DG0808
Demo Guide
PolarFire FPGA PCIe EndPoint and DDR4 Memory Controller Data Plane Using Splash Kit

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.

50200808. 2.0 7/18

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 most current publication.

1.1 Revision 2.0

The document was updated for Libero SoC PolarFire v2.2 release.

1.2 Revision 1.0

The first publication of this document.

2 PolarFire FPGA PCIe EndPoint and DDR4 Memory Controller Data Plane

Microsemi PolarFire® FPGAs contain fully integrated PCIe EndPoint and Root Port subsystems with optimized embedded controller blocks that use the physical layer interface (PHY) of the transceiver. Each PolarFire device includes two embedded PCIe subsystem (PCIESS) blocks that can be configured either separately, or as a pair, using the PCIESS configurator in the Libero® SoC PolarFire software.

The PCIESS is compliant with the PCI Express Base Specification, Revision 2.1. It implements memory-mapped advanced microcontroller bus architecture (AMBA) advanced extensible interface 4 (AXI4) access to the PCIe space, and the PCIe access to the memory-mapped AXI4 space. For more information, see *UG0685: PolarFire FPGA PCI Express User Guide*.

The DDR subsystem addresses memory solution requirements for a wide range of applications with varying power consumption and efficiency levels. The subsystem can be configured to support DDR4 and LPDDR3 memory devices. The subsystem is intended for accessing DDR memories for applications that require high-speed data transfers and code execution. For more information DDR memory controller, see *UG0676: PolarFire FPGA DDR Memory Controller User Guide*.

This document explains how to use the accompanying reference design to demonstrate the high-speed data transfer capability of the PolarFire FPGA using the hardened PCIe EndPoint, and DDR4 controller IP. The PCIe controller, built-in direct memory access (DMA) controller, and the CoreAXI4DMAController IP are used to achieve high-speed, bulk data transfers, as follows:

- The PCIe controller's built-in DMA controller perform bulk-data transfer between contiguous/scatter gather memory locations on a host PC and contiguous memory locations of DDR4/LSRAM.
- The CoreAXI4DMACcontroller performs data transfers between DDR4 memory and LSRAM using the CoreAXI4DMA controller.

The demo also shows how to use pre-synthesized design simulations using PCIe BFM script to initiate the PCIe EndPoint DMA to perform data transfers between LSRAM, DDR4, and PCIe.

The Windows kernel mode PCIe device driver, developed using the Windows Driver Kit (WDK) platform, interacts with the PolarFire PCIe EndPoint from the host PC. A GUI application that runs on the host PC is provided to set up and initiate the DMA transactions between the host PC memory, DDR4, and the LSRAM memories of the PolarFire splash kit through the PCIe interface.

A user application interface is provided for the GUI to interact with the PCIe driver. The GUI can also initiates the DMA transactions between DDR4 and LSRAM through UART IF. If the host PC PCIe slot is not available, the DMA between DDR4 and LSRAM is exercised through UART IF.

The PCIe EndPoint reference design can be programmed using any of the following options:

- Using the stp file: To program the device using the stp file provided along with the design files, see Programming the Device Using FlashPro, page 31.
- Using Libero SoC PolarFire: To program the device using Libero SoC PolarFire, see Libero Design Flow, page 27. Use this option when the reference design is modified.

2.1 Design Requirements

The following table lists the hardware, software, and IP requirements for this demo design.

Table 1 • **Design Requirements**

Requirement	Version
Operating system	64-bit Windows 7 or 10
Hardware	
PolarFire Splash Kit (MPF300T-Splash-KIT) – PolarFire Splash Board – 12 V/5 A power adapter – USB 2.0 A-male to mini-B cable for UART and programming	Rev 2 or later
PCIe Edge card ribbon cable	
Host PC with PCIe compliant slot with x4 or higher width	
Software	
Libero SoC PolarFire	v2.2
Modelsim	10.5c Pro
Synplify Pro	L-2016.09M-SP1-5
IP	
PF_XCVR_REF_CLK	1.0.103
PF_TX_PLL	1.0.112
PF_PCIE	1.0.242
PF_CCC	1.0.113
PF_RESET	2.1.100
PF_OSC	1.0.102
NGMUX	1.0.101
PF_INIT_MONITOR	2.0.103
CoreAXI4Interconnect	2.5.100
COREAXI4DMACONTROLLER	2.0.100
DDR4	2.3.108
CoreAHBLite	5.3.101
CoreAPB	4.1.100
CoreAXItoAHBL	3.2.104
CoreAHBtoAPB3	3.1.100
CoreUART	5.6.102
CoreAPB3	4.1.100
PolarFire SRAM	1.1.125
PCIe_DRI	1.0.101

2.2 Prerequisites

Before you start:

1. Download the design files from the following link:
   http://soc.microsemi.com/download/rsc/?f=mpf_dg0808_liberosocpolarfirev2p2_df
2. Download and install Libero SoC PolarFire on the host PC from the following location:
   https://www.microsemi.com/products/fpga-soc/design-resources/design-software/libero-soc-polarfire#downloads

The latest versions of ModelSim, Synplify Pro, and FTDI drivers are included in the Libero SoC PolarFire installation package.

2.3 Demo Design

The top-level block diagram of the PCIe EndPoint demo design is shown in Figure 1, page 5. Any external PCIe root-port or bridge can establish PCIe link with the PolarFire FPGA PCIe EndPoint and access the control registers, DDR4, and fabric memory through BAR space using the memory write (MWr) and memory read (MRd) transaction layer packets (TLPs). The PCIe EndPoint converts these MWr and MRd TLPs into AXI4 master interface transactions and accesses the fabric memory through CoreAXI4Interconnect IP.

The PCIe Demo application on the host PC initiates the DMA transfers through the PCIe device drivers. The driver on the host PC allocates memory and initiates the DMA Engine in the PolarFire PCIe controller by accessing the PCIe DMA registers through BAR0. The PCIe controller has two independent DMA Engines:

- DMA Engine0: performs DMA from host PC memory to DDR4/LSRAM.
- DMA Engine1: performs DMA from DDR4/LSRAM to host PC memory.

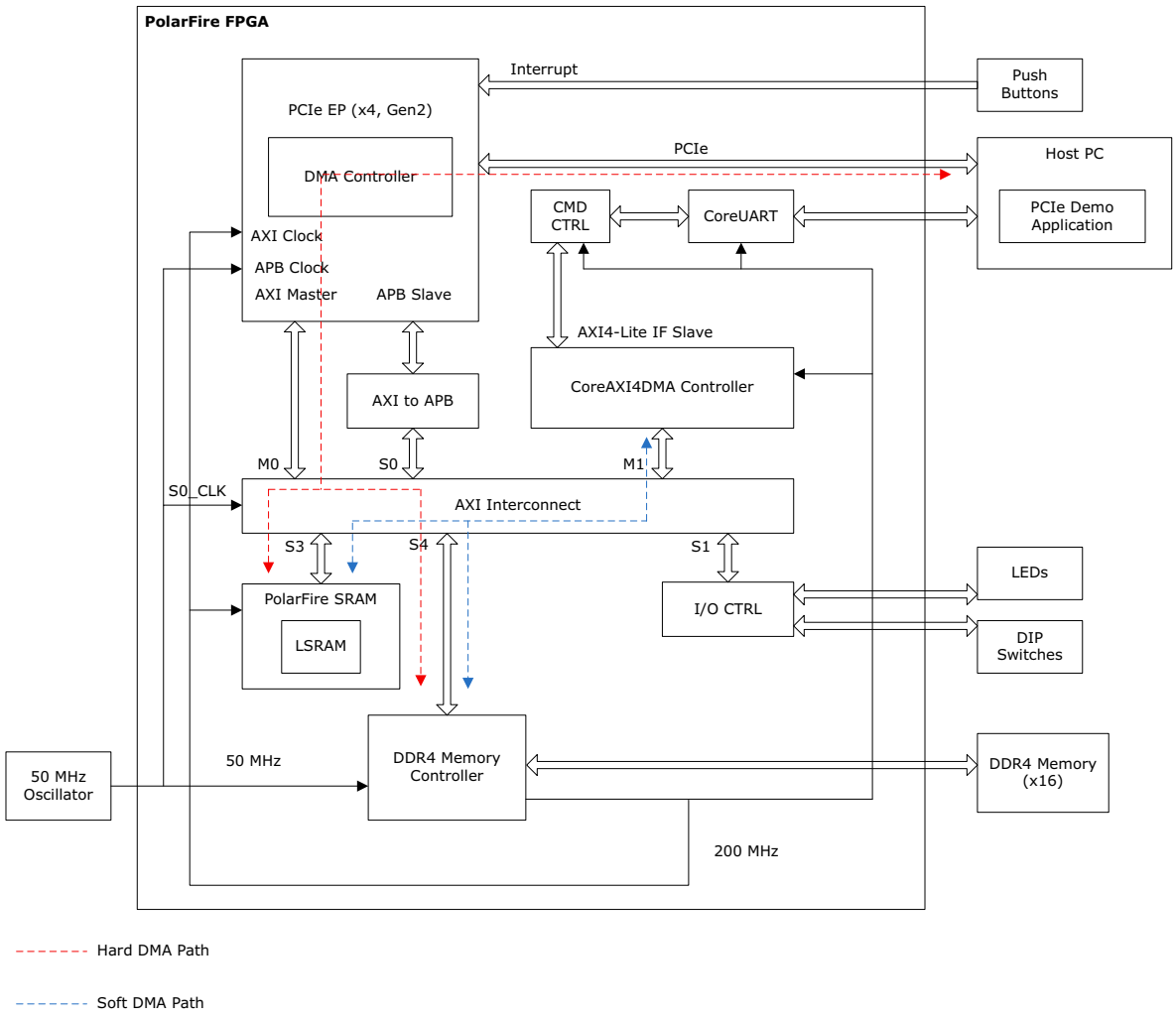
Note: For SGDMA type of DMA operations, the PCIe driver finds the available memory locations and creates the buffer descriptor chain for the different memory locations. It also configures the PCIe DMA for SGDMA and the base address of the first buffer descriptor.

The PCIe demo application initiates CoreAXI4DMA controller IP to perform the DMA between DDR4 memory and LSRAM. The following are the two channels of the CoreAXI4DMA controller IP:

- Channel0: performs DMA from—DDR4 to LSRAM
- Channel1: performs DMA from—LSRAM to DDR4

The host PC application initiates the CoreAXI4DMA controller IP depending on the DMA type through BAR2 when the PCIe edge connector is connected to the host PC PCIe slot. The host PC application also initiates the CoreAXI4DMA controller IP through UART IF. This option is provided to exercise the DDR throughputs when the PolarFire Splash kit is not connected to the host PC PCIe slot.

Figure 1 • **PCIe Demo Design Top-Level Block Diagram**



2.3.1 Design Data Flow

The demo design performs the following control plane operations:

- LED Blink: host PC driver performs BAR2 memory write operation (MWr) to EndPoint. The PCIe controller generates AXI write transaction on AXI_IO_CTRL logic's to blink LEDs.
- DIP Switch Read: host PC driver performs BAR2 memory read operation (MRd) to EndPoint. The PCIe controller generates AXI read transaction on AXI_IO_CTRL logic's to blink LEDs.
- MSI Interrupt Count: when on-board push button is pressed, the PCIe EndPoint generates interrupt to host PC and the host PC driver increments the corresponding interrupt counter.
- Memory Read/Write: host PC driver configures the ATR2 translation address to DDR4/LSRAM base address. It performs BAR2 memory read/write transactions to DDR4/LSRAM memories.

The demo design supports three types of DMA operations.

- Continuous DMA operations
- SDGMA Operations
- Core DMA Operations

2.3.1.1 Continuous DMA Operations

The PCIe DMA0/DMA1 controllers perform DMA between continuous memory locations when SGDMA mode is disabled. The following sections explain the data flow of DMA0 and DMA1.

2.3.1.1.1 DMA0 – Host PC Memory to DDR4/LSRAM

PCIe DMA Engine0 performs continuous DMA from host PC memory to DDR4/LSRAM memories as described in the following steps:

1. PolarFire_PCIe_GUI application sets up the DMA controller through the PCIe link. This includes DMA source and destination, address, and size.
2. DMA controller initiates a read transaction to the PCIe core.
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 is written to the DDR4/LSRAM memories using PCIe AXI master interface.
6. The DMA controller repeats this process (from step 2 to 5) until the DMA size of data transfer is completed.
7. The DMA controller sends the MSI0 interrupt to the host PC, the driver on the host PC detects the interrupt, reads the DMA status, and the number of clock cycles consumed to complete the DMA transaction to the PolarFire_PCIe_GUI application.

2.3.1.1.2 DMA1 – DDR4/LSRAM to Host PC Memory

PCIe DMA Engine1 performs continuous DMA from DDR4/LSRAM memories to host PC memory as described in the following steps.:

1. PolarFire_PCIe_GUI application sets up the DMA controller through the PCIe link. This includes DMA source and destination, address, and size.
2. DMA controller initiates an AXI burst read transaction to read the data from DDR4/LSRAM memories.
3. The DMA controller initiates write transaction to PCIe core with the read data. The PCIe core sends a memory write (MWr) TLP to the host PC.
4. The DMA controller repeats this process (steps 2 and 3) until the DMA size of data transfer is completed.
5. The DMA controller sends the MSI1 interrupt to the host PC. The driver on the host PC detects the interrupt, reads the DMA status, and the number of clock cycles consumed to complete the DMA transaction to the PolarFire_PCIe_GUI application.

2.3.1.2 SGDMA Operations

The PCIe DMA0/DMA1 performs DMA between scattered host PC memory locations and continuous memories of PolarFire when SGDMA mode is enabled.

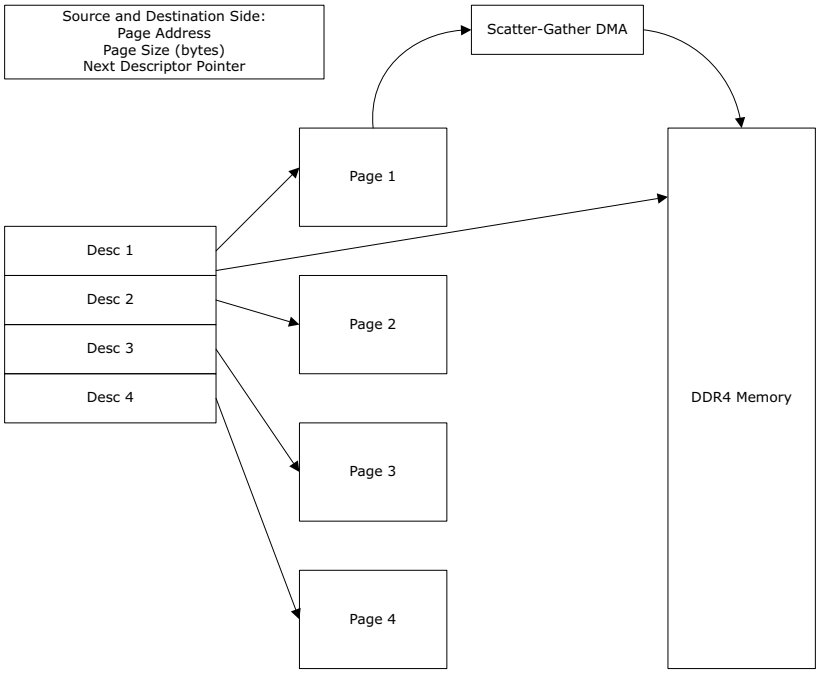
2.3.1.2.1 Host PC Memory to DDR4

PCIe DMA Engine0 performs DMA from host PC memory to DDR4 memories as shown in the following figure.

The following steps describe the SGDMA operation of PCIe DMA0:

1. PolarFire_PCIe_GUI application requests the PCIe driver for SG DMA. The driver on the host PC allocates the available memory location and creates the buffer descriptors with the scattered memory location addresses and location size.
2. The destination DDR4 memory is treated as the continuous memory. The driver configures the PCIe DMA0 with the first buffer descriptor address and initiates the DMA.
3. DMA controller initiates read transaction to the PCIe core with the buffer descriptor address.
4. The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the host PC. The host PC returns a completion (CplD) TLP to the PCIe link.
5. The DMA controller extracts these buffer descriptors and initiates the read transaction to PCIe core with the host PC memory location address in the descriptor.
6. The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the host PC. The host PC returns a completion (CplD) TLP to the PCIe link.
7. This return data is written to the DDR4 memories using PCIe AXI master interface.
8. The DMA controller repeats this process (from step 3 to 7) until the DMA size of data transfer is completed.
9. The DMA controller sends the MSI0 interrupt to the host PC. The driver on the host PC detects the interrupt, reads the DMA status, and the number of clock cycles consumed to complete the DMA transaction to the PolarFire_PCIe_GUI application.

Figure 2 • **DMA0 – Example of SG DMA Operation**



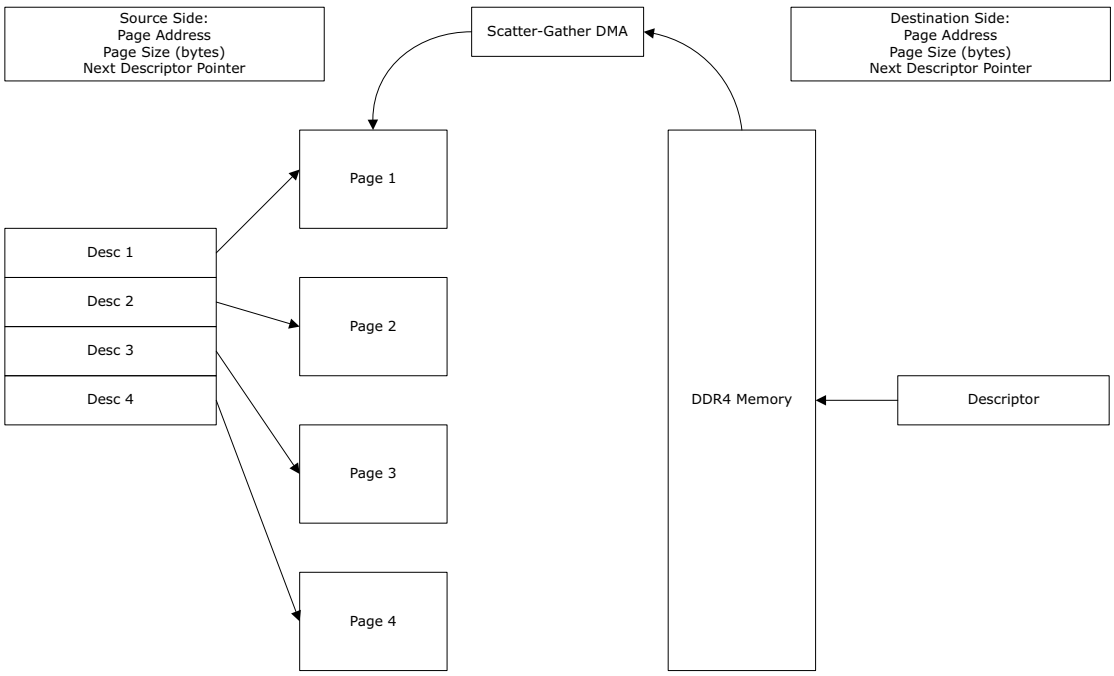
2.3.1.2.2 DDR4 to Host PC Memory:

PCIe DMA Engine1 performs DMA from DDR4 memories to host PC memory as shown in the following figure.

The following steps describe the SGDMA operation of PCIe DMA1:

1. PolarFire_PCIe_GUI application requests the PCIe driver for SG DMA. The driver on the host PC allocates the available memory locations and creates the buffer descriptors with the scattered memory location addresses and location size.
2. The source DDR4 memory is treated as the continuous memory. Single buffer descriptor is created in LSRAM with the base address of DDR4 memory. The LSRAM base address is provided to DMA controller for source descriptor address.
3. The driver configures the PCIe DMA1 with the first host PC destination buffer descriptor address and initiates the DMA.
4. DMA controller initiates read transaction to the PCIe core with the buffer descriptor address.
5. The PCIe core sends the memory read (MRd) transaction layer packets (TLP) to the host PC. The host PC returns a completion (CplD) TLP to the PCIe link.
6. The DMA controller extracts these buffer descriptors and initiates an AXI burst read transaction to read the data from DDR4 memories.
7. With this read data, DMA controller initiates the write transaction to PCIe core with the host PC memory location address in the descriptor.
8. The PCIe core sends the memory write (MWr) transaction layer packets (TLP) to the host PC.
9. The DMA controller repeats this process (from step 4 to 8) until the DMA size of data transfer is completed.
10. The DMA controller sends the MSI1 interrupt to the host PC. The driver on the host PC detects the interrupt, reads the DMA status, and the number of clock cycles consumed to complete the DMA transaction to the PolarFire_PCIe_GUI application.

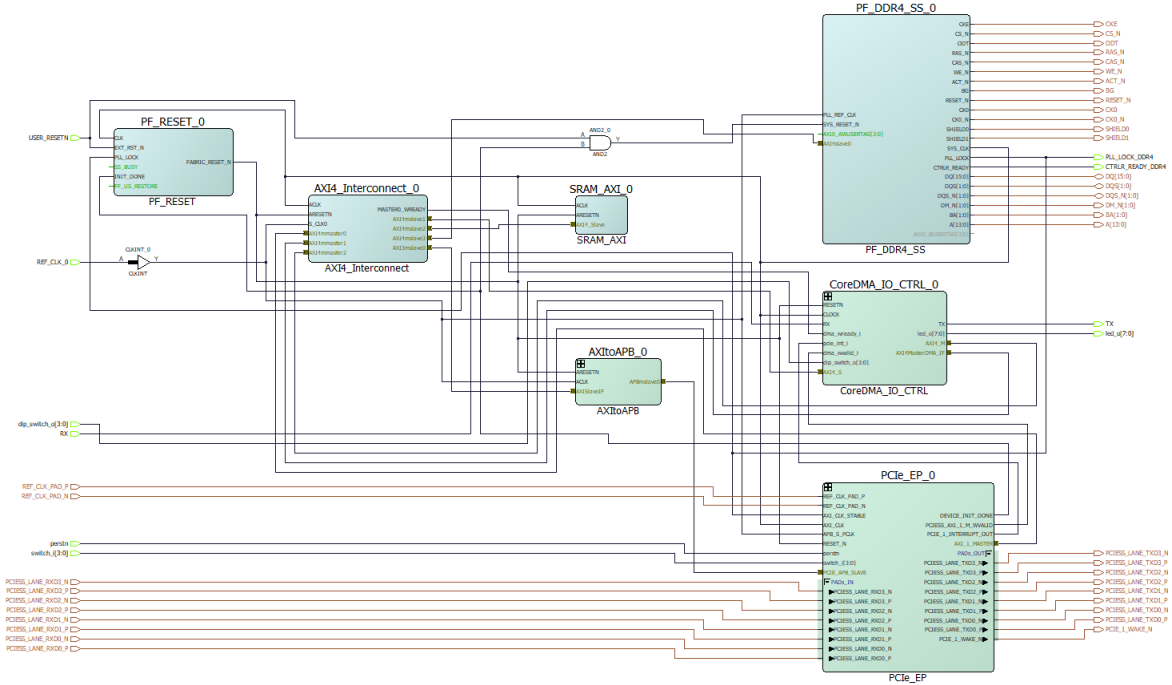
Figure 3 • **DMA1 – Example of SG DMA Operation**



2.3.2 Design Implementation

The following figure shows the Libero SoC PolarFire software design implementation of the PCIe EndPoint reference design.

Figure 4 • **PCIe EndPoint Reference Design**

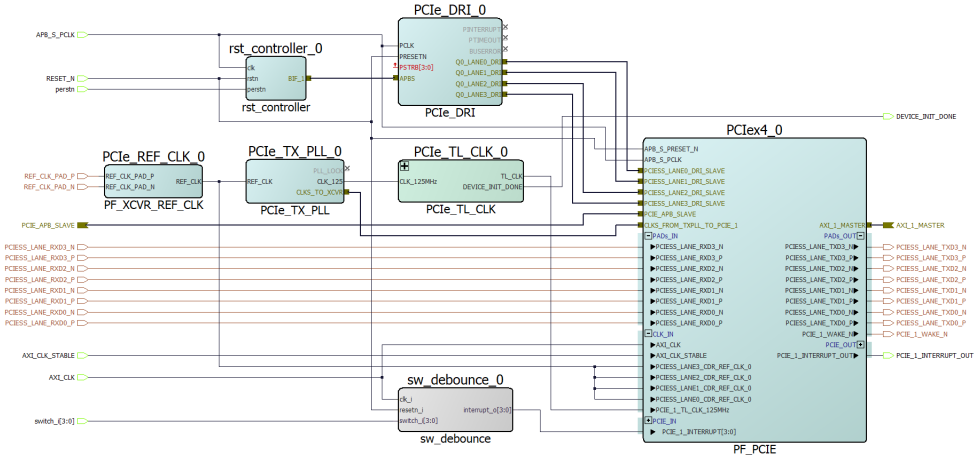


The top-level design includes the following SmartDesign components:

- PCIe_EP
- AXI_to_APB
- CoreDMA_IO_CTRL

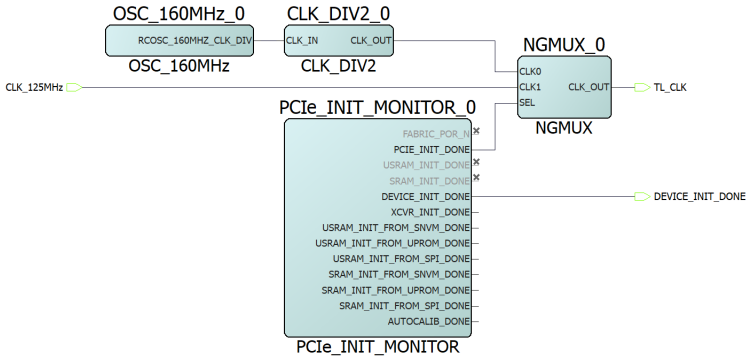
The PCIe_EP SmartDesign implements PCIe EndPoint and its clocking scheme as shown in the following figure. It also includes the sw_debounce module, which is used to suppress bounces from on-board push buttons and to generate a pulse to the PCIe controller interrupt line. The rst_controller logic is used to reset the PCIe EndPoint when host PC generates the EndPoint reset through PCIe PERSTn side band signal. The rst_controller.v fabric logic monitors the PERSTn signal of PCIe Edge card. It performs the assertion and de-assertion of PCIe and PCS soft resets on raising edge of PERSTn signal. It uses the dynamic reconfiguration interface to access the PCIe and PCS soft reset registers.

Figure 5 • **PCIe_EP SmartDesign**



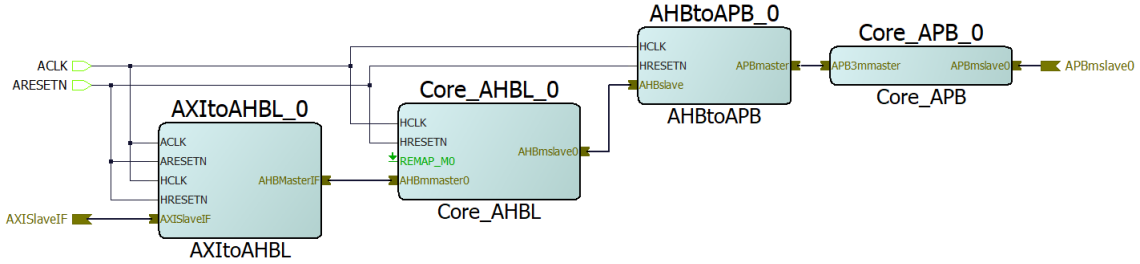
The PCIe_TL_CLK SmartDesign implements PCIe TL CLK for PolarFire devices as shown in the following figure. PCIe TL CLK needs to be connected to CLK_125MHZ of Tx PLL. In PolarFire devices, TL CLK is available only after PCIe initialization. The 80 MHz clock is derived from the on-chip 160 MHz oscillator to drive the TL CLK during PCIe initialization. The NGMUX is used to switch this clock to the required CLK_125MHz after PCIe initialization.

Figure 6 • **PCIe_TL_CLK SmartDesign**



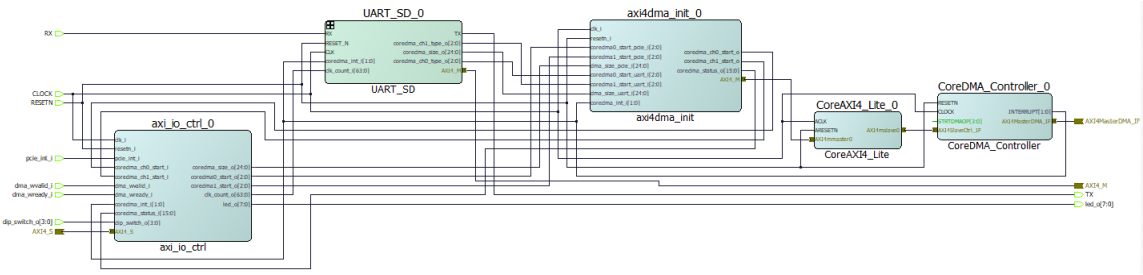
The AXI_to_APB SmartDesign implements AXI to APB using different IP cores as shown in the following figure. AXI to APB IF accesses the PCIe control registers through the PCIe APB IF from the BAR0 space.

Figure 7 • **AXI_to_APB SmartDesign**



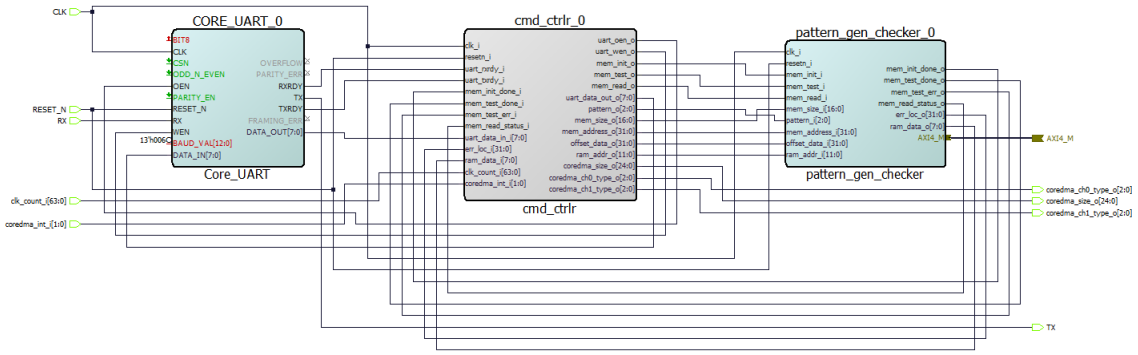
The CoreDMA_IO_CTRL SmartDesign implements fabric registers, CoreDMA4DMA IP initialization, and UART_SD as shown in the following figure.

Figure 8 • **CoreDMA_IO_CTRL SmartDesign**



The UART_SD SmartDesign implements logic required for UART IF as shown in the following figure.

Figure 9 • **UART SmartDesign**



2.3.3 IP Configuration

The following IPs and macros need to be configured before simulating the demo design:

- PCIe
- Transceiver reference clock
- Transmit PLL
- CoreAXI4Interconnect IP
- PolarFire SRAM IP
- DDR4
- CoreRESET_PF
- CoreAXI4DMAController IP

2.3.3.1 PCIe

The PCIESS is configured as an EndPoint with maximum link speed and maximum link width—Gen2 (5.0 Gbps) link speed and x4 link width. The **Simulation Level** in the configurator is set to BFM to simulate the design using PCIe BFM script, as shown in the following figure. The PCIe fabric interface is always the same regardless of the link width or lane rate. APB interface is enabled to access the PCIe DMA and Address translation registers.

Figure 10 • **PCIe Configurator**

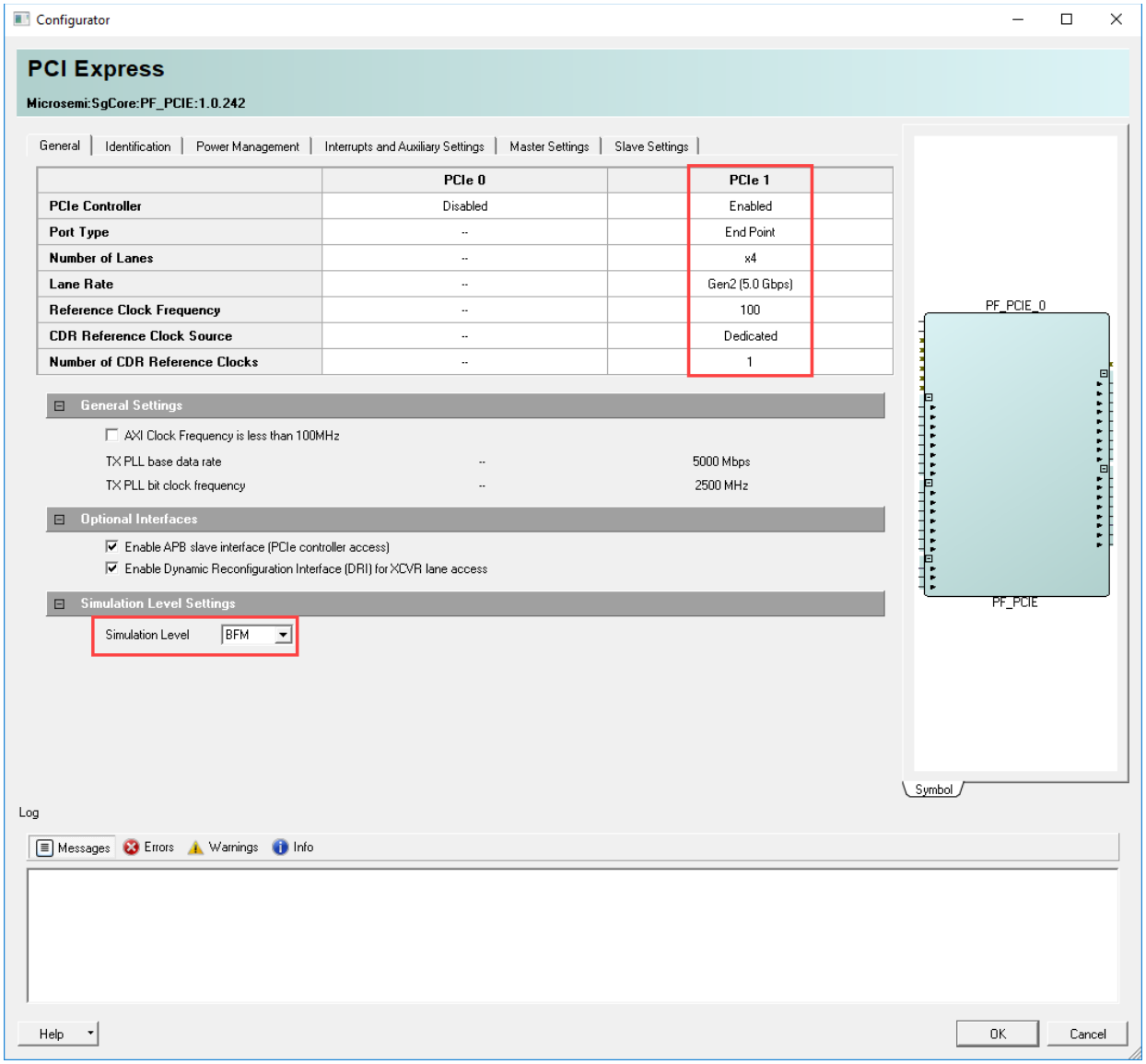


Figure 11 • **PCIe Configurator—BAR 0 Master Settings**

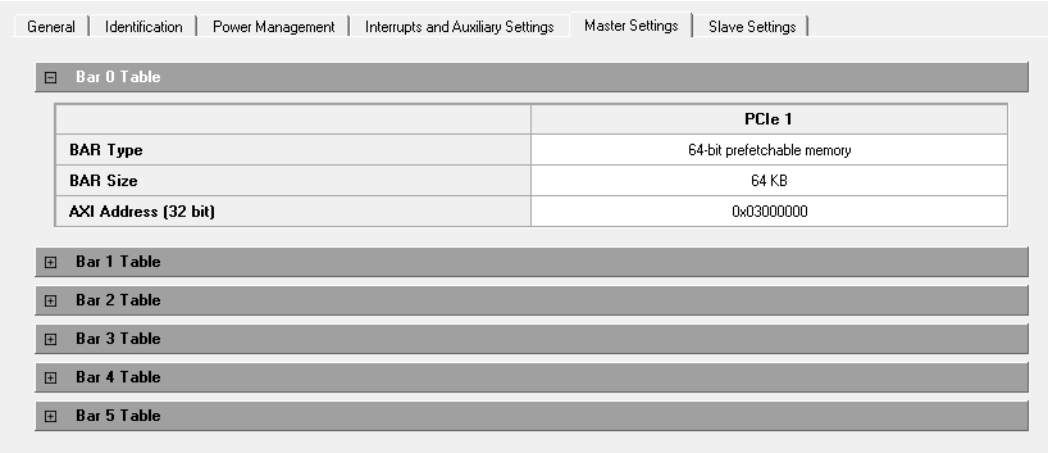
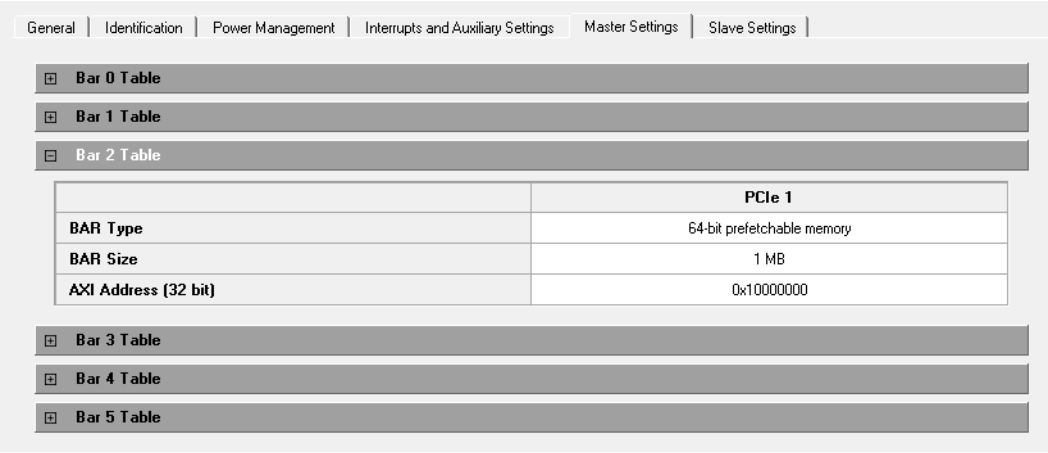


Figure 12 • **PCIe Configurator—BAR 2 Master Settings**



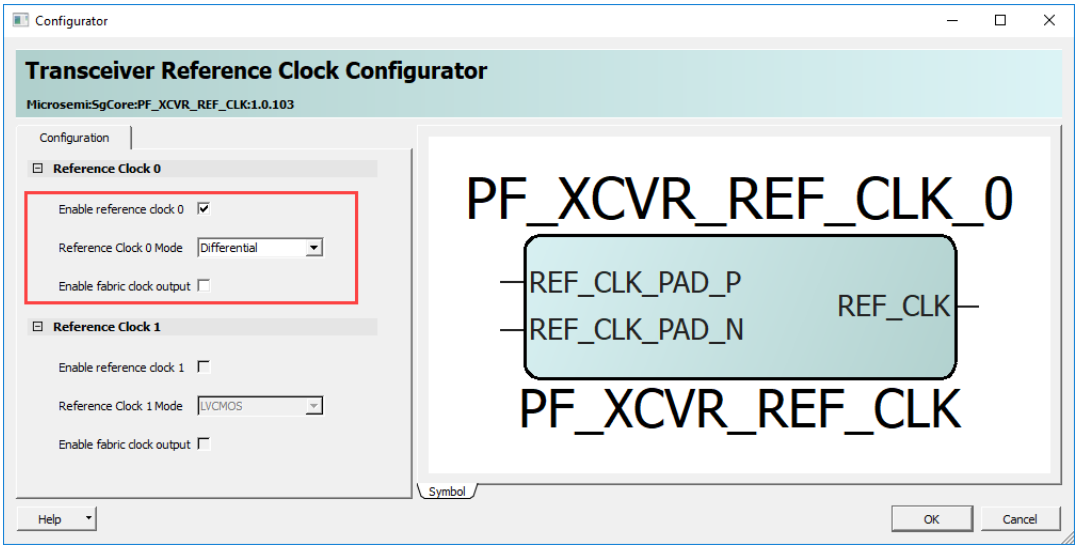
The following two BARs are configured in 64-bit:

- **BAR0**: accesses the PCIe DMA, address translation, and interrupt registers through the PCIe controller's APB interface. The address translation register associated with BAR0 is configured to translate the BAR0 address to the PCIe APB IF base address (0x0300_0000).
- **BAR2**: accesses the fabric control registers and AXI LSRAM, and DDR4 memories. By default, the address translation register associated with BAR2 is configured to access the fabric control registers (0x1000_0000). To access the LSRAM, and DDR4 memories, the driver on the host PC configures the BAR2 address translation register (TRSL_ADDR) to LSRAM (0x3000_0000)/DDR4 (0x4000_0000) memory base address using the PCIe APB IF through BAR0.

2.3.3.2 Transceiver Reference Clock

The transceiver reference clock can be configured either as a differential, or two single-ended REFCLKs. In this reference design, the **Reference Clock 0** is configured as a differential reference clock as shown in the following figure. This demo requires only one REFCLK (Reference Clock 0). The REFCLK is the clock source for transceivers and global clock network in this design.

Figure 13 • **Transceiver Reference Clock Configurator**

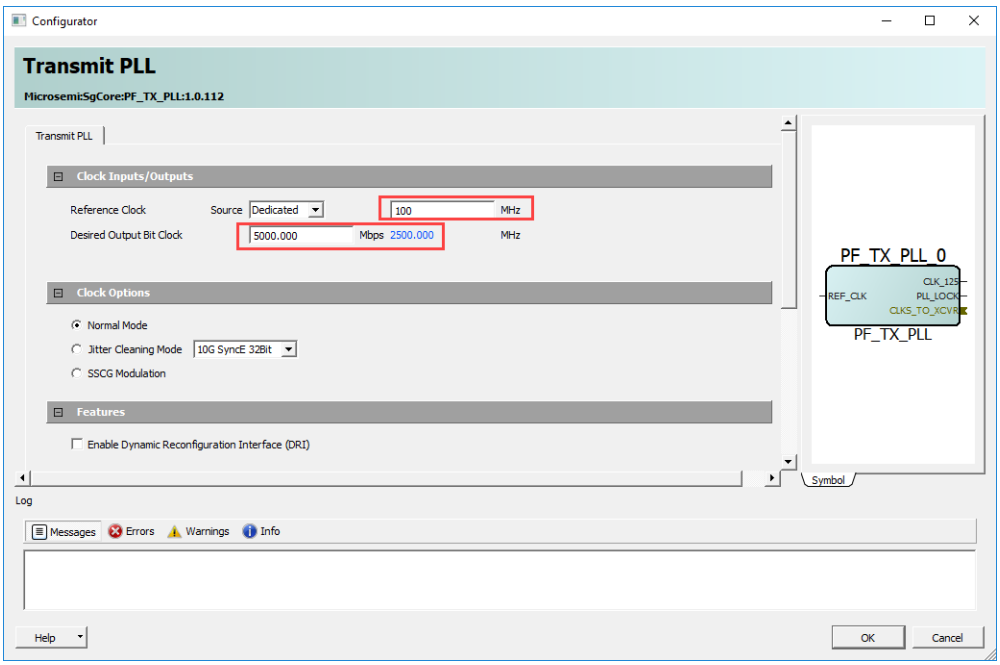


2.3.3.3 Transmit PLL

The Transmit PLL **Reference Clock** and **Desired Output Clock** are set to 100 MHz and 5000 Mbps, respectively, as shown in the following figure.

The PolarFire FPGA transceiver is a half-rate architecture, that is, the internal high-speed path uses both edges of the clock to keep the clock rates down. Therefore, the clock can run at half of the data rate, thereby consuming less dynamic power. The transceiver in PCIe mode requires a 2500 MHz bit clock.

Figure 14 • **Transmit PLL Configurator**

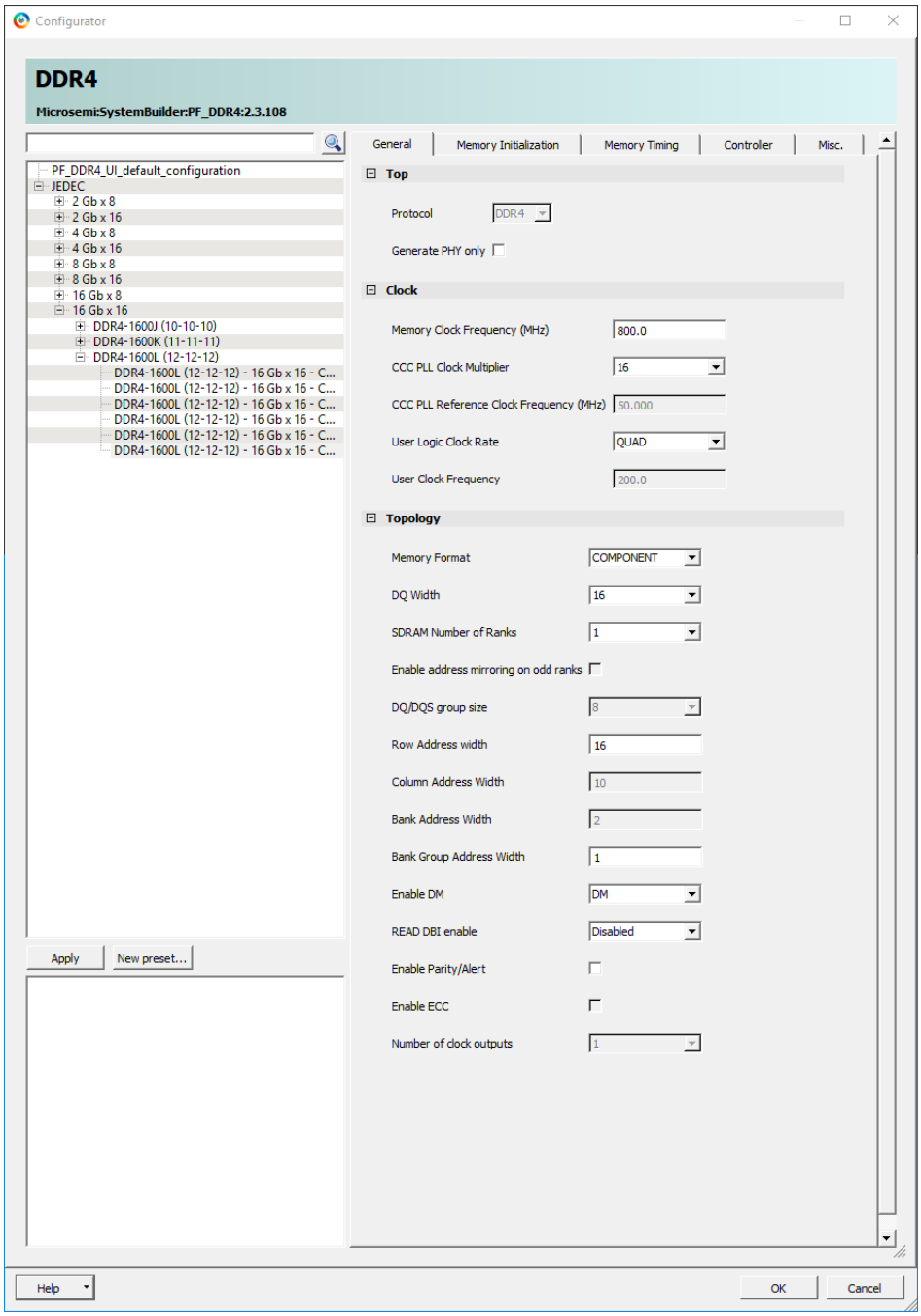


2.3.3.4 DDR4

The DDR4 subsystem is configured to access the 16-bit DDR4 memory through an AXI4 64-bit interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4 memory on PolarFire Splash kit. The following figure shows general configuration settings for the DDR4 memory.

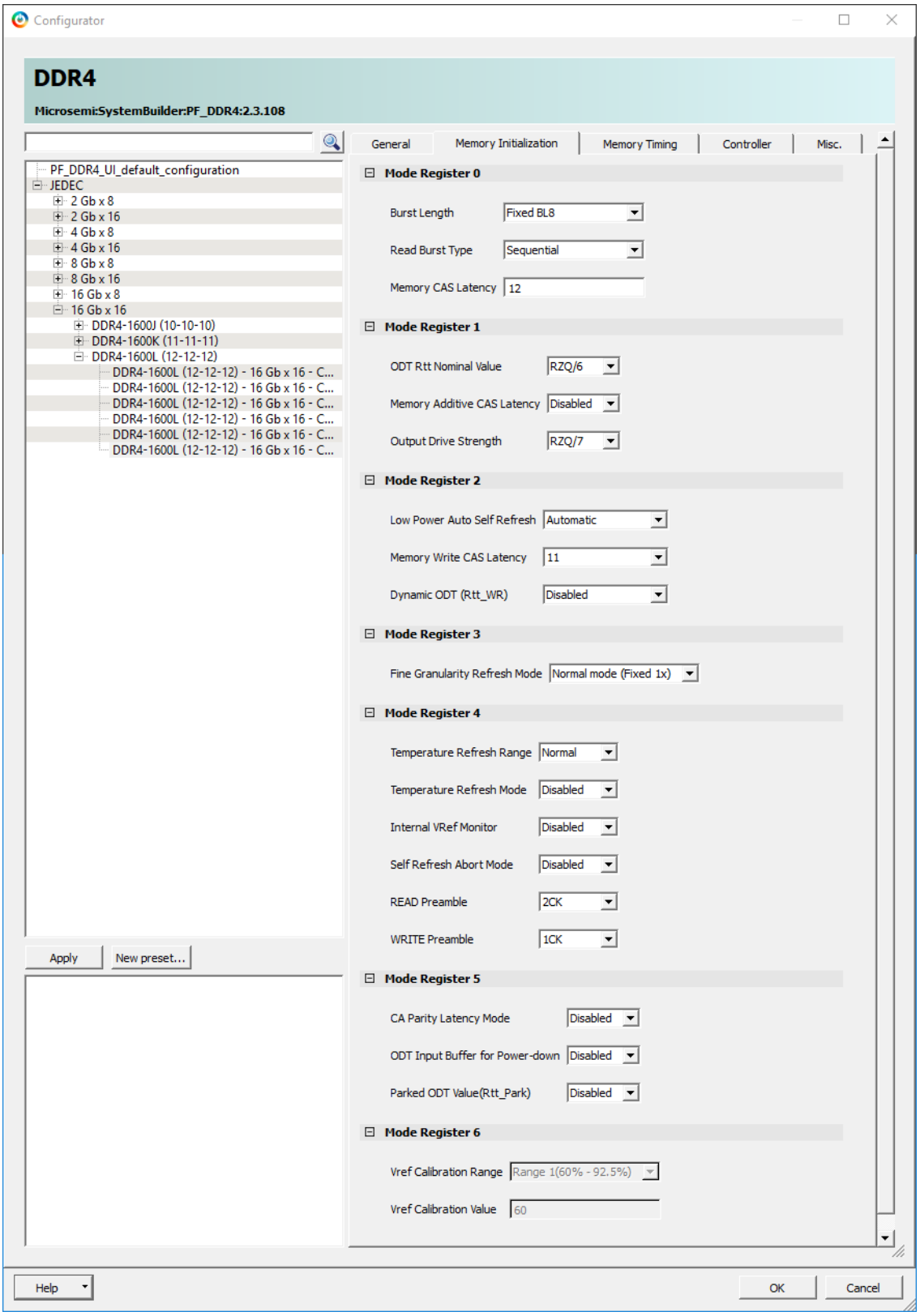
Note: The PolarFire Splash kit supports 32-bit DDR4 memory. This demo design uses only 16-bit DDR4 memory to meet the 200 MHz fabric logic Place and Route timing.

Figure 15 • **DDR4 Configurator**



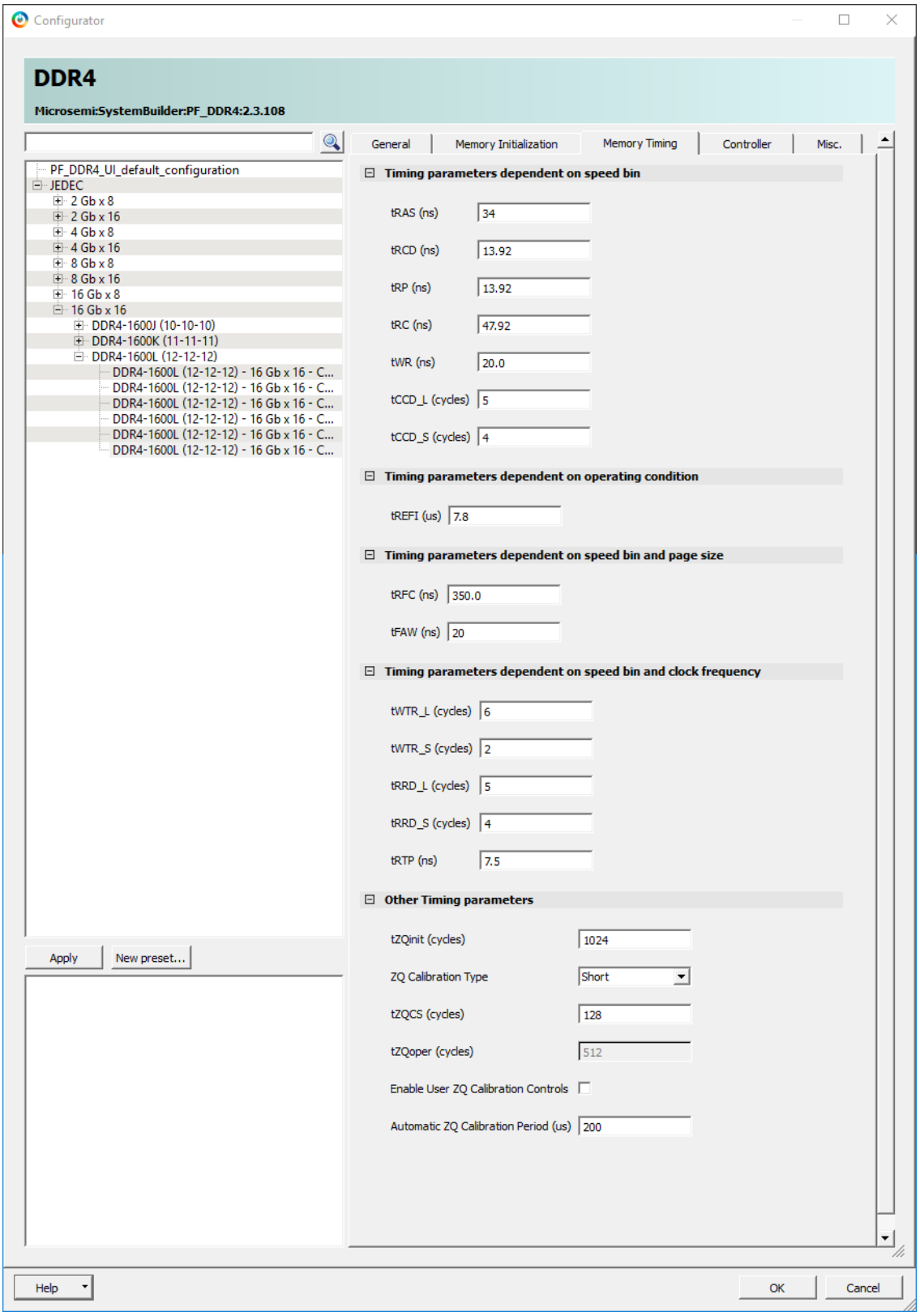
The following figure shows initialization configuration settings for the DDR4 memory.

Figure 16 • **DDR4 Configurator—Memory Initialization**



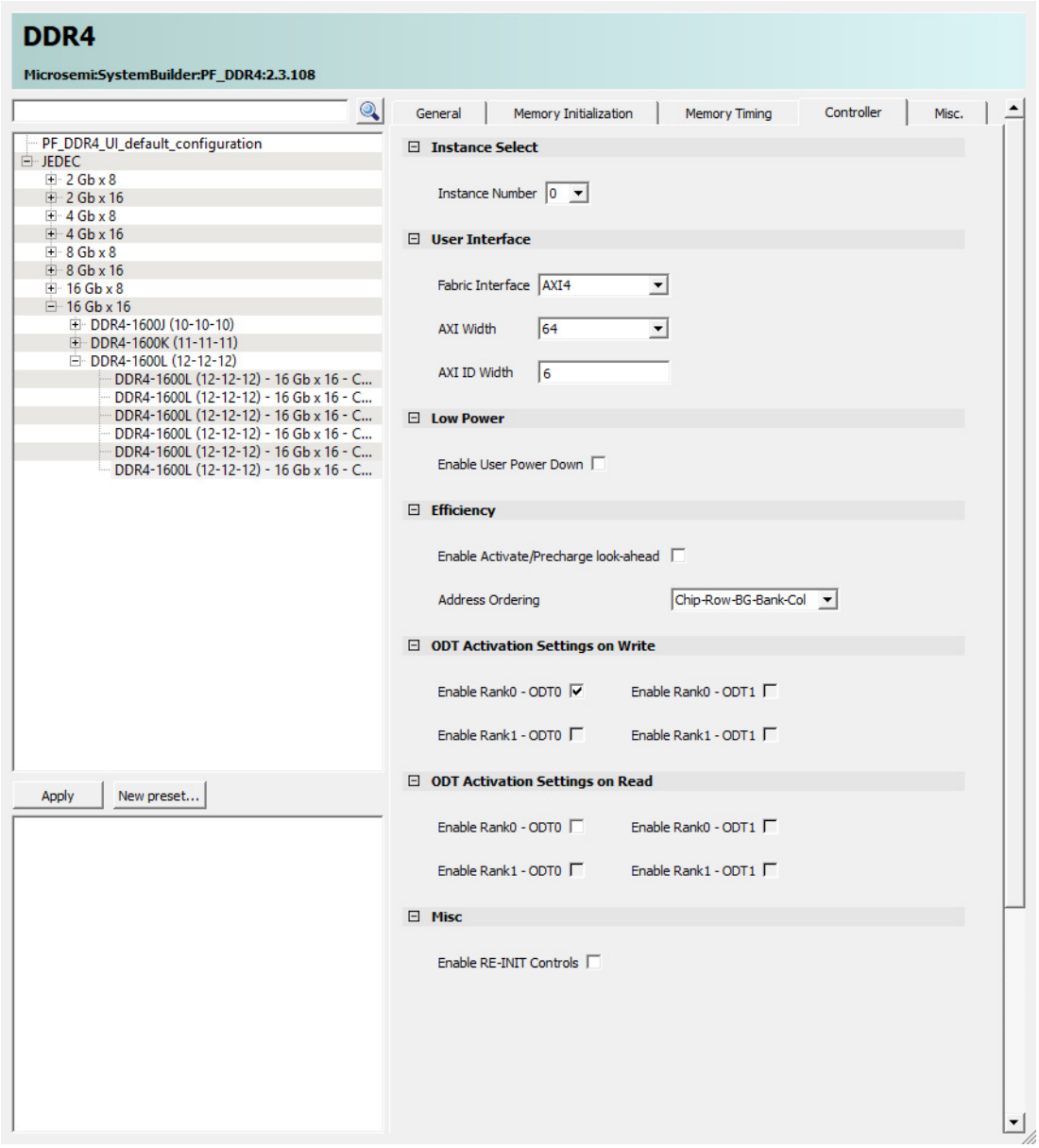
The following figure shows timing configuration settings for the DDR4 memory.

Figure 17 • **DDR4 Configurator—Memory Timing**



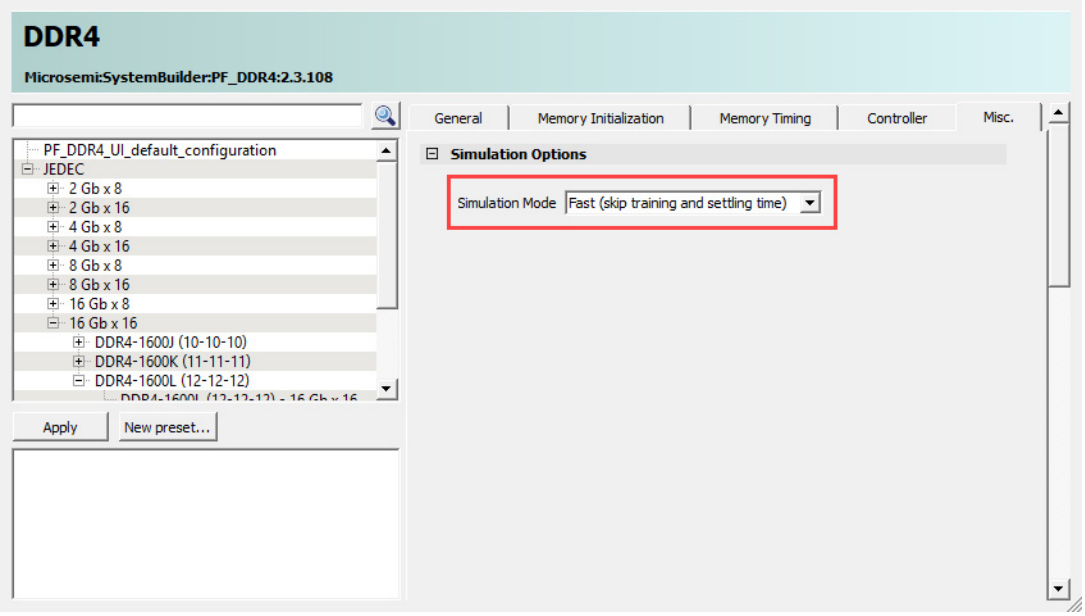
The following figure shows controller configuration settings for the DDR4 memory.

Figure 18 • **DDR4 Configurator—Controller**



The following figure shows miscellaneous configuration settings for the DDR4 memory.

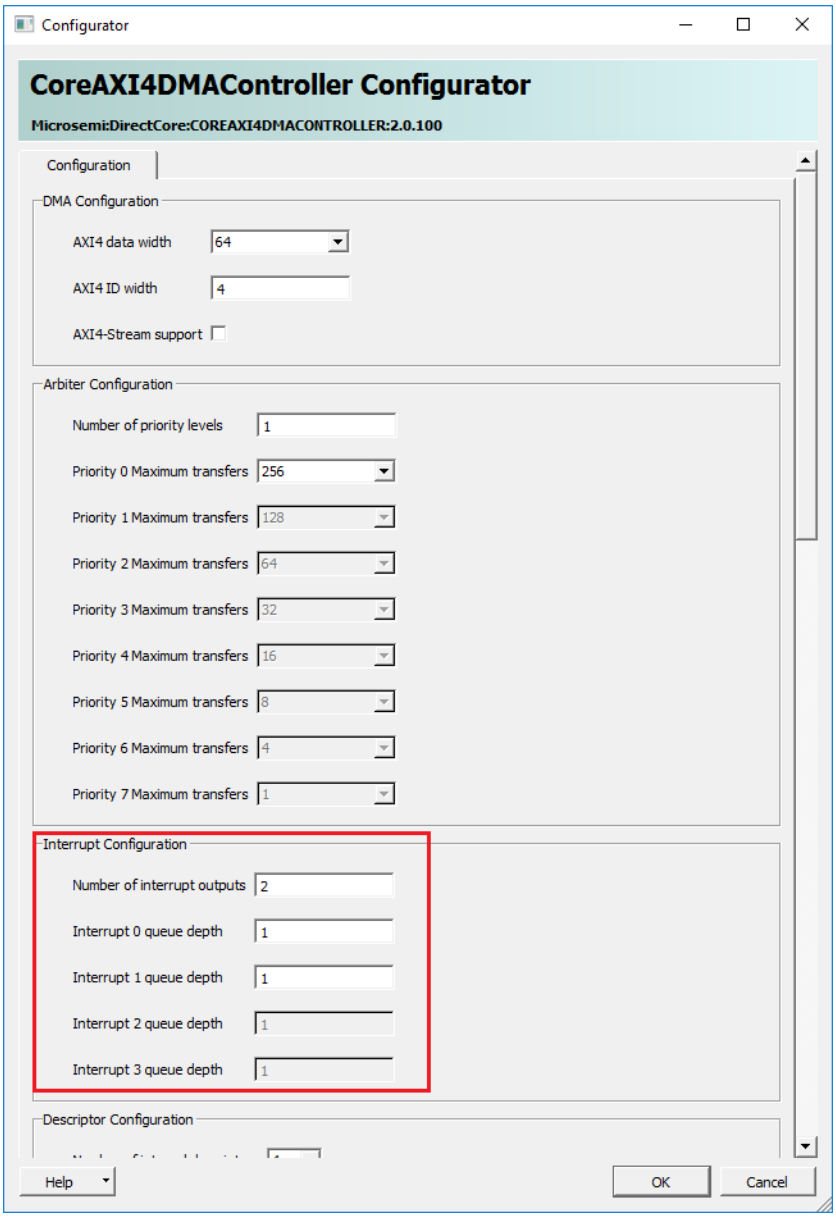
Figure 19 • **DDR4 Configurator—Misc**



2.3.3.5 CoreAXI4DMAController IP

The CoreAXI4DMAController IP is configured for 64-bit AXI4 data width, and to generate interrupts for descriptor0 and descriptor1. Descriptor0 is used for DDR4 to LSRAM DMA and descriptor1 is used for LSRAM to DDR4 DMA. The following figure shows the configuration settings for the CoreAXI4DMAController IP.

Figure 20 • **CoreAXI4DMAController IP Configurator**



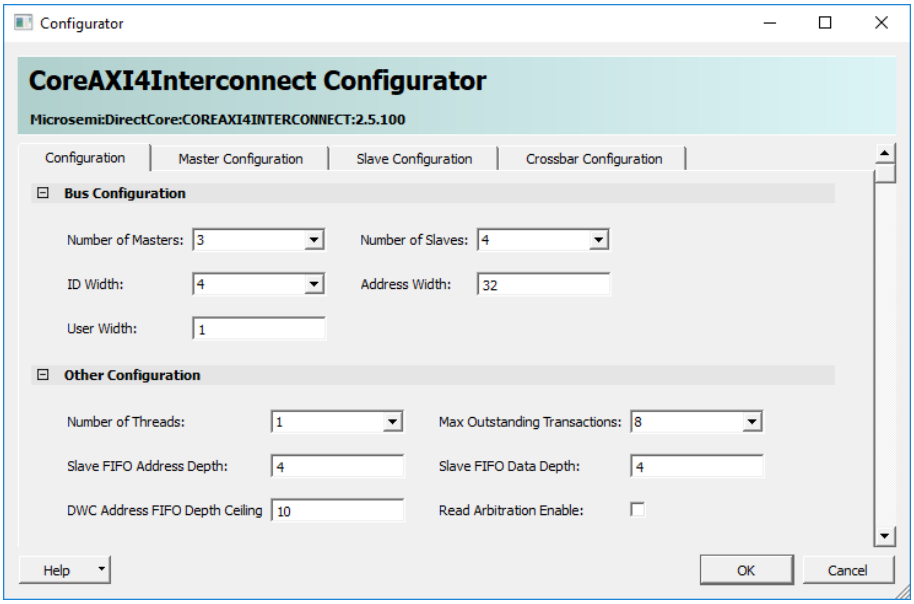
2.3.3.6 CoreAXI4Interconnect IP

The CoreAXI4Interconnect IP is configured for the following master and slave ports:

- Master0: PCIe
- Master1: CoreAXI4DMAController IP
- Slave0: AXItoAPB bridge (0x0000_0000 to 0x0FFF_FFFF)
- Slave1: AXI Slave Fabric Registers (0x1000_0000 to 0x1FFF_FFFF)
- Slave2: AXI4 LSRAM (0x3000_0000 to 0x3FFF_FFFF)
- Slave3: DDR4 Subsystem (0x4000_0000 to 0x4FFF_FFFF)

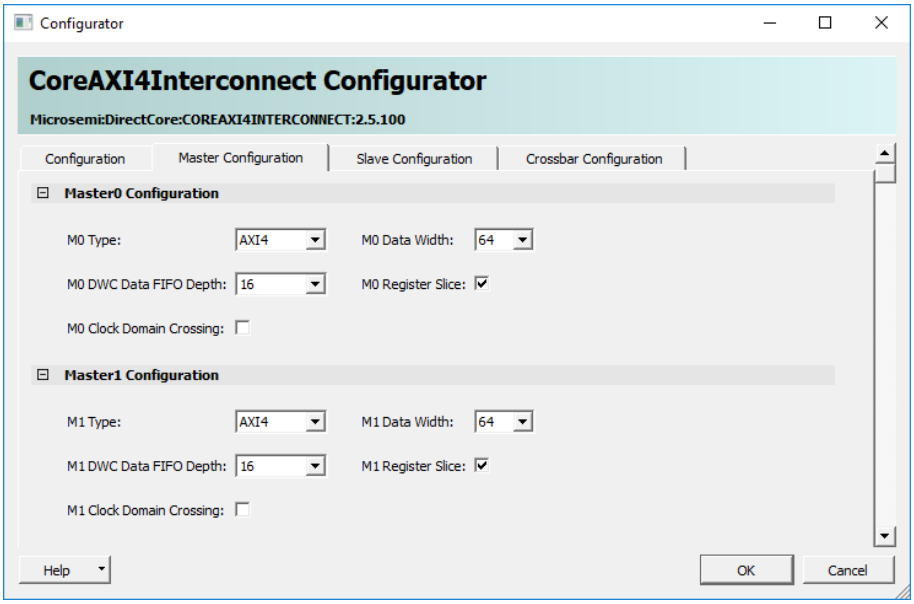
Slave0 is configured to convert AXI4 transactions to AXI3 transactions. The following figure show the CoreAXI4Interconnect IP configurations.

Figure 21 • **CoreAXI4Interconnect IP Core Configuration**



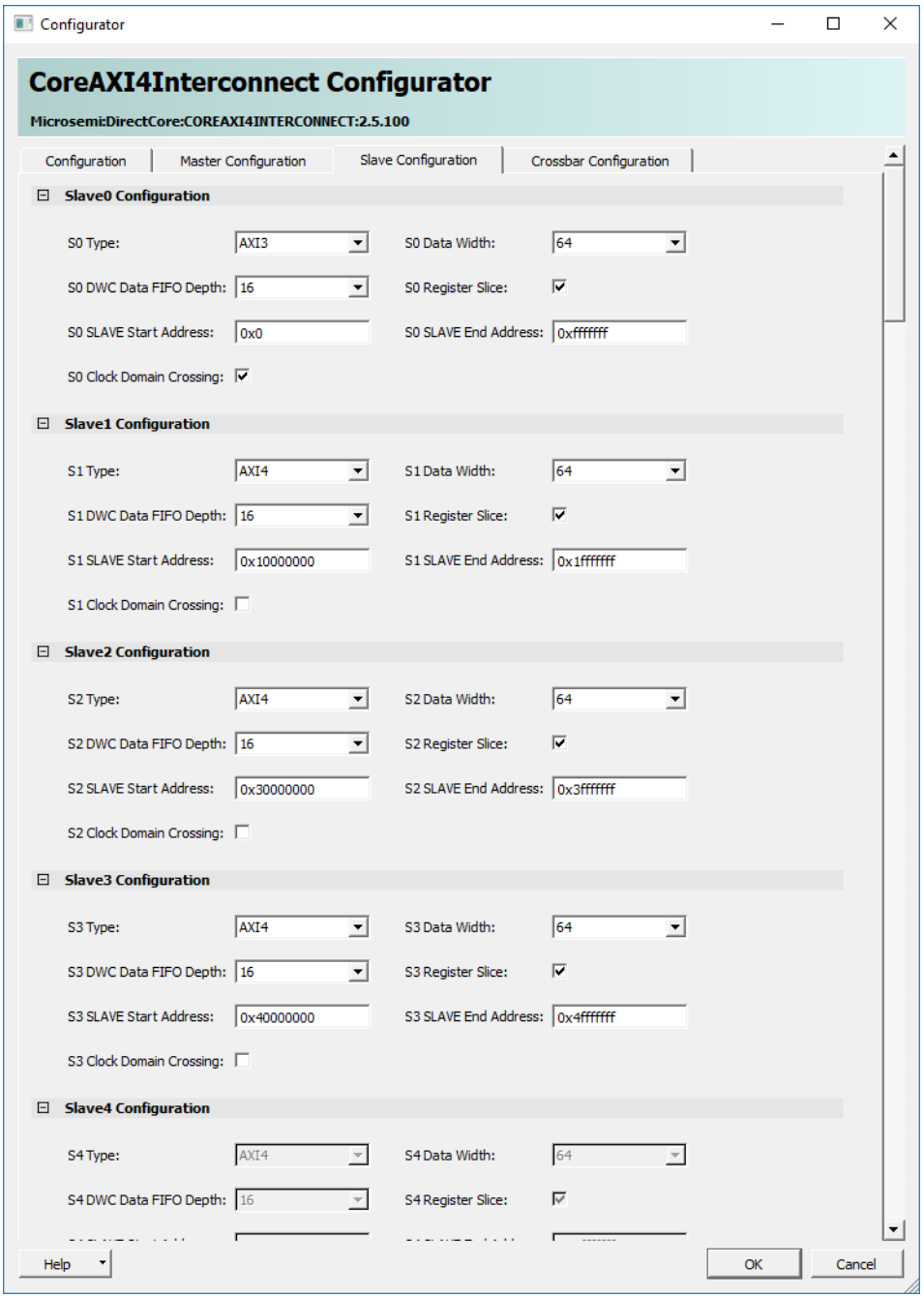
The following figure show the CoreAXI4Interconnect IP master configurations.

Figure 22 • **CoreAXI4Interconnect IP Master Configuration**



The following figure show the CoreAXI4Interconnect IP slave configurations.

Figure 23 • **CoreAXI4Interconnect IP Slave Configuration**

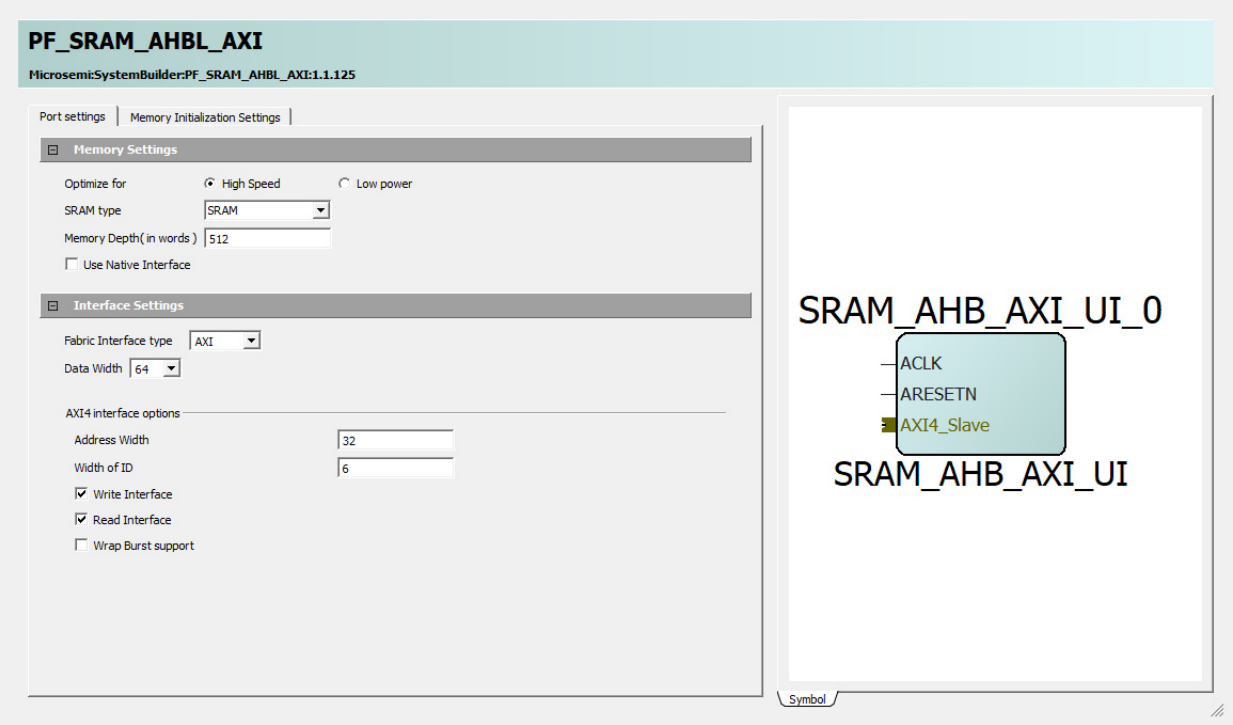


The CoreAXI4Interconnect IP is designed for high bandwidth data movement. It supports bus protocol and bus width converters for each master and slave interface.

2.3.3.7 PolarFire SRAM IP

The SRAM IP is configured to access the 4 KB fabric memory (LSRAM) using the AXI4 interface.

Figure 24 • **PolarFire SRAM IP Configurator**

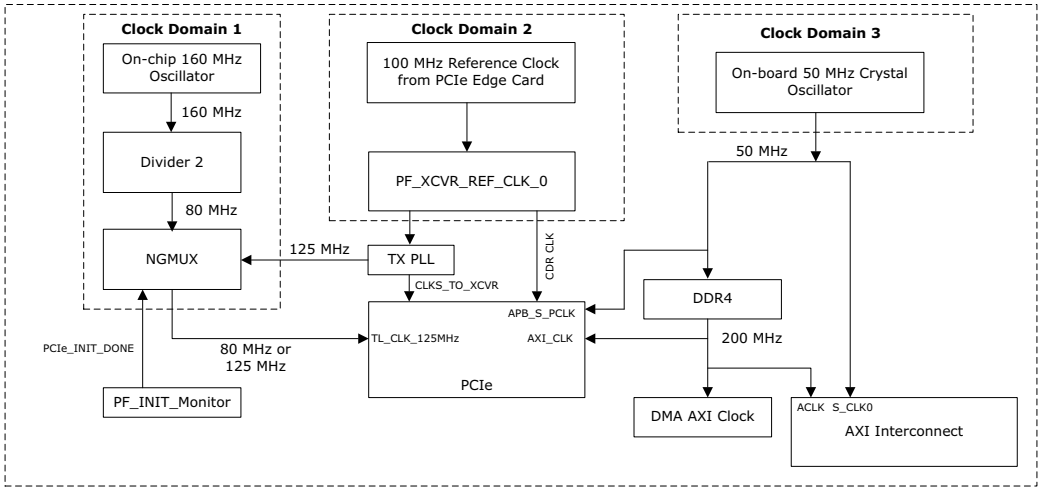


2.4 Clocking Structure

The following figure shows the clocking structure of PCIe EndPoint reference design.

- Clock Domain 1: generates PCIe TL_CLK. At power-up, it uses 80 MHz clock and switches to 125 MHz after completion of PCIe initialization using NGMUX.
- Clock Domain 2: generates CDR reference and XCVR clocks for PCIe.
- Clock Domain 3: generates 50 MHz clock for PCIe APB, DDR4 PLL reference. DDR4 subsystem generates a 200 MHz clock for fabric AXI interface logic.

Figure 25 • **Clocking Structure**



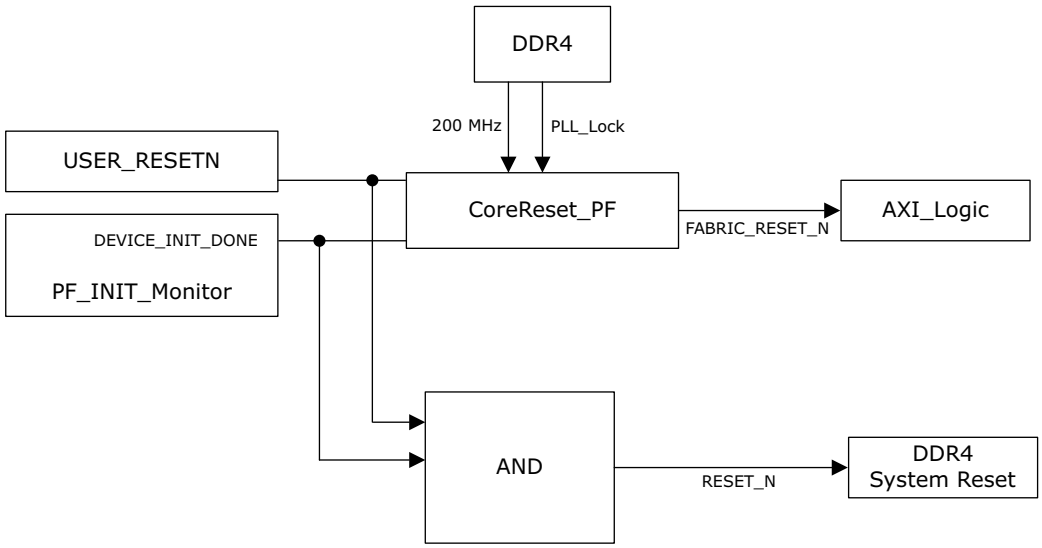
2.5 Reset Structure

The CoreReset_PF synchronizes the external USER_RESETN (**SW2** on PolarFire Splash kit) to DDR4 system clock (200 MHz) and generates the FABRIC_RESET_N, which drives the fabric AXI interface logic. CoreReset_PF uses the DEVICE_INIT_DONE signal, which is asserted when the device initialization is complete. For more information about device initialization, see *UG0725: PolarFire FPGA Device Power-Up and Resets User Guide*.

For more information on CoreReset_PF IP core, see CoreReset_PF handbook from the Libero catalog.

The DDR4 subsystem does not require a synchronization reset as it has the reset synchronization logic. The following figure shows the reset structure in the reference design.

Figure 26 • **Reset Structure**



2.6 Throughput Measurement

The fabric logic uses 32-bit counters to count the number of clock cycles in each DMA transfer. The host PC application starts these counters while initiating the DMA transfers, and the fabric logic stops these counters at the end of the DMA transfer. The DMA Engine interrupts the host PC at the end of the DMA transfer and the host PC application reads the counters to calculate throughput as follows:

Throughput = Transfer Size (Byte) × Clock Frequency/Number of clock cycles taken for a transfer

The throughput includes all of the overhead of the AXI, PCIe, and DMA controller transactions.

2.7 Simulating the Design

Before you start:

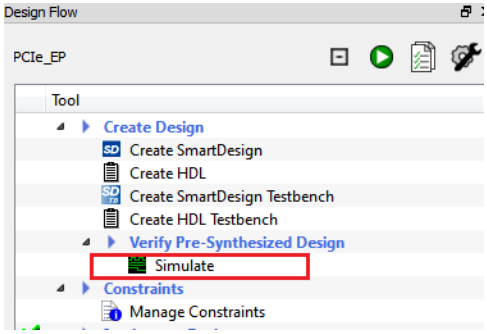
1. Start Libero SoC PolarFire, and in the **Project** menu, click **Open Project**.
2. Browse the Libero Project > PCIe_EP_Demo Libero project folder and open the `PCIe_EP_Demo.prjx` file. The PolarFire PCIe EndPoint project opens.
3. Open the **Design Hierarchy** window and double-click the **PCIe_EP_Demo** component. The SmartDesign page opens on the right pane and displays the high-level design. You can view the design blocks and IP cores instantiated for the PCIe EndPoint interface design.
4. Download the `PF_XCVR_REF_CLK`, `PF_TX_PLL`, `PF_CCC`, `PF_PCIE`, `CoreAXI4Interconnect`, `CoreAXI4DMAController`, `DDR4`, `CoreAHBLite`, `CoreAPB`, `CoreAXItoAHBL`, `CoreAHBLtoAPB`, `CoreUART`, and `PolarFire SRAM IP` cores under **Libero SoC PolarFire > Catalog**.

The PCIe BFM performs 1 KB DMA operations between PCIe and DDR4/LSRAM memories by initiating AXI burst transactions. The PCIe BFM simulation model replaces the entire PCIe EndPoint interface with a simple BFM that can send write transactions and read transactions over the AXI interface. These transactions are driven by a script file (.bfm) and allow easy simulation of the FPGA design connected to a PCIe interface. For more information about BFM commands, see *UG0685: PolarFire FPGA PCI Express User Guide*. The micron DDR4 memory models are instantiated in the testbench for simulating DDR4 memory controllers.

Note: In the Design Flow tab, system verilog is selected, as the memory models from Micron are in the system verilog.

In the **Project settings** > **Design Flow** tab, double-click **Simulate** under **Verify Pre-Synthesized Design** to simulate the design, as shown in the following figure. The ModelSim tool takes about 10 to 15 minutes to complete the simulation.

Figure 27 • **Simulating the Design**



2.7.1 Simulation Flow

The following steps describe the PCIe BFM simulation flow:

1. At the start, the `NSYSREST` signal, reset all the components.
2. DDR4 memory controllers initializes the DDR4 memories and release the CTRLR_READY.
3. The PCIe BFM starts executing the BFM script `PCIex4_PCIex4_0_PF_PCIE_PCIE_1_user.bfm`.
4. The PCIe EndPoint AXI4 master interface initiates write and read burst transactions to SRAM_AXI_0, DDR4 through CoreAXI4Interconnect as per the .bfm script.
5. After 13 µs, the simulation completes. **PCIE1 BFM Simulation Complete – 272 Instructions – NO ERRORS** message is highlighted, as shown in Figure 28, page 26.

The ModelSim transcript window displays the BFM commands execution messages, as shown in the following figure. For more information about BFM commands, see the *SmartFusion2 FPGA Microcontroller Subsystem BFM Simulation User Guide*.

Figure 28 • **Simulation Transcript Window**

```
# SFM: Data Read 30000080 0000002100000022 at 12526.250000ns
# SFM: Data Read 30000088 0000002300000024 at 12531.250000ns
# SFM: Data Read 30000090 0000002500000026 at 12536.250000ns
# SFM: Data Read 30000098 0000002700000028 at 12541.250000ns
# SFM: Data Read 300000a0 000000290000002a at 12546.250000ns
# SFM: Data Read 300000a8 0000002b0000002c at 12551.250000ns
# SFM: Data Read 300000b0 0000002d0000002e at 12556.250000ns
# SFM: Data Read 300000b8 0000002f00000030 at 12561.250000ns
# SFM: Data Read 300000c0 0000003100000032 at 12566.250000ns
# SFM: Data Read 300000c8 0000003300000034 at 12571.250000ns
# SFM: Data Read 300000d0 0000003500000036 at 12576.250000ns
# SFM: Data Read 300000d8 0000003700000038 at 12581.250000ns
# SFM: Data Read 300000e0 000000390000003a at 12586.250000ns
# ------------------------------------------------------------------------
# ----DMA TRANSFER DONE(FROM FABRIC ADDRESS SPACE TO PCIe ADDRESS SPACE)----
# ------------------------------------------------------------------------
# SFM: Data Read 300000e8 0000003b0000003c at 12591.250000ns
# BFM:203:wait 1  starting at 12596 ns
# SFM: Data Read 300000f0 0000003d0000003e at 12596.250000ns
# BFM: ****************************End of PCIe BFM Simulation*********************************
# BFM:206:return
# SFM: Data Read 300000f8 0000003f00000040 at 12601.250000ns
################################################################################
#
# PCIE1 BFM Simulation Complete - 272 Instructions - NO ERRORS
#
################################################################################
#
```

The following figure shows the actual waveform window showing the sequence of data being written and read using the BFM.

Figure 29 • **Simulation Waveform Window**

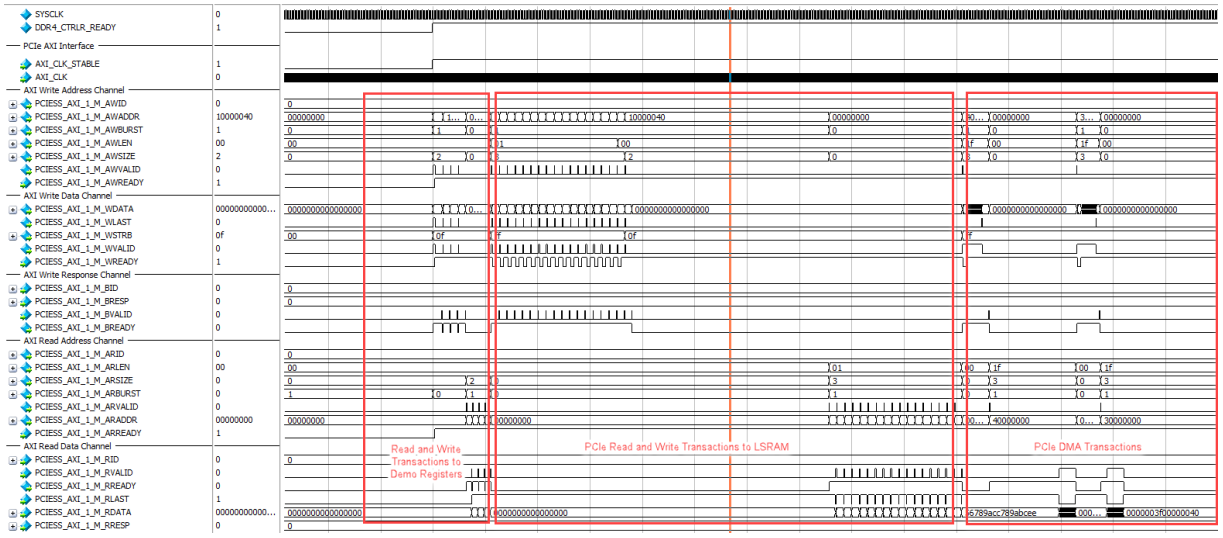
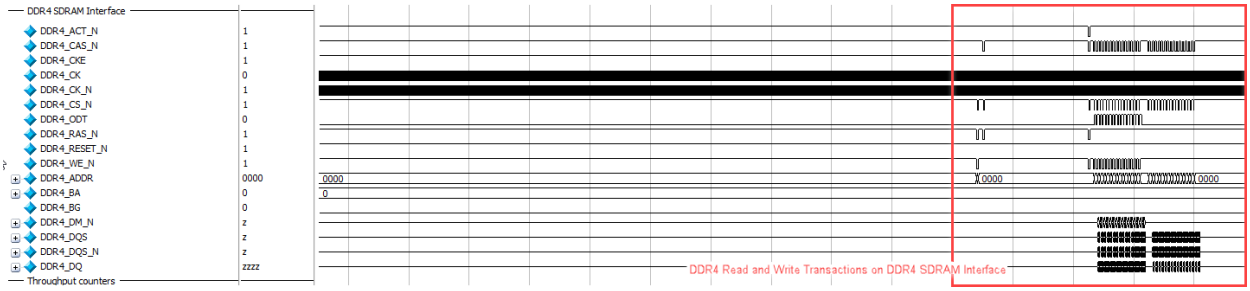


Figure 30 • **Simulation Waveform Window**



3 Libero Design Flow

The Libero design flow involves the following steps:

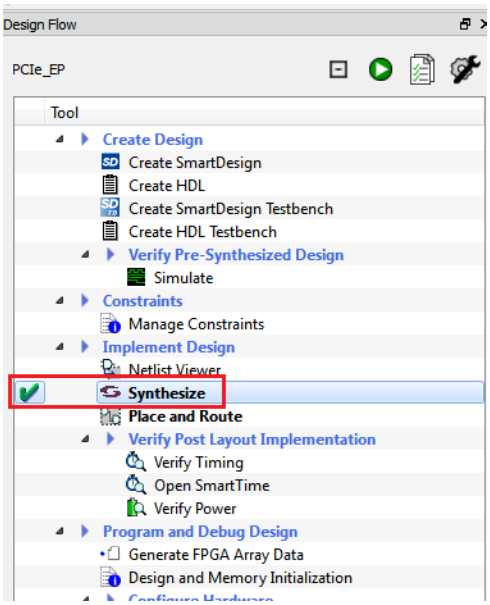
- Synthesize
- Place and route
- Verify timing
- Generate Bitstream
- Run PROGRAM Action

3.1 Synthesize

Go to the **Design Flow** window and double-click **Synthesize**.

When the synthesis is successful, a green tick mark appears as shown in the following figure.

Figure 31 • **Synthesize**



3.1.1 Resource Utilization

The following table lists the resource utilization of the PCIe Endpoint design. These values may vary slightly for different Libero runs, settings, and seed values.

Table 2 • **Resource Utilization**

Type	Used	Total	Percentage
4LUT	29761	299544	9.94
DFF	21840	299544	7.29
I/O Register	0	732	0.00
User I/O	75	244	30.74
– Single-ended I/O	69	244	28.28
– Differential I/O Pairs	3	122	2.46
µSRAM	168	2772	6.06
LSRAM	27	952	2.84

Table 2 • Resource Utilization *(continued)*

Type	Used	Total	Percentage
Math	0	924	0.00
H-Clip Global	8	48	16.67
PLL	1	8	12.50
DLL	1	8	12.50
CRN_INT	1	24	4.17
INIT	1	1	100.00
DRI	1	1	100.00
OSC_RC160MHZ	1	1	100.00
Transceiver Lanes	4	8	50.00
Transceiver PCIe	1	2	50.00
TX_PLL	1	11	9.09
LINK	1	10	10.00
XCVR_REF_CLK	1	6	16.67
PCIE_COMMON	1	1	100.00
ICB_CLKDIV	1	24	4.17
ICB_CLKINT	2	72	2.78
NGMUX	1	12	8.33
ICB_INT	1	12	8.33

3.2 Place and Route

To place and route the design, the TX_PLL, XCVR_REF_CLK, and DDR4 need to be constrained using the **I/O Editor** as shown in the following figures.

Figure 32 • I/O Editor—XCVR View

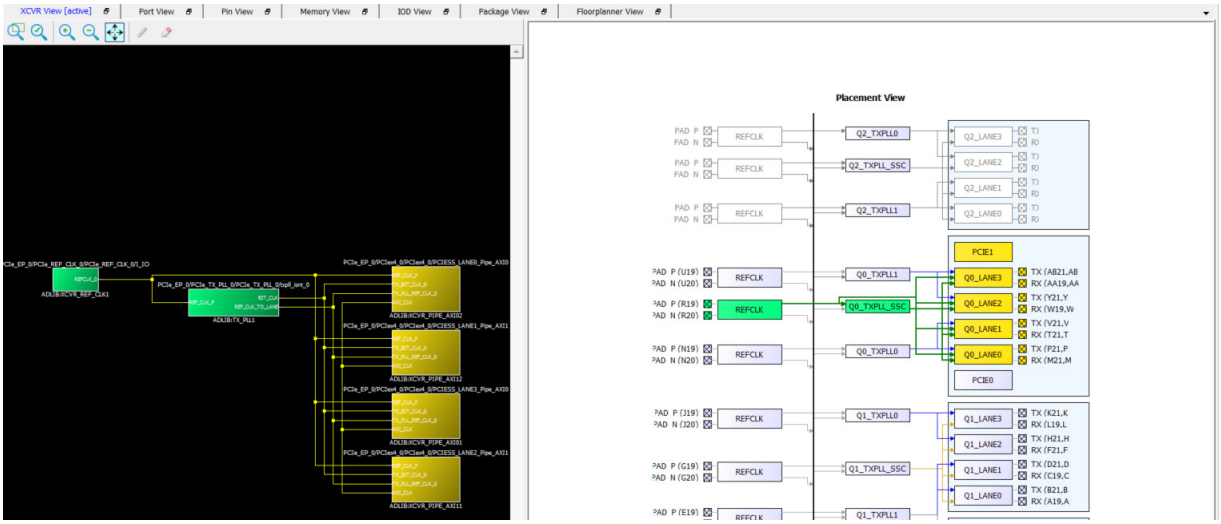
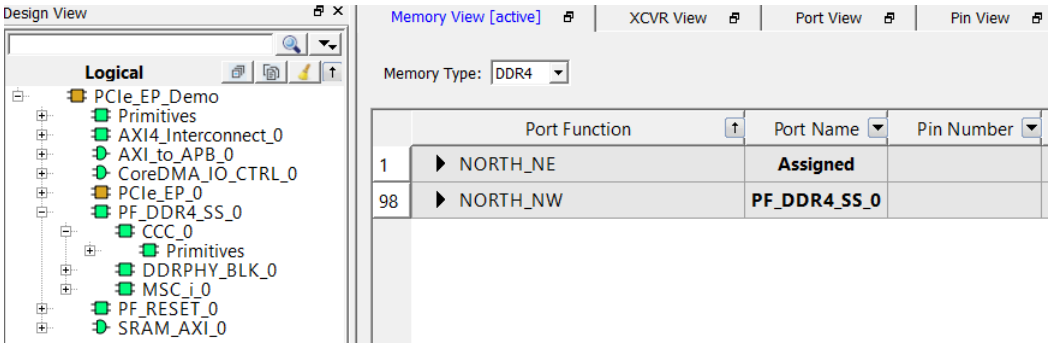
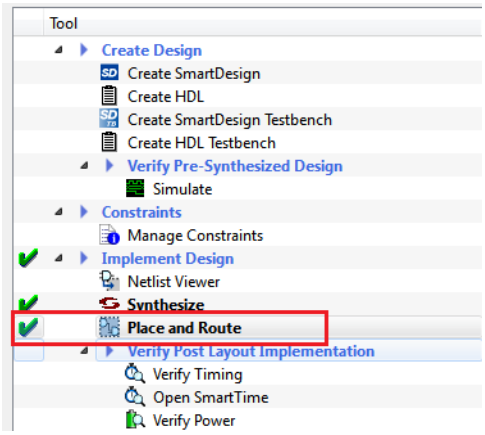


Figure 33 • **I/O Editor—DDR4 Memory View**



Go to the **Design Flow** window and double-click **Place and Route**. When place and route is successful, a green tick mark appears as shown in the following figure.

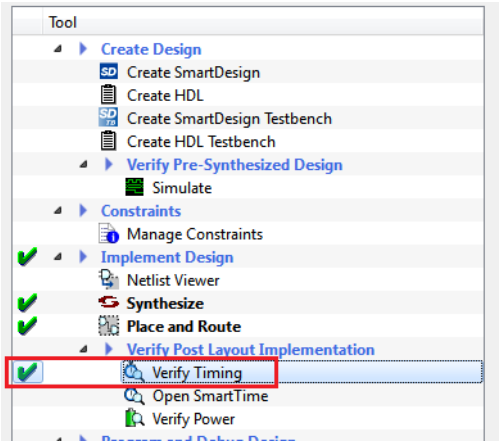
Figure 34 • **Place and Route**



3.3 Verify Timing

Go to the **Design Flow** window and double-click **Verify Timing**. When the design successfully meets the timing requirements, a green tick mark appears as shown in the following figure.

Figure 35 • **Design Flow**



3.4 Generate Bitstream

To generate the bitstream:

1. Double-click **Generate Bitstream** from the **Design Flow** tab. When the bitstream is successfully generated, a green tick mark appears as shown in Figure 37, page 30.
2. Right-click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

3.5 Run PROGRAM Action

After generating the bitstream, the PolarFire device must be programmed. Follow these steps to program the PolarFire device:

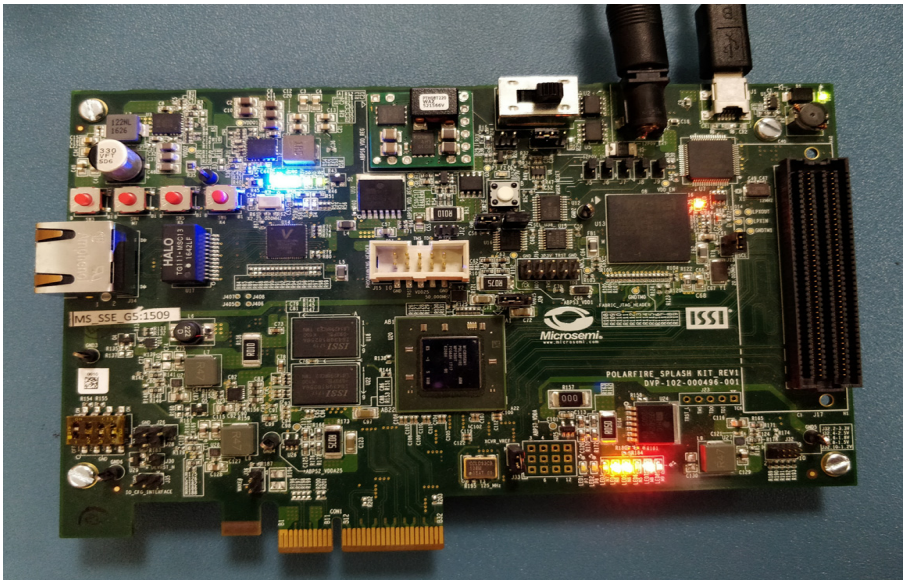
1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Table 3 • **Jumper Settings**

Jumper	Description
J5, J6, J7, J8, J9	Short pin 2 and 3 for programming the PolarFire FPGA through FTDI
J11	Short pin 1 and 2 for programming through the FTDI chip
J10	Short pin 1 and 2 for programming through the FTDI SPI
J4	Short pin 1 and 2 for manual power switching using SW1
J3	Open pin 1 and 2 for 1.0 V

2. Connect the power supply cable to the **J2** connector on the board.
3. Connect the USB cable from the Host PC to **J1** (FTDI port) on the board.
4. Power on the board using the **SW1** slide switch.

Figure 36 • **Board Setup**



5. Double-click **Run PROGRAM Action** from the **Libero > Design Flow** tab.

When the device is programmed successfully, a green tick mark appears as shown in the following figure. See Running the Demo, page 32 to run the PCIe EndPoint demo.

Figure 37 • **Programming the Device**

4 Programming the Device Using FlashPro

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

```
mpf_dg0808_liberosocpolarfirev2p2_df\ProgrammingFile
```

To program the PolarFire device using FlashPro, complete the following steps:

1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Note: The power supply switch must be switched off while making the jumper connections.

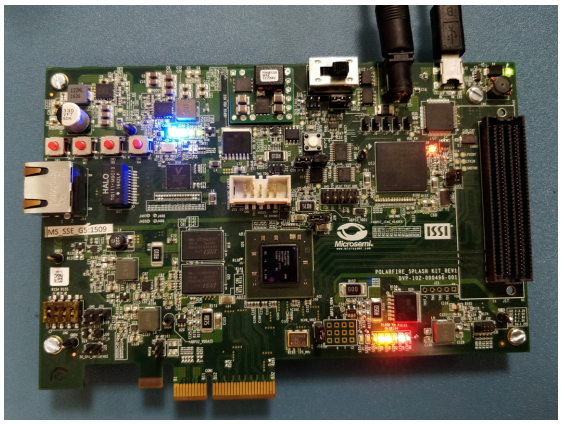
Table 4 • **Jumper Settings**

Jumper	Description
J5, J6, J7, J8, J9	Short pin 2 and 3 for programming the PolarFire FPGA through FTDI
J11	Short pin 1 and 2 for programming through the FTDI chip
J10	Short pin 1 and 2 for programming through the FTDI SPI
J4	Short pin 1 and 2 for manual power switching using SW1
J3	Open pin 1 and 2 for 1.0 V

2. Connect the power supply cable to the **J2** connector on the board.
3. Connect the USB cable from the Host PC to the **J1** (FTDI port) on the board.
4. Power on the board using the **SW1** slide switch.

The following figure shows the board setup of the PolarFire Splash Kit.

Figure 38 • **Board Setup**



5. On the host PC, launch the FlashPro software.
6. Click **New Project** to create a new project.
   In the New Project window, enter a project name.
7. Click **Browse** and navigate to the location where you want to save the project.
8. Select **Single device** as the programming mode and click **OK** to save the project.
9. Click **Configure Device**.
10. Click **Browse**, and navigate to the location where the PCIe_EP_Demo.stp file is located and select the file. The default location is:
    <download_folder>\mpf_dg0808_liberosocpolarfirev2p2_df\ProgrammingFile
11. Click **Open**. The required programming file is selected and ready to be programmed in the device.
12. Click **PROGRAM** to program the device.

When the device is programmed successfully, a **Run PASSED** status is displayed. See Running the Demo, page 32 to run the PCIe EndPoint demo.

5 Running the Demo

This section describes how to install and use the PCIe Demo application. The PolarFire PCIe demo application is a simple graphic user interface (GUI) that runs on the host PC to communicate with the PolarFire PCIe EndPoint device. It provides PCIe link status, driver information, and demo controls. The PolarFire PCIe demo application invokes the PCIe driver installed on the host PC and provides commands to the driver according to the selection made.

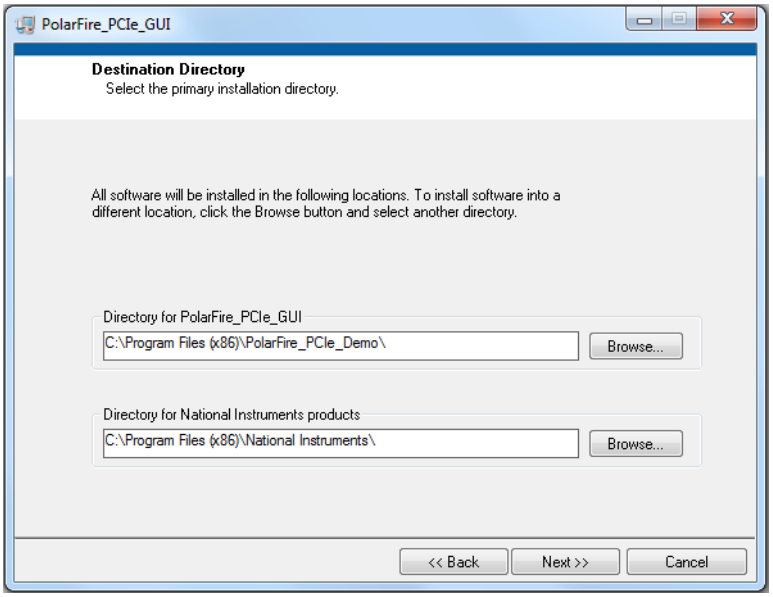
This section also describes how to connect the kit to the Host PC PCIe Slot. If the host PC PCIe slot is not available, the DMA between DDR4 and LSRAM can be exercised through UART IF.

5.1 Installing PCIe Demo Application

To install the PolarFire PCIe Demo application:

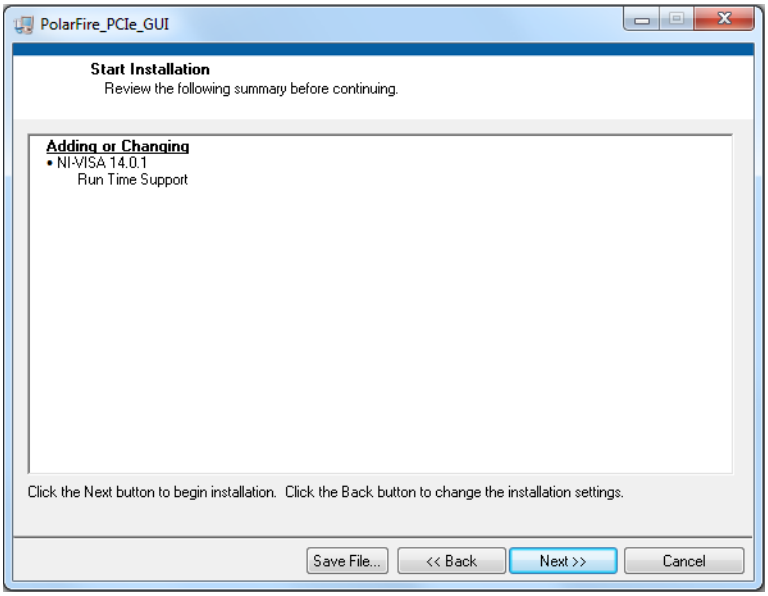
1. Install the GUI_Installer (setup.exe) from the following design files folder: mpf_dg0808_liberosocpolarfirev2p2_df\GUI_Installer.
2. Double-click the setup.exe in the provided GUI installation (GUI_Installer\setup.exe).
3. Apply default options as shown in the following figure.

Figure 39 • **Installing PCIe Demo Application**



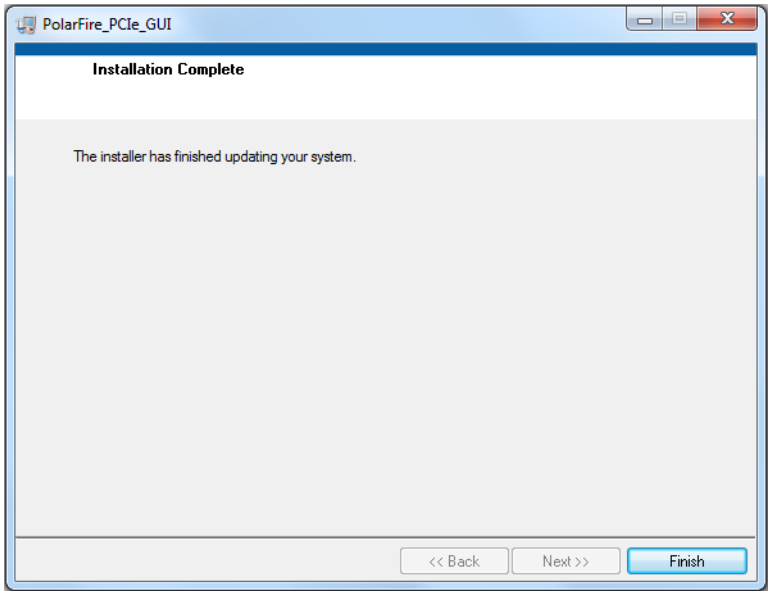
4. Click **Next** to start the installation.

Figure 40 • **PCIe Demo Application Installation Steps**



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

Figure 41 • **Successful Installation of PCIe Demo Application**



5.2 Running the Demo Through PCIe

This section shows how to connect the board to host PC PCIe slot, installing the PCIe drivers and running the demo application.

5.2.1 Connecting the Board to the Host PC PCIe Slot

1. After successful programming, power OFF the PolarFire Splash board and shut down the host PC.
2. Connect the CON3 - PCIe Edge connector of the PolarFire Splash board to the host PC's PCIe slot through the PCI Edge card ribbon cable.
   This demo is designed to work with any PCIe Gen 2 compliant slot. If the host PC does not support Gen 2 compliant slot, the demo switches to Gen 1 mode.

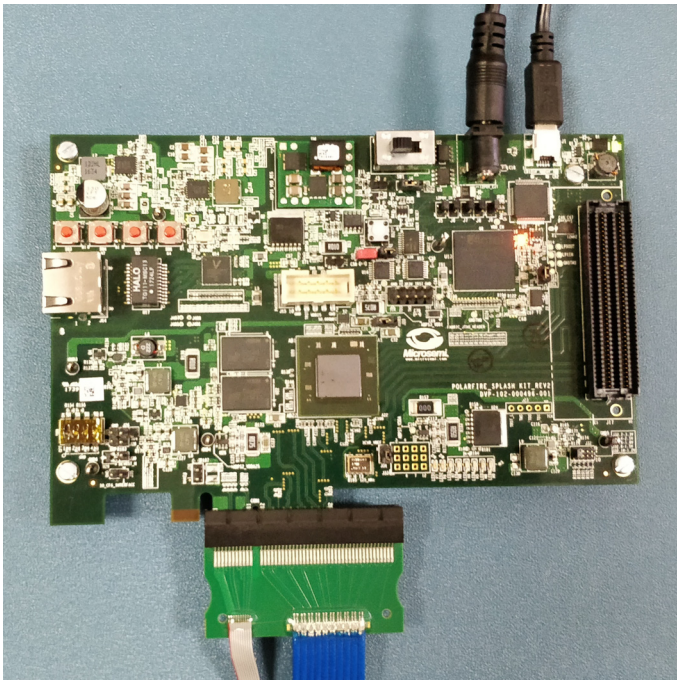
Note: Power OFF the host PC while inserting the PCIe Edge connector. If it is not powered OFF, the PCIe device detection and the selection of Gen1 or Gen2 mode may fail. The device detection and selection depend on the host PC PCIe configuration.

Note: After connecting the board to the host PC, the host PC may power on without manually switching on the PC.

Note: PCIe hot reset is not supported in this version of the demo.

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

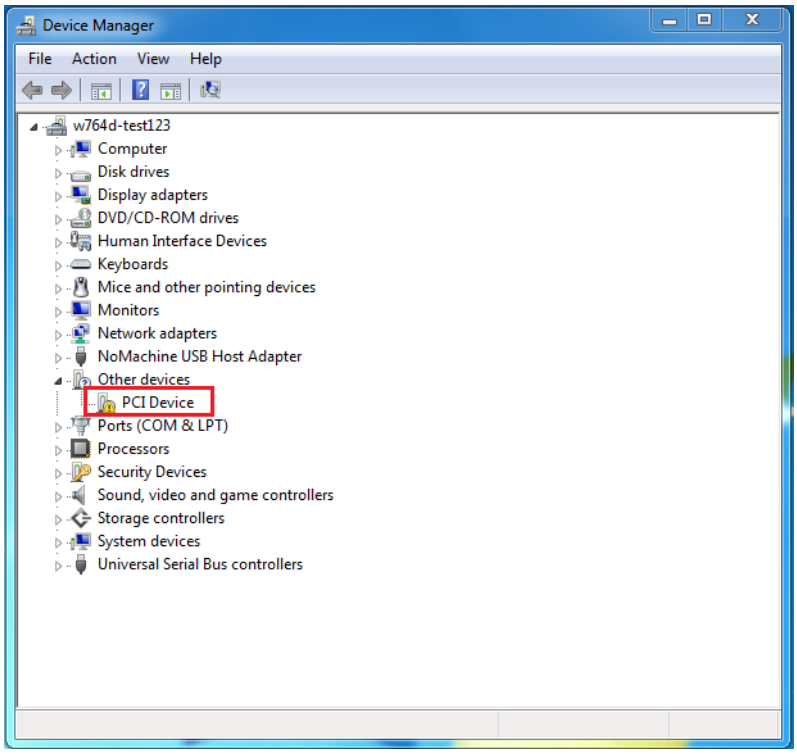
Figure 42 • **PolarFire Splash Kit Setup for Host PC**



3. Power on the power supply switch **SW1**.

4. Power on the host PC and check the **Device Manager** of the Host PC for the PCIe Device. The following figure shows the example **Device Manager** window.

Figure 43 • **Device Manager**



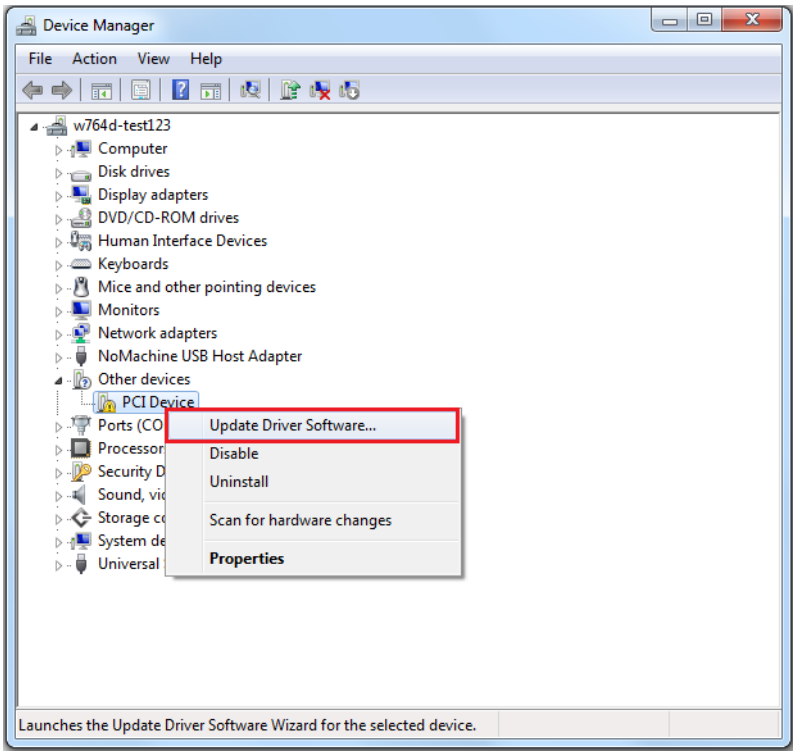
Note: If the device is still not detected, check if the BIOS version in the host PC is the latest, and if PCI is enabled in the host PC BIOS.

5.2.2 Driver Installation

Perform the following steps to install the PCIe drivers on the host PC:

1. Right-click **PCI Device** in the **Device Manager** and select **Update Driver Software...** as shown in the following figure. To install the drivers, administrative rights are required.

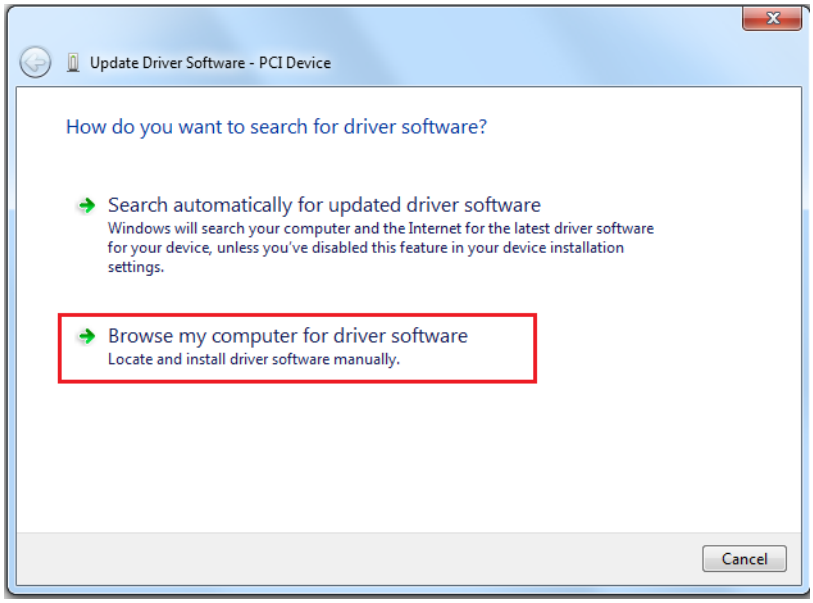
Figure 44 • **Update Driver Software**



Note: Uninstall the existing Microsemi PolarFire drivers on the host PC before proceeding to next step.

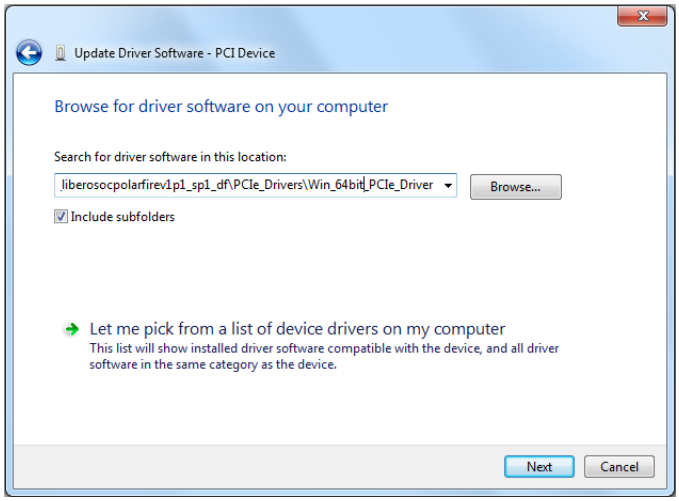
2. In the **Update Driver Software - PCIe Device** window, select **Browse my computer for driver software** as shown in the following figure.

Figure 45 • **Browse for Driver Software**



3. Browse the drivers folder: *mpf_dg0808_liberosocpolarfirev2p2_df\PCIe_Drivers\Win_64bit_PCIe_Driver* and click **Next** as shown in the following figure.

Figure 46 • **Browse for Driver Software Continued**



4. The **Windows Security** dialog box is displayed. Click **Install** as shown in the following figure. After successful driver installation, a message appears. See Figure 48, page 37.

Figure 47 • **Windows Security**

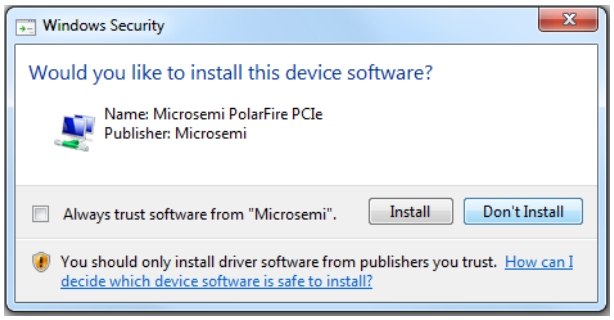
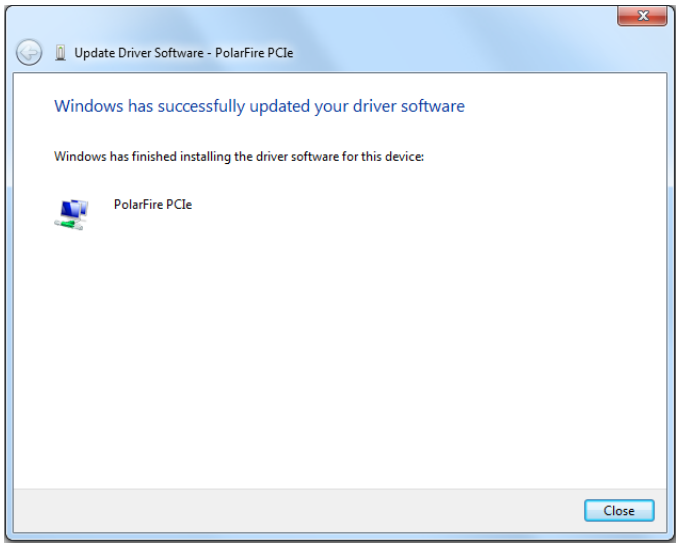


Figure 48 • **Successful Driver Installation**

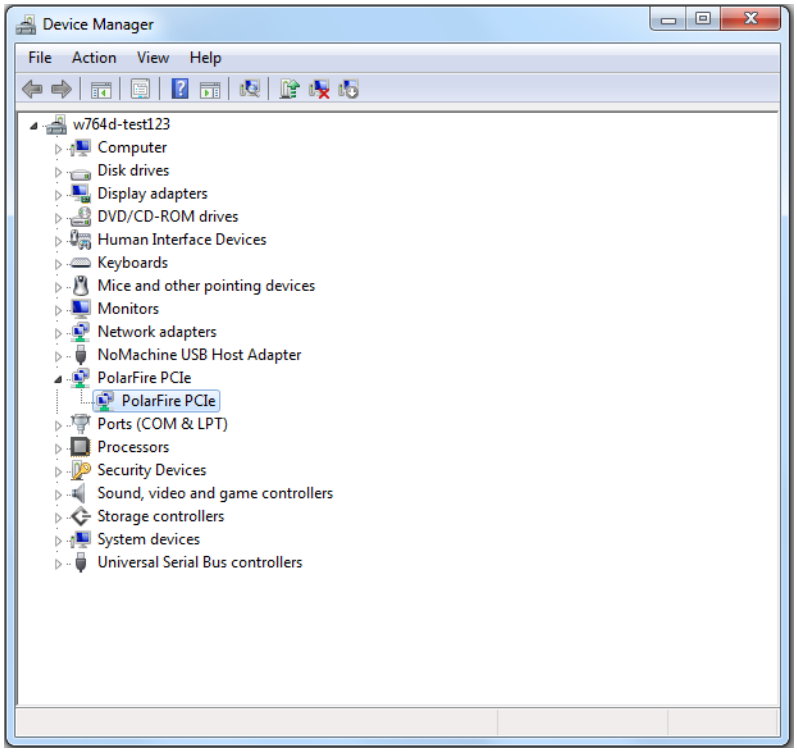


5.2.3 Running the PCIe Demo Application

The following steps describe how to run the demo design:

1. Click to expand the **PolarFire PCIe** device in the host PC **Device Manager** as shown in the following figure.

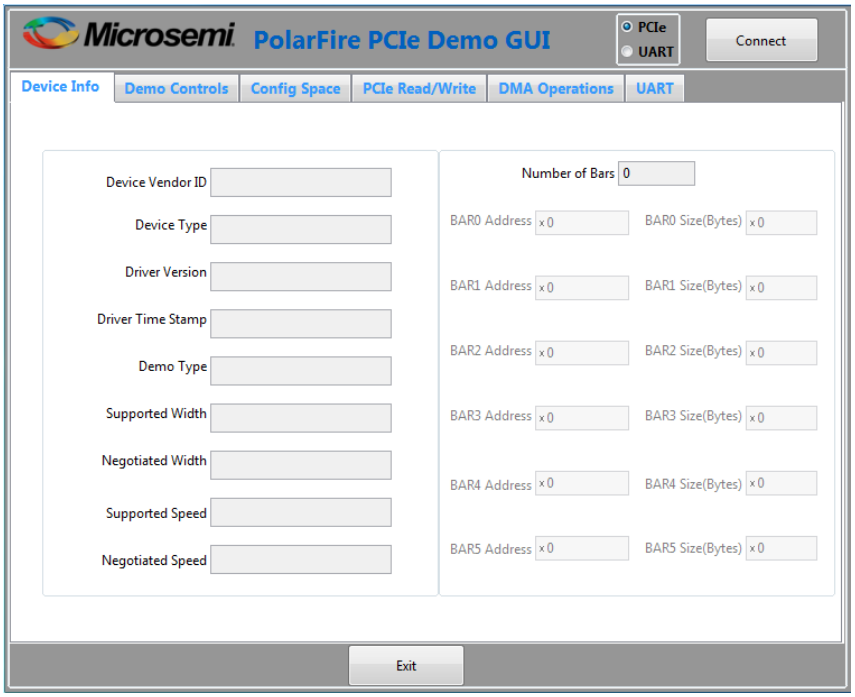
Figure 49 • **Device Manager—PCIe Device Detection**



Note: If a warning message is displayed for PolarFire PCIe driver while accessing, uninstall and re-install the driver.

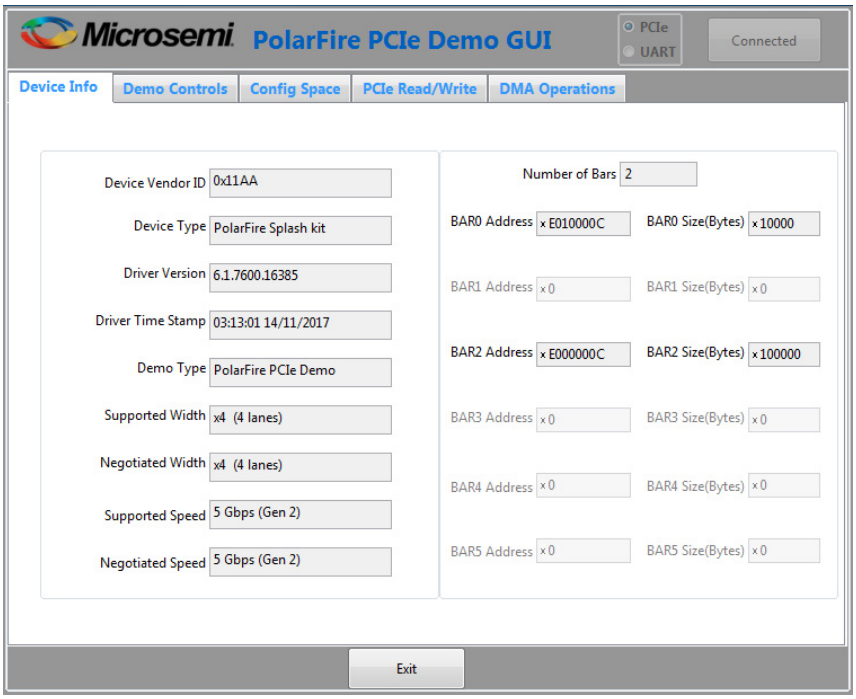
2. Go to **All Programs** > **PolarFire_PCIe_GUI** > **PolarFire_PCIe_GUI**. The **PolarFire PCIe Demo** window is displayed as shown in the following figure.

Figure 50 • **PCIe EndPoint Demo Application**



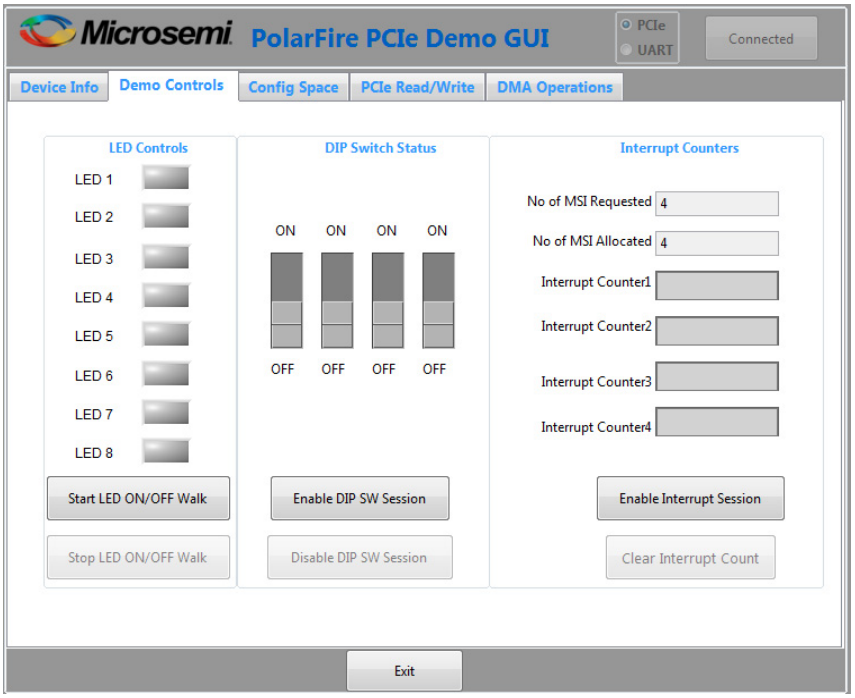
3. Click **Connect**. The application detects and displays the information related to the connected kit such as Device Vendor ID, Device Type, Driver Version, Driver Time Stamp, Demo Type, Supported Link Width, Negotiated Link Width, Supported Speed, Negotiated Speed, Number of Bars, and BAR Address as shown in the following figure.

Figure 51 • **Device Info**



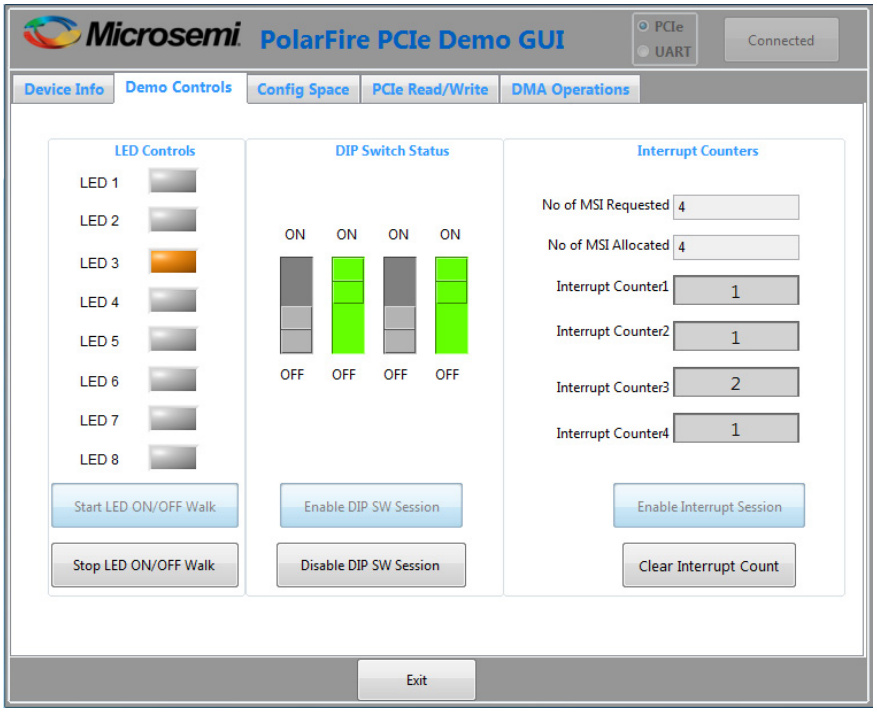
4. Click the **Demo Controls** tab to display the **LED Controls**, **DIP Switch Status**, and **Interrupt Counters** as shown in the following figure.

Figure 52 • **Demo Controls**



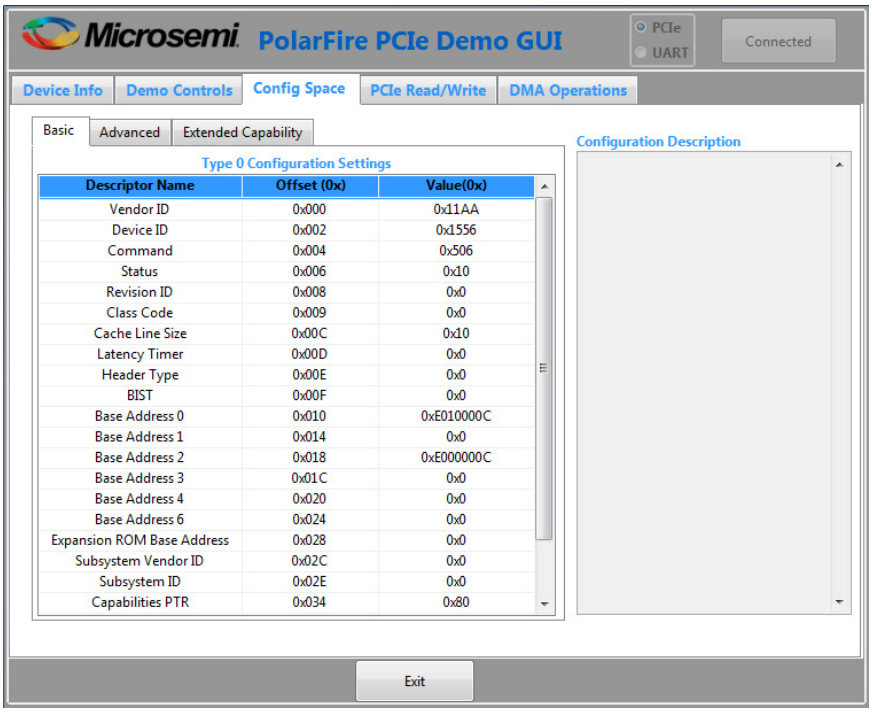
5. Click **Start LED ON/OFF Walk**, **Enable DIP SW Session**, and **Enable Interrupt Session** to view the controlling LEDs (observe LED1 to LED8 on the PolarFire Splash Kit), getting the DIP switch (ON/OFF the DIP1 to DIP4 on the PolarFire Splash Kit) status, and monitoring the interrupts (press SW3 to SW6 on the PolarFire Splash Kit to generate interrupt) simultaneously as shown in the following figure.

Figure 53 • **Demo Controls—Continued**



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

Figure 54 • **Configuration Space**



7. Click the **PCIe Read/Write** tab to perform read and write operations to DDR/LSRAM using BAR2 space.
8. Click **Read** to read the 4 KB memory mapped to BAR2 space for DDR and LSRAM as shown in the following figure.

Figure 55 • **PCIe BAR2 Memory Access—LSRAM**

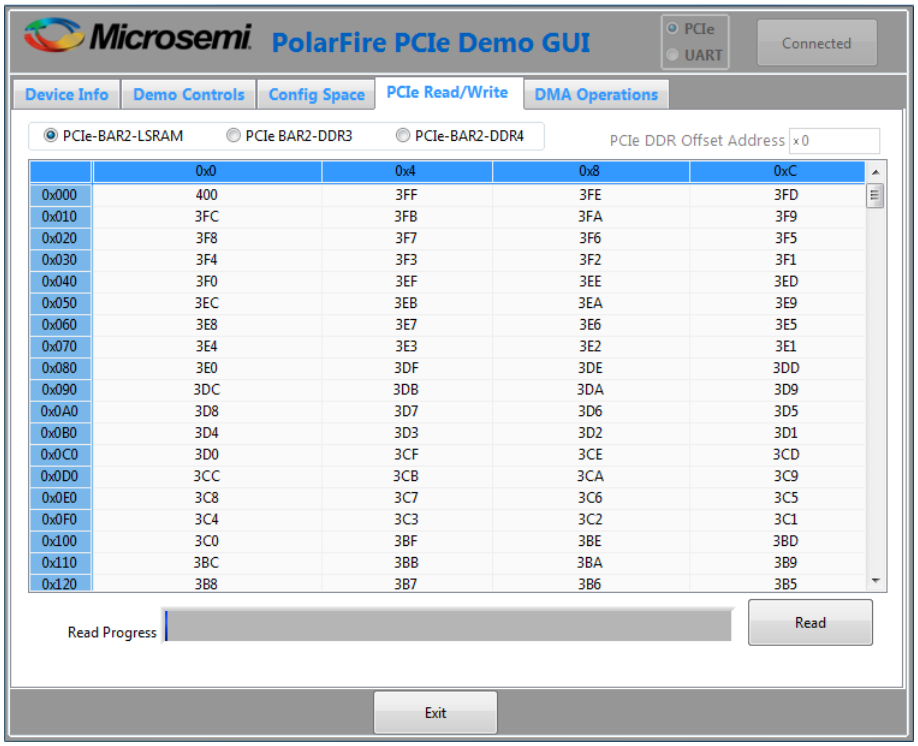
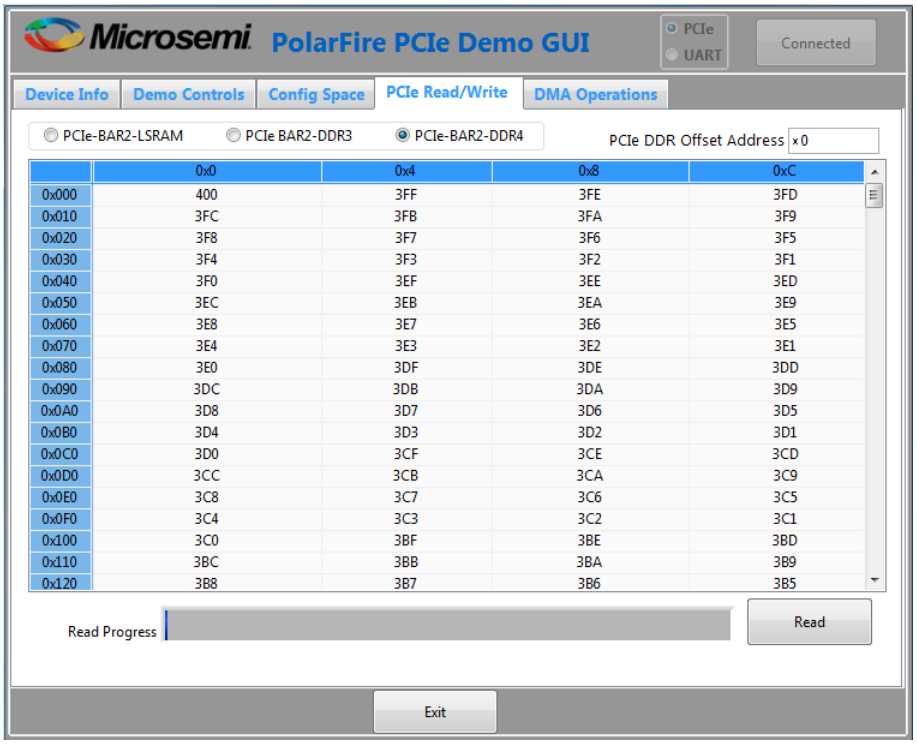


Figure 56 • **PCIe BAR2 Memory Access—DDR4**



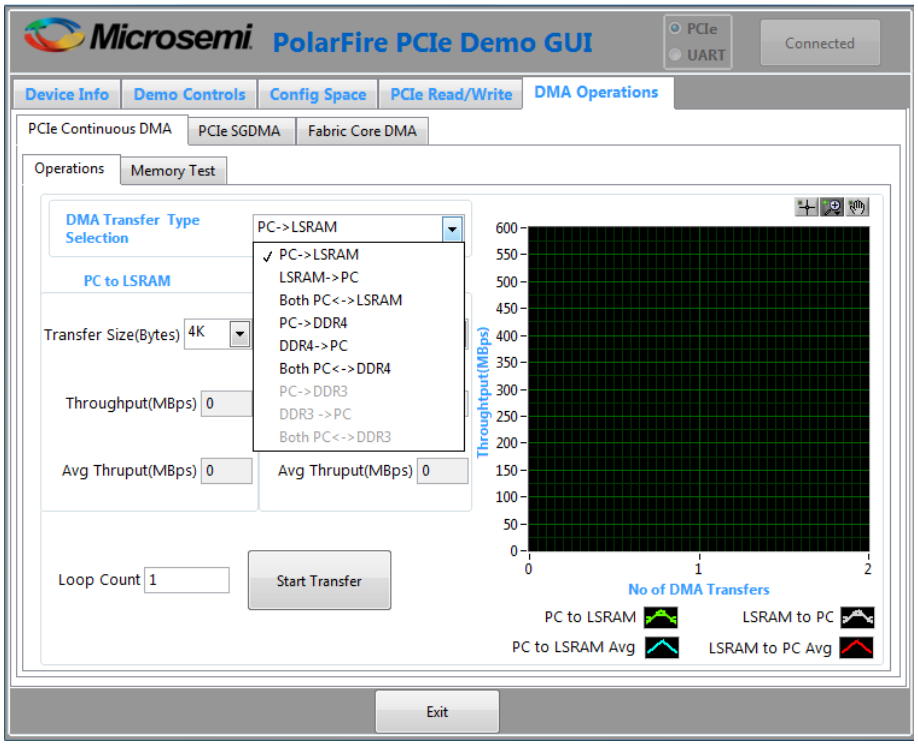
9. Click the **DMA Operations** tab for different DMA operations such as DDR and LSRAM.

5.2.3.1 Continuous DMA—Operations

The following instructions describe running DMA operations between PC and DDR4, PC and LSRAM:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
   - **PC->DDR4**—to transfer the data from host PC to PolarFire DDR4 memory
   - **DDR4->PC**—to transfer the data from PolarFire DDR4 memory to host PC
   - **Both PC<->DDR4**—to transfer the data from host PC to and from PolarFire DDR4 memory
   - **PC->LSRAM**—to transfer the data from host PC to PolarFire LSRAM memory
   - **LSRAM->PC**—to transfer the data from PolarFire LSRAM memory to host PC
   - **Both PC<->LSRAM**—to transfer the data from host PC to and from PolarFire LSRAM memory
2. Select **Transfer Size** (4KB to 64KB) from the drop-down list. Maximum contiguous DMA size is 64KB because the host PC may not have contiguous memory of more than 64 KB. For DMA operations that require more than 64 KB, use SGDMA.
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 Continuous DMA—Operations.

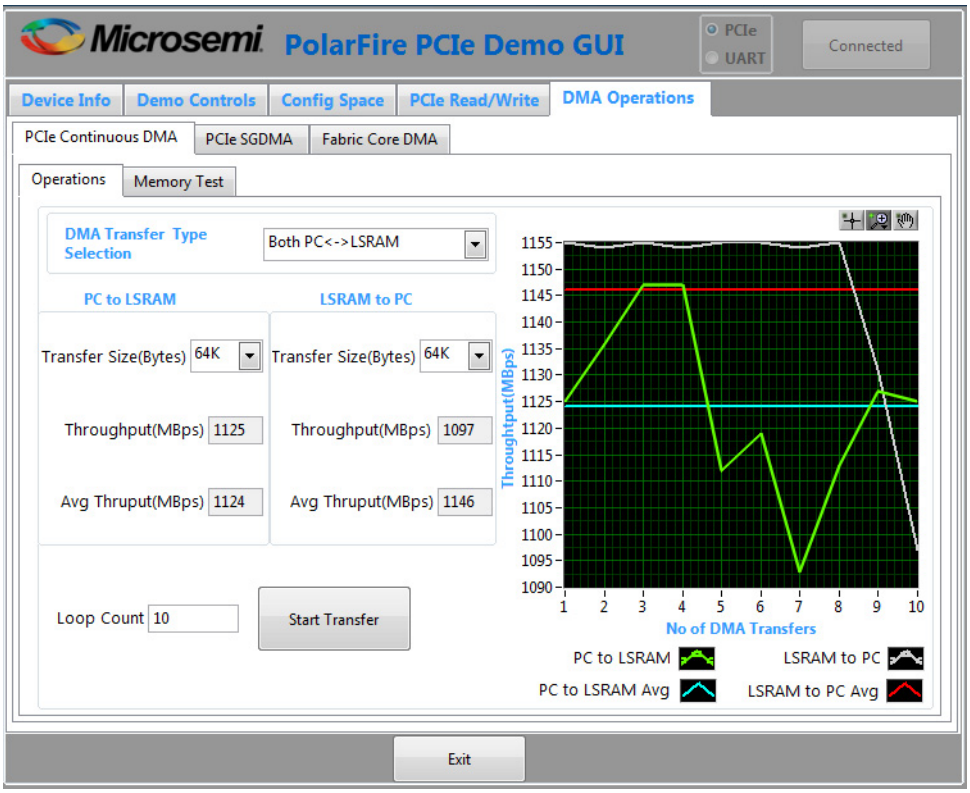
Figure 57 • **Continuous DMA—Operations**



Note: The AXI LSRAM in the design is configured for 4KB. This 4KB is over written if more than 4KB of DMA operation is performed on LSRAM. This option is provided to exercise the throughputs with larger DMA size.

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

Figure 58 • **Continuous DMA Operations with DMA Transfer Type Selection as Both PC and LSRAM**



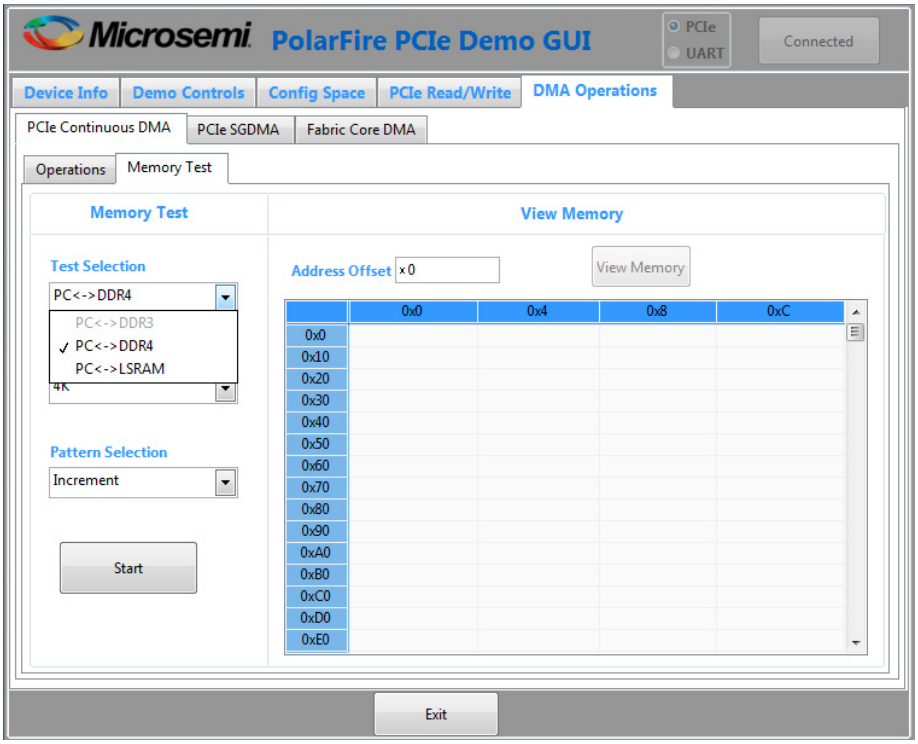
5.2.3.2 Continuous DMA—Memory Test

The following instructions describe running **Memory Test** between PC and DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
   - **PC<->DDR4**—to transfer the data from host PC to and from PolarFire DDR4 memory
   - **PC<->LSRAM**—to transfer the data from host PC to and from PolarFire LSRAM memory
2. Select **Transfer Size** (4KB to 64KB) from the drop-down list.
3. Select **Pattern Selection** from the drop-down list—Increment, Decrement, Random, Fill with Zeros, Fill with Ones, Fill with all A's, and Fill with all 5's.

The following figure shows Continuous DMA—Memory Test tab.

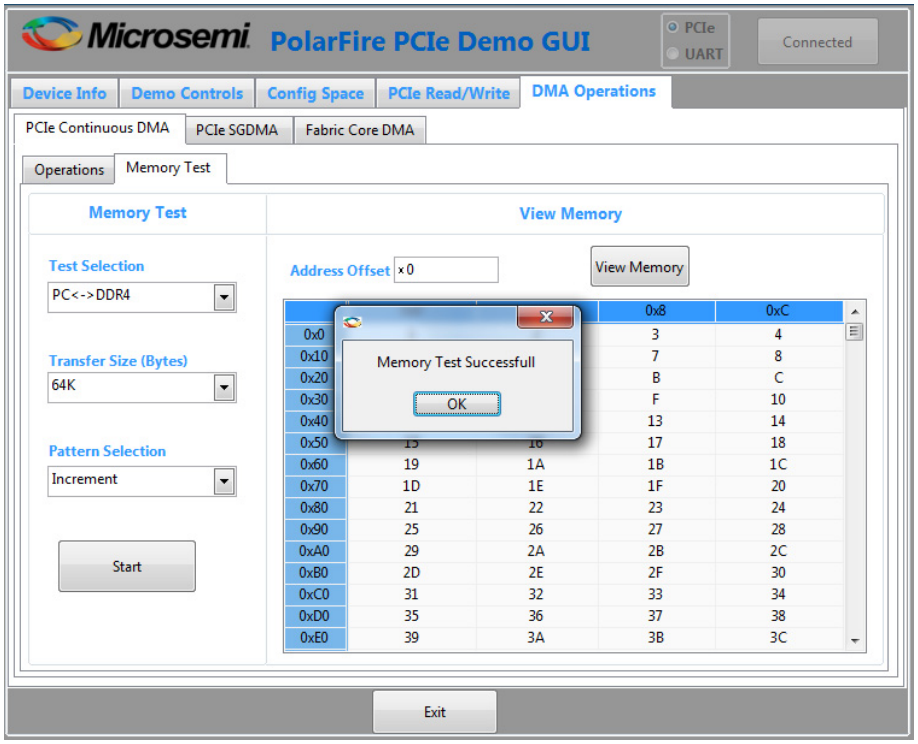
Figure 59 • **DMA Transfer Type Selection—Continuous Memory Test**



4. Click **Start**. GUI performs the following task:
   - The host PC creates a buffer and initializes the memory
   - Initiates the PC to DDR DMA
   - Erases the PC buffer
   - Initializes the DDR to PC DMA
   - Compares the memory against expected memory

Memory Test Successful window appears, as shown in the following figure.

Figure 60 • **Continuous DMA Memory Test—Memory Test Successful**



Note: If memory test fails, the GUI displays the first failed memory location.

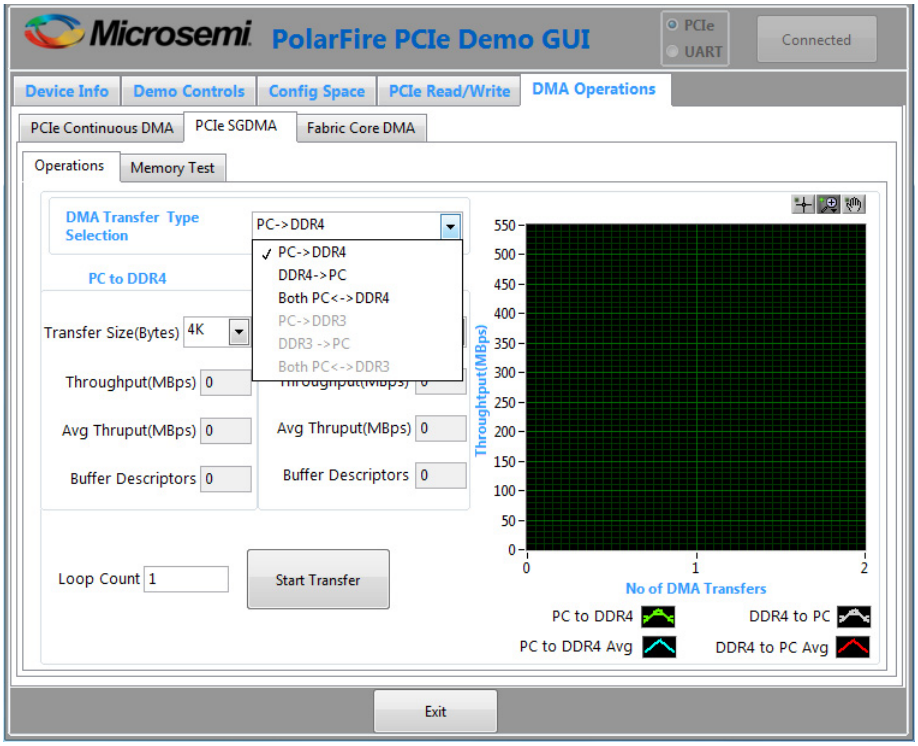
Note: Change the **Offset Address** and click **View Memory** to read the RAM memory content.

5.2.3.3 SGDMA—Operations

The following instructions describe running SGDMA operations between PC and DDR4:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
   - **PC -> DDR4**—to transfer the data from host PC to PolarFire DDR4 memory
   - **DDR4-> PC**—to transfer the data from PolarFire DDR4 memory to host PC
   - **Both PC <->DDR4**—to transfer the data from host PC to and from PolarFire DDR4 memory
2. Select **Transfer Size** (4KB to 64KB) from the drop-down list.
3. Enter the **Loop Count** in the box. The **Buffer Descriptors** show the number of descriptors created by the host driver for each SGDMA operation.
4. Click **Start Transfer**. After a successful DMA operation, the GUI displays the throughput and average throughput in MBps. The following figure shows SGDMA—Operations.

Figure 61 • **SGDMA—Operations**



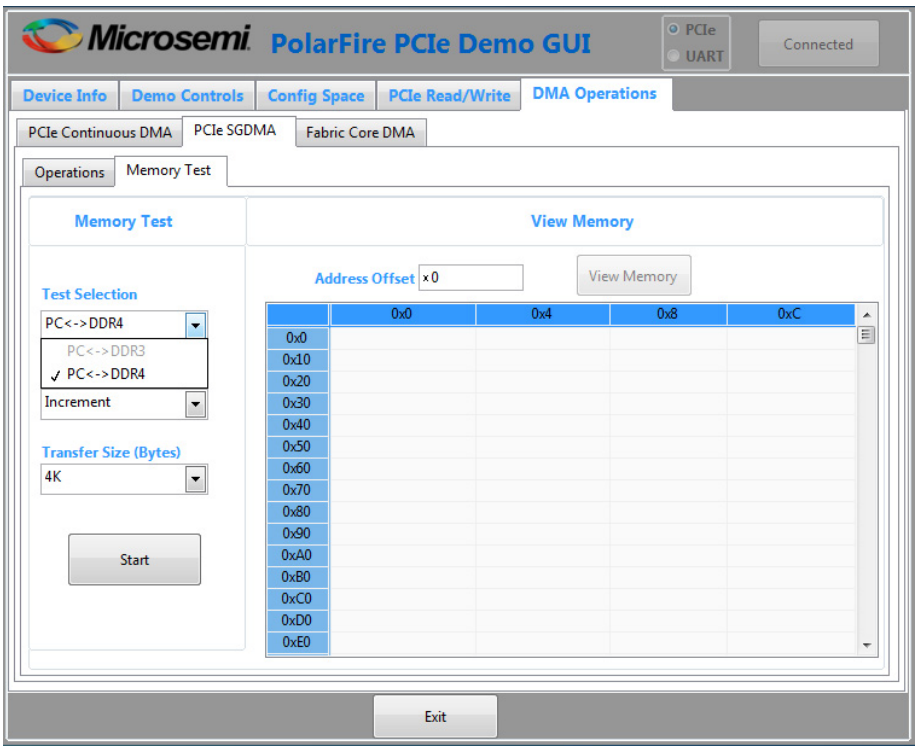
5.2.3.4 SGDMA—Memory Test

The following instructions describe running **Memory Test** between PC and DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
   - **PC<->DDR4**—to transfer the data from host PC to and from PolarFire DDR4 memory
2. Select **Transfer Size** (4KB to 1MB) from the drop-down list.
3. Select **Pattern Selection** from the drop-down list—Increment, Decrement, Random, Fill with Zeros, Fill with Ones, Fill with all A's, and Fill with all 5's.

The following figure shows SGDMA—Memory Test tab.

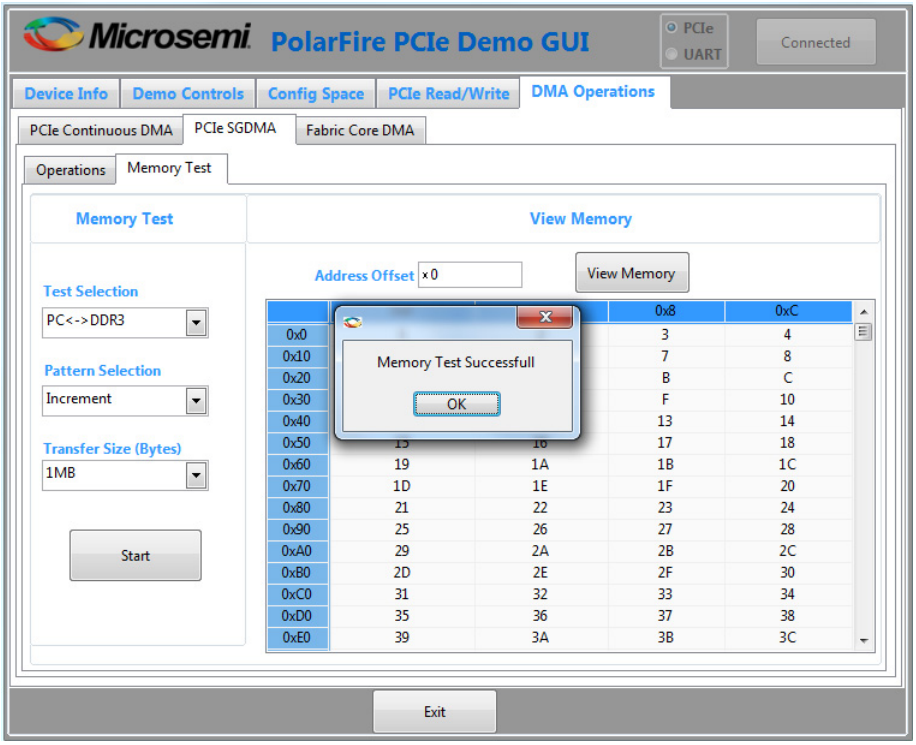
Figure 62 • **SGDMA—Memory Test**



4. Click **Start**. GUI performs the following task:
   - The host PC creates a buffer and initializes the memory
   - Initiates the PC to DDR DMA
   - Erases the PC buffer
   - Initializes the DDR to PC DMA
   - Compares the memory against expected memory

Memory Test Successful window appears, as shown in the following figure.

Figure 63 • **SGDMA Memory Test—Memory Test Successful**



Note: Change the **Offset Address** and click **View Memory** to read the RAM memory content.

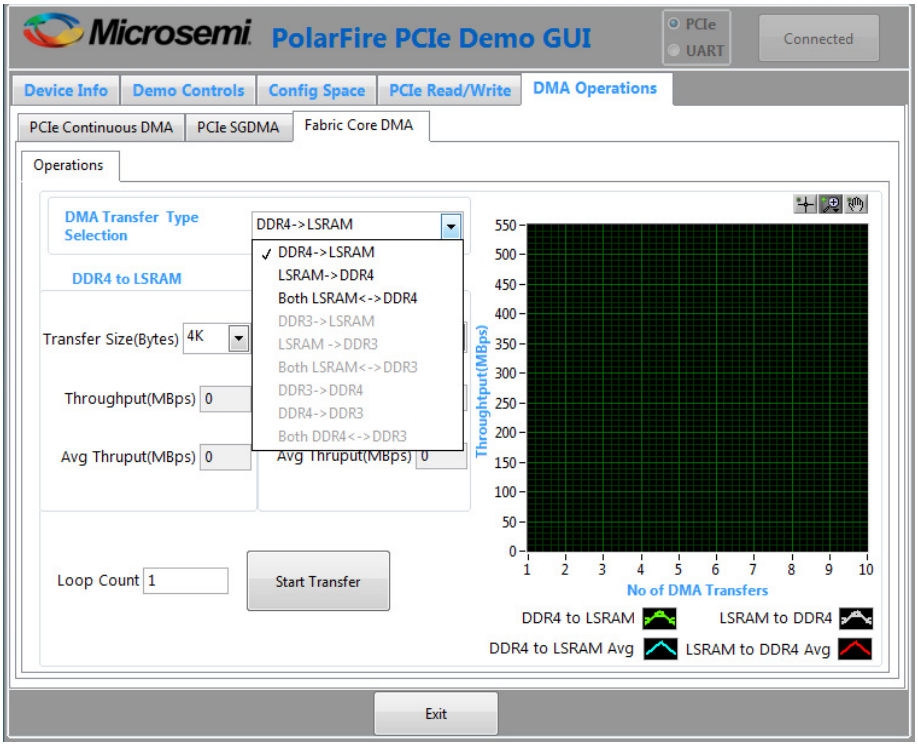
5. Click **OK**.

5.2.3.5 Core DMA—Operations

The following instructions describe running DMA operations between LSRAM and DDR4:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
   - **LSRAM -> DDR4**—to transfer the data from LSRAM to PolarFire DDR4 memory
   - **DDR4-> LSRAM**—to transfer the data from PolarFire DDR4 memory to LSRAM
   - **Both LSRAM <->DDR4**—to transfer the data from LSRAM to and from PolarFire DDR4 memory
2. Select **Transfer Size** (4KB to 1MB) 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 Core DMA—Operations.

Figure 64 • **Core DMA—Operations**



Note: The AXI LSRAM in the design is configured for 4KB. This 4KB is over written if more than 4KB of DMA operation is performed on LSRAM. This option is provided to exercise the throughputs with larger DMA size.

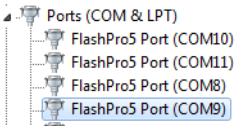
5. Click **Exit** to quit the demo.

5.3 Running the Demo Through UART

The following steps describes how to run a demo using UART if the host PC PCIe slot is not available:

Check the **Device Manager** of the host PC for UART ports.
The following figure shows the example UART ports in the **Device Manager** window.

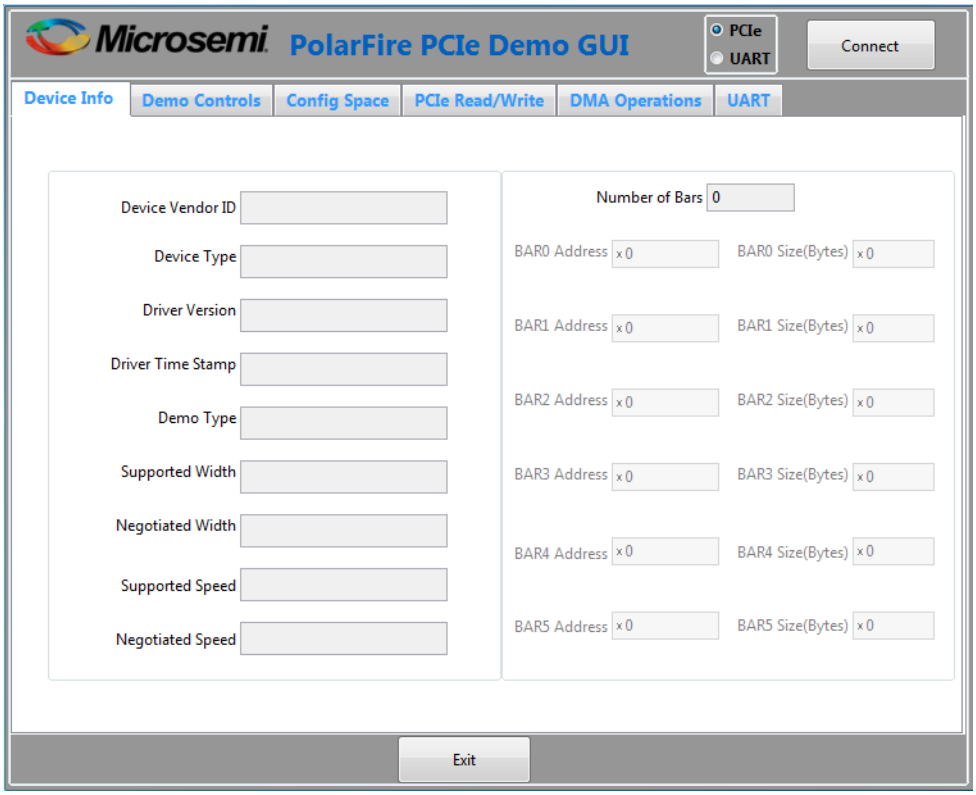
Figure 65 • **Device Manager—UART Ports**



The following steps describe how to run the reference design using UART IF:

1. Go to **All Programs** > **PolarFire_PCIe_GUI** > **PolarFire_PCIe_GUI**. The **PolarFire PCIe Demo** window is displayed as shown in the following figure.

Figure 66 • **PCIe EndPoint Demo Application**



2. Select **UART** radio button and click **Connect**.

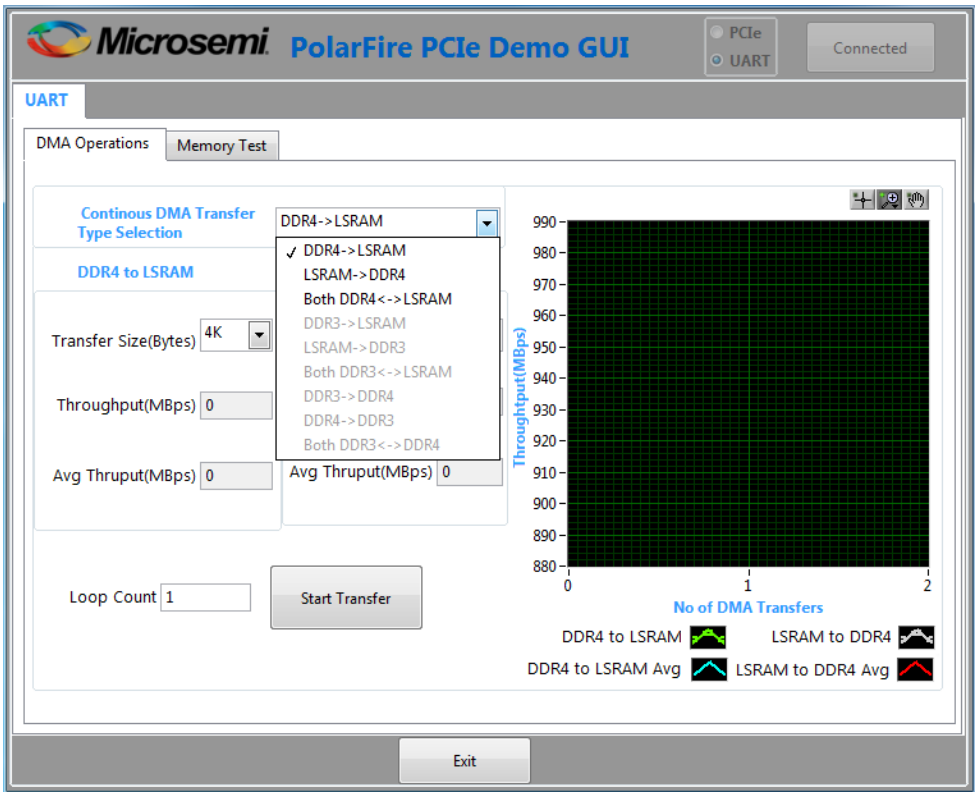
The GUI application scans for UART port and after successful connection, displays the DMA Operations UART tab as shown in Figure 67, page 52.

5.3.1 UART—DMA Operations

The following instructions describe the different ways to read data through LSRAM and DDR:

1. Select one of the following options from the **Continuous DMA Transfer Type Selection** drop-down list:
   - **LSRAM -> DDR4**—to transfer the data from LSRAM to PolarFire DDR4 memory
   - **DDR4-> LSRAM**—to transfer the data from PolarFire DDR4 memory to LSRAM
   - **Both LSRAM <->DDR4**—to transfer the data from LSRAM to and from PolarFire DDR4 memory
2. Select **Transfer Size** (4KB to 512KB) from the drop-down lists.
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 DMA throughput and average throughput from the DDR memory to the LSRAM.

Figure 67 • **UART—DMA Operations**



Note: The AXI LSRAM in the design is configured for 4KB. This 4KB is over written if more than 4KB of DMA operation is performed on LSRAM. This option is provided to exercise the throughputs with larger DMA size.

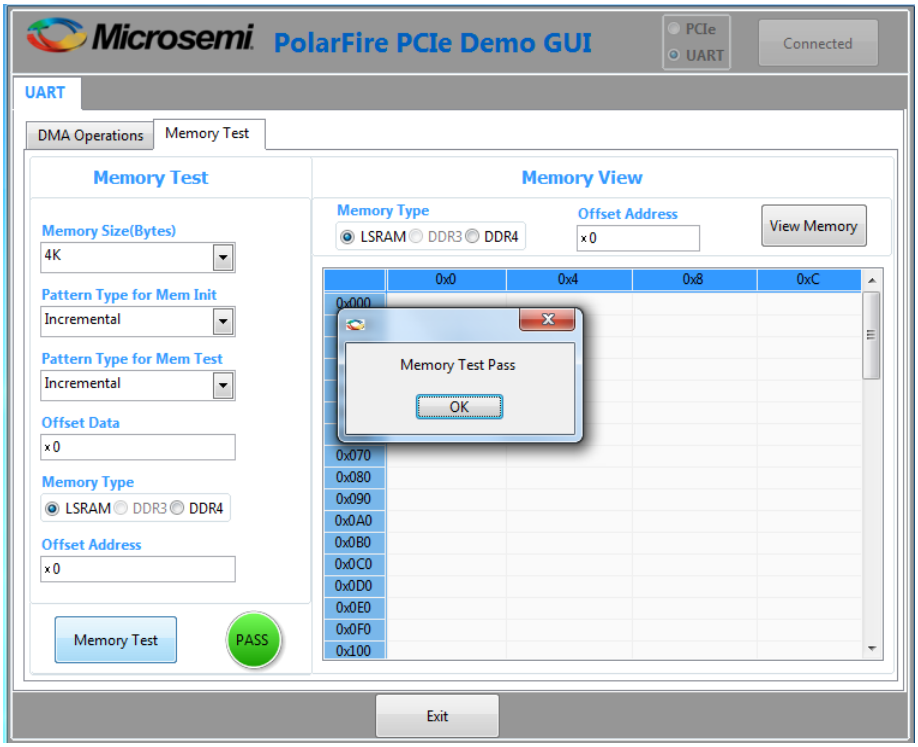
5.3.1.1 UART—Memory Test

The following instructions describe running **Memory Test** between PC and DDR4/LSRAM:

1. Select **Transfer Size** (4KB to 1MB) from the drop-down list.
2. Select **Pattern Selection** from the drop-down list—Increment, Decrement, Fill with Zeros, Fill with Ones, Fill with all A's, and Fill with all 5's. For successful Memory test operation, the **Patter Type for Mem Init** and **Patter Type for Mem Test** should be same.
3. Click **Memory Test**.
   - GUI sends command to fabric logic to initiate the LSRAM/DDR4 memory
   - GUI sends command to fabric logic to read and compare LSRAM/DDR4 memory

The following figure shows UART—Memory Test tab.

Figure 68 • **UART—Memory Test**



Note: Change the **Offset Address** and click **View Memory** to read the RAM memory content.

4. Click **View Memory.** It shows 1KB of RAM memory content.
5. Click **OK**.
6. Click **Exit** to quit the demo.

The following table lists the throughput values observed in Continuous DMA mode.

Table 5 • **PolarFire Throughput Summary—Continuous DMA Mode**

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to LSRAM	64 K	1122	1086
LSRAM to PC		1019	1141
Both PC to and from LSRAM		1019/1128	1074/1125
PC to DDR4	64 K	1070	1049
DDR4 to PC		527[1]	528
Both PC to and from DDR4		1075/527	1050/528

1. The PCIe DMA performs maximum of 32 beat AXI burst transactions (not AXI4's maximum of 256 beat), which causes low read performance of DDR4.

The following table lists the throughput values observed in SGDMA Mode.

Table 6 • **PolarFire Throughput Summary—SGDMA Mode**

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to DDR4	1 MB	1031	1028
DDR4 to PC		530[1]	530
Both PC to and from DDR4		1038/530	1020/530

1. The PCIe DMA performs maximum of 32 beat AXI burst transactions (not AXI4's maximum of 256 beat), which causes low read performance of DDR4.

The following table lists the throughput values observed in Core DMA mode.

Table 7 • **PolarFire Throughput Summary—Core DMA Mode**

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
LSRAM to DDR4	1 MB	1470	1470
DDR4 to LSRAM		1205	1205
Both LSRAM to and from DDR4		1470/1205	1470/1205

6 Appendix: References

This section lists documents that provide more information about the PCIe EndPoint and IP cores used in the reference design.

- For more information about PolarFire transceiver blocks, PF_TX_PLL, and PF_XCVR_REF_CLK, see *UG0677: PolarFire FPGA Transceiver User Guide*.
- For more information about PF_PCIE, see *UG0685: PolarFire FPGA PCI Express User Guide*.
- Fore more information about PF_CCC, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.
- Fore more information about DDR4 memory, see *UG0676: PolarFire FPGA DDR Memory Controller User Guide*.
- For more information about Libero, ModelSim, and Synplify, see the *Microsemi Libero SoC PolarFire* web page.
- For more information about PolarFire FPGA Splash Kit, see *UG0786: PolarFire FPGA Splash Kit User Guide.*
- For more information about CoreAHBLite, see *CoreAHBLite Handbook*.
- For more information about CoreAHBtoAPB3, see *CoreAHBtoAPB3 Handbook.*
- For more information about CoreUART, see *CoreUART Handbook*.