DG0532
Demo Guide
IGLOO2 FPGA PCIe Control Plane with Device Serial Number - Libero SoC v11.8 SP1

Microsemi®

Power Matters.™

Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 7.0

Updated the document for Libero v11.8 SP1 software release.

1.2 Revision 6.0

Updated the document for Libero v11.7 software release.

1.3 Revision 5.0

Updated the document for Libero v11.6 software release.

1.4 Revision 4.0

Updated the document for Libero v11.5 software release.

1.5 Revision 3.0

Updated the design files links.

1.6 Revision 2.0

Updated the document for Libero v11.4 software release.

1.7 Revision 1.0

Revision 1.0 was the first publication of this document.

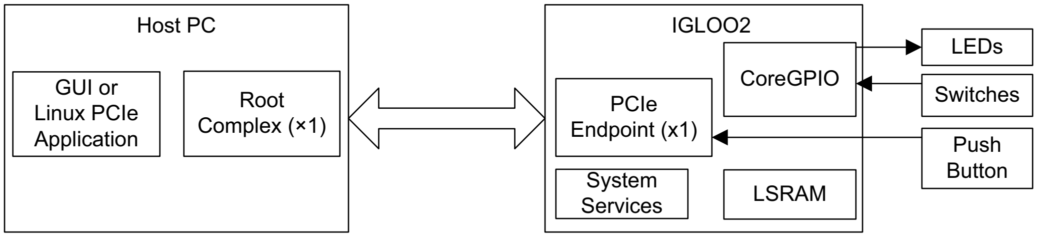
2 IGLOO2 FPGA PCIe Control Plane with Device Serial Number Demo

The IGLOO2 FPGA devices integrate a fourth-generation flash-based FPGA fabric and high-performance communication interfaces on a single chip. The IGLOO2 high-speed serial interface provides a fully integrated peripheral component interface express (PCIe) endpoint implementation and is compliant with the PCIe Base Specification Revision 2.0 and 1.1. See the *UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide* for more information.

This demo shows how the embedded PCIe feature of the IGLOO2 FPGA devices can be used as a low bandwidth control plane interface. It also demonstrates device serial number (DSN) feature embedded in the IGLOO2 device. A sample design is provided to access IGLOO2 PCIe endpoint from the host PC. It runs on both windows and RedHat Linux operating systems (OS). A GUI installer, host PC drivers for Windows OS, and a Linux PCIe application for Linux OS are provided for reading and writing to the IGLOO2 PCIe configuration and memory space.

The following figure shows the top-level block diagram of the PCIe control plane demo. The demo design uses an IGLOO2 PCIe interface with a link width of ×1 lane to interface with a host PC PCIe Gen2 slot. The CoreGPIO IP controls the LEDs and switches on the IGLOO2 Evaluation Kit board through the PCIe interface. The host PC can read and write to the IGLOO2 large SRAM (LSRAM), and can also be interrupted by using the push button on the IGLOO2 Evaluation Kit board. It can read the 128-bit DSN system service.

Figure 1 • **PCIe Control Plane Demo Top-Level Block Diagram**



2.1 Design Requirements

The following table lists the hardware, software, and IP requirements for this demo design.

Table 1 • **Design Requirements**

Design Requirements	**Description**
Hardware Requirements	
IGLOO2 Evaluation Kit[1]: – 12 V adapter – FlashPro4 programmer – USB A to Mini-B cable	Rev D or later
Host PC or Laptop with an available PCIe 2.0 Gen 1 or Gen 2 compliant slot	64-bit Windows 7 OS or 64-bit RedHat Linux OS (Kernel Version: 2.6.18-308)
Express Card slot and PCIe Express card adapter (for Laptop only)	–

Table 1 • Design Requirements (continued)

Design Requirements	Description
Software Requirements	
Libero® SoC Design Suite for viewing the design files	v11.8 SP1
FlashPro Programming Software	v11.8 SP1
Host PC Drivers (provided along with the design files)	–
GUI executable (provided along with the design files)	–

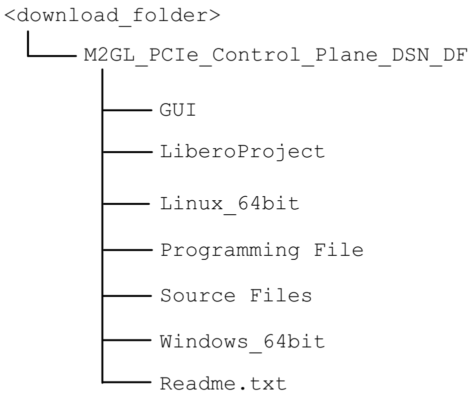
1. The PCIe Express card adapter is not supplied with the IGLOO2 Evaluation Kit

2.2 Demo Design

The design files for this demo can be downloaded from the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2gl_dg0532_liberov11p8_sp1_df*

The following figure shows the top-level structure of the design files. For more information, see the `readme.txt` file.

Figure 2 • Demo Design Files Top-Level Structure



2.2.1 Features

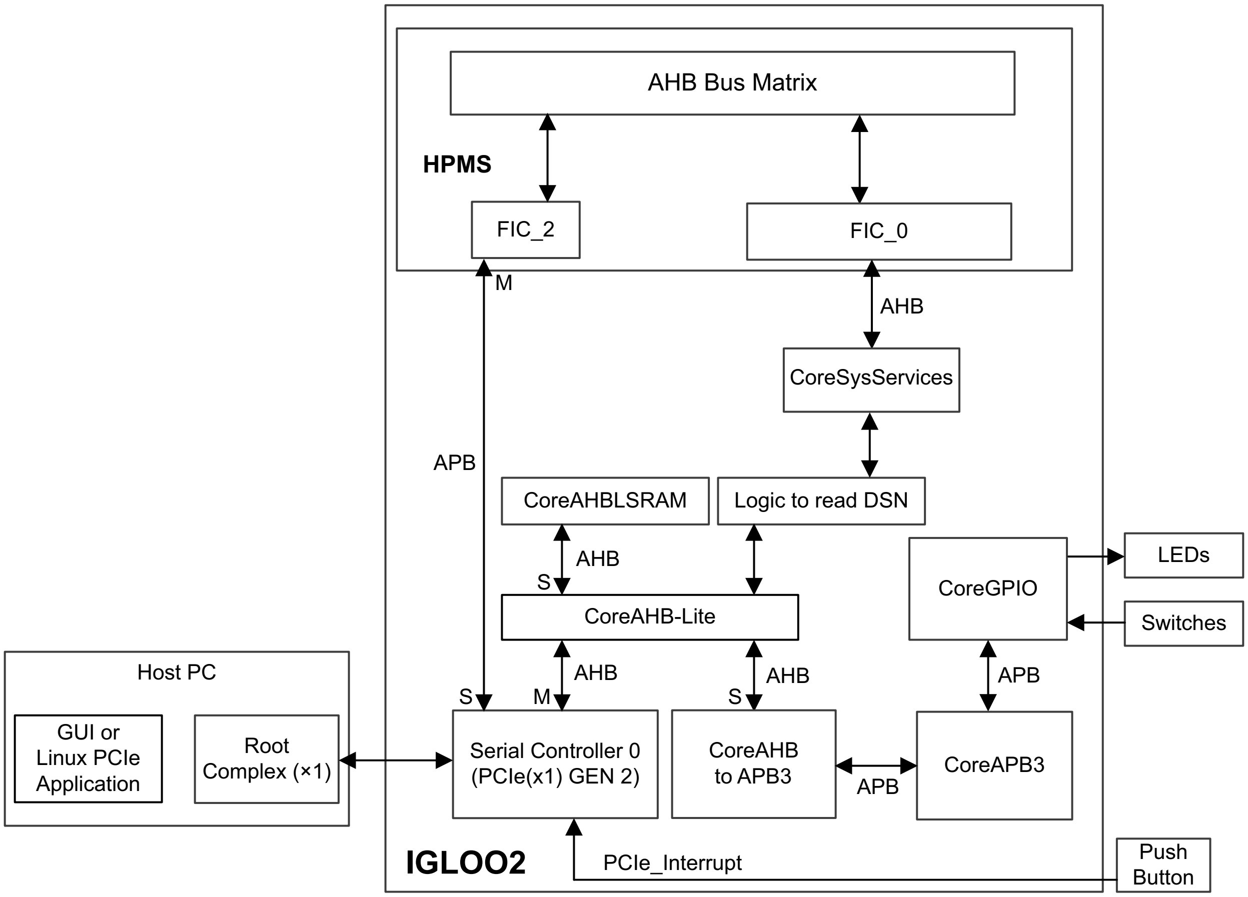
The demo design performs the following tasks:

- Displays PCIe link enable/disable, negotiated link width, and the link speed
- Controls the status of light emitting diodes (LEDs) on the IGLOO2 Evaluation Kit board
- Displays the position of dual in-line package (DIP) switches on IGLOO2 Evaluation Kit board
- Enables read and write to LSRAM
- Accepts and displays interrupts from the push button on the IGLOO2 Evaluation Kit board
- Shows the IGLOO2 PCIe configuration space
- Reads DSN

2.2.2 Description

The demo design accesses the IGLOO2 PCIe EP from the host PC. The following figure shows a detailed block diagram of the design implementation.

Figure 3 • **PCIe Control Plane Block Diagram**



The PCIe EP device receives commands from the host PC through GUI or Linux PCIe application and performs corresponding memory writes to the IGLOO2 fabric address space.

The SERDES_IF_0 is configured for a PCIe 2.0, ×1 link width with GEN2 speed. The PCIe interface to the fabric uses an advanced microcontroller bus architecture (AMBA) high-performance bus (AHB). The AHB master interface of SERDESIF is enabled and connected to the slaves CoreAHBLSRAM and CoreGPIO using the CoreAHBLite, CoreAHBTOAPB, and CoreAPB3 bus interfaces.

SERDES_IF_0 is initialized by high-performance memory subsystem (HPMS), which is configured by the System Builder.

The CoreSysServices Soft IP provides a user interface and AHB-Lite master interface to access the DSN system service. This system service fetches the 128-bit DSN. The DSN is unique to every device that is set during manufacturing. A simple Verilog logic is implemented to read the DSN using CoreSysServices IP and write the same to LSRAM.

See the *UG0450: SmartFusion2 SoC FPGA and IGLOO2 FPGA System Controller User Guide* for more information about system services.

The advanced extensible interface (AXI) master windows of the SERDESIF PCIe provide address translation for accessing one address space from another address space as the PCIe address is different from the IGLOO2 AHB bus matrix address space. The AXI master window 0 is enabled and configured to translate the BAR0 memory address space to the CoreGPIO address space to control the LEDs and DIP switches.

The AXI master window 1 is enabled and configured to translate the BAR1 memory address space to the CoreAHBLSRAM address space to perform read and write from PCIe. The IGLOO2 PCIe BAR0 and BAR1 are configured in 32-bit mode.

CoreGPIO is enabled and configured as below:

- GPIO_OUT [8] connected to user logic to read the DSN
- GPIO_OUT [7:0] connected LEDs
- GPIO_IN [4] indicates that the device serial number is available in LSRAM to display
- GPIO_IN [3:0] connected to DIP switches

The PCIe interrupt line is connected to the **SW4** on the IGLOO2 Evaluation Kit board. The FPGA clocks are configured to run the FPGA fabric at 50 MHz and HPMS at 100 MHz.

2.2.2.1 Simulating the Design

The design supports the BFM_PCIe simulation level to communicate with the SERDESIF block through the master AXI bus interface. Although, no serial communication uses the SERDESIF block, this scenario allows to validate the fabric interface connections. The `SERDESIF_0_user.bfm` file under the *<Libero project>/simulation* folder contains the BFM commands to verify the read or write access to CoreGPIO and CoreAHBLSRAM.

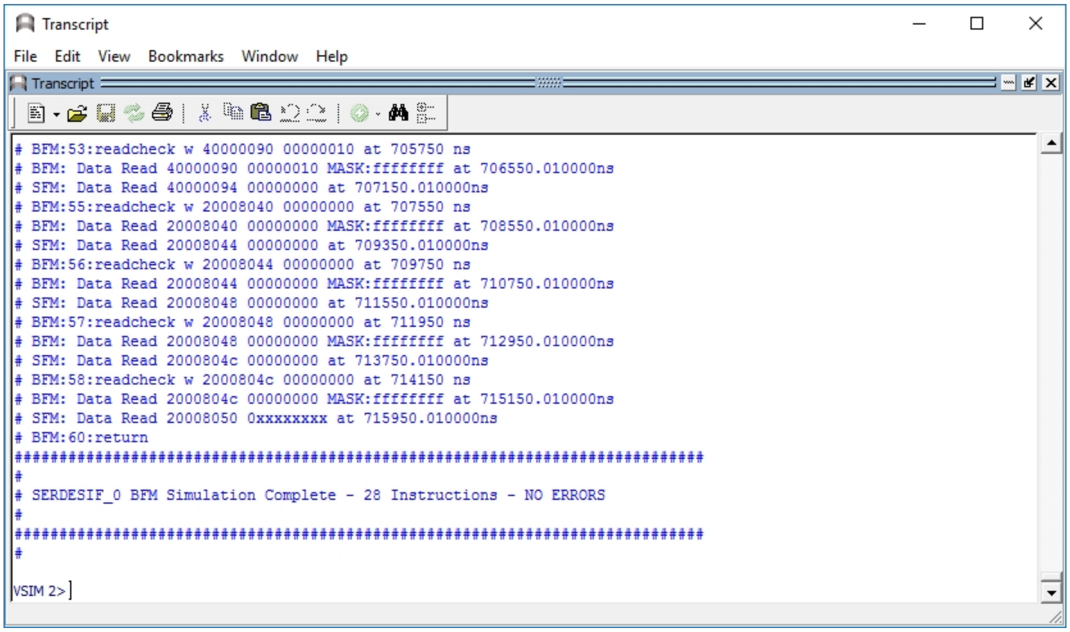
BFM commands added in the `SERDESIF_0_user.bfm` file do the following:

- Write to GPIO_OUT[7:0]
- Write to LSRAM
- Read-check from LSRAM

To run the simulation, double-click **Simulate** under **Verify Pre-Synthesized Design** in the **Design Flow** window of Libero project. ModelSim runs the design for about **720 µs**. The **ModelSim transcript** window displays the BFM commands and the BFM simulation completed with no errors, as shown in the following figure.

Note: In simulation, device serial number, which is read by BFM commands is always zero.

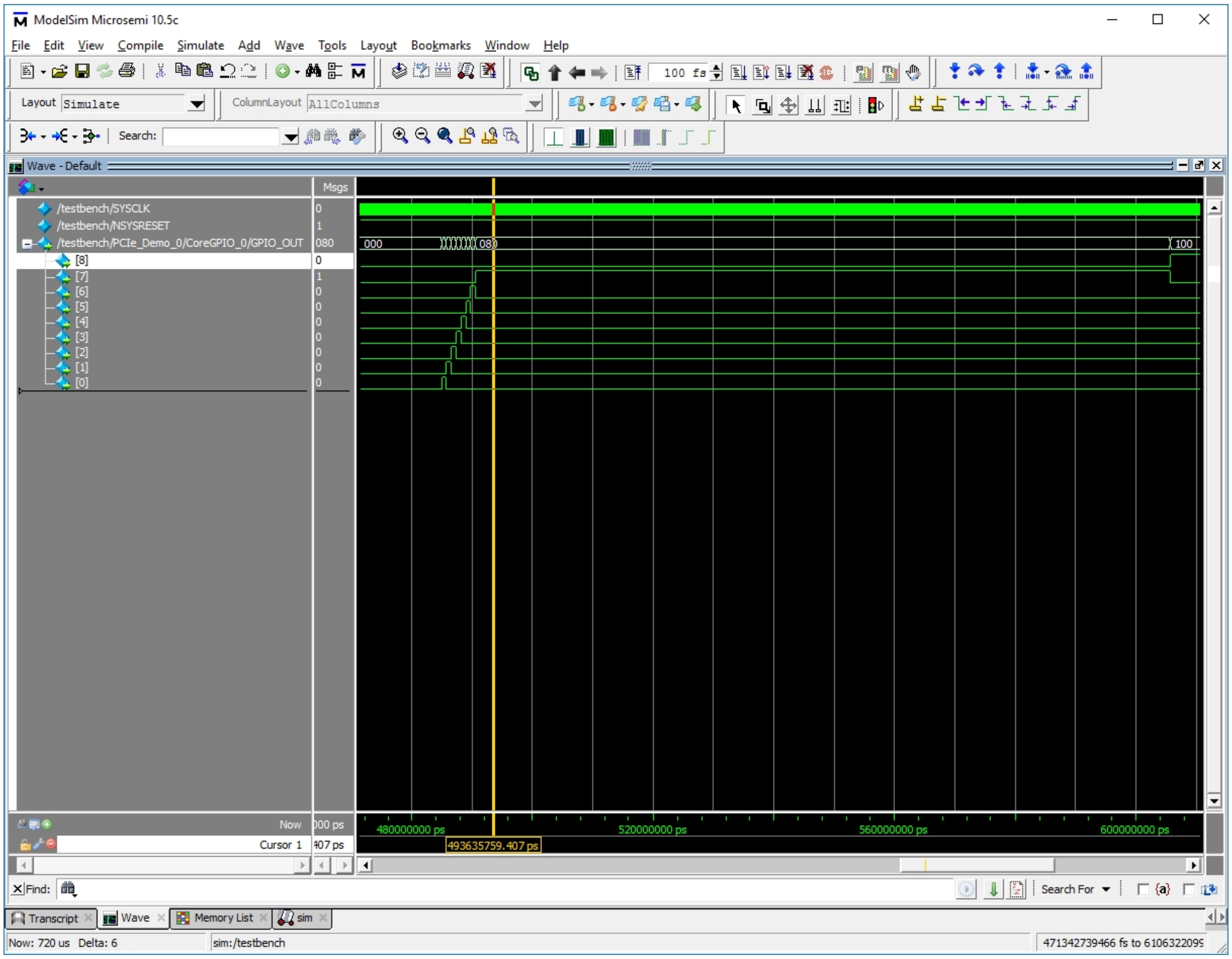
Figure 4 • **SERDES BFM Simulation**



```
# BFM:53:readcheck w 40000090 00000010 at 705750 ns
# BFM: Data Read 40000090 00000010 MASK:ffffffff at 706550.010000ns
# SFM: Data Read 40000094 00000000 at 707150.010000ns
# BFM:55:readcheck w 20008040 00000000 at 707550 ns
# BFM: Data Read 20008040 00000000 MASK:ffffffff at 708550.010000ns
# SFM: Data Read 20008044 00000000 at 709350.010000ns
# BFM:56:readcheck w 20008044 00000000 at 709750 ns
# BFM: Data Read 20008044 00000000 MASK:ffffffff at 710750.010000ns
# SFM: Data Read 20008048 00000000 at 711550.010000ns
# BFM:57:readcheck w 20008048 00000000 at 711950 ns
# BFM: Data Read 20008048 00000000 MASK:ffffffff at 712950.010000ns
# SFM: Data Read 2000804c 00000000 at 713750.010000ns
# BFM:58:readcheck w 2000804c 00000000 at 714150 ns
# BFM: Data Read 2000804c 00000000 MASK:ffffffff at 715150.010000ns
# SFM: Data Read 20008050 0xxxxxxxx at 715950.010000ns
# BFM:60:return
#############################################################################
#
# SERDESIF_0 BFM Simulation Complete - 28 Instructions - NO ERRORS
#
#############################################################################
#

VSIM 2>
```

The following figure shows the waveform window with GPIO_OUT signals.

Figure 5 • **Simulation Result with GPIO_OUT Signals**



2.3 Setting-up the Demo Design

The following steps describe how to setup the demo:

1. Connect the **FlashPro4 programmer** to the J5 connector of the IGLOO2 FPGA Evaluation Kit board.
2. Connect the jumpers on the IGLOO2 FPGA Evaluation Kit board, as shown in the following table.

CAUTION: Switch **OFF** the power supply switch **SW7** while connecting the jumpers.

Table 2 • **IGLOO2 FPGA Evaluation Kit Jumper Settings**

Jumper	Pin (From)	Pin (To)	Comments
J22	1	2	Default
J23	1	2	Default
J24	1	2	Default
J8	1	2	Default
J3	1	2	Default

3. Connect the power supply to the **J6** connector.

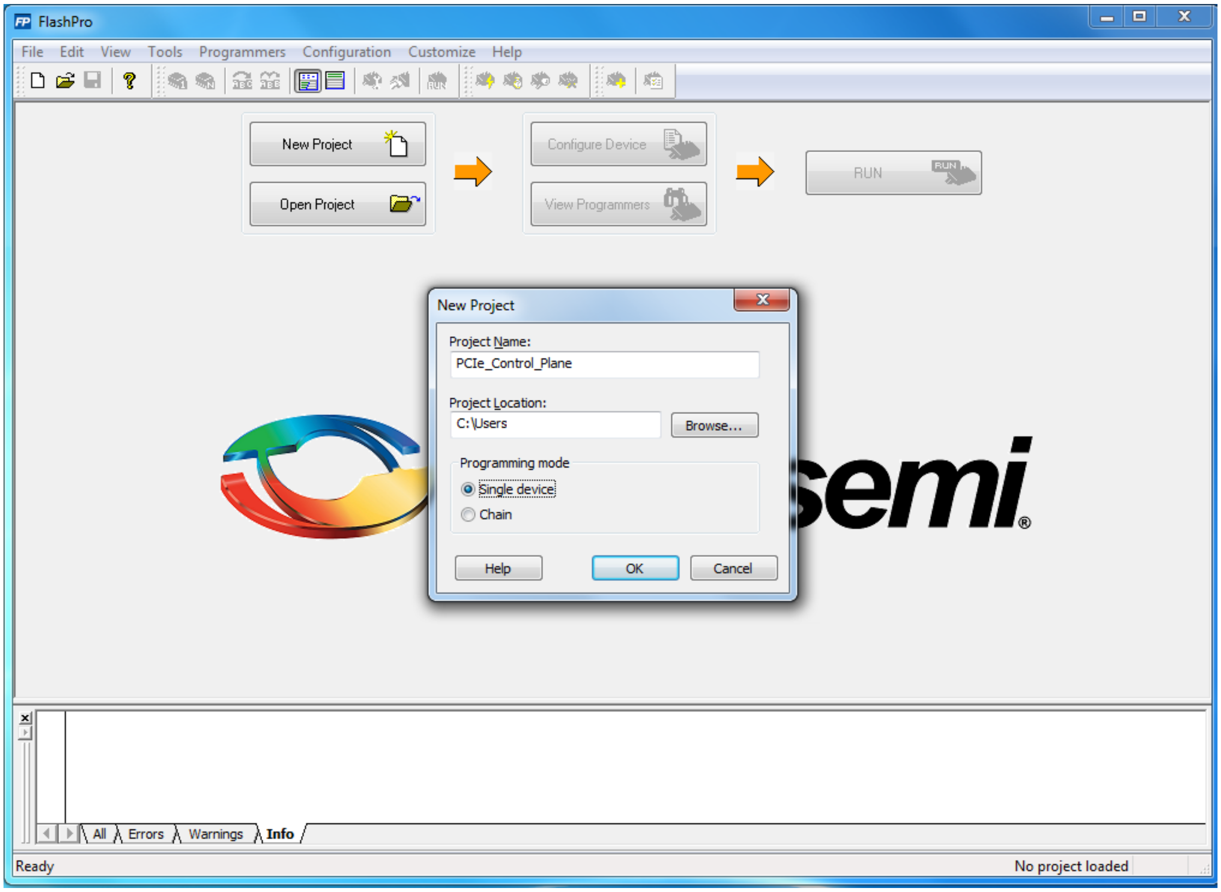
2.3.1 Board Setup

Snapshots of the IGLOO2 Evaluation Kit board with the complete setup is given in the Appendix: IGLOO2 Evaluation Kit Board, page 27.

2.3.2 Programming the IGLOO2 Board

1. Download the demo design from: *http://soc.microsemi.com/download/rsc/?f=m2gl_dg0532_liberov11p8_sp1_df*
2. Switch **ON** the **SW7** power supply switch.
3. Launch the **FlashPro** software.
4. Click **New Project**.
5. In the **New Project** window, type the **Project Name** as **PCIe_Control_Plane**.

Figure 6 • **FlashPro New Project**



New Project

Project Name: PCIe_Control_Plane

Project Location: C:\Users Browse...

Programming mode

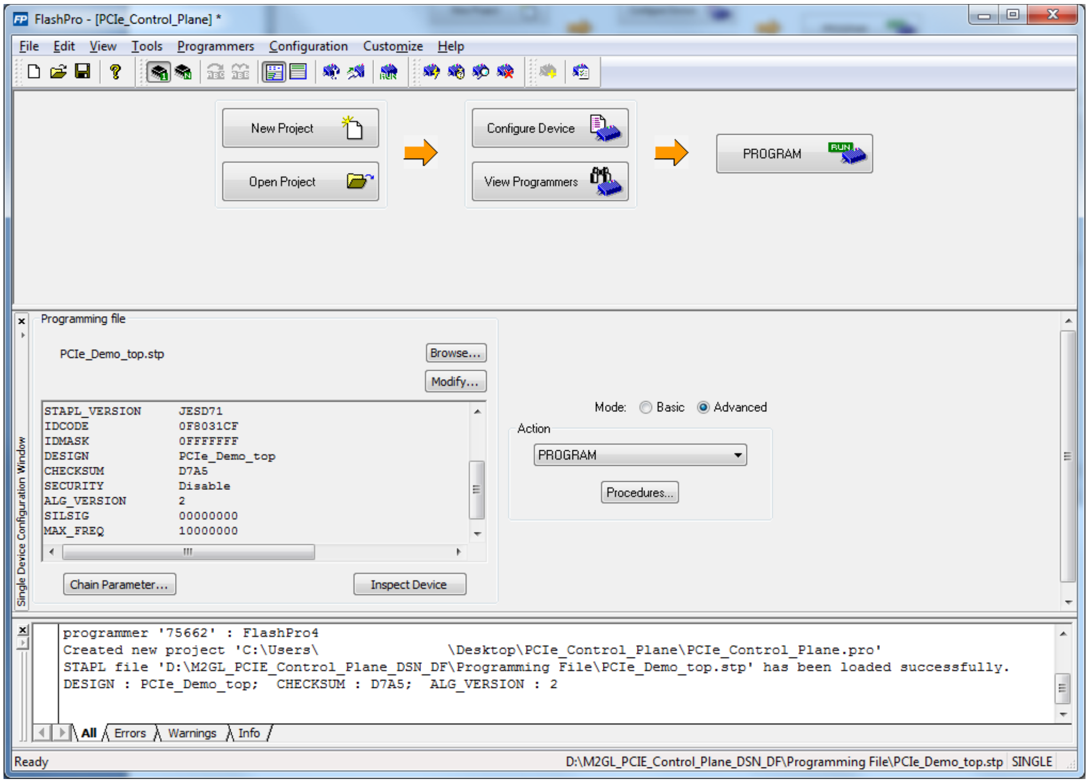
● Single device

○ Chain

Help OK Cancel

6. Click **Browse** and navigate to the location where you want to save the project.
7. Select **Single device** as the **Programming mode**.
8. Click **OK** to save the project.
9. Click **Configure Device** on the FlashPro GUI.
10. Click **Browse** and navigate to the location where the `PCIe_Demo_top.stp` file is located and select the file. The default location is: *<download_folder>\M2GL_PCIE_Control_Plane_DSN_DF\programming_file*.
11. Click **Open**. The required programming file is selected and is ready to be programmed in the device.

Figure 7 • **FlashPro Project Configured**



PCIe_Demo_top.stp

```
STAPL_VERSION    JESD71
IDCODE           0F8031CF
IDMASK           0FFFFFFF
DESIGN           PCIe_Demo_top
CHECKSUM         D7A5
SECURITY         Disable
ALG_VERSION      2
SILSIG           00000000
MAX_FREQ         10000000
```

Mode: ○ Basic ● Advanced

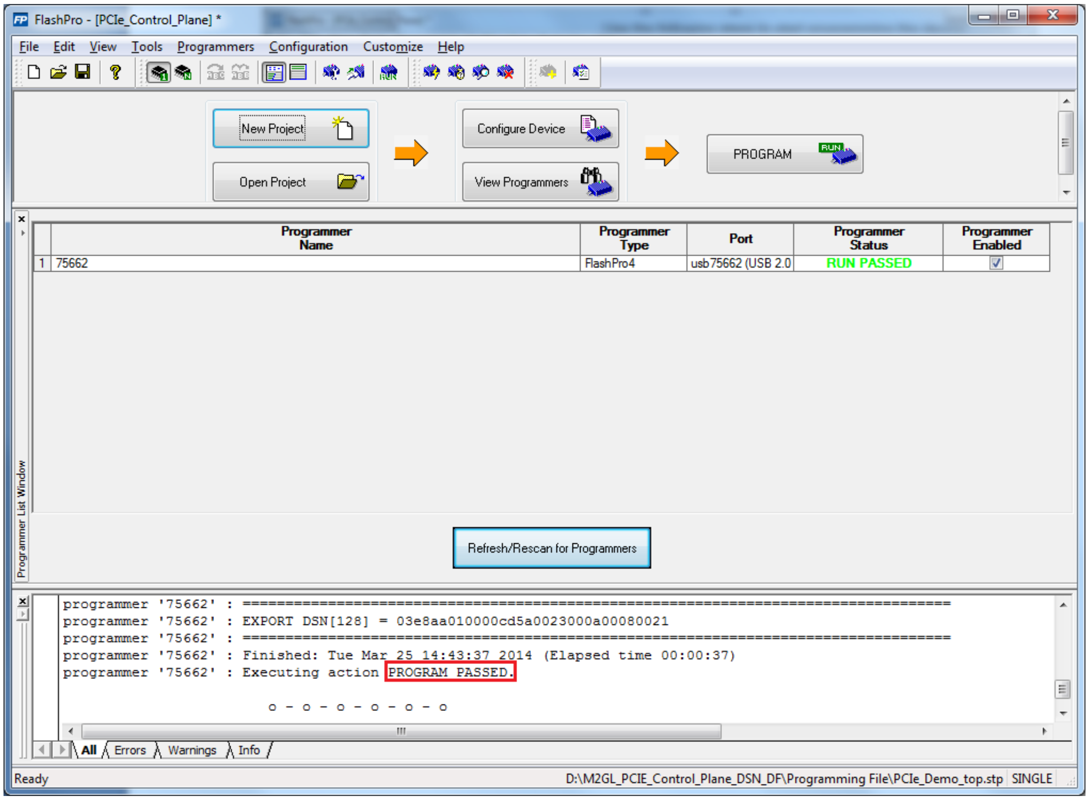
Action: PROGRAM

```
programmer '75662' : FlashPro4
Created new project 'C:\Users\                \Desktop\PCIe_Control_Plane\PCIe_Control_Plane.pro'
STAPL file 'D:\M2GL_PCIE_Control_Plane_DSN_DF\Programming File\PCIe_Demo_top.stp' has been loaded successfully.
DESIGN : PCIe_Demo_top;  CHECKSUM : D7A5;  ALG_VERSION : 2
```

Ready   D:\M2GL_PCIE_Control_Plane_DSN_DF\Programming File\PCIe_Demo_top.stp   SINGLE

12. Click **PROGRAM** to start programming the device. Wait until a message appears indicating **PROGRAM PASSED**, as shown in the following figure.

Figure 8 • **FlashPro Program Passed**


FlashPro - [PCIe_Control_Plane] *
File Edit View Tools Programmers Configuration Customize Help
New Project
Open Project
Configure Device
View Programmers
PROGRAM
RUN
Refresh/Rescan for Programmers
Programmer List Window
All
Errors
Warnings
Info

	Programmer Name	Programmer Type	Port	Programmer Status	Programmer Enabled
1	75662	FlashPro4	usb75662 (USB 2.0	RUN PASSED	☑

```
programmer '75662' : ================================================================================
programmer '75662' : EXPORT DSN[128] = 03e8aa010000cd5a0023000a00080021
programmer '75662' : ================================================================================
programmer '75662' : Finished: Tue Mar 25 14:43:37 2014 (Elapsed time 00:00:37)
programmer '75662' : Executing action PROGRAM PASSED.

                         o - o - o - o - o - o
```

Ready   D:\M2GL_PCIE_Control_Plane_DSN_DF\Programming File\PCIe_Demo_top.stp   SINGLE

2.3.3 Connecting the Evaluation Kit Board to the Host PC

The following steps describe how to connect the IGLOO2 Evaluation Kit board to the host PC:

1. After successful programming, power **OFF** the IGLOO2 Evaluation Kit board and shut down the host PC.
2. Use the following steps to connect the **CON1–PCIe Edge Connector** either to a host PC or laptop:

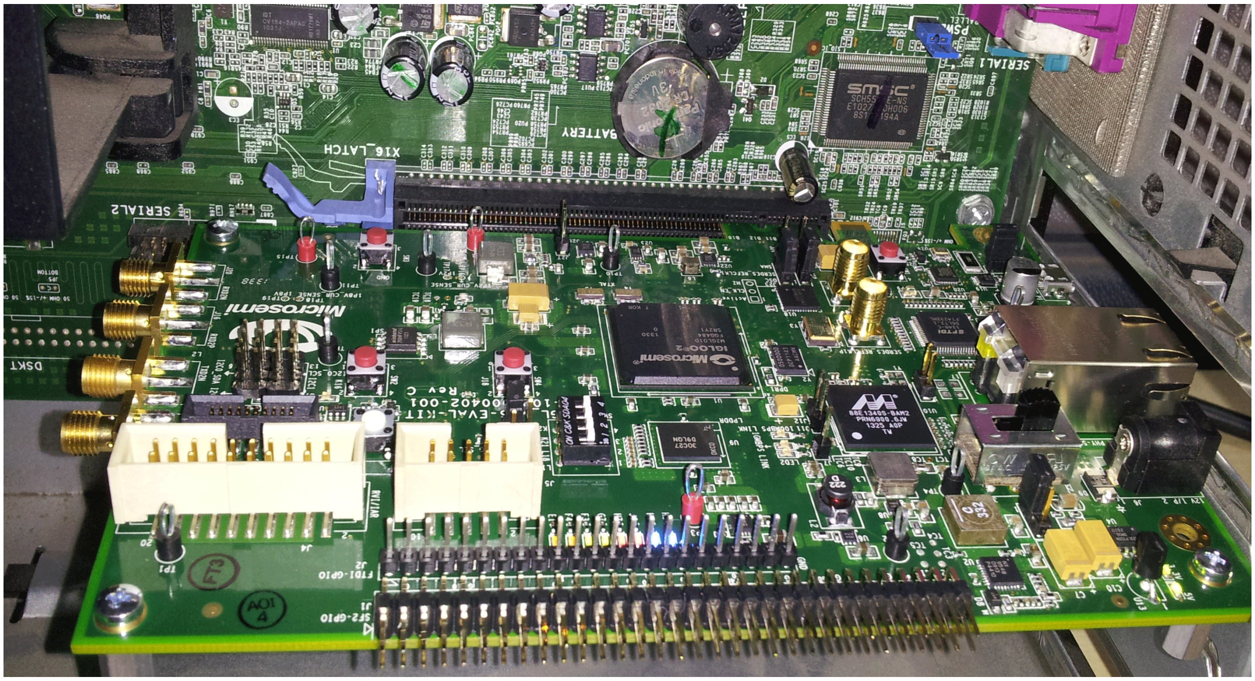
   a. Connect the **CON1–PCIe Edge Connector** to host PC PCIe Gen2 slot or Gen1 slot as applicable. This demo guide is designed to run in any PCIe Gen2 compliant slot. If the host PC does not support the Gen2 compliant slot, the design switches to Gen1 mode.

   b. Connect the **CON1–PCIe Edge Connector** to the laptop PCIe slot using the express card adapter. If you are using a laptop, the express card adapters typically support only Gen1 and the design works on Gen1 mode.

Note: Host PC or laptop must be powered **OFF** while inserting the PCIe Edge Connector. If the system is not powered **OFF**, the PCIe device detection and selection of Gen1 or Gen2 do not occur properly. It is recommended that the host PC or laptop must be powered **OFF** during the PCIe card insertion.

The following figure shows the board setup for the host PC in which IGLOO2 Evaluation Kit board is connected to the host PC PCIe slot. To connect the IGLOO2 Evaluation Kit board to the laptop using Express card adapter, see the Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop, page 28.

Figure 9 • **IGLOO2 Evaluation Kit Setup for Host PC**



2.4 Running the Demo Design

This demo can run on both Windows and RedHat Linux OS.

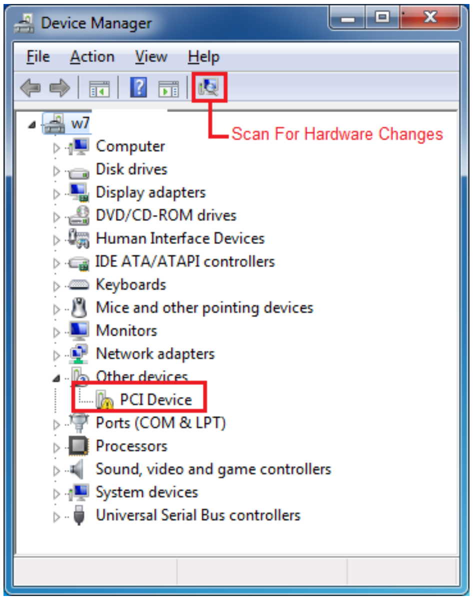
- To run the demo on Windows OS GUI, Microsemi PCIe drivers are provided. See Running the Demo Design on Windows, page 10.
- To run the demo on Linux OS, native RedHat Linux drivers and command line scripts are provided. See Running the Demo Design on Linux, page 17

2.4.1 Running the Demo Design on Windows

The following steps describe how to run the demo design on Windows:

1. Switch **ON** the power supply switch, **SW7**.
2. Power on the host PC and open the host PC **Device Manager** for PCIe device, as shown in the following figure. If the PCIe device is not detected, power cycle the IGLOO2 Evaluation Kit board.
3. Right-click **PCI Device** > **scan for hardware changes** in Device Manager.

Figure 10 • **Device Manager**



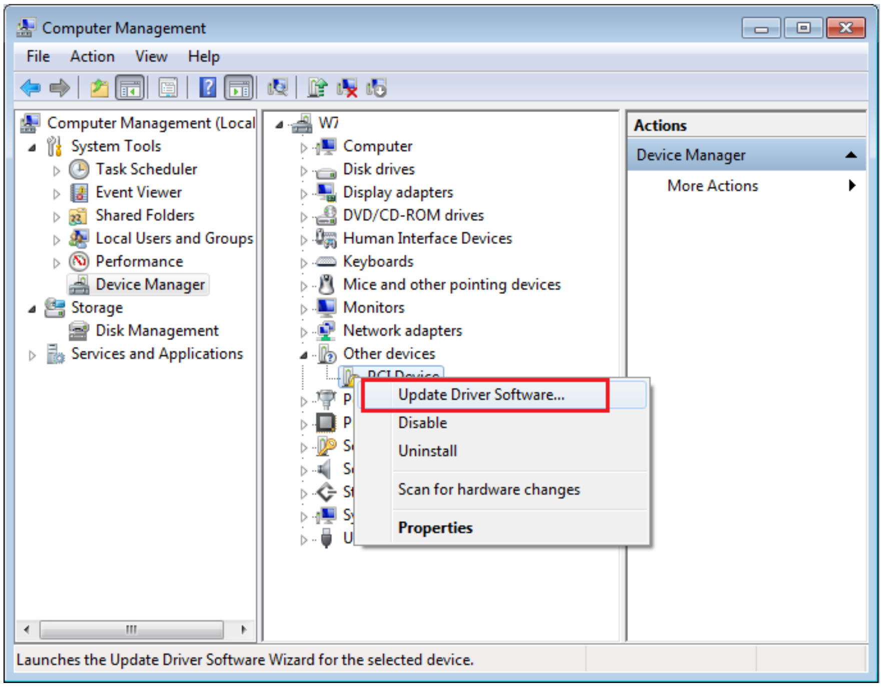
Note: If the device is still not detected, check whether or not the basic input/output system (BIOS) version in host PC is the latest, and if PCIe is enabled in the host PC BIOS.

2.4.1.1 Drivers Installation

Perform the following steps to install the PCIe drivers on the host PC:

1. Right-click **PCI Device** in Device Manager and select **Update Driver Software...**, as shown in the following figure. To install the drivers, administrative rights are required.

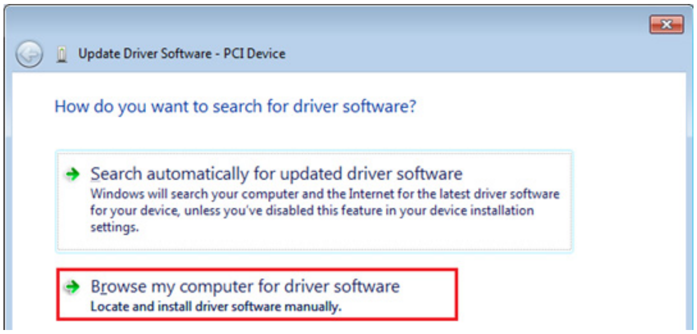
Figure 11 • **Update Driver Software**



Launches the Update Driver Software Wizard for the selected device.

2. In the **Update Driver Software - PCIe Device** window, select the **Browse my computer for driver software** option as shown in the following figure.

Figure 12 • **Browse for Driver Software**


Update Driver Software - PCI Device

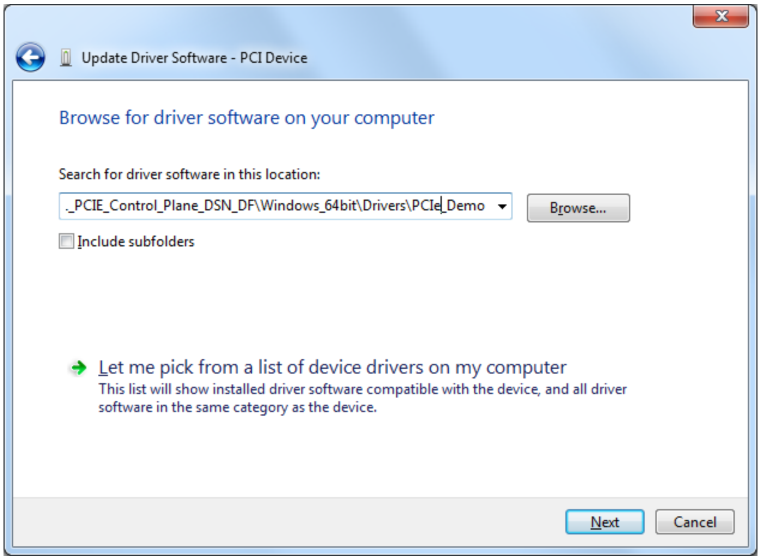
How do you want to search for driver software?

→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software
Locate and install driver software manually.

3. Browse the drivers folder: M2GL_PCIE_Control_Plane_DSN_DF\Windows_64bit\Drivers\PCIe_Demo and click **Next**, as shown in the following figure.

Figure 13 • **Browse for Driver Software Continued**



Update Driver Software - PCI Device

Browse for driver software on your computer

Search for driver software in this location:

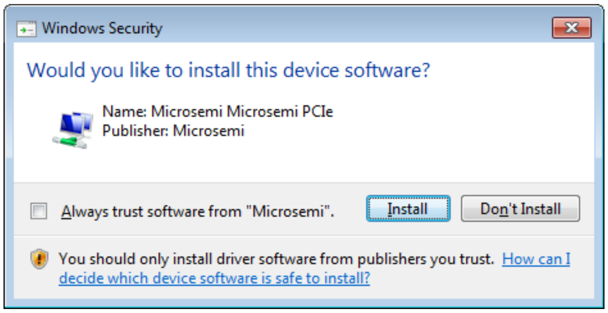
._PCIE_Control_Plane_DSN_DF\Windows_64bit\Drivers\PCIe_Demo

□ Include subfolders

➔ Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

4. The **Windows Security** dialog box is displayed. Click **Install** as shown in the following figure. After successful driver installation, a message appears. See Figure 15, page 12.

Figure 14 • **Windows Security**



Windows Security

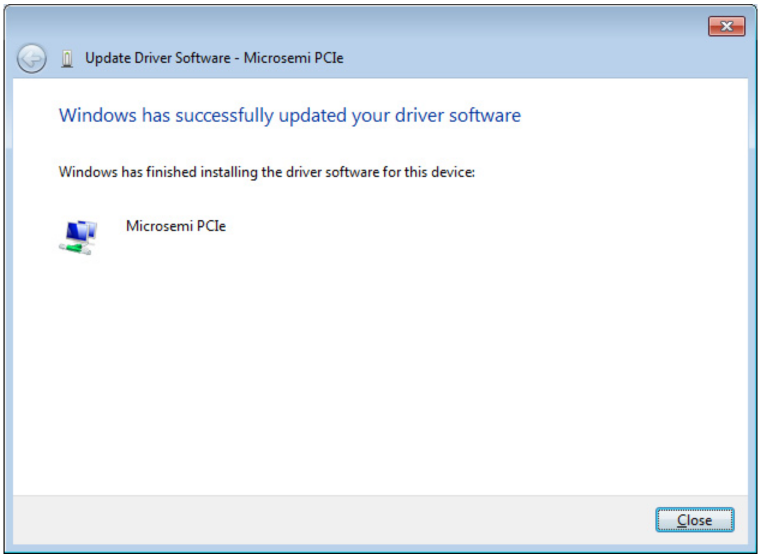
Would you like to install this device software?

Name: Microsemi Microsemi PCIe
Publisher: Microsemi

□ Always trust software from "Microsemi".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Figure 15 • **Successful Driver Installation**



Update Driver Software - Microsemi PCIe

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Microsemi PCIe

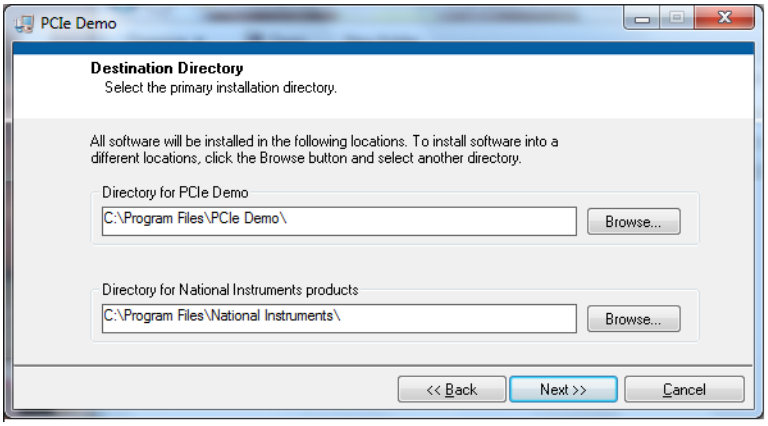
2.4.1.2 PCIe Demo GUI Installation

IGLOO2 PCIe demo GUI is a simple GUI that runs on the host PC to communicate with the IGLOO2 PCIe EP device. The GUI provides the PCIe link status, driver information, and demo controls. The GUI invokes the PCIe driver installed on the host PC and provides commands to the driver according to the user selection.

To install the GUI:

1. Extract the PCIe_Demo_GUI_Installer.rar and locate the files at **M2GL_PCIE_Control_Plane_DSN_DF\GUI**
2. Double-click `setup.exe` in the provided GUI installation (*PCIe_Control_Plane_Demo_GUI_Installer\setup.exe*). Apply default options, as shown in the following figure.
3. To start the installation, click **Next**.

Figure 16 • **GUI Installation**



Destination Directory

Select the primary installation directory.

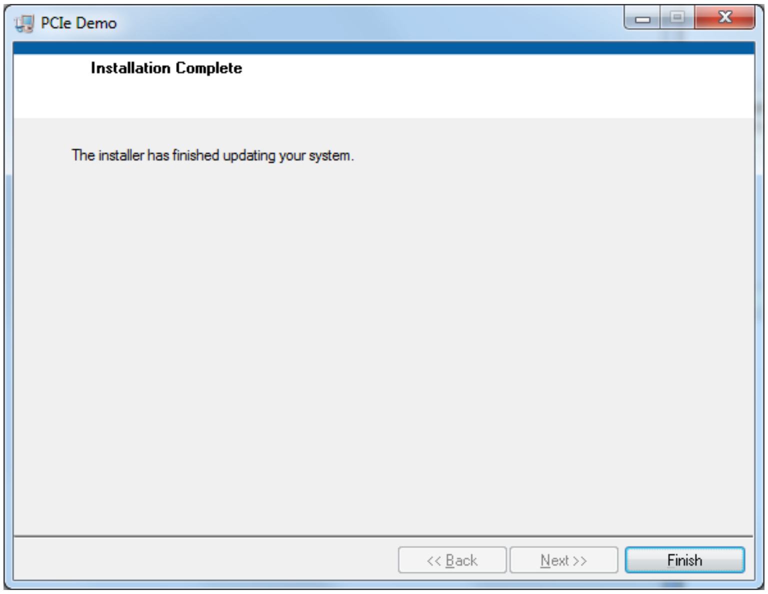
All software will be installed in the following locations. To install software into a different locations, click the Browse button and select another directory.

Directory for PCIe Demo: C:\Program Files\PCIe Demo\

Directory for National Instruments products: C:\Program Files\National Instruments\

4. Click **Finish** to complete the installation.

Figure 17 • **Successful GUI Installation**



Installation Complete

The installer has finished updating your system.

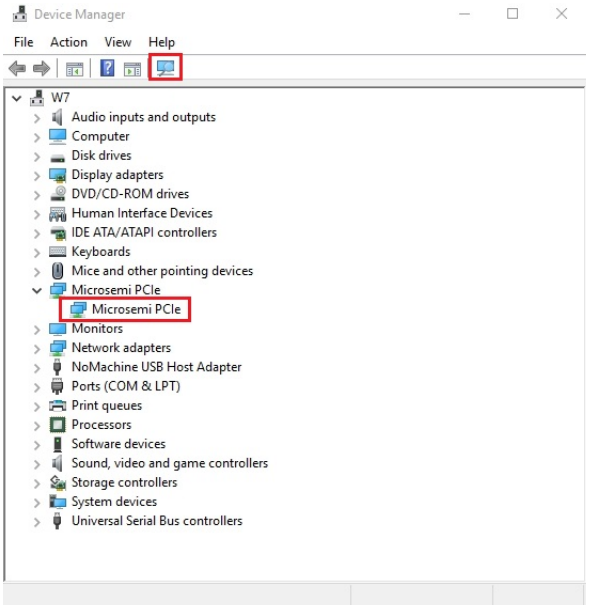
5. Restart the host PC.

2.4.1.3 Running the PCIe GUI

The following steps describe how to run the PCIe GUI:

1. Check the host PC **Device Manager** for the drivers. If the device is not detected, power cycle the IGLOO2 Evaluation Kit board.
2. Click **scan for hardware changes** in Device Manager window. Ensure that the board is switched on.

Figure 18 • **Device Manager—PCIe Device Detection**

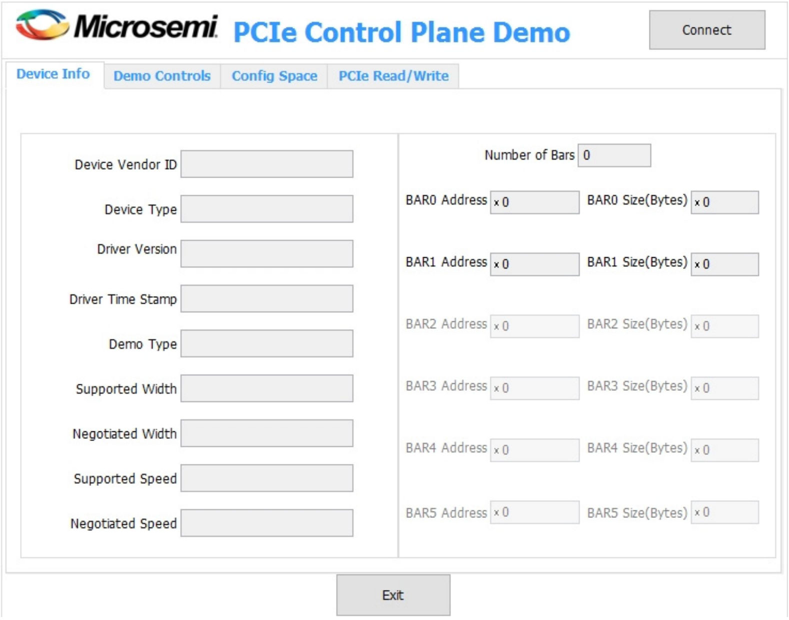


W7
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Microsemi PCIe
  - Microsemi PCIe
- Monitors
- Network adapters
- NoMachine USB Host Adapter
- Ports (COM & LPT)
- Print queues
- Processors
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

Note: If a warning symbol is displayed on the **Microsemi PCIe** in the **Device Manager**, uninstall them and start from step1 of Drivers Installation, page 11.

3. Invoke the GUI from **ALL Programs > PCIe Control Plane Demo**. The GUI is displayed, as shown in the following figure.

Figure 19 • **PCIe Demo GUI**



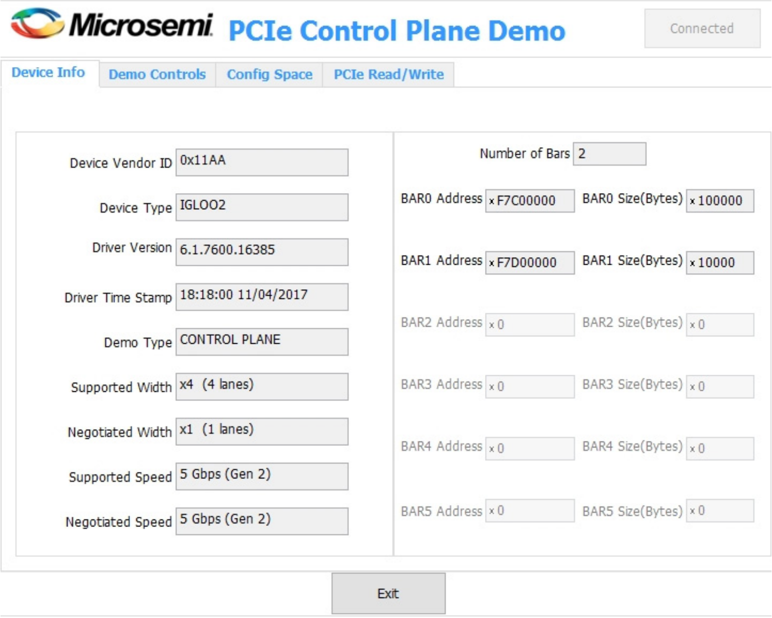
Device Vendor ID
Device Type
Driver Version
Driver Time Stamp
Demo Type
Supported Width
Negotiated Width
Supported Speed
Negotiated Speed

Number of Bars 0

BAR0 Address	x 0	BAR0 Size(Bytes)	x 0
BAR1 Address	x 0	BAR1 Size(Bytes)	x 0
BAR2 Address	x 0	BAR2 Size(Bytes)	x 0
BAR3 Address	x 0	BAR3 Size(Bytes)	x 0
BAR4 Address	x 0	BAR4 Size(Bytes)	x 0
BAR5 Address	x 0	BAR5 Size(Bytes)	x 0

4. Click **Connect**. The application detects and displays the information related to the connected kit such as Device Vendor ID, Device Type, Driver Version, Driver Time Stamp, Demo Type, Supported Width, Negotiated Width, Supported Speed, Negotiated Speed, Number of Bars, and BAR Address as shown in the following figure.

Figure 20 • **Device Info**



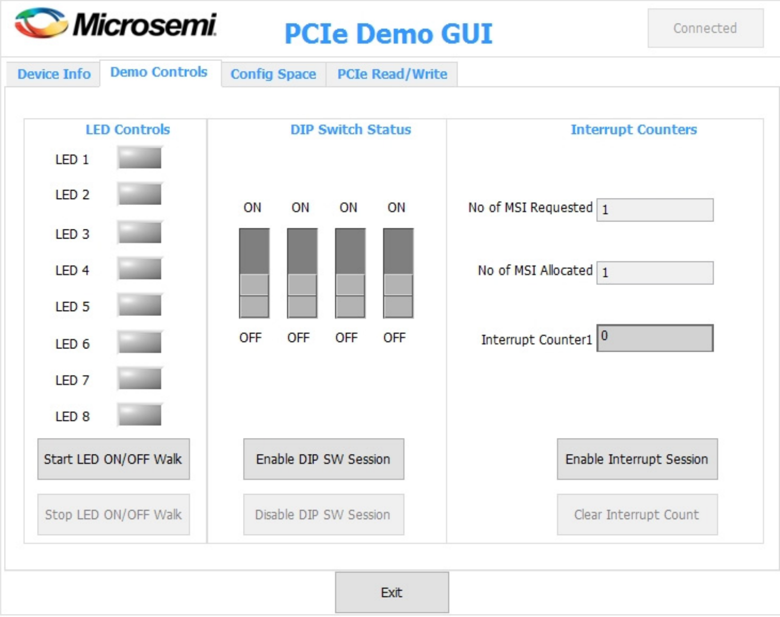
Device Vendor ID 0x11AA
Device Type IGLOO2
Driver Version 6.1.7600.16385
Driver Time Stamp 18:18:00 11/04/2017
Demo Type CONTROL PLANE
Supported Width x4 (4 lanes)
Negotiated Width x1 (1 lanes)
Supported Speed 5 Gbps (Gen 2)
Negotiated Speed 5 Gbps (Gen 2)

Number of Bars 2

BAR0 Address	x F7C00000	BAR0 Size(Bytes)	x 100000
BAR1 Address	x F7D00000	BAR1 Size(Bytes)	x 10000
BAR2 Address	x 0	BAR2 Size(Bytes)	x 0
BAR3 Address	x 0	BAR3 Size(Bytes)	x 0
BAR4 Address	x 0	BAR4 Size(Bytes)	x 0
BAR5 Address	x 0	BAR5 Size(Bytes)	x 0

5. Click the **Demo Controls** tab to display the **LED Controls**, **DIP Switch Status**, and **Interrupt Counters** as shown in the following figure.

Figure 21 • **Demo Controls**

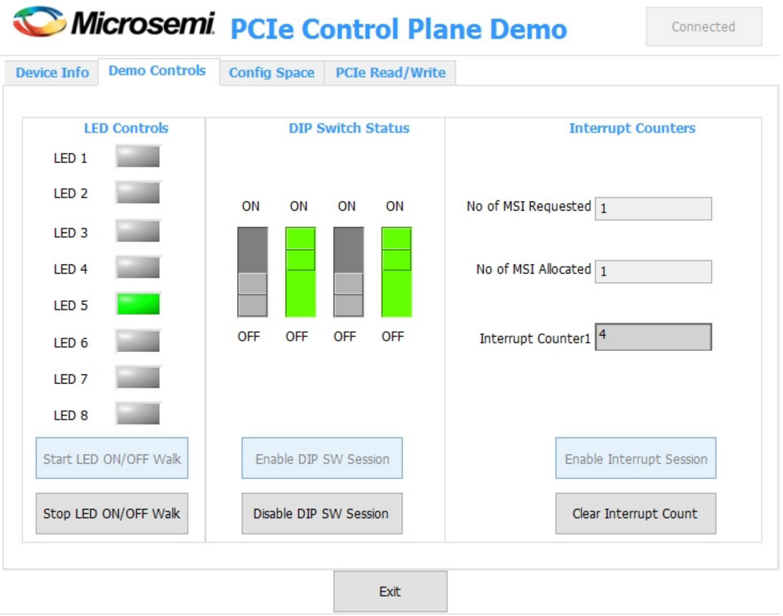


No of MSI Requested 1
No of MSI Allocated 1
Interrupt Counter1 0

6. Click **Start LED ON/OFF Walk**, **Enable DIP SW Session**, and **Enable Interrupt Session** to view controlling LEDs, getting the DIP switch status, and monitoring the interrupts simultaneously as shown in the following figure.
7. Change the DIP switch positions on the IGLOO2 Evaluation Kit board (**SW5**) and observe the similar position of switches in **GUI SWITCH MODULE**.

8. Press **SW4** on the IGLOO2 Evaluation Kit board and observe the interrupt count on the **Interrupt Counter** field in GUI.

Figure 22 • **Demo Controls—Continued**



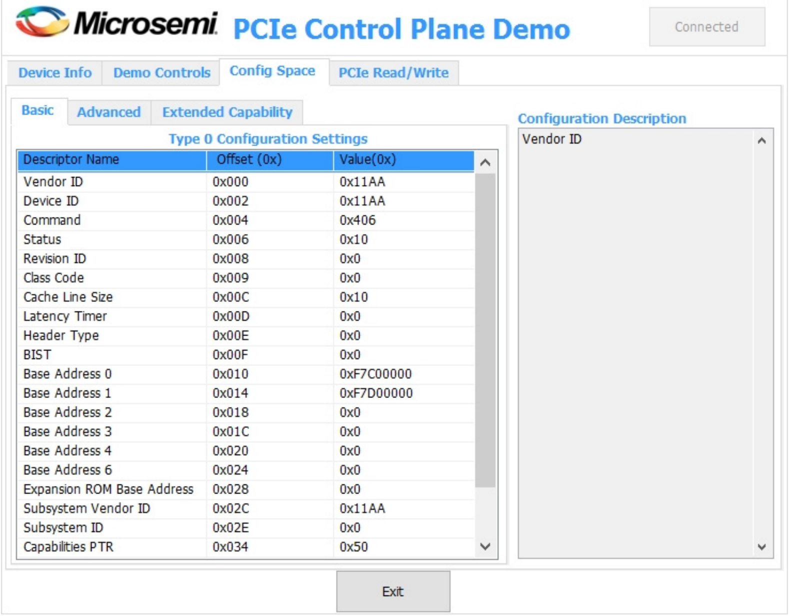
No of MSI Requested 1

No of MSI Allocated 1

Interrupt Counter1 4

9. Click **Config Space** to view details about the PCIe configuration space. The following figure shows the PCIe configuration space.

Figure 23 • **Configuration Space**



Type 0 Configuration Settings

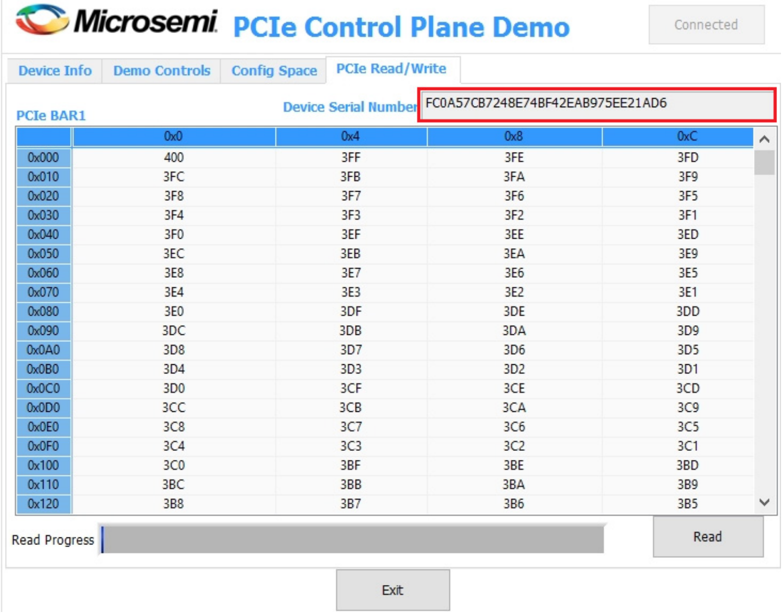
Descriptor Name	Offset (0x)	Value(0x)
Vendor ID	0x000	0x11AA
Device ID	0x002	0x11AA
Command	0x004	0x406
Status	0x006	0x10
Revision ID	0x008	0x0
Class Code	0x009	0x0
Cache Line Size	0x00C	0x10
Latency Timer	0x00D	0x0
Header Type	0x00E	0x0
BIST	0x00F	0x0
Base Address 0	0x010	0xF7C00000
Base Address 1	0x014	0xF7D00000
Base Address 2	0x018	0x0
Base Address 3	0x01C	0x0
Base Address 4	0x020	0x0
Base Address 6	0x024	0x0
Expansion ROM Base Address	0x028	0x0
Subsystem Vendor ID	0x02C	0x11AA
Subsystem ID	0x02E	0x0
Capabilities PTR	0x034	0x50

Configuration Description

Vendor ID

10. Click the **PCIe Read/Write** tab to perform read and writes to LSRAM memory through **BAR1** space. Click **Read** to read the 4 KB memory mapped to BAR1 space as shown in the following figure. Device serial number is also read and displayed in the GUI, as shown in the following figure.

Figure 24 • **PCIe BAR1 Memory Access**



Device Serial Number FC0A57CB7248E74BF42EAB975EE21AD6

	0x0	0x4	0x8	0xC
0x000	400	3FF	3FE	3FD
0x010	3FC	3FB	3FA	3F9
0x020	3F8	3F7	3F6	3F5
0x030	3F4	3F3	3F2	3F1
0x040	3F0	3EF	3EE	3ED
0x050	3EC	3EB	3EA	3E9
0x060	3E8	3E7	3E6	3E5
0x070	3E4	3E3	3E2	3E1
0x080	3E0	3DF	3DE	3DD
0x090	3DC	3DB	3DA	3D9
0x0A0	3D8	3D7	3D6	3D5
0x0B0	3D4	3D3	3D2	3D1
0x0C0	3D0	3CF	3CE	3CD
0x0D0	3CC	3CB	3CA	3C9
0x0E0	3C8	3C7	3C6	3C5
0x0F0	3C4	3C3	3C2	3C1
0x100	3C0	3BF	3BE	3BD
0x110	3BC	3BB	3BA	3B9
0x120	3B8	3B7	3B6	3B5

11. Click **Exit** to quit the demo.

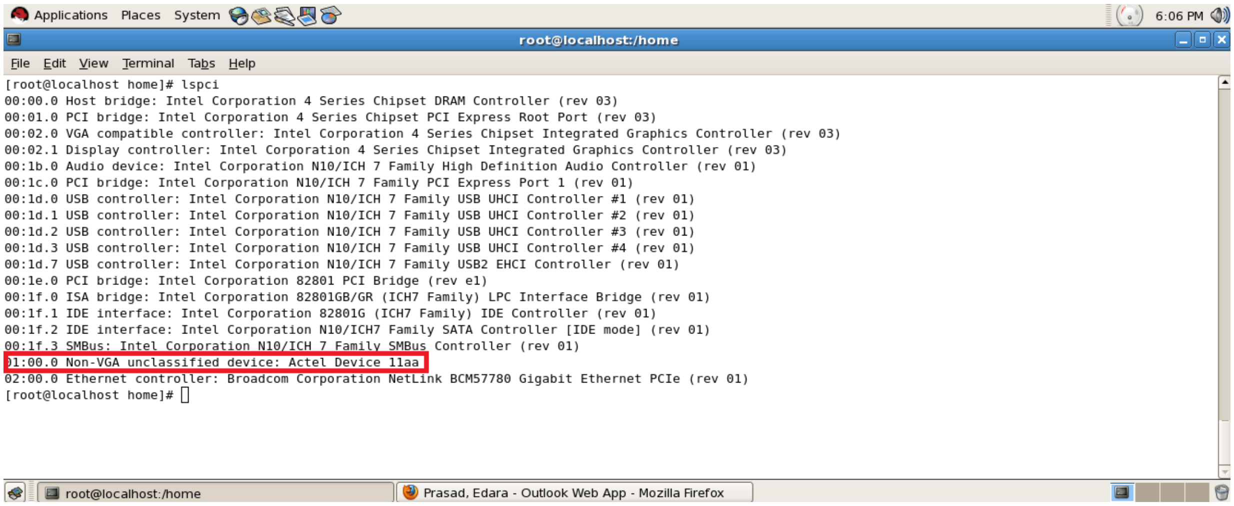
2.4.2 Running the Demo Design on Linux

The following steps describe how to run the demo design on Linux:

1. Switch **ON** the power supply switch on the IGLOO2 Evaluation Kit board.
2. Switch **ON** the RedHat Linux host PC.
3. RedHat Linux Kernel detects the IGLOO2 PCIe end point as Actel Device.
4. On Linux Command Prompt Use `lspci` command to display the PCIe info.

```
# lspci
```

Figure 25 • **PCIe Device Detection**



```
[root@localhost home]# lspci
00:00.0 Host bridge: Intel Corporation 4 Series Chipset DRAM Controller (rev 03)
00:01.0 PCI bridge: Intel Corporation 4 Series Chipset PCI Express Root Port (rev 03)
00:02.0 VGA compatible controller: Intel Corporation 4 Series Chipset Integrated Graphics Controller (rev 03)
00:02.1 Display controller: Intel Corporation 4 Series Chipset Integrated Graphics Controller (rev 03)
00:1b.0 Audio device: Intel Corporation N10/ICH 7 Family High Definition Audio Controller (rev 01)
00:1c.0 PCI bridge: Intel Corporation N10/ICH 7 Family PCI Express Port 1 (rev 01)
00:1d.0 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #1 (rev 01)
00:1d.1 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #2 (rev 01)
00:1d.2 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #3 (rev 01)
00:1d.3 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #4 (rev 01)
00:1d.7 USB controller: Intel Corporation N10/ICH 7 Family USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev e1)
00:1f.0 ISA bridge: Intel Corporation 82801GB/GR (ICH7 Family) LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801G (ICH7 Family) IDE Controller (rev 01)
00:1f.2 IDE interface: Intel Corporation N10/ICH7 Family SATA Controller [IDE mode] (rev 01)
00:1f.3 SMBus: Intel Corporation N10/ICH 7 Family SMBus Controller (rev 01)
01:00.0 Non-VGA unclassified device: Actel Device 11aa
02:00.0 Ethernet controller: Broadcom Corporation NetLink BCM57780 Gigabit Ethernet PCIe (rev 01)
[root@localhost home]#
```

2.4.2.1 Drivers Installation

Enter the following commands in the Linux command prompt to install the PCIe drivers:

1. Create the **igl2** directory under **home/**.
   ```
   # mkdir /home/igl2
   ```
2. Bring the *M2GL_PCIE_Control_Plane_DSN_DF/* design files folder under */home/igl2* directory, which contains the Linux PCIe device driver files and Linux PCIe application utility files.
3. Copy the Linux PCIe Device Driver file (`PCIe_Driver.zip`) from *M2GL_PCIE_Control_Plane_DSN_DF/* design files folder.
   ```
   # cp -rf
   /home/igl2/M2GL_PCIE_Control_Plane_DSN_DF/Linux_64bit/Drivers/PCIe_Driver
   .zip /home/igl2
   ```

```
# unzip PCIe_Driver.zip
```

4. */home/igl2* directory must contain *PCIe_Driver/ inc/* folders.

Execute `ls` command to display the contents of */home/igl2* directory.

```
# ls
```

5. Change to *inc/* directory.

```
#cd /home/igl2/inc
```

6. Edit the `board.h` file for IGLOO2 Evaluation Kit.

```
#vi board.h

#define IGL2

#undef SF2
```

7. To save the selected file, perform `[:wq]`
8. To change the directory, use the following command:

```
#cd /home/igl2/PCIe_Driver
```

9. To compile the Linux PCIe device driver code, execute make command on Linux Command Prompt.

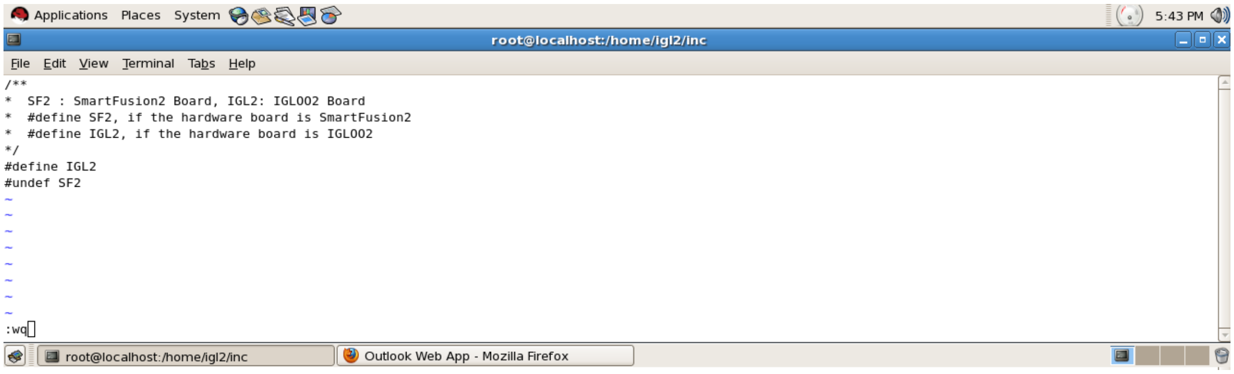
```
#make clean [To clean any *.o, *.ko files]

#make
```

10. The kernel module, `pci_chr_drv_ctrlpln.ko` creates in the same directory.
11. To insert the Linux PCIe device driver as a module, execute `insmod` command on Linux Command Prompt.

```
#insmod pci_chr_drv_ctrlpln.ko
```

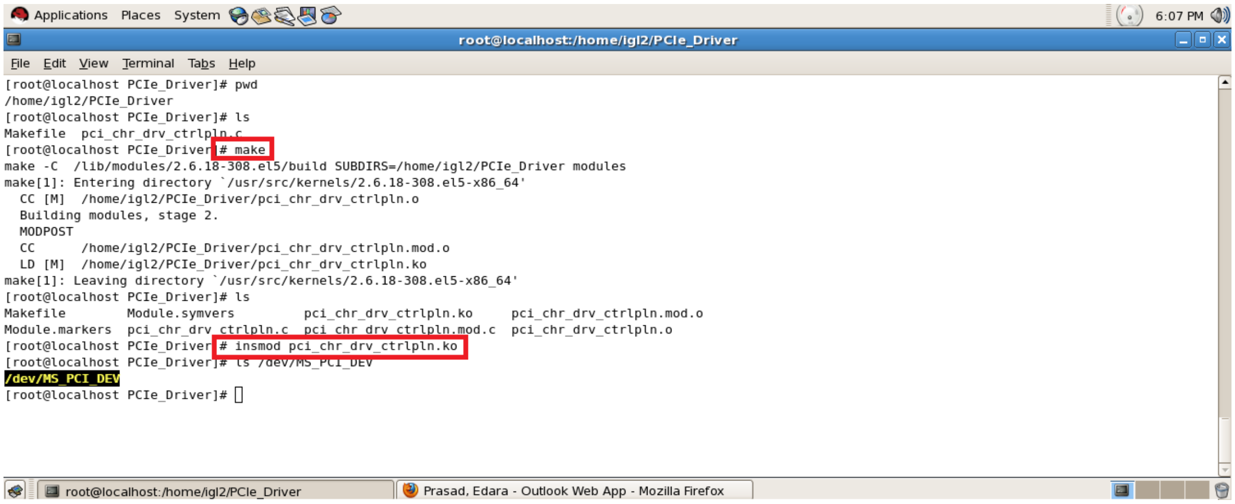
Note: Root Privileges are required to execute this command.

Figure 26 • **Edit board.h File**


Applications Places System
5:43 PM
root@localhost:/home/igl2/inc
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/inc
Outlook Web App - Mozilla Firefox

```
/**
*  SF2 : SmartFusion2 Board, IGL2: IGLOO2 Board
*  #define SF2, if the hardware board is SmartFusion2
*  #define IGL2, if the hardware board is IGLOO2
*/
#define IGL2
#undef SF2
~
~
~
~
~
~
~
~
:wq
```

Figure 27 • **PCIe Device Driver Installation**


Applications
Places
System
6:07 PM
root@localhost:/home/igl2/PCIe_Driver
File
Edit
View
Terminal
Tabs
Help
root@localhost:/home/igl2/PCIe_Driver
Prasad, Edara - Outlook Web App - Mozilla Firefox

```
[root@localhost PCIe_Driver]# pwd
/home/igl2/PCIe_Driver
[root@localhost PCIe_Driver]# ls
Makefile  pci_chr_drv_ctrlpln.c
[root@localhost PCIe_Driver]# make
make -C  /lib/modules/2.6.18-308.el5/build SUBDIRS=/home/igl2/PCIe_Driver modules
make[1]: Entering directory `/usr/src/kernels/2.6.18-308.el5-x86_64'
  CC [M]  /home/igl2/PCIe_Driver/pci_chr_drv_ctrlpln.o
  Building modules, stage 2.
  MODPOST
  CC      /home/igl2/PCIe_Driver/pci_chr_drv_ctrlpln.mod.o
  LD [M]  /home/igl2/PCIe_Driver/pci_chr_drv_ctrlpln.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.18-308.el5-x86_64'
[root@localhost PCIe_Driver]# ls
Makefile        Module.symvers         pci_chr_drv_ctrlpln.ko     pci_chr_drv_ctrlpln.mod.o
Module.markers  pci_chr_drv_ctrlpln.c  pci_chr_drv_ctrlpln.mod.c  pci_chr_drv_ctrlpln.o
[root@localhost PCIe_Driver]# insmod pci_chr_drv_ctrlpln.ko
[root@localhost PCIe_Driver]# ls /dev/MS_PCI_DEV
/dev/MS_PCI_DEV
[root@localhost PCIe_Driver]# 
```

12. After successful Linux PCIe device driver installation, check `/dev/MS_PCI_DEV` is created by using the following Linux command:

```
#ls/dev/MS_PCI_DEV
```

Note: `/dev/MS_PCI_DEV` interface is used to access the IGLOO2 PCIe endpoint from Linux user space.

2.4.2.2 Linux PCIe Application Compilation and PCIe Control Plane Utility Creation

1. Change to */home/igl2* directory.

```
# cd /home/igl2
```

2. Copy the Linux PCIe application utility file (`PCIe_App.zip`) from *M2GL_PCIE_Control_Plane_DSN_DF/* design files folder.

```
# cp -rf
/home/igl2/M2GL_PCIE_Control_Plane_DSN_DF/Linux_64bit/UTIL/PCIe_App.zip
/home/igl2

# unzip PCIe_App.zip
```

3. */home/igl2* directory must contain *PCIe_App/* folder along with `led_blink.sh` and `pcie_config.sh` scripts. Execute `ls` command to display the contents in */home/igl2* directory.

```
# ls
```

4. Compile the Linux user space application `pcie_appln_ctrlpln.c` in /home/igl2/PCIe_App folder by using gcc command.

```
 # cd /home/igl2/PCIe_App

 # gcc -o pcie_ctrlplane pcie_appln_ctrlpln.c
```

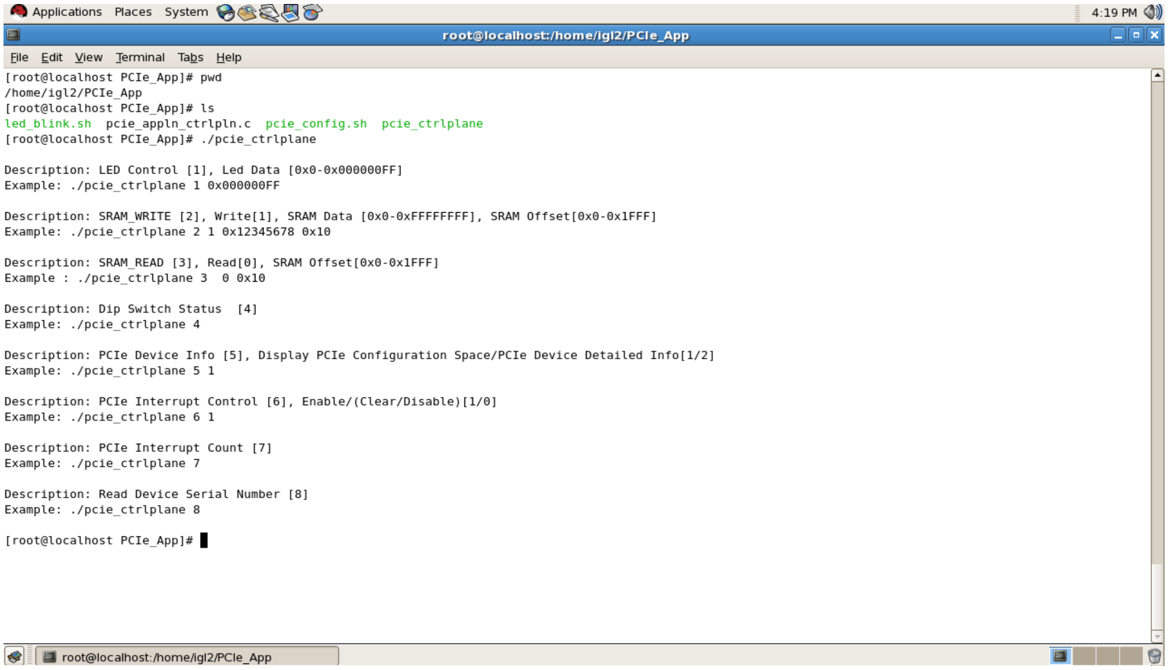
After successful compilation, Linux PCIe application utility `pcie_ctrlplane` creates in the same directory.

5. On Linux Command Prompt, run the `pcie_ctrlplane` utility as:

```
#./pcie_ctrlplane
```

Help menu is displayed, as shown in the following figure.

Figure 28 • **Linux PCIe Application Utility**


Applications Places System
4:19 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App

```
[root@localhost PCIe_App]# pwd
/home/igl2/PCIe_App
[root@localhost PCIe_App]# ls
led_blink.sh  pcie_appln_ctrlpln.c  pcie_config.sh  pcie_ctrlplane
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# █
```

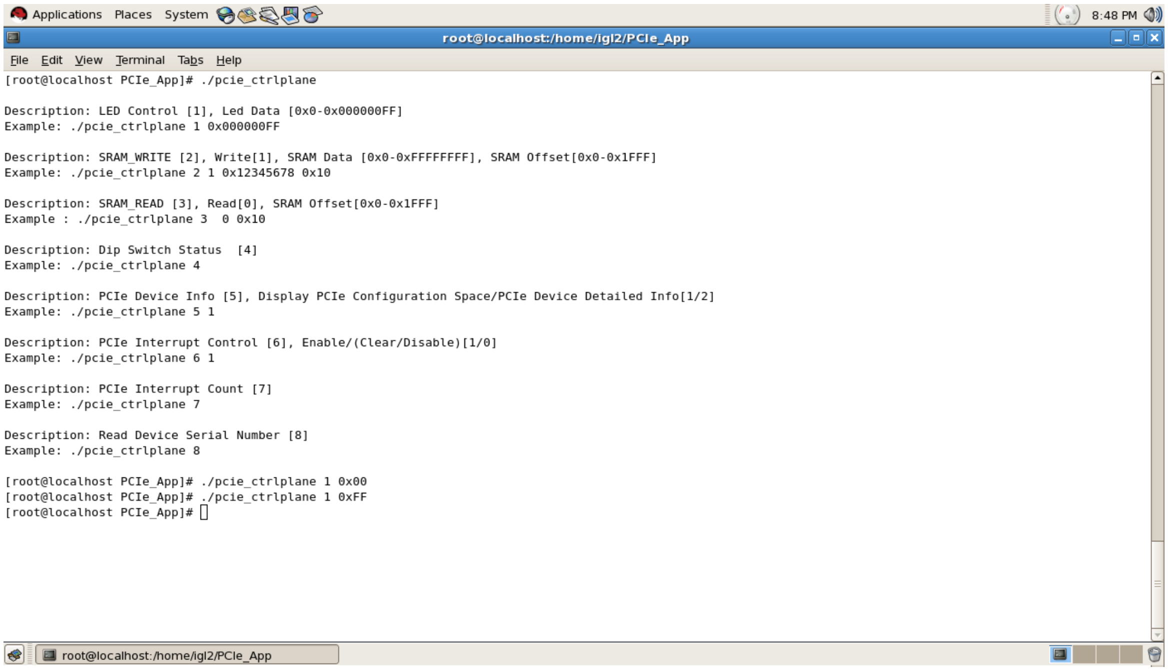
2.4.2.3 Execution of Linux PCIe Control Plane Features

2.4.2.3.1 LED Control

LED1 to LED8 is controlled by writing data to IGLOO2 LED control registers.

```
#./pcie_ctrlplane 1 0x000000FF [LED OFF]
#./pcie_ctrlplane 1 0x00000000 [LED ON]
```

Figure 29 • **Linux Command—LED Control**


Applications Places System
8:48 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 1 0x00
[root@localhost PCIe_App]# ./pcie_ctrlplane 1 0xFF
[root@localhost PCIe_App]# ▯
```

`led_blink.sh`, contains the shell script code to perform LED Walk ON where as `Ctrl+C` kills the shell script and LED Walk turns OFF.

```
#sh led_blink.sh
```

Run the `led_blink.sh` shell script using `sh` command.

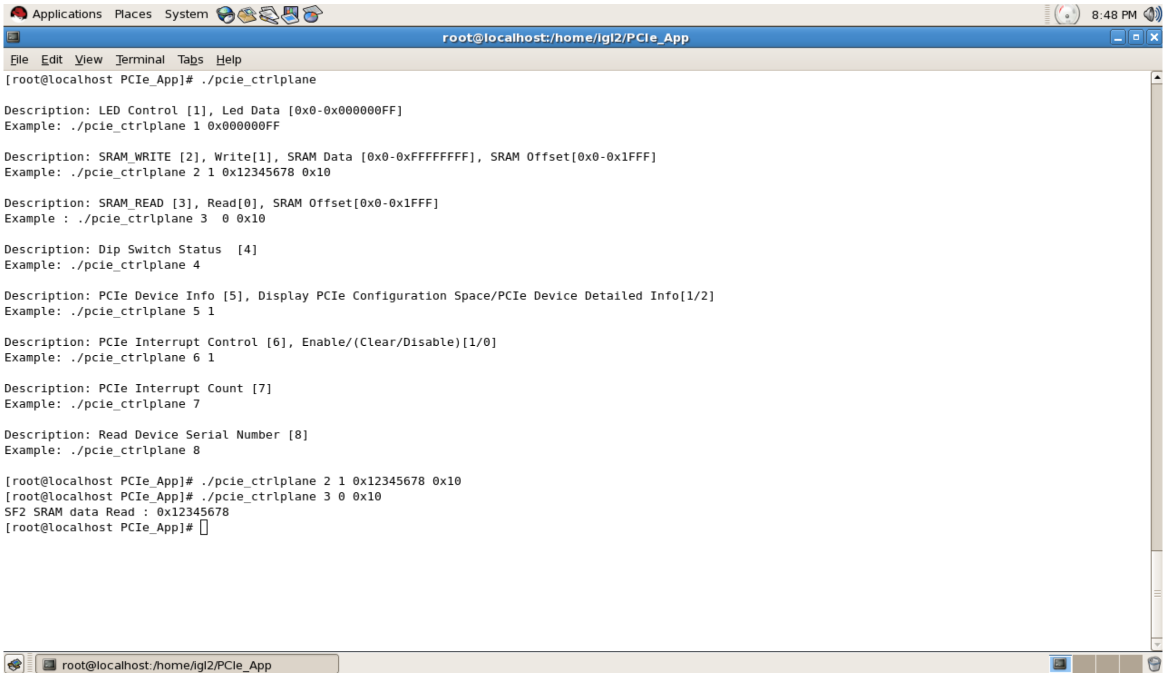
2.4.2.3.2 SRAM Read/Write

32 KB SRAM is accessible for IGLOO2 Evaluation Kit.

```
#./pcie_ctrlplane 2 1 0xFF00FF00 0x1000 [SRAM WRITE]
```

```
#./pcie_ctrlplane 3 0 0x1000 [SRAM READ]
```

Figure 30 • **Linux Command—SRAM Read/Write**



Applications Places System
8:48 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App

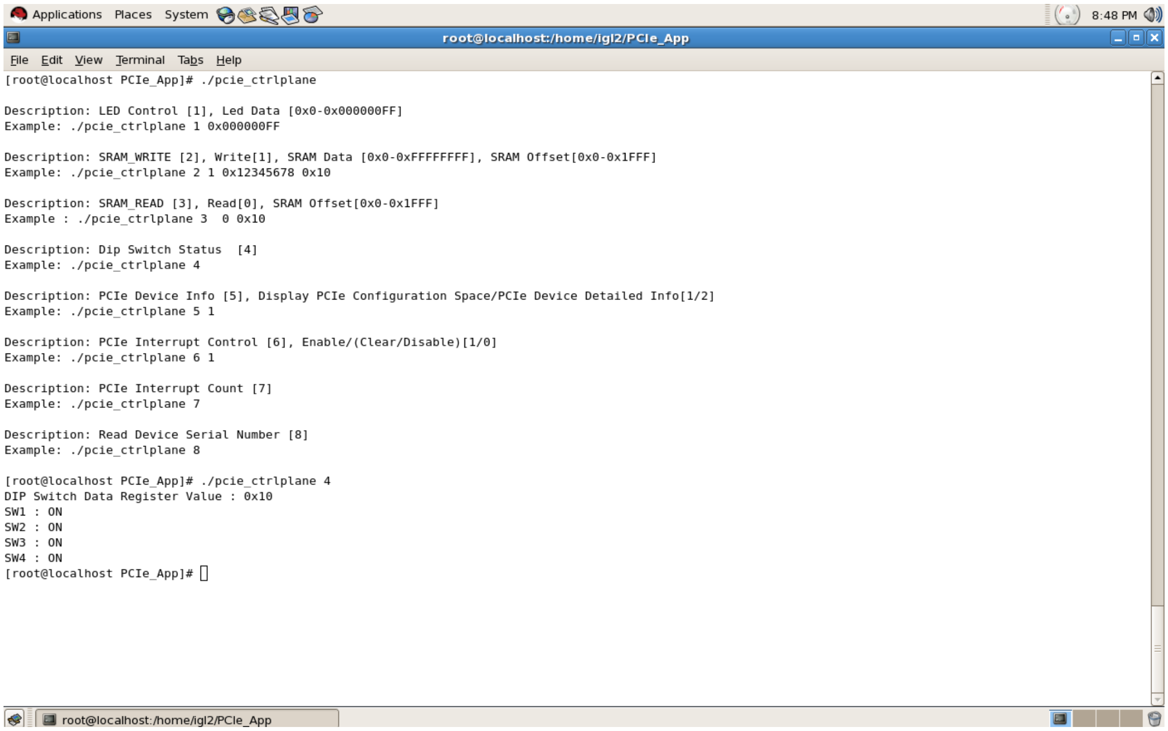
```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 2 1 0x12345678 0x10
[root@localhost PCIe_App]# ./pcie_ctrlplane 3 0 0x10
SF2 SRAM data Read : 0x12345678
[root@localhost PCIe_App]# 
```

2.4.2.3.3 DIP Switch Status

Dip switch on IGLOO2 Evaluation Kit board consists 4 electric switches to hold the device configurations. Linux PCIe utility reads the corresponding switches (ON/OFF) state.

```
#./pcie_ctrlplane 4 [DIP Switch Status]
```

Figure 31 • Linux Command—DIP Switch


Applications
Places
System
8:48 PM
root@localhost:/home/igl2/PCIe_App
File
Edit
View
Terminal
Tabs
Help
root@localhost:/home/igl2/PCIe_App

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 4
DIP Switch Data Register Value : 0x10
SW1 : ON
SW2 : ON
SW3 : ON
SW4 : ON
[root@localhost PCIe_App]# 
```

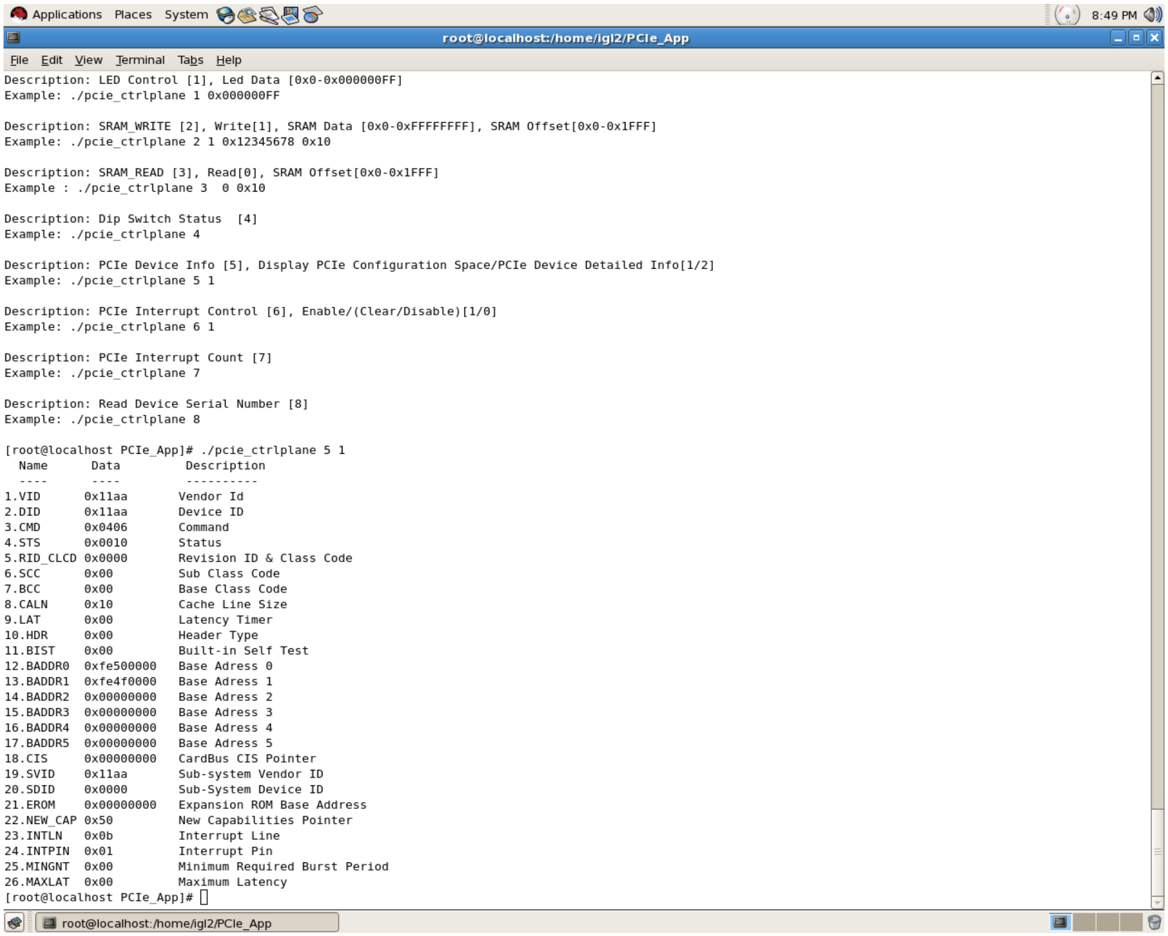
2.4.2.3.4 PCIe Configuration Space Display

PCIe Configuration Space contains the PCIe device data such as Vendor ID, Device ID, and Base Address 0.

Note: Root privileges are required to execute this command.

```
#./pcie_ctrlplane 5 1 [Read PCIe Configuration Space]
```

Figure 32 • **Linux Command—PCIe Configuration Space Display**


Applications
Places
System
8:49 PM
root@localhost:/home/igl2/PCIe_App
File
Edit
View
Terminal
Tabs
Help
root@localhost:/home/igl2/PCIe_App

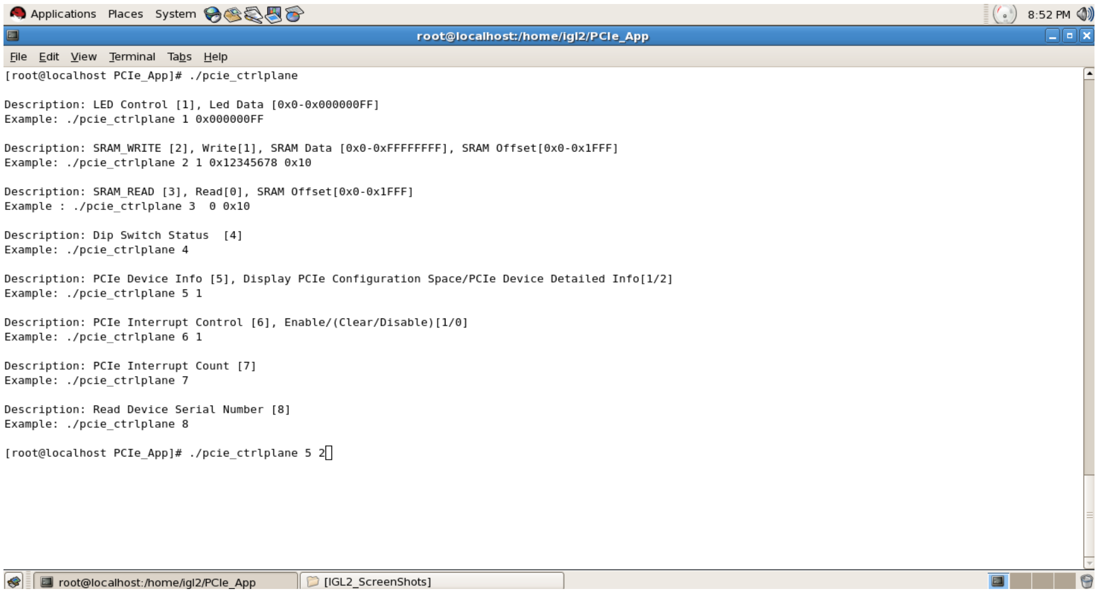
```
Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 5 1
  Name      Data         Description
  ----      ----          ----------
1.VID      0x11aa       Vendor Id
2.DID      0x11aa       Device ID
3.CMD      0x0406       Command
4.STS      0x0010       Status
5.RID_CLCD 0x0000       Revision ID & Class Code
6.SCC      0x00         Sub Class Code
7.BCC      0x00         Base Class Code
8.CALN     0x10         Cache Line Size
9.LAT      0x00         Latency Timer
10.HDR     0x00         Header Type
11.BIST    0x00         Built-in Self Test
12.BADDR0  0xfe500000   Base Adress 0
13.BADDR1  0xfe4f0000   Base Adress 1
14.BADDR2  0x00000000   Base Adress 2
15.BADDR3  0x00000000   Base Adress 3
16.BADDR4  0x00000000   Base Adress 4
17.BADDR5  0x00000000   Base Adress 5
18.CIS     0x00000000   CardBus CIS Pointer
19.SVID    0x11aa       Sub-system Vendor ID
20.SDID    0x0000       Sub-System Device ID
21.EROM    0x00000000   Expansion ROM Base Address
22.NEW_CAP 0x50         New Capabilities Pointer
23.INTLN   0x0b         Interrupt Line
24.INTPIN  0x01         Interrupt Pin
25.MINGNT  0x00         Minimum Required Burst Period
26.MAXLAT  0x00         Maximum Latency
[root@localhost PCIe_App]# 
```

2.4.2.3.5 PCIe Link Speed and Width

Root privileges are required to execute this command.

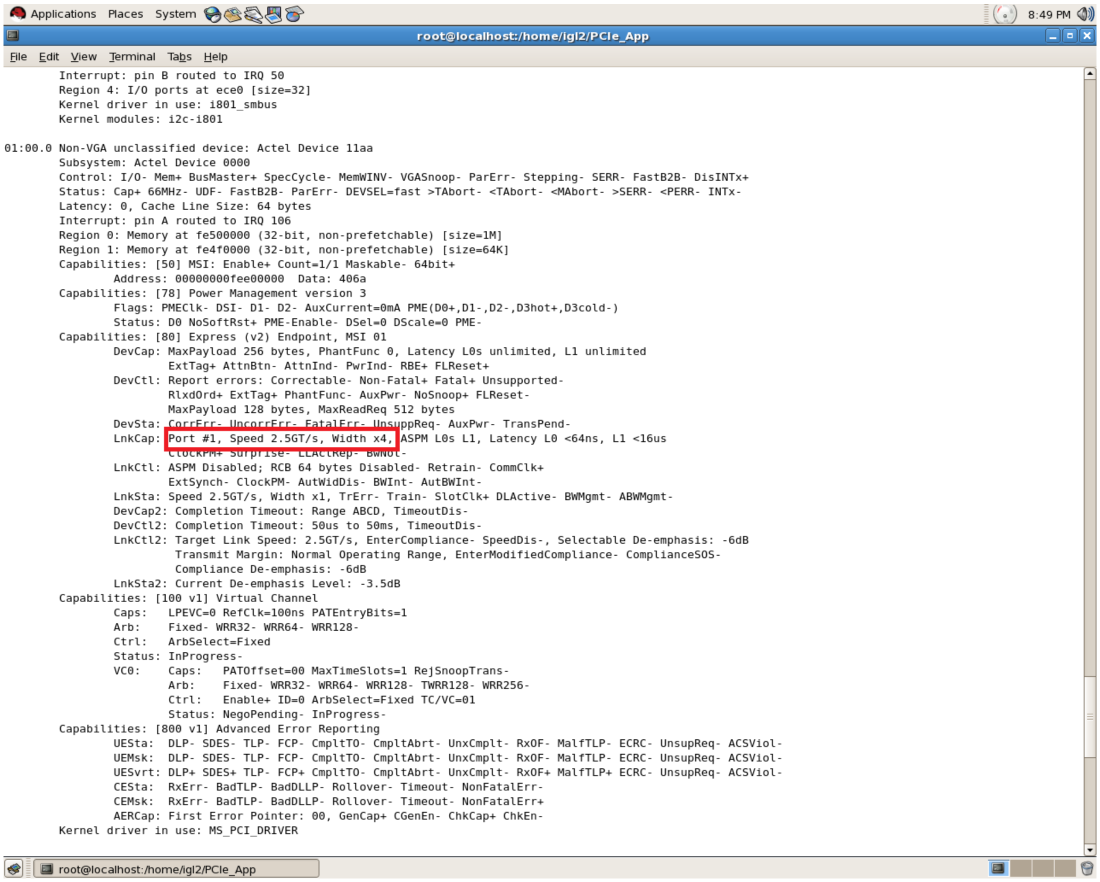
```
#./pcie_ctrlplane 5 2 [Read PCIe Link Speed and Link Width]
```

Figure 33 • Linux Command—PCIe Link Speed and Width


Applications Places System
8:52 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App
[IGL2_ScreenShots]

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 5 2
```

Figure 34 • Linux Command—PCIe Link Speed and Width


Applications Places System
8:49 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App

```
        Interrupt: pin B routed to IRQ 50
        Region 4: I/O ports at ece0 [size=32]
        Kernel driver in use: i801_smbus
        Kernel modules: i2c-i801

01:00.0 Non-VGA unclassified device: Actel Device 11aa
        Subsystem: Actel Device 0000
        Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
        Latency: 0, Cache Line Size: 64 bytes
        Interrupt: pin A routed to IRQ 106
        Region 0: Memory at fe500000 (32-bit, non-prefetchable) [size=1M]
        Region 1: Memory at fe4f0000 (32-bit, non-prefetchable) [size=64K]
        Capabilities: [50] MSI: Enable+ Count=1/1 Maskable- 64bit+
                Address: 00000000fee00000  Data: 406a
        Capabilities: [78] Power Management version 3
                Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold-)
                Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
        Capabilities: [80] Express (v2) Endpoint, MSI 01
                DevCap: MaxPayload 256 bytes, PhantFunc 0, Latency L0s unlimited, L1 unlimited
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset+
                DevCtl: Report errors: Correctable- Non-Fatal+ Fatal+ Unsupported-
                        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
                        MaxPayload 128 bytes, MaxReadReq 512 bytes
                DevSta: CorrErr- UncorrErr- FatalErr- UnsuppReq- AuxPwr- TransPend-
                LnkCap: Port #1, Speed 2.5GT/s, Width x4, ASPM L0s L1, Latency L0 <64ns, L1 <16us
                        ClockPM+ Surprise- LLActRep- BwNot-
                LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- Retrain- CommClk+
                        ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
                LnkSta: Speed 2.5GT/s, Width x1, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
                DevCap2: Completion Timeout: Range ABCD, TimeoutDis-
                DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-
                LnkCtl2: Target Link Speed: 2.5GT/s, EnterCompliance- SpeedDis-, Selectable De-emphasis: -6dB
                         Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
                         Compliance De-emphasis: -6dB
                LnkSta2: Current De-emphasis Level: -3.5dB
        Capabilities: [100 v1] Virtual Channel
                Caps:   LPEVC=0 RefClk=100ns PATEntryBits=1
                Arb:    Fixed- WRR32- WRR64- WRR128-
                Ctrl:   ArbSelect=Fixed
                Status: InProgress-
                VC0:    Caps:   PATOffset=00 MaxTimeSlots=1 RejSnoopTrans-
                        Arb:    Fixed- WRR32- WRR64- WRR128- TWRR128- WRR256-
                        Ctrl:   Enable+ ID=0 ArbSelect=Fixed TC/VC=01
                        Status: NegoPending- InProgress-
        Capabilities: [800 v1] Advanced Error Reporting
                UESta:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
                UEMsk:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
                UESvrt: DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+ ECRC- UnsupReq- ACSViol-
                CESta:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr-
                CEMsk:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr+
                AERCap: First Error Pointer: 00, GenCap+ CGenEn- ChkCap+ ChkEn-
        Kernel driver in use: MS_PCI_DRIVER
```

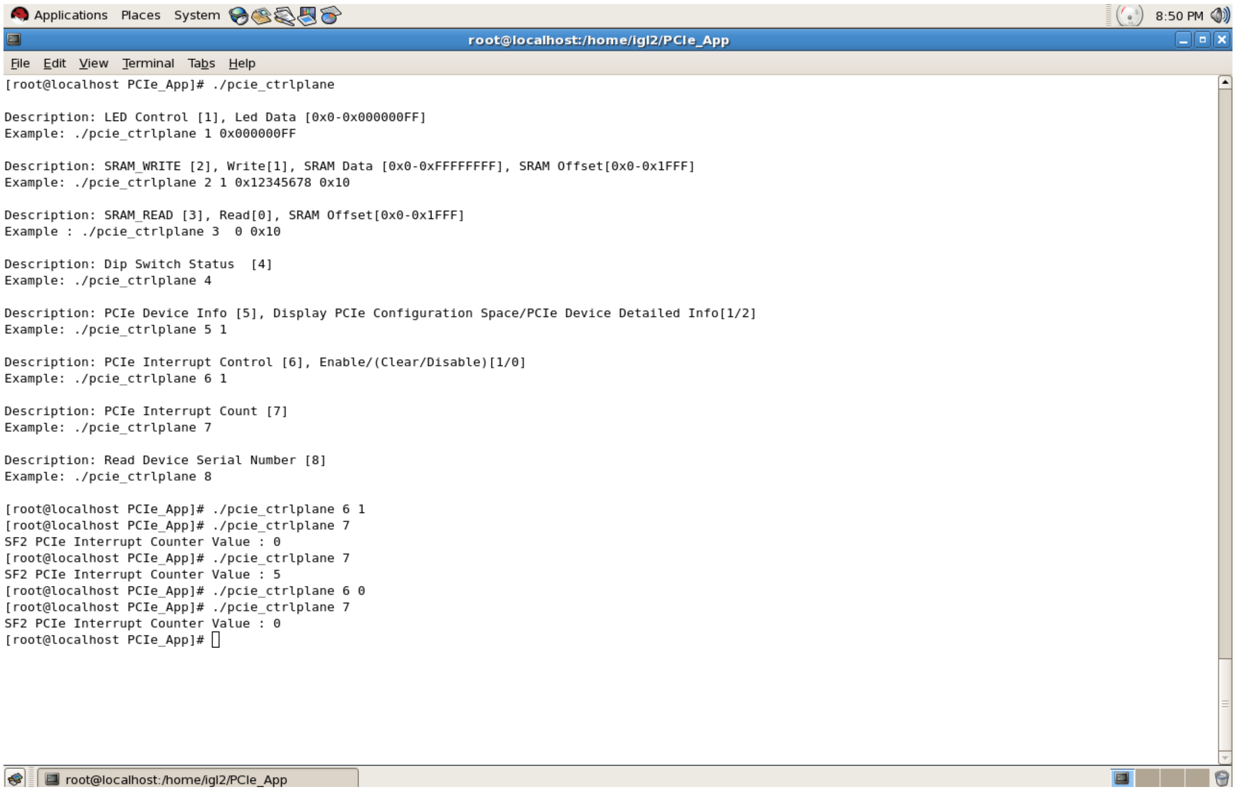
2.4.2.3.6 PCIe Interrupt Control (Enable/Disable) and Interrupt Counter

IGLOO2 Evaluation Kit enables/disables the message signaled interrupts (MSI) interrupts by writing data to its PCIe configuration space.

Interrupt counter holds the number of MSI interrupts got triggered by pressing the **SW4** push button.

```
#. /pcie_ctrlplane 6 0 [Disable Interrupts]

#. /pcie_ctrlplane 6 1 [Enable Interrupts]

#. /pcie_ctrlplane 7 [Interrupt Counter Value]
```

Figure 35 • **Linux Command—PCIe Interrupt Control**


Applications
Places
System
8:50 PM
root@localhost:/home/igl2/PCIe_App
File
Edit
View
Terminal
Tabs
Help
root@localhost:/home/igl2/PCIe_App

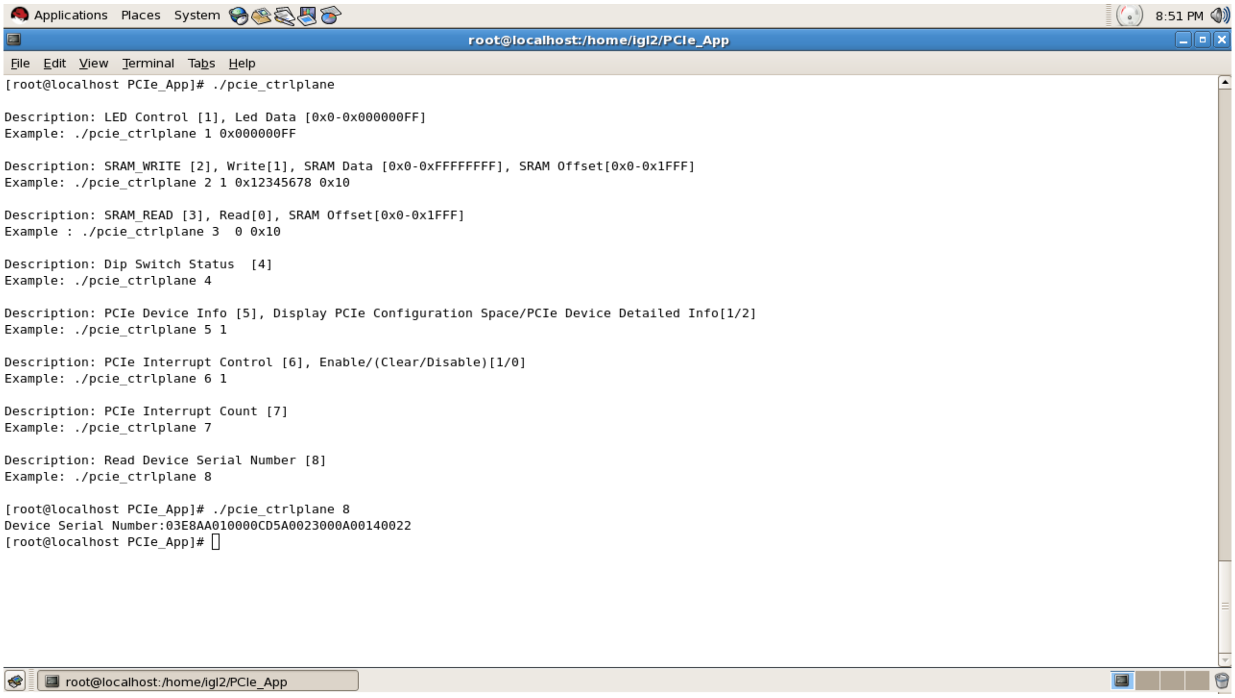
```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 6 1
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 0
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 5
[root@localhost PCIe_App]# ./pcie_ctrlplane 6 0
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 0
[root@localhost PCIe_App]# 
```

2.4.2.3.7 Read Device Serial Number

The IGLOO2 Evaluation Kit device serial number must be read by using the Linux PCIe application utility command.

```
#. /pcie_ctrlplane 8 [Read Device Serial Number]
```

Figure 36 • Linux Command—Read Device Serial Number


Applications Places System
8:51 PM
root@localhost:/home/igl2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/igl2/PCIe_App

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x1FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3], Read[0], SRAM Offset[0x0-0x1FFF]
Example : ./pcie_ctrlplane 3  0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

Description: Read Device Serial Number [8]
Example: ./pcie_ctrlplane 8

[root@localhost PCIe_App]# ./pcie_ctrlplane 8
Device Serial Number:03E8AA010000CD5A0023000A00140022
[root@localhost PCIe_App]# 
```

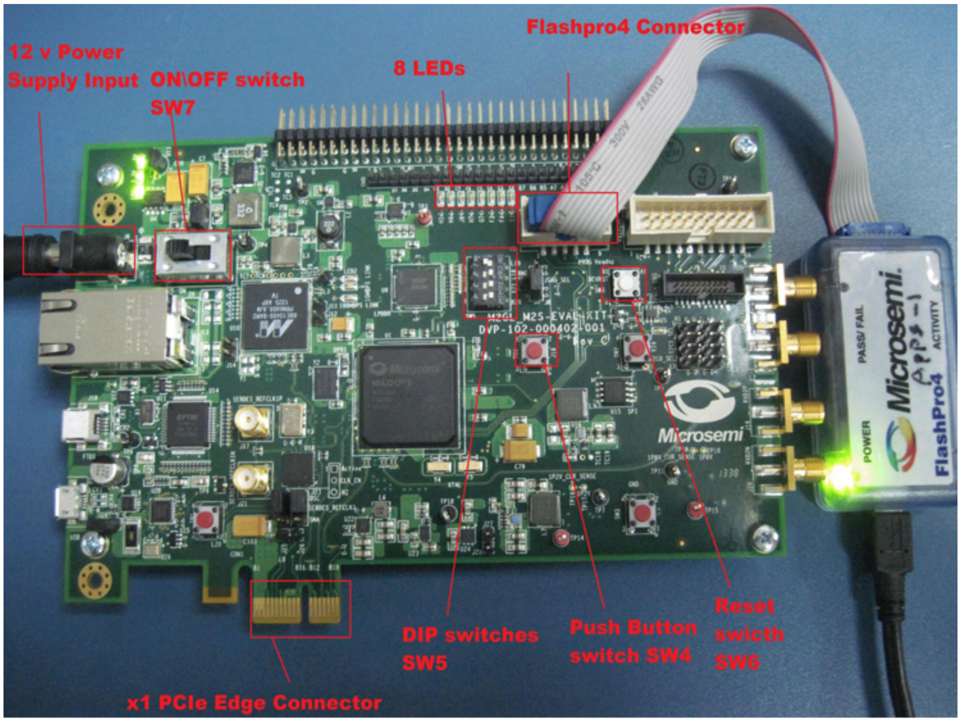
2.5 Conclusion

This demo describes how to access the PCIe endpoint and display the device serial number feature of IGLOO2 by implementing a low bandwidth control plane design with bus functional model (BFM) simulation. This demo provides a GUI for easy control of PCIe endpoint device through Microsemi PCIe drivers for windows platform and also provides a Linux PCIe application for easy control of PCIe endpoint device through the Linux PCIe Device Driver.

3 Appendix: IGLOO2 Evaluation Kit Board

The following figure shows IGLOO2 Evaluation Kit board.

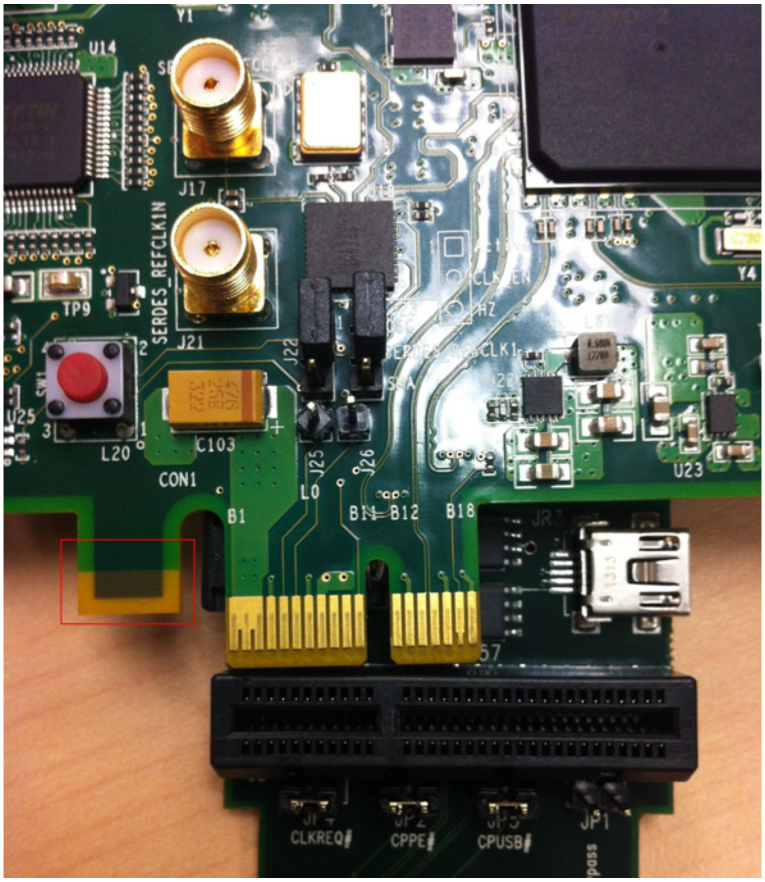
Figure 37 • **IGLOO2 Evaluation Kit Board**



4 Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop

The following figure shows how to line up the IGLOO2 Evaluation Kit PCIe connector with the adapter card slot.

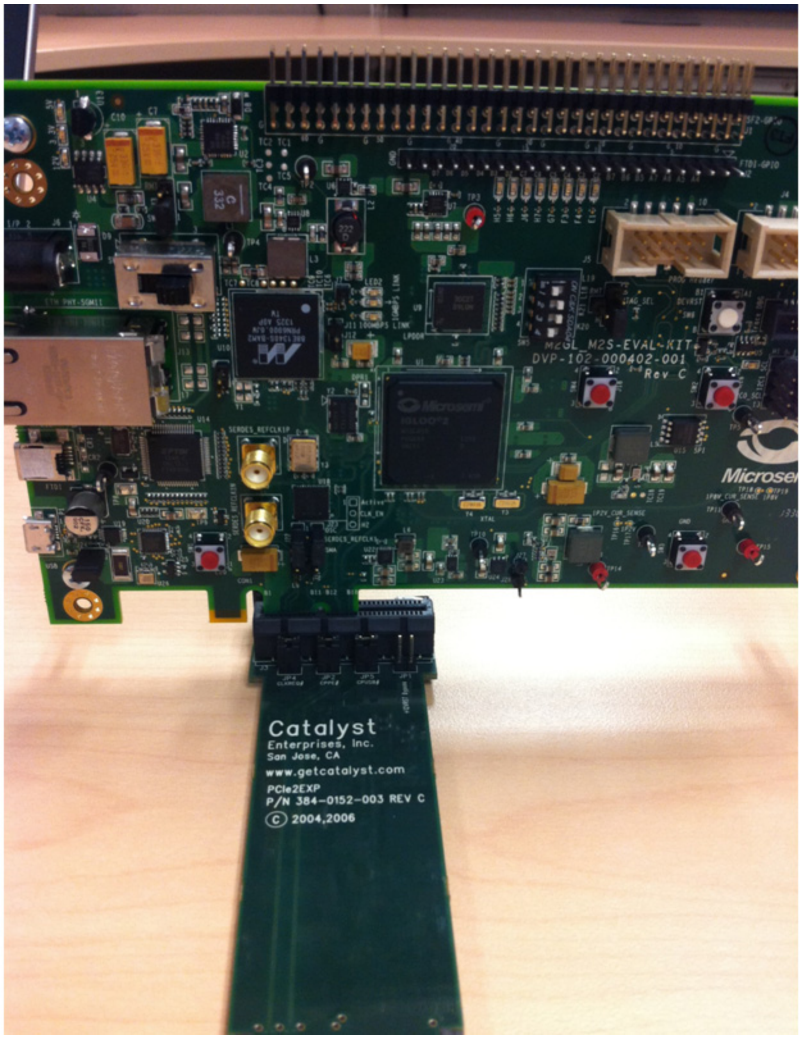
Figure 38 • **Lining-Up the IGLOO2 Evaluation Kit Board**



Note: The Notch (highlighted in red) does not go into the adapter card.

The following figure shows IGLOO2 Evaluation Kit PCIe connector inserted into the PCIe adapter card slot.

Figure 39 • **Inserting the IGLOO2 Evaluation Kit PCIe Connector**



The following figure shows the PCIe adapter card and the IGLOO2 Evaluation Kit connected to the laptop.

Figure 40 • **IGLOO2 Evaluation Kit Connected to the Laptop**