IGLOO2 FPGA PCIe HPMS HPDMA - Libero SoC v11.6

DG0585 Demo Guide

October 2015

Microsemi

Revision History

Date	Revision	Change
21 October 2015	2	Second release
13 February 2015	1	First Release

Confidentiality Status

This is a non-confidential document.

Table of Contents

Preface

About this document

This demo is for IGLOO$^{®}$2 field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

Intended Audience

IGLOO2 devices are used by:
- FPGA designers
- System-level designers

References

The following references are used in this document:
- *UG0448: IGLOO2 FPGA High Performance Memory Subsystem User Guide*
- *UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide*
- *DG0532: IGLOO2 FPGA PCIe Control Plane with Device Serial Number Demo Guide*
- *DG0517: SmartFusion2 and IGLOO2 PCIe Data Plane Demo using 2 Channel Fabric DMA Demo Guide*

Microsemi Publications

Refer to the following web page for a complete and up-to-date listing of IGLOO2 device documentation: *http://www.microsemi.com/products/fpga-soc/fpga/igloo2-fpga*.

PCIe Data Plane Demo Using HPMS HPDMA

Introduction

This demo describes the usage of the following embedded features of the IGLOO2 devices:

- PCIe controller
- High-performance direct memory access (HPDMA) controller
- Microcontroller subsystem double data rate (MDDR) controller

The demo design uses all of these embedded features and few FPGA resources. This demo architecture is resource and power efficient, but does not demonstrate the highest performance of the PCIe link.

The demo design also explains how to use the IGLOO2 FPGA embedded PCI Express feature as a data plane interface. The demo design accesses the PCIe end point (EP) available in the IGLOO2 device from a host PC, and initiates the DMA transactions using the PCIe interface. In this demo, the DMA transfers are performed using the IGLOO2 embedded high-performance memory subsystem (HPMS) HPDMA controller. A host PC `PCIe_Demo` application is provided for setting up and initiating the DMA transactions between the IGLOO2 PCIe end point and host PC. The demo provides host PC device drivers for the IGLOO2 PCIe end point and also PCIe features are used as control plane. Refer to the *DG0532: IGLOO2 FPGA PCIe Control Plane with Device Serial Number Demo Guide* for more information on PCIe control plane.

Microsemi® provides the following PCIe data plane demos for the IGLOO2 devices:

- PCIe Data Plane Demo using HPMS HPDMA (current demo): This demo describes the medium throughput data transfer between the PCIe and MDDR.
- *DG0517: SmartFusion2 and IGLOO2 PCIe Data Plane Demo using 2 Channel Fabric DMA Demo Guide*: This demo describes the high throughput data transfer between PCIe, MDDR, and a large SRAM (LSRAM).

The high-speed serial interface available in the IGLOO2 devices provides a fully hardened PCIe EP implementation, and is compliant to the PCIe Base Specification Revision 2.0 and 1.1. For more information on high-speed serial interface, refer to the *UG0447: IGLOO2 and SmartFusion2 High Speed Serial Interfaces User Guide*.

Design Requirements

Table 1 lists the design requirements for running the PCIe data plane demo using HPMS HPDMA.

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
IGLOO2 Evaluation Kit: • 12 V adapter • FlashPro4 Programmer • USB A to Mini-B cable	Rev C or later
Host PC or Laptop (8 GB RAM)	Windows 64-bit Operating System
Host PC with an available PCIe 2.0 compliant slot (x1 or greater)	–
Software Requirements	
Libero® System-on-Chip (SoC)	v11.6
FlashPro Programming Software	v11.6
PCIe Demo Application	–

Demo Design

Introduction

The demo design files are available for download from the following path in the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2gl_dg0585_hpms_hpdma_liberov11p6_df*

The demo design files include:

- Libero SoC project
- Programming files
- Drivers
- Readme.txt file

Refer to the `Readme.txt` file for the complete directory structure.

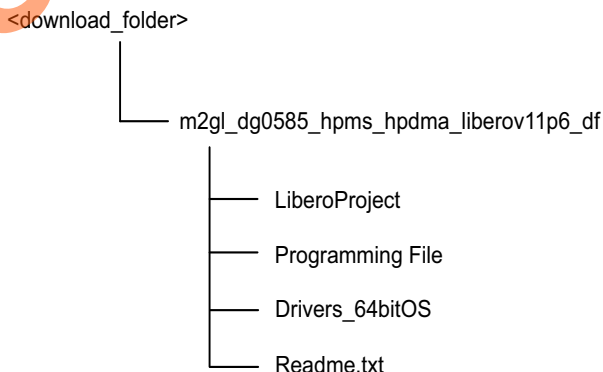Figure 1 shows the top-level structure of the design files.

Figure 1 • **Demo Design Files Top-Level Structure**

Figure 2 shows the demo design. The PCIe_Demo application from the host PC initiates the DMA transfers. The embedded PCIe core in the IGLOO2 device receives these commands and initiates the appropriate AHB transactions. The command decoder logic in the FPGA fabric decodes the AHB transaction commands and provides the inputs to CoreHPDMACtrl IP, which initializes the HPDMA controller to perform the DMA transactions between the low power DDR (LPDDR) memory (available on the IGLOO2 Evaluation Kit), the large static random access memory (LSRAM) (available on the IGLOO2 device), and the host PC. The counter logic implemented in the FPGA fabric calculates the throughput of each DMA transfer for displaying in the application GUI.

The driver on the host PC creates a buffer and passes the physical address of that buffer to the IGLOO2 device. HPDMA uses this address to perform the DMA transfers to the host PC.

The following options are available in the application for the DMA data transactions:

- **LPDDR to Host PC:** In this DMA transfer, HPDMA reads the data from the external LPDDR memory through MDDR and writes to a host PC buffer using the fabric interface controller (FIC_0) and PCIe. The DMA size can be 64 KB, 256 KB, or 1 MB.

- **Host PC to LPDDR:** In this DMA transfer, HPDMA reads the data from the host PC buffer using FIC_0 and PCIe and writes to an external LPDDR memory using MDDR. The DMA size can be 64 KB, 256 KB, or 1 MB.

- **Fabric LSRAM to LPDDR:** In this DMA transfer, HPDMA reads the data from fabric LSRAM using FIC_0 and writes to the external LPDDR memory using MDDR. The DMA size can be 16 KB or 32 KB.

- **LPDDR to Fabric LSRAM:** In this DMA transfer, HPDMA reads the data from the external LPDDR memory using MDDR and writes to the fabric LSRAM using FIC_0. The DMA size can be 16 KB or 32 KB.
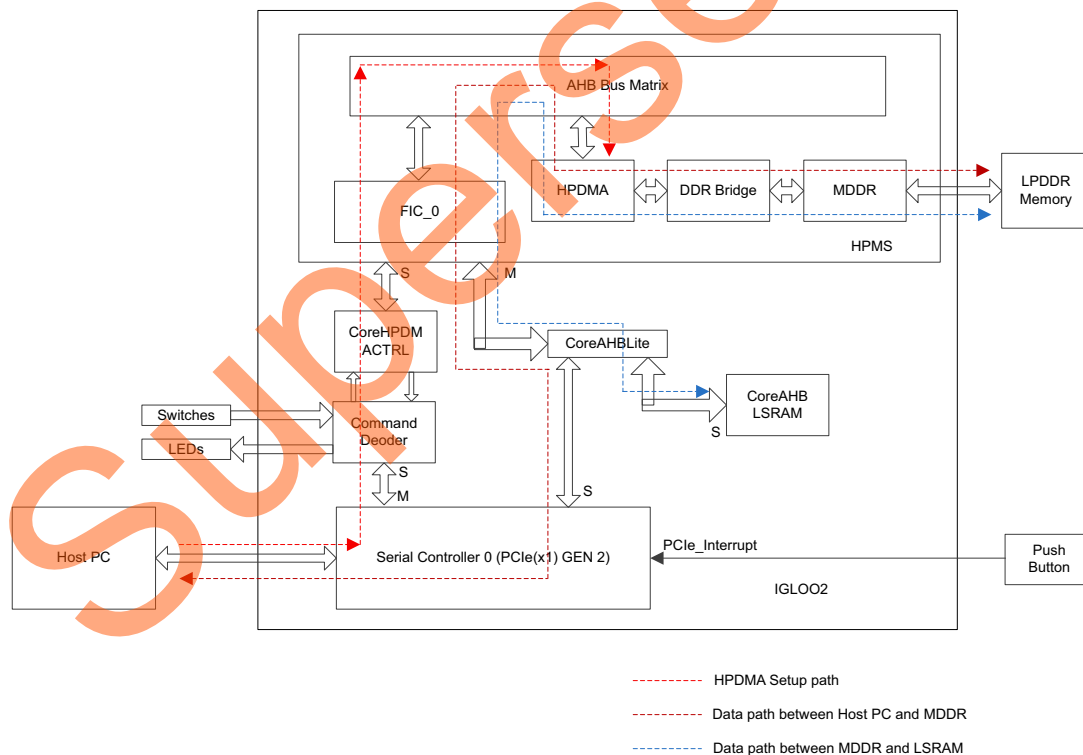

Figure 2 • **PCIe Data Plane Demo Block Diagram**

In this demo, the IGLOO2 MDDR is configured for accessing the LPDDR memory in x16 mode to access the LPDDR memory on the IGLOO2 Evaluation Kit board. The MDDR clock is configured to 160 MHz(320 Mbps DDR) with a 80 MHz DDR_FIC clock for an aggregate memory bandwidth of 640 MBps approximately. The PCIe AHB interface clock and fabric clocks are configured to run at 80 MHz.

Demo Design Features

Following are the demo design features:

- DMA data transfers between the host PC and external LPDDR memory using the HPDMA controller.
- DMA data transfers between the external LPDDR memory and the IGLOO2 LSRAM using the HPDMA controller.
- Displays throughput for each DMA data transfer.
- Continuous DMA transfers to observe the throughput variations.
- Displays the PCIe link enable/disable, negotiated link width, and the link speed.
- Displays the position of DIP switches on the IGLOO2 Evaluation Kit board.
- Displays the PCIe Configuration Space Controlling LEDs on the board through the PCIe_Demo application.
- Read and write operations on scratchpad register in the FPGA fabric.
- Displays the interrupt counts by interrupting the host PC using the Push button on the IGLOO2 Evaluation Kit board.

Demo Design Description

In this demo design, the PCIe link is configured to operate in Gen2 mode using one lane (x1) and a throughput of more than 400 MBps can be achieved. The following four types of data transfers are supported and the sections describe the process of each data transfer:

- PC Memory to DDR Memory
- DDR Memory to Fabric SRAM
- Fabric SRAM to DDR Memory
- Throughput Calculation

PC Memory to DDR Memory

A data transfer from PC memory to the LPDDR device happens in the following sequence:

1. CoreHPDMA is setup over the PCIe link based on the GUI settings.
2. CoreHPDMA sets up the HPMS HPDMA.
3. HPDMA initiates an AHB read transaction through the HPMS fabric interface (FIC) and to the PCIe AHB interface.
4. The PCIe core sends a memory read (MRd) transaction layer packet (TLP) to the host PC.
5. The host PC returns a completion with data (CplD) TLP to the PCIe link.
6. This return data completes the AHB read initiated by HPDMA controller.
7. This data is then written to the LPDDR memory using the MDDR subsystem.
8. HPDMA repeats this process (from step 1 to 7) until the transfer size set in the host PC GUI is completed. For transfers that require more than one buffer descriptor size (64 KB size) of HPDMA, the fabric logic is implemented to reset the buffer descriptor and re-start the transfer process.

This read operation is effective. However, it is not optimal for a PCIe link as the AHB bus of the HPMS does not support a split transaction.

Note: For optimal data transfers from the host PC memory to DDR, use the AXI interface of the PCIe link and a fabric-based DMA controller. The AXI supports split transactions and allows the PCIe link to send a larger read request. While waiting for the read request, the AXI bus can move on to the next transaction. Multiple read requests can be initiated that allow the PCIe link to operate efficiently with read completion data returning.

DDR Memory to Fabric SRAM

The FPGA design has a 32 KB fabric SRAM block. Data transfer from the LPDDR device to the fabric SRAM happens in the following sequence:

1. CoreHPDMA is setup over the PCIe link based on the GUI settings.

2. CoreHPDMA sets up the HPMS HPDMA.

3. HPDMA initiates an AHB read transaction of the LPDDR using the DDR controller of the HPMS.

4. The data is written to the fabric SRAM as an AHB write transaction through the FIC.

5. HPDMA repeats this process (from step 1 to 4) until the transfer size set in the host PC GUI is completed.

Fabric SRAM to DDR Memory

A data transfer from the LPDDR device to PC memory happens in the following sequence:

1. HPDMA is setup over the PCIe link based on the GUI settings.

2. HPDMA initiates an AHB read transaction through the HPMS FIC and to the fabric SRAM.

3. The data is written to the LPDDR Memory through the MDDR subsystem.

4. HPDMA repeats this process (from step 1 to 3) until the transfer size set in the host PC GUI is completed.

Throughput Calculation

This demo implements a timer to measure the throughput of DMA transfers. The throughput measured includes all the overhead of AHB, PCIe, and HPDMA controller transactions. The procedure for measuring throughput is:

1. Setup the CoreHPDMA controller for the complete transfer.

2. Start the timer and HPDMA controller, which initiates the data transfer for the requested number of bytes.

3. Wait until the DMA transfer is completed.

4. Record the number of clock cycles used for steps 2 and 3.

To arrive at a realistic system performance, the throughput calculation takes into account all the overheads during a transfer. The throughput formula is as shown below:

Throughput = Transfer Size (Bytes) / (Number of clock cycles taken for a transfer * Clock Period)

EQ 1

Simulating the Design

The design supports the BFM_PCIe simulation level to communicate with the serializer/de-serializer interface (SERDESIF) block using the master AXI/AHB bus interface. Although, the serial communication does not go through the SERDESIF block, the BFM_PCIe simulation scenario allows validating the fabric interface connections. The `SERDESIF_0_user.bfm` file under the *<Libero project>/simulation* folder has the BFM commands to trigger the DMA transactions.

To run the simulation,

1. Double-click **Simulate** under **Verify Pre-Synthesized Design** in the **Design Flow** window of Libero project.

2. ModelSim runs the design for about 515 us. The **ModelSim Transcript** window displays the BFM commands and the BFM simulation is completed without errors as shown in Figure 3.

```
# Debug: At Time 510355845 ps testbench.u_dram_16_I0.Control_Logic:WRITE: BAnk = 0, Col = 0038
# At Time 510357408 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 036, Data = 6a69
# At Time 510358970 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 037, Data = 6c6b
# At Time 510360533 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 038, Data = 6e6d
# At Time 510362095 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 039, Data = 706f
# At Time 510363658 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03a, Data = 7271
# At Time 510365220 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03b, Data = 7473
# At Time 510366783 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03c, Data = 7675
# At Time 510368345 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03d, Data = 7877
# At Time 510369908 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03e, Data = 7a79
# At Time 510371470 ps testbench.u_dram_16_I0.Write_FIFO_DM_Mask_Logic:WRITE: BAnk = 0, Row = 0000, Col = 03f, Data = 7c7b
# Debug: At Time 511346470 ps testbench.u_dram_16_I0.Control_Logic:PRE: ADDR[10] = 0, BAnk = 00
# testbench.u_dram_16_I0.Power_down_chk: at Time 511371470 ps Entering Power-Down Mode
# BFM:85:poll w 20008018 00000001 at 512228 ns
# BFM: Data Read 20008018 00000001 MASK:ffffffff at 512240.010000ns
# BFM:87:read w 20008014 at 512246 ns
# BFM:91:return
# SFM: Data Read 20008014 000000c3 at 512258.760000ns
# SFM: Data Read 20008018 00000001 at 512265.010000ns
##########################################################################
#
# SERDESIF_0 BFM Simulation Complete - 39 Instructions - NO ERRORS
#
##########################################################################
#
```

Figure 3 •   **Modelsim Transcript Messages**

Figure 4 shows the **Waveform** window for the DMA transactions between PCIe and LPDDR. It shows the commands from the SERDES AHB master, command decoder signals, and data transfer on SERDES AHB slave and MDDR interfaces.
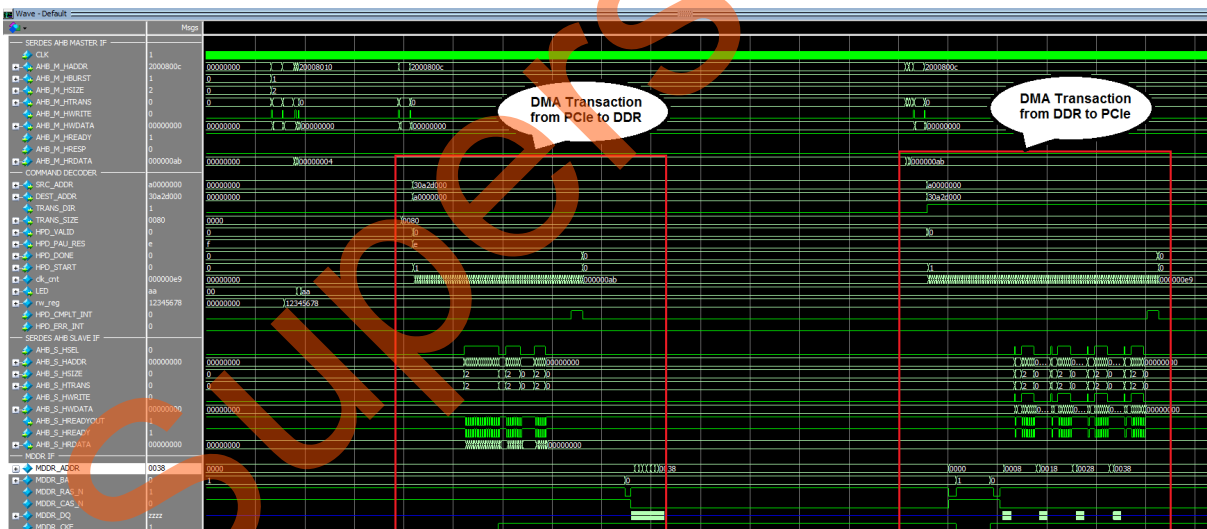

Figure 4 •   **DMA Transactions between PCIe and LPDDR**

Figure 5 shows the **Waveform** window for the DMA transactions between LPDDR and LSRAM. It shows the commands from SERDES AHB master, command decoder signals, and data transfer on CoreLSRAM AHB and MDDR interfaces.
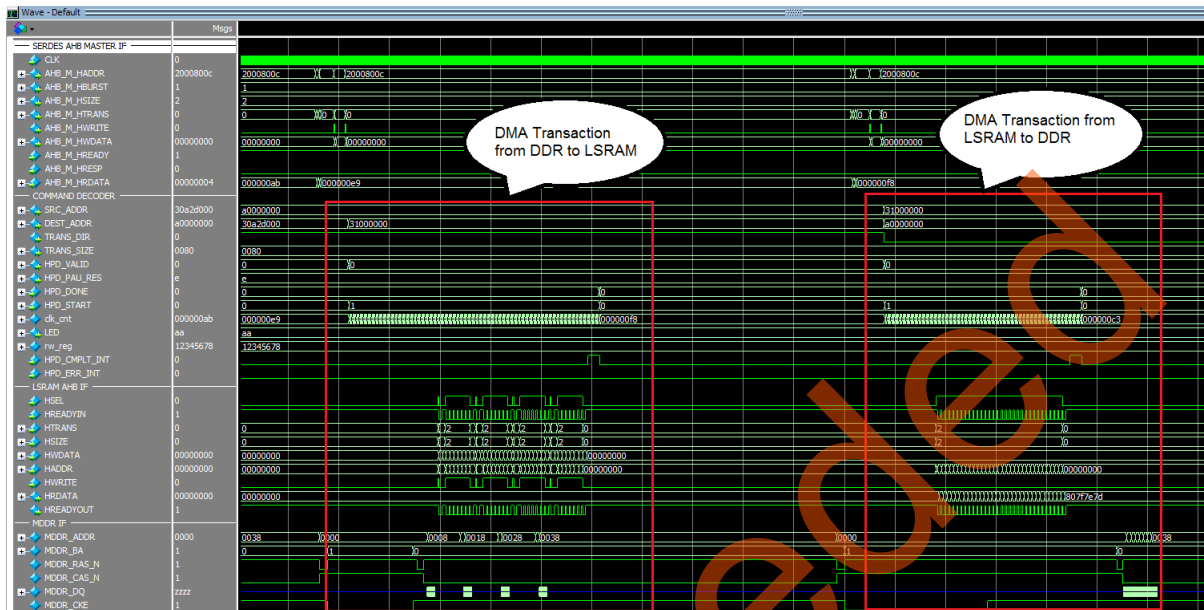

Figure 5 • **DMA Transactions between LPDDR and LSRAM**

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 Table 2.
   CAUTION: Ensure that the power supply switch, SW7, is switched OFF while connecting the jumpers on the IGLOO2 FPGA Evaluation Kit.

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.

Programming the IGLOO2 Board

The following steps describe how to program the demo design:

1. Download the demo design from:
   http://soc.microsemi.com/download/rsc/?f=m2gl_dg0585_hpms_hpdma_liberov11p6_df
2. Switch **ON** the power supply switch, **SW7**.
3. Launch the **FlashPro** software.
4. Click **New Project**.
5. Enter the project name as **PCIe_Data_Plane** in the **New Project** window.
6. Click **Browse** and navigate to the location to save the project.
7. Select a **Single Device** as the Programming mode.
8. Click **OK** to save the project.
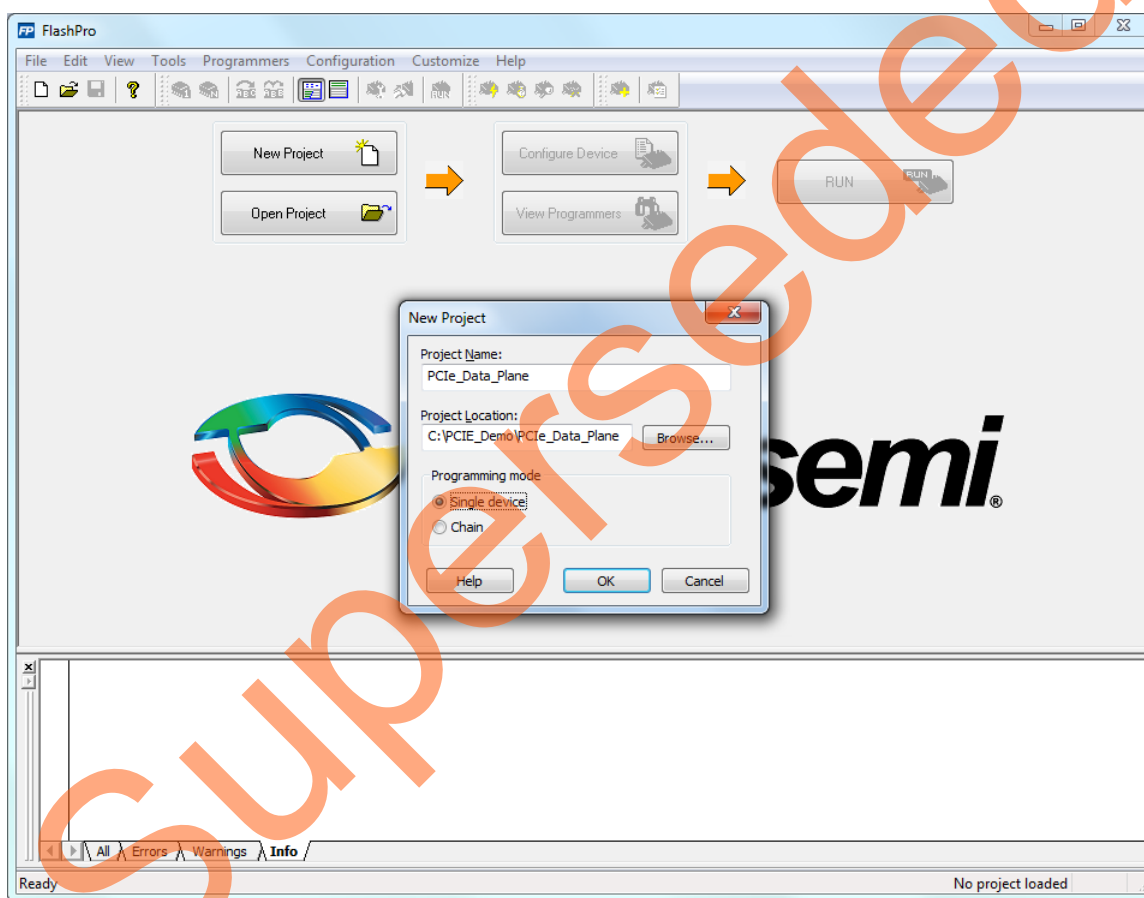
Figure 6 • **FlashPro New Project**

9. Click **Configure Device**.
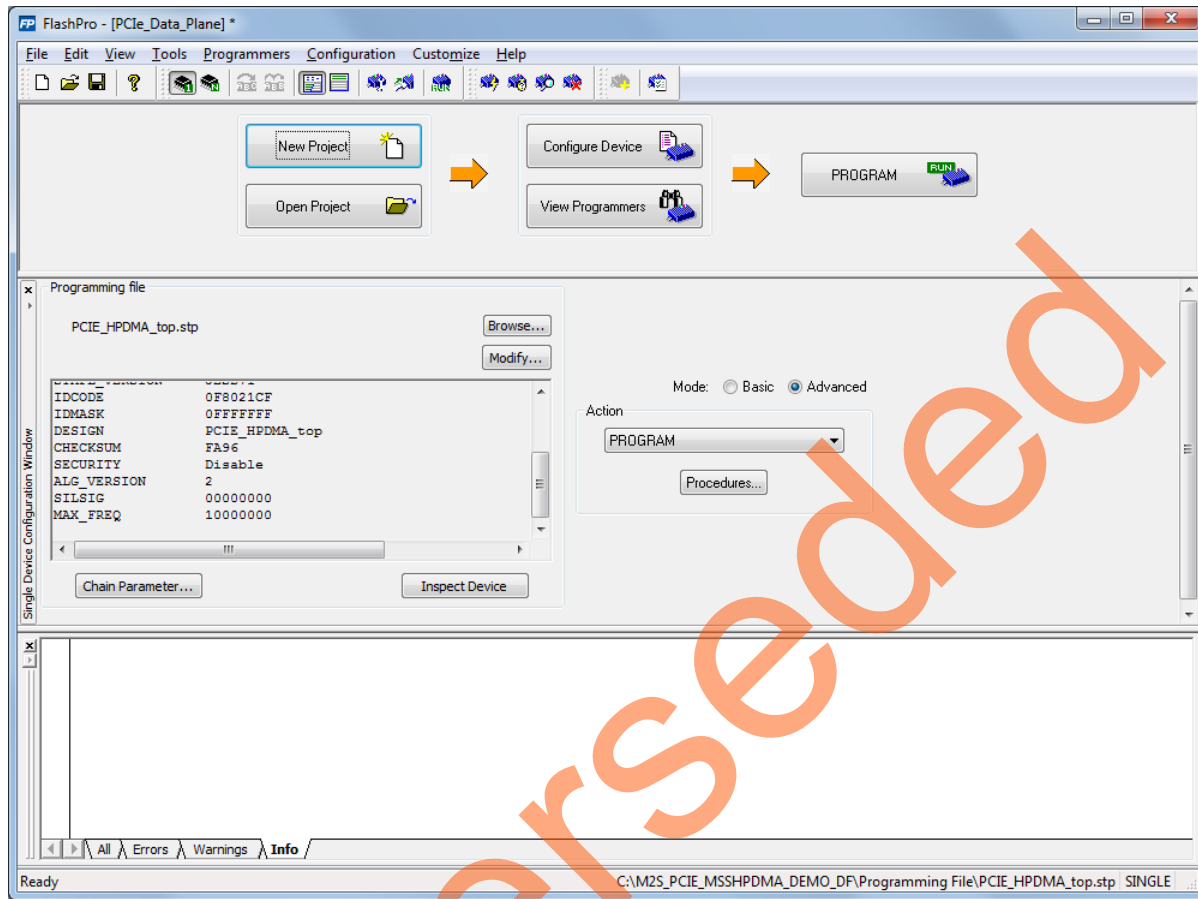
Figure 7 • **FlashPro Project Configuration**

10. Click **Browse** and navigate to the location where the `PCIe_HPDMA_top.stp` file is located and select the file. The default location is:
    <download_folder>\m2gl_dg0585_hpms_hpdma_liberov11p6_df\Programming File
11. Click **Advanced** as **Mode** and select **PROGRAM** as **Action**.

12. Click **PROGRAM** to start programming the device. Wait until the programmer status is changed to **RUN PASSED**.
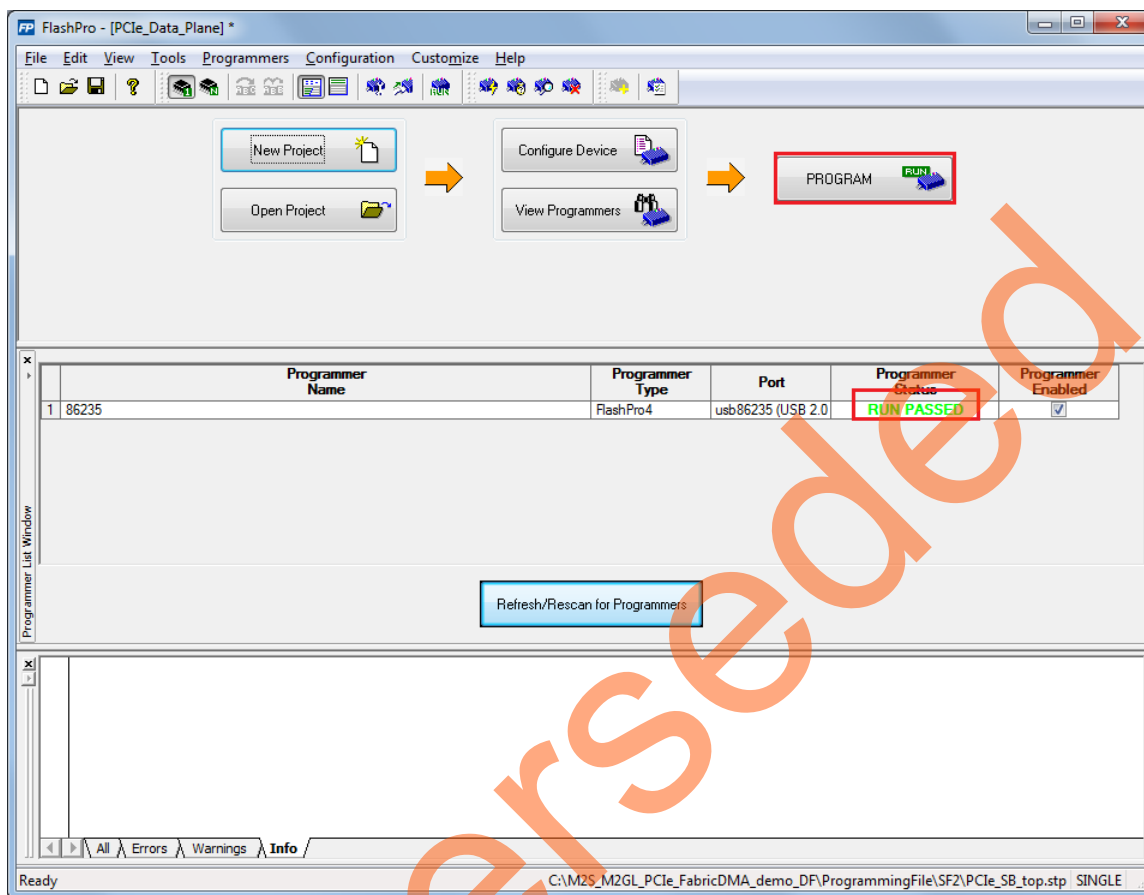
Figure 8 • **FlashPro Programming Passed**

Connecting IGLOO2 Evaluation Kit Board to 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. Connect the **CON1-PCIe Edge Connector** either to a host PC or laptop:

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

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

Note: Host PC or laptop is 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. Power OFF the host PC or laptop during PCIe card insertion.

Figure 9 shows the board setup for the host PC in which the 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, refer to the "Appendix 1: IGLOO2 Evaluation Kit Board Setup for Laptop" on page 36".
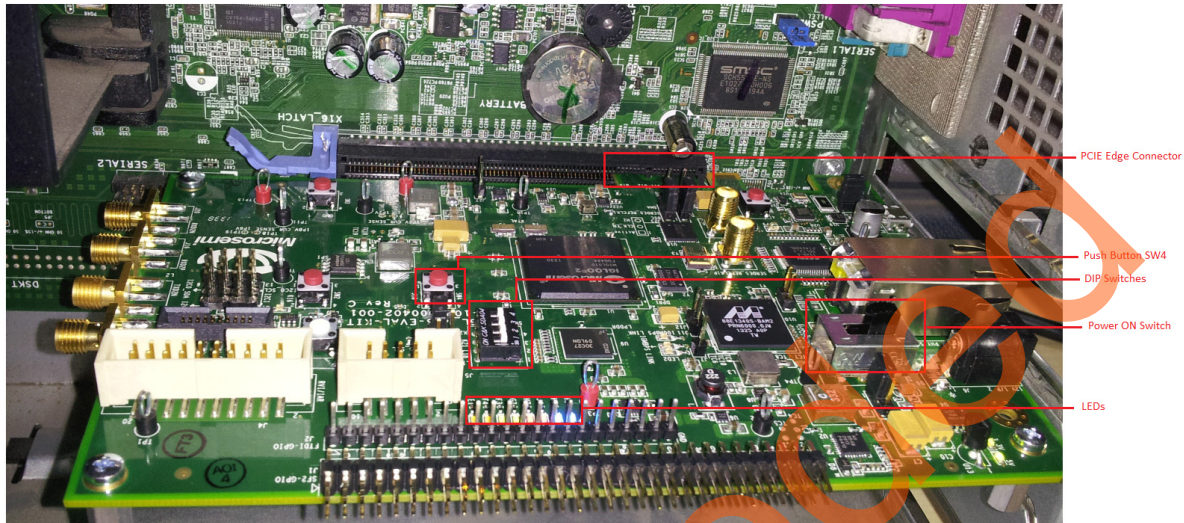
Figure 9 • **IGLOO2 Evaluation Kit Setup**

3. Switch **ON** the power supply switch, **SW7**.

4. Power on the host PC and check the **Device Manager** of the host PC for **PCIe Device**. Figure 10 on page 16 shows the example **Device Manager** window. If the device is not detected, power cycle the IGLOO2 Evaluation Kit and click **scan for hardware changes** option in the **Device Manager** window.
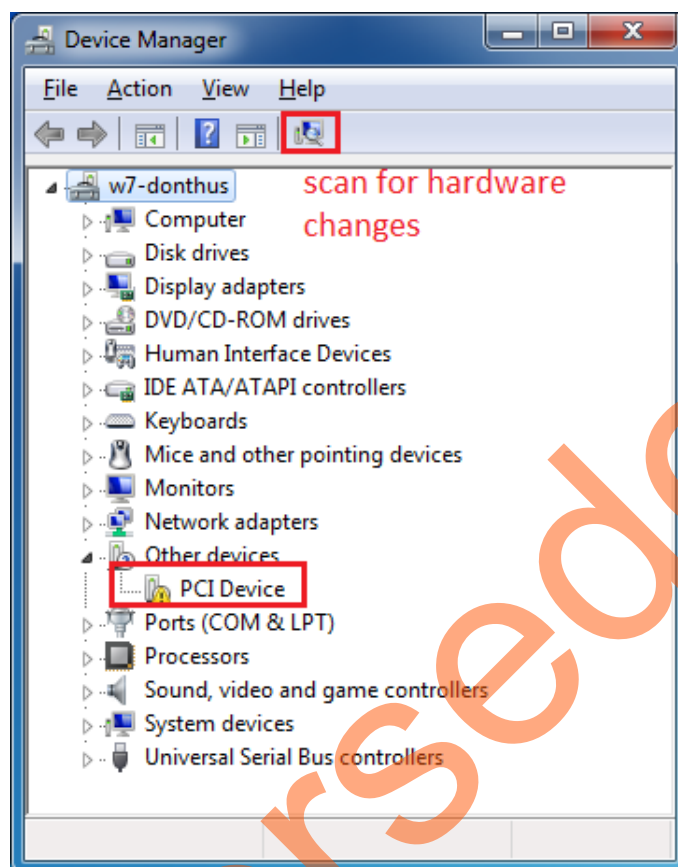
Figure 10 • **Device Manager - PCIe Device Detection**

Note: If the device is still not detected, check if the BIOS version in the host PC is the latest version, and PCI is enabled in the host PC BIOS.

5. If the host PC has any other installed drivers (previous versions of Jungo drivers) for the PCIe device, uninstall them.

6. To uninstall the previous versions of Jungo drivers, follow steps **a** and **b**.

a. Navigate to **Device Manager** and right-click **DEVICE** and click **Uninstall**, as shown in Figure 11. The **Confirm Device Uninstall** dialog box is displayed.
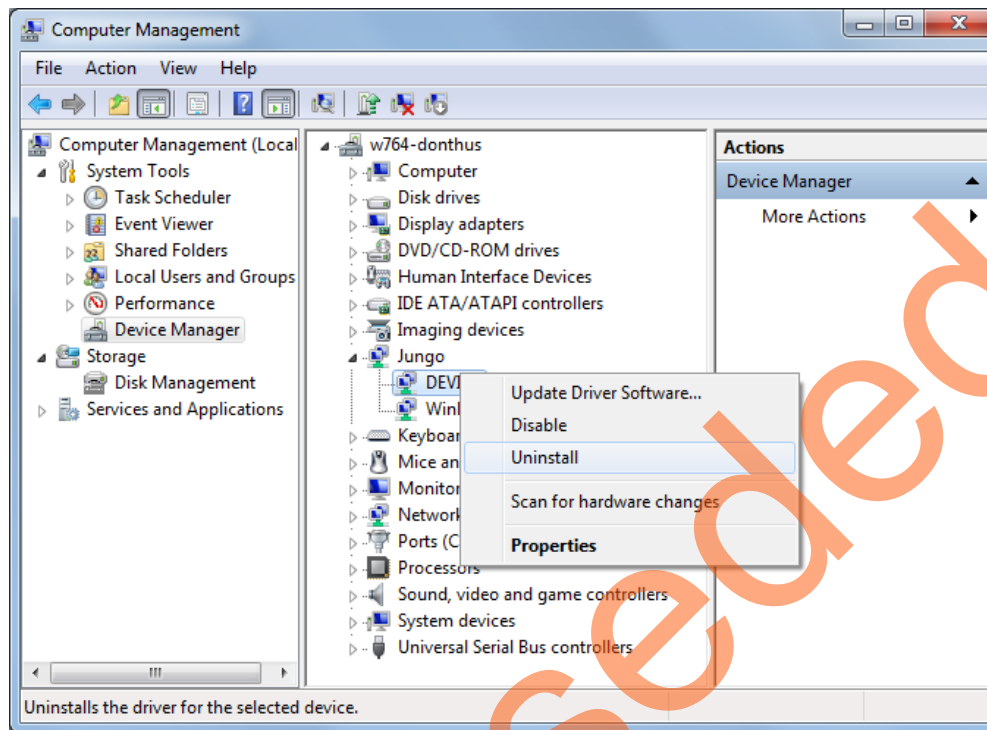
Figure 11 • **Uninstall Jungo Driver**

b. From the **Confirm Device Uninstall** dialog box, select **Delete the driver software for this device** check box.

c. Click **OK**. After uninstalling the previous version of Jungo drivers, ensure that the PCI Device is detected in the **Device Manager** window as shown in Figure 12.
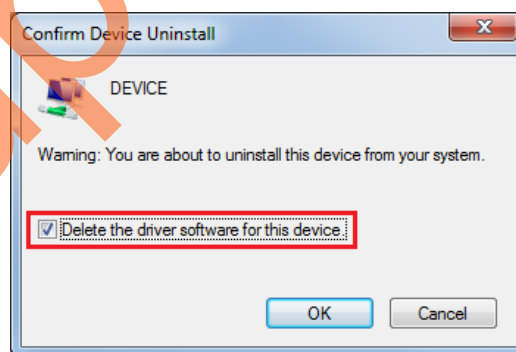
Figure 12 • **Confirm Device Uninstall Dialog Box**

Drivers Installation

The PCIe Demo uses a driver framework provided by Jungo WinDriverPro. The following steps describe how to install the PCIe drivers on the host PC:

1. Extract the `PCIe_Demo.rar` to *C:\ drive*. The PCIe_Demo.rar is at *m2gl_dg0585_hpms_hpdma_liberov11p6_df\Drivers_64bitOS\PCIe_Demo.rar*

2. Run the batch file `Jungo_KP_install.bat` located at *C:\ PCIe_Demo\DriverInstall*

Note: Installing these drivers require administration rights.

3. If the Windows Security dialog appears asking to install or not, click **Install**.
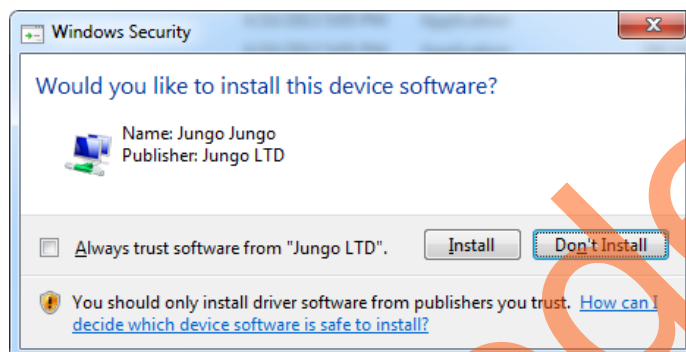
Figure 13 • **Jungo Driver Installation**

Note: If the installation fails, invoke the Command Prompt in Administrator mode and run the batch file `Jungo_KP_install.bat` located at *C:\PCIe_Demo\DriverInstall*

4. If the **Windows Security** dialog box appears asking whether to install or not, click **Install this driver software anyway**.
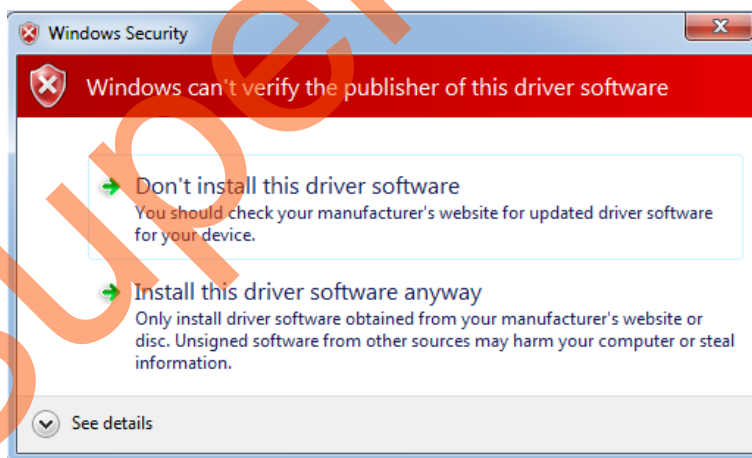
Figure 14 • **Windows Security**

Installing PCIe_Demo Application GUI

The PCIe_Demo application is a simple graphic user interface that runs on the host PC to communicate with the IGLOO2 PCIe end point device. It provides PCIe link status, driver information, and demo controls. The PCIe_Demo application invokes the PCIe driver installed on the host PC and provides commands to the driver according to the selection made.

The following steps describe how to install the PCIe_Demo application GUI:

1. Download the PCIe demo application at:
   http://soc.microsemi.com/download/rsc/?f=PCIe_Demo_GUI_Installer
2. Double-click **setup.exe**. Do not change the default options and click **Next**.
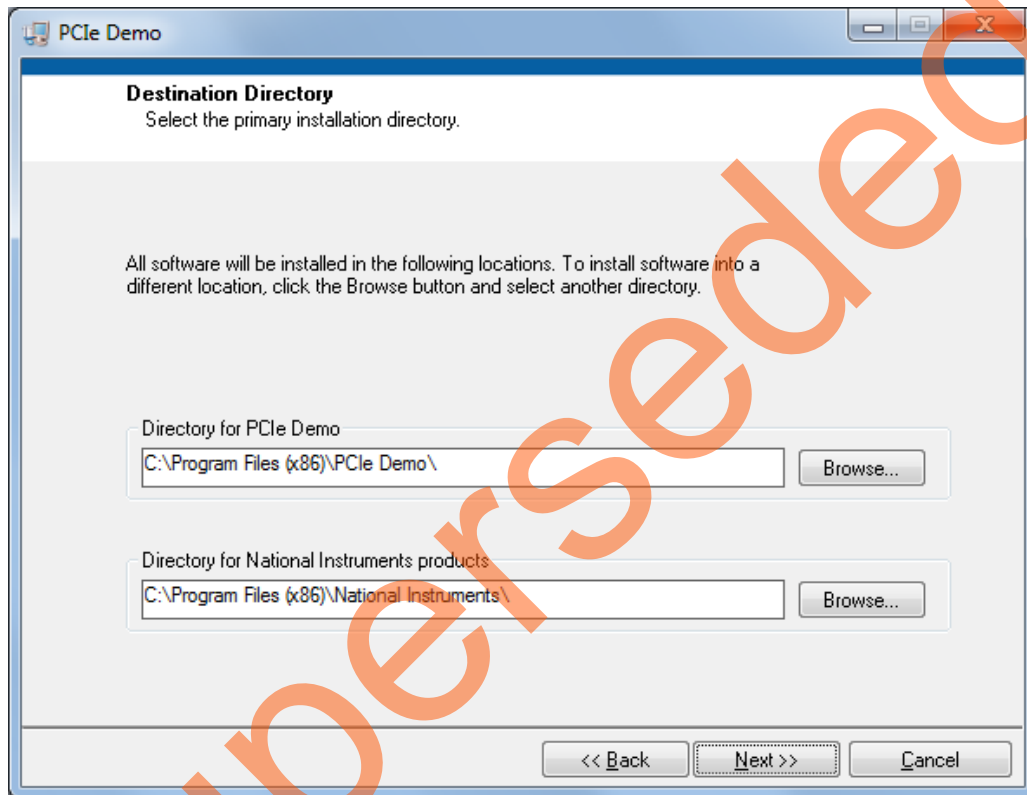
Figure 15 • **GUI Installation**

3. Click **Next** to complete the installation. The **Installation Complete** window is displayed as shown in Figure 16.
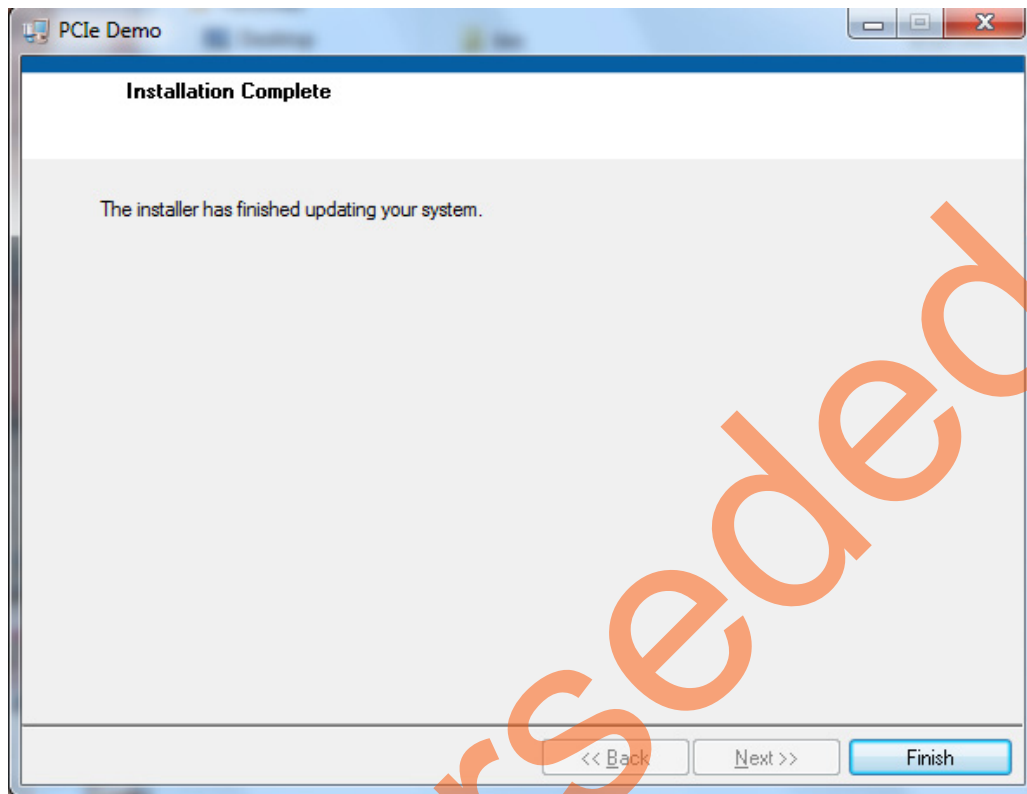
Figure 16 • **Successful Installation of GUI**

4. Shut down the host PC.
5. Power cycle the IGLOO2 Evaluation Kit board.
6. Restart the host PC.

Running the Design

1. Check the host PC **Device Manager** for the drivers. If the device is not detected, power cycle the IGLOO2 Evaluation Kit board and click **scan for hardware changes** in Device Manager.

2. Ensure that the board is switched **ON**.

Figure 17 • **Device Manager - PCIe Device Detection**

Note: If a warning symbol appears on the **DEVICE** or **WinDriver** icons in **Device Manager**, uninstall them and start from step 1 of Driver Installation.

4. Click the **Connect** icon at the top right corner of the PCIe_Demo application. The application detects and displays the connected Kit, demo design, and PCIe link. Figure 19 shows the example messages after the connection is established.
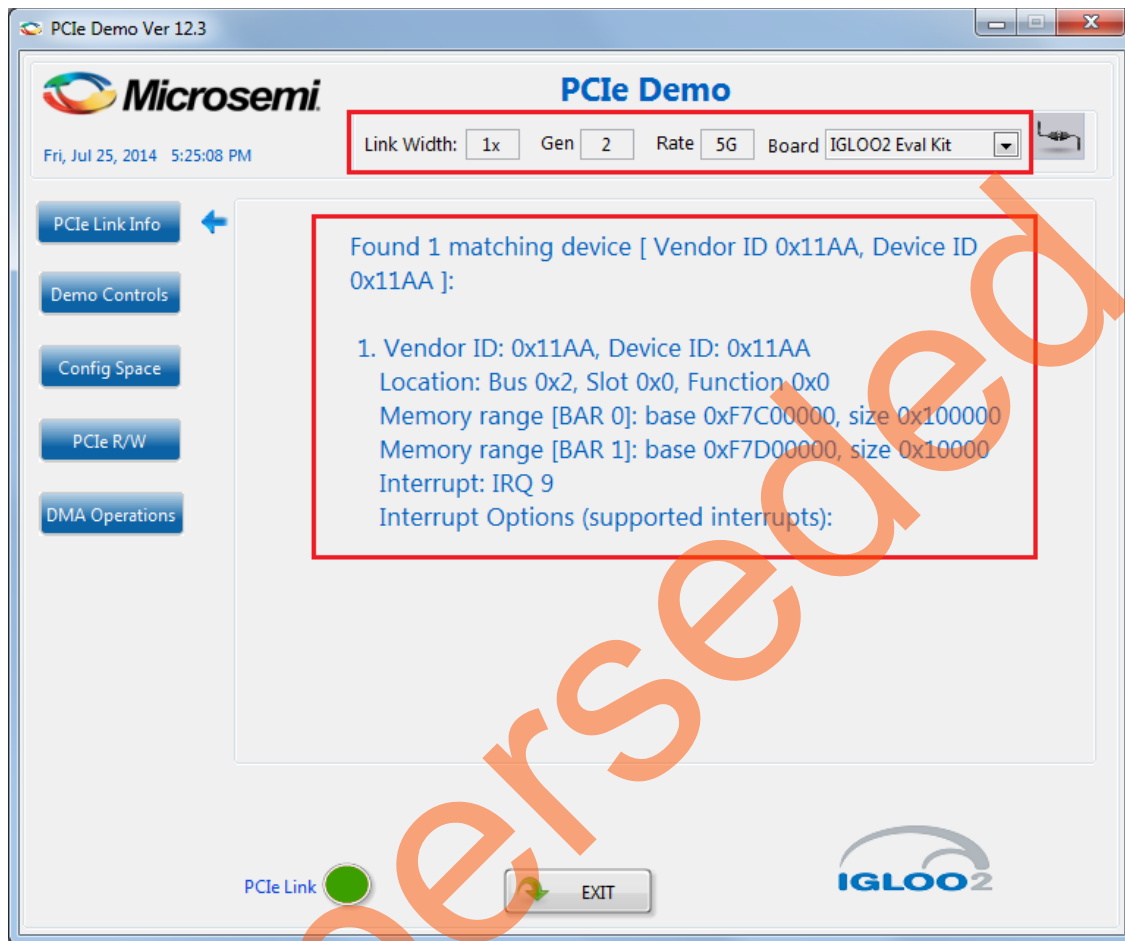
Figure 19 • **PCIe Device Information**

5. Click **Demo Controls** to display the LEDs options and DIP switch positions, as shown in Figure 20.
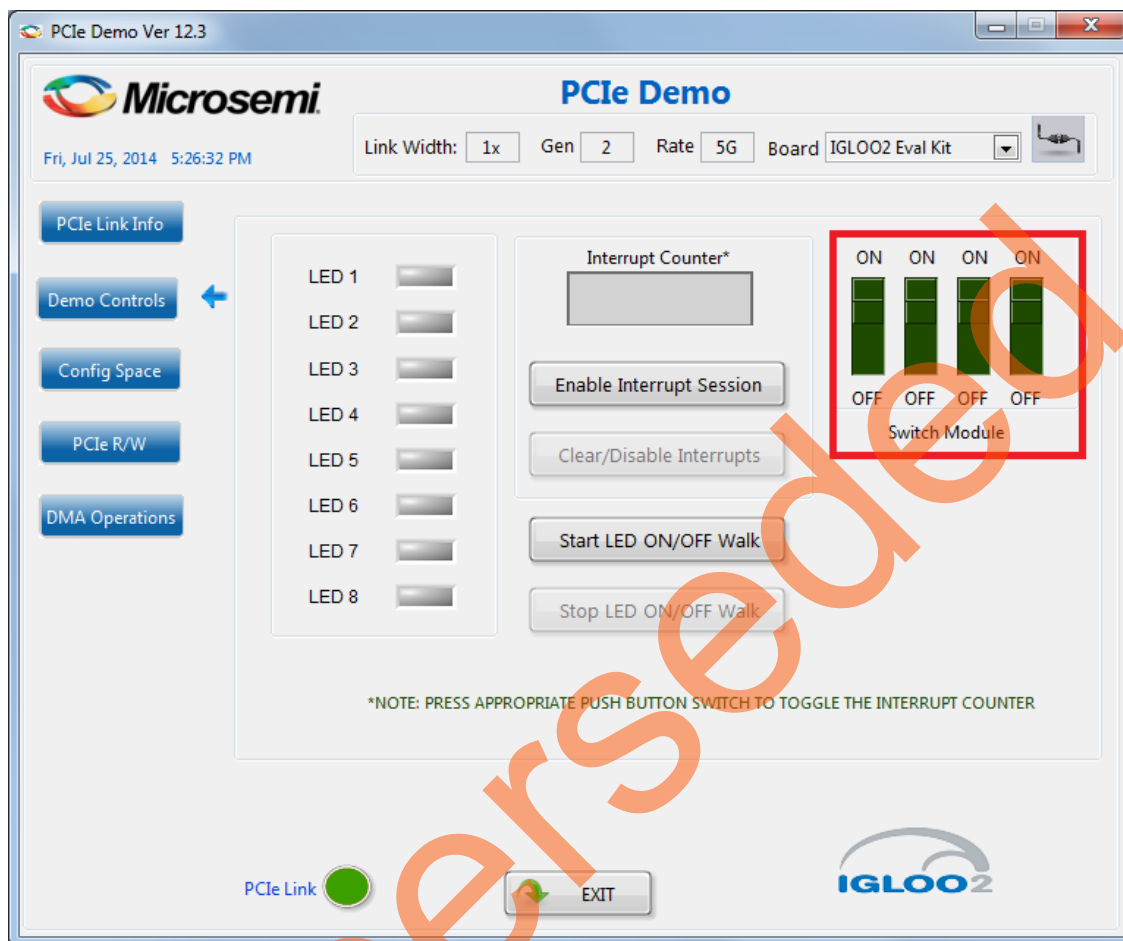
Figure 20 • **Demo Controls**

6. Click LED buttons to switch ON or OFF the LEDs on the board.

7. Click **Start LED ON/OFF Walk** to blink the LEDs on the board.

8. Click **Stop LED ON/OFF Walk** to stop the LEDs blinking.

9. Change the DIP switch positions on the board and observe the same being reflected in the **Switch Module** of the PCIe_Demo application.

10. Click **Enable Interrupt Session** to enable the PCIe interrupt.

11. Press the **Push** button, SW4 on the IGLOO2 Evaluation Kit board. Observe the interrupt count on the **Interrupt Counter** field in PCIe_Demo application as shown in Figure 21.
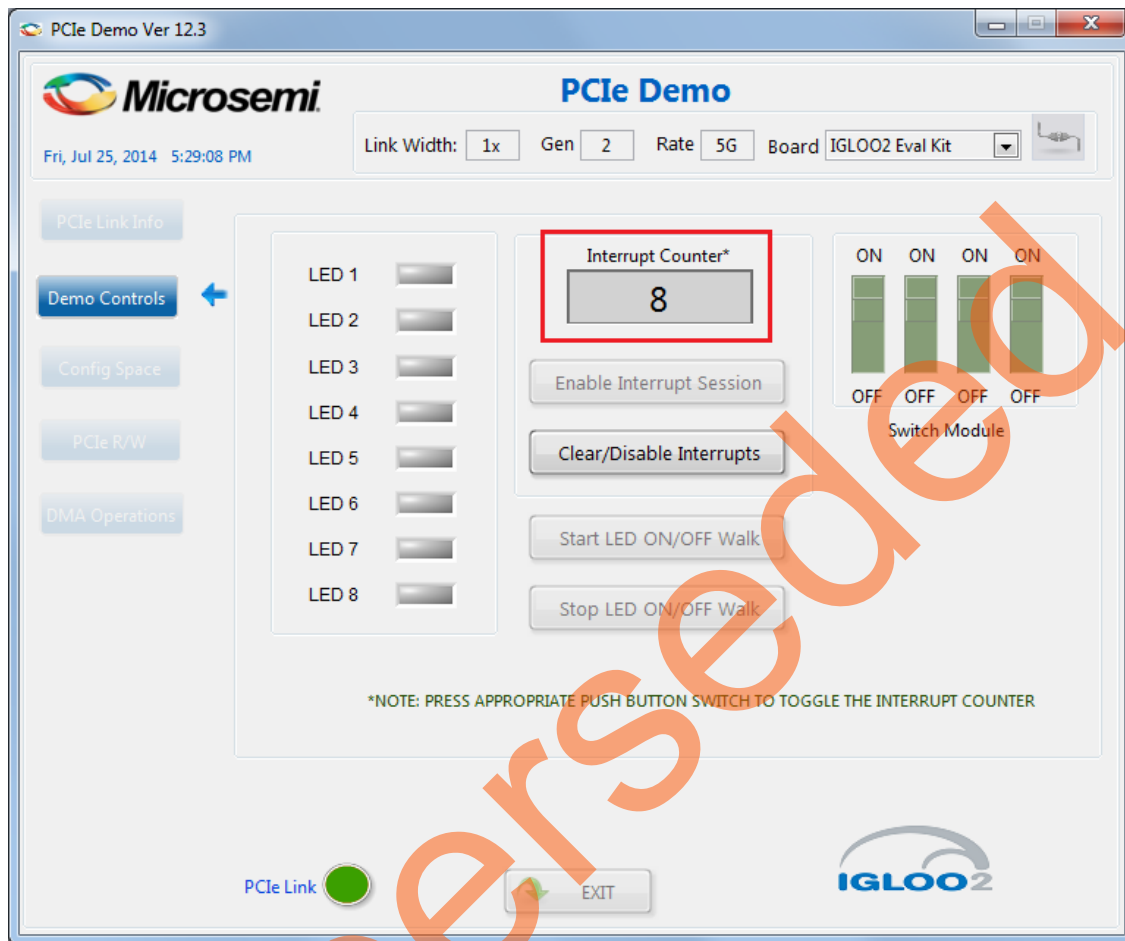
Figure 21 • **PCIe Interrupt**

12. Click **Clear/Disable Interrupts** to clear and disable the PCIe interrupts.

13. Click **Config Space** to view the details about the PCIe configuration space. Figure 22 shows the PCIe configuration space.
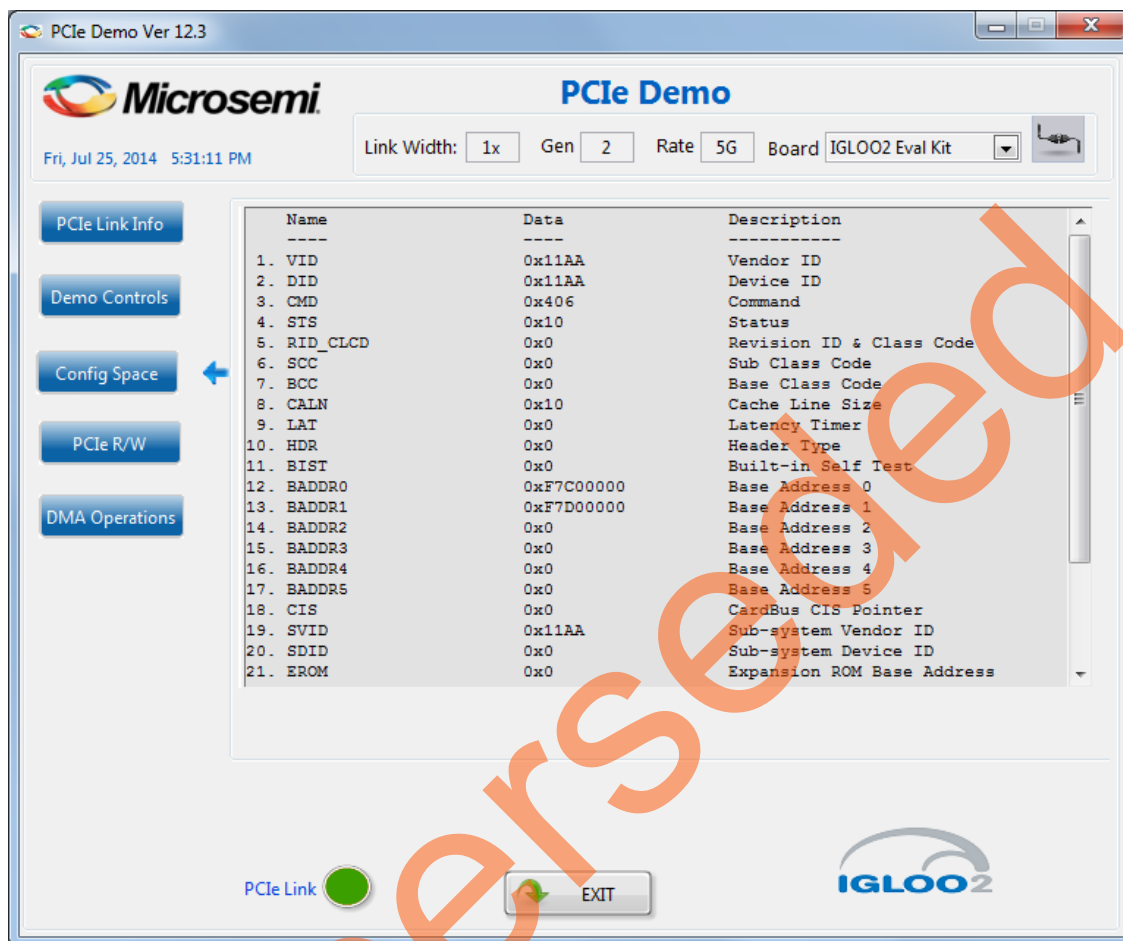

Figure 22 • **PCIe Configuration Space**

14. Click **PCIe R/W** to execute reads and writes to a 32-bit scratchpad register using BAR1 space. Figure 23 on page 27 shows the PCIe R/W panel.
15. Click **DMA Operations** to perform DMA operations. Four types of DMA transactions are possible:
    – PC Memory to DDR
    – DDR to PC Memory
    – DDR to Fabric LSRAM
    – Fabric LSRAM to DDR
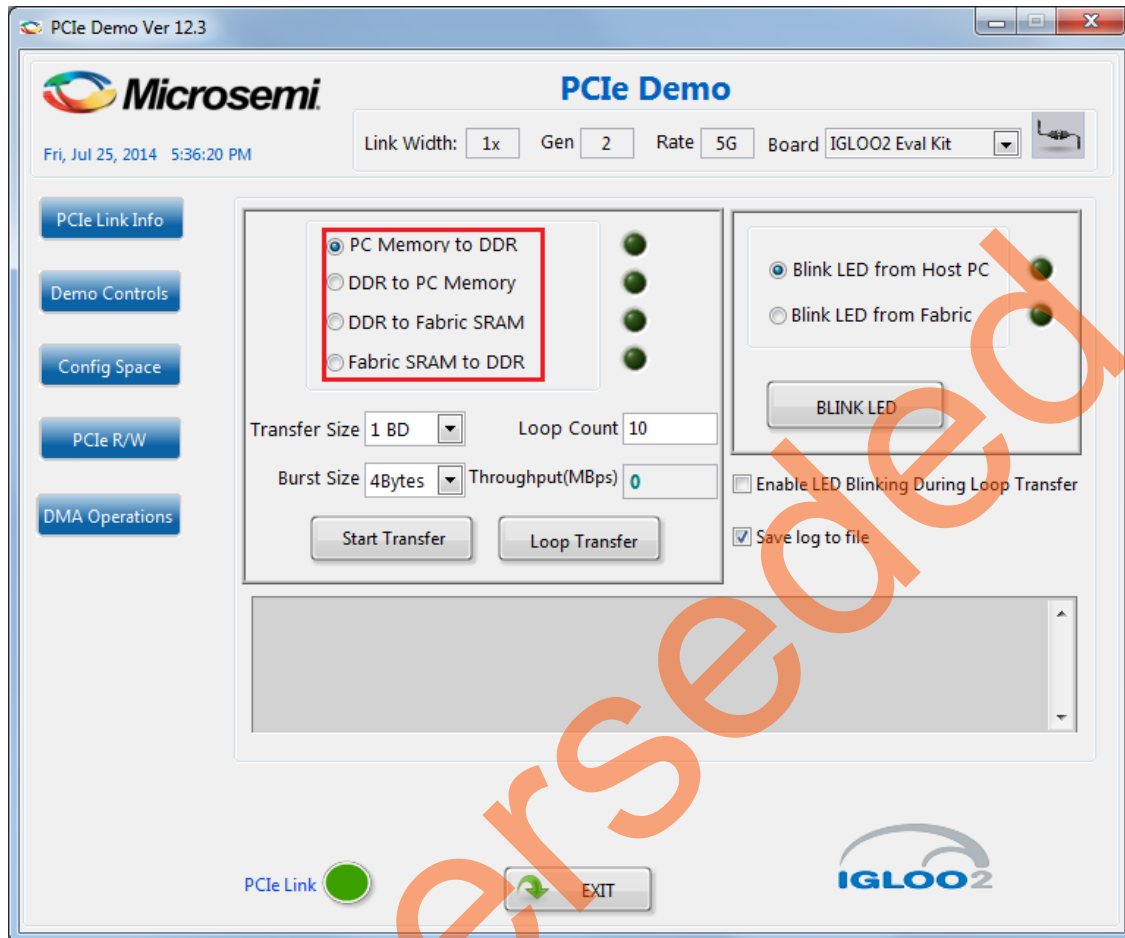
Figure 23 • **DMA Operations**

16. The transfer size can be selected from 64 KB to 1 MB for:

– PC memory to DDR

– DDR to PC memory

The transfer size is provided in steps of HPDMA buffer descriptor size (64 KB). For more information on buffer description, refer to the "HPDMA" chapter in the *UG0448: IGLOO2 FPGA High Performance Memory Subsystem User Guide*.

17. The transfer size can be selected from 16 KB to 32 KB for:
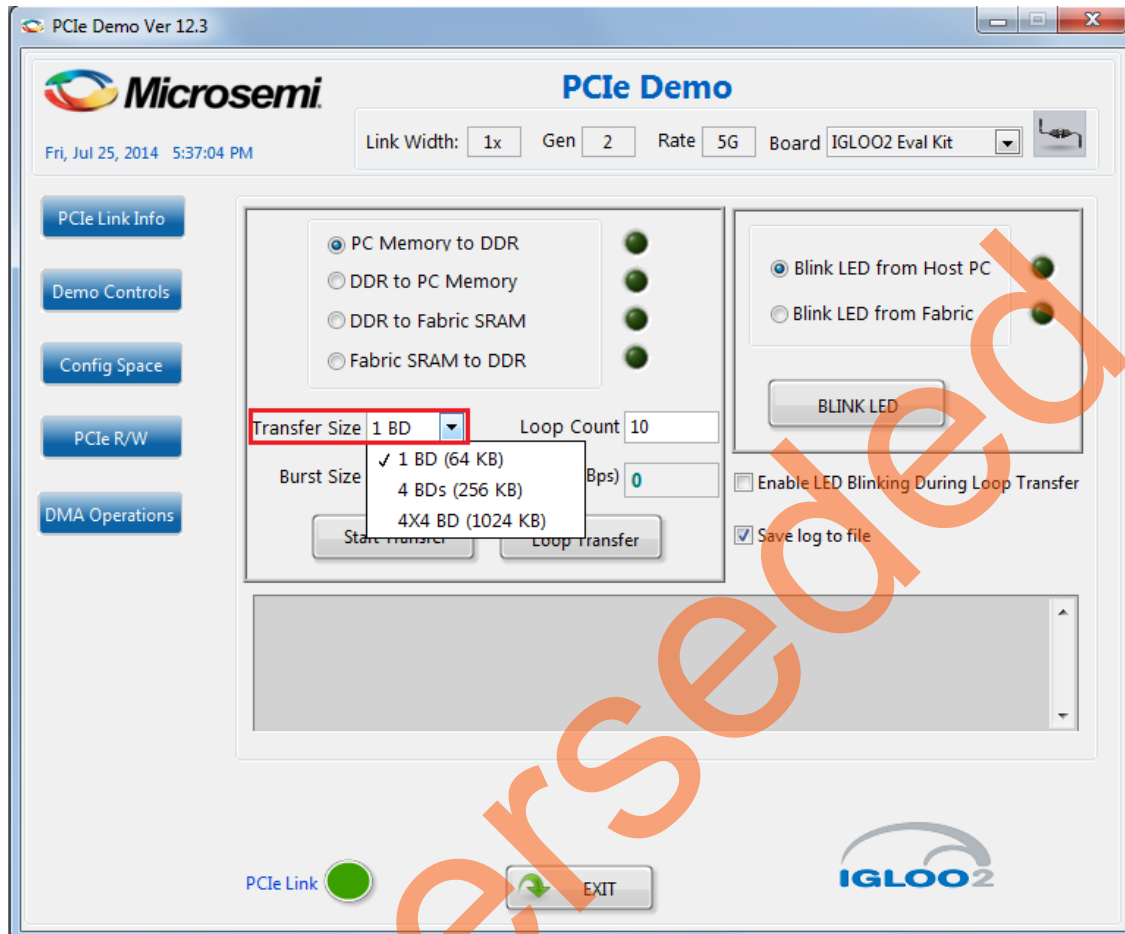
– Fabric LSRAM to DDR

– DDR to Fabric LSRAM

Figure 24 • **Transfer Size Selection**

18. The Burst Size (TLP size) is fixed to 4 bytes as the HPDMA does not support burst, and the actual size of the PCIe packet is the width of a single AHB transfer, which is 4 bytes (32-bit AHB) as shown in Figure 25.
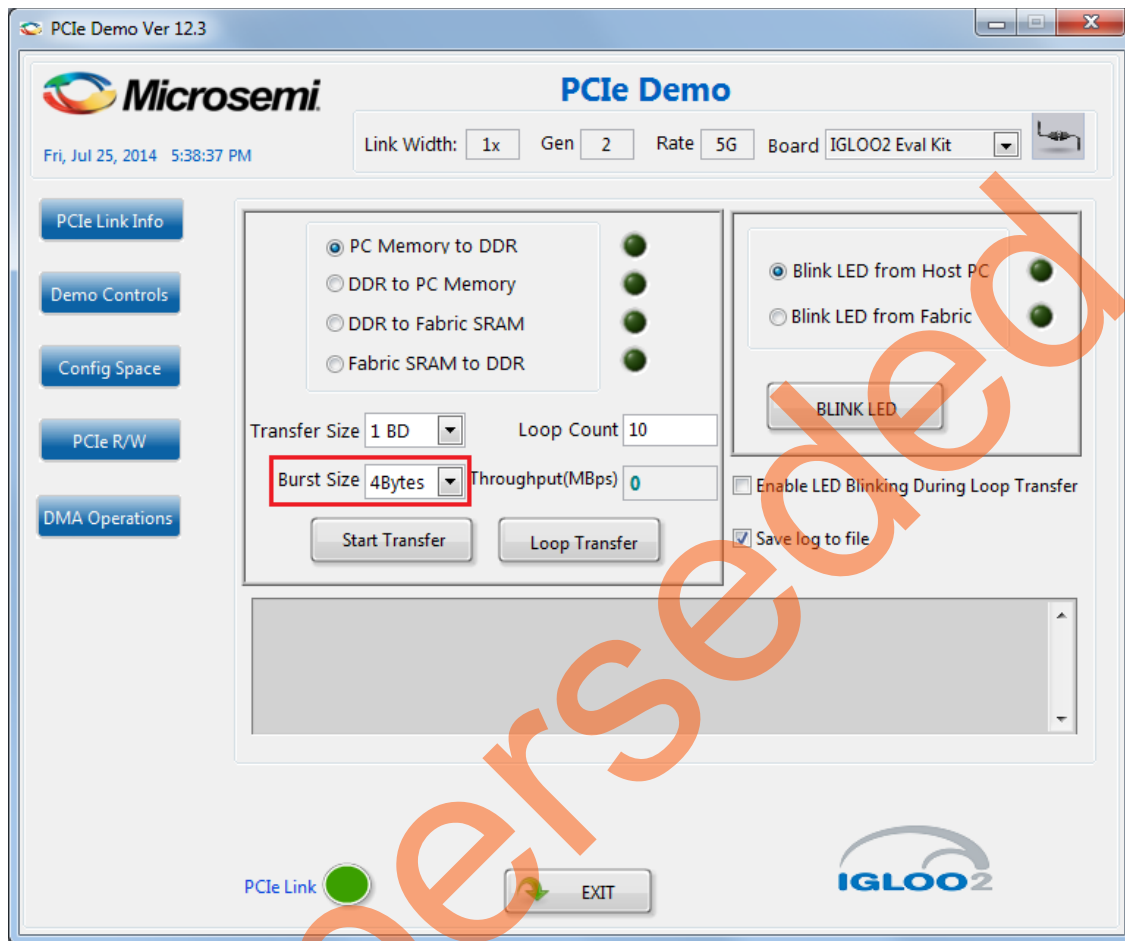

Figure 25 • **Burst Size Selection**

19. Select the type of DMA transfer as **PC Memory to DDR**. Select any **Transfer Size**, and click **Start Transfer** after completion of the data transfer, the GUI displays the throughput in MBps as shown in Figure 26.
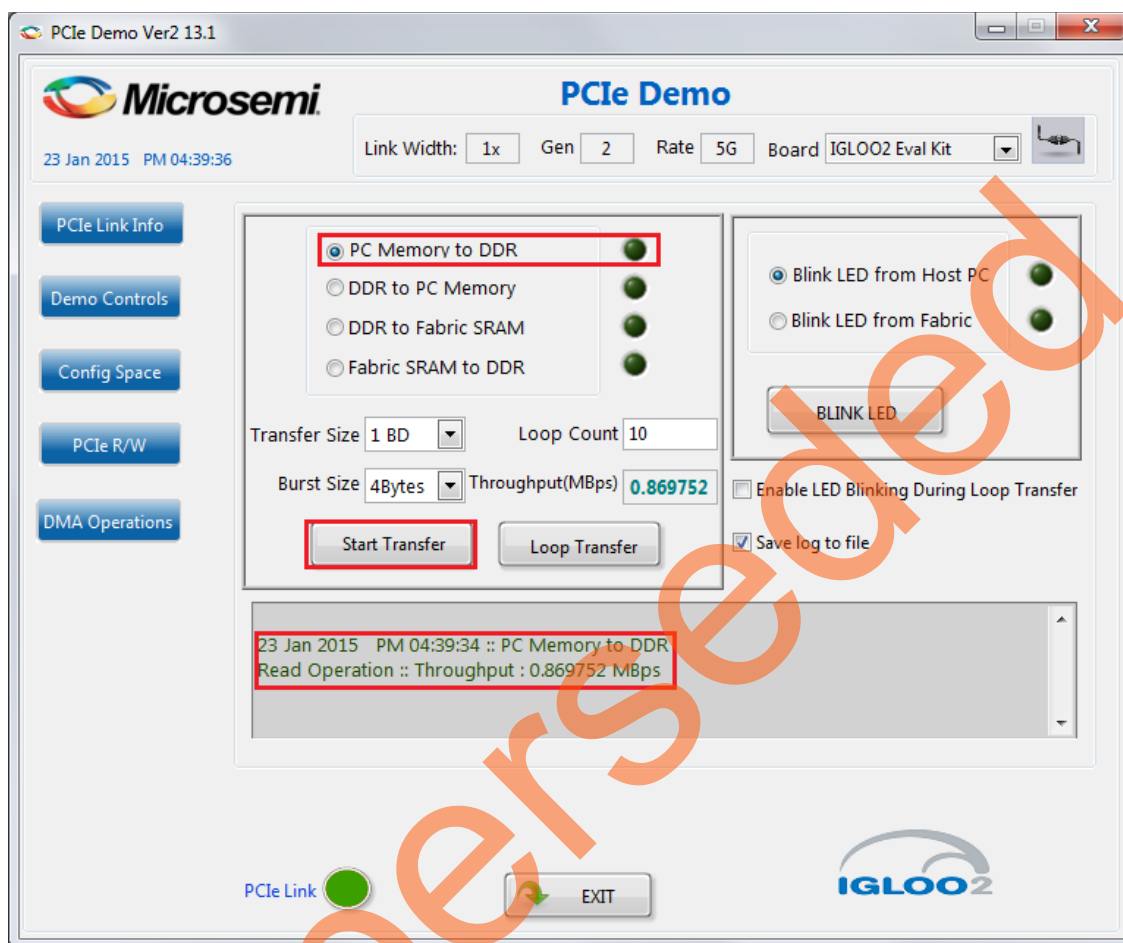
Figure 26 • **DMA Transaction - PC Memory to DDR**

20. Enter the **Loop Count** as 10 in the **Loop Count** box and click **Loop Transfer** to perform 10 sequential DMA transactions. The DMA throughput is displayed in MBps after the completion of data transfer.

Figure 27 shows the throughput of DMA transactions from the host PC to DDR. The average throughput is logged and the log file is stored at *C:\PCIe_Demo\DriverInstall* in the host PC.
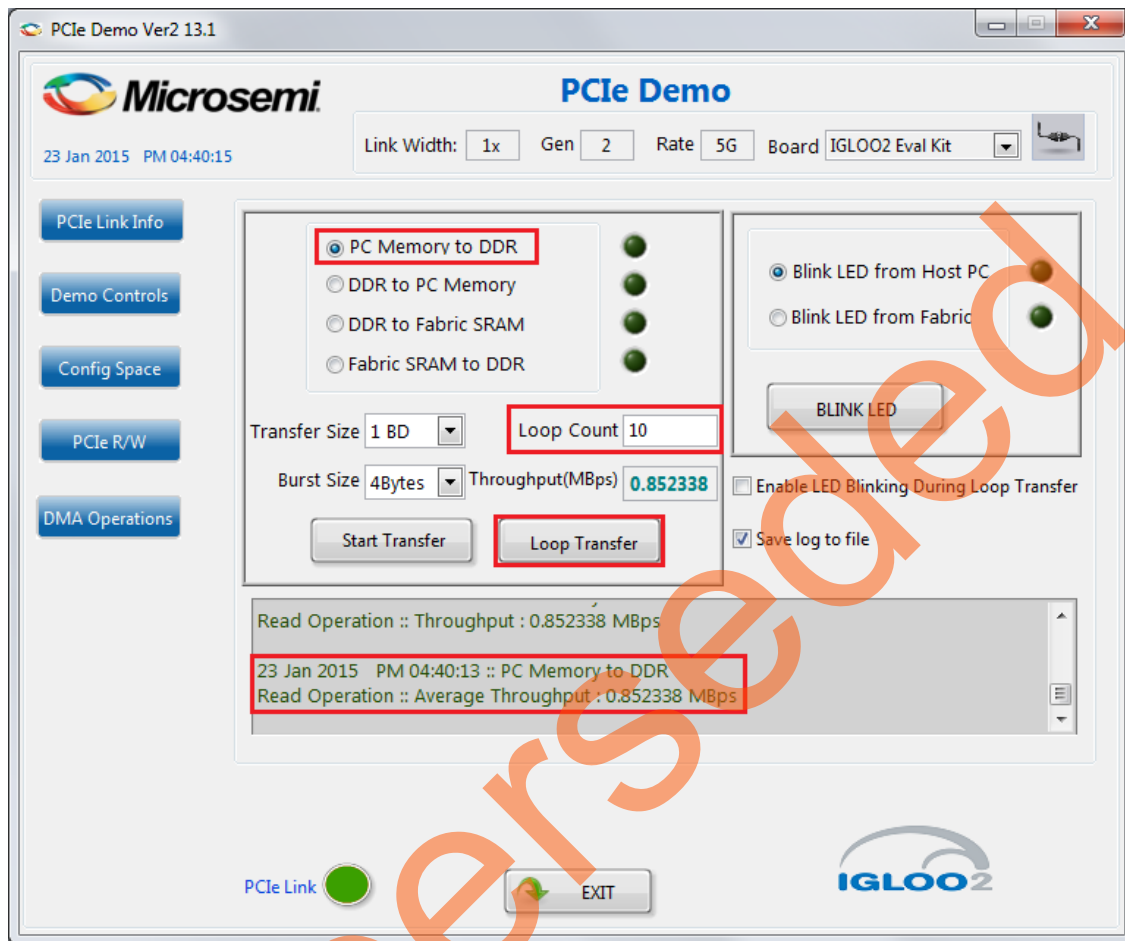

Figure 27 • **PC Memory to DDR DMA Loop Transfer**

21. Select the type of DMA transfer as **DDR to PC Memory**. Select any **Transfer Size** and click **Loop Transfer** as shown in Figure 28.
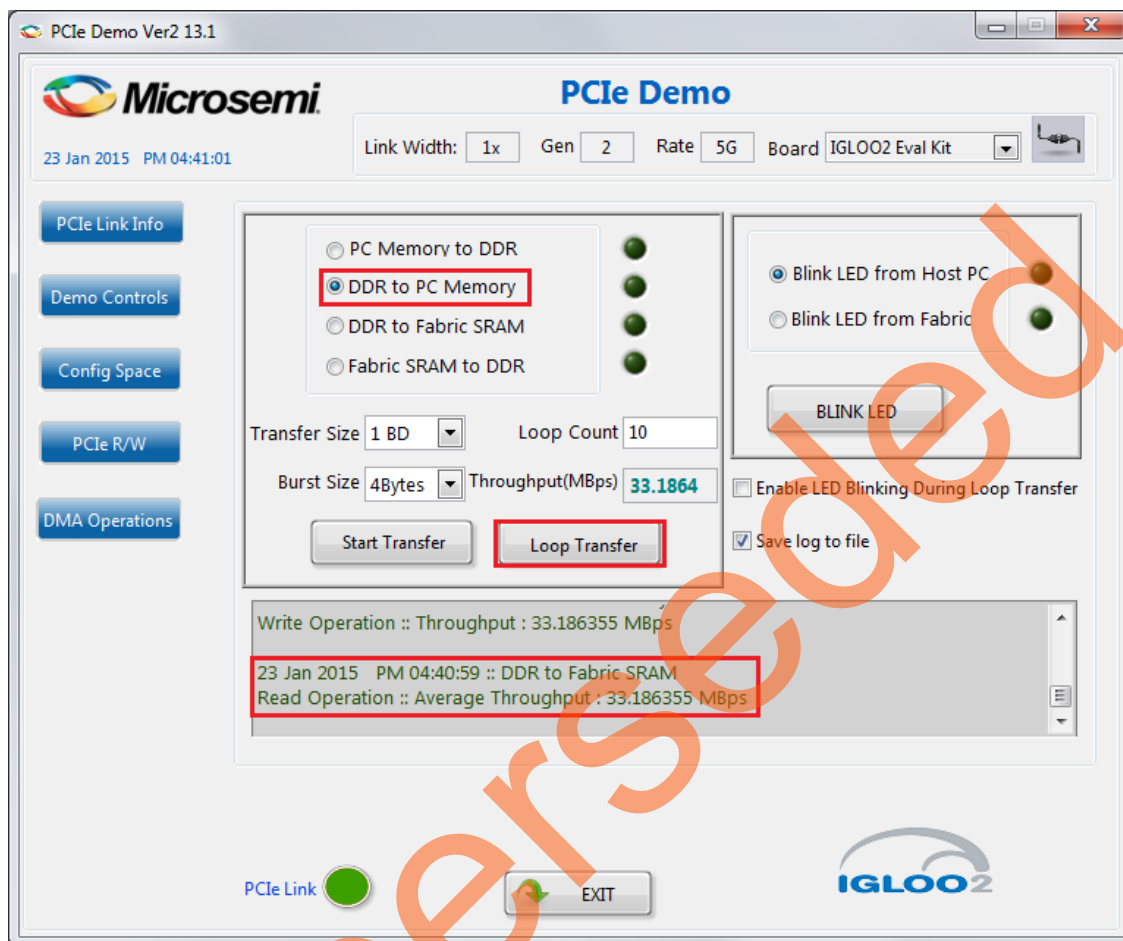

Figure 28 • **DDR to PC Memory DMA Loop Transfer**

22. Select the type of DMA transfer as **DDR to Fabric SRAM**. Select any **Transfer Size** and click **Loop Transfer** as shown in Figure 29.
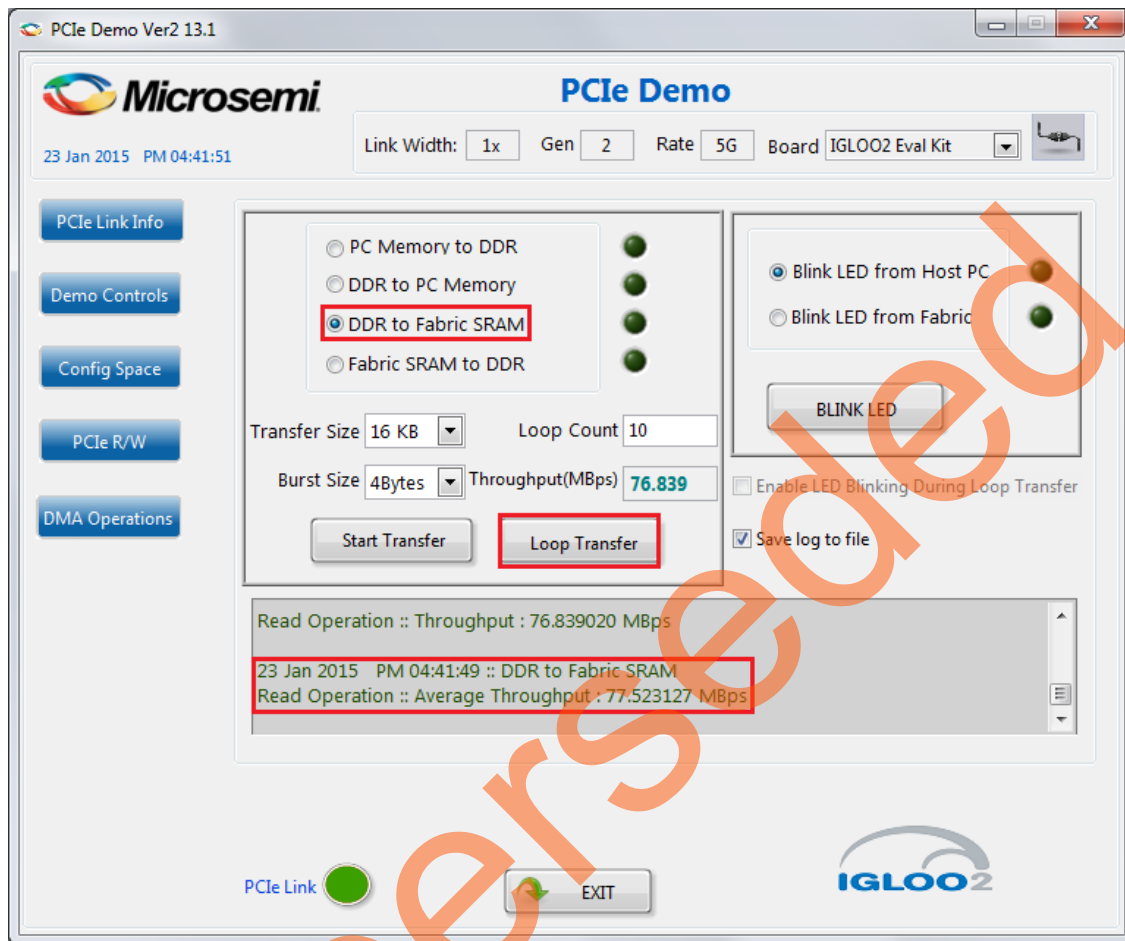

Figure 29 • **LSRAM To DDR DMA Loop Transfer**

23. Select the type of DMA transfer as **Fabric SRAM to DDR**. Select any **Transfer Size** and click **Loop Transfer** as shown in Figure 30.
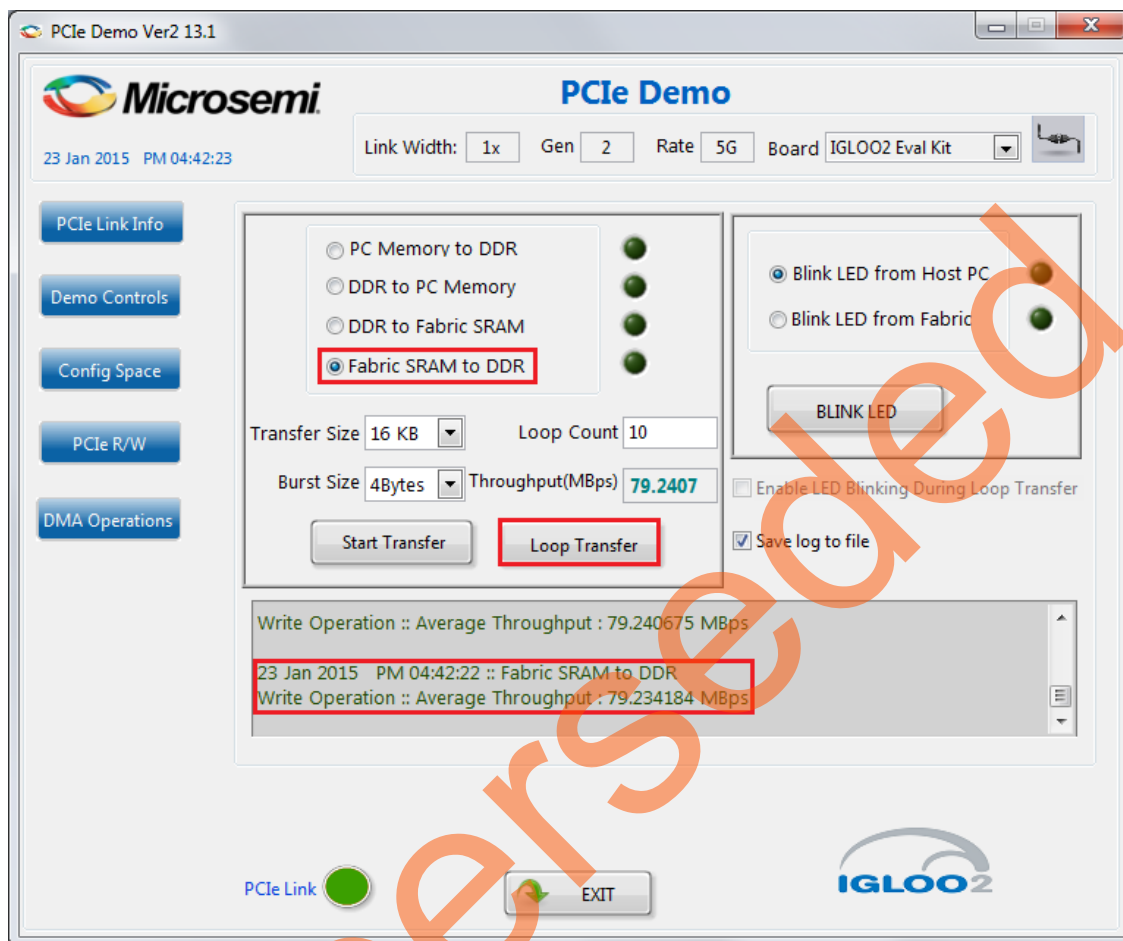

Figure 30 • **DDR to LSRAM DMA Loop Transfer**

24. The LEDs on the board can blink in parallel to the DMA operations by using the LED controls on the right side of the GUI. Select the **Enable the LED Blinking During Loop Transfer** check box to perform the LED blinking from host PC to DMA transfers.

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

Conclusion

This demo shows how to implement a PCIe data plane design using HPMS HPDMA. The data transfer occurs between the host PC IGLOO2 LSRAM, and IGLOO2 DDR memories. The throughput for data transfers depends on the host PC system configuration, and the types of PCIe slots used.

Table 3 lists the throughput values observed in the HP Workstation Z220 - CMT.

Table 3 • **HP Workstation Z220 - CMT - Throughput Values**

DMA Transfer Type	DMA Transfer Size	Throughput (Mega Bytes Per Second)			
		Gen 1		Gen 2	
		Single Transfer	Loop Transfer	Single Transfer	Loop Transfer
Host PC Memory to DDR	• 1 BD (64 KB) • 4 BD (256 KB) • 4 x 4 BD (1 MB)	2.1	2.2	2.604	2.826
DDR to Host PC Memory	• 1 BD (64 KB) • 4 BD (256 KB) • 4 x 4 BD (1 MB)	23.8	23.8	33	33
DDR to Fabric SRAM	• 16 KB • 32 KB	78	77.5	78	77.5
Fabric SRAM to DDR	• 16 KB • 32 KB	79	79	79	79

Appendix 1: IGLOO2 Evaluation Kit Board Setup for Laptop

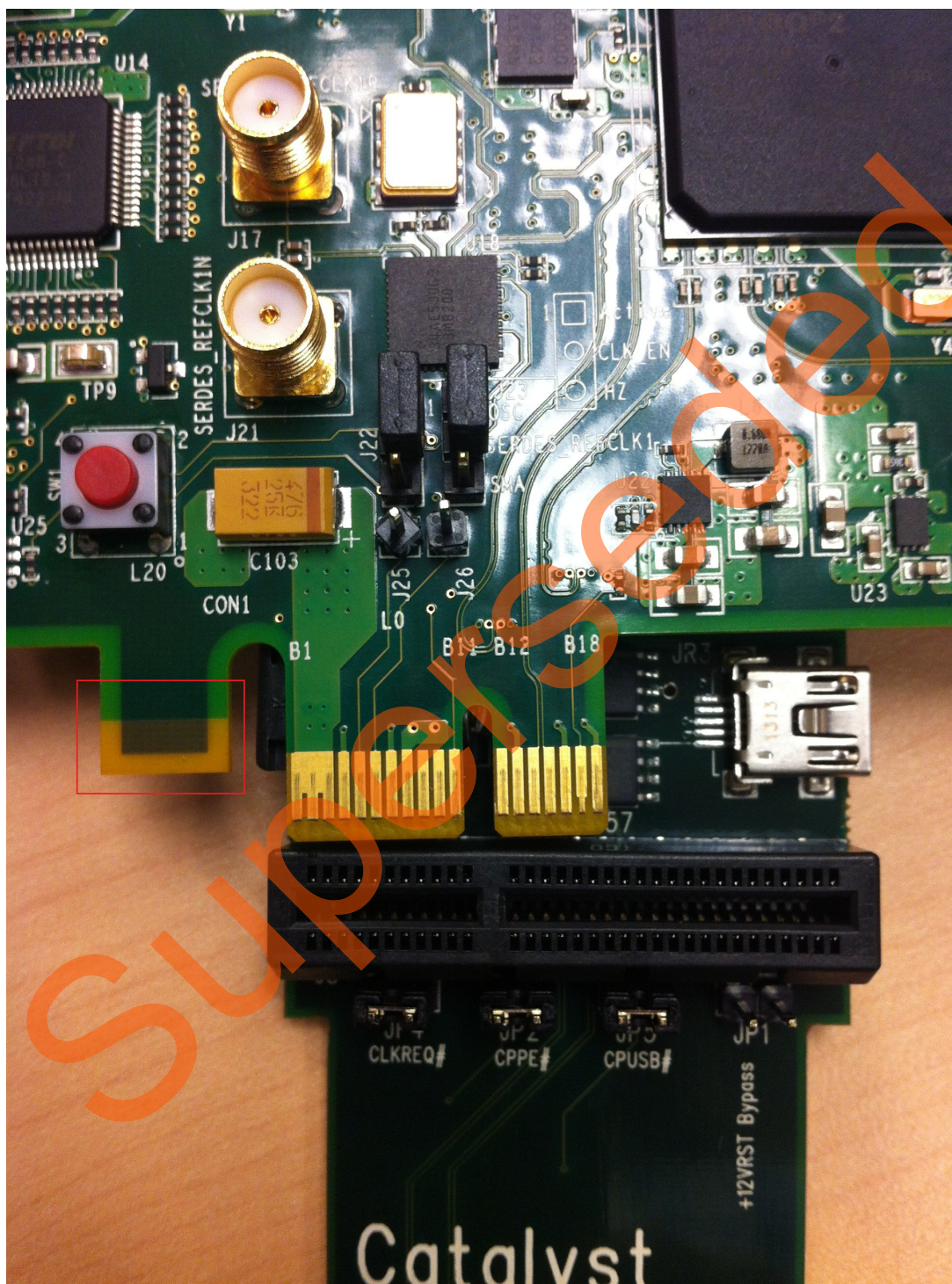Figure 31 shows how to line up the IGLOO2 Evaluation Kit PCIe connector with the adapter card slot.

Figure 31 •   **Lining Up the IGLOO2 Evaluation Kit Board**

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

Figure 32 shows the IGLOO2 Evaluation Kit PCIe connector inserted into the PCIe adapter card slot.

Figure 32 • **Inserting the IGLOO2 Evaluation Kit PCIe Connector into the PCIe Adapter Card Slot**

Figure 33 shows the PCIe adapter card and the IGLOO2 Evaluation Kit connected to the laptop.
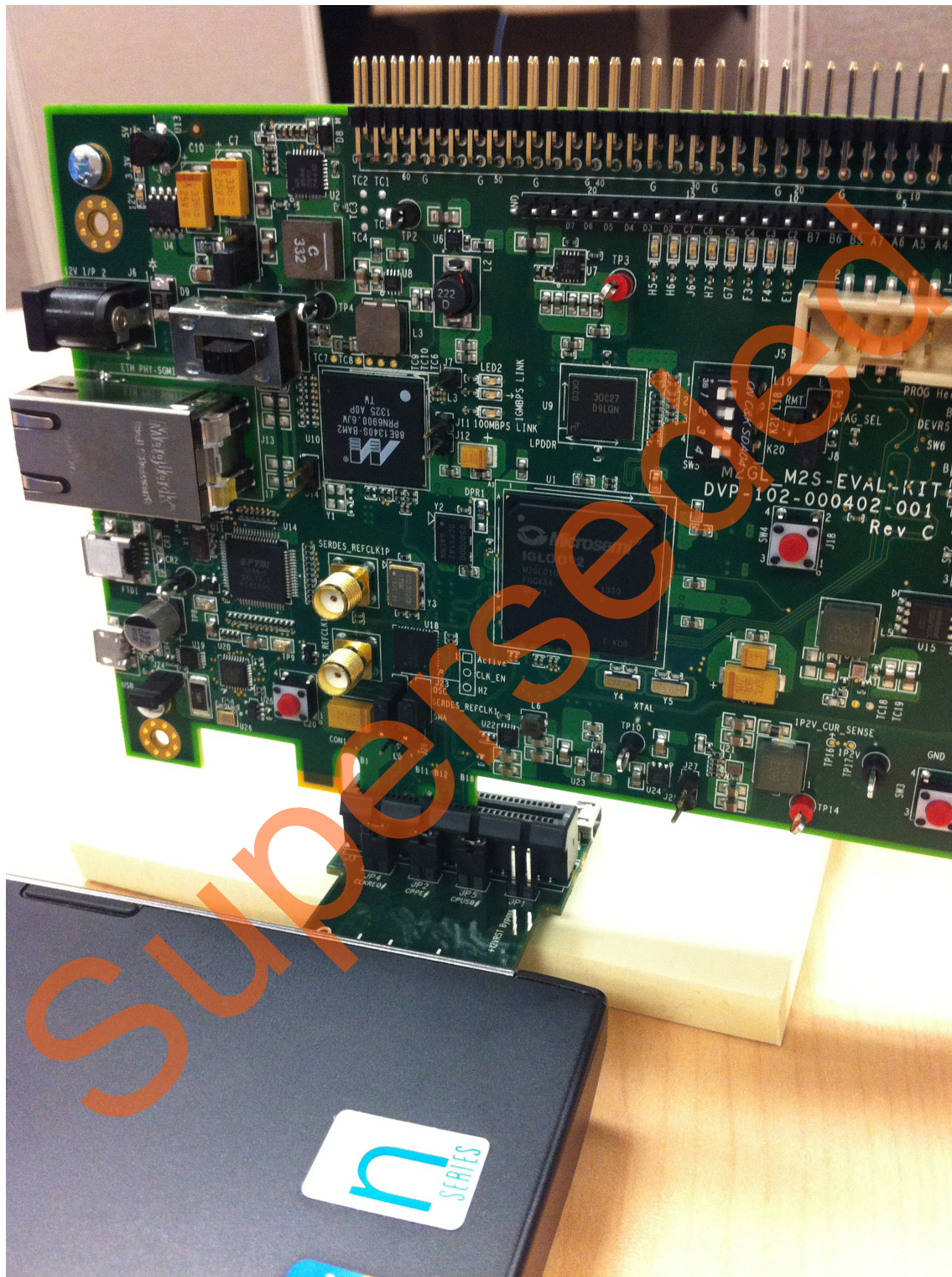

Figure 33 • **PCIe Adapter Card and IGLOO2 Evaluation Kit Connected to Laptop**

Appendix 2: Jumper Locations

Table 4 shows the registers used to interface with the Fabric DMA Controller.

Table 4 • **Fabric DMA Controller**

Register Name	BAR Space	Register Address	Description	
PC_BASE_ADDR	BAR 1	0x8028	Host PC memory base address provided by the driver	
DMA_SIZE	BAR 1	0x8008	Size of DMA transfer	
DMA_DIR	BAR 1	0x800C	DMA direction	
			Direction	**Register value**
			PCIe to DDR	0x11AA1111
			DDR to PCIe	0x11AA2222
			LSRAM to DDR	0x11AA3333
			DDR to LSRAM	0x11AA4444
DMA_CLK_CYCLES	BAR 1	0x8014	Number of clock cycles taken to complete the DMA transfer.	
DMA_STATUS	BAR 1	0x8018	1 - DMA transfer completed 0 - DMA transfer is not completed	
RW_REG	BAR 1	0x0	Scratchpad register for PCIe R/W	
LED_CTRL[7:0]	BAR 0	0xA0	LEDs control register	
SWITCH_STATUS[11:8]	BAR 0	0x90	DIP switch status	

![Microsemi logo]

A – List of Changes

The following table shows important changes made in this document for each revision.

Date	Changes	Page
Revision 2 (October 2015)	Updated the document for Libero v11.6 software release (SAR 71500).	NA
Revision 1 (February 2015)	Initial release	NA

B – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit
http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Visit About Us for sales office listings and corporate contacts.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace 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; 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 3,600 employees globally. Learn more at **www.microsemi.com**.

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

50200585-2/10-15