DG0713
Demo Guide
RTG4 FPGA PCIe Data Plane Demo using Scatter-Gather DMA Controller

Microsemi
a Microchip company

a MICROCHIP company

Microsemi 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
Email: sales.support@microsemi.com
www.microsemi.com

©2021 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), 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. Learn more at www.microsemi.com.

50200713. 6.0 10/21

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 current publication.

1.1 Revision 6.0

Updated the document for Libero SoC v2021.2.

1.2 Revision 5.0

Replaced Figure 15, page 14.

1.3 Revision 4.0

The following is a summary of the changes made in this revision.

- Added Appendix 1: Programming the Device Using FlashPro Express, page 21.
- Added Appendix 2: Running the TCL Script, page 24.
- Removed the references to Libero version numbers.

1.4 Revision 3.0

Updated the document for Libero SoC v12.0 software release.

1.5 Revision 2.0

Updated the document for Libero v11.8 software release.

1.6 Revision 1.0

Revision 1.0 was the first publication of this document.

2 PCIe Data Plane Demo using Scatter-Gather DMA Controller

This demo highlights the high-speed data transfer capability of the RTG4™ device using the PCIe interface. To achieve high-speed and bulk-data transfers, the advanced extensible interface (AXI) based scatter-gather direct memory access (SGDMA) controller is implemented in the FPGA fabric. The SGDMA controller performs a bulk-data transfer between non-contiguous memory locations defined by buffer descriptors. The SGDMA controller is useful in systems, where getting bulk contiguous free memory locations is difficult due to memory management concepts of operating systems.

Windows kernel mode PCIe device driver is developed for interacting with the RTG4 PCIe endpoint from the host PC. These Windows device drivers are developed using the Windows Driver Kit (WDK) platform.

An application GUI (PCIe_Data_Plane_Demo) that runs in the host PC is provided for setting up and initiating the DMA transactions between the host PC memory and the DDR3 memory of the RTG4 development kit through the PCIe interface. A user space application interface is developed for the GUI to interact with the PCIe driver.

The high-speed serial interface (SERDESIF) available in the RTG4 device provides a fully hardened PCIe endpoint implementation and is compliant with the PCIe Base Specification Revision 2.0 and 1.1. For more information about this, refer to *UG0567: RTG4 FPGA High-Speed Serial Interface User Guide*.

2.1 Design Requirements

The following table lists the design requirements to run the design.

Table 1 • **Design Requirements**

Requirement	Version
Hardware	
RTG4 Development Kit	Rev C or later
Host PC with 8 GB RAM and PCIe 2.0 Gen1 compliant slot with x4 or higher width	64-bit Windows 7 and 10
Software	
Libero® System-on-Chip (SoC) FlashPro Express	**Note:** Refer to the `readme.txt` file provided in the design files for the software versions used with this reference design.
Host PC Drivers (provided along with the design files)	–
PCIe Demo application (provided along with the design files)	–

Note: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

2.2 Prerequisites

Before you start:

1. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location: *https://www.microsemi.com/product-directory/design-resources/1750-libero-soc*

2.3 Demo Design

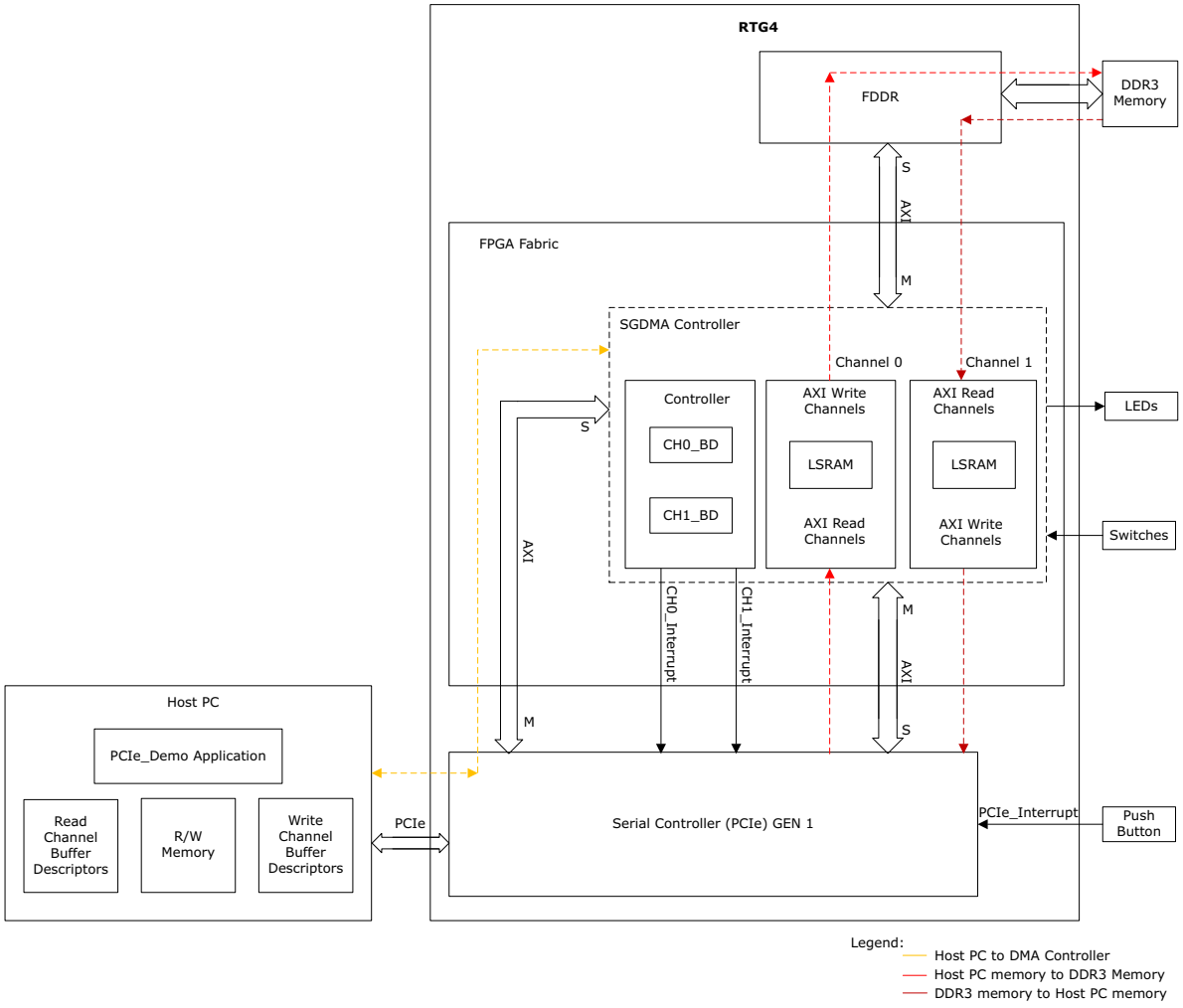
The demo design files are available for download from the following path in the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=rtg4_dg0713_df*

The PCIe_Data_Plane_Demo application on the host PC initiates the DMA transfers through the PCIe device drivers. The drivers on the host PC allocate the memory, create the buffer descriptors, and trigger the SGDMA controller in the FPGA fabric by accessing the controller registers through BAR0 space. The buffer descriptor contains the DMA source address, destination address, DMA size, and the DMA direction details. For more information on the buffer descriptor and its bit descriptions, refer to Buffer Descriptor (16 bytes), page 26.

The SGDMA controller fetches the buffer descriptors from the host PC and performs DMA transfers using the DMA channel 0 and the DMA channel 1. These DMA channels share the AXI read/write channels of the PCIe AXI and FDDR AXI slave interface to perform read and write operations to DDR/PC memories. The DMA channel 0 handles the host PC memory to DDR memory DMA transfer. The DMA channel 1 handles the DDR memory to host PC memory DMA transfer.

The following figure shows the top-level block diagram of the demo design:

Figure 1 • **PCIe Data Plane Demo- Block Diagram**



In this design, the FDDR (west) controller is configured to access the DDR3 memory in x32 mode. The FDDR clock is configured to 320 MHz (640 Mbps DDR) with an 80 MHz DDR_FIC clock for an aggregate memory bandwidth of 1280 Mbps. The PCIe AXI interface clock and fabric DMA controller clock are configured to 80 MHz.

2.3.1 Features

Features of the PCIe Data Plane are:

- High-speed data transfers between the host PC memory and the DDR memory.
- Each buffer descriptor supports the DMA size of 8 B to 1 MB.
- Enables continuous DMA transfers for observing throughput variations.
- The PCIe_Data_Plane_Demo application displays:
  - The PCIe negotiated link width and the link speed.
  - The PCIe configuration space.
  - The position of DIP Switches on the RTG4 Development Kit.
- PCIe_Data_Plane_Demo application controls the LEDs on the RTG4 Development Kit.
- Enables the read and write operations through the BAR1 to 4 KB memory in the FPGA fabric.
- Interrupts the host PC when the Push button is pressed. The PCIe_Data_Plane_Demo application displays the count value of the number of interrupts sent from the board.

2.3.2 Description

The design supports the following types of data transfers:

- Host PC Memory to RTG4 DDR Memory (Read)
- RTG4 DDR Memory to Host PC Memory (Write)

The PCIe_Data_Plane_Demo application requests the PCIe device driver for DMA transfer through the user space application interface. Then, the PCIe device driver finds the available memory locations and initializes the memory read buffers. It also creates the buffer descriptor chain for the different memory locations and sends the base address of the first buffer descriptor. The DMA start command to the SGDMA controller in the FPGA fabric.

To perform the DMA transfers the SGDMA controller fetches the 8 buffer descriptors from the base address of the first buffer descriptor. The SGDMA controller is designed to get the 8 buffer descriptors (128 bytes) using a 16 beat AXI burst for optimum bandwidth utilization.

At the end of the DMA transfer, the fabric DMA controller interrupts the host PC and provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application.

The DMA transfer flow from the host PC memory to the DDR memory flow is shown in Figure 2, page 5. The DMA transfer flow from DDR memory to host PC memory is shown in Figure 3, page 6.

Figure 2 • **Data Transfer from Host PC Memory to RTG4 DDR Memory**

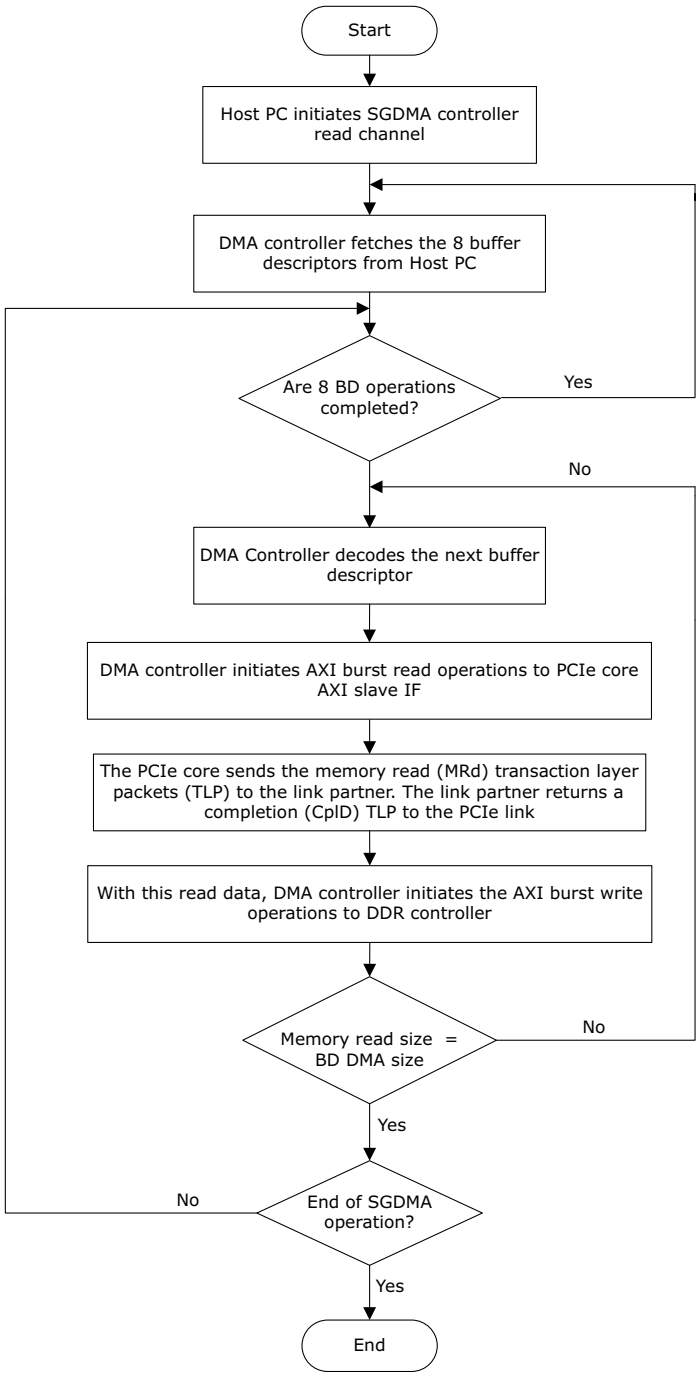
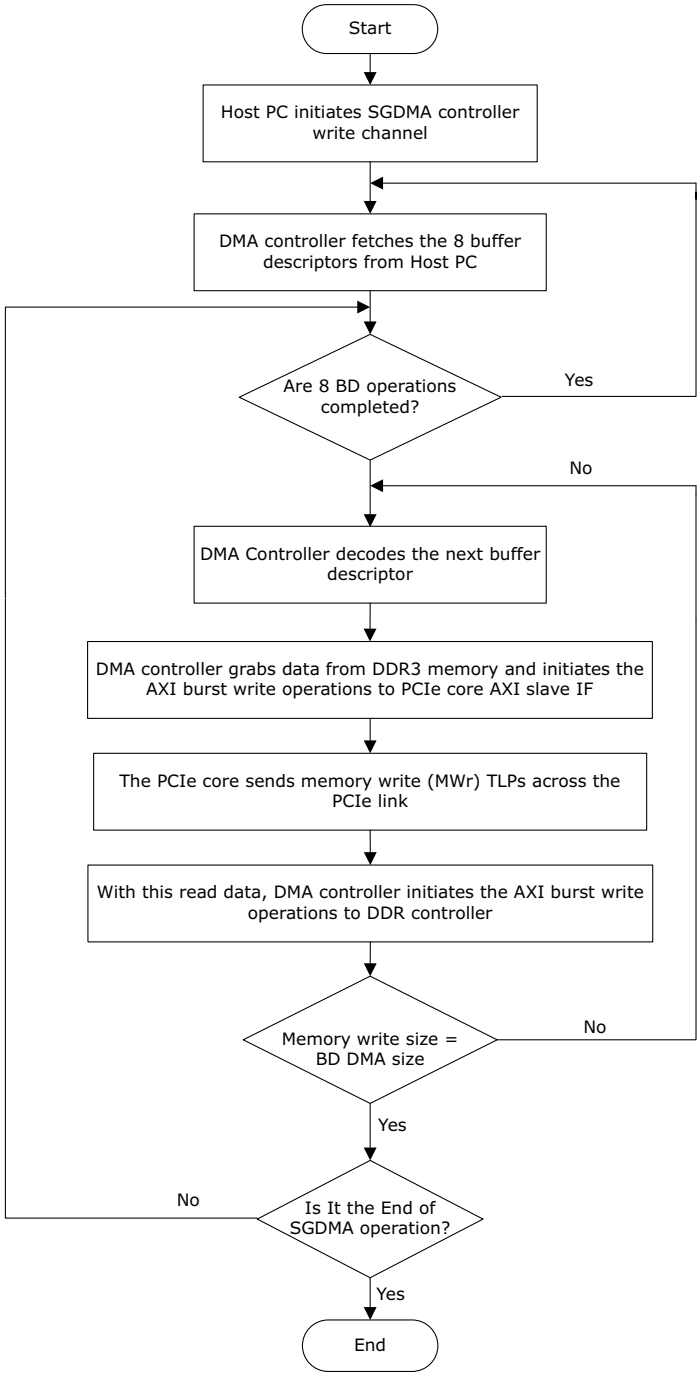


Figure 3 • **Data Transfer from RTG4 DDR Memory to Host PC Memory**



2.3.3 Throughput Calculation

The design implements a timer to measure the throughput of DMA transfers. The throughput includes all of the overhead of the AXI, PCIe, and DMA controller transactions. But it excludes the reading and decoding of buffer descriptors.

The design implements the following steps to measure the throughput*:

1. Fetches the buffer descriptors from the host PC and decodes them.
2. Set up the DMA controller for the complete transfer.
3. Starts the timer and the DMA controller.
4. Initiates the data transfer for the requested number of bytes.
5. Waits until the DMA transfer is completed.
6. Records the number of clock cycles used for data transfer.

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

2.4 Setting up the Demo Design

This section describes the following procedures to set up the design:

2.4.1 Setting Up the Board

The following steps describe how to set up the hardware demo for the RTG4 Development Kit:

1. Connect the jumpers on the RTG4 Development Kit, as shown in the following table.

Table 2 • **Jumper Pin Connections**

Jumper	Pin (From)	Pin (To)	Comments
J11, J17, J19, J23, J26, J21, J32, J27	1	2	Default
J16	2	3	Default
J33	1 3	2 4	Default

Note: Switch OFF the power supply switch **SW6** while connecting the jumpers on the RTG4 Development Kit board.

2. Connect the host PC to the J47 connector using the USB cable.
3. Connect the USB cable (mini USB to Type-A USB cable) to J47 of the RTG4 Development Kit board and another end of the cable to the USB port of the host PC.
4. Switch ON the power supply switch, **SW6**.

2.4.2 Programming the Device

Program the RTG4 Development Kit with the job file provided as part of the design files using FlashPro Express software, refer to Appendix 1: Programming the Device Using FlashPro Express, page 21.

2.4.3 Connecting RTG4 Development Kit to Host PC PCIe Slot

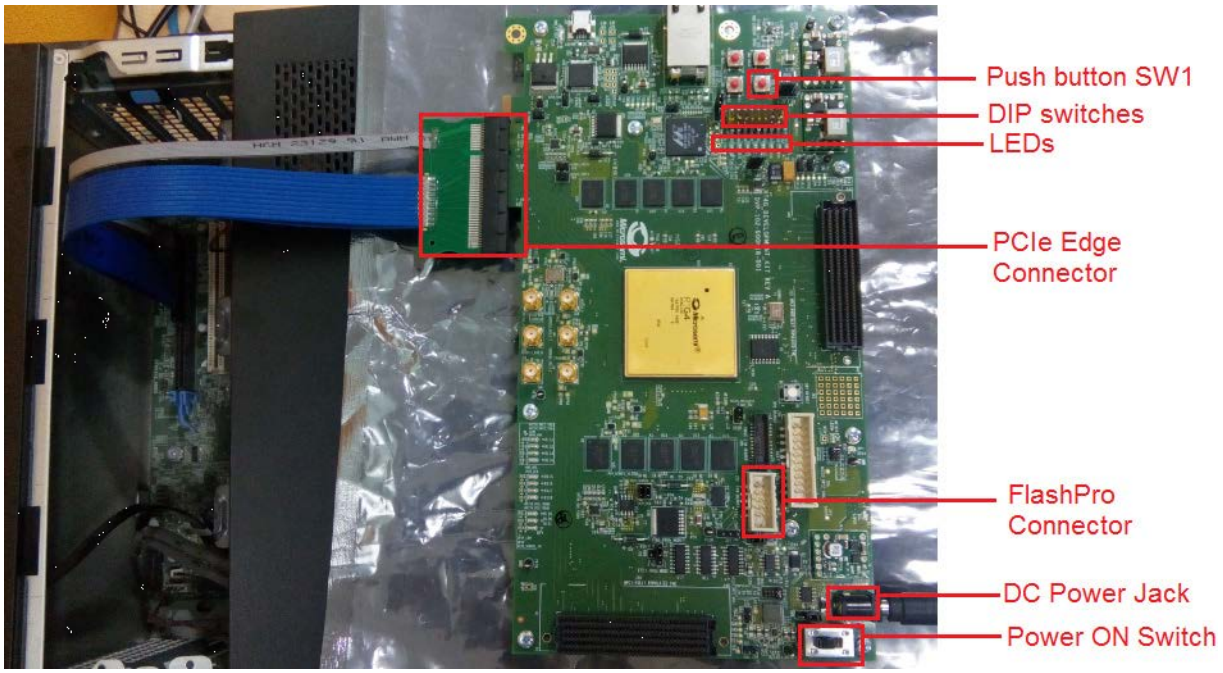
The following steps describe how to connect the RTG4 Development Kit to the host PC:

1. After successful programming, shut down the host PC.
2. Connect the J230 - PCIe Edge connector of the RTG4 Development Kit to the PCIe slot of the host PC through the PCI Edge Card Ribbon Cable.

Note: Ensure that the host PC is powered off while inserting the PCIe Edge Connector. Otherwise, the PCIe device will not be detected properly. After inserting, the host PC may wake-up or start due to a glitch on the PCIe WAKEn pin. The RTG4 device does not support cold sparing, and hence causes the glitch on RTG4 PCIe I/Os while inserting the PCIe edge connector.

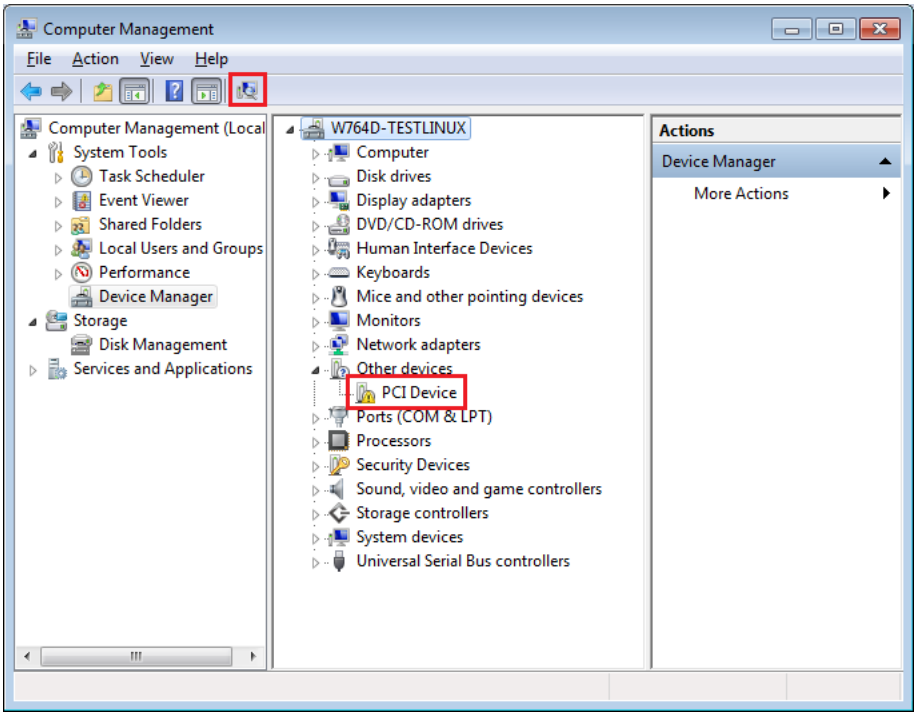
The following figure shows the board setup for the host PC in which the RTG4 Development Kit is connected to the host PC PCIe slot.

Figure 4 • **RTG4 Development Kit Setup**



3. Switch **ON** the host PC and check the Device Manager of the Host PC for PCIe Device. If the device is not detected, power cycle the RTG4 Development Kit and click **scan for hardware changes** in the **Device Manager** window. The following figure shows the device manager window with a PCI device.

Figure 5 • **Device Manager—PCIe Device Detection**



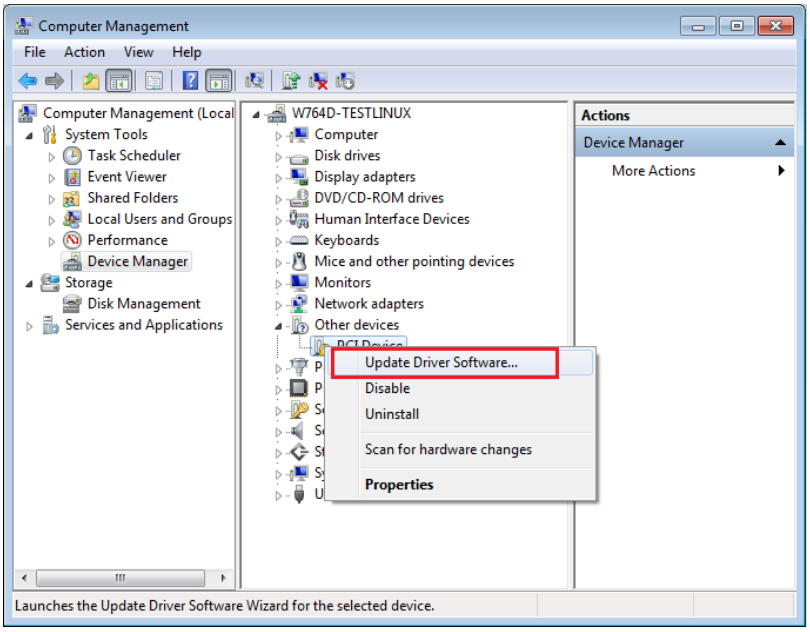
Note: If the device is still not detected, check if the BIOS version in the host PC is the latest and if PCI is enabled in the host PC BIOS.

2.4.4 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.

Figure 6 • **Update Driver Software**



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

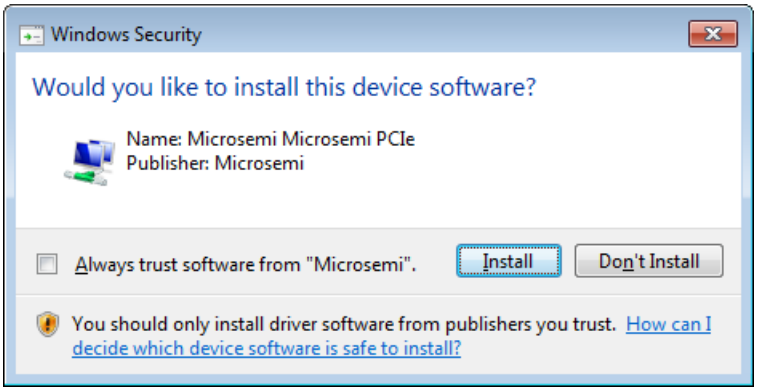
Figure 7 • **Browse for Driver Software**

3. Browse the driver's folder: <download files>/PCIe Drivers/Win_64bit_PCie_Drivers and click **Next** as shown in the following figure.

Figure 8 • **Browse for Driver Software Continued**

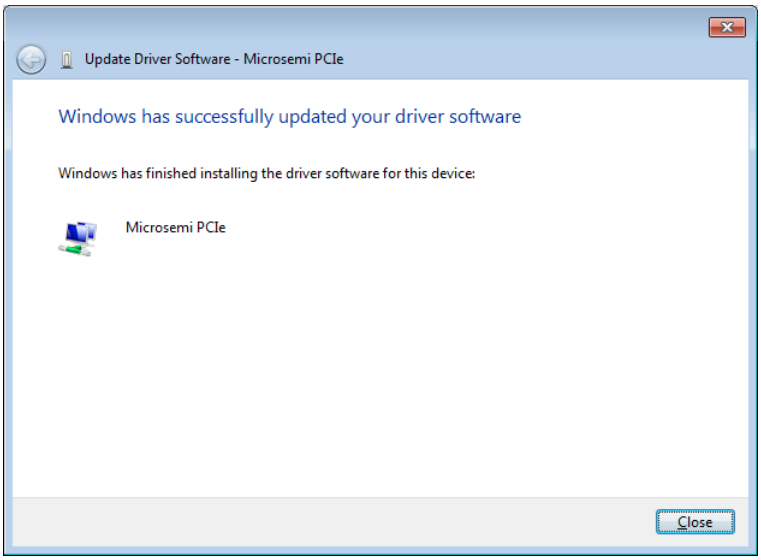
4. **Windows Security** dialog box is displayed and click **Install**, as shown in the following figure.

Figure 9 • **Windows Security**



After successful driver installation, a message window appears, as shown in the following figure.

Figure 10 • **Successful Driver Installation**



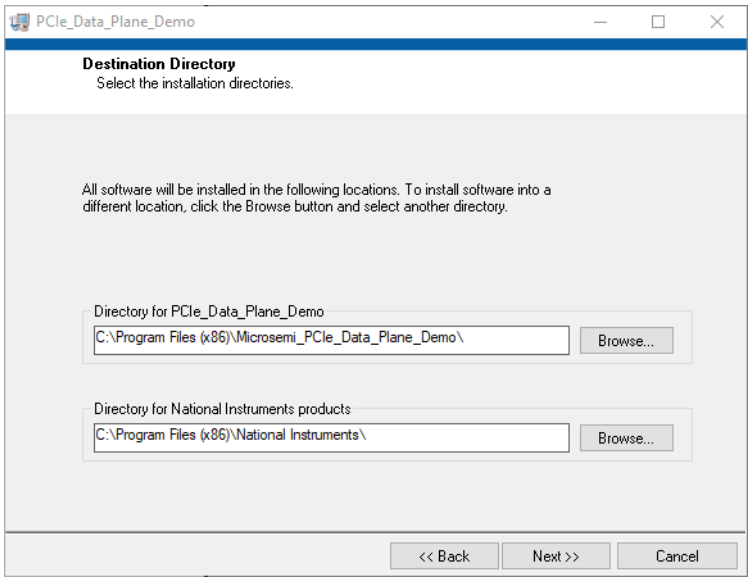
2.4.5 Installing the PCIe_Data_Plane_Demo Application GUI

The PCIe_Data_Plane_Demo application is a simple GUI that runs on the host PC to communicate with the RTG4 PCIe endpoint device. It provides the PCIe link status, driver information, and the demo controls. The PCIe_Data_Plane_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_Data_Plane_Demo application:

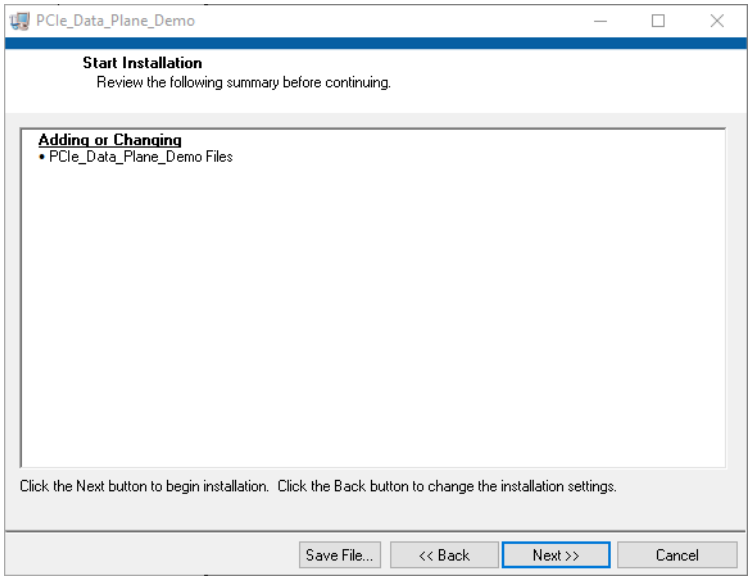
1. Go to <Download Folder>\rtg4_dg0713_df\GUI\PCIe Data Plane Demo Installer V1.0, and double-click setup.exe, do not change the default options as shown in the following figure.

Figure 11 • **PCIe Demo Application Installation**



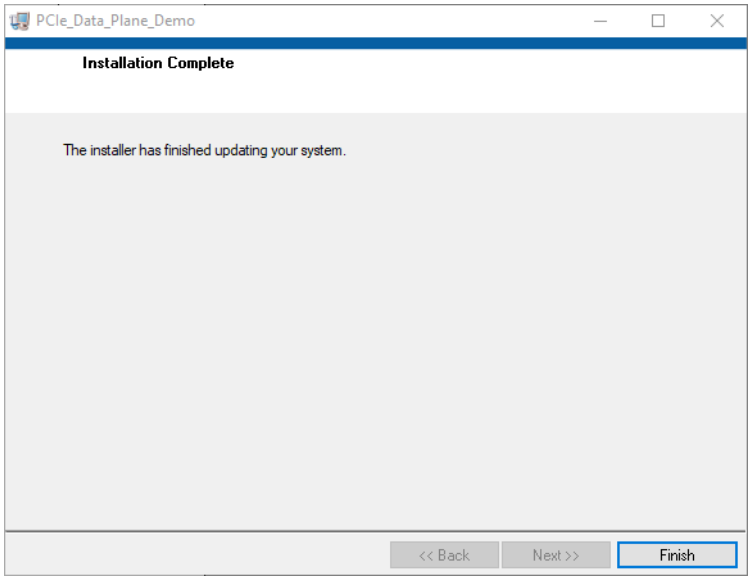
2. Click **Next** to start the installation, as shown in the following figure.

Figure 12 • **PCIe Demo Application Installation**



3. Click **Finish** to complete the installation.
4. The installation completes successfully, as shown in the following figure.

Figure 13 • **Successful Installation of PCIe Demo Application**



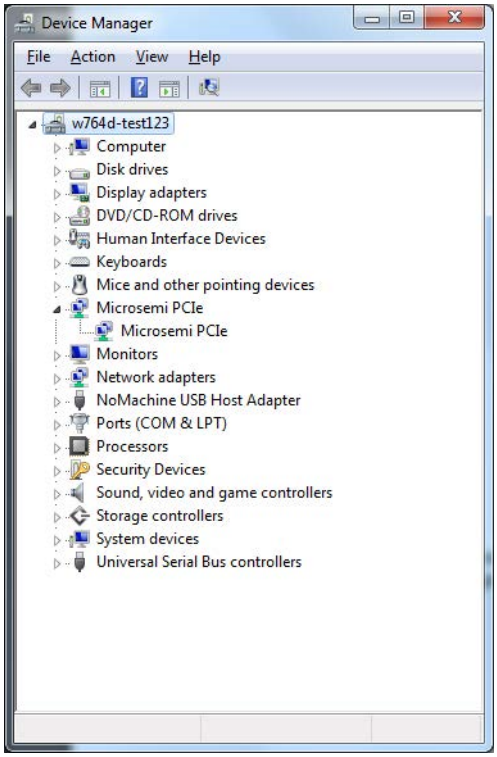
5. Restart the host PC.

2.5 Running the Design

The following steps describe how to run the demo design:

1. Check for the **Microsemi PCIe** device in the host PC **Device Manager** as shown in the following figure.

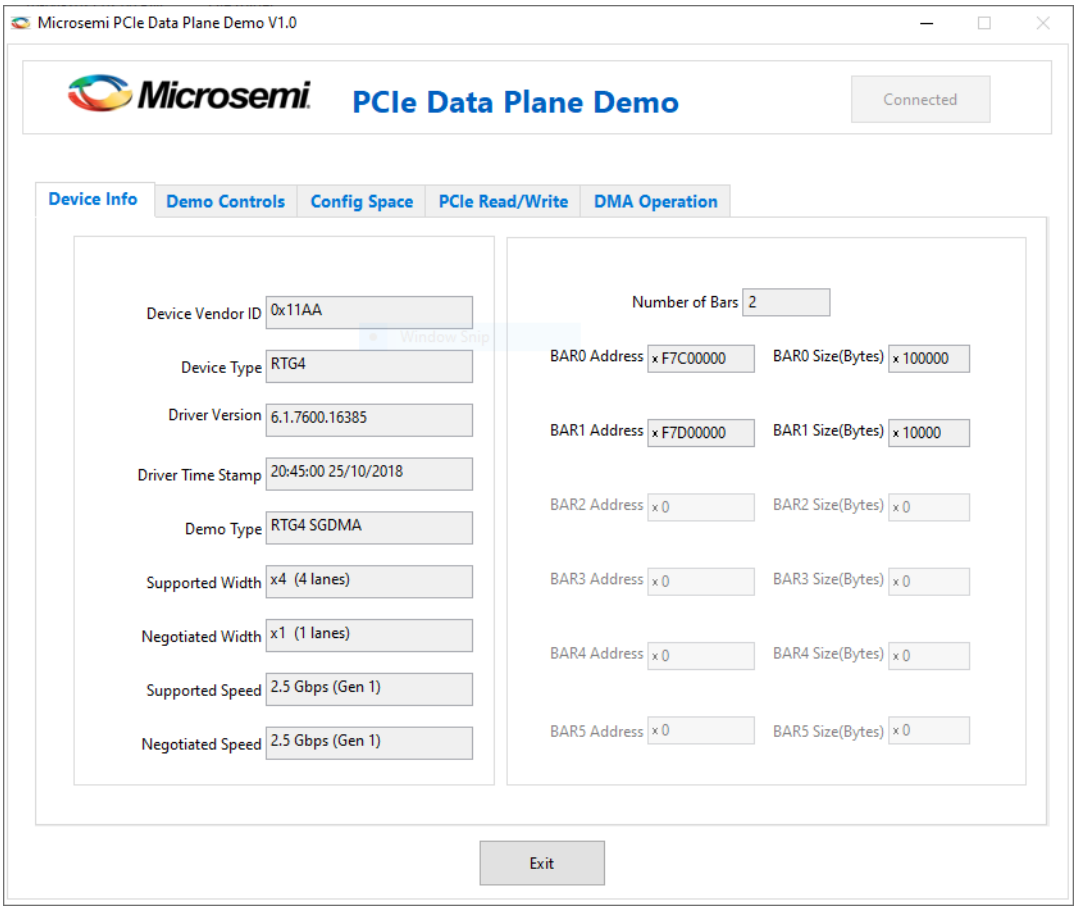
Figure 14 • **Device Manager—PCIe Device Detection**



Note: If a warning appears on the Microsemi PCIe driver in the **Device Manager**, uninstall them and start from Step 1 of driver installation.

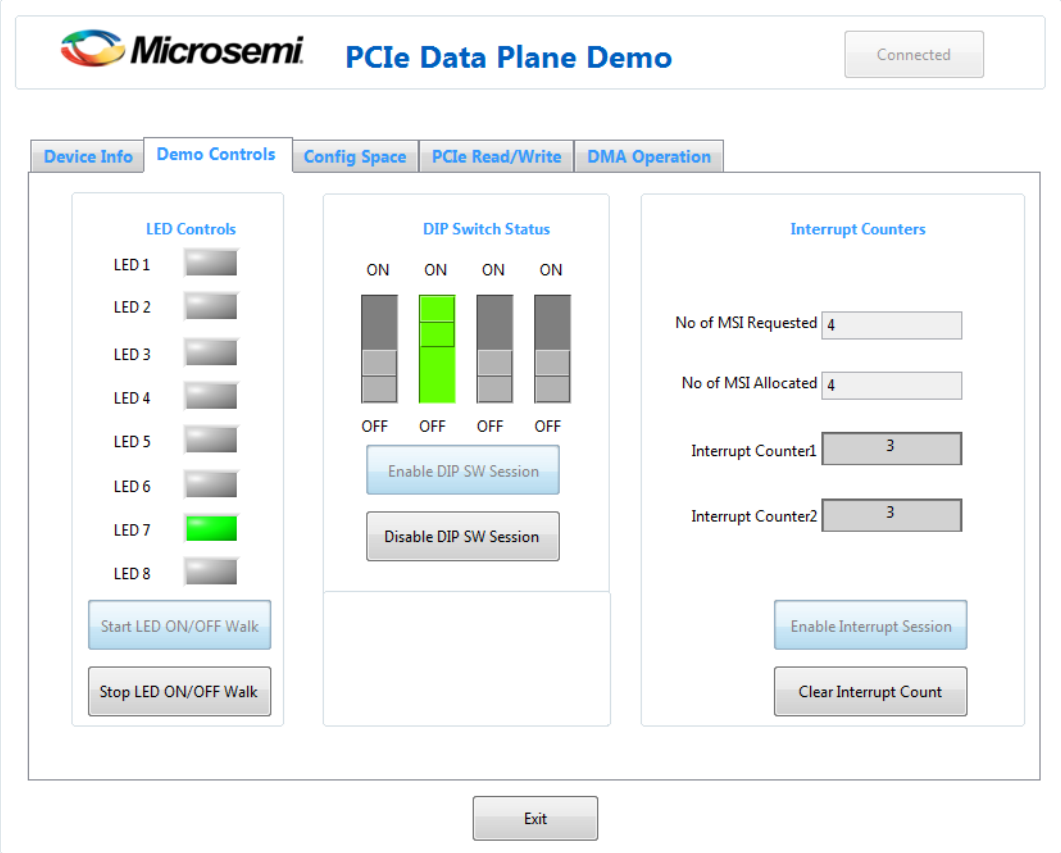
2. Invoke the Microsemi PCIe Data Plane demo application from **All Programs > Microsemi PCIe Data Plane Demo > PCIe_Data_Plane_Demo**.
3. Click **Connect**. The application detects and displays the connected kit, demo type, PCIe link width, and the link speed, as shown in the following figure.

Figure 15 • **Device Info**



4. Click **Demo Controls** tab to display the LEDs options, DIP switch positions, and the interrupt counters. Controlling LEDs, getting the DIP switch status, and monitoring the interrupts can be done simultaneously, as shown in the following figure.

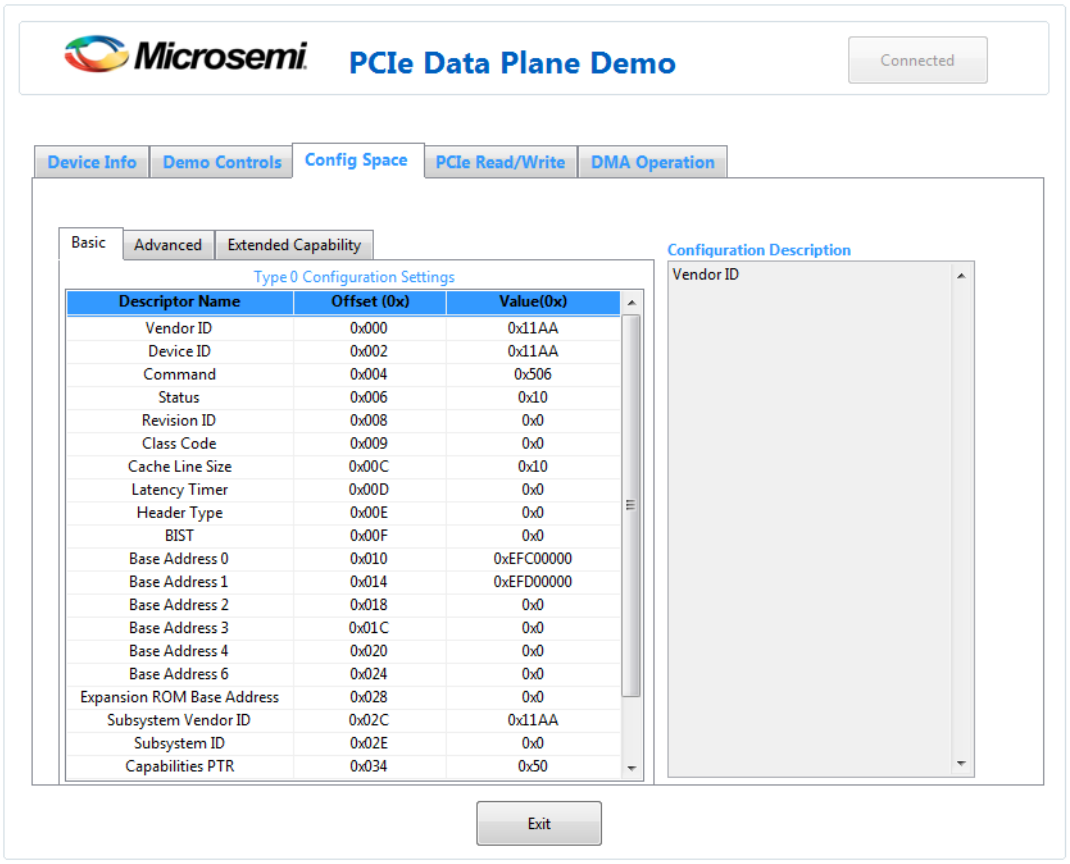
Figure 16 • **Demo Controls Continued**



Note: Interrupt Counter1 and Counter4 are allocated for SW1 and SW2 events. Interrupt Counter2 and Counter3 are allocated for the SGDMA controller.

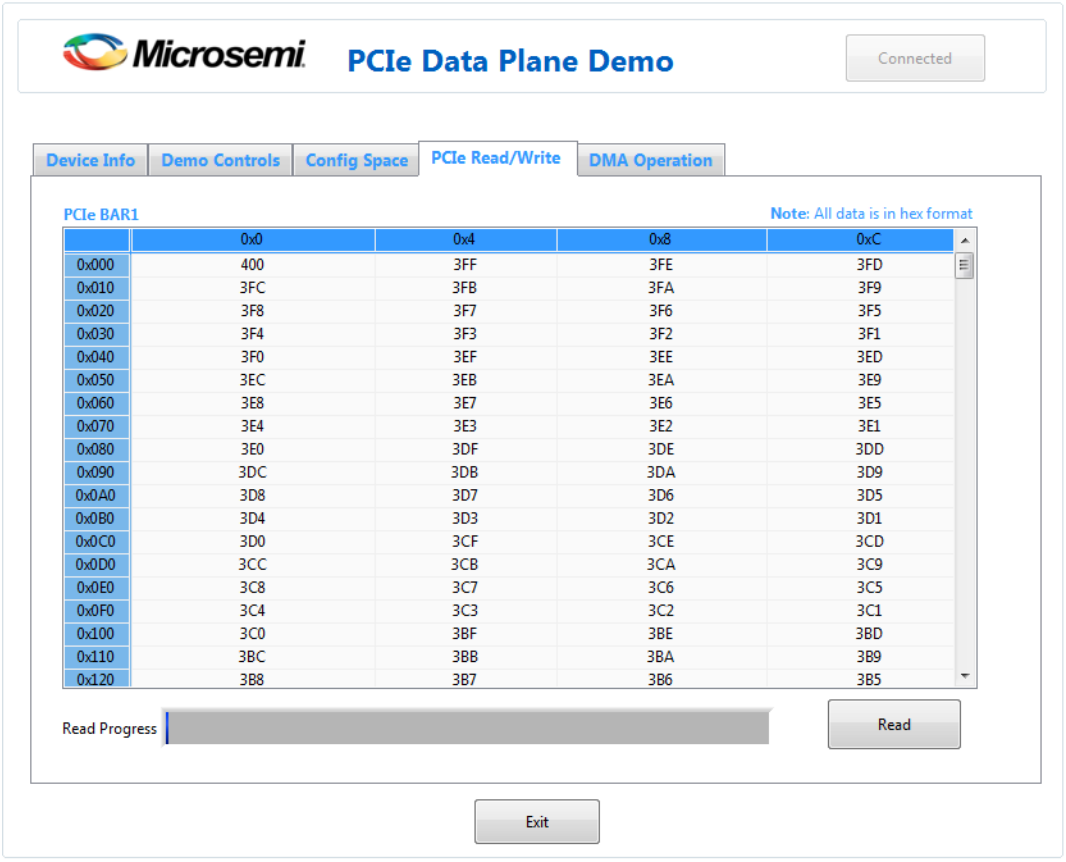
5. Click **Config Space** tab to view the details about the PCIe configuration space as shown in the following figure.

Figure 17 • **Configuration Space**



6. Click **PCIe Read/Write** tab to perform read and write to LSRAM using BAR1 space. Click **Read** to read the 4 KB memory mapped to BAR1 space, as shown in the following figure.

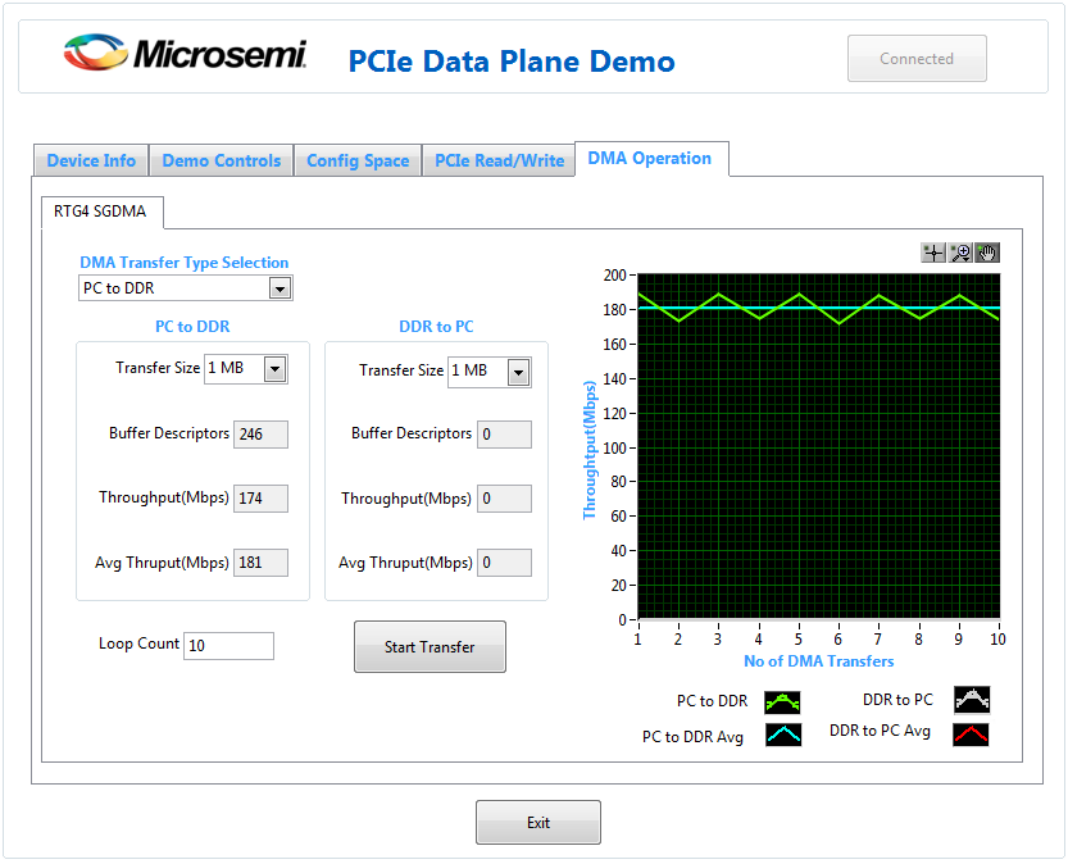
Figure 18 • **PCIe BAR1 memory access**



	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

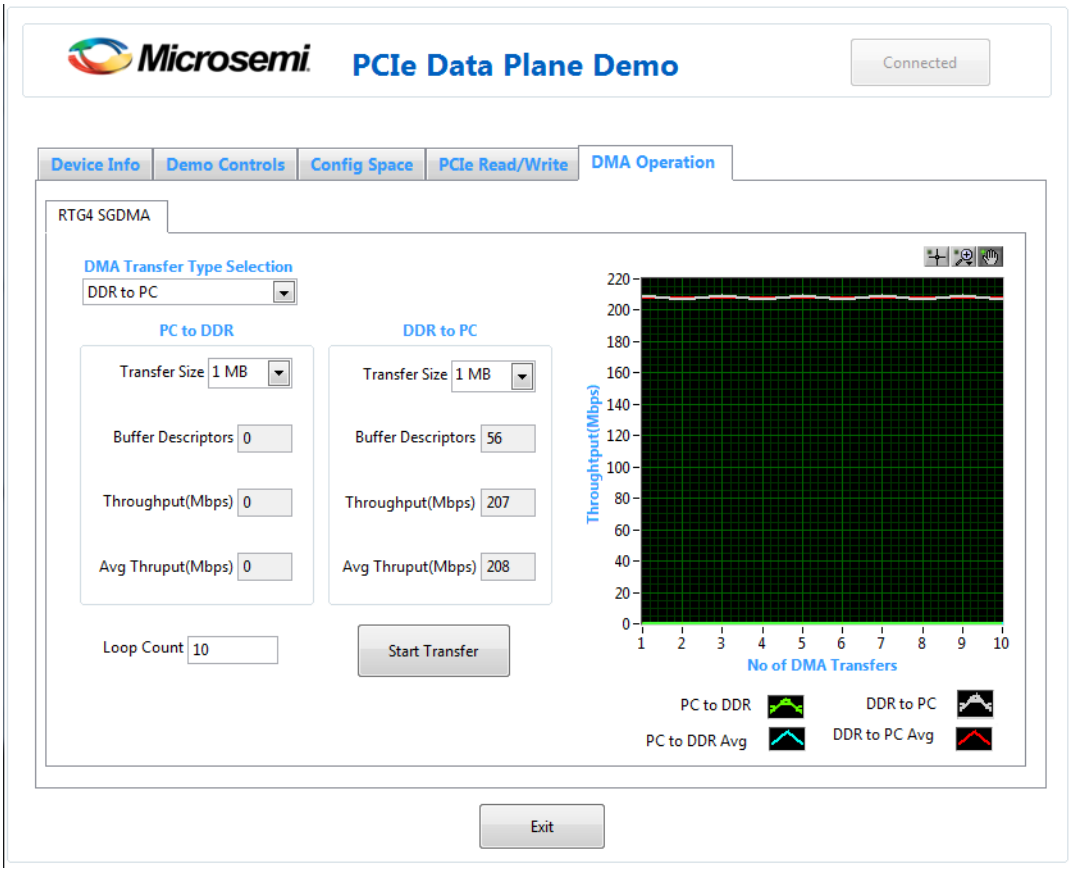
7. Click **DMA Operation** tab. Select **PC to DDR** as the **DMA Transfer Type Selection,** Loop Count, and click **Start Transfer** to transfer the data from host PC to RTG4 DDR memory. The transfer size can be selected from 4 KB to 1 MB. After successful DMA operation, the GUI displays the throughput and the number of buffer descriptors created by the driver as shown in the following figure.

Figure 19 • **PC Memory to DDR DMA**



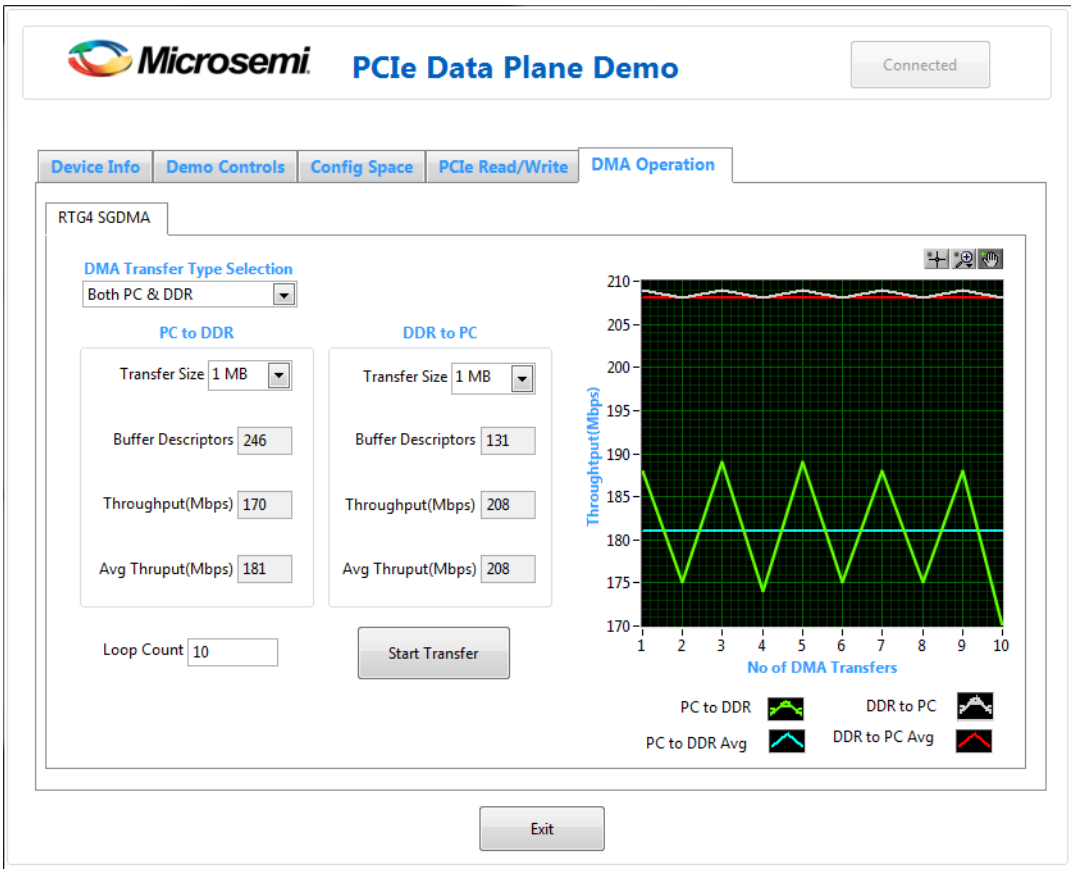
8. Select **DDR to PC** as **DMA Transfer Type Selection** and click **Start Transfer** to transfer the data from RTG4 DDR memory to host PC as shown in the following figure.

Figure 20 • **DDR to PC Memory DMA**



9. Select **Both PC & DDR** as **DMA Transfer Type Selection** and click **Start Transfer** to transfer the data from RTG4 DDR memory to the host PC as shown in the following figure.

Figure 21 • **Both DMA Operations**



10. To write to the BAR1, select the memory location and overwrite the existing value. After successful write operation, GUI shows the updated memory location in green color.

2.5.1 Summary

This demo shows how to implement a PCIe Data Plane design using AXI based SGDMA controller. The throughput for data transfers depends on the host PC system configuration and the type of the PCIe slots used.

The following table lists the throughput values observed on the HP Workstation Z230 PCIe slot 4.

Table 3 • **Throughput Values**

DMA Transfer Type	Throughput (MB/s)
Host PC memory to DDR (read)	170 MB/s
DDR to Host PC memory (write)	220 MB/s
Host PC memory to DDR (read)/DDR to Host PC memory (write)	170/220 MB/s

3 Appendix 1: Programming the Device Using FlashPro Express

This section describes how to program the RTG4 device with the programming job file using FlashPro Express.

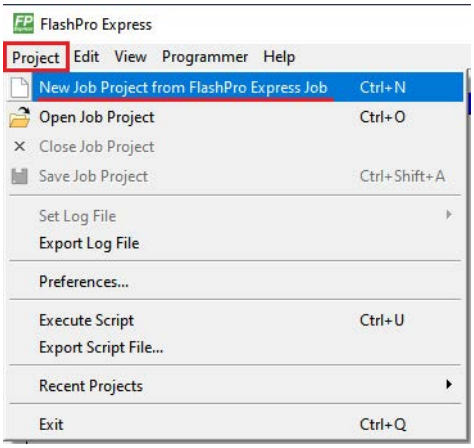
To program the device, perform the following steps:

1. Ensure that the jumper settings on the board are the same as those listed in *Table 3 of UG0617: RTG4 Development Kit User Guide*.
2. Optionally, jumper **J32** can be set to connect pins 2-3 when using an external FlashPro4, FlashPro5, or FlashPro6 programmer instead of the default jumper setting to use the embedded FlashPro5.

Note: The power supply switch, **SW6** must be switched **OFF** while making the jumper connections.

3. Connect the power supply cable to the **J9** connector on the board.
4. Power **ON** the power supply switch **SW6**.
5. If using the embedded FlashPro5, connect the USB cable to connector **J47** and the host PC. Alternatively, if using an external programmer, connect the ribbon cable to the JTAG header **J22** and connect the programmer to the host PC.
6. On the host PC, launch the **FlashPro Express** software.
7. Click **New** or select **New Job Project from FlashPro Express Job** from **Project** menu to create a new job project, as shown in the following figure.

Figure 22 • **FlashPro Express Job Project**



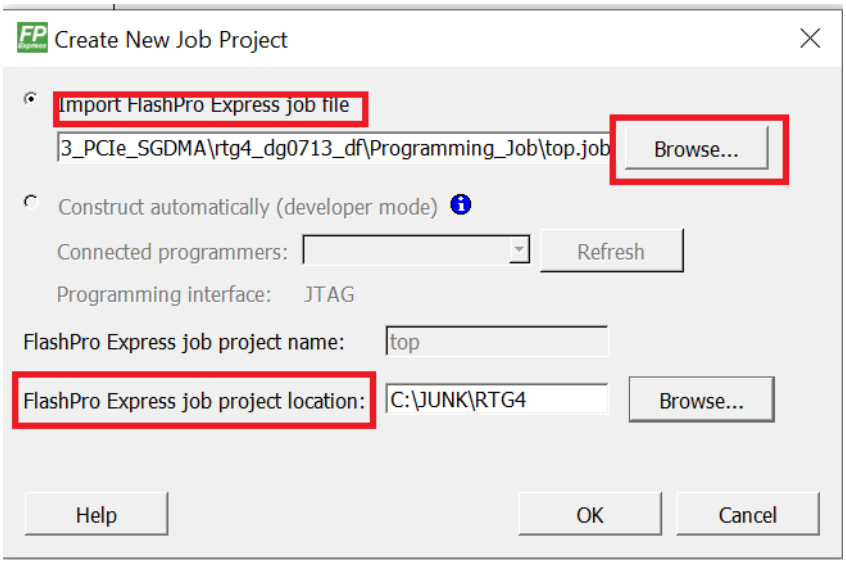
8. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:

- **Programming job file:** Click **Browse**, and navigate to the location where the .job file is located and select the file. The default location is:

  `<download_folder>\rtg4_dg0713_df\Programming_Job`

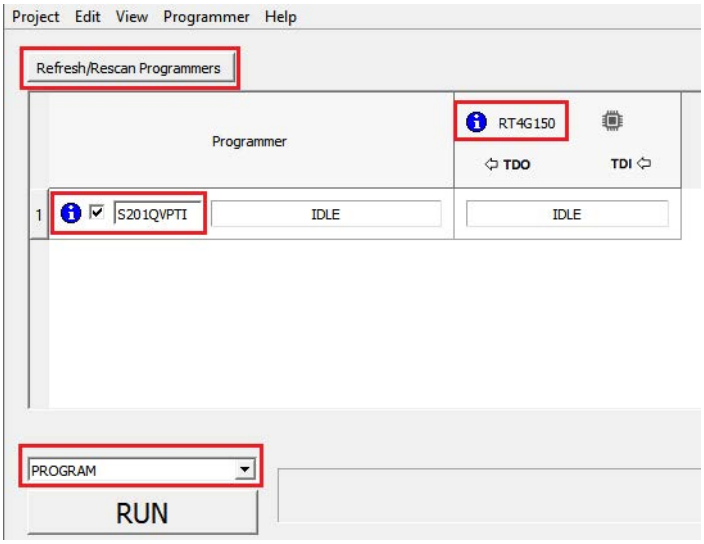
- **FlashPro Express job project location:** Click **Browse** and navigate to the desired FlashPro Express project location.

Figure 23 • **New Job Project from FlashPro Express Job**



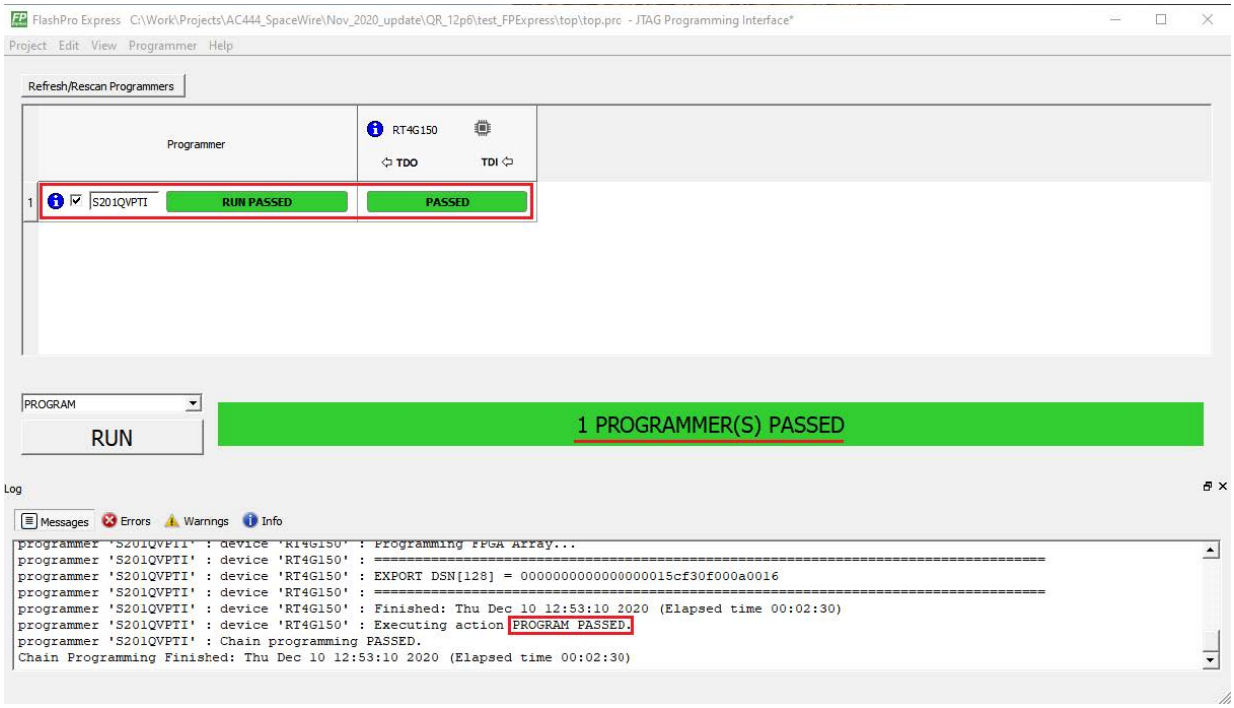
9. Click **OK**. The required programming file is selected and ready to be programmed in the device.
10. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan** Programmers.

Figure 24 • **Programming the Device**



11. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure.

Figure 25 • **FlashPro Express—RUN PASSED**



12. Close **FlashPro Express** or click **Exit** in the Project tab.

4 Appendix 2: Running the TCL Script

TCL scripts are provided in the design files folder under directory TCL_Scripts. If required, the design flow can be reproduced from Design Implementation till generation of job file.

To run the TCL, follow the steps below:

1. Launch the Libero software
2. Select **Project > Execute Script....**
3. Click Browse and select `script.tcl` from the downloaded TCL_Scripts directory.
4. Click **Run**.

After successful execution of TCL script, Libero project is created within TCL_Scripts directory.

For more information about TCL scripts, refer to **rtg4_dg0713_df/TCL_Scripts/readme.txt.**

Refer to *Libero® SoC TCL Command Reference Guide* for more details on TCL commands. Contact Technical Support for any queries encountered when running the TCL script.

5 Appendix 3: Demo Registers and Buffer Descriptor Details

5.1 SGDMA Registers (BAR1)

The following table lists the registers used to interface with the SGDMA controller. These registers are in the BAR1 address space.

Table 4 • **SGDMA Registers**

Register Name	Width	BAR or Address Space	Register Address	Description
RW_REG	32 bit	BAR1	0xFC00	Scratch pad register
DMA_CR	32 bit	BAR1	0xFC04	[15:0] → 0x1 CH0 start [31:16] → 0x1 CH1 start
BDA_CH0 (PC to DDR)	32 bit	BAR1	0xFC08	Host PC buffer descriptor address provided by the driver
DMA_SR_CH0	64 bit	BAR1	0xFC10	DMA status register DMA_SR[63:32] → No.of clks DMA_SR[31:24] → Error code DMA_SR[23:0] → No.of bytes
BDA_CH1 (read)	32 bit	BAR1	0xFC18	Host PC buffer descriptor address provided by the driver
DMA_SR_CH1 (DDR to PC)	64 bit	BAR1	0xFC20	DMA status register DMA_SR[63:32] → No.of clks DMA_SR[31:24] → Error code DMA_SR[23:0] → No.of bytes
CLK_FREQ	32 bit	BAR1	0xFC28	DMA clock frequency. Read only
switches	32 bit	BAR1	0xFC30	DIP switch status
LEDs	32 bit	BAR1	0xFCA0	LEDs control register

5.2 Buffer Descriptor (16 bytes)

The following table lists the buffer descriptor and its bit descriptions.

Table 5 • **Buffer Descriptor and Bit Descriptions**

Reserved[63:32]			Host PC Address[31:0]		
Reserved[63:56]	DDR offset[55:32]	Valid descriptor[31:24]	Last transfer[23]	Direction[22:21]	Size[20:0]

- Host PC Address[31:0]: 32-bit memory buffer address
- Size[20:0]: Size of the DMA operation
  - Maximum DMA size is 1MBytes
  - Minimum DMA size is 8 bytes
- Direction[22:21]: indicates read/ write type of DMA operation
  - “01” is DMA read operation
  - “10” is DMA write operation
- Last transfer[23]: indicates last buffer descriptor of DMA operation
- Valid descriptor[31:24]: “AA” indicates valid buffer descriptor
- DDR offset[55:32]: DDR memory offset address
  - DDR memory base address is 0x0000_0000