DG0622
Demo Guide
RTG4 FPGA PCIe Data Plane Demo using Two Channel Fabric DMA - Libero SoC v12.0
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.
## Contents

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

2 RTG4 FPGA PCIe Data Plane Demo using Two Channel Fabric DMA ........ 2
   2.1 Design Requirements .......................................................... 2
   2.2 Demo Design ...................................................................... 2
      2.2.1 Demo Design Features .................................................... 3
      2.2.2 Demo Design Description .............................................. 4
      2.2.3 Throughput Calculation ............................................... 5
   2.3 Setting Up the Demo Design ................................................ 6
      2.3.1 Programming the Device Using FlashPro Express ........... 6
      2.3.2 Connecting RTG4 Development Kit to Host PC PCIe Slot ... 9
      2.3.3 Drivers Installation ....................................................... 10
      2.3.4 Installing PCIe_Data_Plane_Demo Application GUI .......... 12
   2.4 Running the Design ........................................................... 14
   2.5 Summary .......................................................................... 18

3 Appendix: Known Issues ......................................................... 20

4 Appendix: Register Details ...................................................... 21
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>PCIe Data Plane Demo Block Diagram</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>FlashPro Express Job Project</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3</td>
<td>New Job Project from FlashPro Express Job</td>
<td>7</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Programming the Device</td>
<td>7</td>
</tr>
<tr>
<td>Figure 5</td>
<td>FlashPro Express—RUN PASSED</td>
<td>8</td>
</tr>
<tr>
<td>Figure 6</td>
<td>RTG4 Development Kit Setup</td>
<td>9</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Device Manager - PCIe Device Detection</td>
<td>10</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Update Driver Software</td>
<td>10</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Browse for Driver Software</td>
<td>11</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Browse for Driver Software Continued</td>
<td>11</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Windows Security</td>
<td>11</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Successful Driver Installation</td>
<td>12</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Installing PCIe_Data_Plane_Demo Application</td>
<td>12</td>
</tr>
<tr>
<td>Figure 14</td>
<td>PCIe_Data_Plane_Demo Application Installation</td>
<td>13</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Successful Installation of PCIe_Data_Plane_Demo Application</td>
<td>13</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Device Manager - PCIe Device Detection</td>
<td>14</td>
</tr>
<tr>
<td>Figure 17</td>
<td>PCIe Device Information</td>
<td>15</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Demo Controls</td>
<td>16</td>
</tr>
<tr>
<td>Figure 19</td>
<td>PCIe Configuration Space</td>
<td>16</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Read and Writes to Scratchpad Register</td>
<td>17</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Continuous DMA Operations with DMA Transfer Type Selection as PC to LSRAM</td>
<td>18</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Hot Reset Detection Block</td>
<td>20</td>
</tr>
</tbody>
</table>
Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Design Requirements</td>
<td>2</td>
</tr>
<tr>
<td>Table 2</td>
<td>RTG4 Development Kit Jumper Settings</td>
<td>6</td>
</tr>
<tr>
<td>Table 3</td>
<td>Throughput Summary</td>
<td>19</td>
</tr>
<tr>
<td>Table 4</td>
<td>Register Details</td>
<td>21</td>
</tr>
</tbody>
</table>
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 v12.0 software release.

1.2 Revision 5.0
Updated the document for Libero v11.8 software release.

1.3 Revision 4.0
Updated the document for Libero v11.7SP1 software release (SAR 81140).

1.4 Revision 3.0
Updated the document for Libero v11.7 software release (SAR 78192).

1.5 Revision 2.0
Updated the document for Libero v11.6 software release (SAR 71422).

1.6 Revision 1.0
Initial release.
This demo highlights the high-speed data transfer capability of the RTG4 devices through the PCIe interface. To achieve high-speed data transfer, an advanced extensible interface (AXI) based direct memory access (DMA) controller is implemented in the FPGA fabric. An application, PCIe_Data_Plane_Demo that runs in the host PC is provided for setting up and initiating the DMA transactions from the RTG4 PCIe endpoint to the host PC device. Drivers for connecting the host PC to the RTG4 PCIe endpoint are provided as part of the demo deliverables.

The high-speed serial interface (SERDESIF) available in the RTG4 devices provides a fully hardened PCIe endpoint implementation, and is compliant with the PCIe Base Specification Revision 2.0, 1.1 and 1.0. For more information on this, see the UG0567: RTG4 FPGA High-Speed Serial Interface User Guide.

This demo demonstrates the performance of the PCIe and DDR controller of the RTG4 devices.

### 2.1 Design Requirements

The following table lists the hardware and software requirements of the demo design.

<table>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td></td>
</tr>
<tr>
<td>RTG4 Development Kit</td>
<td>Rev B or later</td>
</tr>
<tr>
<td>Host PC with 8 GB RAM and PCIe 2.0 Gen1 compliant slot with x4 or higher width.</td>
<td>64-bit Windows 7 and 10</td>
</tr>
<tr>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>Libero® SoC</td>
<td>v12.0</td>
</tr>
<tr>
<td>FlashPro Express</td>
<td>v12.0</td>
</tr>
<tr>
<td>PCIe_Data_Plane_Demo Application</td>
<td>--</td>
</tr>
</tbody>
</table>

### 2.2 Demo Design

The demo design files are available for download at: [http://soc.microsemi.com/download/rsc/?f=rt4g_dg0622_liberosocv12p0_df](http://soc.microsemi.com/download/rsc/?f=rt4g_dg0622_liberosocv12p0_df)

Figure 1, page 3 shows the demo design. The PCIe core in the RTG4 devices supports both AXI and AMBA® high-performance bus (AHB) master and slave interfaces. This demo design uses the AXI master and slave interfaces to achieve maximum bandwidth. The PCIe_Data_Plane_Demo application on the host PC initiates the DMA transfers, and the embedded PCIe core in the RTG4 device initiates the AXI transactions through the AXI master interface to the DMA controller in the FPGA fabric. The DMA controller has two independent channels that share the AXI read/write channels of the PCIe AXI slave interface and FDDR AXI slave interface. The DMA controller in the FPGA fabric initiates the DMA channels depending on the type of the DMA transfer. Each channel has a timer to calculate the throughput. It has 4 KB of LSRAM buffer.

DMA channel 0 handles the following DMA transfers:
- Host PC memory to LSRAM
- Host PC memory to DDR memory
- LSRAM to DDR memory
DMA channel 1 handles the following DMA transfers:

- LSRAM to host PC memory
- DDR memory to host PC memory
- DDR memory to LSRAM

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

The FDDR controller is configured to access the DDR3 memory in x32 mode. The FDDR clock is configured to 320 MHz (640 Mbps DDR) with a 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.

### 2.2.1 Demo Design Features

The following are the demo design features:

- DMA data transfers between the host PC memory and the LSRAM
- DMA data transfers between the host PC memory and the DDR memory
- DMA data transfers between the DDR memory and the LSRAM
- Displays throughput for each DMA data transfer
- Enables continuous DMA transfers for observing throughput variations.
- Displays the PCIe link enable/disable, negotiated link width, and link speed on the PCIe_Data_Plane_Demo application.
- Displays the position of DIP Switches on the RTG4 Development Kit on the PCIe_Data_Plane_Demo application.
- Displays the PCIe Configuration Space on the PCIe_Data_Plane_Demo application.
- Controls LEDs on the board according to the command from the PCIe_Data_Plane_Demo application.
- Enables read and write operations to scratchpad register 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.
2.2.2  Demo Design Description

The demo design supports six types of data transfers. The following sections describe the process of each data transfer:

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

2.2.2.1 Host PC Memory to LSRAM (Read)

Data transfer from PC memory to the LSRAM block occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. Fabric DMA controller initiates a 16 beat AXI burst (128 bytes) read transaction to the PCIe AXI slave interface.
3. The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the host PC.
4. The host PC returns a completion (CplD) TLP to the PCIe link.
5. This returned data completes the AXI read initiated by the Fabric DMA controller.
6. This data is stored in the LSRAM.
7. The Fabric DMA controller repeats this process (from step 2 to 6) until the 4 KB size of data transfer is completed.
8. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application.

2.2.2.2 LSRAM to Host PC Memory (Write)

Data transfer from the LSRAM to PC memory occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. Fabric DMA controller reads the LSRAM data and initiates an AXI 16 beat burst write transaction to PCIe AXI slave interface.
3. The PCIe core sends a memory write (MWr) TLP to the host PC.
4. The Fabric DMA controller repeats this process (steps 2 and 3) until the 4 KB size of data transfer is completed.
5. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application.

2.2.2.3 Host PC Memory to DDR Memory (Read)

Data transfer from the PC memory to the DDR memory occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. Fabric DMA controller initiates a 16 beat AXI burst (128 bytes) read transaction to the PCIe AXI interface.
3. The PCIe core sends a MRd TLP to the host PC.
4. The host PC returns a CplD TLP to the PCIe link.
5. This returned data completes the AXI read initiated by the Fabric DMA controller.
6. This data is stored in the dual port LSRAM.
7. The LSRAM data is written to the DDR controller through the AXI interface as an AXI 16 beat burst write transaction. The reads from the host PC memory and the writes to the DDR memory occur independent of each other for achieving high throughput. Empty flags are generated in the Fabric DMA controller to avoid reading unknown data from the LSRAM.
8. The Fabric DMA controller repeats this process (from step 2 to 7) until the 4 KB size of data transfer is completed.
9. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application.
2.2.2.4 DDR Memory to Host PC Memory (Write)
Data transfer from the DDR memory to the PC memory occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. The Fabric DMA controller initiates a 16 beat burst (128 bytes) AXI read transaction from the DDR through the FDDR controller.
3. The data is stored in the dual port LSRAM.
4. The LSRAM data is written to the PCIe core as an AXI 16 beat burst write transaction. The reads from the DDR memory and writes to host PC memory occur independent of each other for achieving high throughput. Empty flags are generated in the Fabric DMA controller to avoid reading unknown data from the LSRAM.
5. The PCIe core sends a MWr TLP to the host PC.
6. The Fabric DMA controller repeats this process (from step 2 to 5) until the 4 KB size of data transfer is completed.
7. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application.

2.2.2.5 LSRAM to DDR Memory (Write)
Data transfer from the LSRAM to the DDR memory occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. The LSRAM data is written to the DDR controller through AXI interface as an AXI 16 beat burst write transaction.
3. The Fabric DMA controller repeats this process (step 2) until the 4 KB size of data transfer is completed.
4. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application for display.

2.2.2.6 DDR Memory to LSRAM (Read)
Data transfer from the DDR memory to the LSRAM occurs in the following sequence:

1. PCIe_Data_Plane_Demo application sets up the Fabric DMA controller through the PCIe link. This includes DMA direction, address, and size (4 KB).
2. The Fabric DMA controller initiates a 16 beat burst AXI read transaction of the DDR through the FDDR controller.
3. The data is stored in the dual port LSRAM.
4. The Fabric DMA controller repeats this process (steps 2 and 3) until the 4 KB size of data transfer is completed.
5. The Fabric DMA controller provides the DMA completion status and the number of clock cycles consumed to complete the DMA transaction to the PCIe_Data_Plane_Demo application for display.

2.2.3 Throughput Calculation
This demo implements a timer to measure the throughput of DMA transfers. The throughput measured includes the entire AXI overhead, PCIe, and DMA controller transactions.

The demo design implements following steps to measure throughput:

1. Setup the DMA controller for the complete transfer.
2. Start the timer and the DMA controller.
3. Initiate the data transfer for the requested number of bytes.
4. Wait until the DMA transfer is completed.
5. Record the number of clock cycles used for steps 2 to 4.

To arrive at a realistic system performance, the throughput calculation takes into account all the overheads during a transfer.

\[
\text{Throughput} = \frac{\text{Transfer Size (Byte)}}{\text{Number of clock cycles taken for a transfer}} \times \text{Clock period}
\]
2.3 Setting Up the Demo Design

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

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

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Pin From</th>
<th>Pin To</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J11, J17, J19, J23, J26, J21, J32, J27</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J16</td>
<td>2</td>
<td>3</td>
<td>Default</td>
</tr>
<tr>
<td>J33</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
</tbody>
</table>

Note: Ensure that the power supply switch, SW6 is switched OFF 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 other end of the cable to the USB port of the host PC.
4. Switch ON the power supply switch, SW6.

2.3.1 Programming the Device Using FlashPro Express

This section describes how to program the RTG4 device with the .job programming file using FlashPro Express. The .job file is available at the following design files folder location:

```
rt4g_dg0622_liberosocv12p0_df\ProgrammingJob
```

To program the device, complete the following steps:

1. On the host PC, launch the FlashPro Express software.
2. 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.
3. 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>\rt4g_dg0622_liberosocv12p0_df\ProgrammingJob`.
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

   ![New Job Project from FlashPro Express Job](image)

4. Click **OK**. The required programming file is selected and ready to be programmed in the device.

5. 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**.

   ![Programming the Device](image)
6. Click **RUN** to program the device. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure.

**Figure 5** • FlashPro Express—RUN PASSED

![FlashPro Express—RUN PASSED](image)

7. Close **FlashPro Express (Project > Exit)**.
2.3.2 Connecting RTG4 Development Kit to Host PC PCIe Slot

The following steps describe how to connect the RTG4 Development Kit board 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 host PC's PCIe slot through the PCI Edge Card Ribbon cable.

**Note:** Ensure that the host PC is switched OFF while inserting the PCIe Edge connector. Else, the PCIe device may not be detected properly. The host PC may wake-up after inserting the PCIe Edge connector, as the RTG4 devices do not support cold sparing.

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 6 • RTG4 Development Kit Setup*
3. Switch ON the host PC and check the **Device Manager of the Host PC for PCIe Device**. The following figure shows the **Device Manager** window. If the device is not detected, power cycle the RTG4 Development Kit and click **scan for hardware changes** (highlighted in the figure) in the **Device Manager** window.

**Figure 7 • Device Manager - PCIe Device Detection**

![Device Manager](image)

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

### 2.3.3 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**... See the following figure.

**Figure 8 • Update Driver Software**

![Update Driver Software](image)
2. In the **Update Driver Software - PCIe Device** window, select the **Browse my computer for driver software** option. See the following figure.

*Figure 9 • Browse for Driver Software*

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

3. Browse the drivers folder: `<download files>/PCIe Drivers/Win_64bit_PCIe_Drivers` and click **Next**. See the following figure.

*Figure 10 • Browse for Driver Software Continued*

![Browse for Driver Software Continued](image2.png)

4. **Windows Security** dialog box is displayed and click **Install**. See the following figure.

After successful driver installation, a message window appears. See *Figure 12, page 12.*

*Figure 11 • Windows Security*

![Windows Security](image3.png)
2.3.4 Installing 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 PCIe link status, driver information and 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>\rt4g_dg0622_liberosocv12p0_df\GUI\PCIe Data Plane Demo Installer V1.0`, Double-click `setup.exe`. Do not change the default options and click **Next**.

2. Click **Next** to start the installation.
3. Click **Finish** to complete the installation. The following figure shows that the installation has been successfully completed.

**Figure 15** • **Successful Installation of PCIe_Data_Plane_Demo Application**

4. Shut down the host PC.
5. Power cycle the RTG4 Development Kit.
6. Restart the host PC.
2.4 Running the Design

The following steps describe how to run the demo design:

1. Check the host PC Device Manager for the drivers. If the device is not detected, power cycle the RTG4 Development Kit and click scan for hardware changes (highlighted in the following figure) in the Device Manager window.

The following figure shows an example Device Manager window.

*Figure 16 • Device Manager - PCIe Device Detection*

Note: If a warning appears on the DEVICE or WinDriver in the Device Manager, uninstall them and start from running the design of driver installation.
2. Invoke the 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.

The following figure shows the example messages after the connection is established.

*Figure 17 • PCIe Device Information*
4. Click the **Demo Controls** tab to display the LEDs options, DIP switch positions (SW5 – DIP1, DIP2, DIP3, and DIP4), and the interrupt counters. Controlling LEDs, getting the DIP switch status, and monitoring the interrupts can be done simultaneously. See the following figure.

**Figure 18 • Demo Controls**

![Demo Controls Figure]

**Note:** Interrupt Counter1 and Counter2 are allocated for SW1 and SW2 events.

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

**Figure 19 • PCIe Configuration Space**

![PCIe Configuration Space Figure]
6. Click the **PCIe Read/Write** tab to perform read and writes to 4 KB LSRAM using BAR1 space. Click **Read** to read the 4 KB memory mapped to BAR1 space. The following figure shows the PCIe Read/Write panel. Double-click any bar location to write the memory mapped to BAR1 space.

**Figure 20** • Read and Writes to Scratchpad Register

7. Click the **DMA Operations** tab.
The following instructions describe running DMA operations between PC and LSRAM, PC and DDR, DDR and LSRAM:

1. Select one of the following options from the DMA Transfer Type Selection drop-down list:
   - **PC to LSRAM**—to transfer the data from host PC to LSRAM memory
   - **LSRAM to PC**—to transfer the data from LSRAM memory to host PC
   - **Both PC & LSRAM**—to transfer the data from host PC to and from LSRAM memory
   - **PC to DDR**—to transfer the data from host PC to DDR memory
   - **DDR to PC**—to transfer the data from DDR memory to host PC
   - **Both PC & DDR**—to transfer the data from host PC to and from DDR memory
   - **DDR to LSRAM**—to transfer the data from DDR memory to LSRAM memory
   - **LSRAM to DDR**—to transfer the data from LSRAM memory to DDR memory
   - **Both DDR & LSRAM**—to transfer the data from DDR memory to and from LSRAM memory

2. Select **Transfer Size** (4KB to 1 MB) from the drop-down list.

3. Enter the **Loop Count** in the box.

4. Click **Start Transfer**. After a successful DMA operation, the GUI displays the throughput and average throughput in MBps.

The following figure shows the throughput and average throughput in MBps.

**Figure 21** • Continuous DMA Operations with DMA Transfer Type Selection as PC to LSRAM

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

### 2.5 Summary

This demo shows how to implement a PCIe Data Plane Design using AXI based fabric DMA controller. The throughput for data transfers depends on the Host PC system configuration and the type of PCIe slots used.
The following table lists the throughput values observed on the HP Workstation Z220 PCIe slot 4.

<table>
<thead>
<tr>
<th>DMA Transfer Type</th>
<th>Throughput Summary</th>
<th>Throughput in Mbyte/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X1 Lane</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gen1</td>
</tr>
<tr>
<td>Host PC to LSRAM</td>
<td>Read</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>Write</td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>R/W</td>
<td>180/237</td>
</tr>
<tr>
<td>DDR to LSRAM</td>
<td>Read</td>
<td>577</td>
</tr>
<tr>
<td></td>
<td>Write</td>
<td>585</td>
</tr>
<tr>
<td></td>
<td>R/W</td>
<td>575/585</td>
</tr>
<tr>
<td>Host PC to DDR</td>
<td>Read</td>
<td>178</td>
</tr>
<tr>
<td></td>
<td>Write</td>
<td>232</td>
</tr>
<tr>
<td></td>
<td>R/W</td>
<td>177/232</td>
</tr>
</tbody>
</table>

The table shows the throughput values in Mbyte/sec for different DMA transfer types, including X1 Lane and Gen1, with specific values for single and loop transfers.
The following steps describe the known issues:

1. The PCIe reset (fundamental reset or in-band reset) causes the endpoint device state machines, hardware logic, port states, and configuration registers (except for the sticky registers) to initialize the default conditions. During a host initiated PCIe reset process, the SERDES PCIe endpoint reset must be generated in a proper sequence, and the endpoint device must be reinitialized correctly.

2. If the PCIe endpoint is not reset properly, corrupt data may be passed through the PCIe link. The demo design implements proper endpoint reset, as shown in Figure 22, page 20.

3. It detects the hot reset, and resets the SERDES core and AXI interface using the SERDES soft reset register through the CoreABC IP module. The logic performs the following operations:
   a. The HOTRESET_DETECT logic detects the hot reset from the root port by monitoring the LTSSM[5] signal from SERDES block and generates HOTRESET.
   b. The GPIO_IN[0] of the CoreGPIO module is connected to the HOTRESET signal. The GPIO_IN[1] is connected to the PCIE_L2P2_ACTIVE signal from the SERDES block.
   c. The CoreGPIO generates an interrupt to the CoreABC IP module on the positive edge of the HOTRESET signal or the negative edge of the PCIE_L2P2_ACTIVE signal.
   d. The CoreGPIO interrupt is connected to IO_IN[0] of the CoreABC IP module, the CoreABC monitors IO_IN[0]. If the CoreGPIO interrupt is high, CoreABC resets the SERDES core and AXI interface using the SERDES soft reset register.

*Figure 22 • Hot Reset Detection Block*
The following table lists shows the registers used to interface with the Fabric DMA controller. These registers are in BAR1 address space.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC_BASE_ADDR</td>
<td>0x8028</td>
<td>Host PC memory base address provided by the driver.</td>
</tr>
<tr>
<td>DMA_DIR</td>
<td>0x8008</td>
<td>DMA direction:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Direction – Register Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. PCIe → DDR memory – 0x11AA0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. DDR → PCIe memory – 0x11AA0002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. LSRAM → DDR memory – 0x11AA0003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. DDR → LSRAM memory – 0x11AA0004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. PCIe → LSRAM memory – 0x11AA0005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. LSRAM → PCIe memory – 0x11AA0006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To reset the DMA, the register value is 0x11AA0007.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Before initiating DMA transactions, reset the DMA with the register value, 0x11AA0007.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. The DMA transactions 1 and 2, 3 and 4, or 5 and 6 can be performed simultaneously by writing the corresponding values one after another.</td>
</tr>
<tr>
<td>DMA_CH0_STATUS</td>
<td>0x8100</td>
<td>DMA Channel-0 status:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DMA_CH0_STATUS[31]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: DMA operation completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: DMA operation not completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DMA_CH0_STATUS[15:0] = CLK count</td>
</tr>
<tr>
<td>DMA_CH1_STATUS</td>
<td>0x8108</td>
<td>DMA Channel-1 status:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DMA_CH1_STATUS[31]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: DMA operation completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: DMA operation not completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DMA_CH1_STATUS[15:0] = CLK count</td>
</tr>
<tr>
<td>RW_REG</td>
<td>0x0</td>
<td>Scratchpad register for PCIe R/W.</td>
</tr>
<tr>
<td>LED_CTRL</td>
<td>0xA0</td>
<td>LEDs control register.</td>
</tr>
<tr>
<td>SWITCH_STATUS</td>
<td>0x90</td>
<td>DIP switch status.</td>
</tr>
<tr>
<td>CLK_FREQ</td>
<td>0x8038</td>
<td>DMA controller clock frequency.</td>
</tr>
<tr>
<td>BAR1 memory</td>
<td>0x9000 – 0x9FFF</td>
<td>Memory connected to BAR1.</td>
</tr>
</tbody>
</table>

**Note:** For the DDR memory, the source memory address is fixed as 0x0100_0000 and the destination memory address is fixed as 0x0000_0000.