DG0622
Demo Guide
RTG4 FPGA PCIe Data Plane Demo using Two Channel Fabric DMA

Microsemi
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

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.

50200622. 9.0 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 9.0

Updated the document for Libero SoC v2021.2.

1.2 Revision 8.0

Replaced Figure 13, page 14.

1.3 Revision 7.0

The following is a summary of the changes made.

- Added Appendix 1: Programming the Device Using FlashPro Express, page 18.
- Added Appendix 2: Running the TCL Script, page 21.
- Removed the references to Libero version numbers.

1.4 Revision 6.0

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

1.5 Revision 5.0

Updated the document for Libero v11.8 software release.

1.6 Revision 4.0

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

1.7 Revision 3.0

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

1.8 Revision 2.0

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

1.9 Revision 1.0

Initial release.

2 RTG4 FPGA PCIe Data Plane Demo using Two Channel Fabric DMA

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, refer to *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 1 • **Design Requirements**

Requirement	Version
Hardware	
RTG4 Development Kit	Rev B 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)	**Note:** Refer to the `readme.txt` file provided in the design files for the software versions used with this reference design.
FlashPro Express	
PCIe_Data_Plane_Demo Application	–

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:

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 at:
http://soc.microsemi.com/download/rsc/?f=rtg4_dg0622_df

Figure 1, page 4 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.

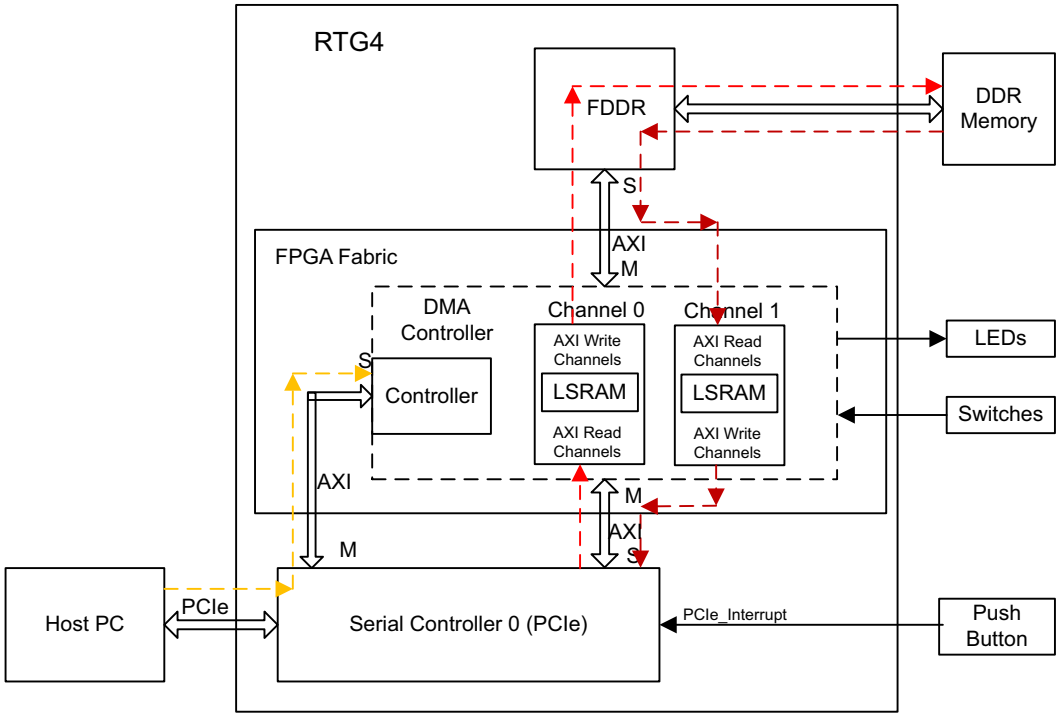
DMA channel 0 handles the following DMA transfers:

- 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**



Legend:
- Path from Host PC to the DMA controller
- DMA Channel 0 path
- DMA Channel 1 path

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 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 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.

2.3.2 Demo Design Description

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

2.3.2.1 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.

2.3.2.2 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.

2.3.2.3 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 an 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.

2.3.2.4 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 an 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.

2.3.2.5 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 the 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.

2.3.2.6 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.

2.3.3 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 the following steps to measure throughput:

1. Set up 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 realistic system performance, the throughput calculation takes into account all the overheads during a transfer.

$$\text{Throughput} = \text{Transfer Size (Byte)} / \text{Number of clock cycles taken for a transfer} \times \text{Clock period}$$

2.4 Setting Up the Demo Design

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 • **RTG4 Development Kit Jumper Settings**

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: 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 another end of the cable to the USB port of the host PC.
4. Switch ON the power supply switch, **SW6**.

2.4.1 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 18.

2.4.2 Connecting RTG4 Development Kit to Host PC PCIe Slot

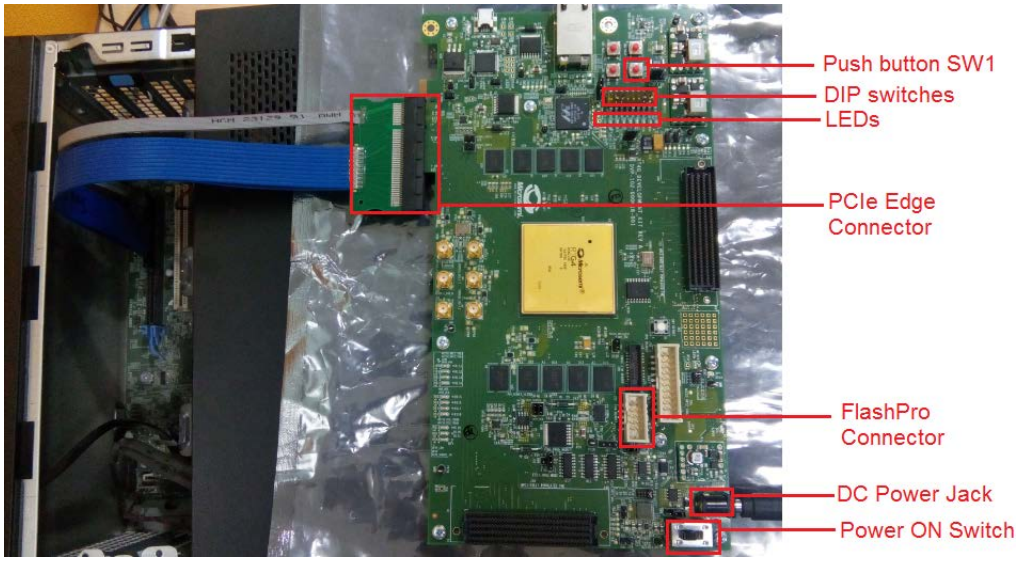
The following steps describe how to connect the RTG4 Development Kit board to the host PC:

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.

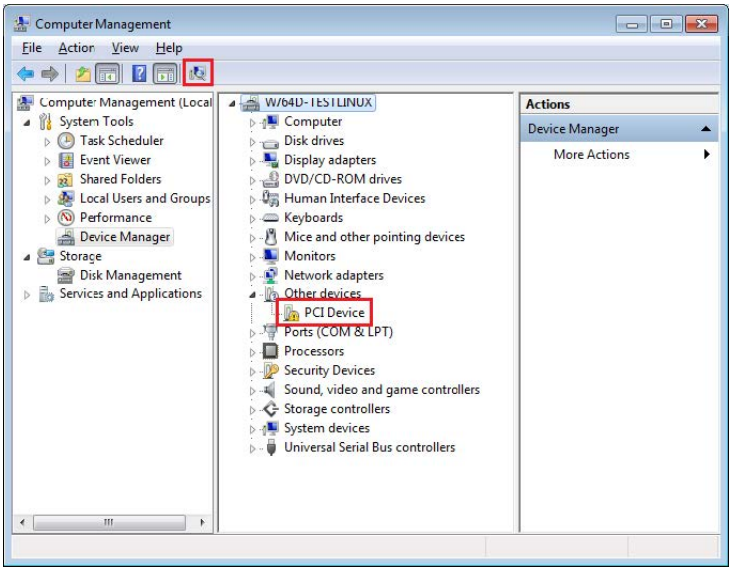
The following figure shows the board setup for the host PC in which the RTG4 Development Kit is connected to the host PC PCIe slot.

Figure 2 • **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 3 • **Device Manager - PCIe Device Detection**



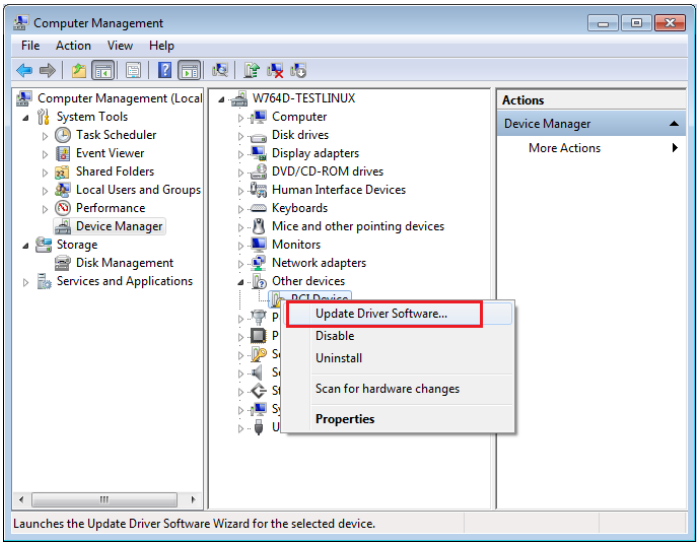
Note: If the device is still not detected, check if the BIOS version in the host PC is the latest and if PCI is enabled in the host PC BIOS.

2.4.3 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...** as shown in the following figure.

Figure 4 • **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.

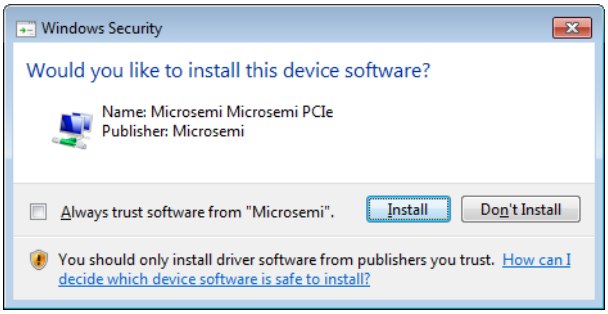
Figure 5 • **Browse for Driver Software**

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

Figure 6 • **Browse for Driver Software Continued**

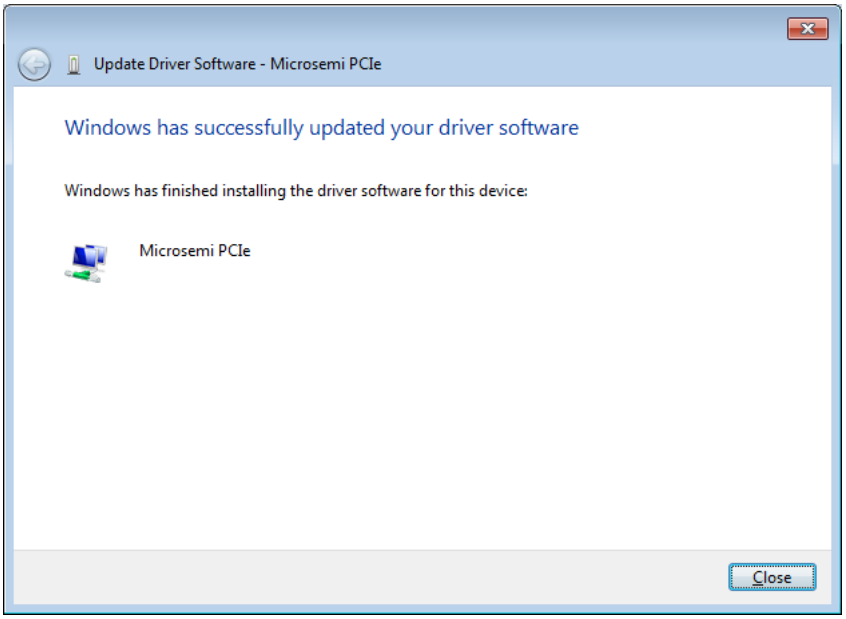
4. **Windows Security** dialog box is displayed and click **Install**, as shown in the following figure.

Figure 7 • **Windows Security**



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

Figure 8 • **Successful Driver Installation**



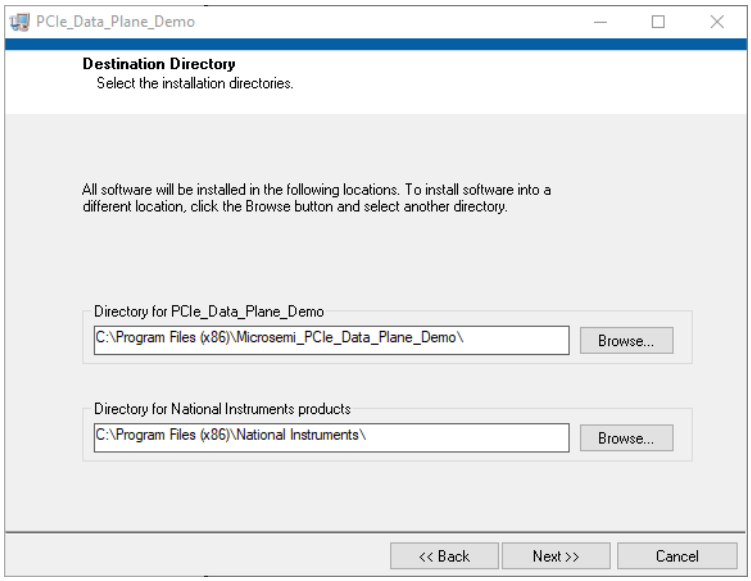
2.4.4 Installing PCIe_Data_Plane_Demo Application GUI

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:

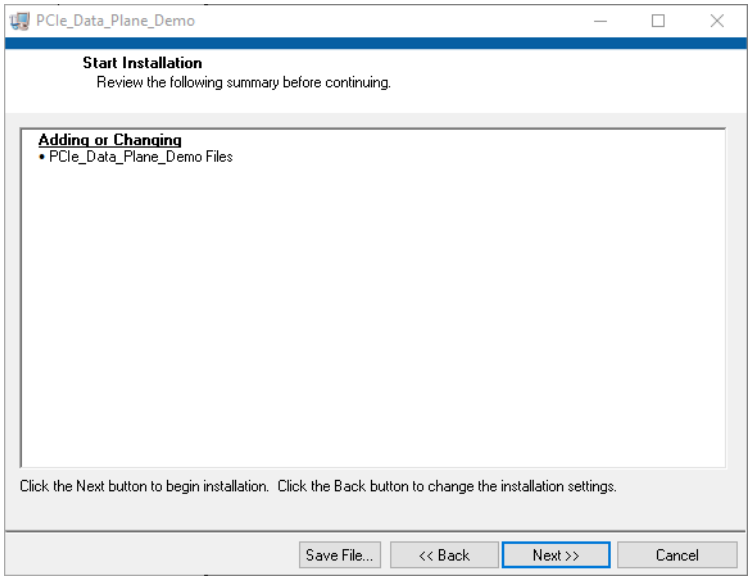
1. Go to *<Download Folder>\rtg4_dg0622_df\GUI\PCIe Data Plane Demo Installer V1.0*, Double-click `setup.exe`. Do not change the default options and click **Next**.

Figure 9 • **Installing PCIe_Data_Plane_Demo Application**



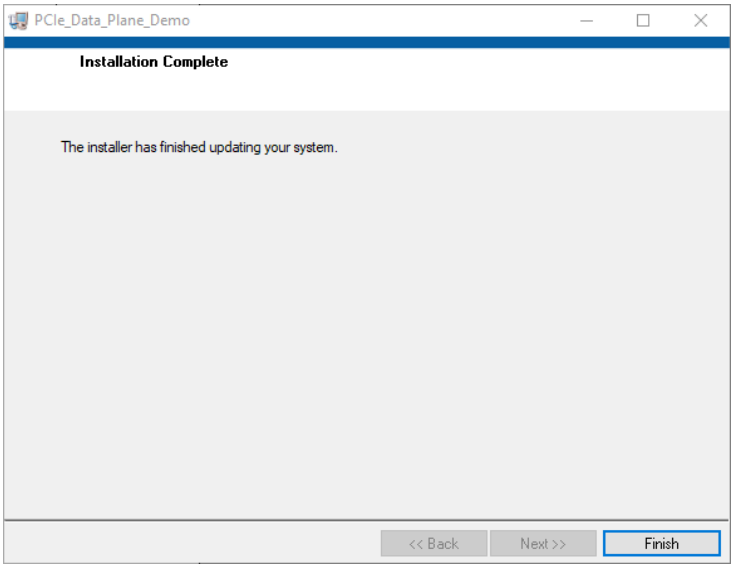
2. Click **Next** to start the installation.

Figure 10 • **PCIe_Data_Plane_Demo Application Installation**



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

Figure 11 • **Successful Installation of PCIe_Data_Plane_Demo Application**



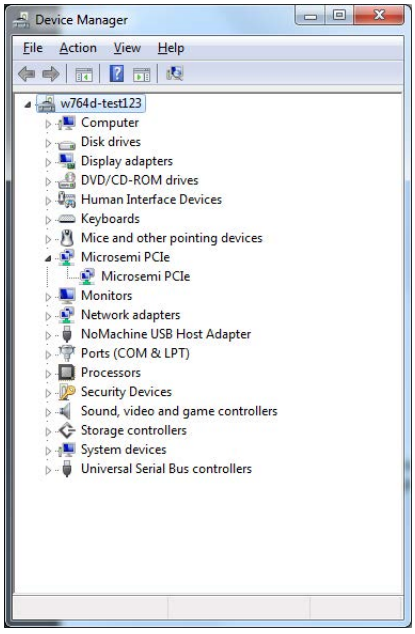
4. Shut down the host PC.
5. Power cycle the RTG4 Development Kit.
6. Restart the host PC.

2.5 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 12 • **Device Manager - PCIe Device Detection**

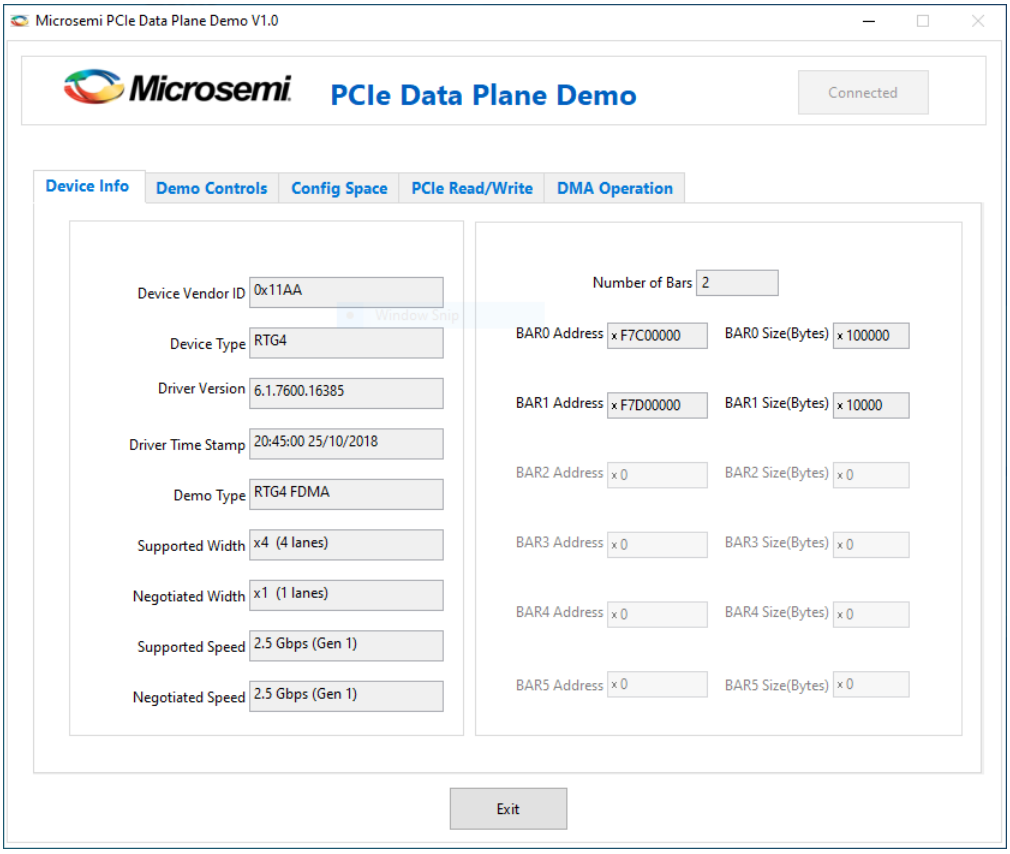


Note: If a warning appears on the DEVICE or WinDriver in the Device Manager, uninstall them and start from running the design of driver installation.

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 link speed.

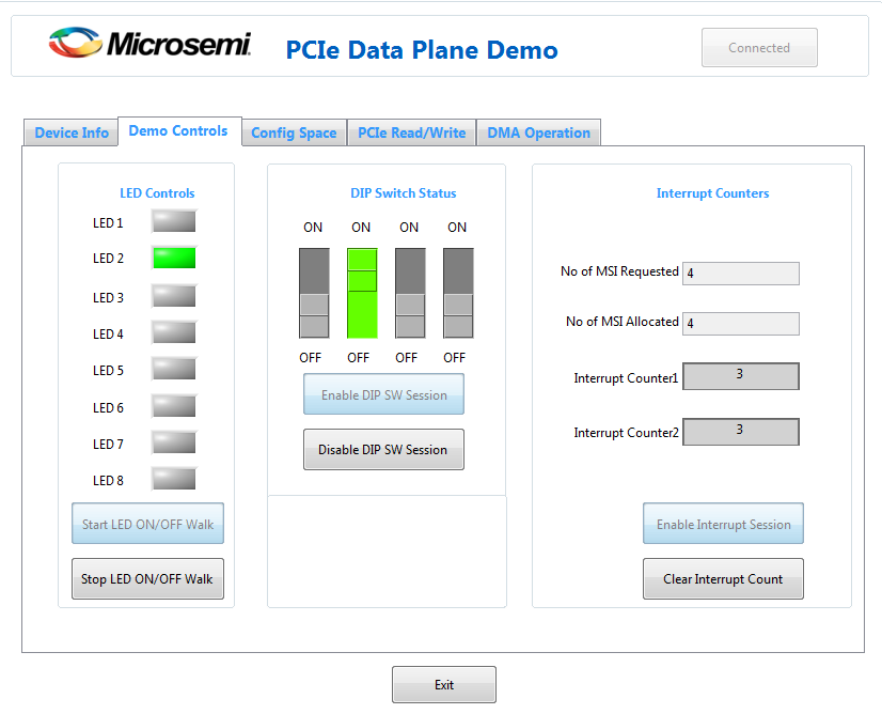
The following figure shows the example messages after the connection is established.

Figure 13 • **PCIe Device Information**



4. Click **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, as shown in the following figure.

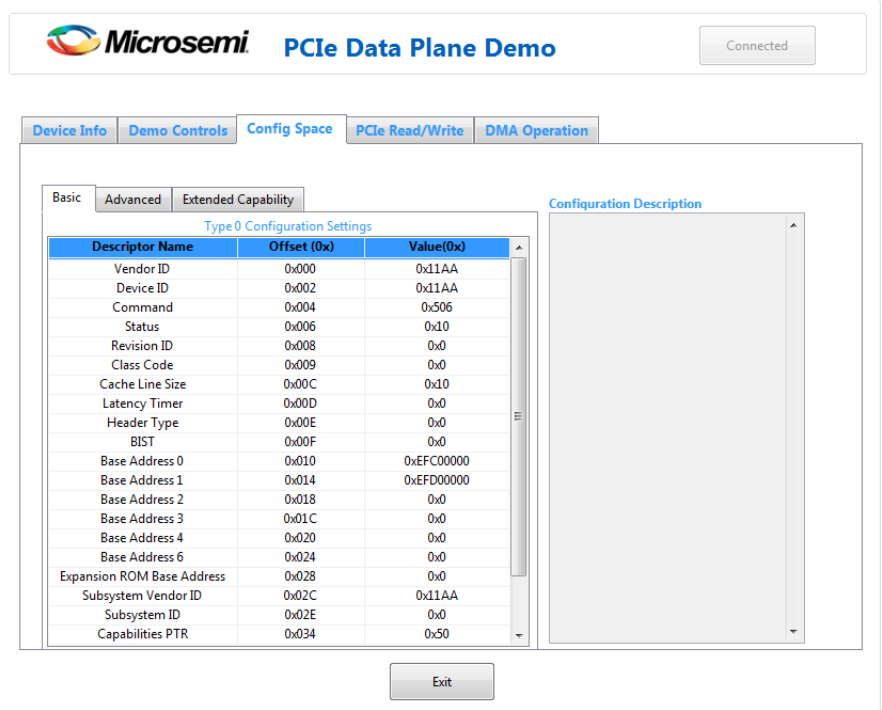
Figure 14 • **Demo Controls**



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

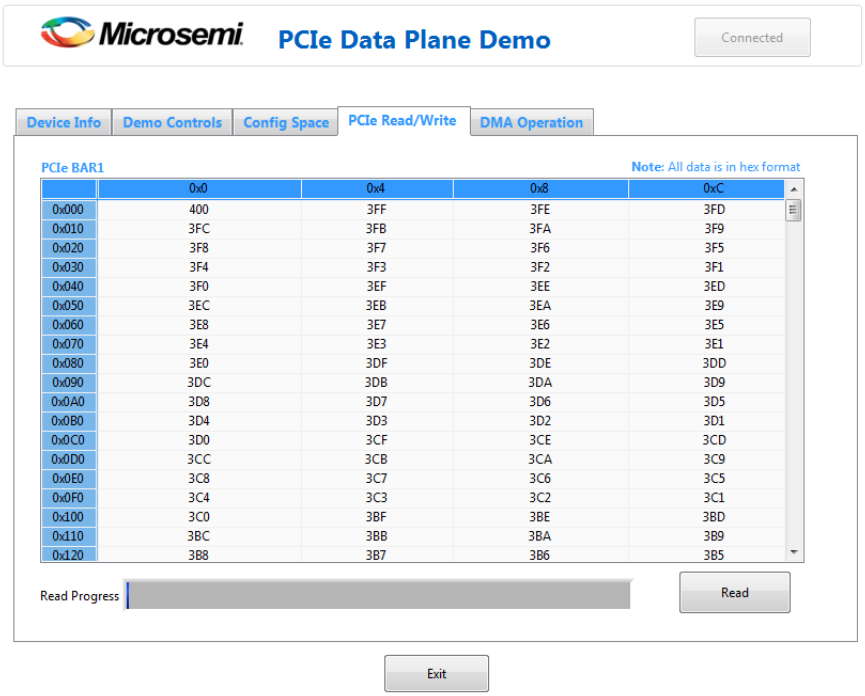
5. Click **Config Space** to view the details of the PCIe configuration space. The following figure shows the PCIe configuration space.

Figure 15 • **PCIe Configuration Space**



6. Click **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 16 • **Read and Writes to Scratchpad Register**



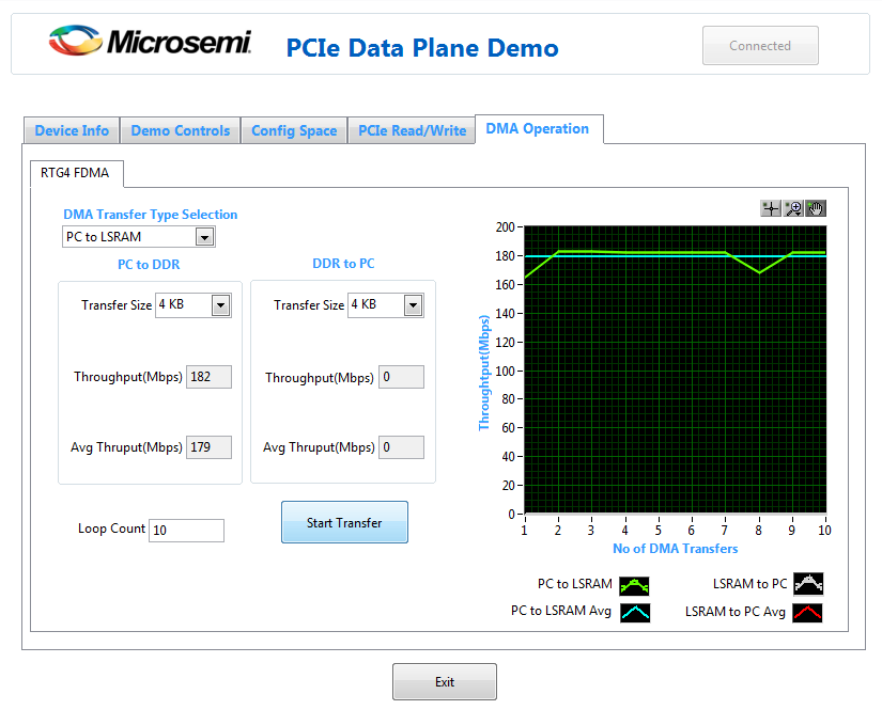
7. Click **DMA Operations** tab.

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



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

2.6 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 3 • **Throughput Summary**

		Throughput in Mbyte/sec	
		X1 Lane	
		Gen1	
DMA Transfer Type		**Single xfer**	**Loop xfer (50)**
Host PC to LSRAM	Read	180	177
	Write	237	236
	R/W	180/237	180/236
DDR to LSRAM	Read	577	571
	Write	585	582
	R/W	575/585	570/580
Host PC to DDR	Read	178	178
	Write	232	232
	R/W	177/232	177/231

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.

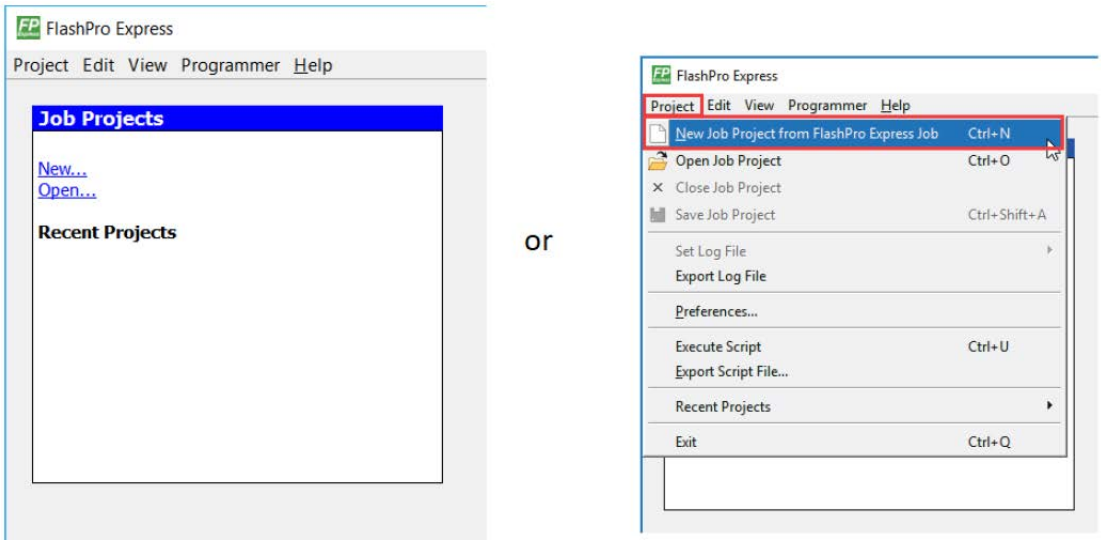
To program the device, perform the following steps:

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.

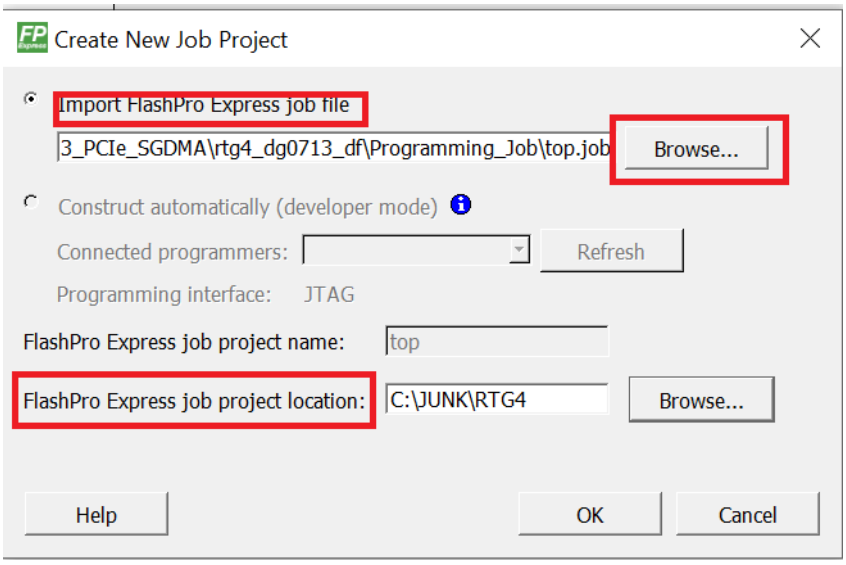
3. Connect the power supply cable to the **J9** connector on the board.
4. Power **ON** the power supply switch **SW6**.
5. If using the embedded FlashPro5, connect the USB cable to connector **J47** and the host PC. Alternatively, if using an external programmer, connect the ribbon cable to the JTAG 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 18 • **FlashPro Express Job Project**



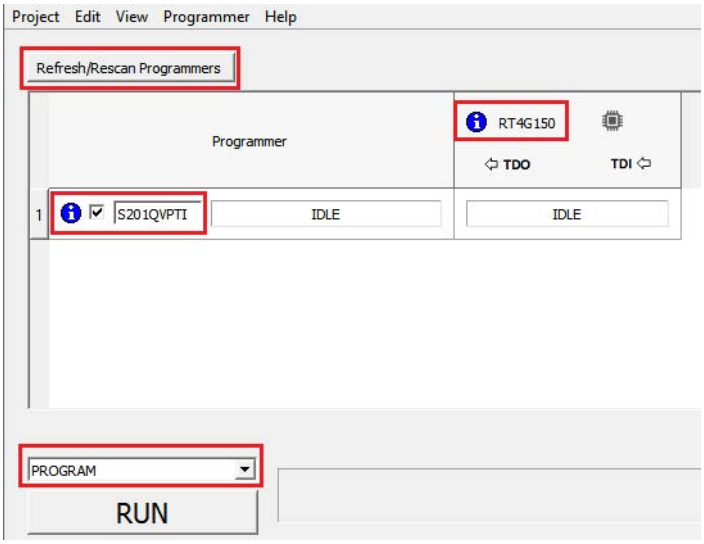
8. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
   - **Programming job file**: Click **Browse**, and navigate to the location where the .job file is located and select the file. The default location is:
     `<download_folder>\rtg4_dg0622_df\Programming_Job`
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

Figure 19 • **New Job Project from FlashPro Express Job**



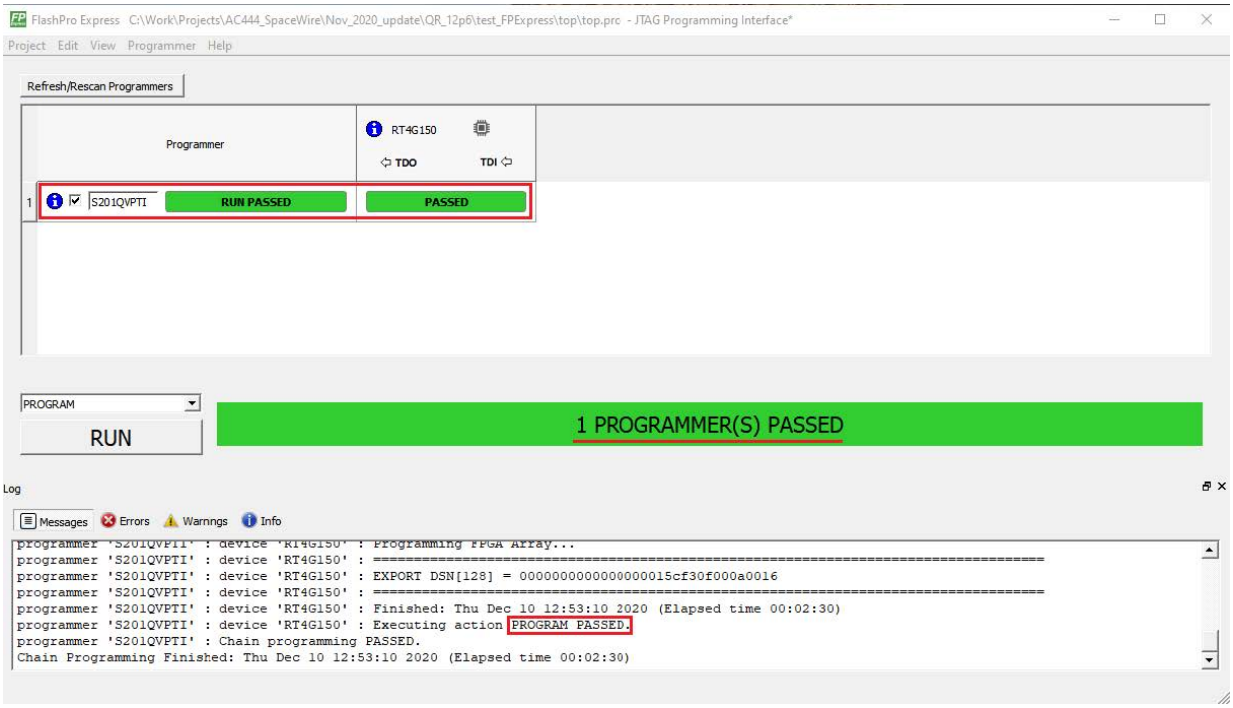
9. Click **OK**. The required programming file is selected and ready to be programmed in the device.
10. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan**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

Figure 20 • **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 21 • **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_dg0622_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.

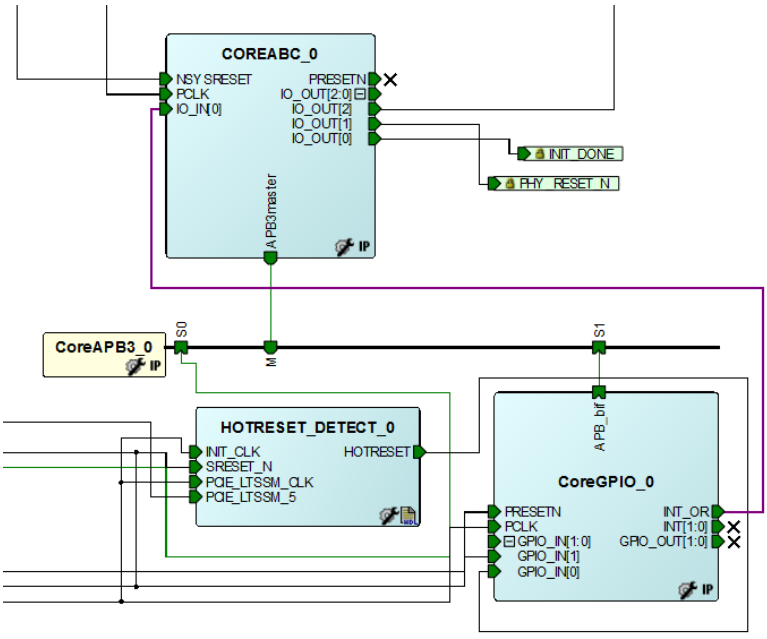
5 Appendix 3: Known Issues

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 a proper endpoint reset, as shown in Figure 22.
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:

- The HOTRESET_DETECT logic detects the hot reset from the root port by monitoring the LTSSM[5] signal from the SERDES block and generates HOTRESET.
- 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.
- 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.
- 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**



6 Appendix 4: Register Details

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

Table 4 • **Register Details**

Register Name	Register Address	Description
PC_BASE_ADDR	0x8028	Host PC memory base address provided by the driver.
DMA_DIR	0x8008	DMA direction: **Direction – Register Value** 1. PCIe → DDR memory – 0x11AA0001 2. DDR → PCIe memory – 0x11AA0002 3. LSRAM → DDR memory – 0x11AA0003 4. DDR → LSRAM memory – 0x11AA0004 5. PCIe → LSRAM memory – 0x11AA0005 6. LSRAM → PCIe memory – 0x11AA0006 To reset the DMA, the register value is 0x11AA0007. 1. Before initiating DMA transactions, reset the DMA with the register value, 0X11AA0007. 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 other.
DMA_CH0_STATUS	0x8100	DMA Channel-0 status: • DMA_CH0_STATUS[31] 1: DMA operation completed 0: DMA operation not completed • DMA_CH0_STATUS[15:0] = CLK count
DMA_CH1_STATUS	0x8108	DMA Channel-1 status: • DMA_CH1_STATUS[31] 1: DMA operation completed 0: DMA operation not completed • DMA_CH1_STATUS[15:0] = CLK count
RW_REG	0x0	Scratchpad register for PCIe R/W
LED_CTRL	0xA0	LEDs control register
SWITCH_STATUS	0x90	DIP switch status
CLK_FREQ	0x8038	DMA controller clock frequency
BAR1 memory	0x9000 – 0x9FFF	Memory connected to BAR1

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