Lattice mVision DisplayPort to PCIe Demo

User Guide

FPGA-UG-02227-1.1

March 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
ATA	Advanced Technology Attachment
AUX	Auxiliary Channel
CPU	Central Processing Unit
DMA	Direct Memory Access
DP	DisplayPort
DSI	Display Serial Interface
DSP	Digital Signal Processor
EBR	Embedded Block RAM
FMC	FPGA Mezzanine Card
FPS	Frames Per Second
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HPD	Hot Plug Detect
LUT	Look-Up Table
MIPI	Mobile Industry Processor Interface
MPCS	Multiple-Protocol Physical Coding Sublayer
MSA	Main Stream Attribute
OS	Operating System
PC	Personal Computer
PCIe	Peripheral Component Interconnect Express
RX	Receiver
SPI	Serial Peripheral Interface
VDMA	Video Direct Memory Access
VTG	Video Timing Generator

1. Introduction

The CertusPro™-NX DisplayPort (DP) to Peripheral Component Interconnect Express (PCIe®) bridge design demonstrates the functionality of transferring DisplayPort video data to a computer through PCIe with a direct memory access (DMA) engine. This demo is based on the CertusPro-NX PCIe Bridge Board with Linux® operating system (OS) driver support that shows transfer of 4K60 video data to the computer memory. The board renders the data as video on the computer screen using the software driver.

2. Hardware and Software Requirements

2.1. Hardware Requirements

- CertusPro-NX PCIe Bridge Board RevC
- Modular FPGA mezzanine card (FMC) adapter
- DisplayPort receiver daughter card
- Host personal computer (PC) (specifications: Intel CPU i7-14700 on B760 Aorus Elite AX or an equivalent PC that supports 4K60 DisplayPort output)
- Monitor
- Mini USB Type A cable for programming the bitstream
- DisplayPort cable

2.2. Software Requirements

- The Lattice™ Radiant™ software version 2023.2 or later
- Ubuntu® 24.04
- Lattice mVision™: DP-to-PCIe demo software

To download the demo files, refer to the Lattice mVision DisplayPort to PCIe Demonstration web page.

3. Demo Design Overview

3.1. System Design

This design requires DisplayPort source, such as laptop or Graphics Processing Unit (GPU), to provide a 4K60 video source with video format RGB 8 bits per colour (bpc) to the FPGA. The DisplayPort receiver (RX) receives the video stream and sends the video stream to the PCIe interface through a video bridge. The PCIe Video DMA (VDMA) processes the video data movement between video interfaces and the target central processing unit (CPU) memory. The video data is displayed in a host PC.

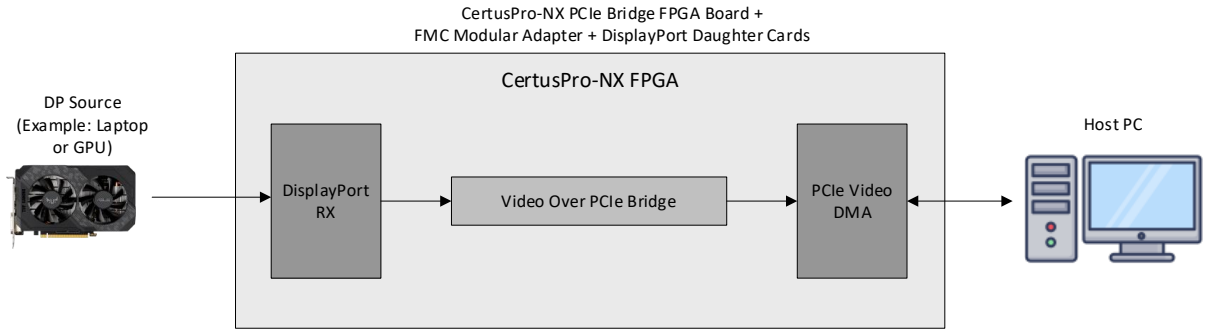

Figure 3.1. Top-Level Architecture of the System Design

3.1.1. DisplayPort Receiver

The DisplayPort IP in this demo focuses only on the DisplayPort receiver as described in the following subsections.

3.1.1.1. Configuration

The DisplayPort receiver can start automatically on power up.

3.1.1.2. Training

When the DisplayPort receiver is active, the receiver pulls up the Hot Plug Detect (HPD) line to indicate the presence of a sink to the source. The source starts the training process using the auxiliary channel (AUX) interface. The training sequence includes clock recovery, channel equalization, and symbol lock. Training is initiated by AUX channel, but the training data (TPS1, TPS2, TPS3, or TPS4) is transmitted over the main channel.

3.1.1.3. Normal Operation

When the training is complete, video data starts coming on the main channel. The main channel data is converted to parallel data, deskewed, aligned, and descrambled. The descrambled data is formatted to detect different control symbols and video. The pixel data is demapped from lane formatted data and the video timing data is generated by the video timing generator (VTG) based on the received video information in the main stream attribute (MSA) data field. The video data consisting of pixel data and timing information is sent out through the Unified Video Streaming interface.

3.1.2. PCIe Video DMA

The DMA engine provides a high-performance DMA data transmission between PCIe and a local user interface (Unified Video Streaming interface). The high-performance data movement between the host memory and the FPGA local memory is achieved using the PCIe link and the driver software running on the host. Refer to the Software Setup section for the driver software.

4. Setting Up the Demo

4.1. Hardware Setup

This section covers the hardware setup and the steps to program the demo to the serial peripheral interface (SPI) memory of the FPGA board.

This demo is tested on the following setup:

- Input video source: Intel® UHD Graphics 770 (from Intel i7-14700)
- Host PC: Intel CPU i7-14700 + Gigabyte B760 Aorus Elite AX

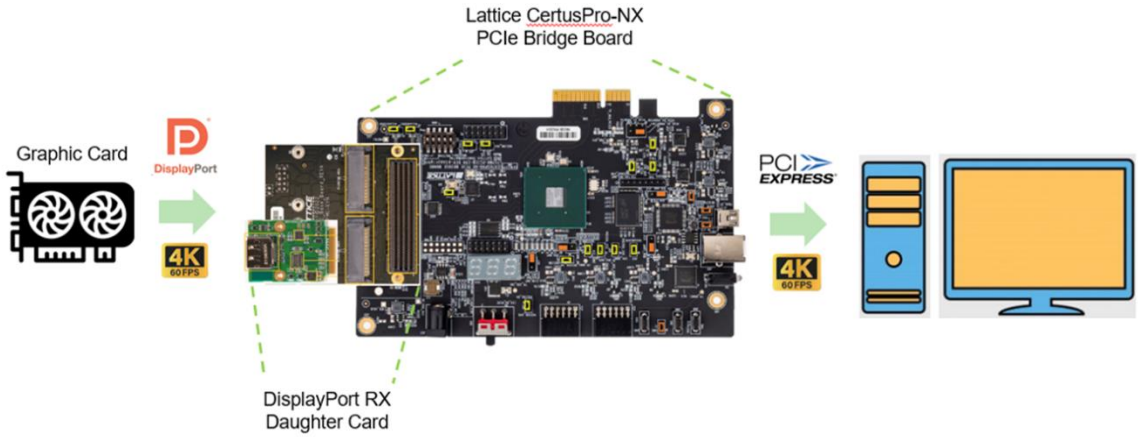

Figure 4.1. DP-to-PCIe Hardware Setup

4.1.1. CertusPro-NX PCIe Bridge Board RevC

The CertusPro-NX PCIe Bridge Board has no jumpers to configure out of the default configuration. To ensure PCIe x4 is configured, jumper J4 is shorted at 2–4 by default. The card is plugged into the PCIe slot, external power is provided by the system, and the 12 V source switch receives power from the PCIe slot. Video source such as laptop that runs the Lattice Radiant software is connected to the board using the mini USB Type A cable.

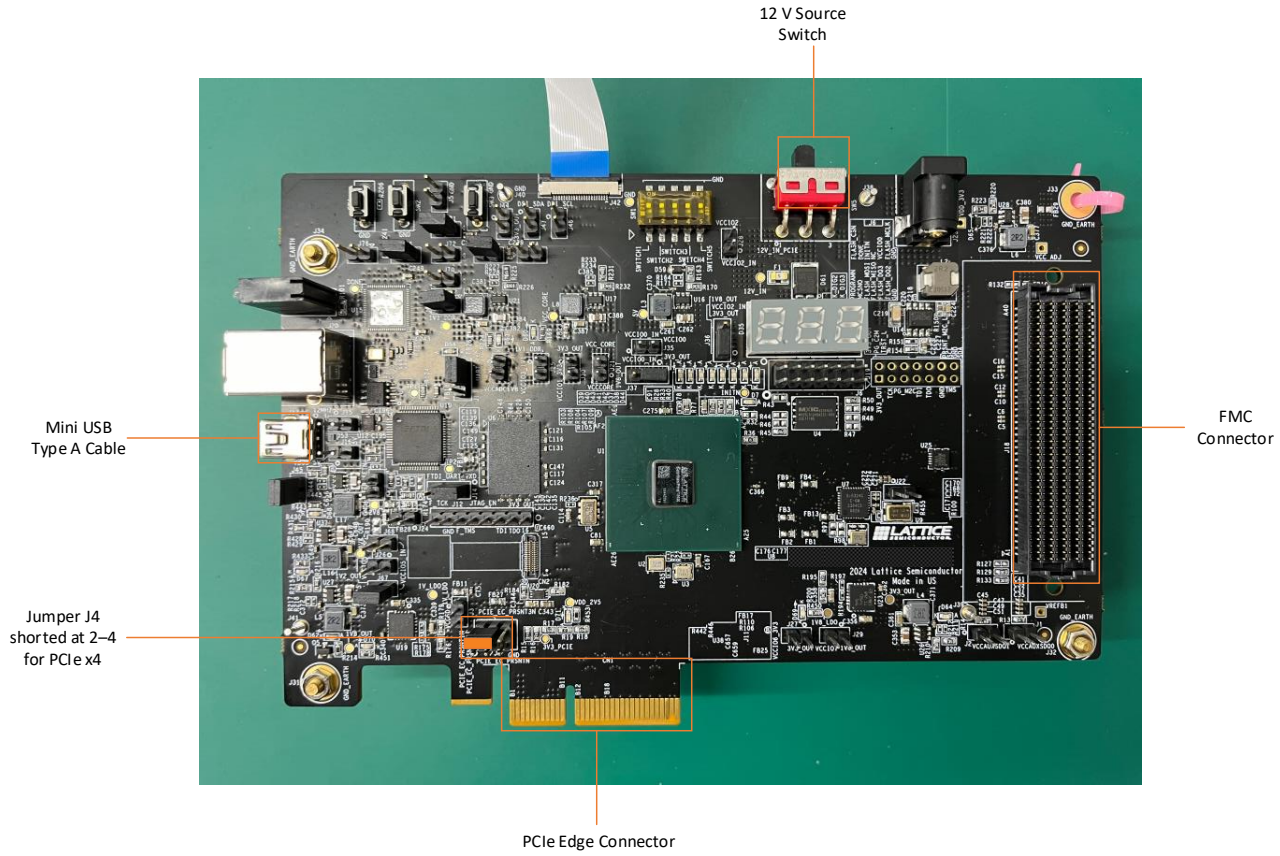

Figure 4.2. CertusPro-NX PCIe Bridge Board RevC

4.1.2. Modular FMC Adapter and Daughter Card

The modular FMC adapter power is provided from the FMC connector (J1). The DisplayPort receiver daughter card connects to the modular FMC adapter through the J2 connector.

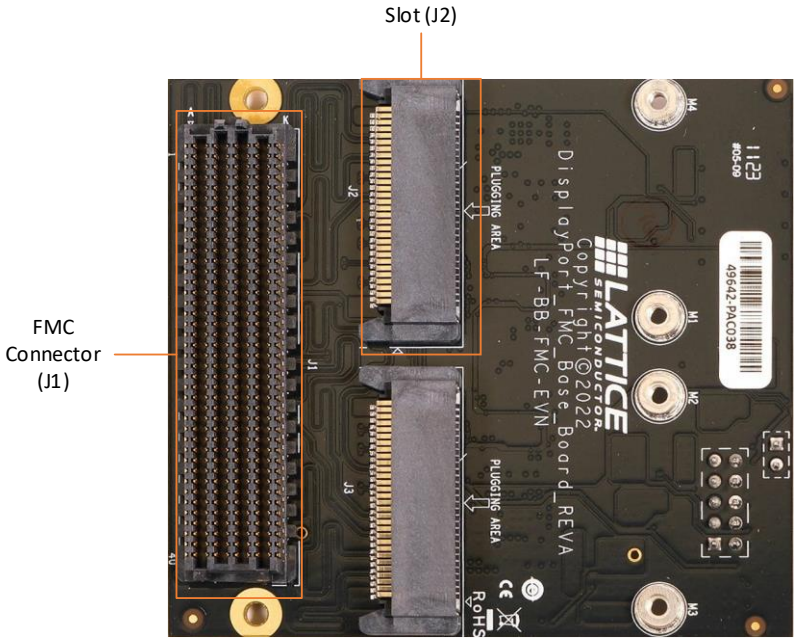

Figure 4.3. Modular FMC Adapter

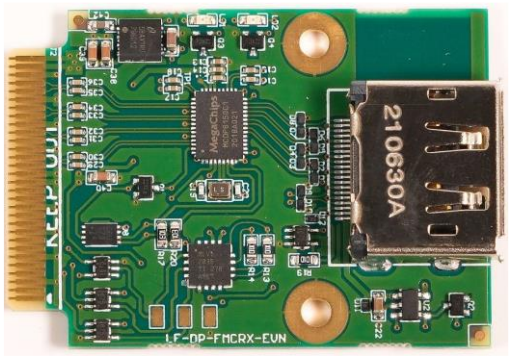

Figure 4.4. DisplayPort Receiver Daughter Card

4.1.3. Host PC

The host PC executes the demo software. To set up the system, ensure the PCIe slot in the host PC supports PCIe x4. Insert the CertusPro-NX PCIe Bridge Board with modular FMC adapter and DisplayPort RX daughter card into the PCIe slot. Connect the DisplayPort RX daughter card to the on-board DisplayPort connector of the motherboard by using the DisplayPort cable. Refer to the Software Setup section for the details of software setup.

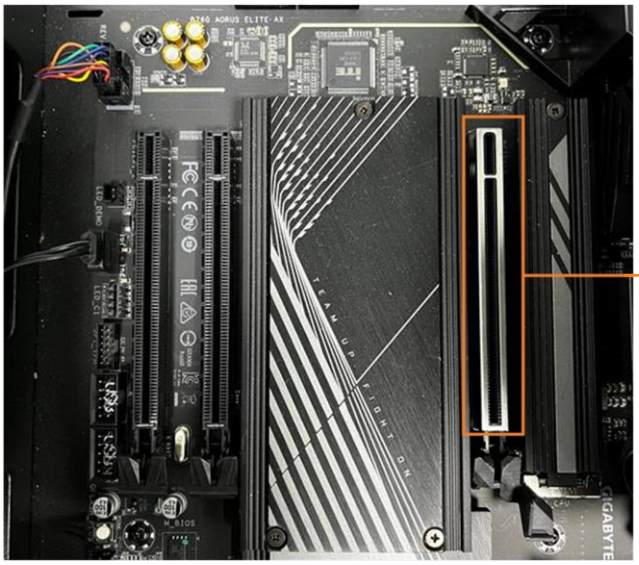

Figure 4.5. Sample PCIe Slot that Supports Up to PCIe x16

4.2. Programming the FPGA

To program the FPGA, ensure the board is inserted into the host PC, then follow these steps:

1. Create a new project using the Lattice Radiant Programmer software. In the **Getting Started** dialog box, input the **Project Name** and **Location** as shown in Figure 4.6.
2. Select **Create a new project from a scan**. The values are indicated in the **Cable**, **Port**, and **TCK Divider Setting (0-30×)** fields.
3. Click **OK**.

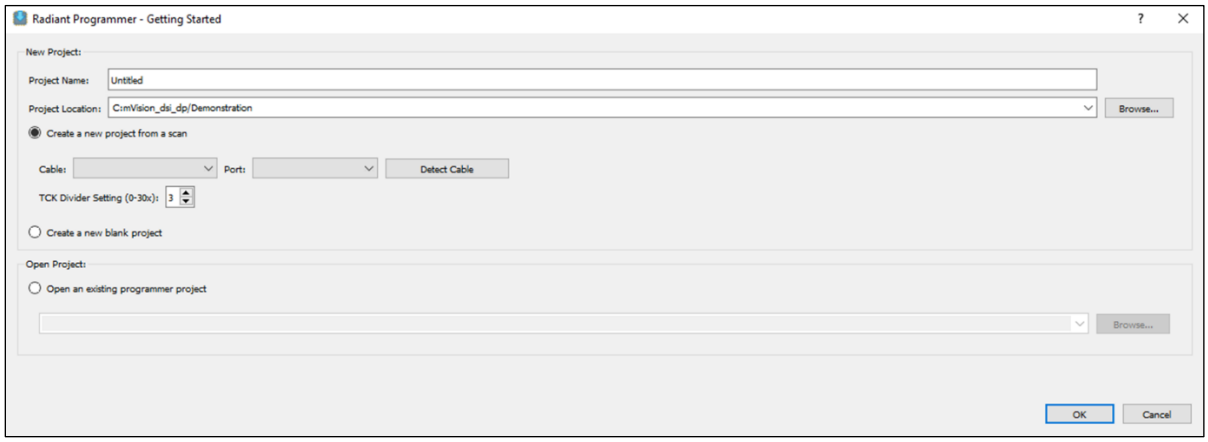

Figure 4.6. Creating a New Project from a Scan

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

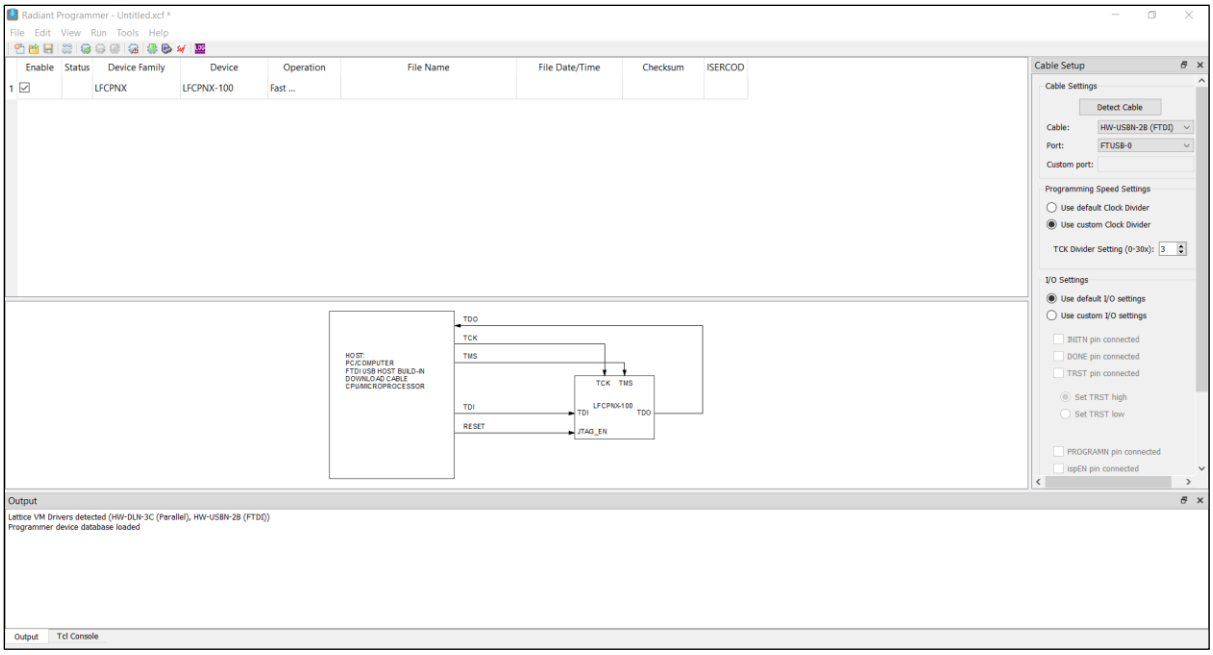

Figure 4.7. Lattice Radiant Programmer Window

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

Figure 4.8. CertusPro-NX Device Settings

To select the programming settings, follow these steps:

1. Browse and select the Programming file, mvision_dp_pcie.bit.
2. Click **OK**.
3. Double-click **Operation** to open the **Device Properties** dialog box.
4. Select the settings as shown in the figure below.

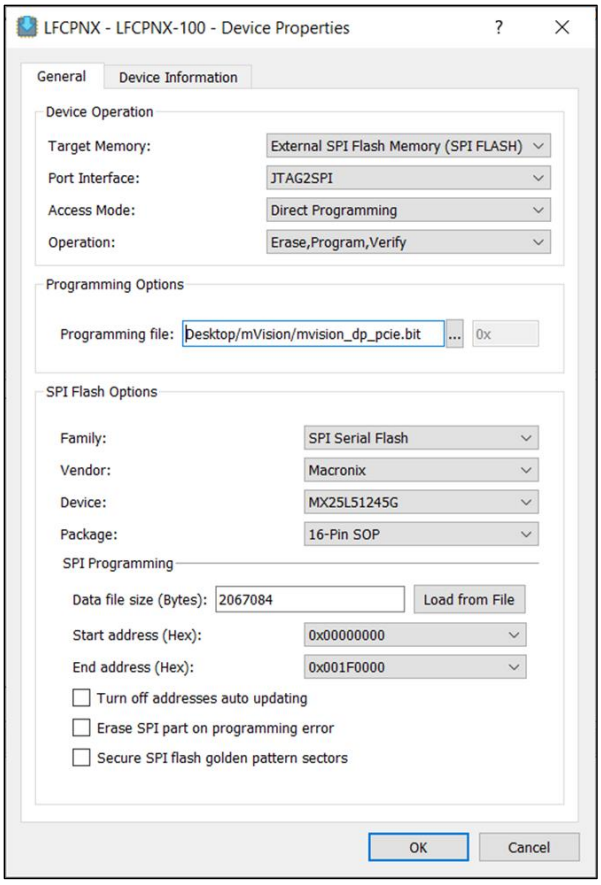

Figure 4.9. Device Properties Window for the CertusPro-NX SPI Flash Programming

5. Click **Programming** from the menu bar to start programming.

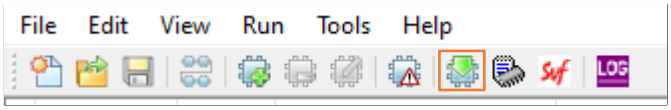

Figure 4.10. Programmer Menu Bar

When the FPGA programming is successful, the output console displays the *Operation: successful* message.

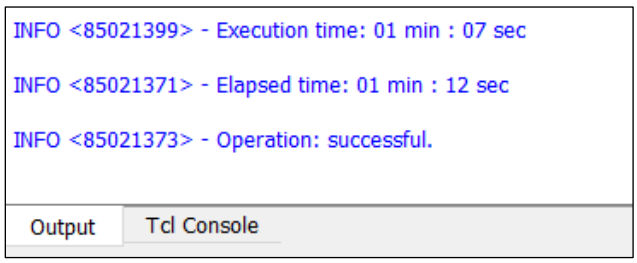

Figure 4.11. Programmer Output Window

If the programming operation is unsuccessful, refer to the Troubleshooting section.

6. After programming, check the status LEDs on the board, power cycle the board, and reboot the host PC.

4.2.1. Status LED

Figure 4.12. CertusPro-NX Status LEDs

Table 4.1. Description of the Status LED

Sl. No	Name	Description
1	DONE	Green – Lights if configuration is successful.

4.3. Software Setup

To set up the software for this demo, follow these steps:

1. Create a directory and copy all software related folders from the demo .zip file into this directory. For this demo, the folder name is *mVision_VDMA_SW_RGB_DEMO*.
2. In Linux, open a terminal to the directory where the example application folder is located.
3. For the first boot, in a Linux terminal, run the following command and look for *Lattice Semiconductor Corporation Device 0001 (rev 01)*.
   ```
   lspci -v
   ```
4. Run the following command to go into the script folder.
   ```
   cd ./mVision_VDMA_SW_RGB_DEMO/scripts/
   ```
5. If you run the application for the first time or after making changes to the lscvdma code, use the following command:
   ```
   sudo sh run-all.sh compile
   ```
   This command compiles and builds the lscvdma module, loads the module, and runs the demo if the module is loaded successfully.

Note: When *sudo* command is entered, enter *admin-ubuntu* when prompted for password.

6. To run the application without recompiling the module, use the following command:

```
sudo sh run-all.sh
```

7. If the software is set up successfully, the message is shown in the figure below.

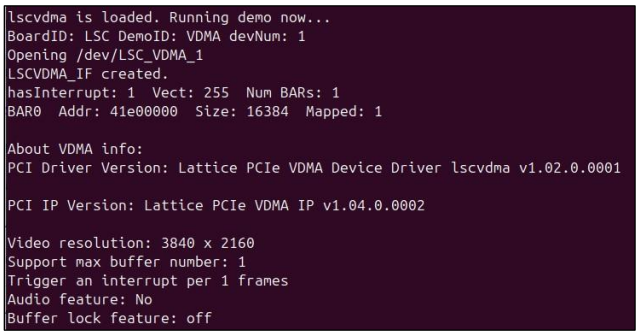

Figure 4.13. Software Running Message

For issues on software setup, refer to the Troubleshooting section.

5. Software GUI Overview

The software graphical user interface (GUI) provides the following display information:

- Resolution
- Display frames per second (FPS)
- Display rate
- PCIe FPS
- PCIe rate
- Lost frames and total frames

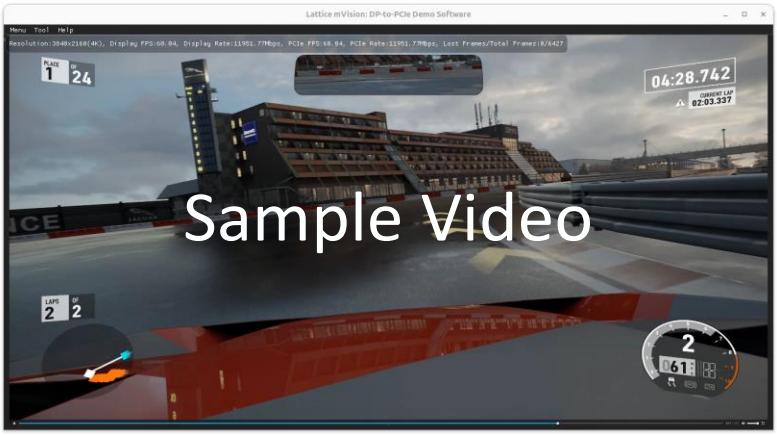

Figure 5.1. Demo Software GUI Displaying a Sample Video

6. Troubleshooting

6.1. Error when Updating SPI Flash

- If there is verification error when programming the .bit file, try changing the TCK frequency by setting **TCK Divider Setting** to greater than 3. Restart programming by clicking the **Program** button.

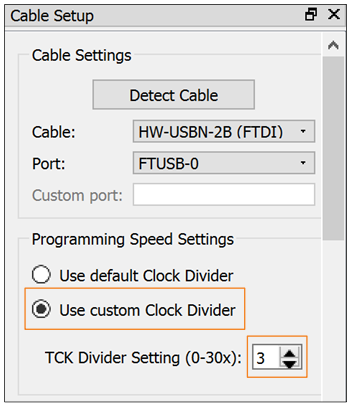

Figure 6.1. TCK Frequency Settings

If the verification error problem still occurs, press and hold the **PROGRAMN** push button before clicking the **Program** button.

- If the device is not detected on Port: **FTUSB-0**, click **Detect Cable** and select Port: **FTUSB-1**.

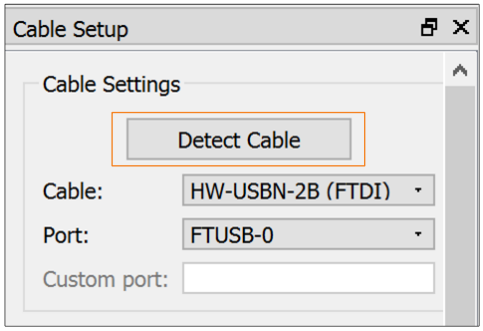

Figure 6.2. Port Selection

6.2. Issues in Running the Automated Scripts

6.2.1. Driver Module Insertion Error and Missing Device File

```
 existing lscvdma...
lscvdma is removed.
insmod: ERROR: could not insert module ../drvr/lscvdma.ko: Operation not permitt
ed
Unable to load lscvdma.
```

Figure 6.3. Driver Module Insertion Error Message

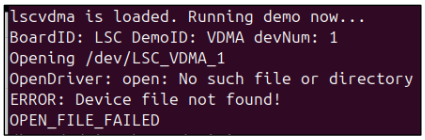

Figure 6.4. Missing Device File Message

To fix the driver module insertion error and to detect the missing device file, follow these steps:

1. Ensure that the FPGA board is connected properly to the host PC.
2. Run *lspci* to check the advanced technology attachment (ATA) controller. Ensure the controller is *Lattice Semiconductor Corporation Device 0001 (rev 01)*.
3. Reboot the host PC.

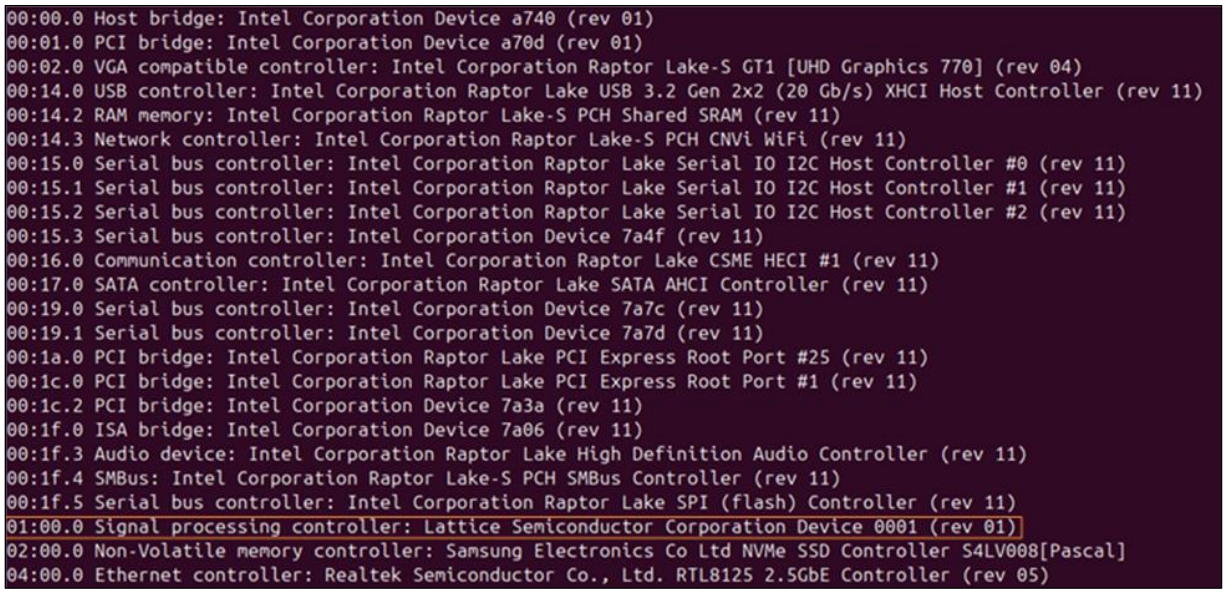

Figure 6.5. Running the lspci Command

6.2.2. Software GUI Displaying a Black Screen

To fix the black screen error, follow these steps:

1. Go to **Settings > Displays > Resolution**.
2. Ensure that the video source resolution and refresh rate are set to 3840x2160 and 60.00 Hz respectively.

Figure 6.6. Software GUI Displaying a Black Screen

```
 existing lscvdma...
lscvdma is removed.
lscvdma is loaded. Running demo now...
BoardID: LSC DemoID: VDMA devNum: 1
Opening /dev/LSC_VDMA_1
LSCVDMA_IF created.
hasInterrupt: 1  Vect: 255  Num BARs: 1
BAR0  Addr: 41e00000  Size: 16384  Mapped: 1

About VDMA info:
PCI Driver Version: Lattice PCIe VDMA Device Driver lscvdma v1.02.0.0001

PCI IP Version: Lattice PCIe VDMA IP v1.04.0.0002

Video resolution: 3840 x 2160
Support max buffer number: 1
Trigger an interrupt per 1 frames
Audio feature: No
Buffer lock feature: off
frame index not update, output black screen.
PCI IP Version:
Lattice PCIe VDMA IP v1.04.0.0002
```

Figure 6.7. Software GUI Black Screen Message

Appendix A. Resource Utilization

The table below shows a sample resource utilization of the CertusPro-NX DP to PCIe bridge demo design.

Table A.1. Resource Utilization of the CertusPro-NX DP to PCIe Bridge Demo Design

DP to PCIe Bridge Demo Design	LUT4	Registers	EBR	DSP
Total	15,936	19,969	37	30

References

- CertusPro-NX web page
- CertusPro-NX PCIe Bridge Board web page
- Modular FMC Adapter and DisplayPort Daughter Cards web page
- Lattice Radiant Software web page
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- 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 1.1, March 2025

Section	Change Summary
Hardware and Software Requirement	• Updated the host PC requirements in the Hardware Requirements section. • Updated the Software Requirements section.
Setting Up the Demo	• Added description about PCIe x4 configuration using Jumper J4 in the CertusPro-NX PCIe Bridge Board RevC section. • Updated Figure 4.2. CertusPro-NX PCIe Bridge Board RevC. • Updated the Host PC section. • Added a prerequisite before programming the FPGA in the Programming the FPGA section.

Revision 1.0, December 2024

Section	Change Summary
All	Production release.

www.latticesemi.com