DG0713
Demo Guide
RTG4 FPGA PCIe Data Plane Demo using Scatter-Gather DMA Controller - Libero SoC v12.0
## Contents

1 Revision History ................................................................. 1  
1.1 Revision 3.0 ........................................................................... 1  
1.2 Revision 2.0 ........................................................................... 1  
1.3 Revision 1.0 ........................................................................... 1  

2 PCIe Data Plane Demo using Scatter-Gather DMA Controller ............. 2  
2.1 Design Requirements ................................................................ 2  
2.2 Demo Design ........................................................................... 2  
.... 2.2.1 Features ......................................................................... 3  
.... 2.2.2 Description ..................................................................... 4  
.... 2.2.3 Throughput Calculation .................................................... 7  
2.3 Setting up the Demo Design ....................................................... 7  
.... 2.3.1 Setting Up the Board .......................................................... 7  
.... 2.3.2 Programming the Device Using FlashPro Express ................. 8  
.... 2.3.3 Connecting RTG4 Development Kit to Host PC PCIe Slot ........ 10  
.... 2.3.4 Drivers Installation ............................................................ 11  
.... 2.3.5 Installing the PCie_Data_Plane_Demo Application GUI .......... 13  
2.4 Running the Design ............................................................... 15  
.... 2.4.1 Summary ....................................................................... 22  

3 Appendix: Demo Registers and Buffer Descriptor Details ................... 23  
3.1 SGDMA Registers (BAR1) ....................................................... 23  
3.2 Buffer Descriptor (16 bytes) .................................................... 24
Figures

Figure 1  PCIe Data Plane Demo- Block Diagram .................................................................................. 3
Figure 2  Data Transfer from Host PC Memory to RTG4 DDR Memory ........................................... 5
Figure 3  Data Transfer from RTG4 DDR Memory to Host PC Memory ............................................. 6
Figure 4  FlashPro Express Job Project ............................................................................................. 8
Figure 5  New Job Project from FlashPro Express Job ...................................................................... 8
Figure 6  Programming the Device ..................................................................................................... 9
Figure 7  FlashPro Express—RUN PASSED .................................................................................... 9
Figure 8  RTG4 Development Kit Setup ........................................................................................... 10
Figure 9  Device Manager—PCIe Device Detection .......................................................................... 11
Figure 10 Update Driver Software ..................................................................................................... 11
Figure 11 Browse for Driver Software ............................................................................................... 12
Figure 12 Browse for Driver Software Continued ............................................................................. 12
Figure 13 Windows Security .............................................................................................................. 12
Figure 14 Successful Driver Installation ............................................................................................ 13
Figure 15 PCIe Demo Application Installation ................................................................................... 13
Figure 16 PCIe Demo Application Installation ................................................................................... 14
Figure 17 Successful Installation of PCIe Demo Application ............................................................... 14
Figure 18 Device Manager—PCIe Device Detection .......................................................................... 15
Figure 19 Device Info ........................................................................................................................... 16
Figure 20 Demo Controls Continued .................................................................................................. 17
Figure 21 Configuration Space ............................................................................................................ 18
Figure 22 PCIe BAR1 memory access ................................................................................................. 19
Figure 23 PC Memory to DDR DMA .................................................................................................. 20
Figure 24 DDR to PC Memory DMA ................................................................................................... 21
Figure 25 Both DMA Operations ........................................................................................................ 22
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</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>Jumper Pin Connections</td>
<td>7</td>
</tr>
<tr>
<td>Table 3</td>
<td>Throughput Values</td>
<td>22</td>
</tr>
<tr>
<td>Table 4</td>
<td>SGDMA Registers</td>
<td>23</td>
</tr>
<tr>
<td>Table 5</td>
<td>Buffer Descriptor and Bit Descriptions</td>
<td>24</td>
</tr>
</tbody>
</table>
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 3.0
Updated the document for Libero SoC v12.0 software release.

1.2 Revision 2.0
Updated the document for Libero v11.8 software release.

1.3 Revision 1.0
Revision 1.0 was the first publication of this document.
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 SGM DMA controller performs a bulk-data transfer between non-contiguous memory locations defined by buffer descriptors. The SGM DMA 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 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, see the 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>
<thead>
<tr>
<th>Table 1 • Design Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Requirements</strong></td>
</tr>
<tr>
<td>RTG4 Development Kit</td>
</tr>
<tr>
<td>Host PC with 8 GB RAM and PCIe 2.0 Gen1 compliant slot with x4 or higher width</td>
</tr>
<tr>
<td><strong>Software Requirements</strong></td>
</tr>
<tr>
<td>Libero® SoC</td>
</tr>
<tr>
<td>FlashPro Express</td>
</tr>
<tr>
<td>Host PC Drivers (provided along with the design files)</td>
</tr>
<tr>
<td>PCIe Demo application (provided along with the design files)</td>
</tr>
</tbody>
</table>

2.2 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 liberosocv12p0_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 SGM DMA 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, see Buffer Descriptor (16 bytes), page 24.

The SGM DMA 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 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 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 the 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.
2.2.2 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 and the DMA start command to SGMIA controller in the FPGA fabric.

To perform the DMA transfers the SGMIA controller fetches the 8 buffer descriptors from the base address of the first buffer descriptor. The SGMIA 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

1. Start
2. Host PC initiates SGDMA controller read channel
3. DMA controller fetches the 8 buffer descriptors from Host PC
4. Are 8 BD operations completed?
   - Yes
      - DMA Controller decodes the next buffer descriptor
      - DMA controller initiates AXI burst read operations to PCIe core AXI slave IF
      - The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the link partner. The link partner returns a completion (CplD) TLP to the PCIe link
      - With this read data, DMA controller initiates the AXI burst write operations to DDR controller
   - No
      - DMA Controller decodes the next buffer descriptor
      - DMA controller initiates AXI burst read operations to PCIe core AXI slave IF
      - The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the link partner. The link partner returns a completion (CplD) TLP to the PCIe link
      - With this read data, DMA controller initiates the AXI burst write operations to DDR controller
   - Memory read size = BD DMA size
     - Yes
     - End of SGDMA operation?
       - Yes
         - End
     - No
6. No
Figure 3 • Data Transfer from RTG4 DDR Memory to Host PC Memory

Start

Host PC initiates SGDMA controller write channel

DMA controller fetches the 8 buffer descriptors from Host PC

Are 8 BD operations completed?

Yes

DMA Controller decodes the next buffer descriptor

DMA controller grabs data from DDR3 memory and initiates the AXI burst write operations to PCIe core AXI slave IF

The PCIe sends memory write (MWr) TLPS across the PCIe link

With this read data, DMA controller initiates the AXI burst write operations to DDR controller

Memory write size = BD DMA size

Yes

No

Is It the End of SGDMA operation?

Yes

No

End
2.2.3 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 host PC and decodes them.
2. Setup 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)

2.3 Setting up the Demo Design

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

- Setting Up the Board
- Programming the Demo Design
- Connecting RTG4 Development Kit to Host PC PCIe Slot
- Drivers Installation
- Installing the PCIe_Data_Plane_Demo Application GUI

2.3.1 Setting Up the Board

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 2 • Jumper Pin Connections

<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></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

   **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 other end of the cable to the USB port of the host PC.
4. Switch ON the power supply switch, SW6.
2.3.2 Programming the Device Using FlashPro Express

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

```
rtg4_dg0713_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.

![FlashPro Express Job Project](image1)

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>\rtg4_dg0713_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](image2)

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.

**Figure 6 • Programming the Device**

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 7 • FlashPro Express—RUN PASSED**

7. Close **FlashPro Express (Project > Exit)**.
2.3.3 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 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 8 • 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 PCI device.
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.

2.3.4 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**. See the following figure.

*Figure 10 • Update Driver Software*
2. In the Update Driver Software - PCIe Device window, select the Browse my computer for driver software option. See the following figure.

*Figure 11 • Browse for Driver Software*

3. Browse the drivers folder: <download files>/PCIe Drivers/Win_64bit_PCIE_Drivers and click Next. See the following figure.

*Figure 12 • Browse for Driver Software Continued*

4. Windows Security dialog box is displayed and click Install. See the following figure. After successful driver installation, a message window appears. See Figure 14, page 13.

*Figure 13 • Windows Security*
2.3.5 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_liberosocv12p0_dflGUI_PCIe Data Plane Demo Installer V1.0`, and double-click setup.exe, do not change the default options as shown in the following figure.

2. Click **Next** to start the installation. See **Figure 16**, page 14.
Three steps are described:

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

4. The installation completes successfully. See the following figure.

**Figure 17 • Successful Installation of PCIe Demo Application**

5. Restart the host PC.
2.4 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. See the following figure.

*Figure 18* • 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. See the following figure.

*Figure 19 • Device Info*
4. Click the **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. See the following figure.

*Figure 20 • Demo Controls Continued*

![PCle Data Plane Demo](image)

*Note:* Interrupt Counter1 and Counter4 are allocated for SW1 and SW2 events. Interrupt Counter2 and Counter3 are allocated for SGDMA controller.
5. Click the **Config Space** tab to view the details about the PCIe configuration space. See the following figure.

*Figure 21 • Configuration Space*
6. Click the **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. See the following figure.

*Figure 22 • PCIe BAR1 memory access*
7. Click the **DMA Operations** 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 8 bytes to 1,048,576 bytes (1 MB). After successful DMA operation, the GUI displays the throughput and the number of buffer descriptors created by the driver. See following figure.

*Figure 23 • 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. See the following figure.

*Figure 24 • 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 host PC. See the following figure.

**Figure 25** • 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.

### 2.4.1 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

<table>
<thead>
<tr>
<th>DMA Transfer Type</th>
<th>Throughput (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host PC memory to DDR (read)</td>
<td>170 MB/s</td>
</tr>
<tr>
<td>DDR to Host PC memory (write)</td>
<td>220 MB/s</td>
</tr>
<tr>
<td>Host PC memory to DDR (read)/DDR to Host PC memory (write)</td>
<td>170/220 MB/s</td>
</tr>
</tbody>
</table>
3 Appendix: Demo Registers and Buffer Descriptor Details

3.1 SGDMA Registers (BAR1)

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

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

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

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

<table>
<thead>
<tr>
<th>Reserved[63:32]</th>
<th>Host PC Address[31:0]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Valid descriptor[31:24]</td>
</tr>
<tr>
<td></td>
<td>Last transfer[23]</td>
</tr>
<tr>
<td></td>
<td>Direction[22:21]</td>
</tr>
<tr>
<td></td>
<td>Size[20:0]</td>
</tr>
</tbody>
</table>

- 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