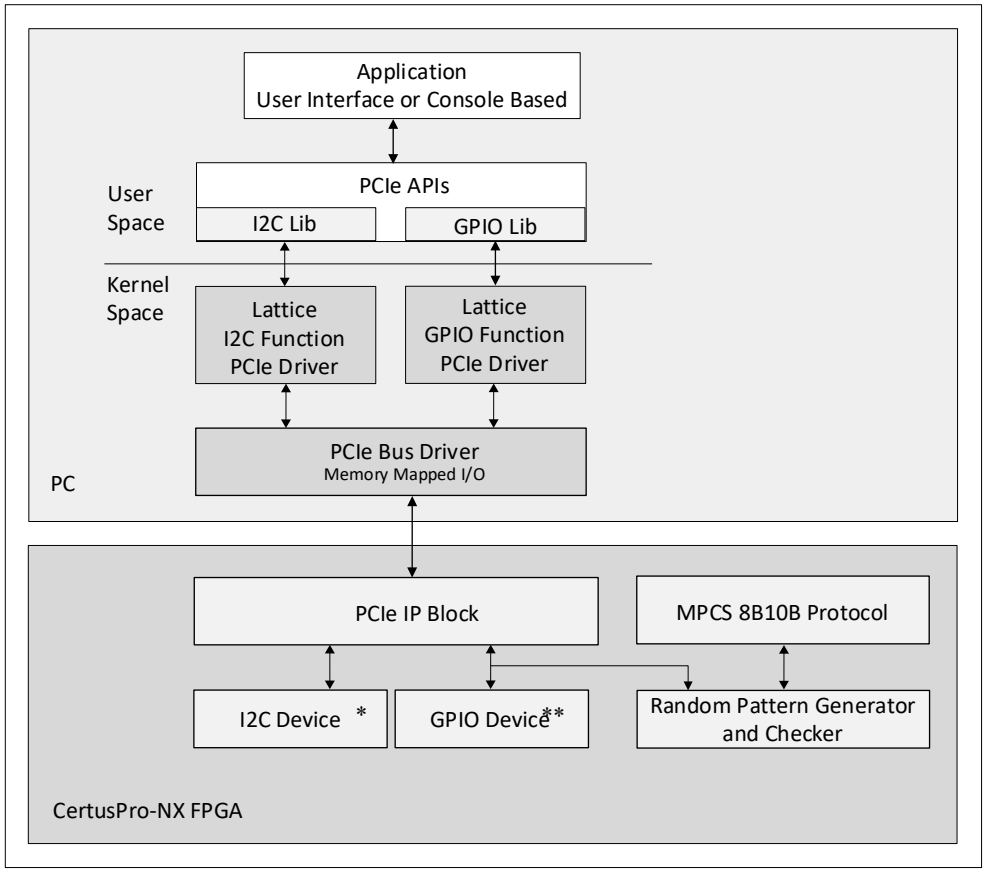

Figure 3.1. Relationship between Hardware and Software Components

3.2. Design Overview

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

- User interface application
- Driver API
- Device drivers
- Device hardware (FPGA design)

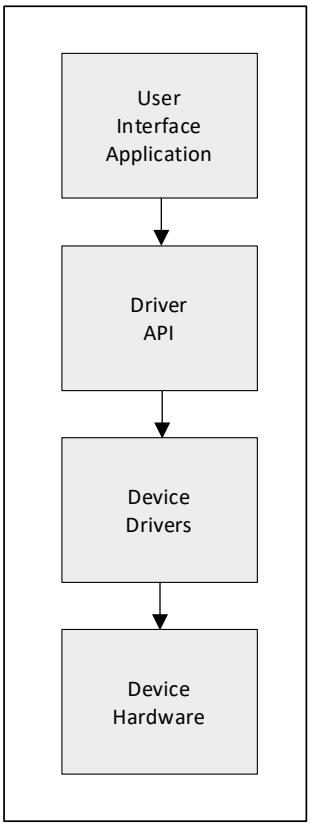

Figure 3.2. Mixed Mode Demo Software Design for PCIe

3.2.1. User Interface Application

The Mixed Mode demo includes a user interface written in Qt, and uses the driver application programming interface (API) to communicate with the device hardware to send control plane reads/writes to registers in the IP. The driver API is a C++ dynamic library that provides an interface to access the hardware.

On Windows, the *multifunction_lib.dll* library is a DLL that bridges the user space demo applications to the kernel space driver code and provides routines to control the IP modules.

On Linux, the library is named *libmem_rw.so*.

3.2.2. Device Drivers

The hardware drivers provide access to the FPGA board. On Windows, the *lpcie_gpio.sys* and *lpcie_i2c.sys*, and drivers support the Mixed Mode demo. On Linux, *gpio_main.ko* and *i2c_main.ko* support the demo.

3.2.3. Device Hardware (FPGA Design)

The following figure shows the top-level architecture of the FPGA design. PCIe hard IP is used on the FPGA side to implement the PCIe endpoint while MPCS is used to implement TSE protocol. The endpoint interfaces with the application logic for each function goes through an arbiter. The arbiter is also used to trigger the start of random pattern generator and checker used in the TSE protocol. For the initial configuration of the PCIe IP and MPCS IP, the LMMI interface is used. The following figure shows a block diagram of the FPGA design.

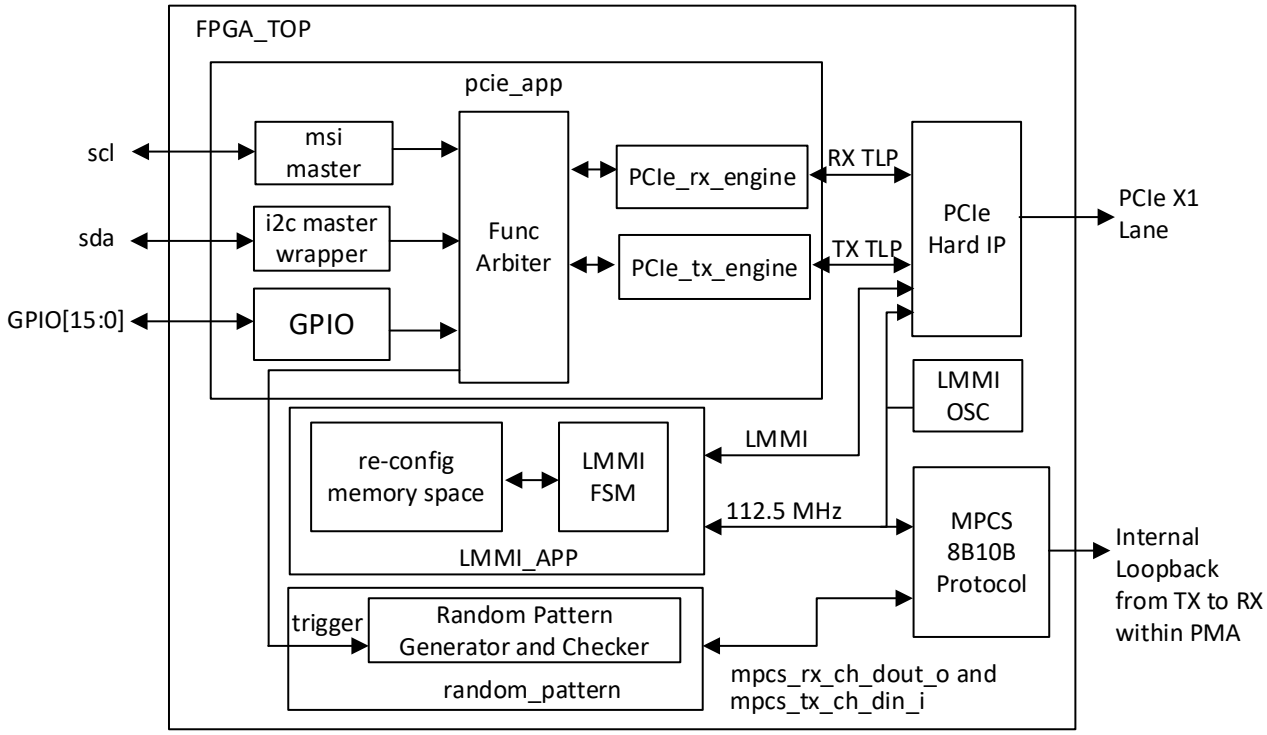

Figure 3.3. Mixed Mode Design

4. Running the Reference Design

4.1. Building the Lattice Radiant Project

To generate the bitstream file, follow these steps:

1. Open the Lattice Radiant software.
2. Click **Open project** and browse to the *.rdf* file. The file is LFCPNX_100_Mixed_Mode.rdf, which is in the *Hardware*\ *CertusPro_NX_Mixed_Mode* *LFCPNX_100_Mixed_Mode* folder.
3. After your 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 that you want to generate.

   Note: The options needed to regenerate the *.bit* file are selected by default.
5. After selecting your preferred reports, click **Run All**.

   This creates a *.bit* file with your project current name in the *impl_1* folder.

4.2. Hardware Configuration to Run the Reference Design

For CertusPro-NX devices, there are no jumpers to configure beyond the default. Refer to the CertusPro-NX Versa Board User Guide (FPGA-EB-02053) for a detailed list of jumpers on the board. 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 the following figure.

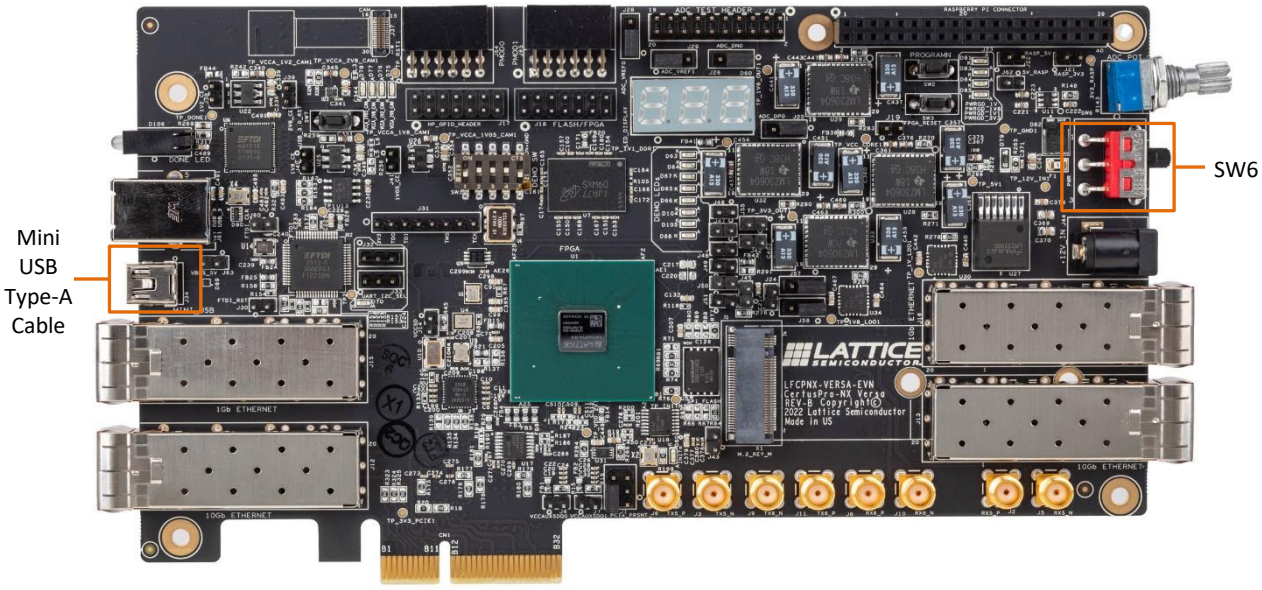

Figure 4.1. CertusPro-NX Versa Evaluation Board Connection

4.2.1. Programming the FPGA

To program the FPGA, follow these steps:

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 4.2.
2. Select **Create a new project from scan**. Values are indicated in the **Cable**, **Port**, and **TCK Divider Setting (0-30×)** fields.
3. Click **OK**.

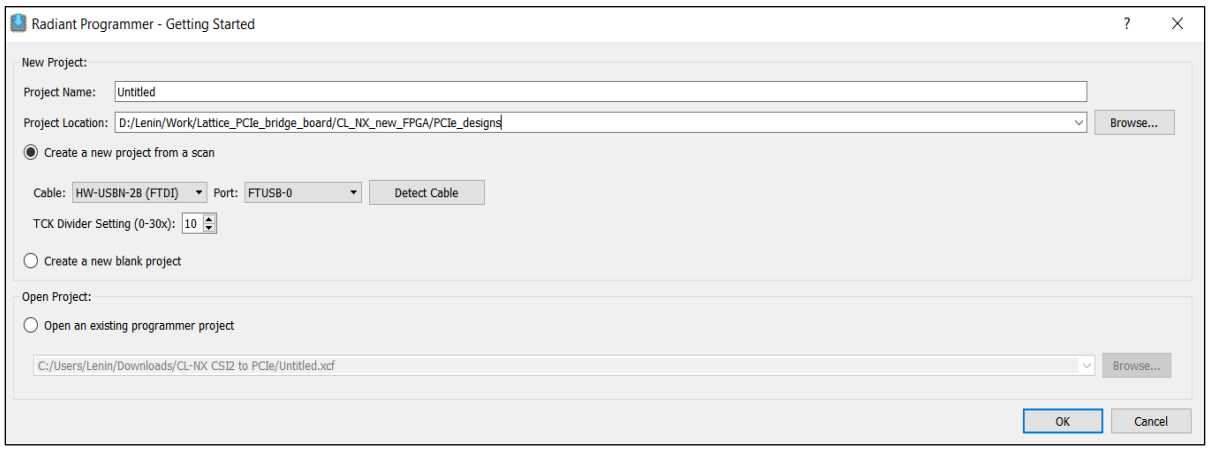

Figure 4.2. Creating a New Project from a Scan

4. The main interface opens as shown in the figure below.

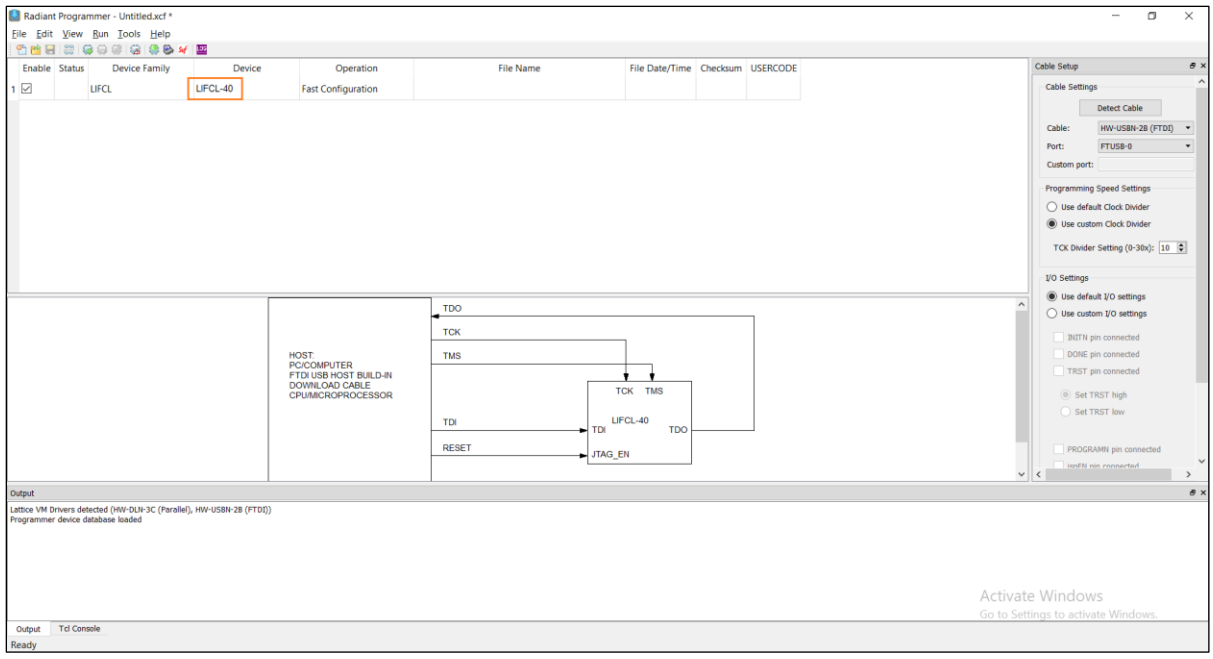

Figure 4.3. Lattice Radiant Programmer Window

5. If the Programmer settings do not match the settings shown in Figure 4.4 for CertusPro-NX devices, select these settings manually from the drop-down menu.

Figure 4.4. CertusPro-NX FPGA Device Settings

To select programming settings, follow these steps:

1. Browse and select the Programming file (*LFCPNX_100_Multi_Mode.bit*) from the *Demonstration\Bitstream* folder.
2. Click **OK**.
3. Double-click under **Operation** to open the **Device Properties** dialog box.
4. Select the settings as shown in the following figure for CertusPro-NX devices.

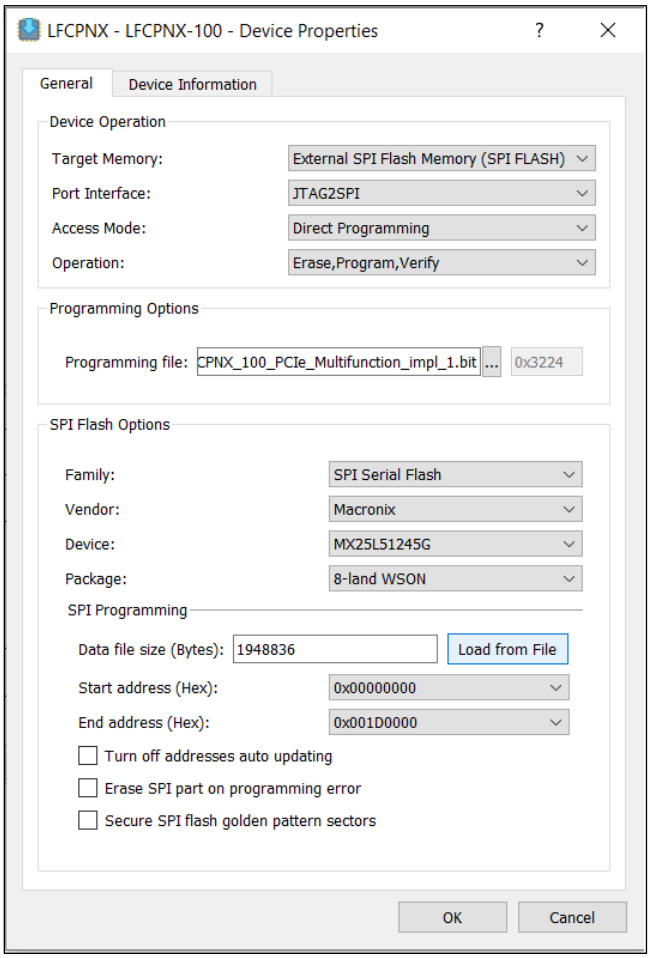

Figure 4.5. Device Properties Window for CertusPro-NX SPI Flash Programming (with Macronix Flash)

5. Click the **Programming** button from the menu bar shown in the following figure to start programming.

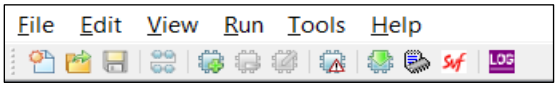

Figure 4.6. Programmer Menu Bar

When the FPGA programming is successful, the output console displays an **Operation: successful** message as shown in the following figure

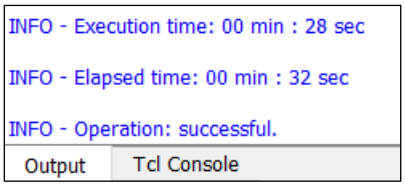

Figure 4.7. Programmer Output Window

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

4.2.2. Status LED

For CertusPro-NX devices, the programming Done LED lights up in green if configuration is successful. The LED is located as shown in the following figure

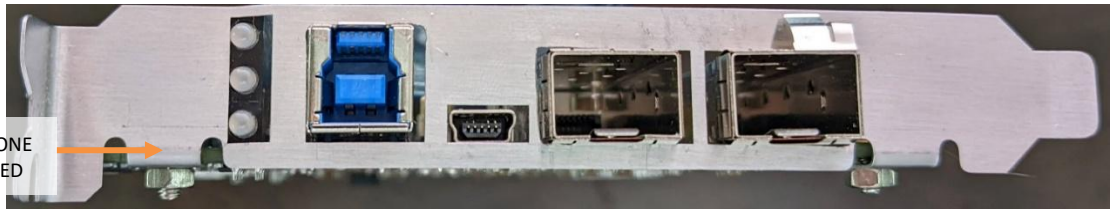

Figure 4.8. CertusPro-NX Programming Done LED

4.3. Software Setup

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

4.3.1. Software Setup and Installation for Windows

Before installing the driver, the Driver Signature Enforcement feature must be disabled.

4.3.1.1. Disabling Driver Signature Enforcement Permanently

To permanently disable Driver Signature Enforcement, follow these steps:

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

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

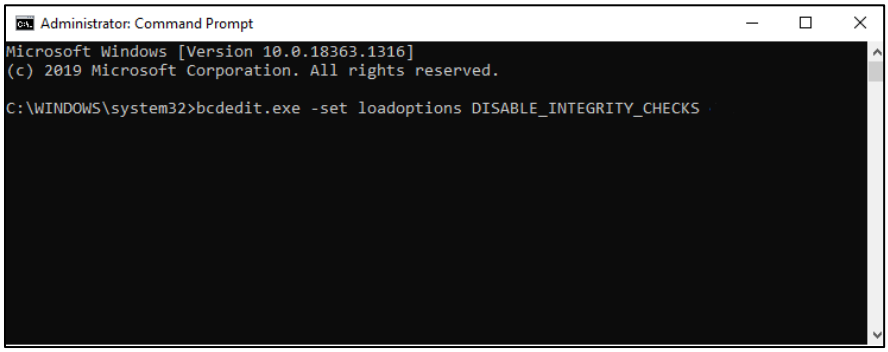

Figure 4.9. Running Disable Integrity Checks Command

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

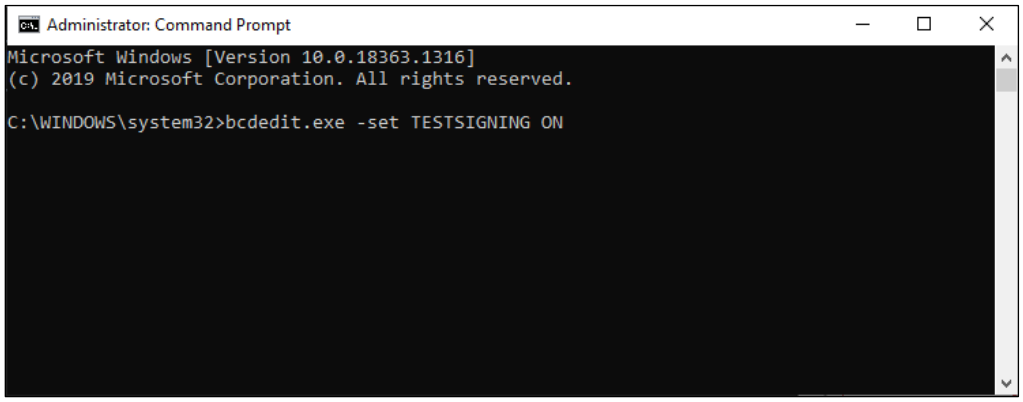

Figure 4.10. Running Test Sign On Command

3. Close the command prompt and restart your PC.

4.3.1.2. Disabling Driver Signature Enforcement Temporarily

Note: If Driver Signature Enforcement is already disabled, skip this section and proceed to the Driver Installation section.

To disable Driver Signature Enforcement temporarily on Windows 10, follow these steps:

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 the following figure, release the Shift key.
4. Click **Troubleshoot**.

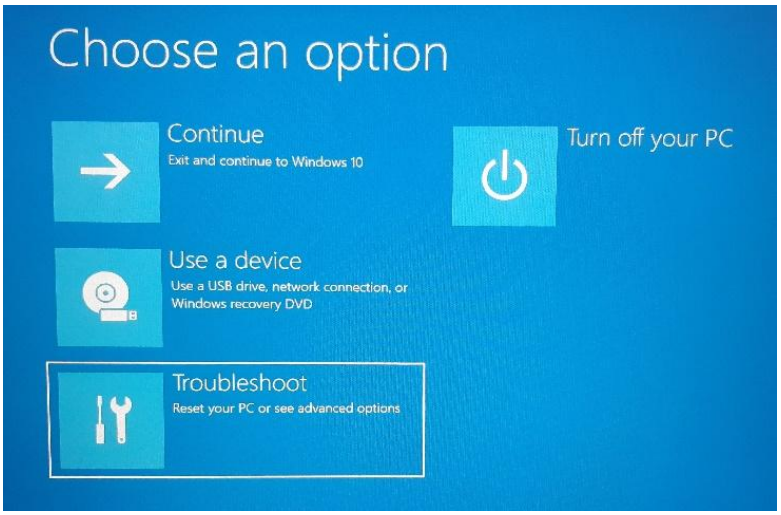

Figure 4.11. Troubleshoot Option

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

Figure 4.12. Advanced Options

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

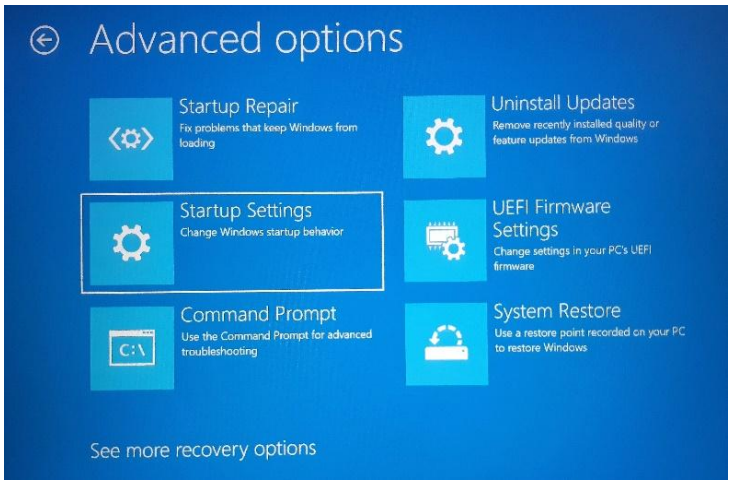

Figure 4.13. Select Startup Settings

7. Press **Enter** to restart.

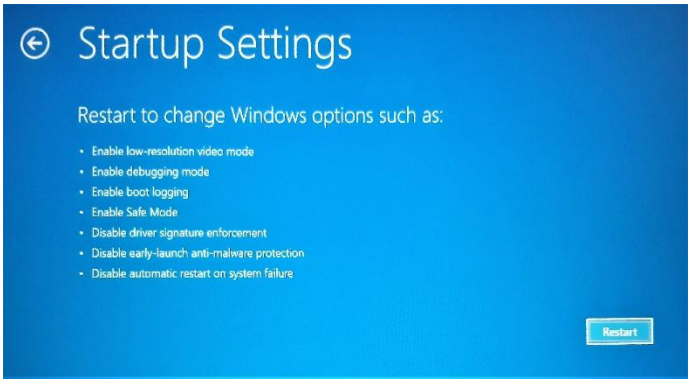

Figure 4.14. Restarting Windows

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

4.3.1.3. Driver Installation

There are two ways to install the device driver:

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

Installing Mixed Demo Device Driver through the User Interface Installer

The Multifunction Demo device driver can be 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 Multifunction Demo application in your system.

The framework supported version is Windows 10 with WDF 1.25 or earlier.

To install the Multifunction Demo device driver through the user interface, follow these steps:

1. In the *Demonstration\Windows10\Application* folder, double click *setup.exe*.
2. The welcome page appears. Click **Next**.

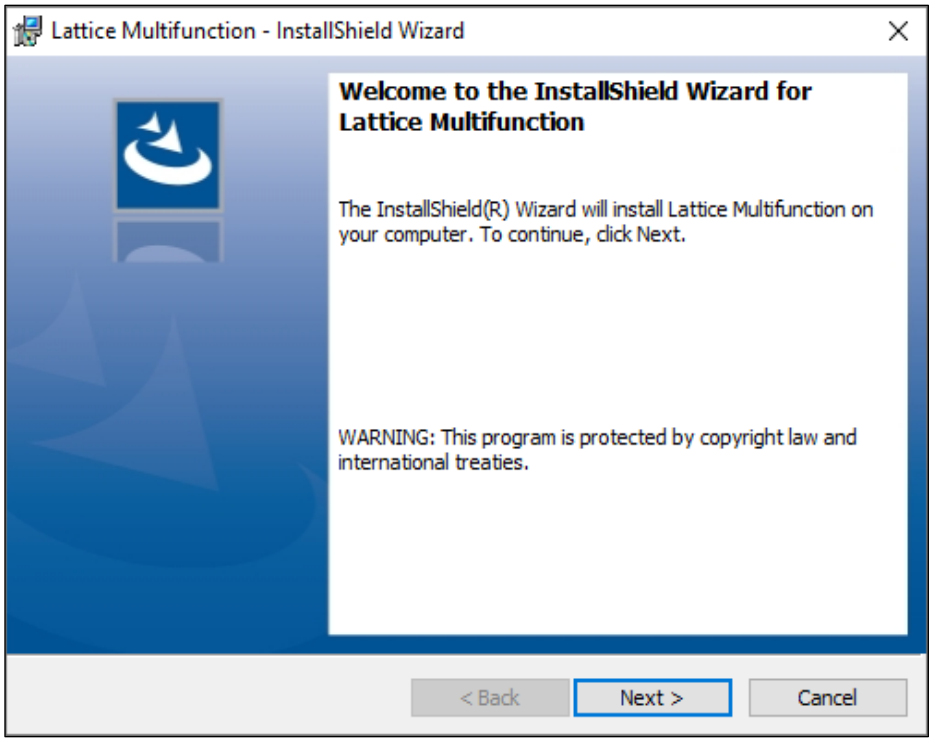

Figure 4.15. Windows Installer: Welcome Page

3. Provide the location where you want to install the application. Click **Next**.

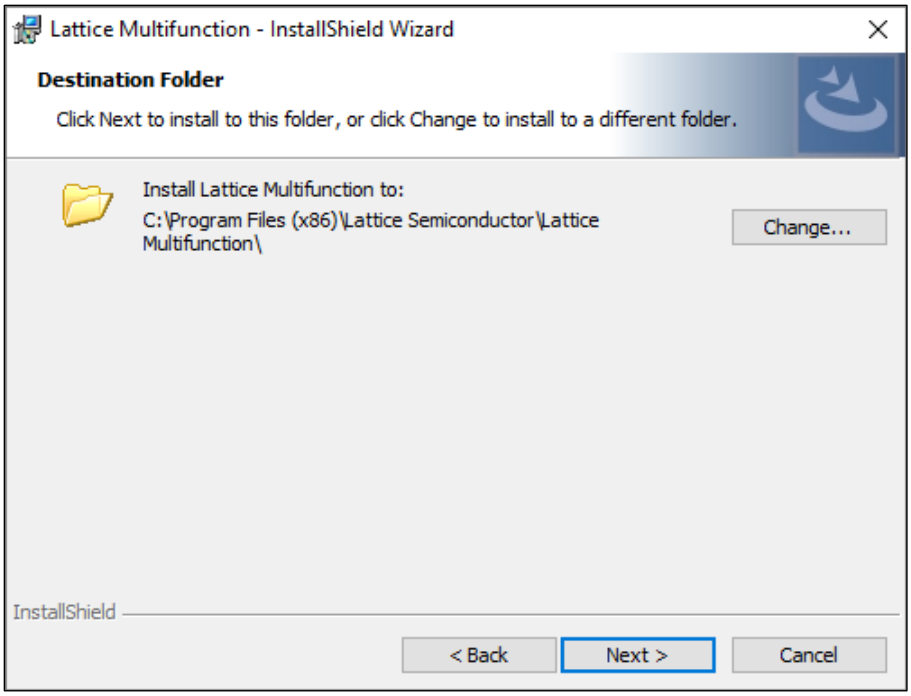

Figure 4.16. Windows Installer: Destination Folder Page

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

   Note: Administrative access is required to run this command.

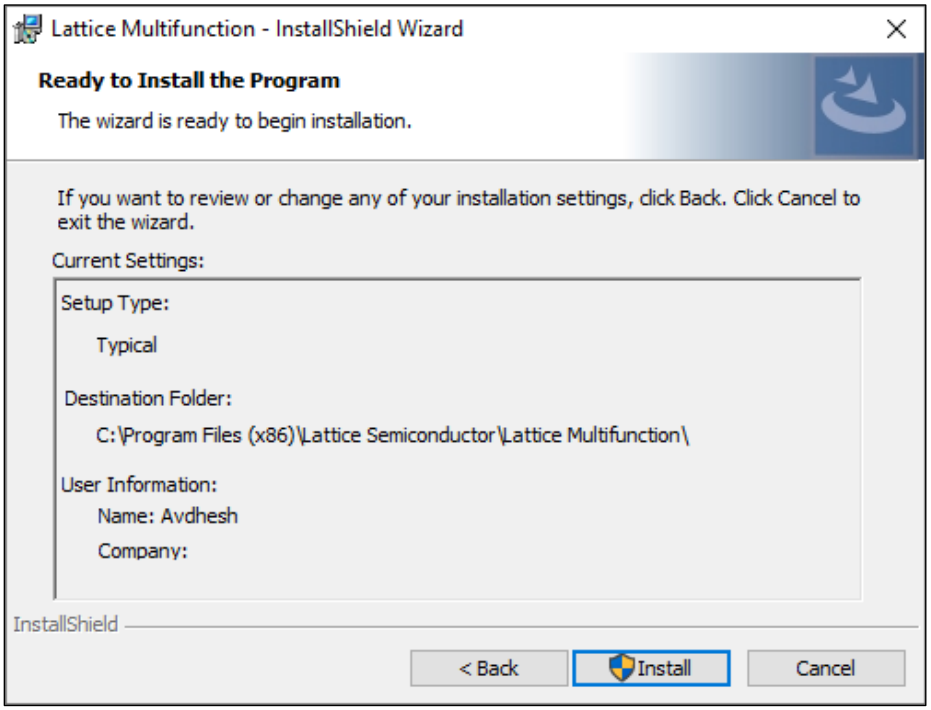

Figure 4.17. Windows Installer: Summary Page

5. The installation of the Multifunction Demo application starts. When the software installation is completed, the drivers are installed.

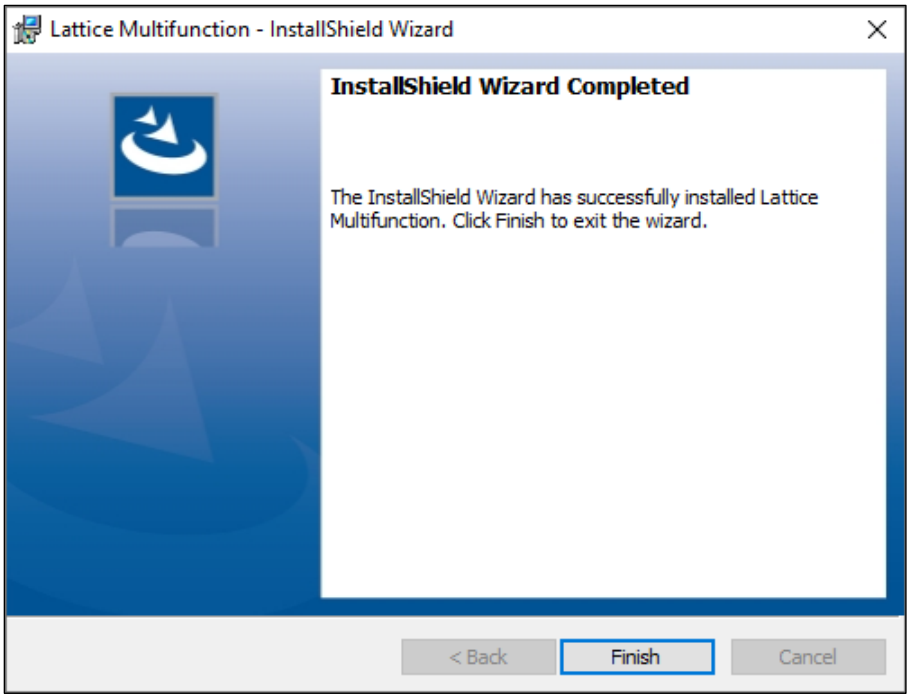

Figure 4.18. Windows Installer: Application Installed

6. A message box appears. Click **Yes**.

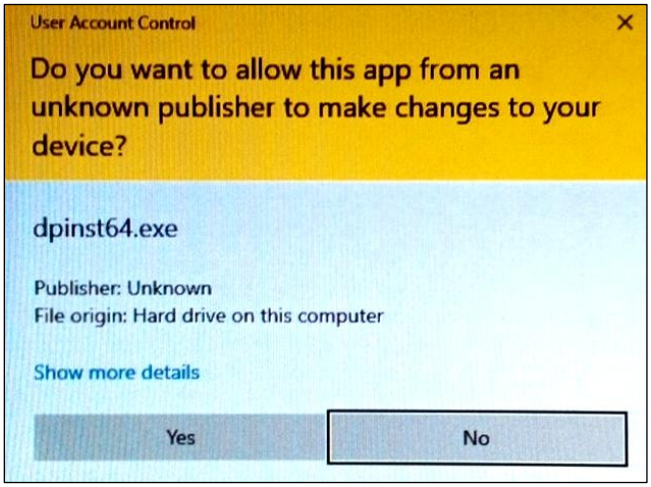

Figure 4.19. Device Configuration Prompt

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

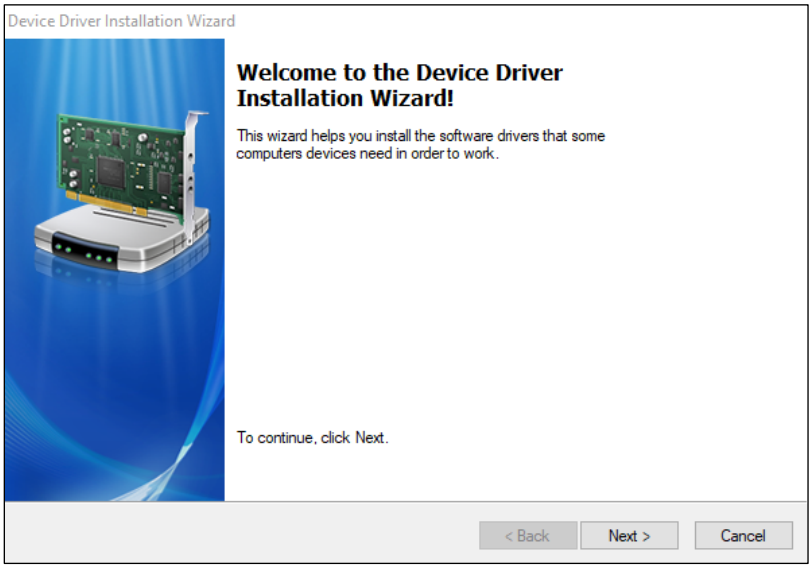

Figure 4.20. Device Driver Installation Wizard

8. If you receive a Windows Security prompt, select **Install this driver software anyway** as shown in the following figure.

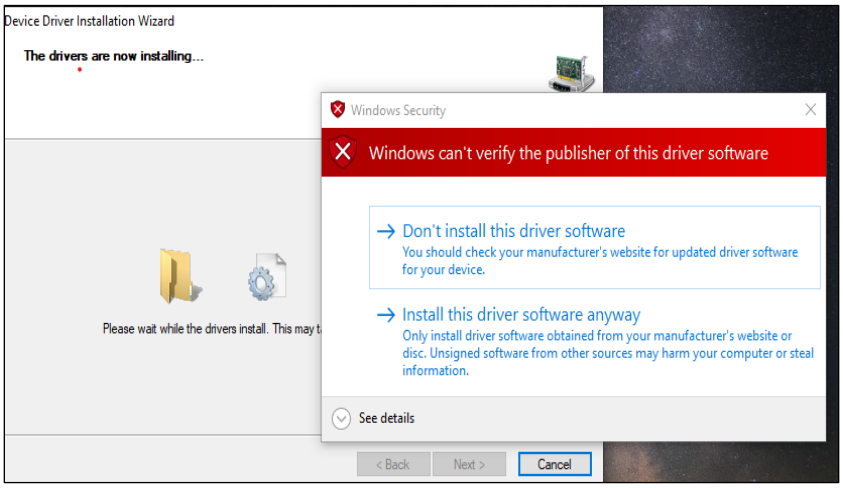

Figure 4.21. Windows Security in Driver Installation

9. If the driver is installed successfully, a message is displayed as shown in the following figure.

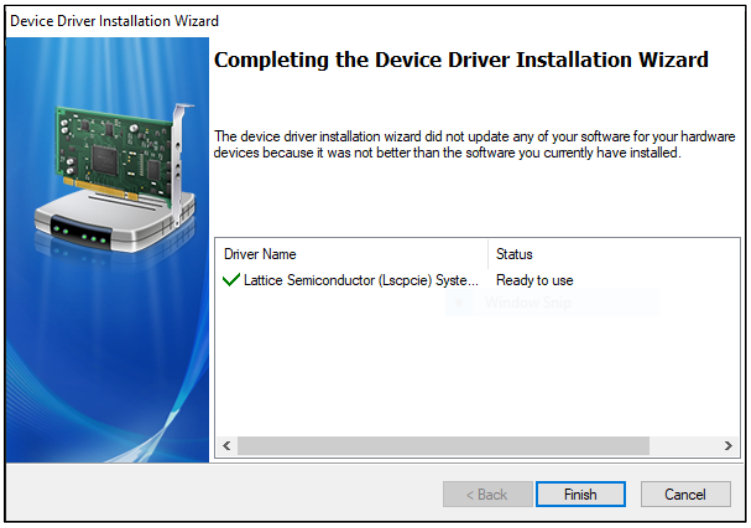

Figure 4.22. Device Driver Installation Completed

Installing Multifunction Demo Device Driver Manually

The drivers for the Multifunction Demo can be found in the *Demonstration\Windows10\Driver* folder.

To install the Multifunction Demo device driver manually, follow these steps:

1. Open **Device Manager**, which shows the connected PCIe devices, as shown in Figure 4.23.

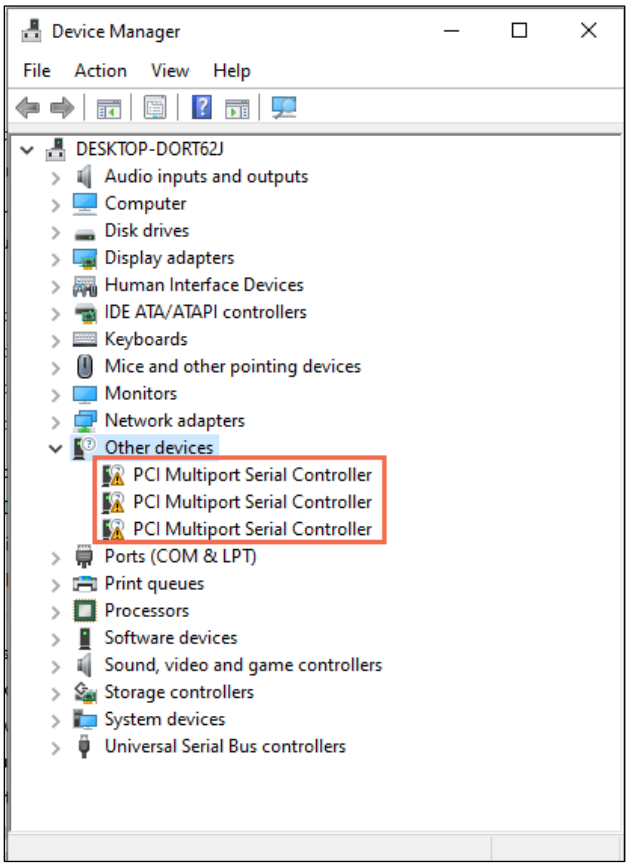

Figure 4.23. Device Manager

2. Right-click each device and select **Properties** as shown in the figure below.

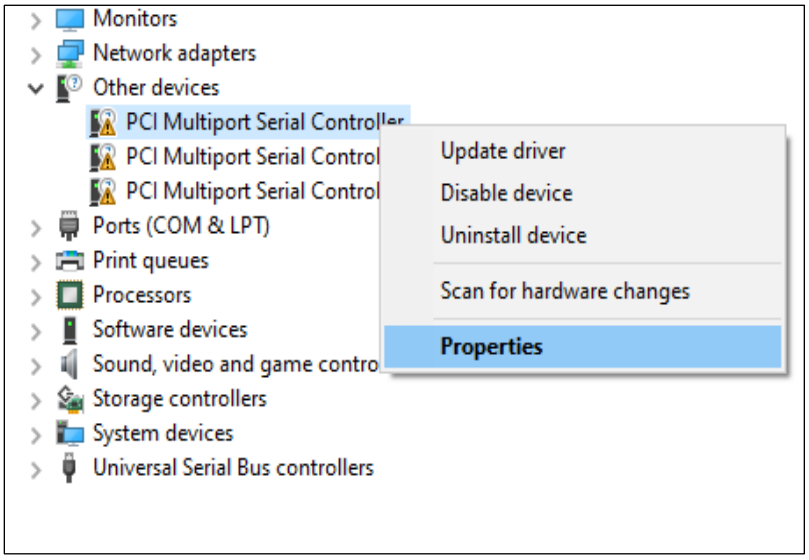

Figure 4.24. Showing Device Properties

3. The information in **Hardware IDs** is needed to install the correct driver for the corresponding device. The Hardware IDs of the I2C and GPIO devices are shown below:

Table 4.1. Device IDs

Sl. No	Certus-NX
I2C	PCI\VEN_1204&DEV_9C2F&SUBSYS_E00419AA
GPIO	PCI\VEN_1204&DEV_9C30&SUBSYS_E00419AA

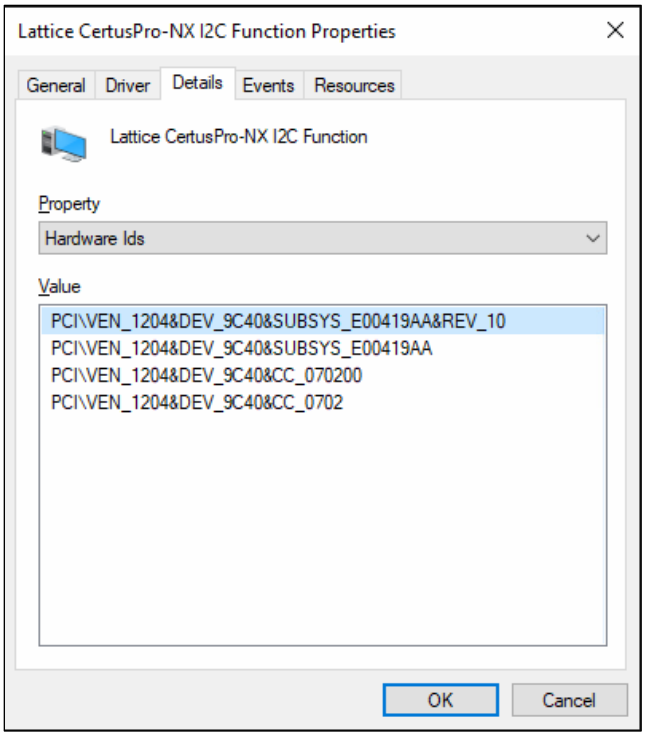

Figure 4.25. Hardware IDs of CertusPro-NX I2C Device

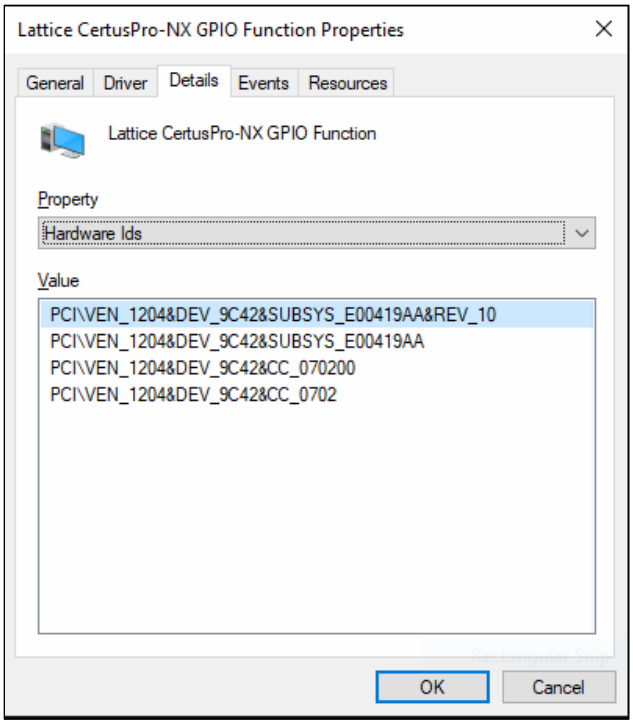

Figure 4.26. Hardware IDs of CertusPro-NX GPIO Device

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

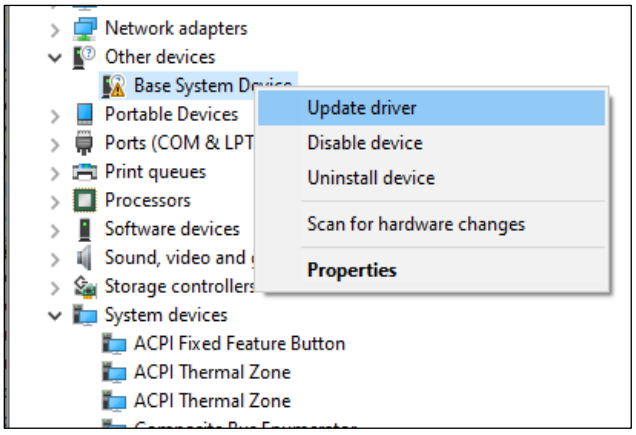

Figure 4.27. Update Driver Menu in Device Manager

5. Select the **Browse my computer for driver software** option as shown in the following figure.

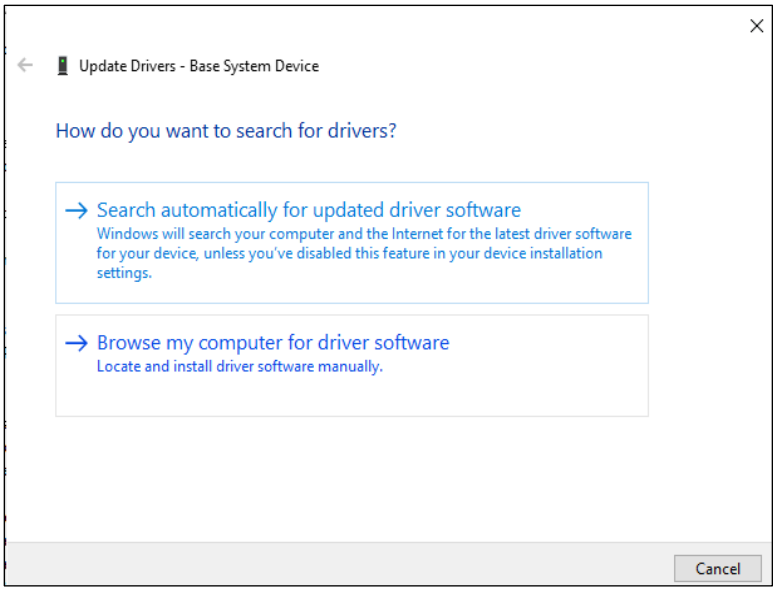

Figure 4.28. Update Driver Options

6. Browse for I2C or GPIO driver.

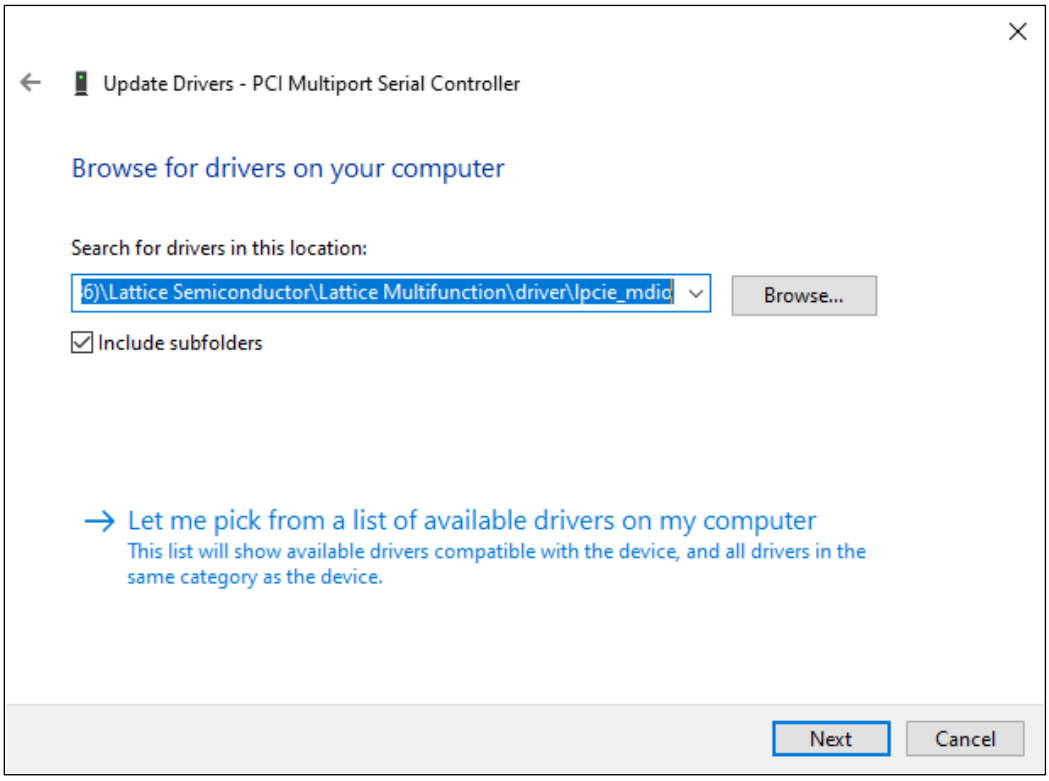

Figure 4.29. Browse the Driver for Device

7. If you receive a Windows Security prompt, select **Install this driver software anyway** as shown in the following figure.

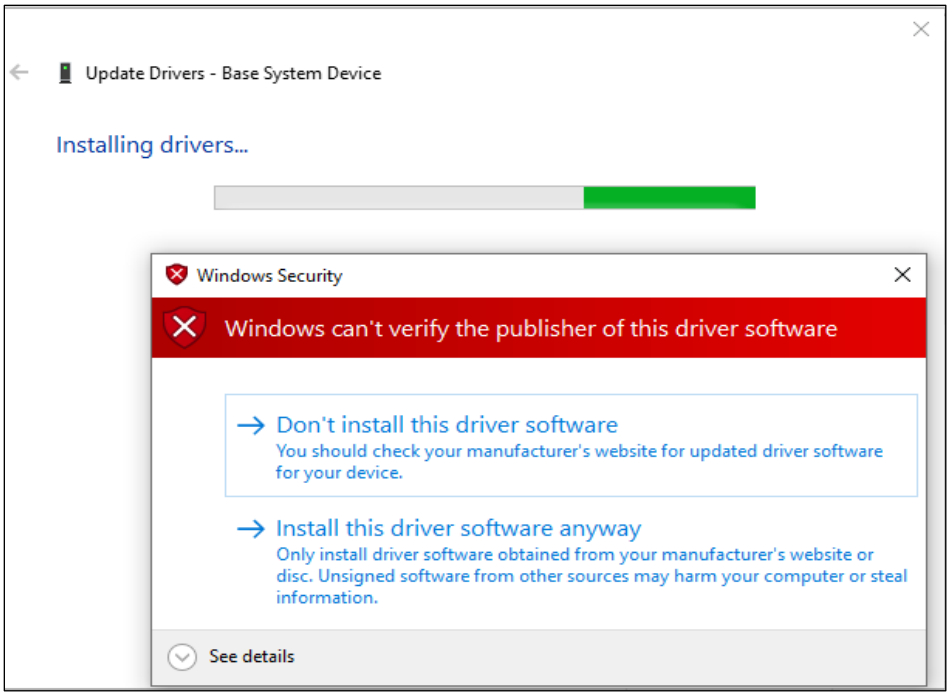

Figure 4.30. Windows Security in Device Manager

8. If the driver is installed successfully, a message is displayed as shown in the following figure.

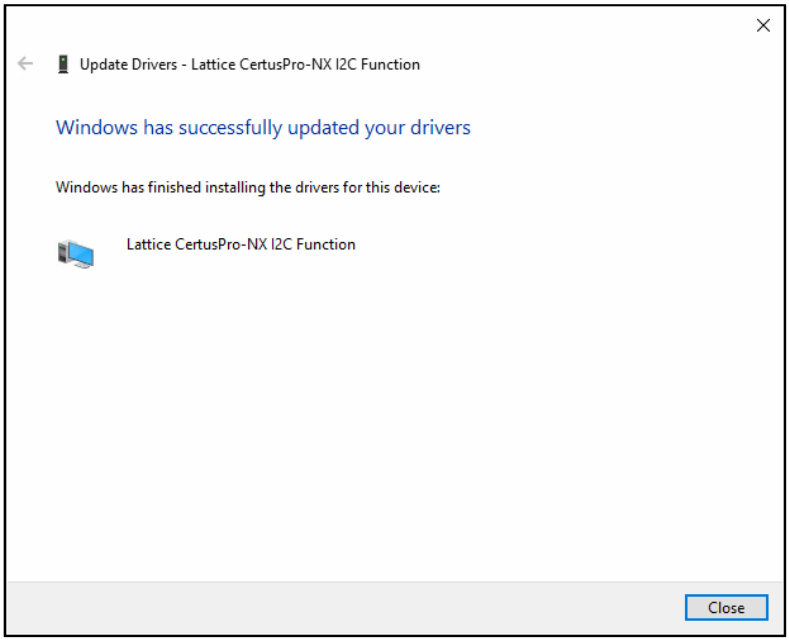

Figure 4.31. I2C Driver Installation Status Message for CertusPro-NX

9. Follow steps 3 through 8 for GPIO devices. After successful installation, all the two devices are displayed in the Device Manager as shown in the following figure.

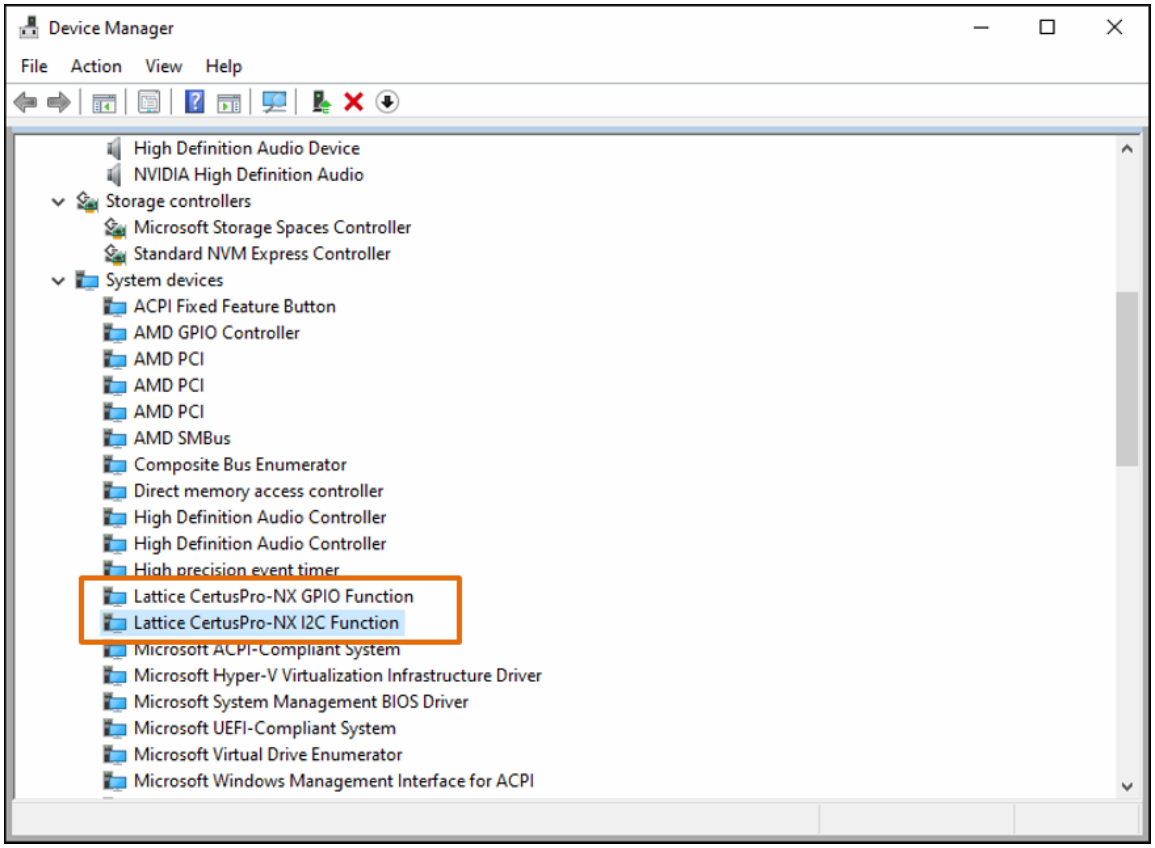

Figure 4.32. I2C and GPIO Device Drivers for CertusPro-NX in Device Manager

4.3.1.4. Uninstall the Driver Manually

Follow these steps to uninstall the device driver:

1. Open **Device Manager** window. Look for the device you want to uninstall.
2. Right-click on the device driver and select **Uninstall device** as shown in the following figure.

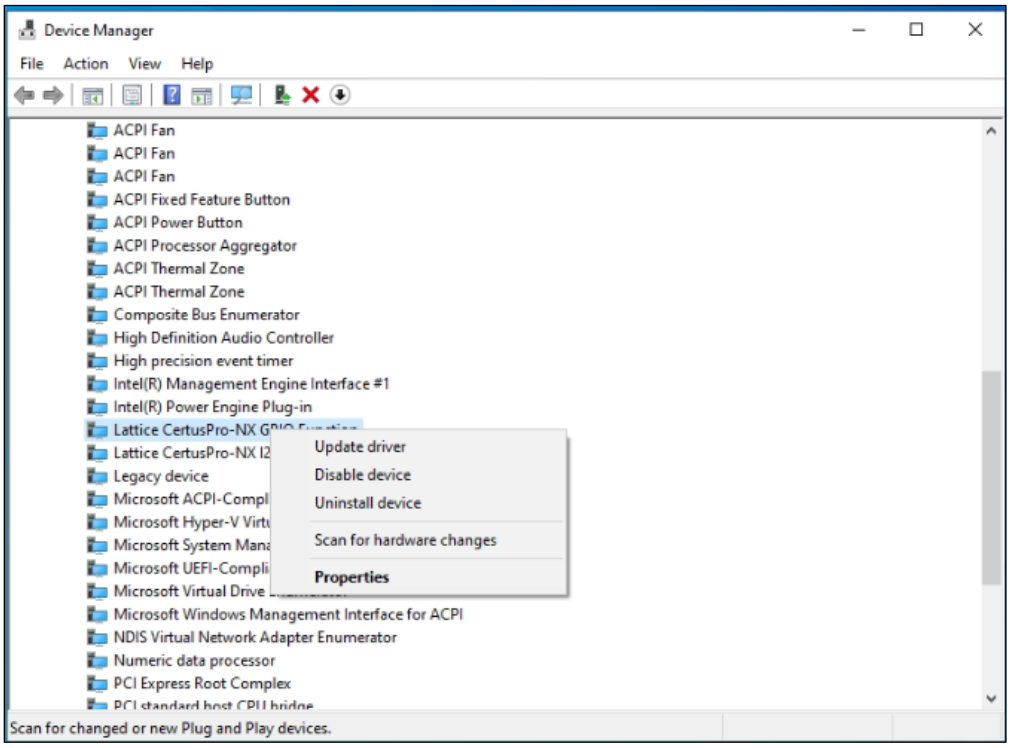

Figure 4.33. Select Uninstall Device

3. Checked the option **Delete the driver software for this device** and click on **Uninstall** as shown in the following figure.

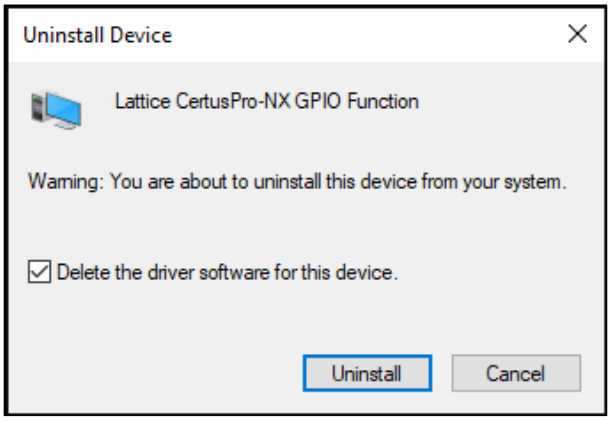

Figure 4.34. Uninstall and Delete Driver Software for this Device

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

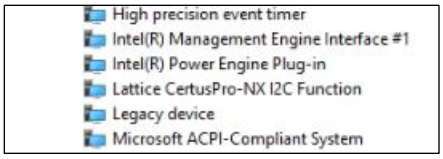

Figure 4.35. Uninstalled Device is Not Listed in the Device Manager

4.3.2. Software Setup for Linux

4.3.2.1. Supported Operating System

- Distribution: Ubuntu
- Description: Ubuntu 18.04.6 LTS
- Release: 18.04.6
- OS Type: 64 bit
- Codename: bionic

4.3.2.2. Required Packages

To check whether the required packages are installed, run the following commands on a terminal as shown below.

```
make -v
```

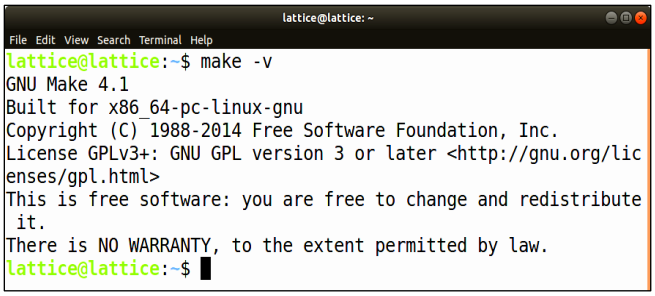

Figure 4.36. make-v Command

```
gcc -v
```

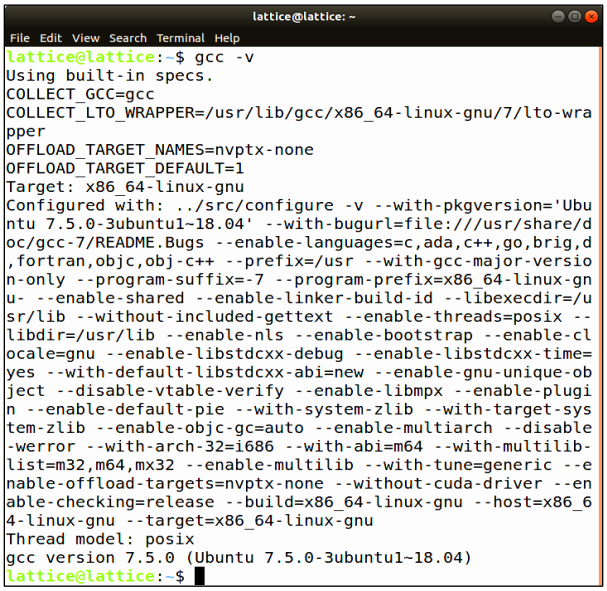

Figure 4.37. gcc -v Command

```
g++ -v
```

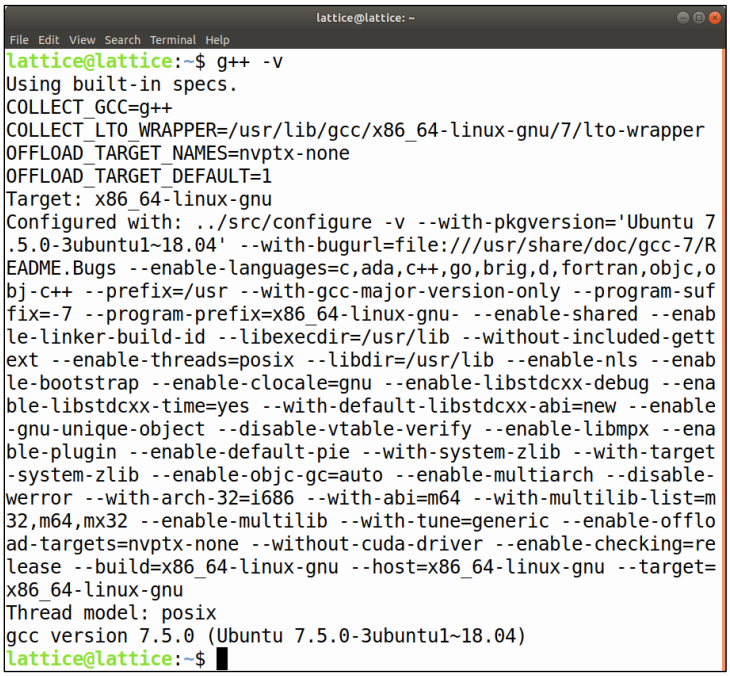

Figure 4.38. g++ -v Command

4.3.2.3. 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
```

4.3.2.4. Automatic Setup and Installation

To set up the demo in automatic mode, follow these steps:

1. Go to the *Demonstration/Linux* directory.
2. Change the permission of the *install.sh* file by running the chmod command.
   ```
   sudo chmod 777 install.sh
   ```
3. Run the following command, which builds the driver and API library, install the driver, and launch the QT use interface application.
   ```
   sudo ./install.sh
   ```
4. To uninstall the driver, go to the *Demonstration/Linux* directory and run the following command.
   ```
   sudo ./uninstall.sh
   ```

4.3.2.5. Manual Setup and Installation

Note: You may skip this section if you have installed the driver successfully in the previous section.

Before installing the driver, the driver must be built.

To build the driver, follow these steps:

1. Go to the *Demonstration/Linux* directory.
2. Run the following command on the terminal.
   ```
   sudo chmod 777 -R Source_Code
   ```
3. Go to the *Source_Code* directory.
4. Run the following commands.
   ```
   sudo make clean
   sudo make
   ```
   This builds the driver and API library.
5. Make sure that the driver is not installed.
6. To remove an existing driver, run one or all the following command in a terminal window:
   ```
   sudo rmmod gpio_main.ko
   sudo rmmod i2c_main.ko
   sudo rmmod mdio_main.ko
   ```

To install the drivers, follow these steps:

1. Go to the *drv_src/gpio_drv/* directory.
2. Run the following command.
   ```
   sudo insmod gpio_main.ko
   ```
   Repeat the same procedure for the i2c_drv and mdio_drv directories. The mdio_drv directory is not applicable for CertusPro-NX devices.
3. To launch the user interface application, go to the *app_src/gui/deploy/* directory and run the following command.
   ```
   sudo ./MFDemo.sh
   ```

4.4. Using the Mixed Mode Demo Application User Interface

The PCI Express Device Info page displays information about the device driver and device PCI configuration registers. The data displayed is read-only and cannot be edited. If there is a problem loading the driver or accessing the device, an error message appears on this page. The **Run Counter** button in the **Functionality Test** tab is used to trigger off the start of random pattern generator and checker that is used in the TSE protocol.

4.4.1. Functionality Test Tab

You can read or write from or to the device using controls present on this tab. The following figure shows the **Functionality** Test tab.

Figure 4.39. Functionality Test Tab

4.4.2. GPIO

This section is used to interact with the GPIO endpoint function.

4.4.2.1. Input Switch

There are five input LEDs in the GPIO input box. These LEDs change the status when you change the position of the input switches on the board. The input LEDs are mapped to SW1 on the board, and there is one LED per input switch. To start reading input from the switches, click the **Read Input** button. An example of the LEDs is shown in the following figure.

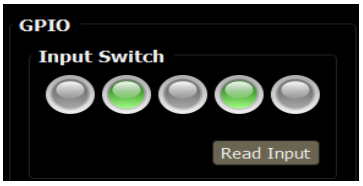

Figure 4.40. GPIO Input Switch

4.4.2.2. Output LED

The output LED section shows the status of eight output LEDs. These LEDs represent the five bits of a counter, which begin to increase when you click the **Run Counter** button on the user interface, as shown in the following figure.

Figure 4.41. GPIO Output LEDs

4.4.2.3. Run Counter

The **Run Counter** switches the Demo LED within the CertusPro-NX Versa Evaluation Board to be turned ON and OFF following the GPIO Output LED as shown in the following figure.

Seven out of eight (LED_0 till LED_6) of the Demo LED follows the GPIO Output LED.

Figure 4.42. Demo LEDs

The **Run Counter** trigger starts the Random Pattern Generator and Checker used in the TSE protocol. The data transmitted through the TSE protocol is looped back from TX to RX at the PMA using the serial loopback feature.

The Random Pattern Generator and Checker signals is shown in the Reveal Analyzer.

4.4.2.4. Reveal Analyzer

You can open the Reveal Analyzer using the Radiant software version 2022 onwards by clicking on the highlighted button shown in the following figure.

Figure 4.43. Open Reveal Analyzer

4.4.2.5. Configure the Reveal Analyzer

The Reveal Analyzer can be configured as shown in the following figure. The Reveal Analyzer source file is mixed_mode.rvl, which is in the *Hardware*\ *CertusPro_NX_Mixed_Mode* *LFCPNX_100_Mixed_Mode* folder. Select **Scan** and **OK** to configure the Reveal Analyzer.

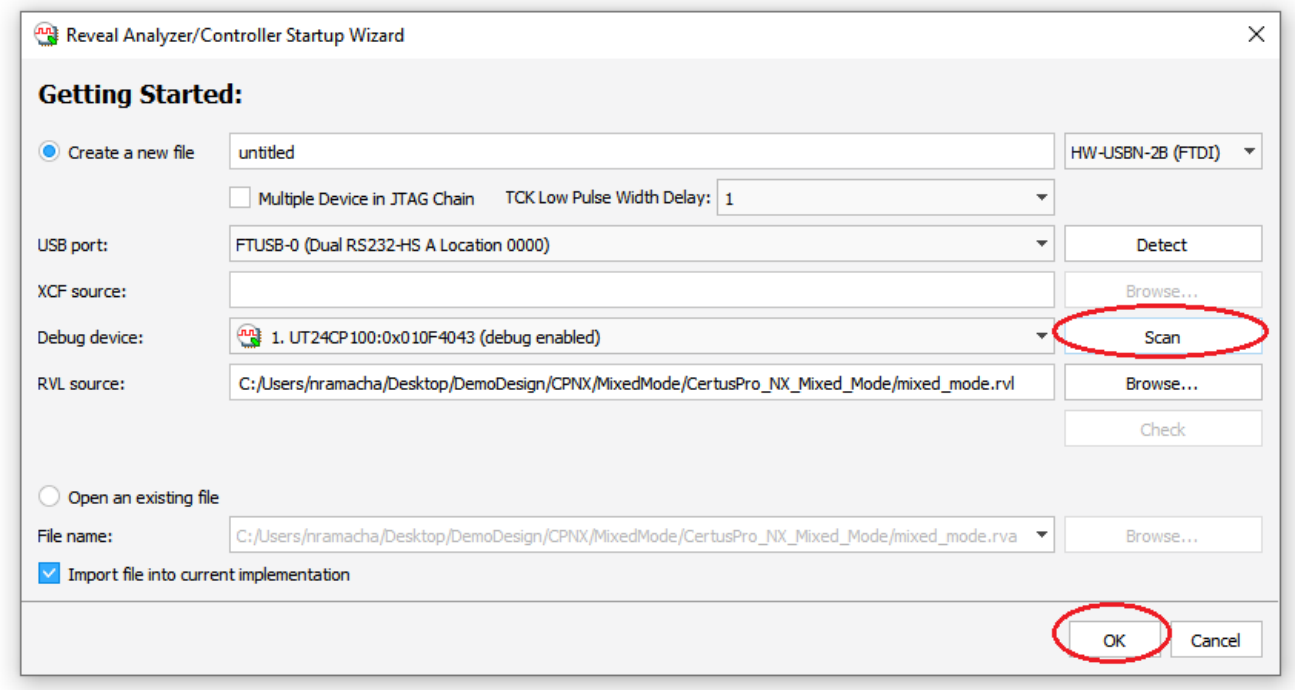

Figure 4.44. Configure Reveal Analyzer

4.4.2.6. Reveal Analyzer Capture

The Run Counter triggers and start the Random Pattern Generator and Checker. The Reveal Analyzer in Figure 4.45 shows the behavior before the Random Pattern Generator and Checker starts. Before the Random Pattern Generator starts, the Mixed Mode demo designs transmits word alignment pattern *BC* through the TSE protocol to the TX. The Mixed Mode demo design uses serial loopback from TX to RX, hence the RX will receive BC on the first byte and attempt to determine the word boundary. Word boundary determined correctly by the RX is indicated by assertion of signal *mpcs_get_lsync* as shown in Figure 4.45. After the word boundary is determined, the Mixed Mode demo design \ checks whether the random Run Counter has triggered a start. When the Run Counter is triggered to start, the Random Pattern Generator will start transmitting random patterns, this can be observed through *mpcs_tx_ch_din_byte0, mpcs_tx_ch_din_byte1, mpcs_tx_ch_din_byte2* and *mpcs_tx_ch_din_byte3* signals. The signal received at the RX is sent over to *mpcs_rx_ch_dout_byte0, mpcs_rx_ch_dout_byte1, mpcs_rx_ch_dout_byte2* and *mpcs_rx_ch_dout_byte3* signals which sends the signal to Random Pattern Checker. If the Random Pattern Checker receives 32 bytes of data correctly, the *random_pattern_detected* signal asserts indicating that data has been transmitted and received with no bit errors using the TSE protocol as shown in Figure 4.46.

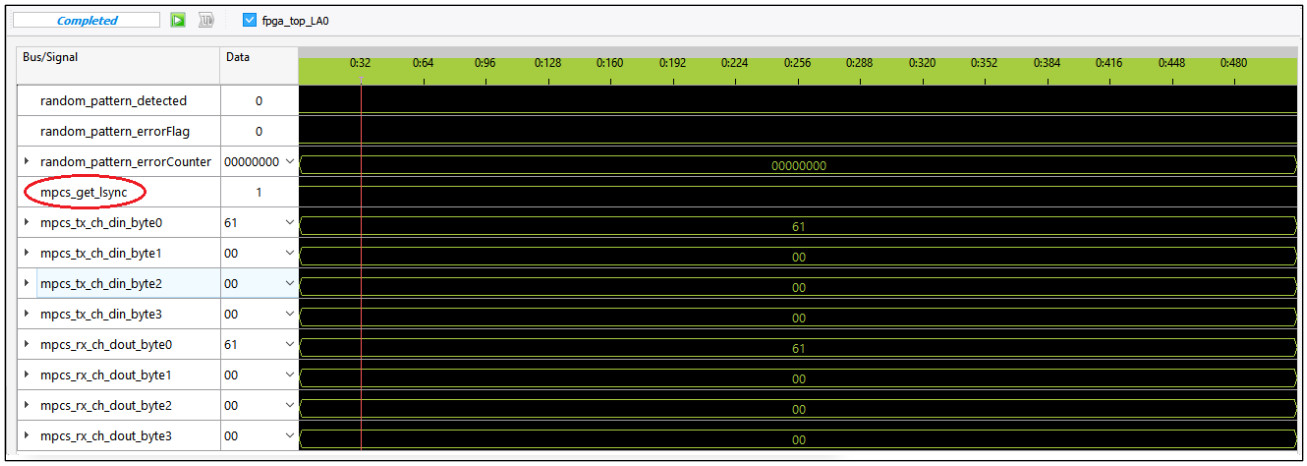

Figure 4.45. Reveal Analyzer Capture Before Pressing the Run Counter

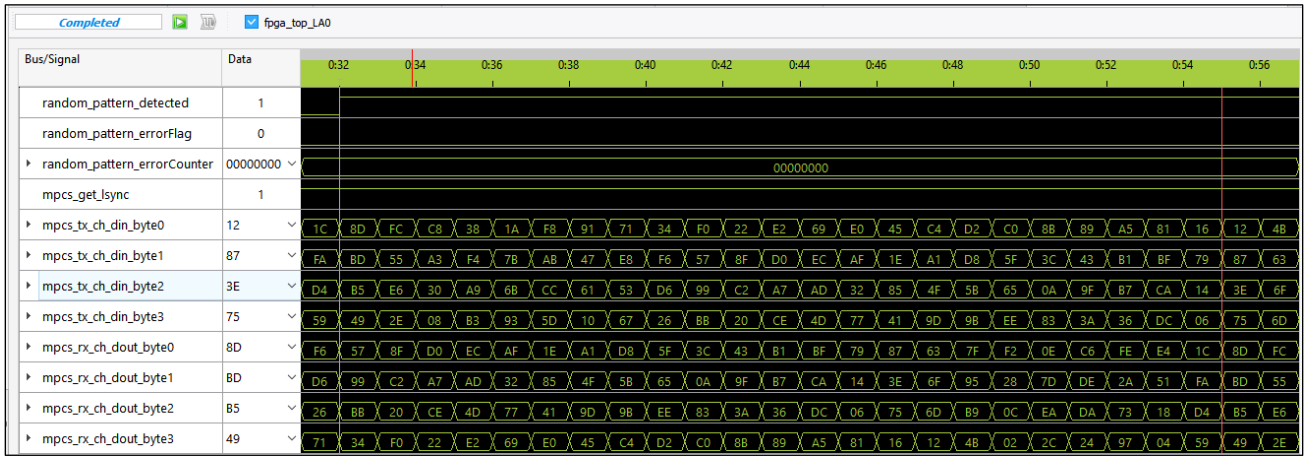

Figure 4.46. Reveal Analyzer Capture After Pressing the Run Counter

4.4.2.7. Pattern Detected

The Demo LED (LED_7) switches ON indicating the transmitted random pattern has been detected with no bit errors using the TSE protocol. This is also indicated by the *random_pattern_detected* signal assertion.

Figure 4.47. LED_7 Switches ON to Indicate Pattern Detected with No Bit Errors

References

- PCIe Multifunction Demo for Lattice Nexus-based FPGAs (FPGA-UG-02150)
- Mixed Mode PCIe and 8B10B Protocol Demo for CertusPro-NX (FPGA-UG-02192)
- CertusPro-NX Versa Board User Guide (FPGA-EB-02053)
- Tri-Speed Ethernet IP User Guide (FPGA-IPUG-02084)
- PCI Express for Nexus FPGAs web page
- Tri-Speed Ethernet IP Core web page
- CertusPro-NX web page
- Lattice Radiant FPGA design software
- Lattice Solutions Reference Designs web page
- Lattice Solutions IP Cores web page
- Lattice Radiant Software User Guide
- 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 1.0, July 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com