Lattice mVision CSI-2 PCIe Bridge Demo

User Guide

FPGA-UG-02125-1.1

March 2022

Disclaimers

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

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
C2H	Card to Host (Endpoint to Root)
CSI	Camera Serial Interface
DMA	Direct Memory Access
FPGA	Field Programmable Gate Array
ISP	Image Signal Processing
MIPI	Mobile Industry Processor Interface
OS	Operating System
PCIe	Peripheral Component Interconnect Express
SG-DMA	Scatter-Gather DMA
SG-Element	Scatter-Gather Element
SG-List	Scatter-Gather List
TLP	Transaction Layer Packet

1. Introduction

This document provides technical information and instructions of CrossLink™-NX/CertusPro™-NX MIPI CSI-2 to PCIe Bridge Demo. This design demonstrates the functionality of transferring MIPI CSI-2 camera video data to computer via PCIe with a Direct Memory Access (DMA) engine. This DMA solution can provide up to 3.1Gbps for data throughput Card to Host (C2H) without onboard DDR3 Memory, via PCIe Gen2x1 interface.

This demo is based on Lattice CrossLink-NX PCIe Bridge Board (or CertusPro-NX Versa Board), with Windows 7/Windows 10 and Linux Operating System (OS) drivers support. Total there are two modes: camera mode and color bar mode. The demo defaults to the camera mode, with the functionality of transferring camera data to computer memory and then software on a computer shows the video with the data. You can change to the color bar mode via the demo switch on the board. You can use the color bar mode to evaluate the maximum data throughput with the OS and hardware environments. Details of exchange between the two modes are discussed in later sections.

The demo package includes the following files:

- CrossLink-NX bitstream (.bit), or CertusPro-NX bitstream
- Windows driver
- Windows application software
- Linux driver source code
- Linux application software source code

Demo design hardware requirements:

- CrossLink-NX PCIe Bridge Board (Figure 1.1), or CertusPro-NX Versa Board (Figure 1.2)
- Sony IMX258 sensor module
- Computer with Linux OS, or
- Computer with Windows OS. Dual-Channel Memory is needed if there is no graphics card on Windows OS.
- USB cable for programing the CrossLink-NX/CertusPro-NX device

Demo design software requirements:

- Radiant 3.1 programmer software

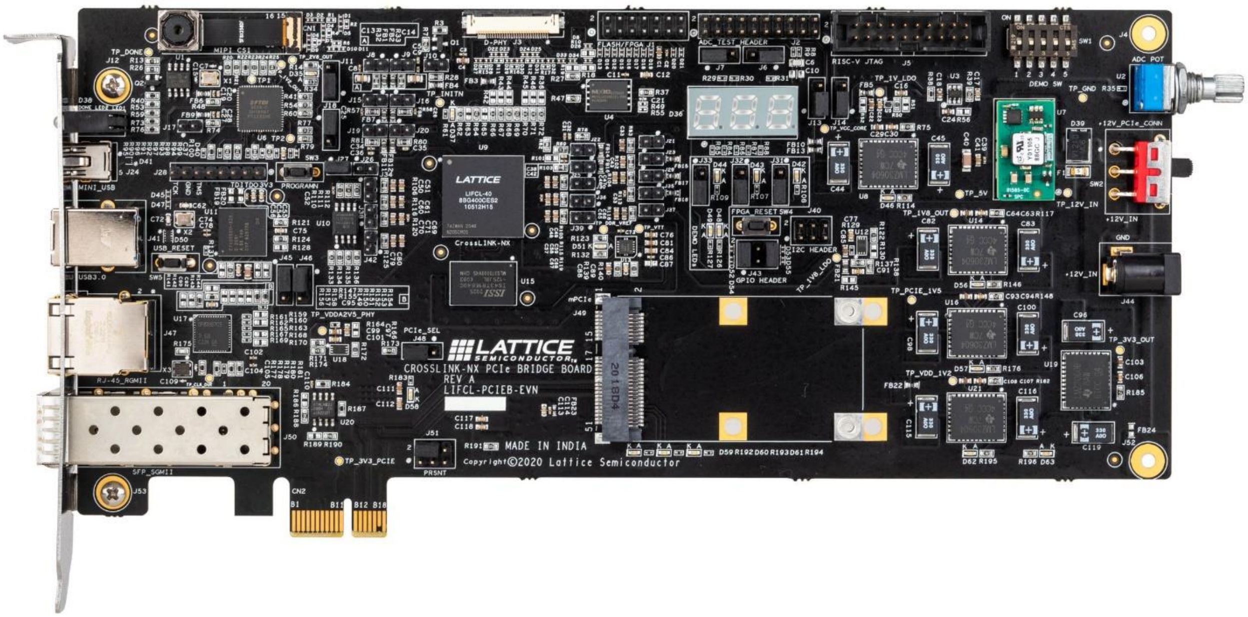


Figure 1.1. CrossLink-NX PCIe Bridge Board

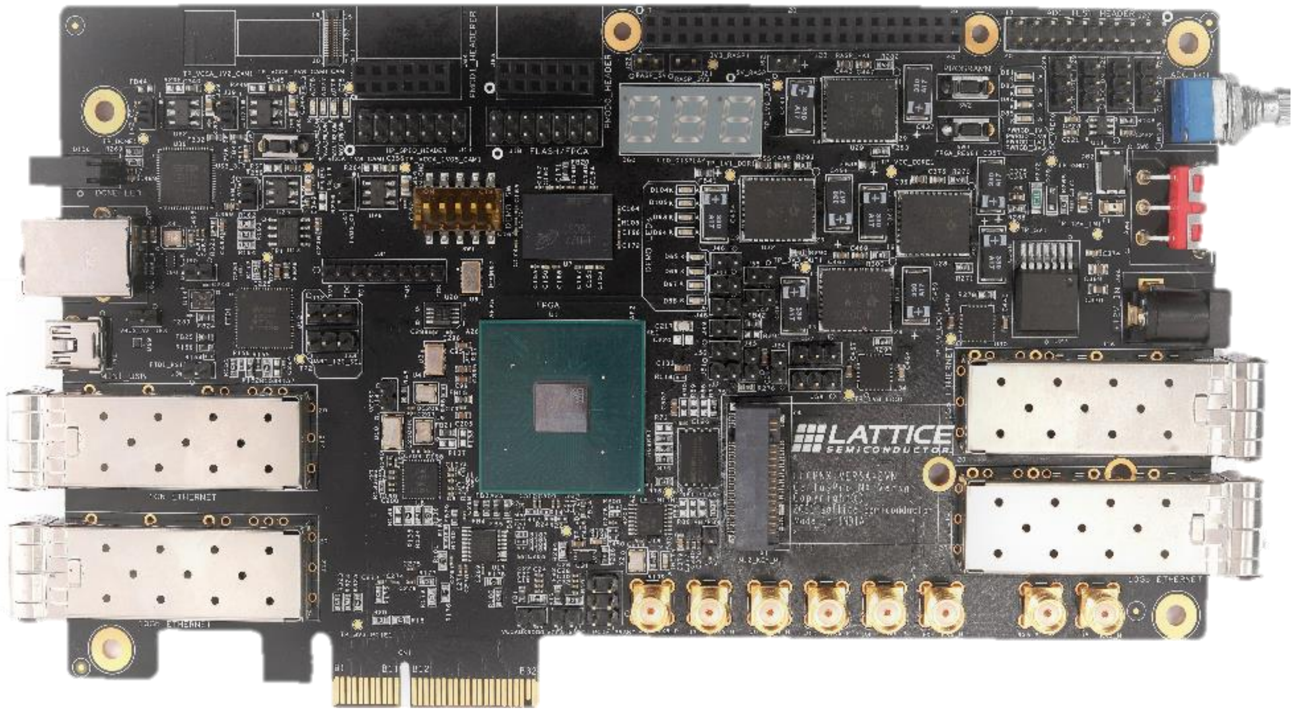


Figure 1.2. CertusPro-NX Versa Board

2. Demo Design Overview

2.1. FPGA Design Description

Figure 2.1 shows the block diagram of FPGA Design. The CrossLink-NX/CertusPro-NX PCIe hard core is configured as TLP interface. PCIe Interface block implements TLP decoder, generator, and some bridge logic to exchange data with DMA Engine module. The DMA Engine checks the computer-side frame buffer status, and then transfers video data to a specific frame buffer (allocated by OS) via PCIe interface. The video data format from IMX258 sensor is RAW8 (1080p50), so that ISP module is used to convert RAW to RGB, and then YUV422. The color Bar Gen module can generate a video test pattern whenever PCIe interface is ready to transfer.

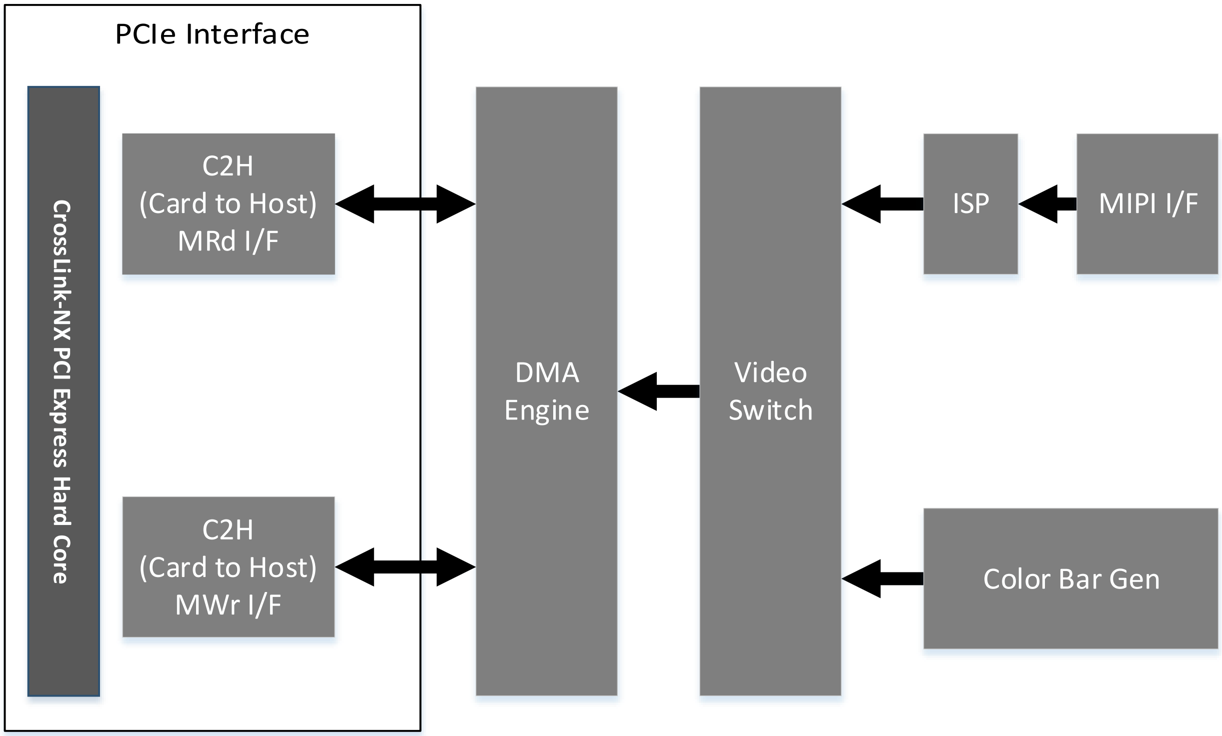


Figure 2.1. FPGA Design Block Diagram

2.2. DMA Architecture

DMA is a technique for transferring blocks of data between the computer system memory and peripherals (such as PCIe Endpoint) without a processor (such as system CPU) involved in each transfer. DMA not only offloads the system processing elements, but also transfers data at much higher rates than processor for read and write.

Scatter-Gather DMA (SG-DMA) augments this technique by providing data transfers from one non-contiguous block of memory to another by means of a series of smaller contiguous-block transfers. The DMA solution inside this demo implements a stream based DMA engine with scatter-gather capability. Stream based data is ready for transmission via PCIe continuously, and the data does not have a unique local address. Without this DMA solution, a large external memory is needed to buffer local data at FPGA side, and to wait for PCIe to be ready to transfer these data. Compared to stream based DMA, the DMA engine which needs external large memory is usually referred as memory mapped DMA.

The driver and application software implements several frame buffers for video data receive from the PCIe Endpoint (FPGA), and each frame buffer has a descriptor buffer to record the scatter gather list (SG-List). Figure 2.2 shows the relationship between the DMA frame buffer and description buffer. The SG-List records the necessary control information for data transfers:

- Numbers of SG-Elements
- The Location of each SG-Element

Each SG-Element contains:

- Physical address for each small buffer
- Length for each small buffer

After the DMA engine is enabled and started, the PCIe Endpoint (FPGA) launches Memory Read (MRd) operations from descriptor buffer to get these control information. Then the PCIe Endpoint launches Memory Write (MWr) operations to the specific frame buffer whenever the PCIe interface is not busy and the FPGA has data to transfer. The driver and application software on the computer side read the video data inside the frame buffer, and then display them on the screen.

The demo implements 16 frame buffers to evaluate the maximum data rate on different OS and hardware. Eight frame buffers are enough for most PCIe video bridge design.

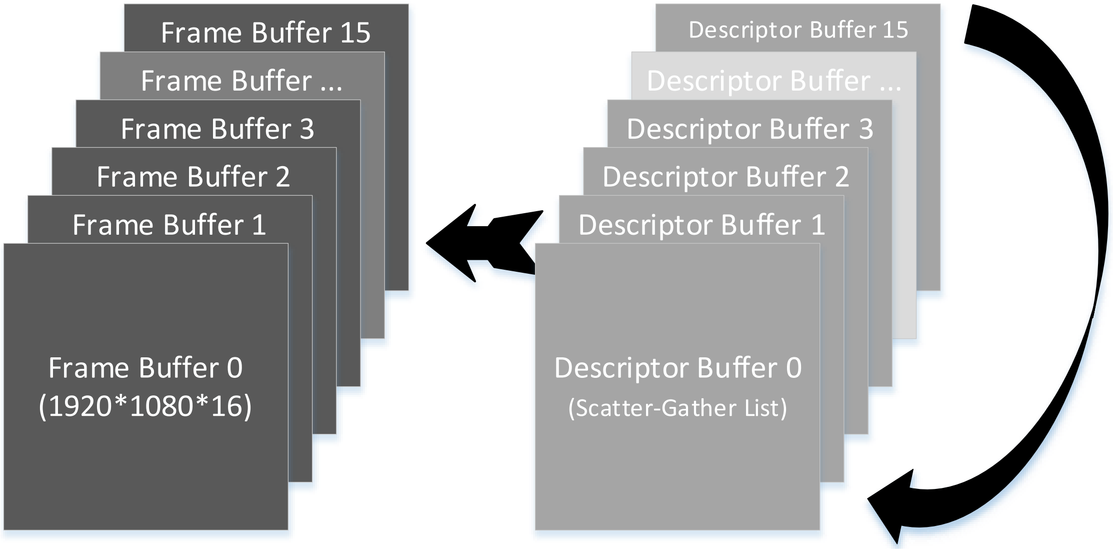


Figure 2.2. DMA Frame Buffer and Descriptor Buffer

3. CrossLink-NX PCIe Bridge Board

This section describes the CrossLink-NX PCIe Bridge Board for the CSI-2 to PCIe Bridge demo.

3.1. Hardware Description

The CrossLink-NX PCIe Bridge Board has an on-board camera connector (CN1, at the top left corner of this board). Sony IMX258 sensor module should be connected to CN1 for the demonstration.

There is a 5-bit demo switch (SW1) at the top right corner of this board. The CSI-2 to PCIe Bridge demo uses this demo switch to select the current function between the default camera mode and the color bar mode.

The CrossLink-NX device and SPI flash device on the board can be programmed over JTAG via the USB port. The instructions to set up the CrossLink-NX PCIe Bridge Board for this demo are discussed later in this document.

3.2. Jumper Settings

Following jumper settings of CrossLink-NX PCIe Bridge Board (Table 3.1) are required to enable the CSI-2 to PCIe Bridge demo.

Table 3.1. CrossLink-NX PCIe Bridge Board Jumper Settings

Jumper Name	Settings	Description
J48	Connects 2-3	Using PCIe Edge Connector
J51	Connects 2-4	—
SW2	Switch to UP	Using 12V_PCIe_CONN power

4. CertusPro-NX Versa Board

This section describes the CertusPro-NX Versa Board for the CSI-2 to PCIe Bridge demo.

4.1. Hardware Description

The CertusPro-NX Versa Board has an on-board camera connector (J37, at the top left corner of this board). Sony IMX258 sensor module should be connected to J37 for the demonstration.

There is a 5-bit demo switch (SW1) on this board. The CSI-2 to PCIe Bridge demo uses this demo switch to select the current function between the default camera mode and the color bar mode.

The CertusPro-NX device and SPI flash device on the board can be programmed over JTAG via the USB port. The instructions to set up the CertusPro-NX Versa Board for this demo are discussed later in this document.

4.2. Jumper Settings

Following jumper settings of CertusPro-NX Versa Board (Table 4.1) are required to enable the CSI-2 to PCIe Bridge demo.

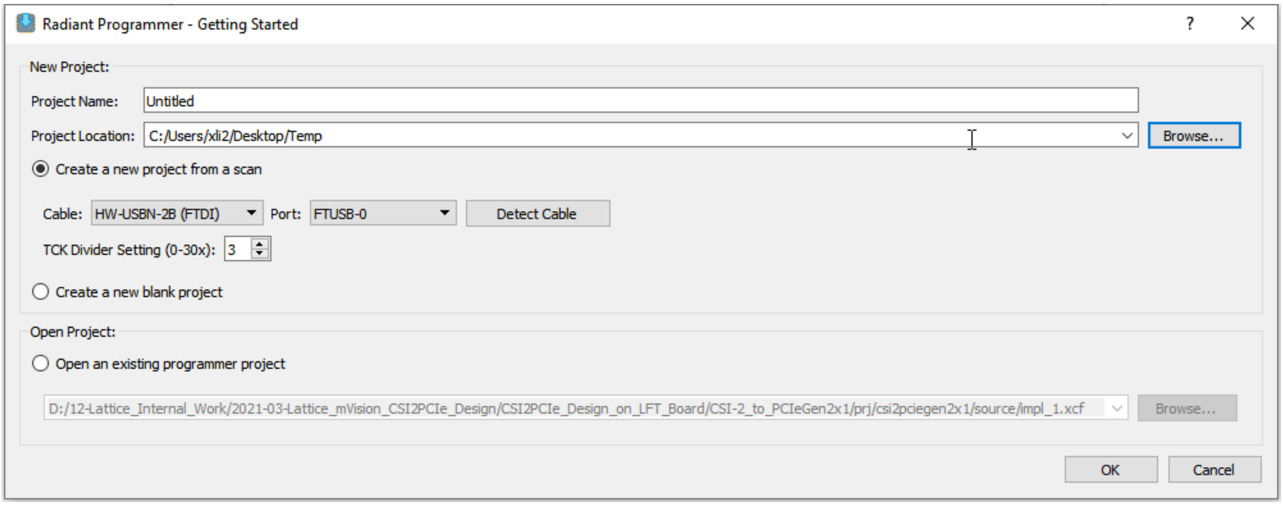
Table 4.1. CertusPro-NX Versa Board Jumper Settings

Jumper Name	Settings	Description
J1	Connects 1-2	Using PCIe x1 mode
SW6	Switch to UP	Using 12V_PCIe_CONN power

5. Program CrossLink-NX/CertusPro-NX Device

The following steps show you how to download the bitstream to CrossLink-NX device on CrossLink-NX PCIe Bridge board (or CertusPro-NX device on CertusPro-NX Versa board):

1. Connect the board to PC via USB mini port.
2. Connect the board to DC power adapter (12V).
3. Select 12 V DC power.
4. Start Radiant Programmer, Version 3.1 or later version, from your computer.
5. The Radiant Programmer – Getting Started dialog box pops up (Figure 5.1). By default, the *Create a new project from a scan* option is selected. Check and confirm other settings are correctly reflected the real components connected. Click OK, if everything is correct.



Radiant Programmer - Getting Started

New Project:

Project Name: Untitled

Project Location: C:/Users/xli2/Desktop/Temp

● Create a new project from a scan

Cable: HW-USBN-2B (FTDI) Port: FTUSB-0

TCK Divider Setting (0-30x): 3

○ Create a new blank project

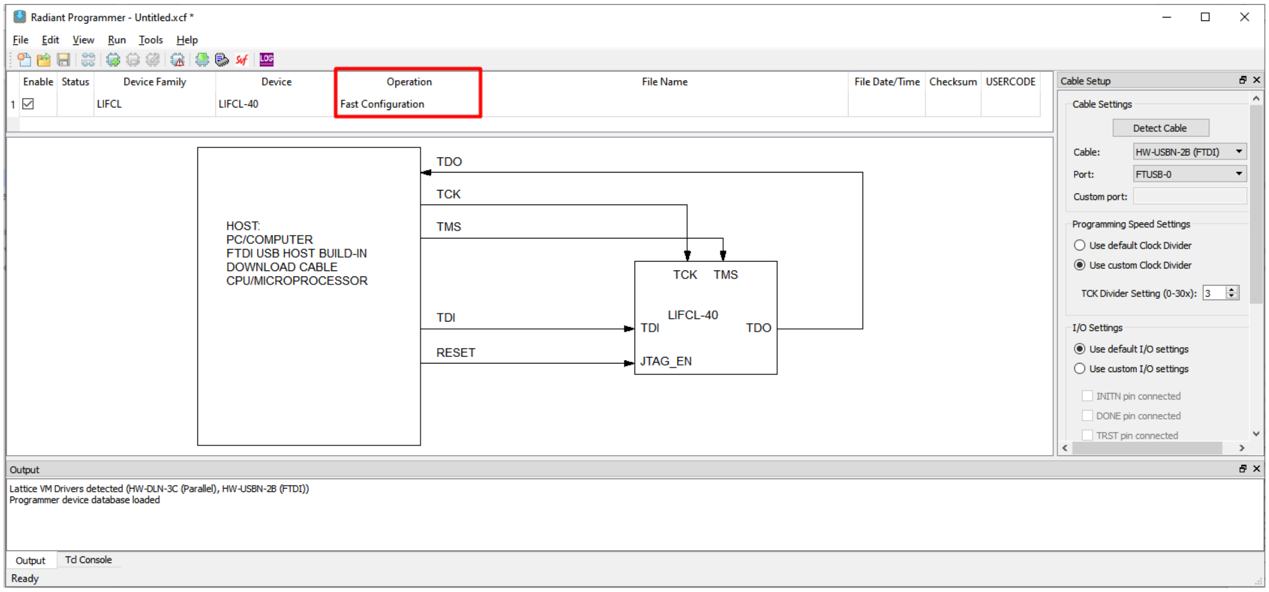
Open Project:

○ Open an existing programmer project

D:/12-Lattice_Internal_Work/2021-03-Lattice_mVision_CSI2PCIe_Design/CSI2PCIe_Design_on_LFT_Board/CSI-2_to_PCIeGen2x1/prj/csi2pciegen2x1/source/impl_1.xcf

Figure 5.1. Getting Started Dialog in Lattice Radiant Programmer

6. A new window pops up as Figure 5.2 shows. Double-click Fast Configuration from the Operation column.



Radiant Programmer - Untitled.xcf *

	Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	☑		LIFCL	LIFCL-40	Fast Configuration				

Cable Setup

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

○ Use default Clock Divider

● Use custom Clock Divider

TCK Divider Setting (0-30x): 3

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

□ DONE pin connected

□ TRST pin connected

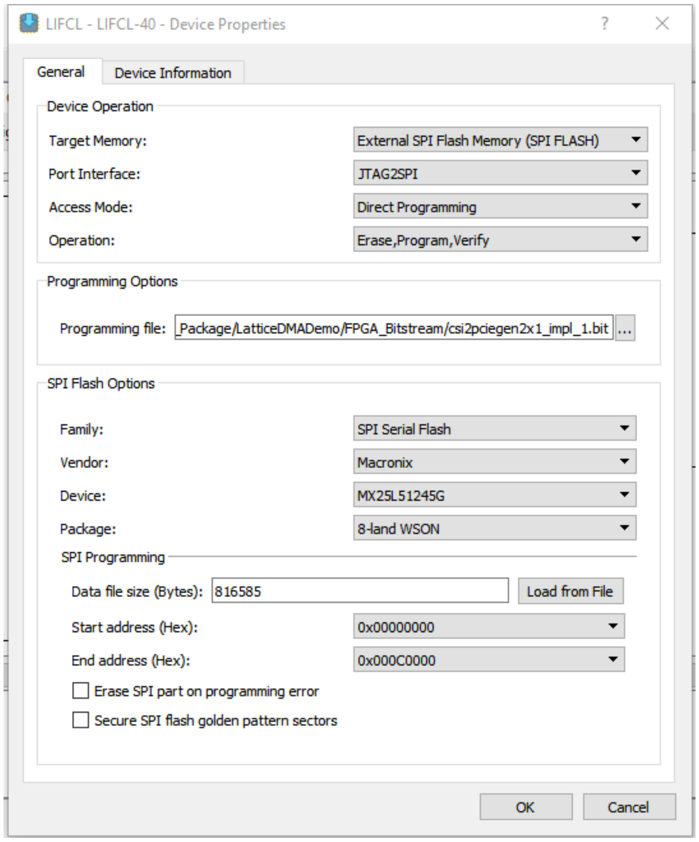
Output

```
Lattice VM Drivers detected (HW-DLN-3C (Parallel), HW-USBN-2B (FTDI))
Programmer device database loaded
```

Ready

Figure 5.2. Radiant Programmer Main Window

7. The Device Properties dialog pops up. Changed the settings as shown in Figure 5.3. Click OK.



LIFCL - LIFCL-40 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: Package/LatticeDMADemo/FPGA_Bitstream/csi2pciegen2x1_impl_1.bit ...

SPI Flash Options

Family: SPI Serial Flash

Vendor: Macronix

Device: MX25L51245G

Package: 8-land WSON

SPI Programming

Data file size (Bytes): 816585 Load from File

Start address (Hex): 0x00000000

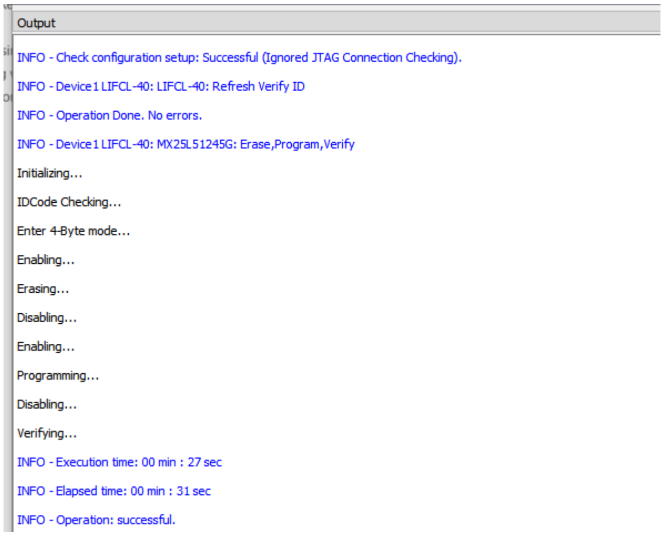
End address (Hex): 0x000C0000

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

Figure 5.3. Device Properties Dialog

8. Click the Program icon from the toolbar, or choose the Design > Program menu item from Radiant Programmer, to program the CrossLink-NX/CertusPro-NX board. Wait for the programming to finish. Check the programming status and result from the output pane, as shown in Figure 5.4.



Output

```
INFO - Check configuration setup: Successful (Ignored JTAG Connection Checking).
INFO - Device1 LIFCL-40: LIFCL-40: Refresh Verify ID
INFO - Operation Done. No errors.
INFO - Device1 LIFCL-40: MX25L51245G: Erase,Program,Verify
Initializing...
IDCode Checking...
Enter 4-Byte mode...
Enabling...
Erasing...
Disabling...
Enabling...
Programming...
Disabling...
Verifying...
INFO - Execution time: 00 min : 27 sec
INFO - Elapsed time: 00 min : 31 sec
INFO - Operation: successful.
```

Figure 5.4. Programmer Output Window

9. Select 12 V power from PCIe edge connector.

6. Install Driver for OS

This section provides the processes and guidelines of installing drivers on both Linux OS and Windows OS.

6.1. Linux OS

Linux systems require device drivers to correspond to specific kernel versions, so we provide the driver and application software directly to support different version Linux OS.

Follow steps below to install drivers on Linux OS:

1. Start Linux OS. Take Ubuntu 20.04 LTS for example [Kernel Version: 5.4.0-42].
2. Open the Terminal. Find *LatticeDMADemo* folder location.
3. Input: cd *LatticeDMADemo/DriverandApplication/linux*
4. Input: cd *Driver*
5. Input: sudo apt-get install dkms
6. Input: make
7. Input: sudo insmod pcie-dma.ko

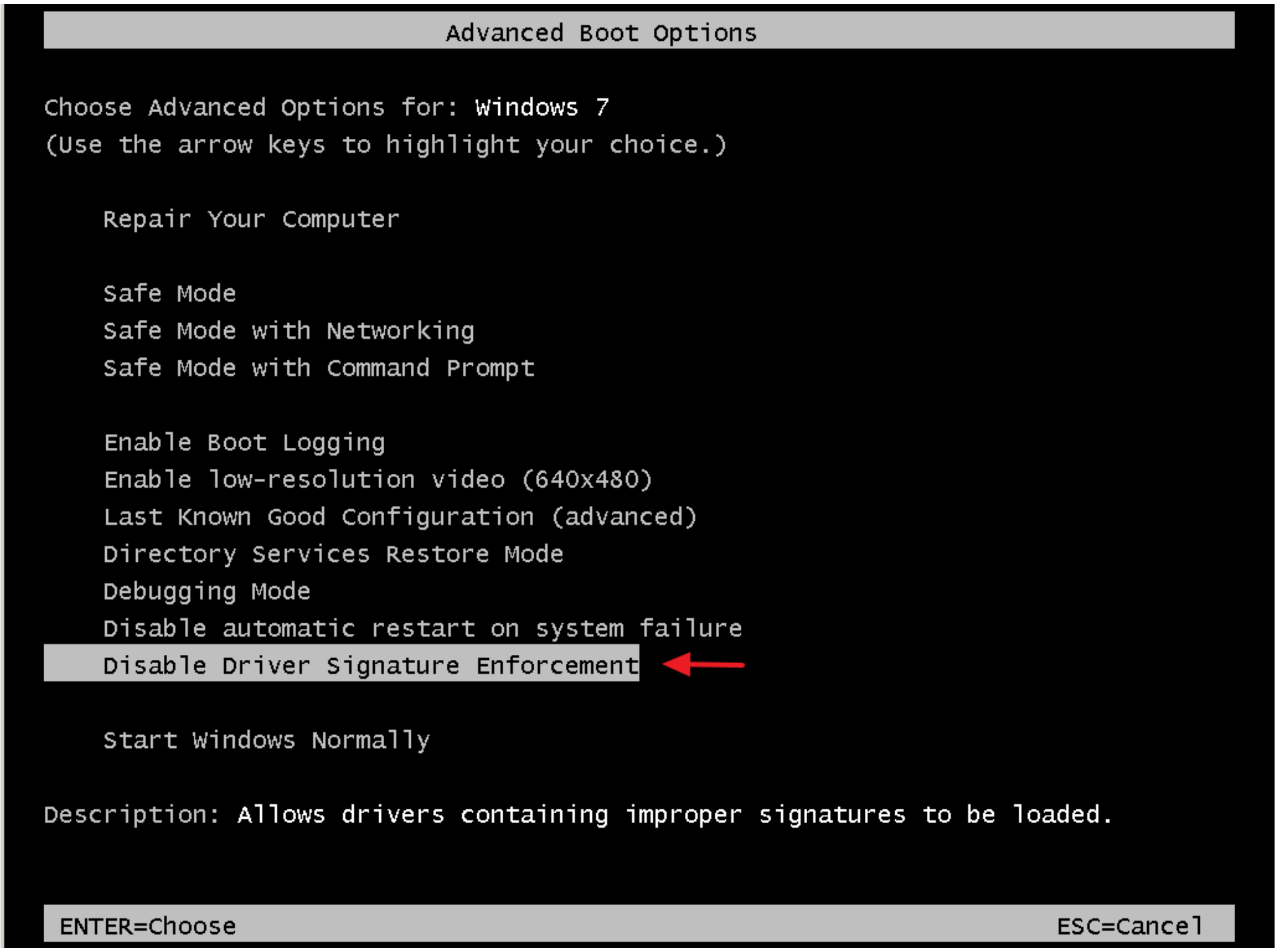
Now the driver of the requested Linux OS is ready for the application software.

Note: The driver must be reinstalled using only Step 7 (Input: sudo insmod pcie-dma.ko) every time the computer is restarted.

6.2. Windows 7 OS

You need to temporarily disable the operating system digital signature enforcement checking, considering this driver does not have a valid digital signature certification. Follow steps below:

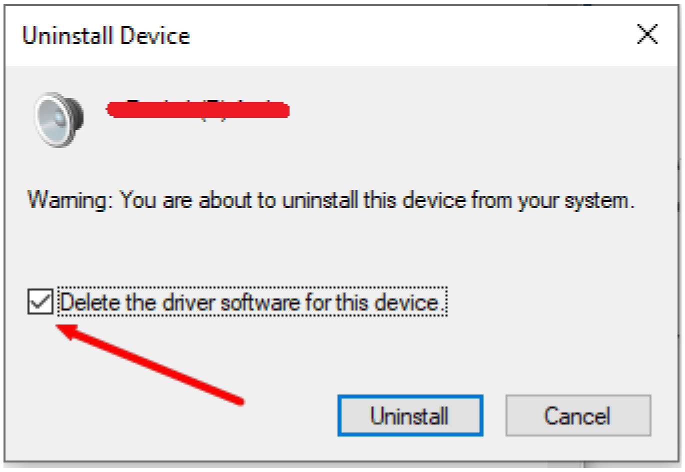
1. Start Windows 7 OS, and log in using the Administrator account.
2. Open cmd.exe. Input “bcdedit.exe -set loadoptions DDISABLE_INTEGRITY_CHECKS”.
3. Restart computer. Press the **F8** key on keyboard to enter Advanced Boot Options window. Refer to Figure 6.1.



```
                          Advanced Boot Options

Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)

    Repair Your Computer

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally

Description: Allows drivers containing improper signatures to be loaded.


 ENTER=Choose                                                       ESC=Cancel
```

Figure 6.1. Advanced Boot Options

4. Open the Device Manager, and find "PCI Data Capture and Signal Processing Controller" under Other Devices.
5. You need to uninstall device and delete the old version driver, if it is previously installed. After that, install the new version driver for this device. Refer to Figure 6.2.



Uninstall Device

[redacted]

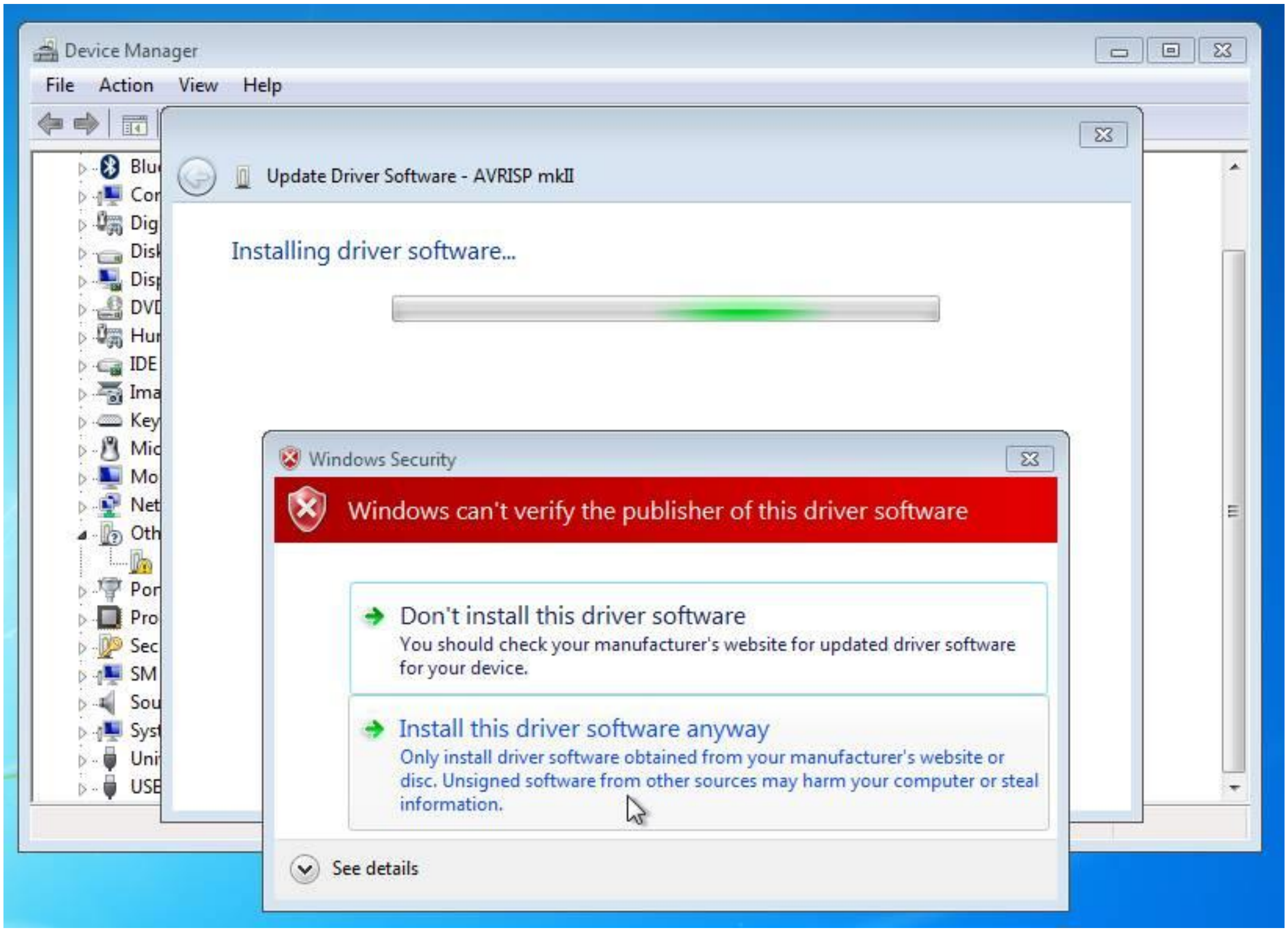
Warning: You are about to uninstall this device from your system.

☑ Delete the driver software for this device.

Figure 6.2. Uninstall Old Drivers if Needed

6. Right-click the device, then select *update driver*, and select the driver location.

7. During the installation procedure, Windows inform you that it cannot verify the publisher of this driver software. Ignore the warning message, and choose *Install this driver software anyway* to complete the installation, as shown in Figure 6.3.



Update Driver Software - AVRISP mkII

Installing driver software...

Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Figure 6.3. Install This Driver Software Anyway

8. The installation completes successfully if no error message is shown.

Now the operating system digital signature enforcement checking is disabled, and the driver is ready for the demo.

Note: After the next restart, Driver Signature Enforcement is automatically enabled again to prevent new drivers from being installed.

6.3. Windows 10 OS

You need to temporarily disable the operating system digital signature enforcement checking, considering this driver does not have a valid digital signature certification. Follow steps below:

1. Start Windows 10 OS, and log in using the Administrator account.
2. Open the cmd.exe. Input “bcdedit.exe -set loadoptions DDISABLE_INTEGRITY_CHECKS”.
3. Open **Windows Settings** page.
4. Click **Update & Security**
5. Select **Recovery** from the right pane. Click the **Restart now** button below **Advanced startup.**
6. Once the computer reboots, you need to choose the **Troubleshoot** option (Figure 6.4).

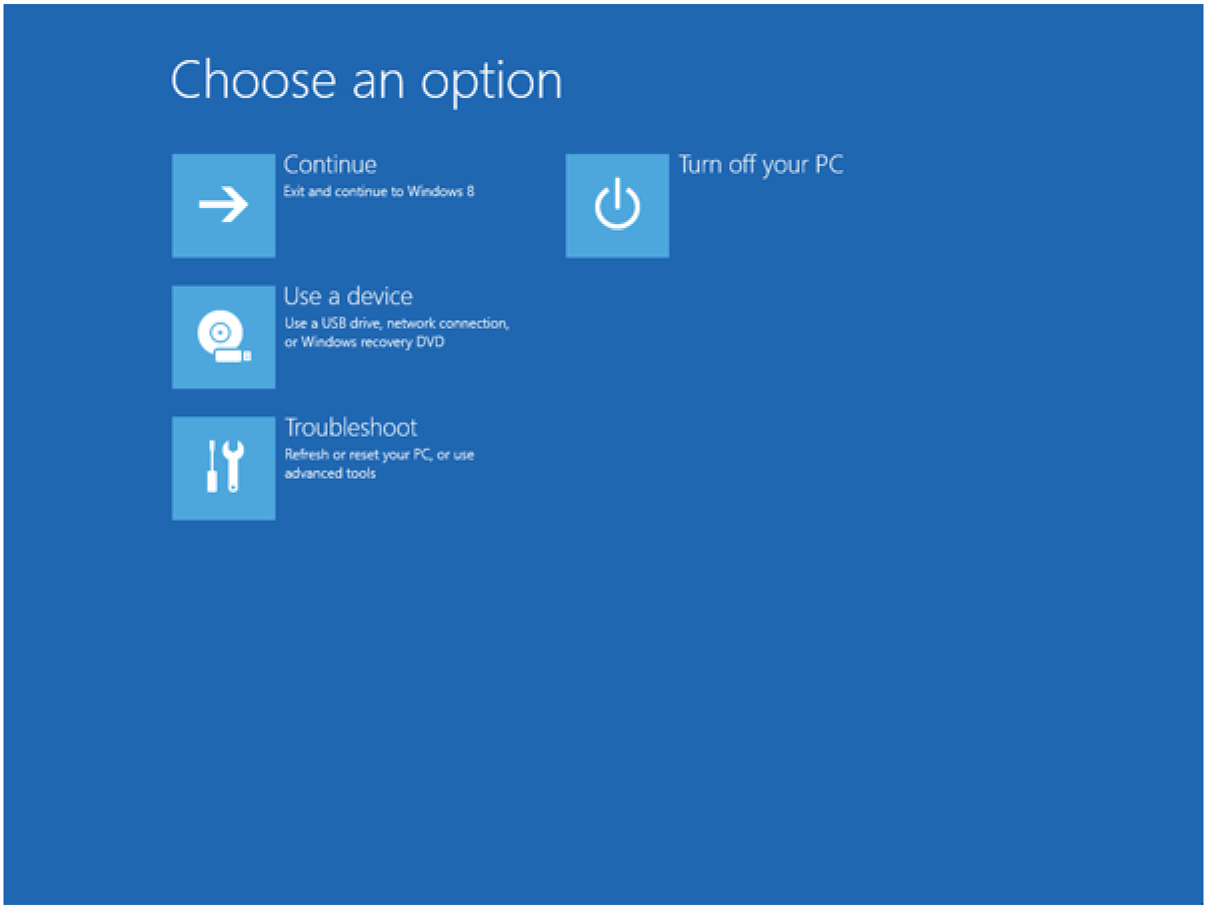
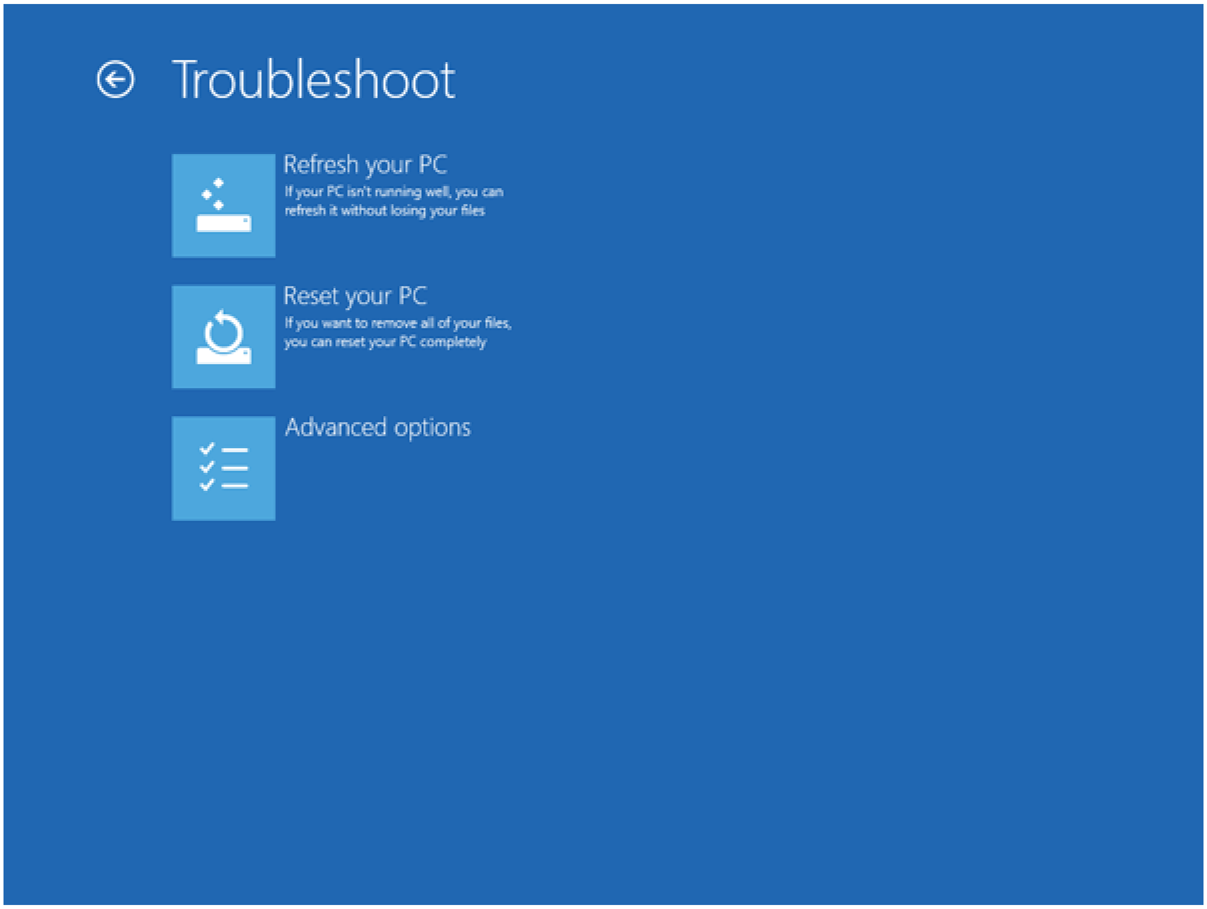

Choose an option
Continue
Exit and continue to Windows 8
Turn off your PC
Use a device
Use a USB drive, network connection, or Windows recovery DVD
Troubleshoot
Refresh or reset your PC, or use advanced tools

Figure 6.4. Choose the Troubleshoot Option

7. Go to **Advanced Options** (Figure 6.5).



Troubleshoot

Refresh your PC

If your PC isn't running well, you can refresh it without losing your files

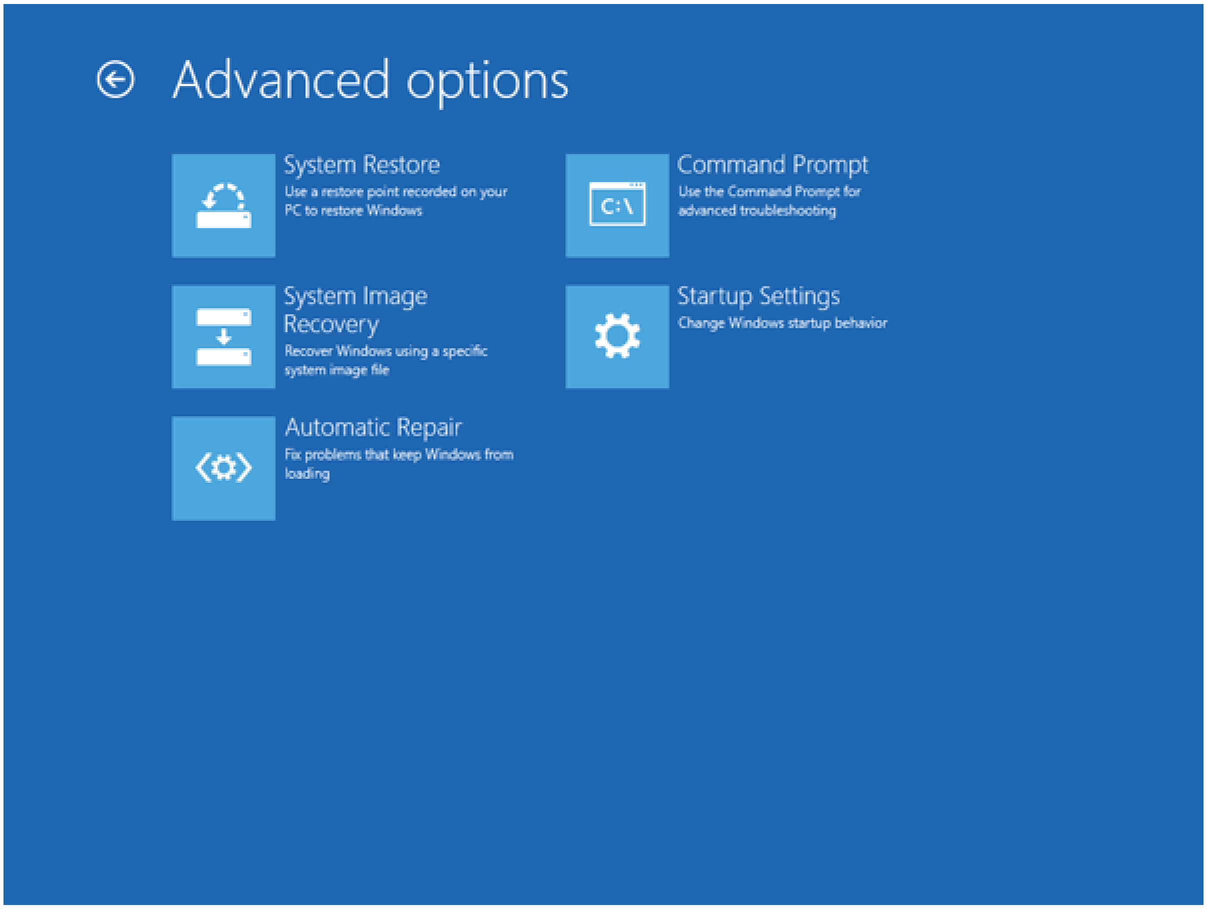
Reset your PC

If you want to remove all of your files, you can reset your PC completely

Advanced options

Figure 6.5. Troubleshoot > Advanced options

8. In the Advanced Options window, choose **Startup Settings** (Figure 6.6).



Advanced options

System Restore
Use a restore point recorded on your PC to restore Windows

Command Prompt
Use the Command Prompt for advanced troubleshooting

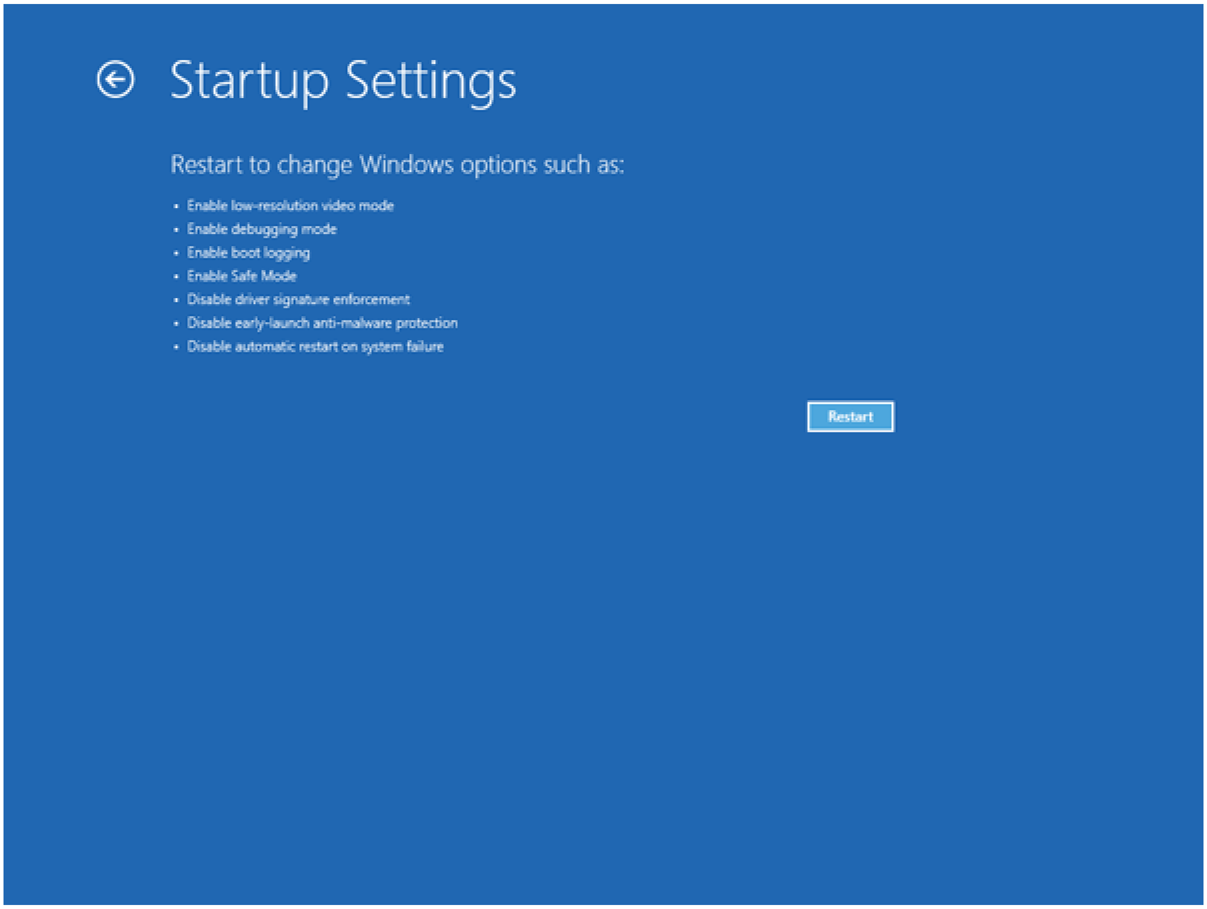
System Image Recovery
Recover Windows using a specific system image file

Startup Settings
Change Windows startup behavior

Automatic Repair
Fix problems that keep Windows from loading

Figure 6.6. Advance Options > Startup Settings

9. Click the **Restart** button on from the Startup Settings to reboot the computer again (Figure 6.7).



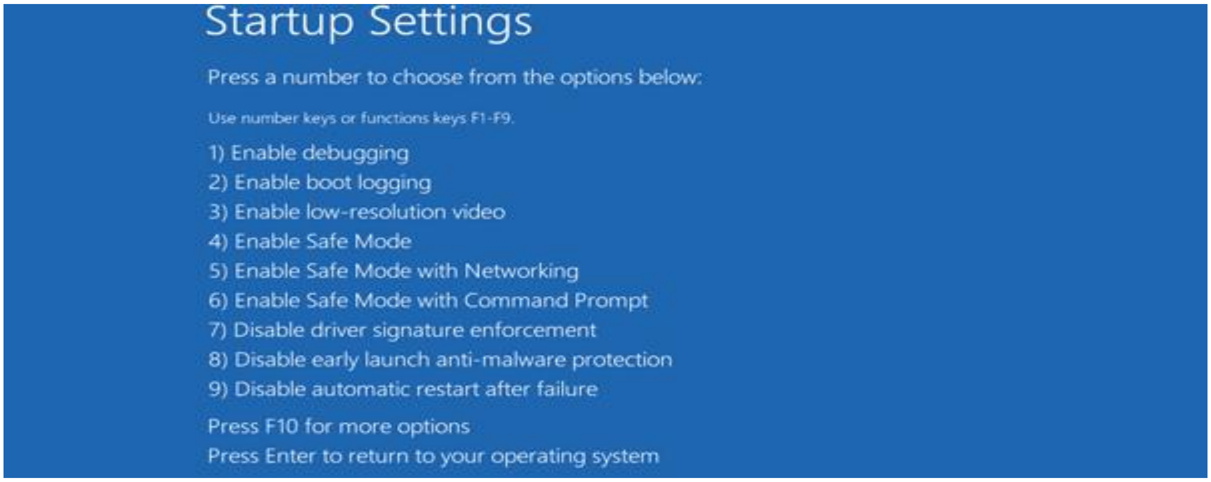
Startup Settings

Restart to change Windows options such as:

- Enable low-resolution video mode
- Enable debugging mode
- Enable boot logging
- Enable Safe Mode
- Disable driver signature enforcement
- Disable early-launch anti-malware protection
- Disable automatic restart on system failure

Figure 6.7. Startup Settings > Restart

10. After restart in Startup Settings window, press the **F7** key from the keyboard to select “**Disable driver signature enforcement**” (Figure 6.8).



Startup Settings

Press a number to choose from the options below:

Use number keys or functions keys F1-F9.

1) Enable debugging
2) Enable boot logging
3) Enable low-resolution video
4) Enable Safe Mode
5) Enable Safe Mode with Networking
6) Enable Safe Mode with Command Prompt
7) Disable driver signature enforcement
8) Disable early launch anti-malware protection
9) Disable automatic restart after failure

Press F10 for more options
Press Enter to return to your operating system

Figure 6.8. Startup Settings [Next Page]

11. The computer is rebooted to proceed to install unsigned drivers. You need to uninstall the device and delete the driver that was previously installed, if you updated the driver for that device (Figure 6.2).

12. During the installation procedure, Windows informs you that it cannot verify the publisher of this driver software. Ignore the warning message, and choose *Install this driver software anyway* to complete the installation, as shown in Figure 6.3.
13. The installation completes successfully if the last window shows no error message.

Now the operating system digital signature enforcement checking is disabled, and the driver is ready for the demo.

Note: After the next restart, Driver Signature Enforcement is automatically enabled again to prevent new drivers from being installed.

7. Run This Demo

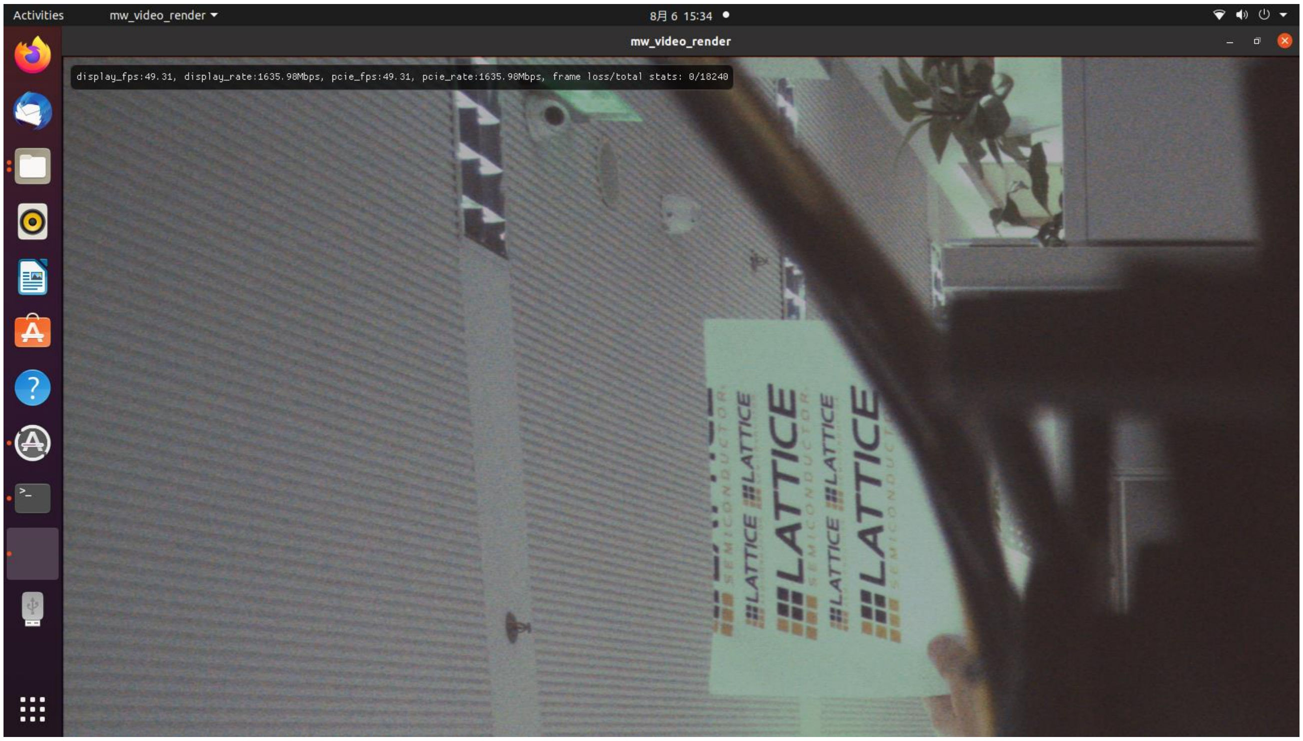
This section shows you how to run this demo design on target computer with either Linux OS or Window OS.

7.1. Linux OS

As mentioned earlier, we provide the driver and application software source code directly. You need to compile the application source code at first. Follow steps below to compile the application source code:

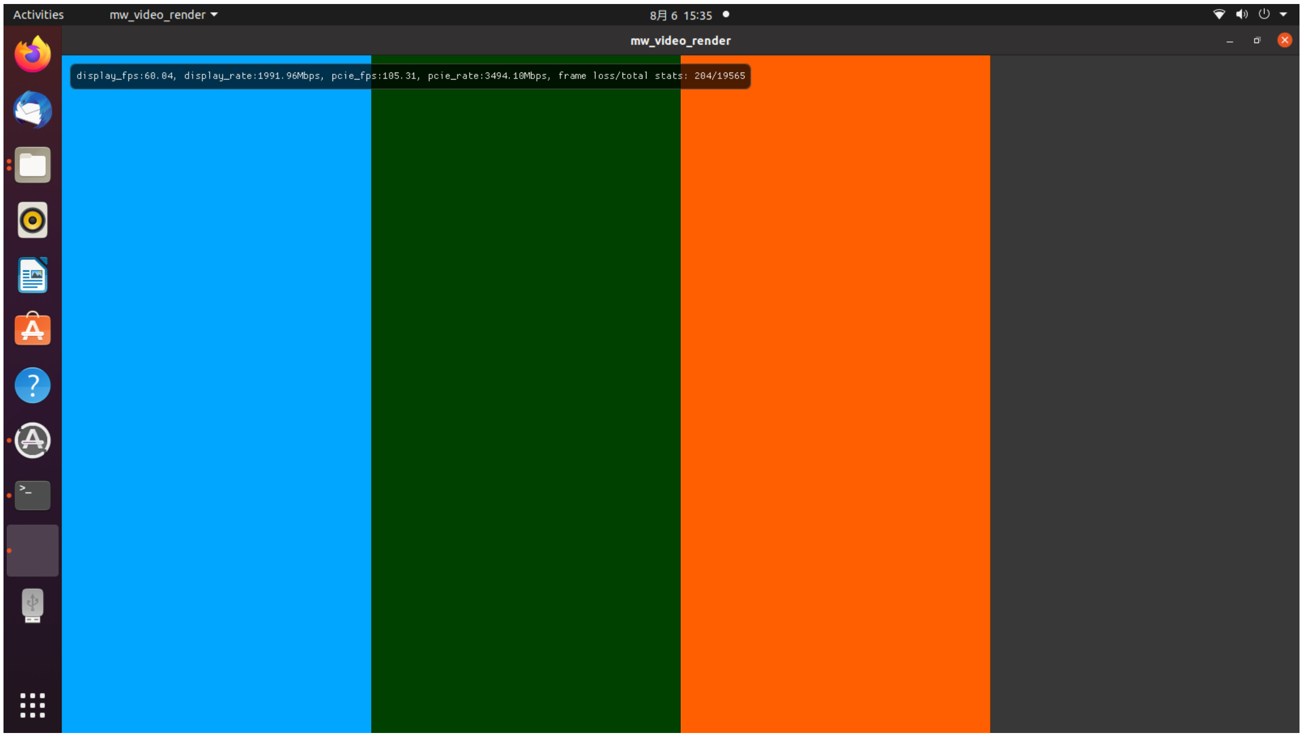
1. Start Linux OS. Take Ubuntu 20.04 LTS for example [Kernel Version: 5.4.0-42].
2. Open the Terminal. Find *LatticeDMADemo* folder location.
3. Input: cd *LatticeDMADemo/DriverandApplication/linux/App*
4. Input: mkdir build, then input: cd build
5. Input: sudo apt-get install libglew-dev
6. Input: sudo apt-get install libglfw3 libglfw3-dev
7. Input: cmake ..
8. Input: make
9. Input: cd src
10. Now, you can run this demo by inputting sudo ./pcie_dma

The default functionality is MIPI CSI-2 to PCIe Bridge. Sony IMX258 sensor data is transferred to computer memory via PCIe, then display on screen (Figure 7.1). You can change from the default camera mode to the color bar mode by changing the demo switch state. After changing to color bar mode, you can see screen as shown in Figure 7.2. You can exchange between the two modes.



```
display_fps:49.31, display_rate:1635.98Mbps, pcie_fps:49.31, pcie_rate:1635.98Mbps, frame loss/total stats: 0/18240
```

Figure 7.1. CSI-2 to PCIe Bridge

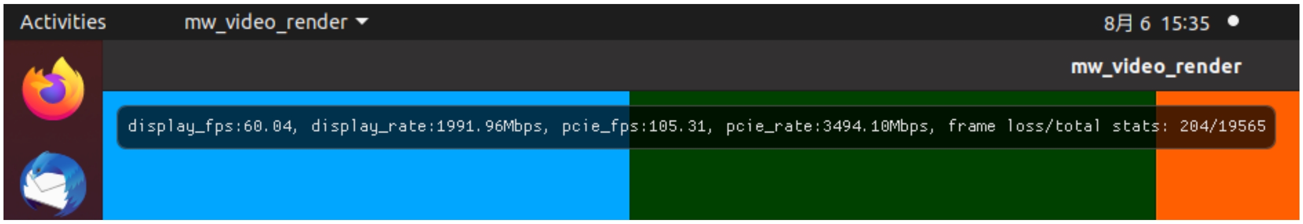


```
display_fps:60.04, display_rate:1991.96Mbps, pcie_fps:105.31, pcie_rate:3494.10Mbps, frame loss/total stats: 204/19565
```

Figure 7.2. Color Bar Mode

As shown in Figure 7.3, the top left of the window displays the following information:

- Displayed frames per second
- Displayed data rate
- PCIe transmitted frames per second
- PCIe transmitted throughput rate
- The number of frames discarded by software
- The total number of frames transmitted via PCIe



```
display_fps:60.04, display_rate:1991.96Mbps, pcie_fps:105.31, pcie_rate:3494.10Mbps, frame loss/total stats: 204/19565
```

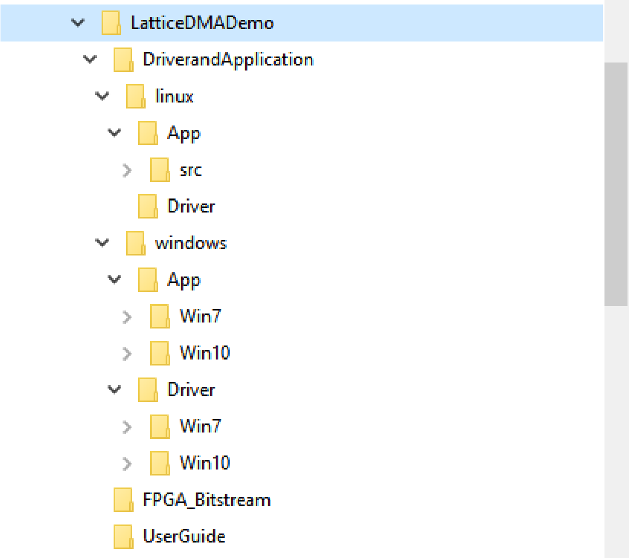
Figure 7.3. Display Information

7.2. Windows OS

For Windows OS, double-click pcie_dma_test.exe. A window similar to Figure 7.1 pops up, showing the default camera mode. You can switch between the camera mode and the color bar mode by changing the demo switch state.

8. Demo Package Directory Structure

Figure 8.1 shows the directory structure of this demo package. The *DriverandApplication/linux* folder contains driver and application software source code for Linux OS; the *DriverandApplication/windows* folder contains driver and application software files for both Win7 and Win10 OS. The “*FPGA_Bitstream*” folder contains the bitstream file for CrossLink-NX FPGA device.



```
LatticeDMADemo
  DriverandApplication
    linux
      App
        src
      Driver
    windows
      App
        Win7
        Win10
      Driver
        Win7
        Win10
  FPGA_Bitstream
  UserGuide
```

Figure 8.1. Demo Package Directory Structure

References

- PCIe X1 IP Core - Lattice Radiant Software (FPGA-IPUG-02091)
- CrossLink-NX PCIe Bridge Board Hardware (FPGA-EB-02040)
- PCIe X4 IP Core - Lattice Radiant Software (FPGA-IPUG-02126)
- CertusPro-NX Versa Board (FPGA-EB-02053)

Technical Support Assistance

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

Revision History

Revision 1.1, February 2022

Section	Change Summary
Introduction	• added the support of the CertusPro-NX Versa board. • updated to Radiant 3.1 Programmer software. • updated Figure 1.1. CrossLink-NX PCIe Bridge Board and Figure 1.2. CertusPro-NX Versa Board.
Demo Design Overview	Added the support of the CertusPro-NX Versa board.
CrossLink-NX PCIe Bridge Board	Added the support of the CertusPro-NX Versa board.
CertusPro-NX Versa Board	Newly added section for the support of the CertusPro-NX Versa board.
Program CrossLink-NX/CertusPro-NX Device	• added the support of the CertusPro-NX Versa board; • updated to *Radiant Programmer* in Step 8; • updated to *Select 12 V power* in Step 9.
References	Newly added the last two reference documents: PCIe X4 IP Core - Lattice Radiant Software (FPGA-IPUG-02126) and Lattice CertusPro-NX Versa Board (FPGA-EB-02053).

Revision 1.0, March 2021

Section	Change Summary
All	Production release.

LATTICE
SEMICONDUCTOR®

www.latticesemi.com