Lattice mVision MIPI Video Sensor to PCIe Bridge Demo

User Guide

FPGA-UG-02168-1.1

June 2023

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

Glossary 4
1. Introduction 5
2. Demo Design Overview 6
2.1. FPGA Design Description 6
2.2. DMA Architecture 6
3. CertusPro-NX Versa Board 8
3.1. Hardware Description 8
3.2. Jumper Settings 8
4. Programming the CertusPro-NX Device 9
5. Installing Driver and Running Demo on Linux OS 12
References 13
Technical Support Assistance 14
Revision History 15

Figures

Figure 1.1. CertusPro-NX Versa Board 5
Figure 2.1. FPGA Design Block Diagram 6
Figure 2.2. Descriptor Usage with the PCIe DMA System Architecture 7
Figure 4.1. Getting Started Dialog in Lattice Radiant Programmer 9
Figure 4.2. Radiant Programmer Main Window 9
Figure 4.3. Device Properties Dialog 10
Figure 4.4. Programmer Output Window 11
Figure 5.1. CSI-2 to PCIe Bridge 12

Tables

Table 3.1. CertusPro-NX Versa Board Jumper Settings 8

Glossary

A list of terms used in this document.

Term	Definition
C2H	Card to Host, also named as Peripheral-to-Host (P2H) or Card-to-System (C2S)
CSI	Camera Serial Interface
Descriptor	Defines some control bits and the address and length of the allocated target CPU memory.
DMA	Direct Memory Access
FPGA	Field-Programmable Gate Array
H2C	Host to Card
ISP	Image Signal Processing
MIPI	Mobile Industry Processor Interface
OS	Operating System
PCIe	Peripheral Component Interconnect Express
SG-DMA	Scatter-Gather DMA
SG-Element	Descriptor
SG-List	A series of descriptors for single frame buffer in host memory
S2C	System to Card
TLP	Transaction Layer Packet

1. Introduction

This document provides technical information and instructions of the CertusPro™-NX MIPI CSI-2 to PCIe Bridge demo. This design demonstrates the functionality of transferring MIPI CSI-2 sensor video data to a computer through PCIe with a Direct Memory Access (DMA) engine.

This demo is based on the CertusPro-NX Versa Board with Linux Operating System (OS) driver support. The demo shows transfer of sensor data to the computer memory and rendering of the data as video on the computer screen using the software driver.

The demo package includes the following files:

- CertusPro-NX bitstream (.bit)
- Linux driver source code
- Linux application software source code

Demo design hardware requirements:

- CertusPro-NX Versa Board (Figure 1.1)
- Sony IMX258 sensor module
- Computer with Linux OS
- USB cable for programing the CertusPro-NX device

Demo design software requirements:

- Radiant 3.0 (or later version) software

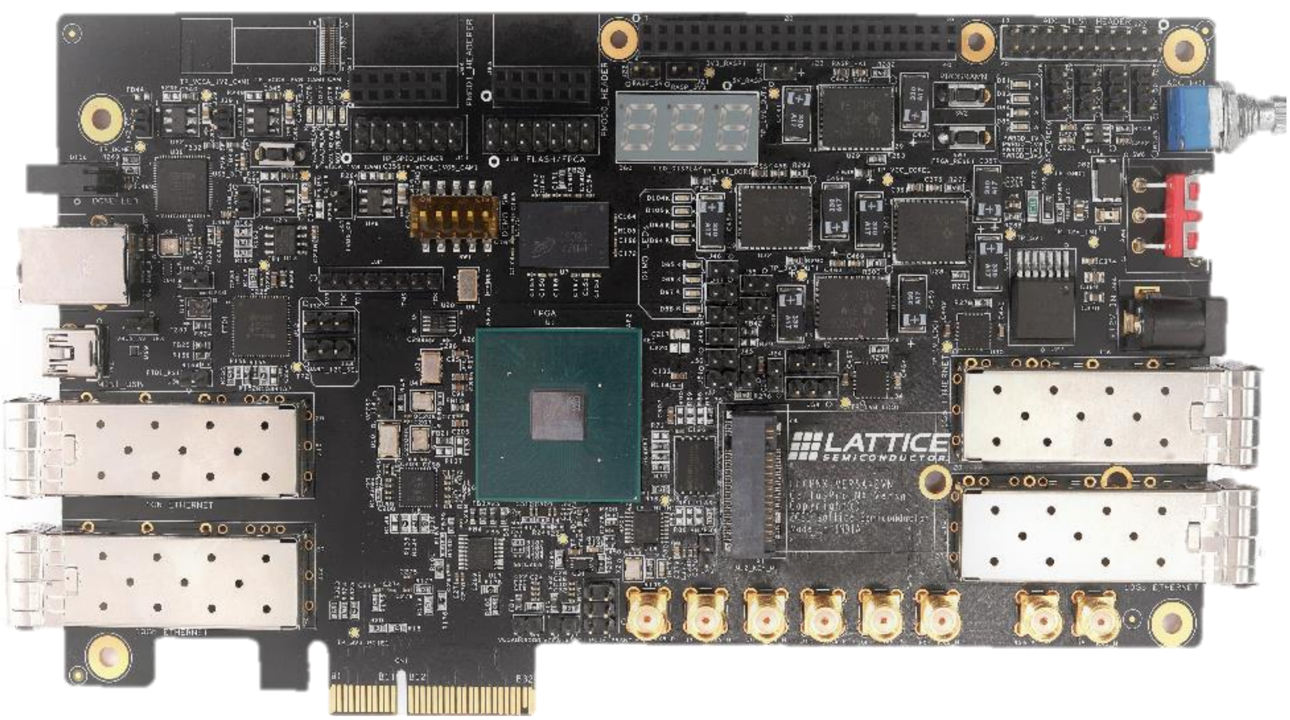

Figure 1.1. CertusPro-NX Versa Board

2. Demo Design Overview

2.1. FPGA Design Description

Figure 2.1 shows the block diagram of the FPGA design. VDMA for PCIe sub-system is used in this design. This VDMA for PCIe block provides high-performance DMA data transmission between PCIe and the local native video interface. The High-performance data movement between the host memory and FPGA local memory is achieved using the PCIe link and the driver software running on the host.

The PCIe hard IP, PCIe TLP Decoding and Encoding, PCIe Flow Control Management, TLP Transmitter Arbiter, PCIe Reset Sequence, and VDMA Controller are wrapped into this VDMA for PCIe sub-system (Figure 2.1). The video data format from the IMX258 sensor is RAW8 (1080p50) and the ISP module is used to convert RAW to RGB, and then to YUV422.

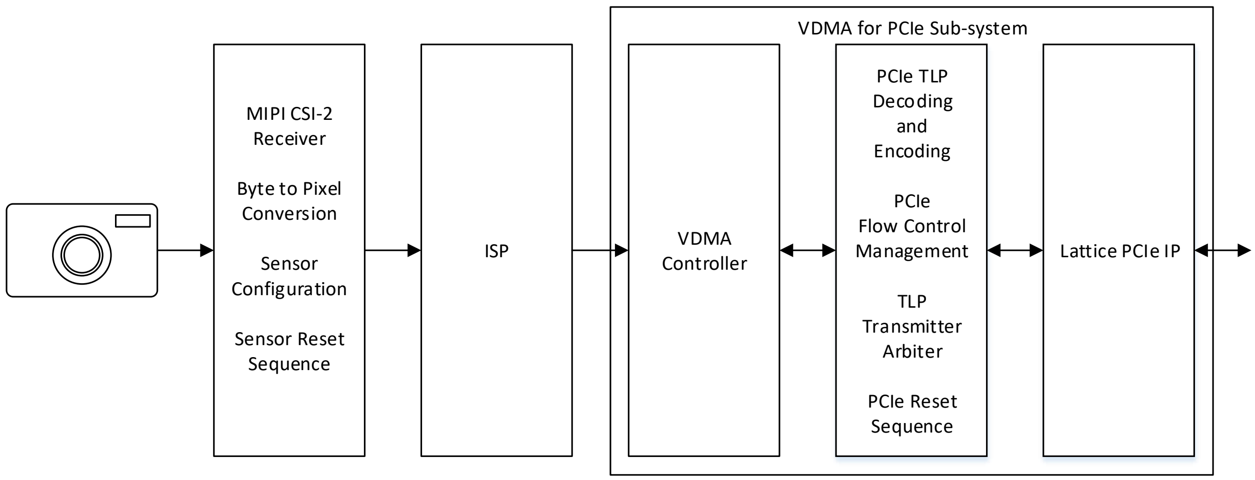

Figure 2.1. FPGA Design Block Diagram

2.2. DMA Architecture

At the most basic level, the PCIe DMA engine typically moves data between the host memory and the memory that resides in the FPGA which is often but not always on an add-in card. When data is moved from the host memory to the FPGA memory, it is called Host-to-Card (H2C) transfer or System-to-Card (S2C) transfer. Conversely, when data is moved from the FPGA memory to the host memory, it is called a Card to Host (C2H) or Card to System (C2S) transfer.

These two terms (H2C and C2H) help delineate which way the data is flowing, as opposed to using read and write, which may be confusing.

In a typical operation, the application software in the host needs to move data from or to the FPGA PCIe Endpoint. To accomplish this transfer, the software in the host sets up buffer space in the host system memory and creates descriptors, which the DMA engine uses to move the data.

It is quite difficult to request the OS to allocate a large contiguous (in physical address) memory for the buffer that is used by DMA transmission. Scatter-Gather DMA (SG-DMA) provides data transfer from one non-contiguous block of memory to another by means of a series of smaller contiguous-block transfer. This SG-DMA technique reduces the processor load by reducing the number of interrupts it needs to handle.

The VDMA block implements the SG-DMA engine transfer and merges non-contiguous memory to a continuous address space, and vice versa. A series of descriptors are used to specify the data to be transferred. The series of descriptors, for single frame buffer in host memory, is named as SG-List, whereas a single descriptor is also named as SG-Element. The size of the SG-List is the number of the descriptor (SG-Element) for this frame buffer.

Figure 2.2 shows the descriptor usage with the PCIe DMA system architecture.

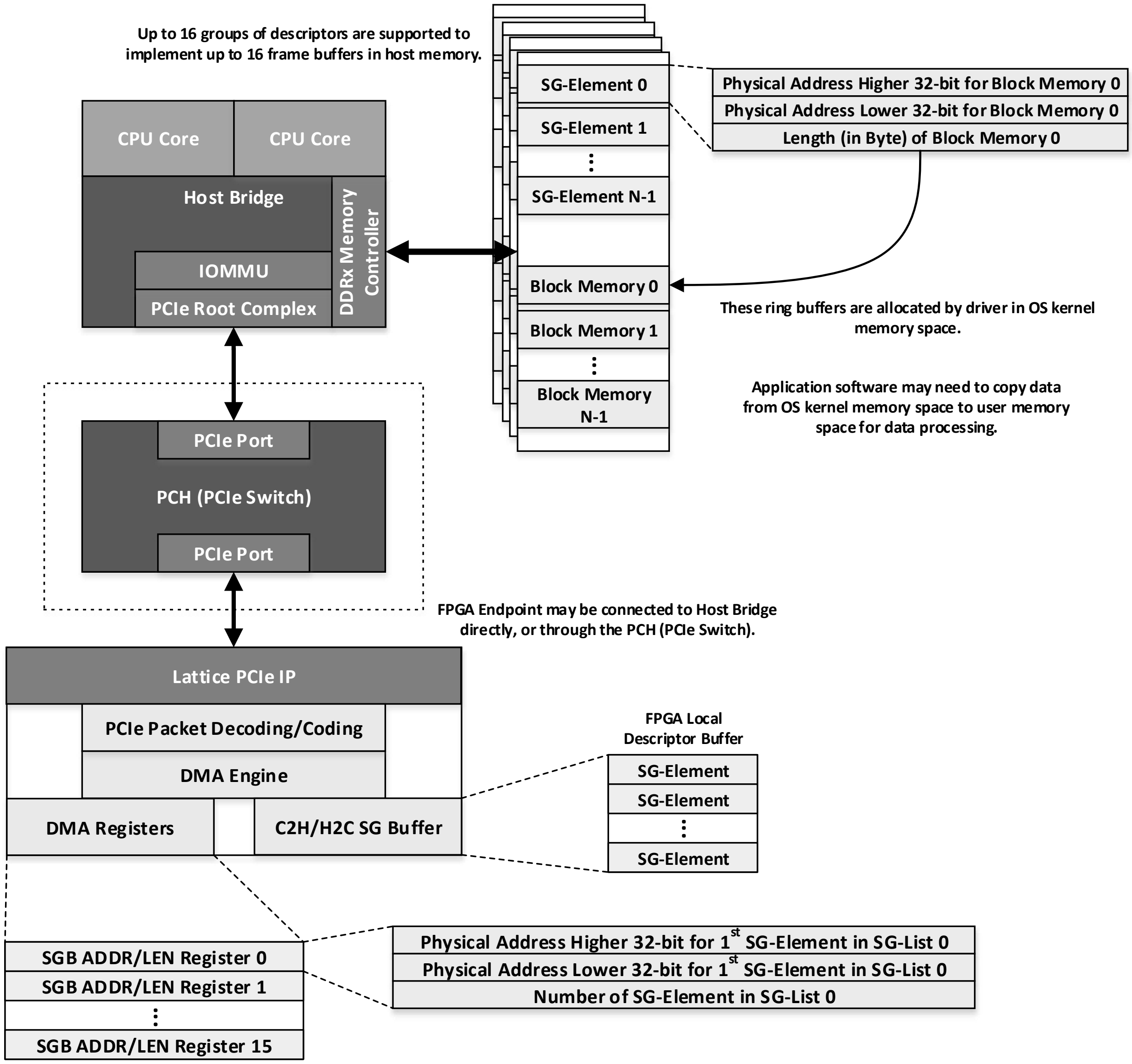

Figure 2.2. Descriptor Usage with the PCIe DMA System Architecture

3. CertusPro-NX Versa Board

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

3.1. Hardware Description

The CertusPro-NX Versa Board has an on-board sensor connector, J37, at the top left corner. The Sony IMX258 sensor module should be connected to J37.

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

3.2. Jumper Settings

The following jumper settings of the CertusPro-NX Versa Board (Table 3.1) are required to enable the CSI-2 to PCIe Bridge demo.

Table 3.1. CertusPro-NX Versa Board Jumper Settings

Jumper Name	Settings	Description
J1	Connect 2-4	Using PCIe x4 mode
SW6	Switch to UP	Using 12V_PCIe_CONN power

4. Programming the CertusPro-NX Device

The following steps show how to download the bitstream to the CrossLink-NX device on the CrossLink-NX PCIe Bridge Board, or CertusPro-NX device on the CertusPro-NX Versa Board:

1. Connect the board to the PC through the USB mini port.
2. Connect the board to the DC power adapter (12 V).
3. Select 12 V DC power.
4. Start Radiant Programmer, version 3.0 or later version, from the computer.
5. The **Radiant Programmer – Getting Started** dialog box opens (Figure 4.1). By default, the **Create a new project from a JTAG scan** option is selected. Check and confirm if other settings are correctly selected and if the required components are connected.
6. Click **OK**.

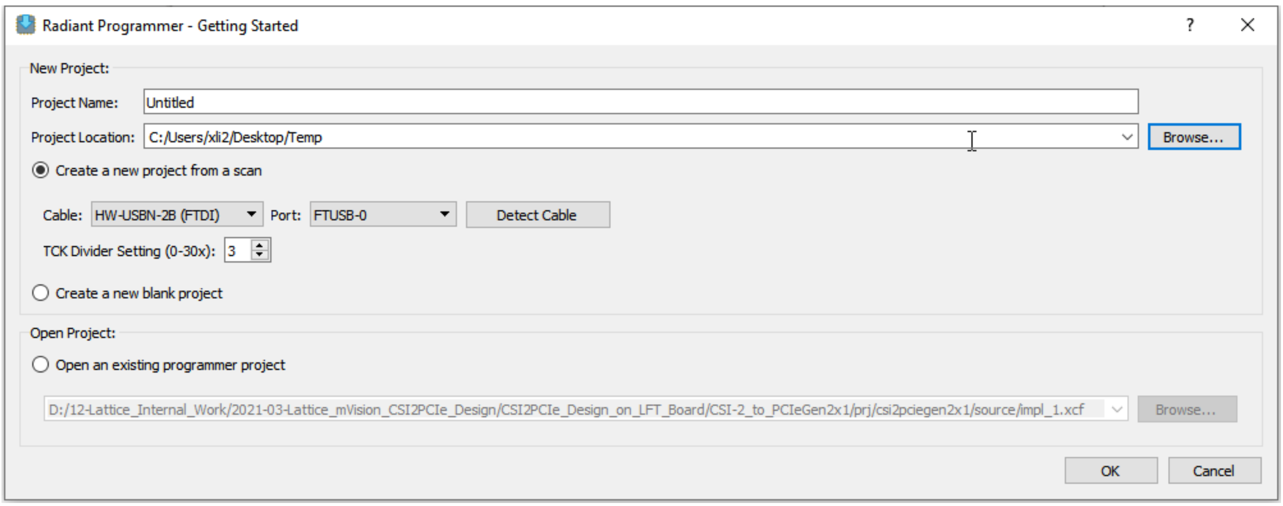

Figure 4.1. Getting Started Dialog in Lattice Radiant Programmer

7. The main interface opens, as shown in Figure 4.2. Double-click **Fast Configuration** under **Operation**.

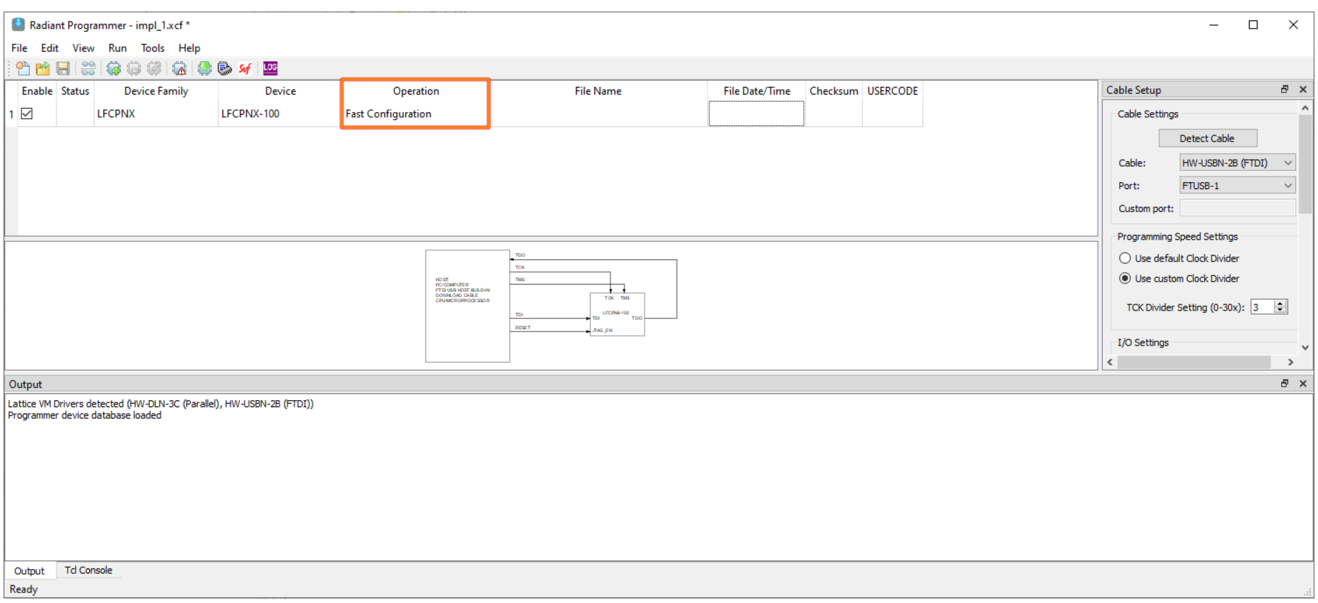

Figure 4.2. Radiant Programmer Main Window

8. The **Device Properties** dialog box opens. Changed the settings as shown in Figure 4.3.
9. Click **OK**.

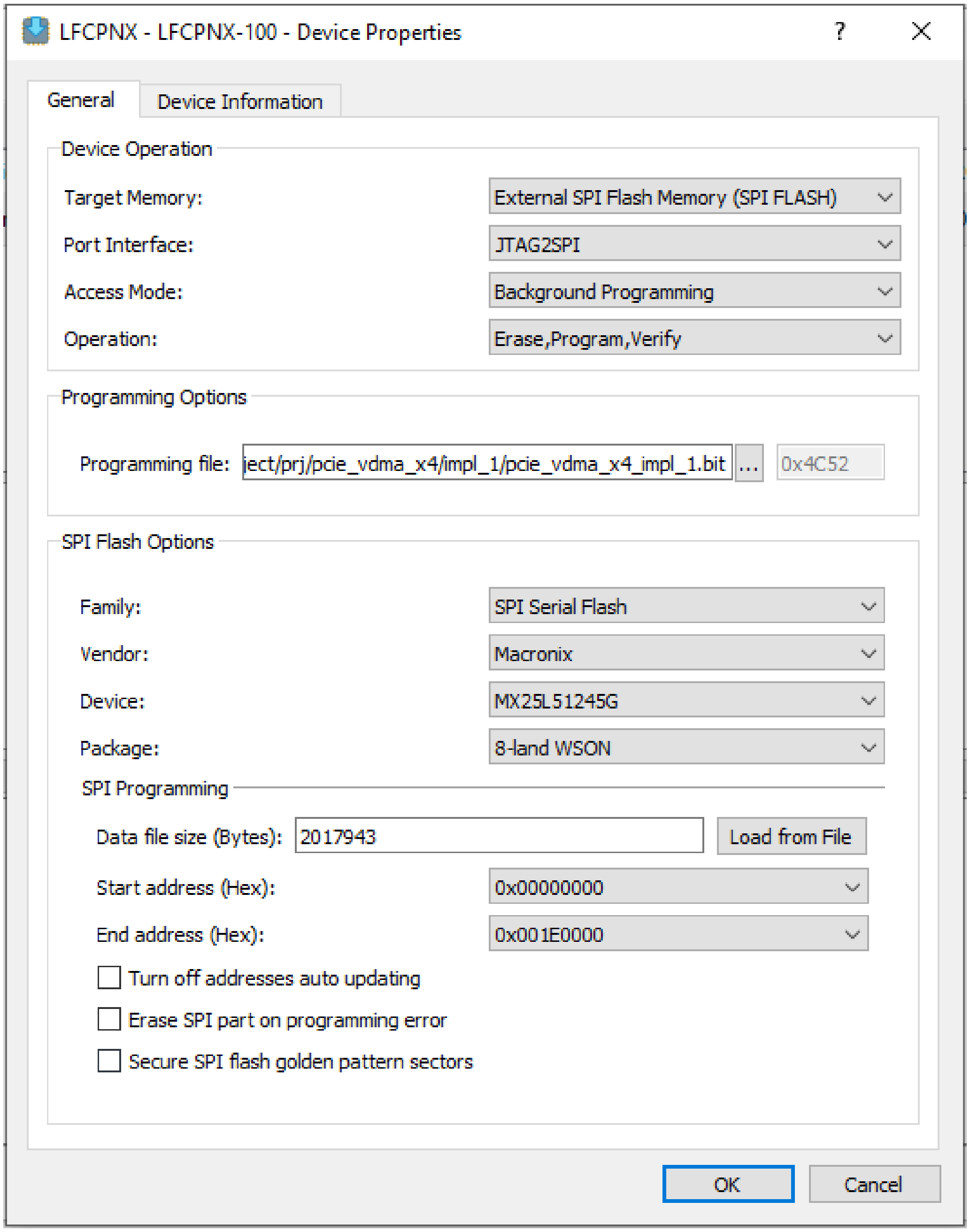

Figure 4.3. Device Properties Dialog

10. Click **Program**  from the toolbar, or choose the **Design > Program** from the Radiant Programmer, to program the CrossLink-NX/CertusPro-NX board. Wait for the programming to finish.

11. Check the programming status and result from the output pane, as shown in Figure 4.4.

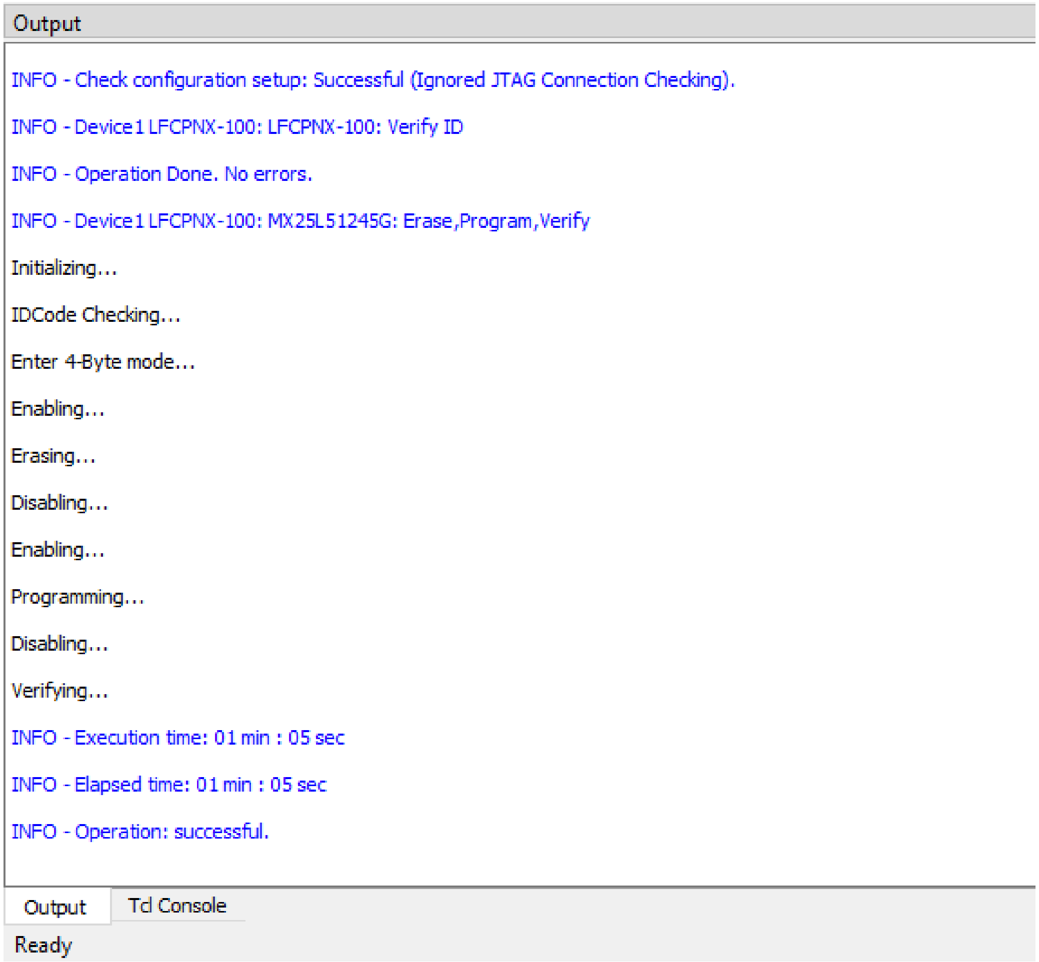

Figure 4.4. Programmer Output Window

12. Select 12 V power from PCIe edge connector.

5. Installing Driver and Running Demo on Linux OS

This section provides the steps and instructions for installing drivers and running the demo on a Linux-based target computer. It is recommended to use Ubuntu 20.04.4 LTS 64-bit as the operating system.

The device drivers must correspond to the kernel version. The provided installation script allows the user to compile the driver for the installed kernel.

Before compiling the source code of driver and application software, install the following packages:

- make
- gcc
- g++
- libglfw3-dev
- mesa-utils
- libglew-dev
- libasound2-dev

To install drivers on Linux OS, follow the steps:

1. Insert the board into the PCIE Slot.

Note: Make sure that the secure boot is disabled before booting the PC.

2. Boot the PC into Linux.

Note: Make sure that the LED D68 on the board is not lit up. If it is lit up, this indicates PCIE Link is down. Restart the PC.

3. Open the terminal and locate the *lscvdma* folder.
4. Input `sudo chmod 777 install.sh`
5. Make sure that the secure boot is disabled as it interferes with the process of loading the kernel module of the demo.

   To do this on ubuntu, use command: `$ sudo mokutil --sb-state`. This command should output if the secure boot is enabled/disabled. If enabled, disable this in the BIOS UEFI settings.
6. Input `sudo ./install.sh`. This command generates the .ko files and insert the drivers. It also opens the GUI.

Notes:

1. Every time the pc is restarted; run `$sudo ./install.sh`.
2. If encounter an issue with `install.sh`, restart the system.

The driver of the requested Linux OS is ready for the application software and the Sony IMX258 sensor data is transferred to computer memory through PCIe. Figure 5.1 shows the data on screen.

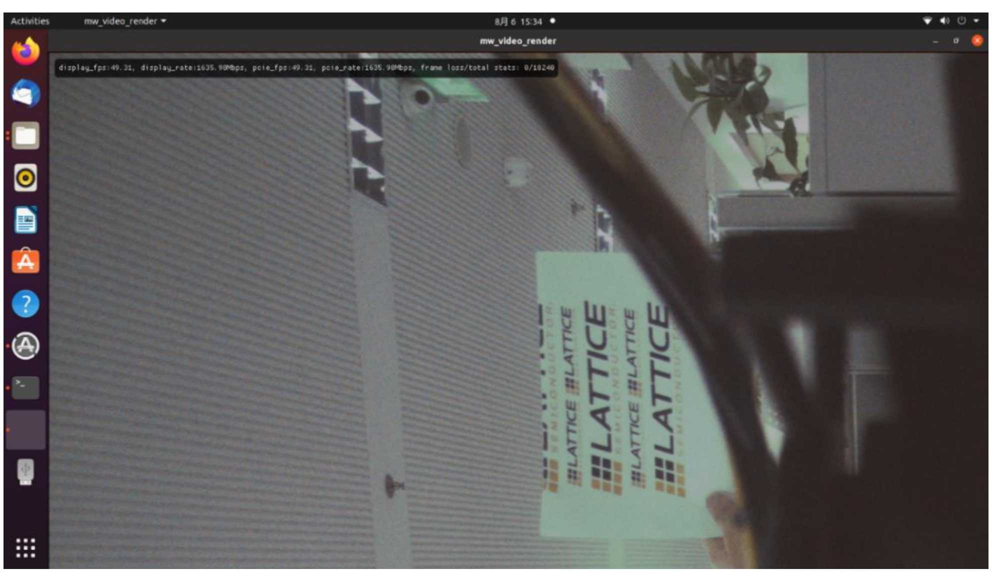

Figure 5.1. CSI-2 to PCIe Bridge

References

- 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.

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

Revision History

Revision 1.1, June 2023

Section	Change Summary
Installing Driver and Running Demo on Linux OS	Updated the steps for installing drivers on Linux OS.
Technical Support Assistance	Added reference link to the Lattice Answer Database.

Revision 1.0, September 2022

Section	Change Summary
All	Initial release.

LATTICE SEMICONDUCTOR®

www.latticesemi.com