Contents

1 Revision History .......................................................... 1
  1.1 Revision 7.0 ........................................................................ 1
  1.2 Revision 6.0 ........................................................................ 1
  1.3 Revision 5.0 ........................................................................ 1
  1.4 Revision 4.0 ........................................................................ 1
  1.5 Revision 3.0 ........................................................................ 1
  1.6 Revision 2.0 ........................................................................ 1
  1.7 Revision 1.0 ........................................................................ 1

2 IGLOO2 FPGA PCIe Control Plane with Device Serial Number Demo ............. 2
  2.1 Design Requirements .......................................................... 2
  2.2 Demo Design ....................................................................... 3
    2.2.1 Features ................................................................------- 3
    2.2.2 Description ................................................................... 4
  2.3 Setting-up the Demo Design .................................................... 6
    2.3.1 Board Setup ................................................................... 7
    2.3.2 Programming the IGLOO2 Board ..................................... 7
    2.3.3 Connecting the Evaluation Kit Board to the Host PC ........... 9
  2.4 Running the Demo Design ...................................................... 10
    2.4.1 Running the Demo Design on Windows ............................ 10
    2.4.2 Running the Demo Design on Linux ................................. 17
  2.5 Conclusion .......................................................................... 26

3 Appendix: IGLOO2 Evaluation Kit Board ........................................... 27

4 Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop ...................... 28
Figures

Figure 1   PCIe Control Plane Demo Top-Level Block Diagram  .................................................. 2
Figure 2   Demo Design Files Top-Level Structure ................................................................. 3
Figure 3   PCIe Control Plane Block Diagram ....................................................................... 4
Figure 4   SERDES BFM Simulation ...................................................................................... 5
Figure 5   Simulation Result with GPIO_OUT Signals ........................................................ 6
Figure 6   FlashPro New Project ............................................................................................ 7
Figure 7   FlashPro Project Configured ................................................................................ 8
Figure 8   FlashPro Program Passed ..................................................................................... 8
Figure 9   IGLOO2 Evaluation Kit Setup for Host PC ............................................................. 9
Figure 10  Device Manager .................................................................................................10
Figure 11  Update Driver Software .....................................................................................11
Figure 12  Browse for Driver Software ...............................................................................11
Figure 13  Browse for Driver Software Continued ..............................................................12
Figure 14  Windows Security ...............................................................................................12
Figure 15  Successful Driver Installation ..........................................................................12
Figure 16  GUI Installation .................................................................................................13
Figure 17  Successful GUI Installation ..............................................................................13
Figure 18  Device Manager—PCIe Device Detection ..........................................................14
Figure 19  PCIe Demo GUI .................................................................................................14
Figure 20  Device Info ........................................................................................................15
Figure 21  Demo Controls ..................................................................................................15
Figure 22  Demo Controls—Continued ..............................................................................16
Figure 23  Configuration Space ..........................................................................................16
Figure 24  PCIe BAR1 Memory Access .............................................................................17
Figure 25  PCIe Device Detection .....................................................................................17
Figure 26  Edit board.h File ...............................................................................................18
Figure 27  PCIe Device Driver Installation ........................................................................19
Figure 28  Linux PCIe Application Utility .........................................................................20
Figure 29  Linux Command—LED Control ......................................................................20
Figure 30  Linux Command—SRAM Read/Write ...............................................................21
Figure 31  Linux Command—DIP Switch ..........................................................................22
Figure 32  Linux Command—PCIe Configuration Space Display ......................................23
Figure 33  Linux Command—PCIe Link Speed and Width ................................................24
Figure 34  Linux Command—PCIe Link Speed and Width ................................................24
Figure 35  Linux Command—PCIe Interrupt Control .......................................................25
Figure 36  Linux Command—Read Device Serial Number ...............................................26
Figure 37  IGLOO2 Evaluation Kit Board ...........................................................................27
Figure 38  Lining-Up the IGLOO2 Evaluation Kit Board ......................................................28
Figure 39  Inserting the IGLOO2 Evaluation Kit PCIe Connector .........................................29
Figure 40  IGLOO2 Evaluation Kit Connected to the Laptop ..............................................30
Tables

Table 1  Design Requirements ................................................................. 2
Table 2  IGLOO2 FPGA Evaluation Kit Jumper Settings .......................... 6
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>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>IGLOO2 Evaluation Kit¹:</td>
<td>Rev D or later</td>
</tr>
<tr>
<td>– 12 V adapter</td>
<td></td>
</tr>
<tr>
<td>– FlashPro4 programmer</td>
<td></td>
</tr>
<tr>
<td>– USB A to Mini-B cable</td>
<td></td>
</tr>
<tr>
<td>Host PC or Laptop with an available PCIe 2.0 Gen 1 or Gen 2 compliant slot</td>
<td>64-bit Windows 7 OS or 64-bit RedHat Linux OS (Kernel Version: 2.6.18-308)</td>
</tr>
<tr>
<td>Express Card slot and PCIe Express card adapter (for Laptop only)</td>
<td>–</td>
</tr>
</tbody>
</table>

¹Rev D or later or later
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

```plaintext
<download_folder>
  M2GL_PCIE_Control_Plane_DSN_DF
    GUI
    LiberoProject
    Linux_64bit
    Programming File
    Source Files
    Windows_64bit
    Readme.txt
```

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.
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**

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Pin (From)</th>
<th>Pin (To)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J22</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J23</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J24</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J8</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J3</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
</tbody>
</table>

3. Connect the power supply to the J6 connector.
2.3.1 Board Setup
Snapsots 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.

![FlashPro New Project](image)

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

![Update Driver Software](image1)

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

![Browse for Driver Software](image2)
3. Browse the drivers folder:
M2GL_PCIE_Control_Plane_DSN_DFI\Windows_64bit\Drivers\PCIe_Demo and click **Next**, as shown in the following figure.

*Figure 13 • Browse for Driver Software Continued*

![Browse for Driver Software](image1.png)

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](image2.png)

*Figure 15 • Successful Driver Installation*

![Successful Driver Installation](image3.png)
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_DFIGUI
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.

![GUI Installation](image1.jpg)

4. Click Finish to complete the installation.

![Successful GUI Installation](image2.jpg)

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—PCle Device Detection

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
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*

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*

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*

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

*Figure 23 • Configuration Space*
10. Click the **PCIe Read/Write** tab to perform read and writes to LSRAM memory through **BAR1** space. Click **Read** to read the 4 KBytes 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*

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.

   ```bash
   # lspci
   *
   *Figure 25 • PCIe Device Detection*
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
12. After successful Linux PCIe device driver installation, check `/dev/MS_PCI_DEV` is created by using the following Linux command:

```
# /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
```

---

**Figure 27 • PCIe Device Driver Installation**

![Image of PCIe Device Driver Installation](image)
Help menu is displayed, as shown in the following figure.

**Figure 28 • Linux PCIe Application Utility**

### 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**
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.

```bash
#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.

```bash
#./pcie_ctrlplane 2 1 0xFF00FF00 0x1000 [SRAM WRITE]
#./pcie_ctrlplane 3 0 0x1000 [SRAM READ]
```

*Figure 30 • Linux Command—SRAM Read/Write*
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*
2.4.2.3.4 PCIe Configuration Space Display

PCle 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

```bash
root@localhost:~# ./pcie_ctrlplane 5 1
Name            Data            Description
---------------------------------------------------------------------------
1. VID 0x0800 Vendor ID
2. D_ID 0x0000 Device ID
3. CMD 0x0000 Command
4. STS 0x010 Status
5. REV_ID 0x0000 Revision ID & Class Code
6. S_CC 0x00 Sub Class Code
7. B_CC 0x00 Base Class Code
8. C_LSN 0x10 Cache Line Size
9. LAT 0x00 Latency Timer
10. H_TMP 0x00 Header Type
11. BST 0x00 Built-in Self Test
12. ADS 0x00000000 Base Address 0
13. ADS 0x00000000 Base Address 1
14. ADS 0x00000000 Base Address 2
15. ADS 0x00000000 Base Address 3
16. ADS 0x00000000 Base Address 4
17. ADS 0x00000000 Base Address 5
18. CIS 0x100000000 CardBus CIS Pointer
19. SVID 0x00 Sub-system Vendor ID
20. S_ID 0x00 Sub-System Device ID
21. RMM 0x00000000 Expansion ROM Base Address
22. H_CIP 0x00000000 New Capability Pointer
23. INTR 0x00 Interrupt Line
24. INTCON 0x0000 Interrupt Command
25. MAX_PMT 0x00 Maximum Required Burst Period
26. SKPLAT 0x00 Maximum Latency
```
2.4.2.3.5 PCIe Link Speed and Width

Root privileges are required to execute this command.

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

**Figure 33 • Linux Command—PCIe Link Speed and Width**

```bash
root@localhost:~# pctl_app
```

```bash
Description: LED Control [3], Link Data [00-00000000]
Example: ./pcie_ctrlplane 1 0000000000
Description: SRAM_WRITE [2], Register[1], SRAM Data [00-00000000]
Example: ./pcie_ctrlplane 2 0 00000000
Description: GPIO [1], Test[0], GPIO Offset[00-00000000]
Example: ./pcie_ctrlplane 1 0
Description: Dip Switch Status [4]
Example: ./pcie_ctrlplane 4
Description: PCIe Device Info [5], Display PCIe Configuration Space Device Details Info[1/2]
Example: ./pcie_ctrlplane 5 1
Description: PCIe Interrupt Control [6], Enable[1/0]
Example: ./pcie_ctrlplane 6 1
Description: Read Device Serial Number [7]
Example: ./pcie_ctrlplane 7
```

**Figure 34 • Linux Command—PCIe Link Speed and Width**

```bash
root@localhost:~# pctl_app
```

```bash
Interrupt: pin 8 routed to IRQ 30
Device: 0x0000:00:03.0 [PCI bridge]
Status: Capable, 66MHz, 32bit, 256K
devbus:0000:00:03.0 [PCI bridge]: Capable, 66MHz, 32bit, 256K
```

**Captions**

- [Figure 33](#)
- [Figure 34](#)
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**
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

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.
The following figure shows IGLOO2 Evaluation Kit board.

*Figure 37 • IGLOO2 Evaluation Kit Board*
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