

DG0756
Demo Guide
PolarFire FPGA PCIe EndPoint, DDR3, and DDR4
Memory Controller Data Plane



a  MICROCHIP company



a  MICROCHIP company

Microsemi Headquarters

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

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

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 10.0	1
1.2	Revision 9.0	1
1.3	Revision 8.0	1
1.4	Revision 7.0	1
1.5	Revision 6.0	1
1.6	Revision 5.0	1
1.7	Revision 4.0	1
1.8	Revision 3.0	1
1.9	Revision 2.0	1
1.10	Revision 1.0	1
2	PolarFire FPGA PCIe EndPoint, DDR3, and DDR4 Memory Controller Data Plane	2
2.1	Design Requirements	3
2.2	Prerequisites	3
2.3	Demo Design	4
2.3.1	Design Data Flow	6
2.3.2	Design Implementation	9
2.4	Clocking Structure	13
2.5	Reset Structure	14
2.6	Throughput Measurement	14
2.7	Simulating the Design	14
2.7.1	Simulation Flow	15
3	Libero Design Flow	17
3.1	Synthesize	17
3.1.1	Resource Utilization	17
3.2	Place and Route	18
3.3	Verify Timing	19
3.4	Generate Bitstream	19
3.5	Run PROGRAM Action for Evaluation Kit	20
3.6	Run PROGRAM Action for Splash Kit	21
4	Running the Demo	22
4.1	Installing PCIe Demo Application	22
4.2	Running the Demo Through PCIe	23
4.2.1	Connecting the Board to the Host PC PCIe Slot	23
4.2.2	Driver Installation	26
4.2.3	Running the PCIe Demo Application	28
4.3	Running the Demo Through UART	34
4.3.1	UART—DMA Operations	34
4.3.2	Running the Demo Design on Linux	37
4.4	Throughput Summary of Evaluation Kit	44
4.5	Throughput Summary of Splash Kit	45
5	Appendix 1: DDR3 and DDR4 Power Measurement	46
5.1	DDR3	46
5.2	DDR4	46

6	Appendix 2: DDR4 Configuration	47
7	Appendix 3: Programming the Device Using FlashPro Express	51
8	Appendix 4: Running the TCL Script	54
9	Appendix 5: References	55

Figures

Figure 1	PCIe Demo Design Top-Level Block Diagram	5
Figure 2	DMA0 – Example of SG DMA Operation	7
Figure 3	DMA1 – Example of SG DMA Operation	8
Figure 4	PCIe EndPoint Reference Design	9
Figure 5	PCIe_EP SmartDesign	10
Figure 6	PCIe_TL_CLK SmartDesign	10
Figure 7	CoreDMA_IO_CTRL SmartDesign	11
Figure 8	UART SmartDesign	11
Figure 9	AXI_to_APB SmartDesign	12
Figure 10	Clocking Structure	13
Figure 11	Reset Structure	14
Figure 12	Simulating the Design	15
Figure 13	Simulation Transcript Window	16
Figure 14	Simulation Waveform Window	16
Figure 15	Simulation Waveform Window	16
Figure 16	I/O Editor—XCVR View	18
Figure 17	I/O Editor—DDR3 Memory View	18
Figure 18	I/O Editor—DDR4 Memory View (For Evaluation Kit)	18
Figure 19	I/O Editor—DDR4 Memory View (For Splash Kit)	19
Figure 20	Design Flow	19
Figure 21	Evaluation Kit Board Setup	20
Figure 22	Splash Kit Board Setup	21
Figure 23	Programming the Device	21
Figure 24	Installing PCIe Demo Application	22
Figure 25	Successful Installation of PCIe Demo Application	23
Figure 26	PolarFire Evaluation Kit Setup for Host PC	24
Figure 27	PolarFire Splash Kit Setup for Host PC	25
Figure 28	Device Manager	25
Figure 29	Update Driver Software	26
Figure 30	Browse for Driver Software	26
Figure 31	Browse for Driver Software Continued	27
Figure 32	Windows Security	27
Figure 33	Successful Driver Installation	27
Figure 34	Device Manager—PCIe Device Detection	28
Figure 35	PCIe EndPoint Demo Application	28
Figure 36	Device Info	29
Figure 37	Demo Controls	29
Figure 38	Configuration Space	30
Figure 39	PCIe BAR2 Memory Access—LSRAM	30
Figure 40	Continuous DMA Operations with DMA Transfer Type Selection as Both PC and LSRAM	31
Figure 41	Continuous DMA Memory Test—Memory Test Successful	32
Figure 42	Device Manager—UART Ports	34
Figure 43	UART—DMA Operations	35
Figure 44	UART—Memory Test	36
Figure 45	PCIe Device Detection	37
Figure 46	PCIe Device Driver Installation	38
Figure 47	Linux PCIe Application Utility	38
Figure 48	Device Information	39
Figure 49	Blink LEDs	39
Figure 50	Dip Switch Status	39
Figure 51	For Interrupt	40
Figure 52	For Bar Read/Write	41
Figure 53	For PCIe configspace	42
Figure 54	For DMA Operation	43

Figure 55	For Continuous DMA write/read DDR3	43
Figure 56	DDR4 Configurator	47
Figure 57	DDR4 Configurator—Memory Initialization	48
Figure 58	DDR4 Configurator—Memory Timing	49
Figure 59	DDR4 Configurator—Controller	50
Figure 60	DDR4 Configurator—Misc	50
Figure 61	FlashPro Express Job Project	51
Figure 62	New Job Project from FlashPro Express Job	52
Figure 63	Programming the Device	52
Figure 64	FlashPro Express—RUN PASSED	53

Tables

Table 1	Design Requirements	3
Table 2	Resource Utilization—Evaluation Kit	17
Table 3	Resource Utilization—Splash Kit	17
Table 4	Jumper Settings	20
Table 5	Jumper Settings	21
Table 6	PolarFire Throughput Summary—PCIe Continuous DMA Mode	44
Table 7	PolarFire Throughput Summary—PCIe SGDMA Mode	44
Table 8	PolarFire Throughput Summary—Fabric Core DMA Mode	44
Table 9	PolarFire Throughput Summary—PCIe Continuous DMA Mode	45
Table 10	PolarFire Throughput Summary—PCIe SGDMA Mode	45
Table 11	PolarFire Throughput Summary—Fabric Core DMA Mode	45

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 10.0

Added Appendix 4: Running the TCL Script, page 54.

1.2 Revision 9.0

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

- Updated the document for Libero SoC v12.3.
- Removed the references to Libero version numbers.

1.3 Revision 8.0

The document was updated for Libero SoC v12.0 release.

1.4 Revision 7.0

Merged Splash kit related content and updated the document for Libero SoC PolarFire v2.3 release.

1.5 Revision 6.0

The following is a summary of the changes made in revision 6.0 of this document.

- The document was updated for Libero SoC PolarFire v2.2 release.
- Information about DDR power measurement was added. See [Appendix 1: DDR3 and DDR4 Power Measurement](#), page 46.

1.6 Revision 5.0

The document was updated for Libero SoC PolarFire v2.1 release.

1.7 Revision 4.0

The document was updated to include features and enhancements introduced in the Libero SoC PolarFire v2.0 release.

1.8 Revision 3.0

The following is a summary of the changes made in revision 3.0 of this document.

- The document was updated to include features and enhancements introduced in the Libero SoC PolarFire v1.1 SP1 release.
- Information about setting up the device and running the demo was added, see [Programming the Device Using FlashPro](#), page 21, and [Running the Demo](#), page 22.
- List of reference was added. For more information, see [Appendix 5: References](#), page 55.

1.9 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- The document was updated for Libero SoC PolarFire v1.1 release.
- Information about resource utilization was added. For more information, see [Resource Utilization](#), page 17.

1.10 Revision 1.0

The first publication of this document.

2 PolarFire FPGA PCIe EndPoint, DDR3, 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 software.

The PCIESS is compliant with the PCI Express Base Specification, Revision 3.0 with Gen1/2 speed. 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, DDR3, DDR3L, 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 about 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, Soft DDR3, 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 performs bulk-data transfer between contiguous/scatter gather memory locations on a host PC and contiguous memory locations of DDR3/DDR4/LSRAM.
- The CoreAXI4DMAController performs data transfers between DDR3/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, DDR3, 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, DDR3, DDR4, and the LSRAM memories of the PolarFire Evaluation/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 initiate the DMA transactions between DDR3/DDR4 and LSRAM through UART IF. If the host PC PCIe slot is not available, the DMA between DDR3/DDR4 and LSRAM is exercised through UART IF.

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

- Using the job file: To program the device using the job file provided along with the design files, see [Appendix 3: Programming the Device Using FlashPro Express](#), page 51.
- Using Libero SoC: To program the device using Libero SoC, see [Libero Design Flow](#), page 17. Use this option when the reference design is modified.

Note: The user can debug the PCIe features: PCIe lane status, LTSSM state machine, and other available PCIe features using SmartDebug. For more information about PCIE Debug using SmartDebug, see *SmartDebug User Guide*.

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 Linux CentOS Kernel version 3.10.0
Hardware	
PolarFire Evaluation Kit (MPF300TS-FCG1152I) or PolarFire Splash Kit (MPF300T-1FCG484)	Rev D or later Rev 2 or later
PCIe Edge card ribbon cable (not provided with the kit)	
Host PC with PCIe compliant slot with x4 or higher width	
Software	
Libero SoC	
Modelsim	Refer to the <code>readme.txt</code> file provided in the design files for the software versions used with this reference design.
Synplify Pro	

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 begin:

- For demo design files download the link:
For Evaluation kit:
http://soc.microsemi.com/download/rsc/?f=mpf_dg0756_eval_df
For Splash kit:
http://soc.microsemi.com/download/rsc/?f=mpf_dg0756_splash_df
- 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#downloads>

The latest versions of ModelSim, Synplify Pro, and FTDI drivers are included in the Libero SoC 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 a PCIe link with the PolarFire FPGA PCIe EndPoint and access the control registers, DDR3, 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 DDR3/DDR4/LSRAM.
- DMA Engine1: performs DMA from DDR3/DDR4/LSRAM to host PC memory.

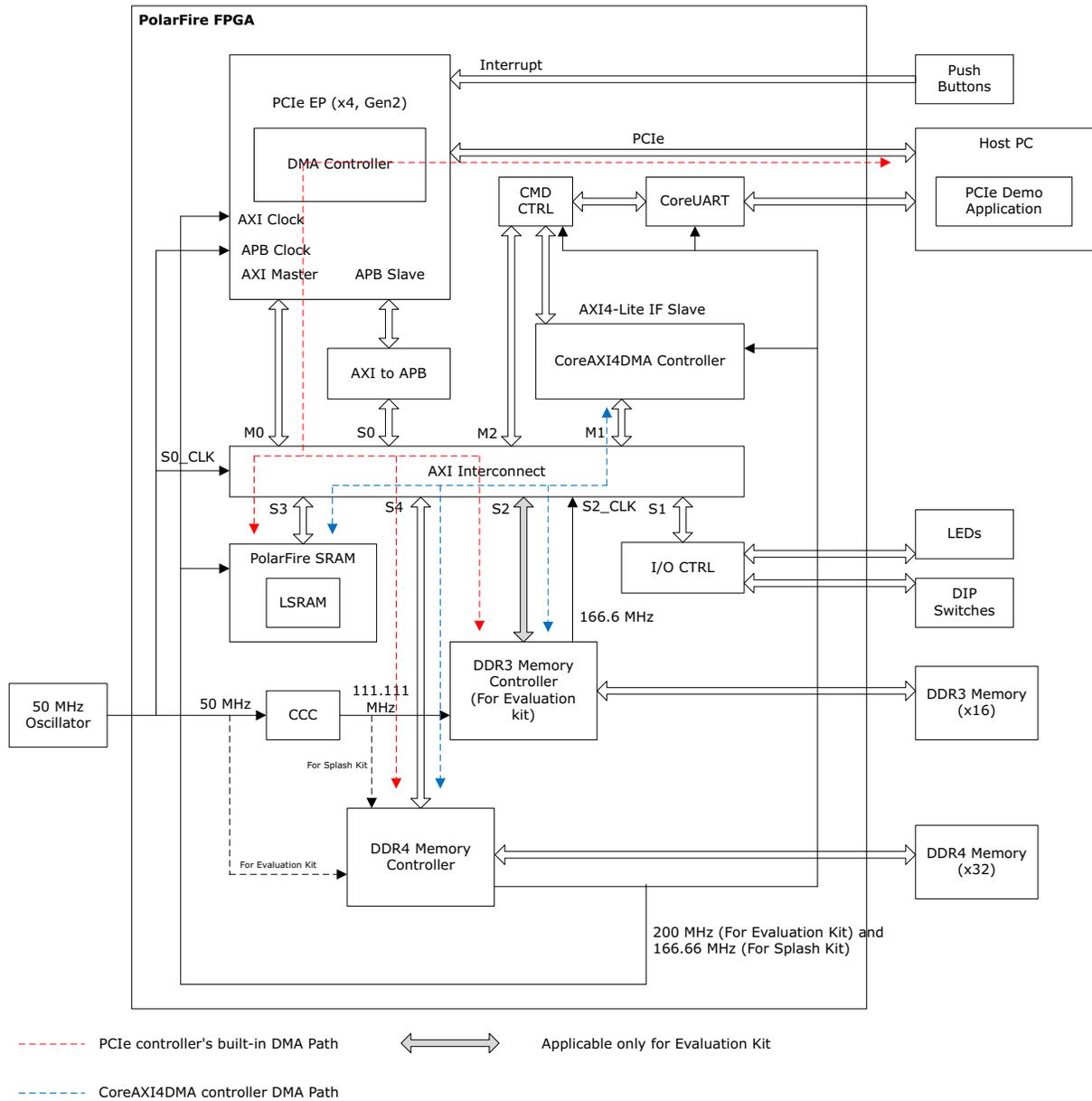
Note: For SGDMA type of DMA operations, the PCIe driver finds the available memory locations and creates the buffer descriptor chain for the different memory locations. It also configures the PCIe DMA for SGDMA and the base address of the first buffer descriptor.

The PCIe demo application initiates CoreAXI4DMA controller IP to perform the DMA between DDR3 memory and LSRAM. The following are the two channels of the CoreAXI4DMA controller IP:

- Channel0: performs DMA from—DDR3 to DDR4, DDR3 to LSRAM, and DDR4 to LSRAM
- Channel1: performs DMA from—DDR4 to DDR3, LSRAM to DDR3, and 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 Evaluation/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 DDR3/DDR4/LSRAM base address. It performs BAR2 memory read/write transactions to DDR3/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 DDR3/DDR4/LSRAM

PCIe DMA Engine0 performs continuous DMA from host PC memory to DDR3/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 DDR3/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 – DDR3/DDR4/LSRAM to Host PC Memory

PCIe DMA Engine1 performs continuous DMA from DDR3/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 DDR3/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.

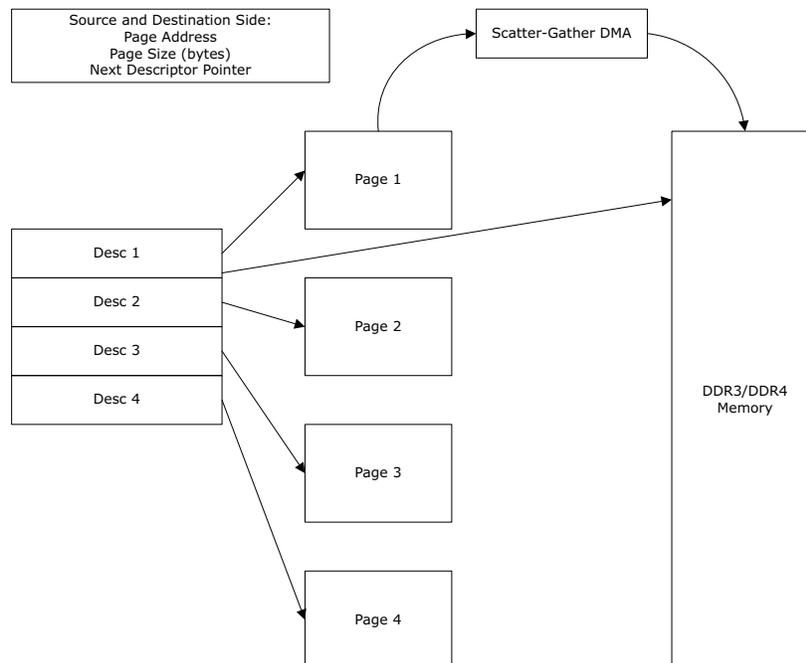
2.3.1.2.1 Host PC Memory to DDR3/DDR4

PCIe DMA Engine0 performs DMA from host PC memory to DDR3/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 DDR3/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 DDR3/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



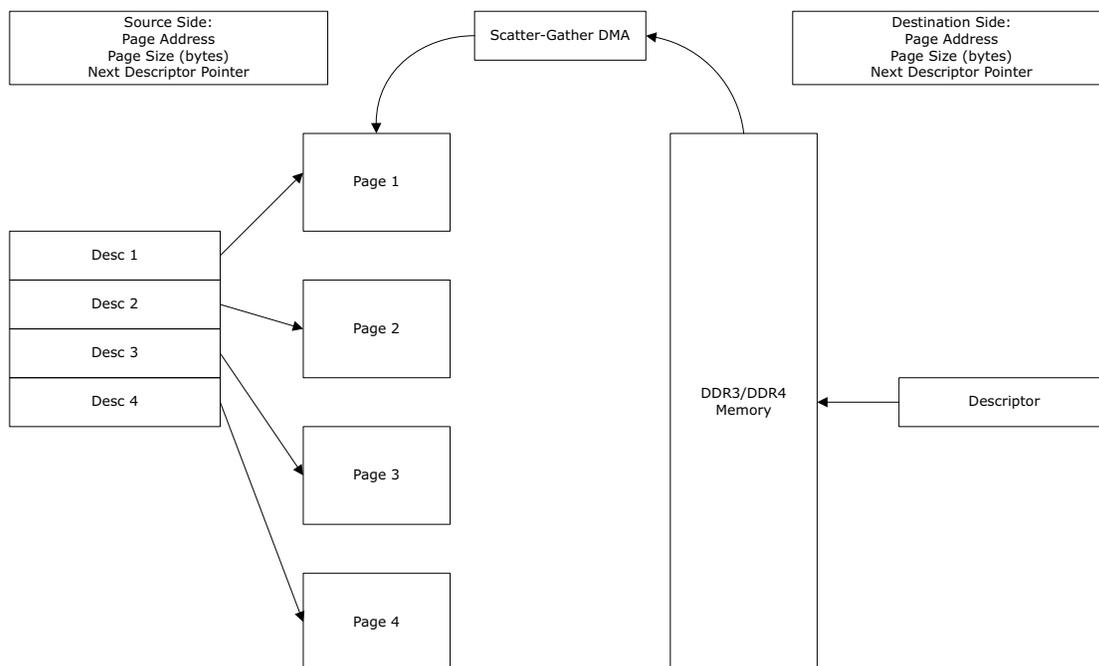
2.3.1.2.2 DDR3/DDR4 to Host PC Memory:

PCIe DMA Engine1 performs DMA from DDR3/DDR4 memories to host PC memory as shown in the following figure.

The following steps describe the SGDMA operation of PCIe DMA1:

1. PolarFire_PClE_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 DDR3/DDR4 memory is treated as the continuous memory. Single buffer descriptor is created in LSRAM with the base address of DDR3/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 DDR3/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_PClE_GUI application.

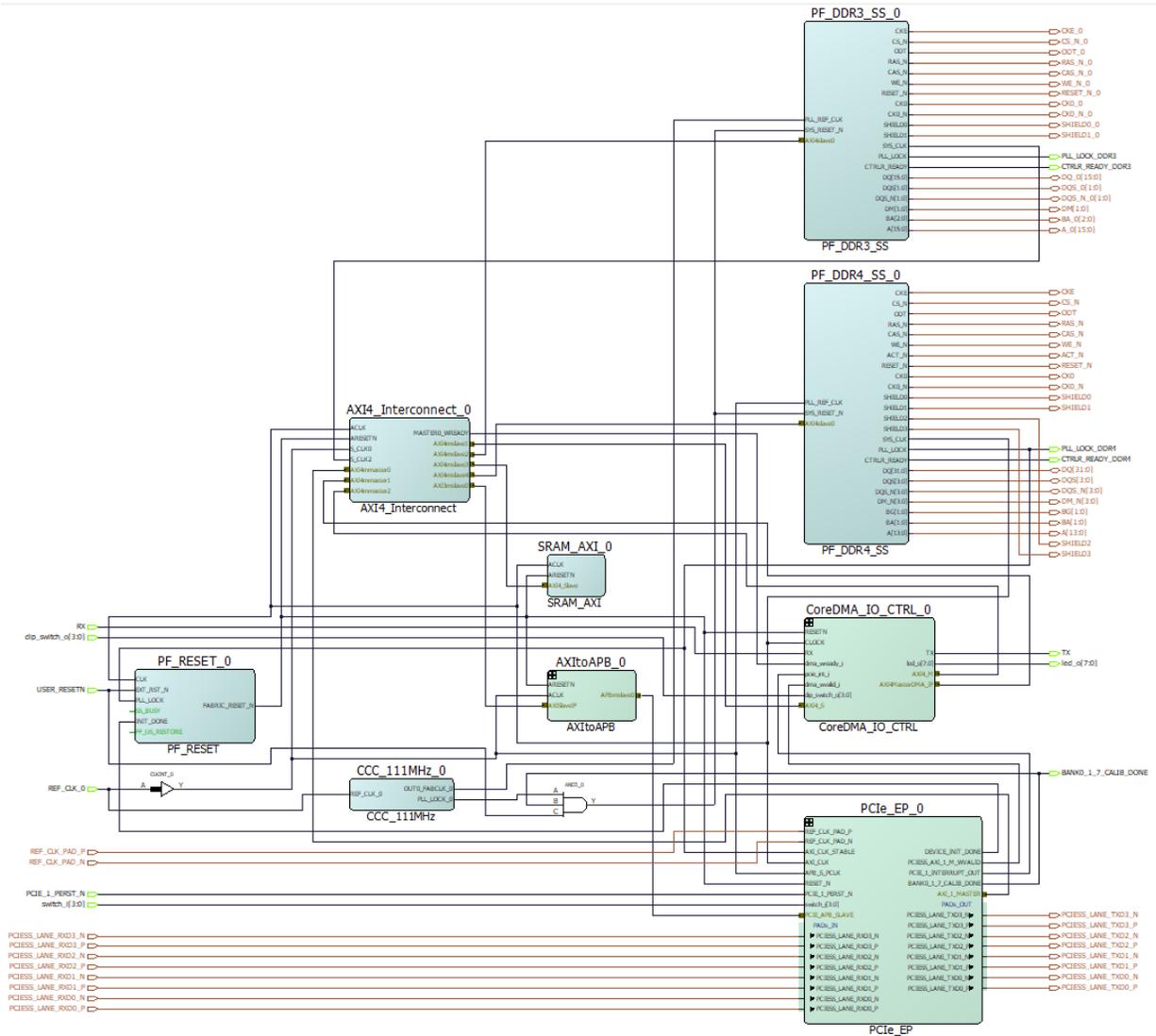
Figure 3 • DMA1 – Example of SG DMA Operation



2.3.2 Design Implementation

The following figure shows the Libero SoC software top-level design implementation of the PCIe EndPoint reference design.

Figure 4 • PCIe EndPoint Reference Design



The top-level design includes the following SmartDesign components, memory controller subsystems, and AXI4Interconnect IP.

- PCIe EP subsystem
- CoreDMA and UART subsystem
- AXItoAPB
- DDR3 subsystem
- DDR4 subsystem
- AXI LSRAM
- AXI4Interconnect IP

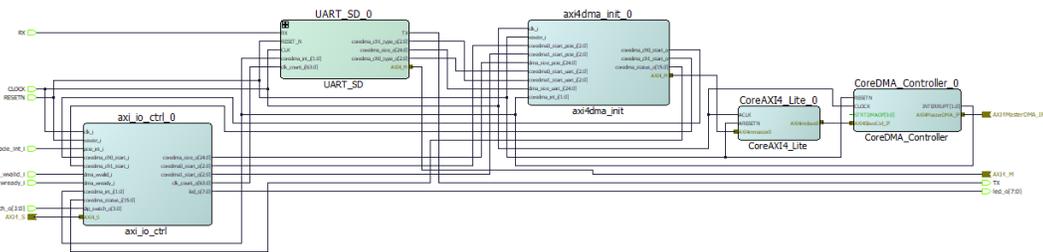
2.3.2.2 CoreDMA and UART Subsystem

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

axi4dma_init logic initiates the CoreDMA through the AXI4Lite interface to perform the DMA as per commands from GUI. axi_io_ctrl block receives commands from PCIe BAR space and controls the IOs or axi4dma_init logic.

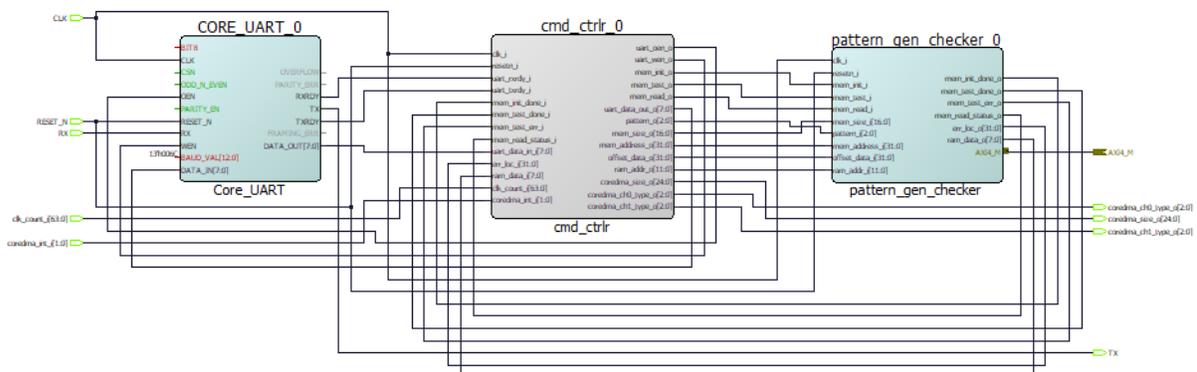
The CoreAXI4DMAController IP is configured for 64-bit AXI4 data width, and to generate interrupts for descriptor0 and descriptor1. Descriptor0 is used for—DDR3 to DDR4, DDR3 to LSRAM, and DDR4 to LSRAM DMA and descriptor1 is used for—DDR4 to DDR3, LSRAM to DDR3, and LSRAM to DDR4 DMA.

Figure 7 • CoreDMA_IO_CTRL SmartDesign



The UART_SD SmartDesign implements logic required to communicate with UART IF as shown in the following figure. cmd_ctrl block receives commands from UART and triggers the logic to perform CoreDMA/DDR memory initialization. pattern_gen_checker block initializes the DDR memory with the specified pattern and compares against the specified pattern.

Figure 8 • UART SmartDesign



2.3.2.3 Memory Controller Subsystem

2.3.2.3.1 DDR3

The DDR3 subsystem is configured to access the 16-bit DDR3 memory through an AXI4 interface. The “PolarFire evaluation kit DDR3 memory” preset is applied to configure all of the memory initialization and timing parameters in the DDR3 configurator.

Note: DDR3 is applicable only for Evaluation kit demo design.

2.3.2.3.2 DDR4

The DDR4 subsystem is configured to access the 32-bit DDR4 memory through an AXI4 64-bit interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4 memory on the PolarFire Evaluation/Splash kit. For more information about DDR4 subsystem configuration, see Appendix 2: DDR4 Configuration, page 47.

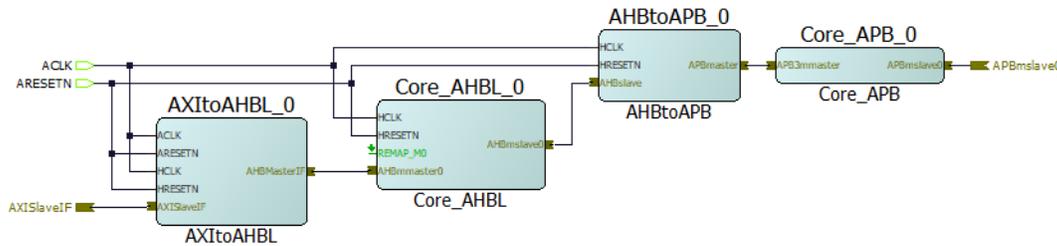
2.3.2.3.3 AXI LSRAM

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

2.3.2.4 AXI to APB SmartDesign

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

Figure 9 • AXI_to_APB SmartDesign



2.3.2.5 CoreAXI4Interconnect IP

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

- Master0: PCIe
- Master1: CoreAXI4DMAController IP
- Master2: Pattern generator and checker logic (pattern_gen_checker block)
- Slave0: AXItoAPB bridge (0x0000_0000 to 0x0FFF_FFFF)
- Slave1: AXI Slave Fabric Registers (0x1000_0000 to 0x1FFF_FFFF)
- Slave2: DDR3 Subsystem (0x2000_0000 to 0x2FFF_FFFF) (Not enabled for Splash kit)
- Slave3: AXI4 LSRAM (0x3000_0000 to 0x3FFF_FFFF)
- Slave4: DDR4 Subsystem (0x4000_0000 to 0x4FFF_FFFF)

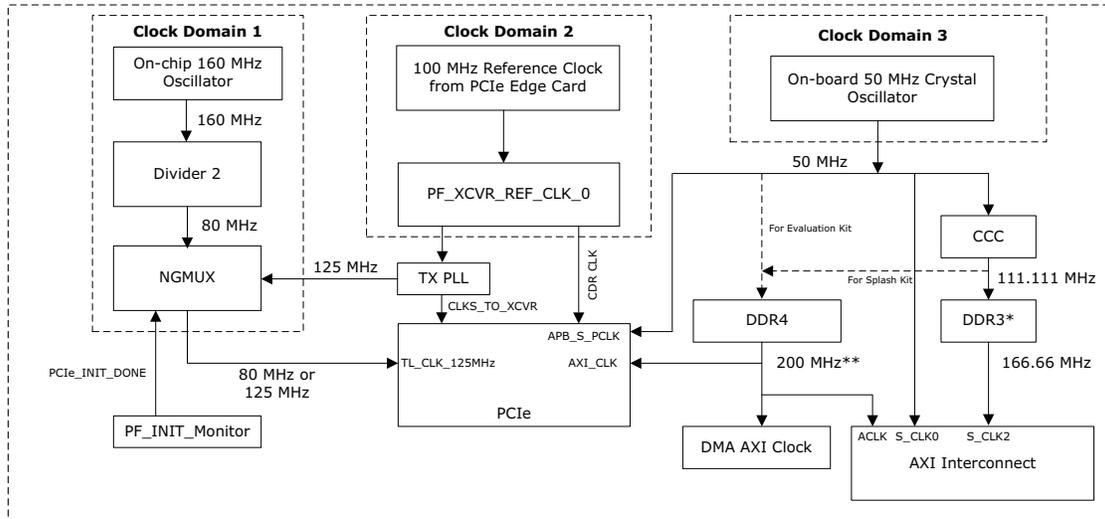
Slave0 is configured to convert AXI4 transactions to AXI3 transactions.

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.
- Clock Domain 2: generates CDR reference and XCVR clocks for PCIe.
- Clock Domain 3: generates 50 MHz clock for PCIe APB, DDR4 PLL reference, and CCC reference clocks. DDR4 subsystem generates a 200 MHz (166.66 MHz for Splash kit) clock for fabric AXI interface logic. DDR3 subsystem generates 166.66 MHz clock and is connected to AXI interconnect slave2 CDC interface.

Figure 10 • Clocking Structure



*: DDR3 is applicable only for Evaluation demo design.

** : On Splash kit, 166.66 MHz is used.

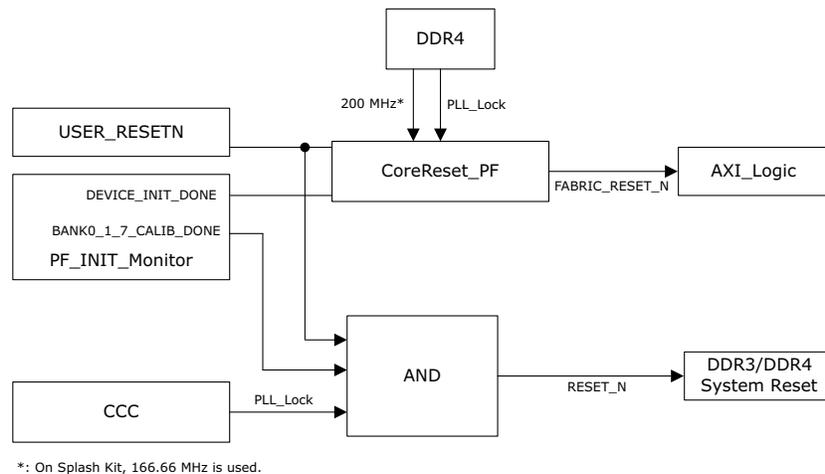
2.5 Reset Structure

The CoreReset_PF synchronizes the external USER_RESETN (**SW6** on PolarFire Evaluation kit and **SW2** on PolarFire Splash kit) to the 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 DDR3/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 11 • 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 begin:

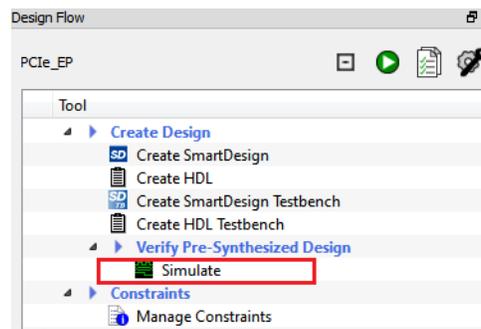
1. Start Libero SoC, in the **Project** menu, and click **Open Project**.
2. Browse the Libero Project > PCIe_EP_Demo_EvalKit or PCIe_EP_Demo_SplashKit Libero project folder and open the `Libero_Project.prjx` file. The PolarFire PCIe EndPoint project opens.
3. Open the **Design Hierarchy** window and 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`, `DDR3`, `DDR4`, `CoreAHBLite`, `CoreAPB`, `CoreAXItoAHBL`, `CoreAHBLtoAPB`, `CoreUART`, and PolarFire SRAM IP cores under **Libero SoC > Catalog**.

The PCIe BFM performs 1 KB DMA operations between PCIe and DDR3, DDR4 and 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 DDR3 and DDR4 memory models are instantiated in the testbench for simulating DDR3 and 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 12 • 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. DDR3 and DDR4 memory controllers initializes the DDR3/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`, DDR3, DDR4 through CoreAXI4Interconnect as per the .bfm script.
5. After 18 μ s, the simulation completes. **PCIE1 BFM Simulation Complete – 282 Instructions – NO ERRORS** message is displayed for **Evaluation kit**, as shown in [Figure 13](#), page 16.
6. After 13 μ s, the simulation completes. **PCIE1 BFM Simulation Complete – 272 Instructions – NO ERRORS** message is displayed for **Splash kit**.

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](#).

3 Libero Design Flow

The Libero design flow involves the following processes:

- Synthesize
- Place and route
- Verify timing
- Design and Memory Initialization
- 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 [Figure 20](#), page 19.

3.1.1 Resource Utilization

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

Table 2 • Resource Utilization—Evaluation Kit

Type	Used	Total	Percentage
4LUT	46386	299544	15.49
DFF	38186	299544	12.75
I/O Register	0	510	0.00
User I/O	154	512	30.08
– Single-ended I/O	138	512	26.95
– Differential I/O Pairs	8	256	3.13

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

Table 3 • Resource Utilization—Splash Kit

Type	Used	Total	Percentage
4LUT	33704	299544	11.25
DFF	26964	299544	9.00
I/O Register	0	242	0.00
User I/O	99	244	40.57
– Single-ended I/O	89	244	36.48
– Differential I/O Pairs	5	122	4.10

3.2 Place and Route

To place and route the design, the TX_PLL, XCVR_REF_CLK, DDR3, DDR4, and CCC need to be constrained using the I/O Editor as shown in the following figures.

Figure 16 • I/O Editor—XCVR View

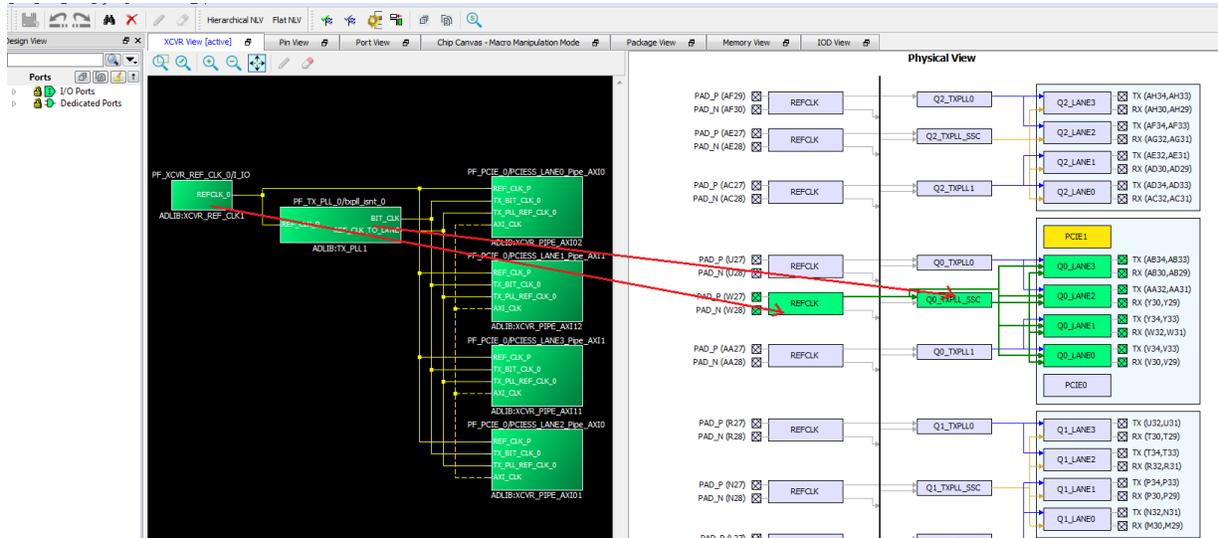


Figure 17 • I/O Editor—DDR3 Memory View

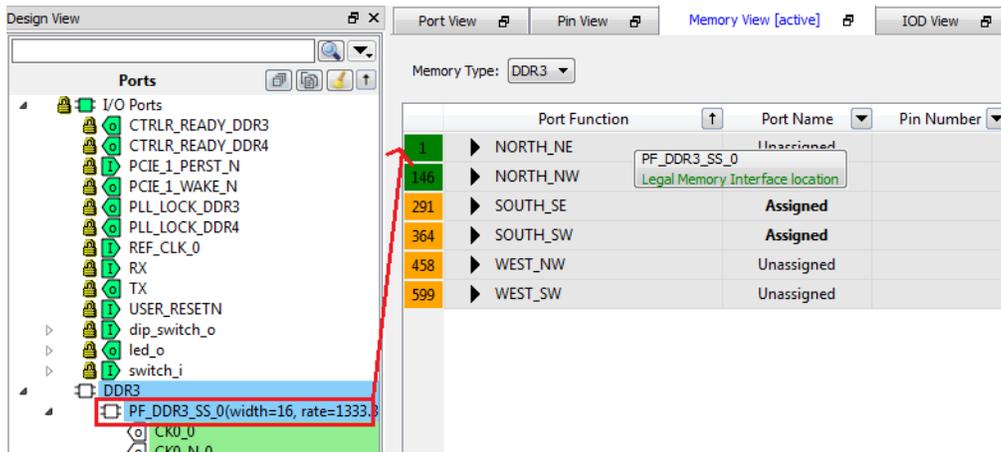


Figure 18 • I/O Editor—DDR4 Memory View (For Evaluation Kit)

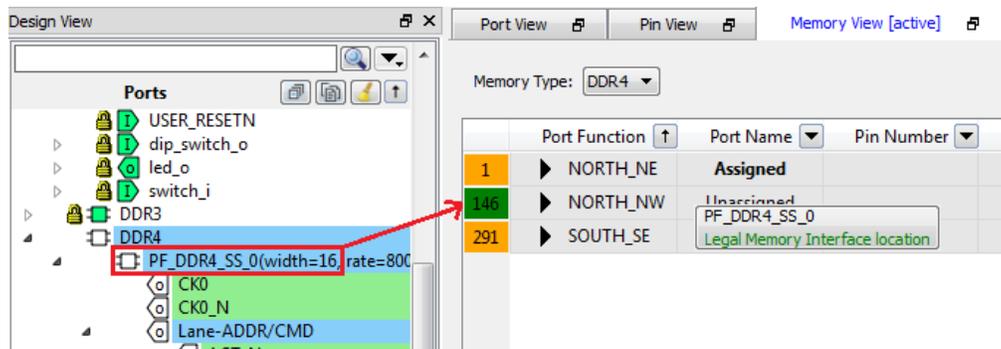
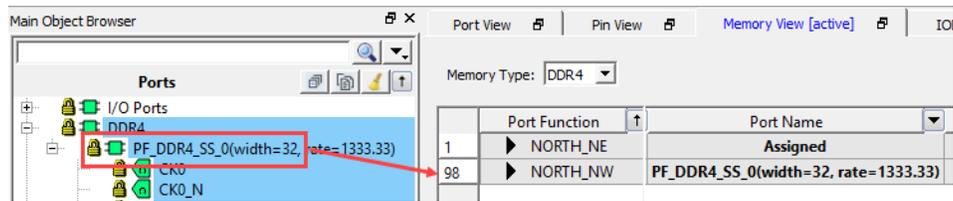
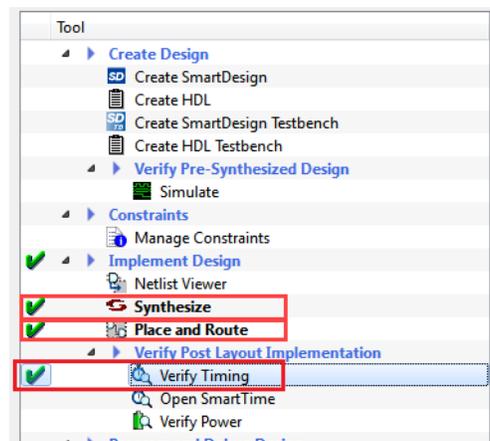


Figure 19 • I/O Editor—DDR4 Memory View (For Splash Kit)


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 Figure 20, page 19.

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 20 • Design Flow


3.4 Generate Bitstream

To generate the bitstream, perform the following steps:

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 23, page 21.
2. Right-click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

3.5 Run PROGRAM Action for Evaluation Kit

After generating the bitstream, the PolarFire device must be programmed. To program the PolarFire device, perform the following steps:

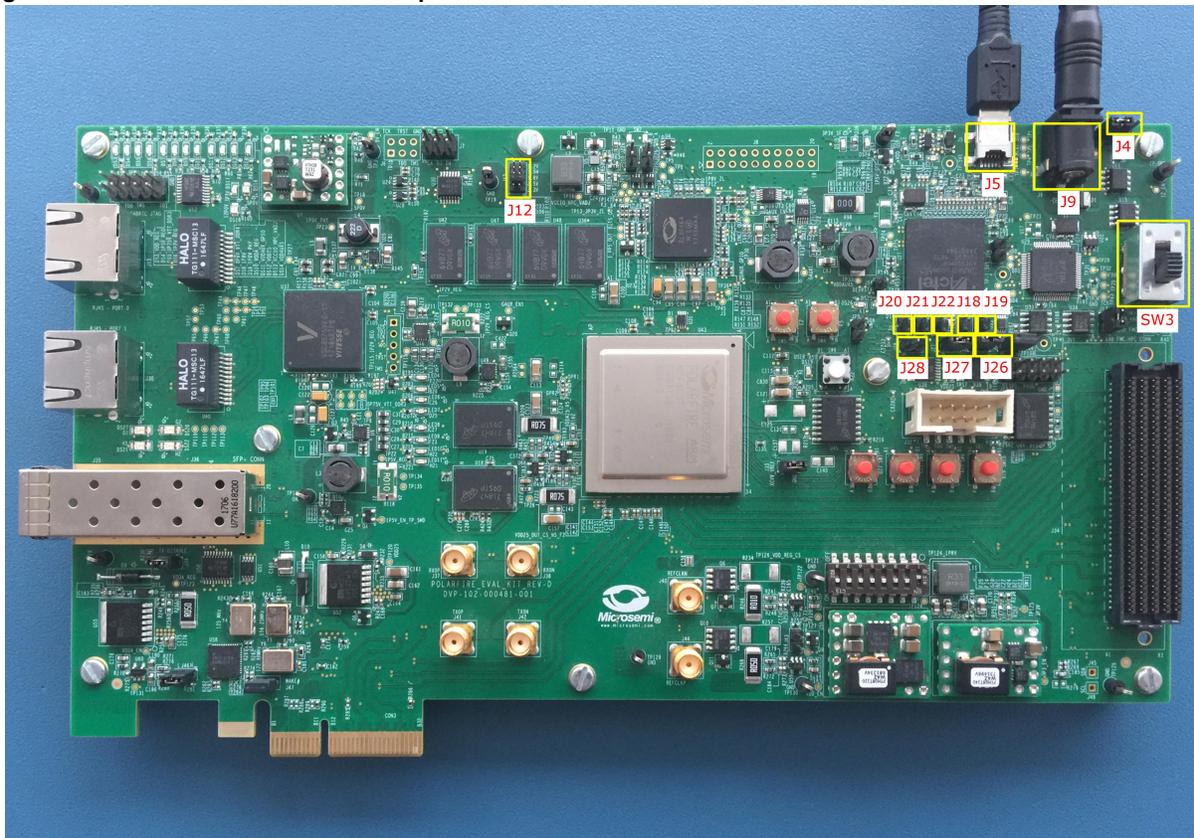
1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Table 4 • Jumper Settings

Jumper	Description
J18, J19, J20, J21, and J22	Short pin 2 and 3 for programming the PolarFire FPGA through FTDI
J28	Short pin 1 and 2 for programming through the on-board FlashPro5
J26	Short pin 1 and 2 for programming through the FTDI SPI
J27	Short pin 1 and 2 for programming through the FTDI SPI
J4	Short pin 1 and 2 for manual power switching using SW3
J12	Short pin 3 and 4 for 2.5 V

2. Connect the power supply cable to the **J9** connector on the board.
3. Connect the USB cable from the Host PC to **J5** (FTDI port) on the board.
4. Power on the board using the **SW3** slide switch.

Figure 21 • Evaluation Kit Board Setup



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

3.6 Run PROGRAM Action for Splash Kit

After generating the bitstream, the PolarFire device must be programmed. To program the PolarFire device, perform the following steps:

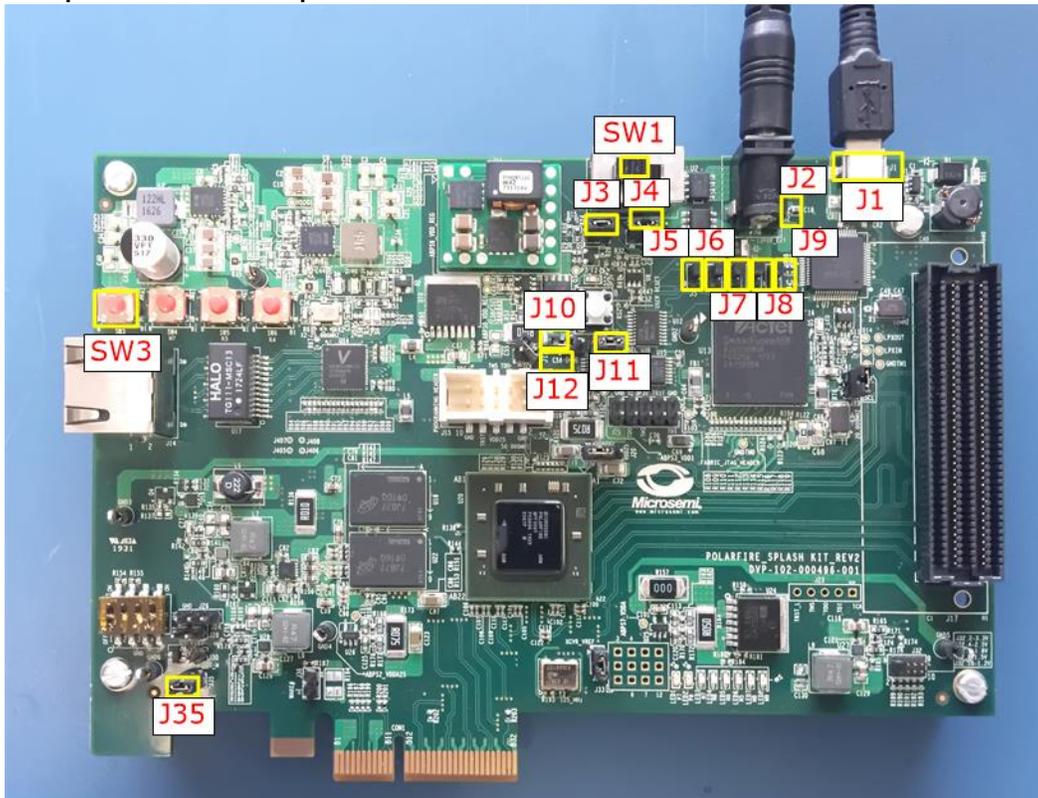
1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Table 5 • Jumper Settings

Jumper	Description
J5, J6, J7, J8, and 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 22 • Splash Kit 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 22 to run the PCIe EndPoint demo.

Figure 23 • Programming the Device



4 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.

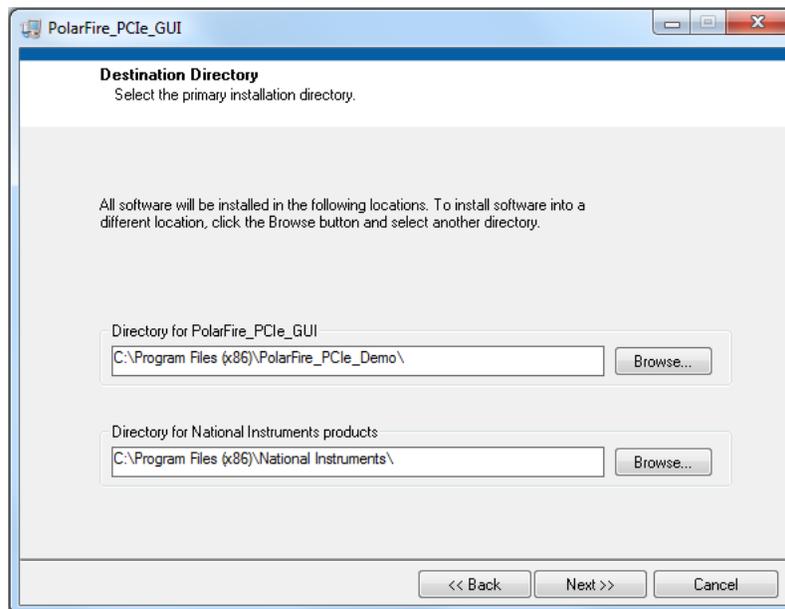
This section also describes how to connect the kit to the Host PC PCIe Slot. If the host PC PCIe slot is not available, the DMA between DDR3/DDR4 and LSRAM can be exercised through UART IF.

4.1 Installing PCIe Demo Application

To install the PolarFire PCIe Demo application, perform the following steps:

1. Install the GUI_Installer (setup.exe) from the following design files folder:
mpf_dg0756_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.

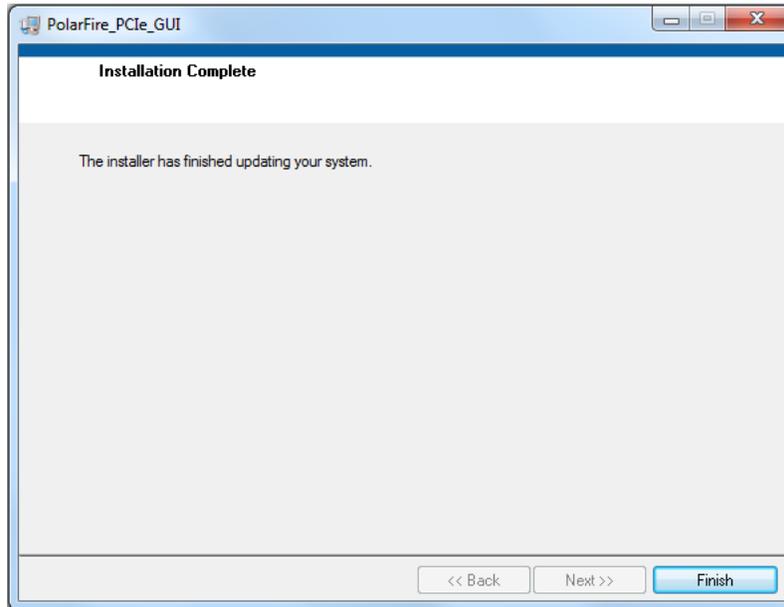
Figure 24 • Installing PCIe Demo Application



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

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

Figure 25 • Successful Installation of PCIe Demo Application



4.2 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.

4.2.1 Connecting the Board to the Host PC PCIe Slot

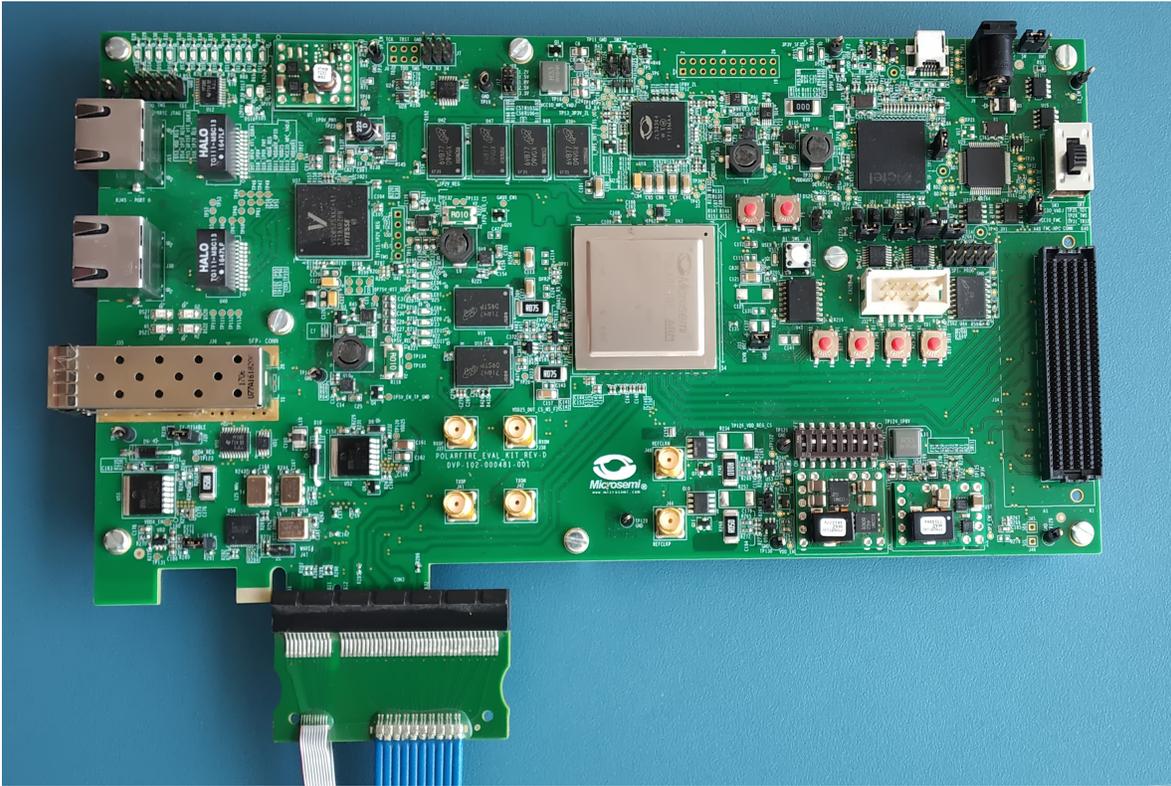
1. After successful programming, power OFF the PolarFire Evaluation/Splash board and shut down the host PC.
2. Connect the CON3 - PCIe Edge connector of the PolarFire Evaluation/Splash board to the host PC's PCIe slot through the PCI Edge card ribbon cable.
This demo is designed to work with any PCIe 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.

The following figure shows the board setup for the host PC in which the PolarFire Evaluation Kit is connected to the host PC PCIe slot using PCIe Edge Card Ribbon cable (not supplied with the kit).

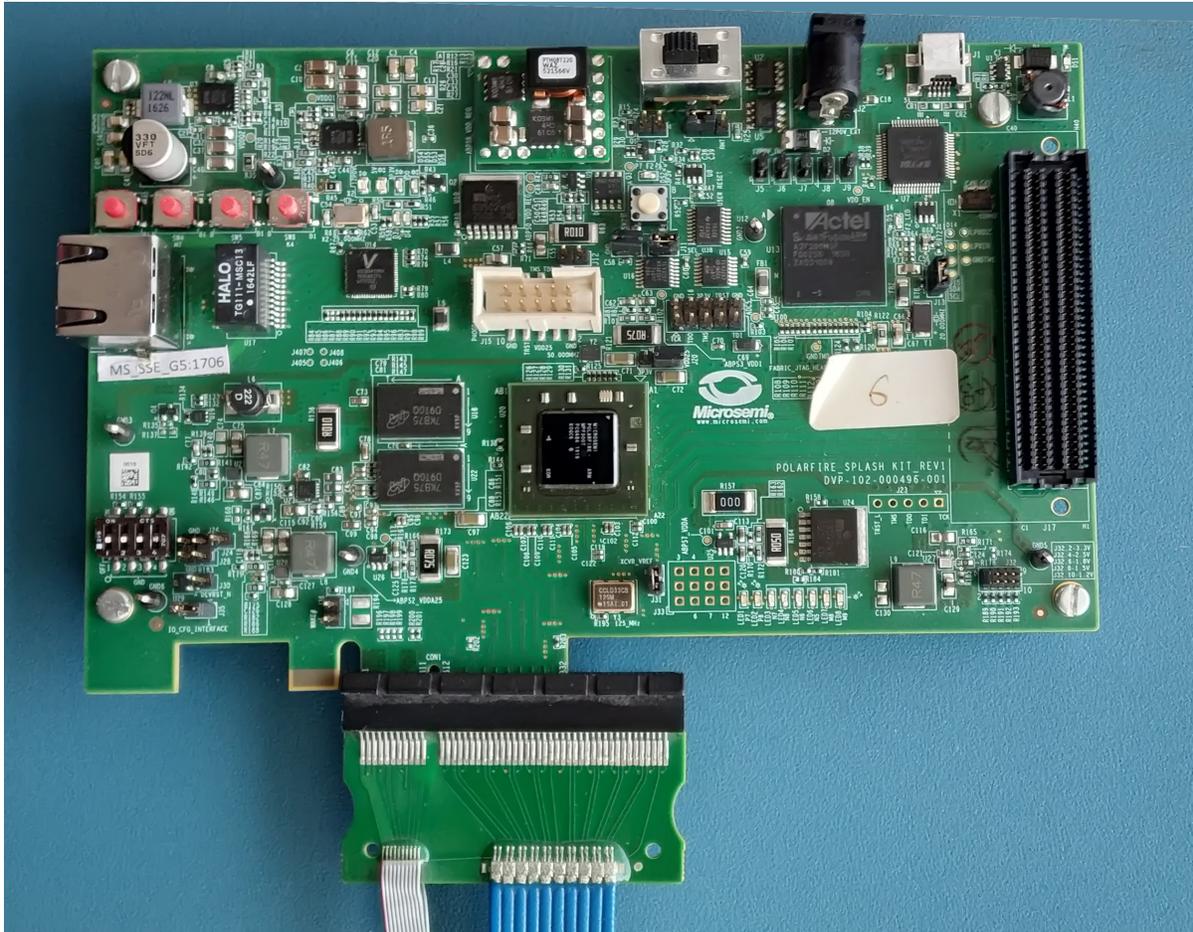
Figure 26 • PolarFire Evaluation Kit Setup for Host PC



3. Power on the power supply switch **SW3**.

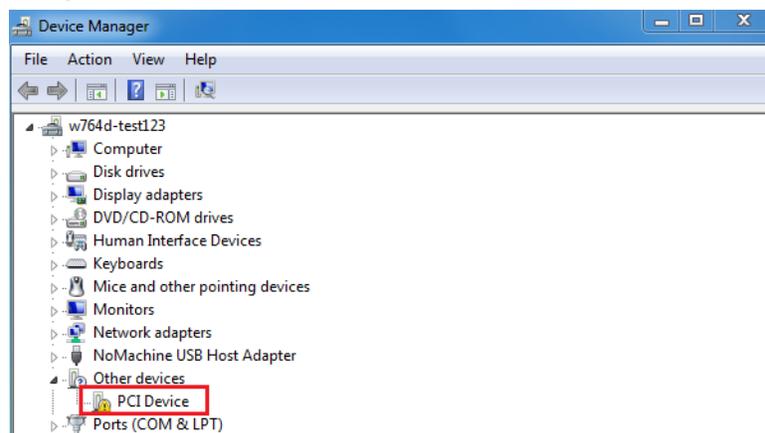
The following figure shows the board setup for the host PC in which the PolarFire Splash Kit is connected to the host PC PCIe slot.

Figure 27 • PolarFire Splash Kit Setup for Host PC



4. Power on the power supply switch **SW1**.
5. 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 28 • Device Manager



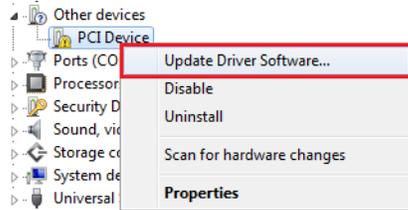
Note: If the device is still not detected, check if the BIOS version in the host PC is the latest and if PCI is enabled in the host PC BIOS.

4.2.2 Driver Installation

To install the PCIe drivers on the host PC, perform the following steps:

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

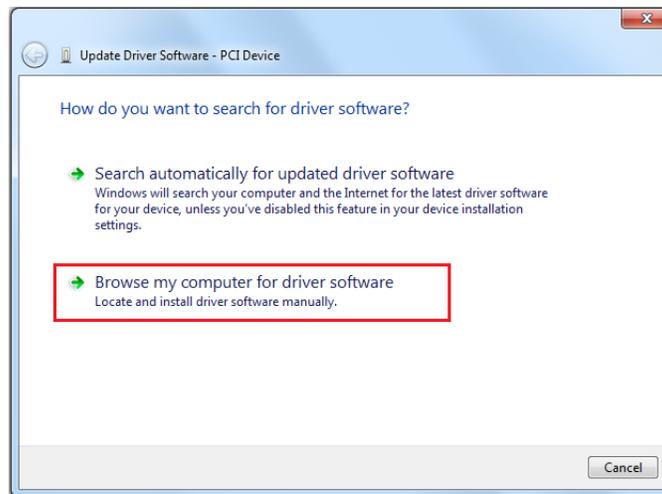
Figure 29 • Update Driver Software



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

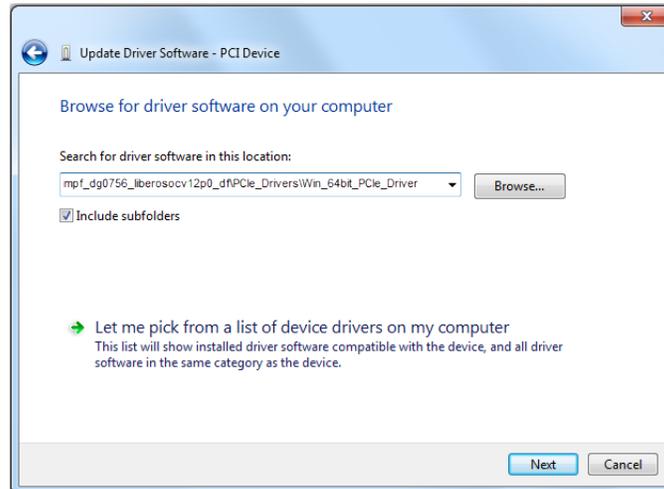
2. In the **Update Driver Software - PCI Device** window, select **Browse my computer for driver software** as shown in the following figure.

Figure 30 • Browse for Driver Software



3. Browse the drivers folder: `mpf_dg0756_dfPCIe_Drivers\Win_64bit_PCIe_Driver` and click **Next** as shown in the following figure.

Figure 31 • 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 33, page 27.

Figure 32 • Windows Security

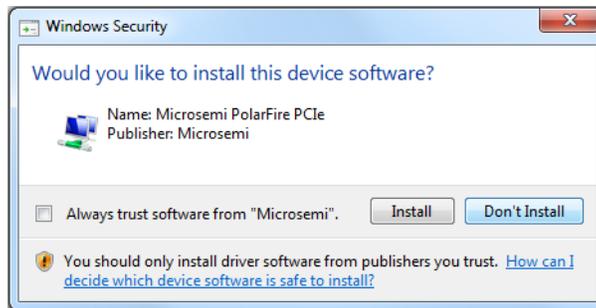
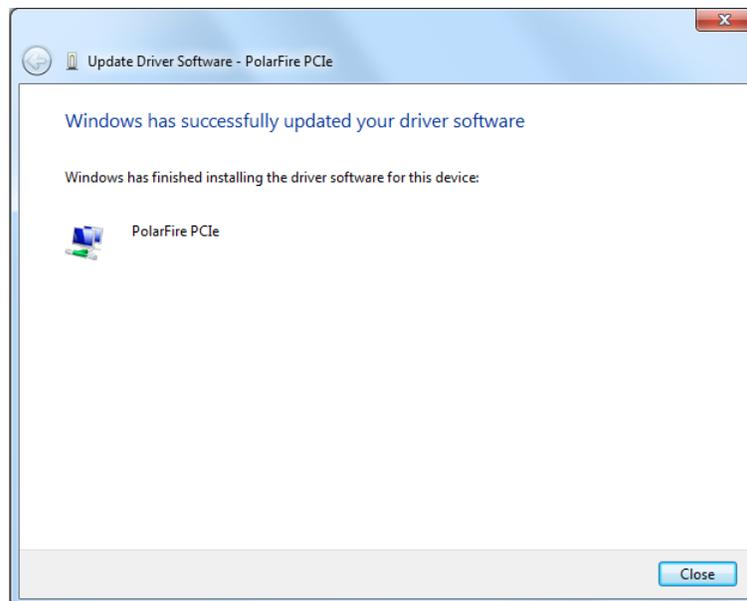


Figure 33 • Successful Driver Installation

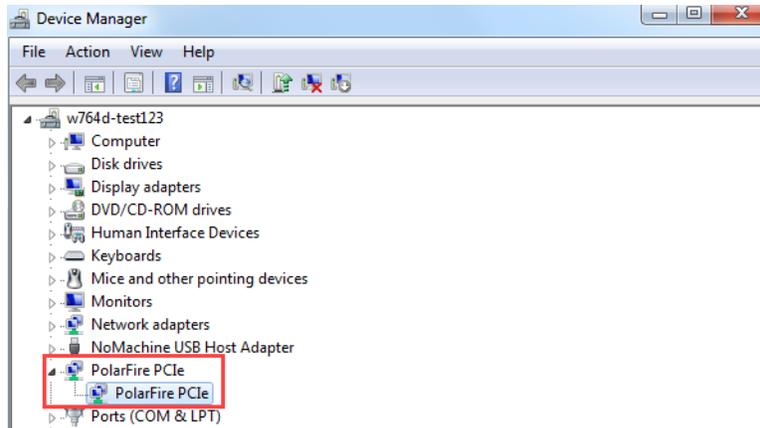


4.2.3 Running the PCIe Demo Application

To run the demo design, perform the following steps:

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

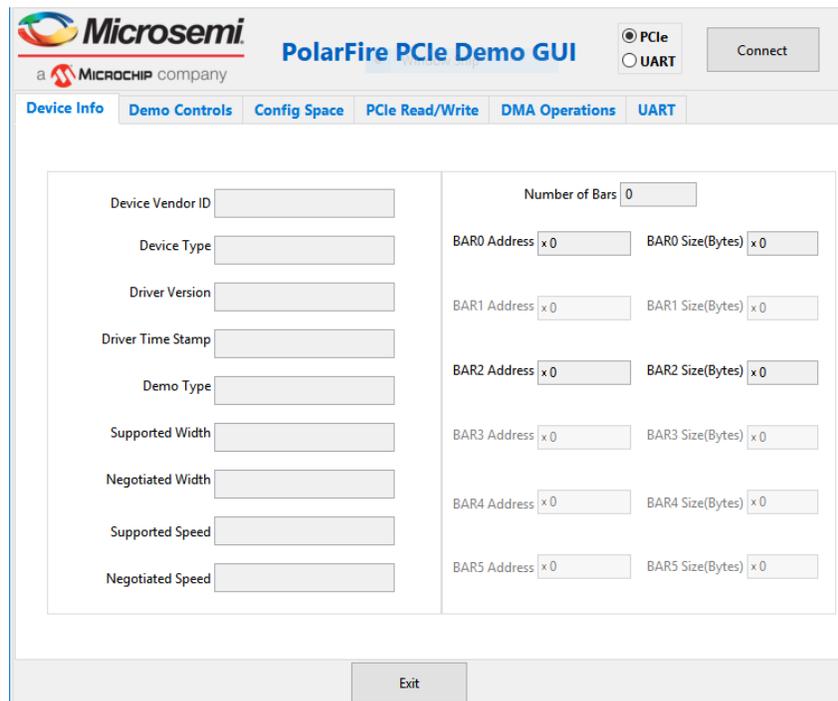
Figure 34 • Device Manager—PCIe Device Detection



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

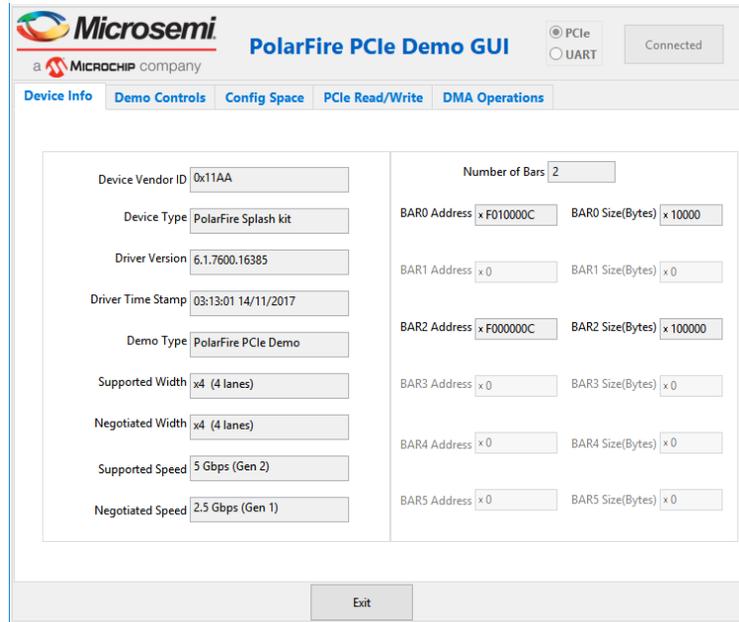
2. Go to **All Programs > PolarFire_PCIE_GUI > PolarFire_PCIE_GUI**. The **PolarFire PCIe Demo** window is displayed as shown in the following figure.

Figure 35 • PCIe EndPoint Demo Application



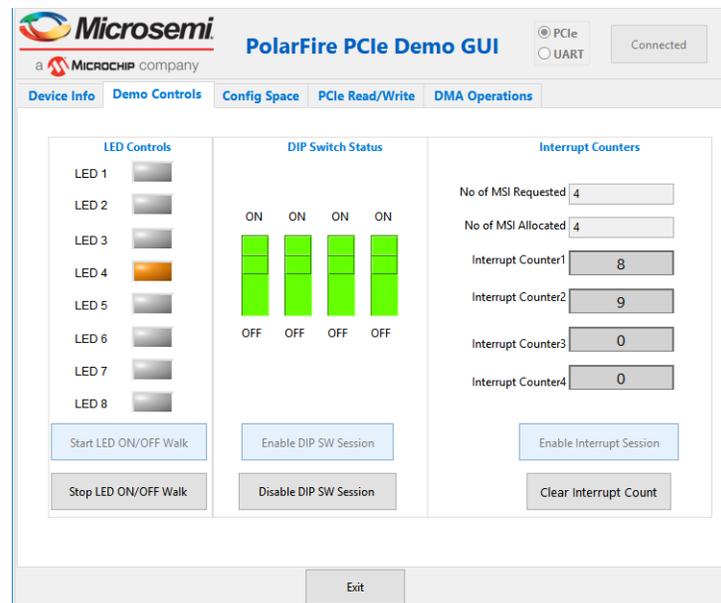
- Click **Connect**. The application detects and displays the information related to the connected kit such as Device Vendor ID, Device Type, Driver Version, Driver Time Stamp, Demo Type, Supported Width, Negotiated Width, Supported Speed, Negotiated Speed, Number of Bars, and BAR Address as shown in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

Figure 36 • Device Info



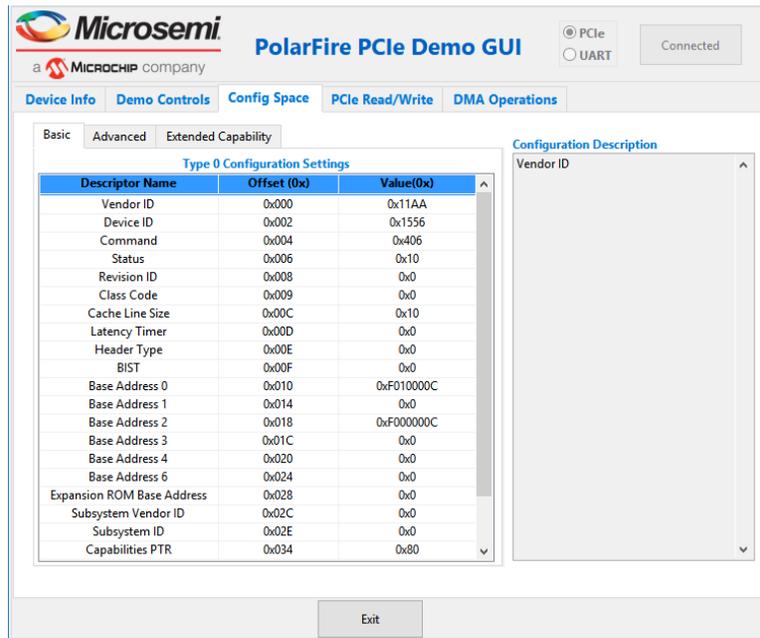
- Click **Demo Controls** tab to display the **LED Controls**, **DIP Switch Status**, and **Interrupt Counters**.
- Click **Start LED ON/OFF Walk**, **Enable DIP SW Session**, and **Enable Interrupt Session** to view the controlling LEDs (observe LED4 to LED11 on the PolarFire Evaluation Kit and LED1 to LED8 on the PolarFire Splash Kit), getting the DIP switch (ON/OFF the DIP1 to DIP4 on the PolarFire Evaluation/Splash Kit) status, and monitoring the interrupts (press SW7 to SW10 on the PolarFire Evaluation Kit and SW3 to SW6 on the PolarFire Splash Kit to generate interrupt) simultaneously as shown in the following figure.

Figure 37 • Demo Controls



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

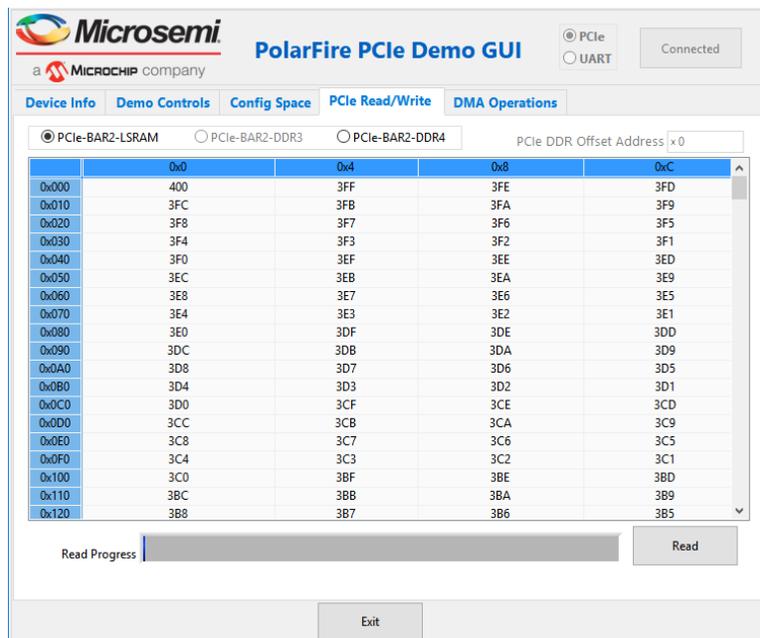
Figure 38 • Configuration Space



- Click **PCIe Read/Write** tab to perform read and write operations to DDR/LSRAM using BAR2 space.
- Select **LSRAM/DDR3/DDR4** and then click **Read** to read the 4 KB memory mapped to BAR2 space for DDR and LSRAM as shown in the following figure.

Note: PCIe BAR2-DDR3 is applicable only for Evaluation Kit.

Figure 39 • PCIe BAR2 Memory Access—LSRAM



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

Note: DDR3 DMA options are not applicable for Splash kit demo.

4.2.3.1 Continuous DMA—Operations

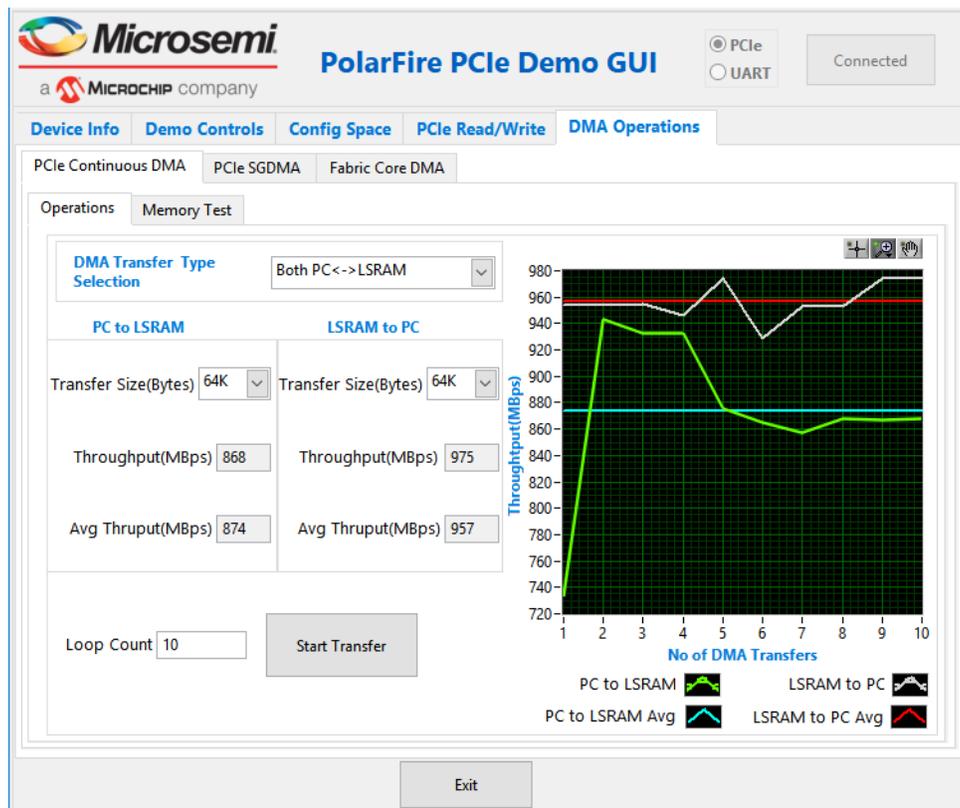
The following instructions describe running DMA operations between PC and DDR3, PC and DDR4, PC and LSRAM:

- Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
 - PC->DDR3**—to transfer the data from host PC to PolarFire DDR3 memory
 - DDR3->PC**—to transfer the data from PolarFire DDR3 memory to host PC
 - Both- PC<->DDR3**—to transfer the data from host PC to and from PolarFire DDR3 memory
 - 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
- Select **Transfer Size** (4 KB to 64 KB) from the drop-down list. The maximum contiguous DMA size is 64 KB because the host PC may not have a contiguous memory of more than 64 KB. For DMA operations that require more than 64 KB, use SGDMA.
- Enter the **Loop Count** in the box.
- Click **Start Transfer**. After a successful DMA operation, the GUI displays the throughput and average throughput in MBps.

Note: The AXI LSRAM in the design is configured for 4 KB. This 4 KB is over written if more than 4 KB 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 40 • Continuous DMA Operations with DMA Transfer Type Selection as Both PC and LSRAM



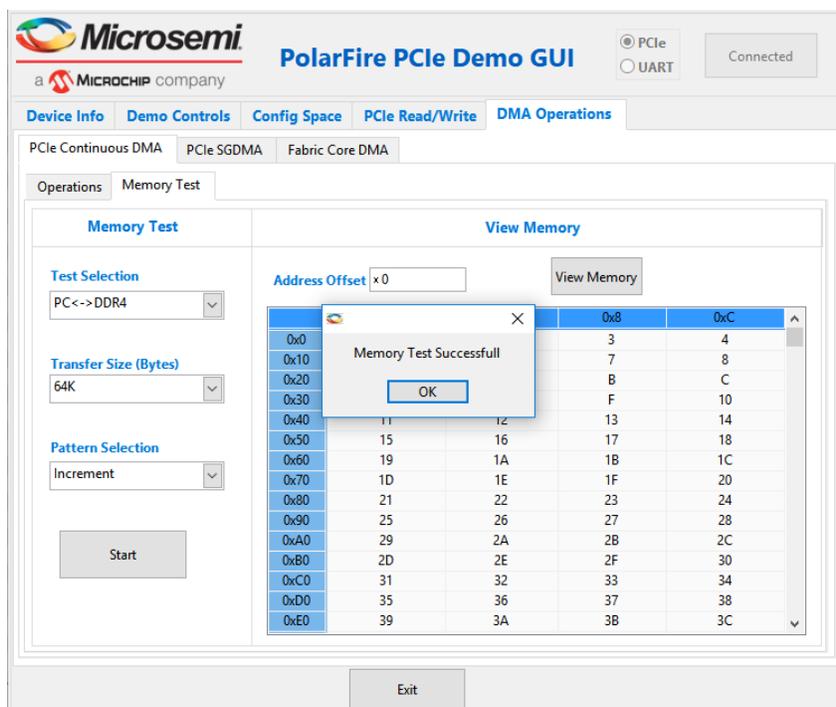
4.2.3.2 Continuous DMA—Memory Test

The following instructions describe running **Memory Test** between PC and DDR3/DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
 - **PC<->DDR3**—to transfer the data from host PC to and from PolarFire DDR3 memory
 - **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** (4 KB to 64 KB) 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.
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 41 • 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.

4.2.3.3 SGDMA—Operations

The following instructions describe running SGDMA operations between PC and DDR3, PC and DDR4:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
 - **PC -> DDR3**—to transfer the data from host PC to PolarFire DDR3 memory
 - **DDR3-> PC**—to transfer the data from PolarFire DDR3 memory to host PC
 - **Both PC <->DDR3**—to transfer the data from host PC to and from PolarFire DDR3 memory
 - **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** (4 KB to 64 KB) 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.

4.2.3.4 SGDMA—Memory Test

The following instructions describe running **Memory Test** between PC and DDR3/DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
 - **PC<->DDR3**—to transfer the data from host PC to and from PolarFire DDR3 memory
 - **PC<->DDR4**—to transfer the data from host PC to and from PolarFire DDR4 memory
2. Select **Transfer Size** (4 KB to 1 MB) 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.
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.

5. Click **OK**.

4.2.3.5 Core DMA—Operations

The following instructions describe running DMA operations between LSRAM and DDR3, LSRAM and DDR4, DDR3 and DDR4:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
 - **LSRAM -> DDR3**—to transfer the data from LSRAM to PolarFire DDR3 memory
 - **DDR3-> LSRAM**—to transfer the data from PolarFire DDR3 memory to LSRAM
 - **Both LSRAM <->DDR3**—to transfer the data from LSRAM to and from PolarFire DDR3 memory
 - **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
 - **DDR4 -> DDR3**—to transfer the data from DDR4 to DDR3 memory
 - **DDR3 -> DDR4**—to transfer the data from DDR3 to DDR4 memory
 - **Both DDR4 <-> DDR3**—to transfer the data from DDR4 to and from DDR3 memory
2. Select **Transfer Size** (4 KB to 1 MB) from the drop-down list.
3. Enter the **Loop Count** in the box.
4. Click **Start Transfer**. After a successful DMA operation, the GUI displays the throughput and average throughput in MBps.

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

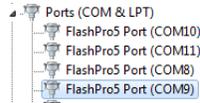
5. Click **Exit**.

4.3 Running the Demo Through UART

The following steps describe 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 of UART ports in the **Device Manager** window.

Figure 42 • Device Manager—UART Ports



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

1. Go to **All Programs > PolarFire_PcGui > PolarFire_PcGui**. The **PolarFire PCIe Demo** window is displayed.
2. Select **UART** radio button and click **Connect**.

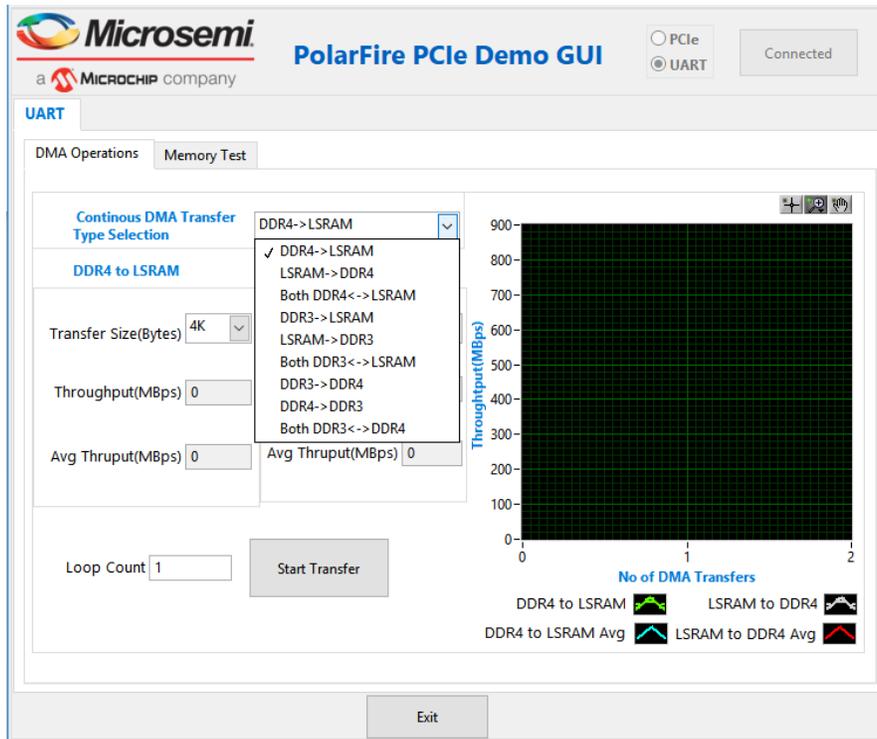
The GUI application scans for UART port and after successful connection, displays the DMA Operations UART tab as shown in Figure 43, page 35.

4.3.1 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:
 - **DDR3 -> LSRAM**: to transfer the data from DDR3 to PolarFire LSRAM memory.
 - **LSRAM -> DDR3**: to transfer the data from PolarFire LSRAM memory to DDR3.
 - **Both DDR3 <-> LSRAM**: to transfer the data from DDR3 to and from PolarFire LSRAM memory.
 - **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
 - **DDR4 -> DDR3**—to transfer the data from DDR4 to DDR3 memory
 - **DDR3 -> DDR4**—to transfer the data from DDR3 to DDR4 memory
 - **Both DDR4 <-> DDR3**—to transfer the data from DDR4 to and from DDR3 memory
 - **Both DDR3<->DDR4**: to transfer the data from DDR3 to and from DDR4 memory.
2. Select **Transfer Size** (4 KB to 512 KB) 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 43 • UART—DMA Operations



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

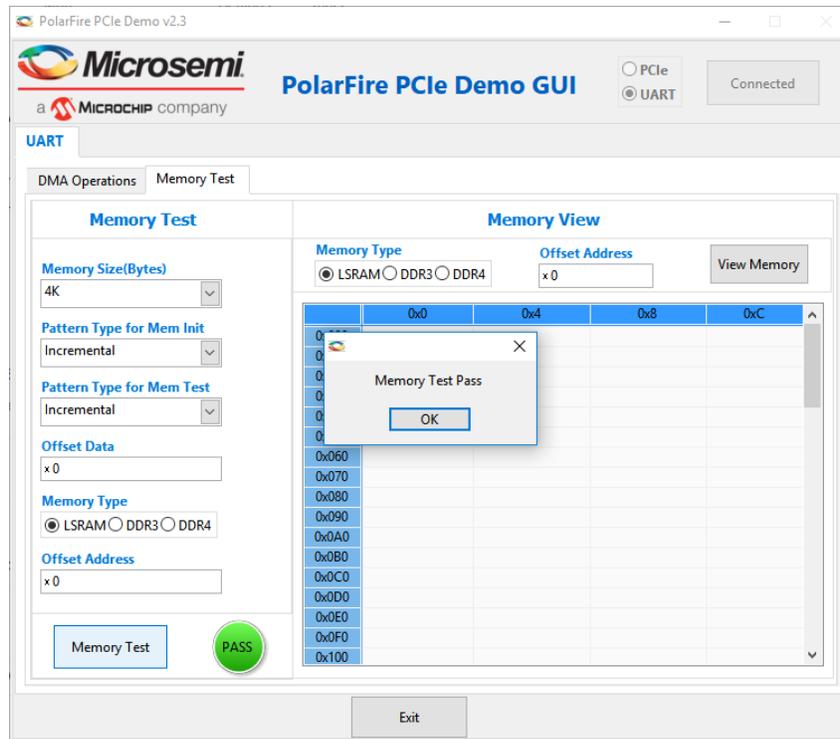
4.3.1.1 UART—Memory Test

The following instructions describe running **Memory Test** between PC and DDR3/DDR4/LSRAM:

1. Select **Transfer Size** (4 KB to 1 MB) 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/DDR3/DDR4 memory
 - GUI sends command to fabric logic to read and compare LSRAM/DDR3/DDR4 memory

The following figure shows UART—Memory Test tab.

Figure 44 • UART—Memory Test



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

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

4.3.2 Running the Demo Design on Linux

The following steps describe how to run the demo design on Linux:

1. Switch **ON** the power supply switch on the PolarFire Evaluation Kit board or PolarFire Splash Kit.
2. Switch **ON** the CentOS Linux Host PC.
3. CentOS Linux Kernel detects the PolarFire Evaluation Kit board or PolarFire Splash Kit PCIe end point as Actel Device.
4. On Linux Command Prompt Use `lspci` command to display the PCIe info.
`lspci`

Figure 45 • PCIe Device Detection

```
[root@localhost linux_pcie_driver]# lspci
00:00.0 Host bridge: Intel Corporation Xeon E5/Core i7 DMI2 (rev 07)
00:01.0 PCI bridge: Intel Corporation Xeon E5/Core i7 II0 PCI Express Root Port 1a (rev 07)
00:02.0 PCI bridge: Intel Corporation Xeon E5/Core i7 II0 PCI Express Root Port 2a (rev 07)
00:03.0 PCI bridge: Intel Corporation Xeon E5/Core i7 II0 PCI Express Root Port 3a in PCI Express M
00:04.0 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 0 (rev 07)
00:04.1 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 1 (rev 07)
00:04.2 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 2 (rev 07)
00:04.3 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 3 (rev 07)
00:04.4 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 4 (rev 07)
00:04.5 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 5 (rev 07)
00:04.6 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 6 (rev 07)
00:04.7 System peripheral: Intel Corporation Xeon E5/Core i7 DMA Channel 7 (rev 07)
00:05.0 System peripheral: Intel Corporation Xeon E5/Core i7 Address Map, VTd_Misc, System Managem
00:05.2 System peripheral: Intel Corporation Xeon E5/Core i7 Control Status and Global Errors (rev
00:05.4 PIC: Intel Corporation Xeon E5/Core i7 I/O APIC (rev 07)
00:11.0 PCI bridge: Intel Corporation C600/X79 series chipset PCI Express Virtual Root Port (rev 05)
00:16.0 Communication controller: Intel Corporation C600/X79 series chipset MEI Controller #1 (rev
00:16.2 IDE interface: Intel Corporation C600/X79 series chipset IDE-r Controller (rev 05)
00:16.3 Serial controller: Intel Corporation C600/X79 series chipset KT Controller (rev 05)
00:19.0 Ethernet controller: Intel Corporation 82579LM Gigabit Network Connection (Lewisville) (rev
00:1a.0 USB controller: Intel Corporation C600/X79 series chipset USB2 Enhanced Host Controller #2
00:1b.0 Audio device: Intel Corporation C600/X79 series chipset High Definition Audio Controller (r
00:1c.0 PCI bridge: Intel Corporation C600/X79 series chipset PCI Express Root Port 2 (rev b5)
00:1c.5 PCI bridge: Intel Corporation C600/X79 series chipset PCI Express Root Port 5 (rev b5)
00:1c.6 PCI bridge: Intel Corporation C600/X79 series chipset PCI Express Root Port 3 (rev b5)
00:1c.7 PCI bridge: Intel Corporation C600/X79 series chipset PCI Express Root Port 4 (rev b5)
00:1d.0 USB controller: Intel Corporation C600/X79 series chipset USB2 Enhanced Host Controller #1
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev a5)
00:1f.0 ISA bridge: Intel Corporation C600/X79 series chipset LPC Controller (rev 05)
00:1f.2 RAID bus controller: Intel Corporation C600/X79 series chipset SATA RAID Controller (rev 05)
00:1f.3 SMBus: Intel Corporation C600/X79 series chipset SMBus Host Controller (rev 05)
02:00.0 Serial Attached SCSI controller: Intel Corporation C602 chipset 4-Port SATA Storage Control
04:00.0 Non-VGA unclassified device: Actel Device 1556
05:00.0 VGA compatible controller: NVIDIA Corporation GK107GL [Quadro K600] (rev a1)
```

4.3.2.1 Drivers Installation

Enter the following commands in the Linux command prompt to install the PCIe drivers:

1. Change to `g5_endpoint_driver` directory by using the following command:


```
#cd <source_code_directory>/ g5_endpoint_driver/linux_pcie_driver
```
2. Enter the `make` command on Linux Command Prompt to compile the Linux PCIe device driver code.


```
#make clean [To clean any *.o, *.ko files]
#make
```
3. The kernel module, `mpci.ko`, is created in the same directory.
4. Enter `insmod` command to insert the Linux PCIe device driver as a module.


```
#insmod mpci.ko
```

Note: Root privileges are required to execute this command.

Figure 46 • PCIe Device Driver Installation

```
[root@localhost linux_pcie_driver]# make
make -W -Wall -Wstrict-prototypes -Wmissing-prototypes -C /lib/modules/3.10.0-1062.7.1.el7.x86_64/build SUBDIRS=/root/linux_pcie_driver modules
make[1]: Entering directory /usr/src/kernels/3.10.0-1062.7.1.el7.x86_64
  CC [M] /root/linux_pcie_driver/mpci.o
  CC [M] /root/linux_pcie_driver/mdma.o
  CC [M] /root/linux_pcie_driver/misrdpc.o
  LD [M] /root/linux_pcie_driver/mpci.o
Building modules, stage 2.
MODPOST 1 modules
  CC /root/linux_pcie_driver/mpci.mod.o
  LD [M] /root/linux_pcie_driver/mpci.ko
make[1]: Leaving directory /usr/src/kernels/3.10.0-1062.7.1.el7.x86_64
[root@localhost linux_pcie_driver]# insmod mpci.ko
[root@localhost linux_pcie_driver]# ls /dev/MS_PCI_DEV
/dev/MS_PCI_DEV
```

4.3.2.1.1 Linux PCIe Application Compilation

1. Compile the Linux user space application as follows:

```
# cd <source_code_directory>/ g5_endpoint_driver/linux_pcie_app
# make all
```

After successful compilation, Linux PCIe application utility `pcie_app` creates in the same directory.

2. On Linux Command Prompt, run the `pcie_app` utility as:

```
# ./pcie_app
```

Help menu is displayed, as shown in the following figure.

Figure 47 • Linux PCIe Application Utility

```
[root@localhost linux_pcie_app]# ls
Makefile pcie_appln_dma.c pcie_appln.h pcie_appln_main.c
[root@localhost linux_pcie_app]# make
gcc -c pcie_appln_main.c -o pcie_appln_main.o
gcc -c pcie_appln_dma.c -o pcie_appln_dma.o
gcc pcie_appln_main.o pcie_appln_dma.o -Wall -lm -o pcie_app
[root@localhost linux_pcie_app]# ls
Makefile pcie_app pcie_appln_dma.c pcie_appln_dma.o pcie_appln.h pcie_appln_main.c pcie_appln_main.o
[root@localhost linux_pcie_app]# ./pcie_app
welcome to PCI Demo

-----
PCI Device [/dev/MS_PCI_DEV] opened
1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
```

4.3.2.1.2 Device Information

Enter 1 to get device information.

Figure 48 • Device Information

```

-----
PCI Device [/dev/MS_PCI_DEV] opened
1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
1
demo_type = PolarFire PCIe Demo
device_status = Microsemi Device Detected
device_type = PolarFire Evaluation kit
number of BARs enabled = 2
bar0_add      = 0xe2b0000c
bar0_size     = 10000
bar2_add      = 0xe2a0000c
bar2_size     = 100000

```

Note: PCI device enumeration takes place during the boot time and base address register starting address will not be the same for every boot.

4.3.2.1.3 Blink LEDs

Enter 2 to blink leds.

Figure 49 • Blink LEDs

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
2
-----
Enter Led data
55
Blink LEDs success

```

4.3.2.1.4 DIP Switch Status

Enter 3 to get dip switch status.

Figure 50 • Dip Switch Status

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
3
-----
Blink LEDs success

SW1 : ON
SW2 : ON
SW3 : ON
SW4 : ON

```

4.3.2.1.5 For Interrupt

Enter **4** then enter 1 for interrupt count or 2 for clear.

Figure 51 • For Interrupt

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
4
-----
1. for ISR count, 2. for ISR clean
1
counter 1 = 0
counter 2 = 0
counter 3 = 0
counter 4 = 0

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
4
-----
1. for ISR count, 2. for ISR clean
1
counter 1 = 0
counter 2 = 0
counter 3 = 0
counter 4 = 1

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
4
-----
1. for ISR count, 2. for ISR clean
2
counter 1 = 0
counter 2 = 0
counter 3 = 0
counter 4 = 0

```

4.3.2.1.6 For Read/Write to Bar Space

Enter **5** to read or write to bar space.

Figure 52 • For Bar Read/Write

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation

5
-----
provide type 1.DDR-3, 2.LSRAM,3.DDR-4
1
1. Read from bar space
2. Write to bar space
2
Enter the offset
0x10
Provide the data to write
0x0a
Write successful

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
5
-----
provide type 1.DDR-3, 2.LSRAM,3.DDR-4
1
1. Read from bar space
2. Write to bar space
1
Enter the offset
0x10
read value = 0xa

```

4.3.2.1.7 For PCIe configspace

Enter **6** then enter **2** for full pcie configuration read.

Figure 53 • For PCIe configspace

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA operation
6
-----
1.Read from PCIe ConfigSpace offset
2.Read PCIe ConfigSpace
2
Vendor ID: 0x11aa
Device ID: 0x1556
Cmd Reg: 0x506
Stat Reg: 0x10
Revision ID: 0x0
Class Prog: 0x0
Device Class: 0x0
Cache Line Size: 0x10
Latency Timer: 0x0
Header Type: 0x0
BIST: 0x0
BAR0: Addr:0xe2b0000c
BAR1: Addr:0x0
BAR2: Addr:0xe2a0000c
BAR3: Addr:0x0
BAR4: Addr:0x0
BAR5: Addr:0x0
CardBus CIS Pointer: 0x0
Subsystem Vendor ID: 0x0
Subsystem Device ID: 0x0
Expansion ROM Base Address: 0x0
IRQ Line: 0x3
IRQ Pin: 0x4
Min Gnt: 0x0
Max Lat: 0x0
MSIEnable: 0x1
MultipleMessageCapable: 0x4
MultipleMessageEnable: 0x4
CapableOf64Bits: 0x1
PerVectorMaskCapable: 0x0

```

4.3.2.1.8 For PCIe DMA Operation

Enter 7 for PCIe DMA operation.

- Maximum transfer length is $4k < \text{transfer length} < 1 \text{ MB}$.
- For DDR3/4 maximum transfer length is $4k < \text{transfer length} < 1 \text{ MB}$.
- For LSRAM maximum transfer length is $4k < \text{transfer length} < 64k\text{B}$.

Figure 54 • For DMA Operation

```

1. Device info
2. Blink LEDs
3. Dip switch status
4. For Interrupt
5. Read/Write to bars space
6. PCIe configSpace information
7. PCIe DMA opertion
7
-----
Read/write to memeory using DMA
-----
1. G5 Continuous Dma Write to DDR3
2. G5 Continuous Dma Read from DDR3
3. G5 Continuous Dma write/read DDR3
4. G5 Continuous Dma Write to DDR4
5. G5 Continuous Dma Read from DDR4
6. G5 Continuous Dma write/read DDR4
7. G5 Continuous Dma Write to LSRAM
8. G5 Continuous Dma Read from LSRAM
9. G5 Continuous Dma write/read LSRAM
10. G5 SG Dma Write to DDR3
11. G5 SG Dma Read from DDR3
12. G5 SG Dma write/read DDR3
13. G5 SG Dma Write to DDR4
14. G5 SG Dma Read from DDR4
15. G5 SG Dma write/read DDR4

```

Enter 3 for Continuous DMA write/read DDR3.

Figure 55 • For Continuous DMA write/read DDR3

```

3
Tx size
4096
Rx size
4096
Enter the data pattern 1.inc 2. dec 3.rand 4.zero's 5.one's 6. AAA 7. 555
1
Write throughput = 648 MBPS
Read throughput = 412 MBPS

```

4.4 Throughput Summary of Evaluation Kit

The following tables list the throughput values observed.

Table 6 • PolarFire Throughput Summary—PCIe Continuous DMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to LSRAM	64 K	1091	1069
LSRAM to PC		1104	1147
Both PC to and from LSRAM		1124/1154	1093/1149
PC to DDR4	64 K	998	970
DDR4 to PC		523 ¹	523
Both PC to and from DDR4		998/523	972/523
PC to DDR3	64 K	468	460
DDR3 to PC		327 ¹	327
Both PC to and from DDR3		468/327	465/327

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 DDR3/DDR4.

Table 7 • PolarFire Throughput Summary—PCIe SGDMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to DDR4	1 MB	986	984
DDR4 to PC		524 ¹	524
Both PC to and from DDR4		986/524	980/524
PC to DDR3	1 MB	469	472
DDR3 to PC		325 ¹	325
Both PC to and from DDR3		473/325	473/325

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 DDR3/DDR4.

Table 8 • PolarFire Throughput Summary—Fabric Core DMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
LSRAM to DDR4	1 MB	1470	1470
DDR4 to LSRAM		1245	1245
Both LSRAM to and from DDR4		1470/1245	1470/1245
LSRAM to DDR3	1 MB	574	574
DDR3 to LSRAM		553	553
Both LSRAM to and from DDR3		574/554	574/554
DDR4 to DDR3	1 MB	574	574
DDR3 to DDR4		553	553
Both DDR4 to and from DDR3		574/553	574/553

Note: DDR3 throughput is less due to AXI interconnect CDC path limitation.

4.5 Throughput Summary of Splash Kit

The following table lists the throughput values observed.

Table 9 • PolarFire Throughput Summary—PCIe Continuous DMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to LSRAM	64 K	1268	1270
LSRAM to PC		1156	1156
Both PC to and from LSRAM		1271/1145	1236/1152
PC to DDR4	64 K	1050	1187
DDR4 to PC		527 ¹	527
Both PC to and from DDR4		1242/528	1231/528

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.

Table 10 • PolarFire Throughput Summary—PCIe SGDMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
PC to DDR4	1 MB	1215	1201
DDR4 to PC		528 ¹	528
Both PC to and from DDR4		1216/528	1207/528

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.

Table 11 • PolarFire Throughput Summary—Fabric Core DMA Mode

DMA Transfer Type	DMA Size	Throughput (MBps)	Average Throughput (MBps)
DDR4 to LSRAM	1 MB	1259	1258
LSRAM to DDR4		1470	1470
Both LSRAM to and from DDR4		1257/1470	1258/1470

5 Appendix 1: DDR3 and DDR4 Power Measurement

5.1 DDR3

The PolarFire Evaluation Kit board has a current sense resistor (R118) for 1.5 V power rail. Measure the voltage across the R118 using test points TP134 and TP135 and use the following equations to get the DDR3 power. This measurement includes PolarFire DDR3 IO power consumption and actual Micron DDR3 memory power consumption.

$$\text{Current(mA)} = \frac{\text{MeasureVoltage(mV)}}{R}$$

$$\text{Power(mW)} = \text{Current} \times \text{Voltage}$$

While running the demo, the measured voltage across R118 is 6.3 mV and resistor value is 0.01 Ω .

$$\begin{aligned}\text{Current (mA)} &= 6.3 / 0.01 = 630 \text{ mA} \\ \text{Power} &= 630 \times 1.5 = 945 \text{ MW}\end{aligned}$$

5.2 DDR4

The PolarFire Evaluation Kit board has a current sense resistor (R222) for 1.2 V power rail. Measure the voltage across the R222 using test points TP132 and TP133 and use the following equations to get the DDR4 power. This measurement includes PolarFire DDR4 IO power consumption and actual Micron DDR4 memory power consumption.

$$\text{Current(mA)} = \frac{\text{MeasureVoltage(mV)}}{R}$$

$$\text{Power(mW)} = \text{Current} \times \text{Voltage}$$

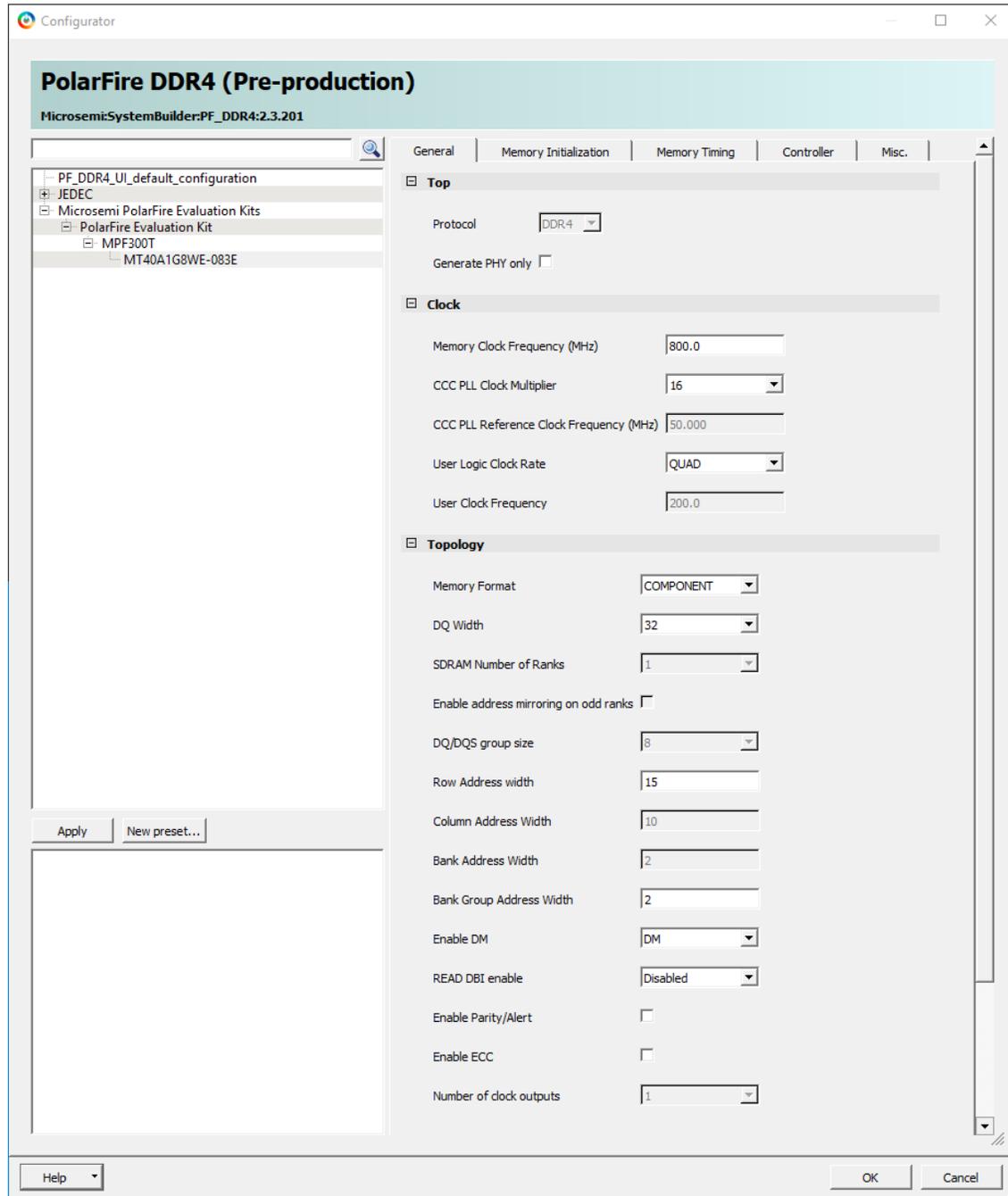
While running the demo, the measured voltage across R222 is 2.4 mV and resistor value is 0.01 Ω .

$$\begin{aligned}\text{Current (mA)} &= 2.4 / 0.01 = 240 \text{ mA} \\ \text{Power} &= 240 \times 1.2 = 288 \text{ MW}\end{aligned}$$

6 Appendix 2: DDR4 Configuration

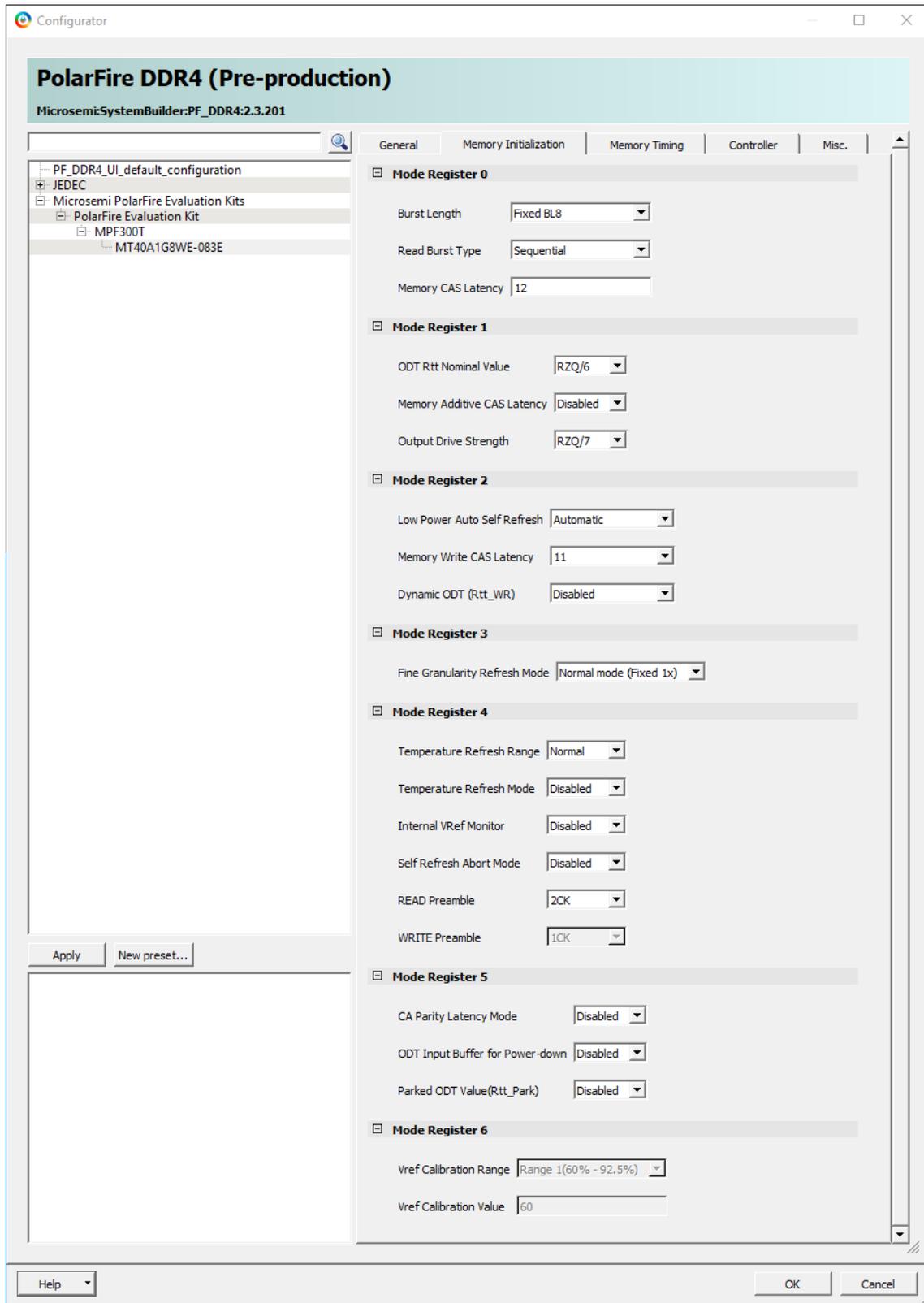
The DDR4 subsystem is configured to access the 32-bit DDR4 memory through an AXI4 64-bit interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4 memory on the PolarFire Evaluation kit. The following figures show general configuration settings for the DDR4 memory.

Figure 56 • DDR4 Configurator



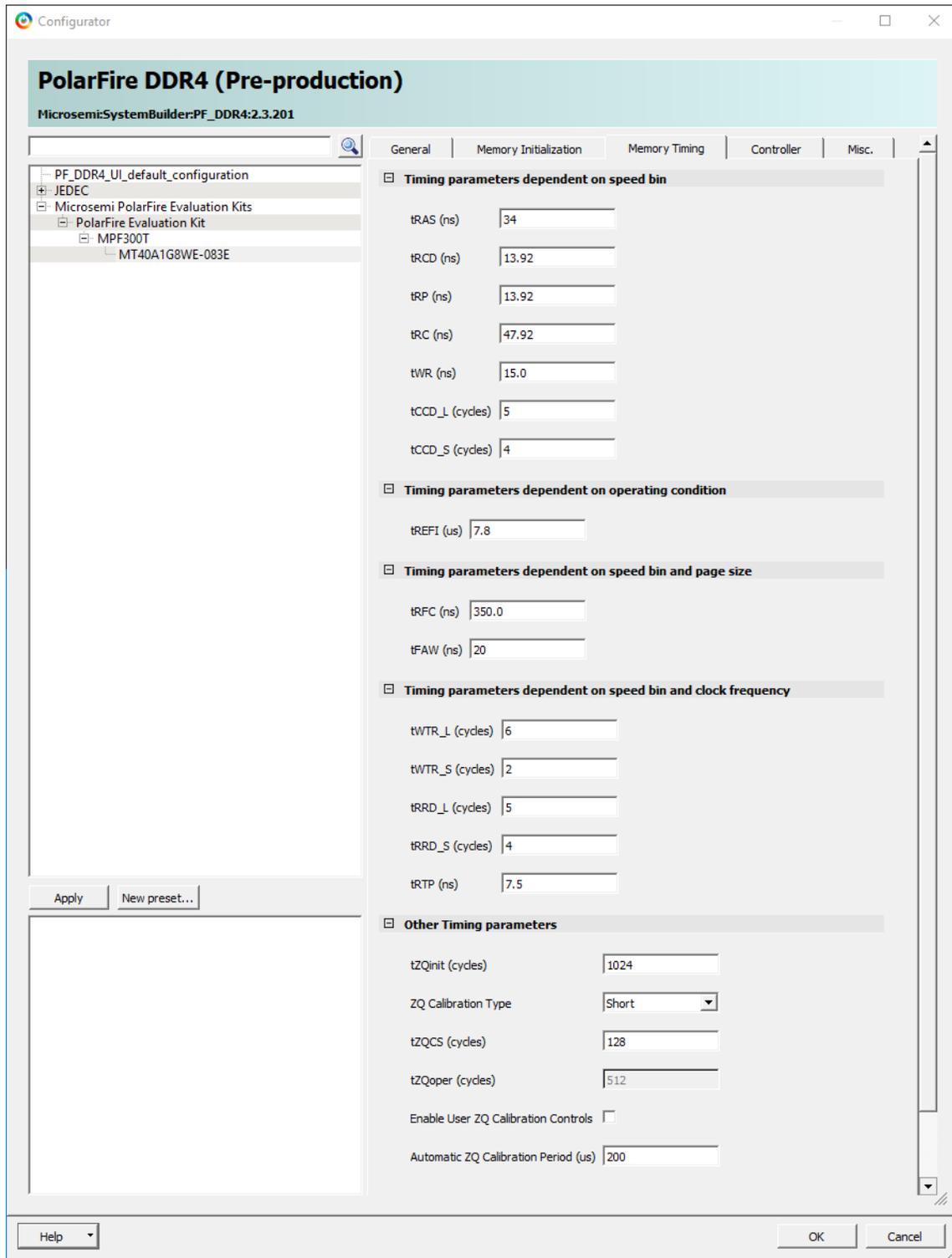
The following figure shows initialization configuration settings for the DDR4 memory.

Figure 57 • DDR4 Configurator—Memory Initialization



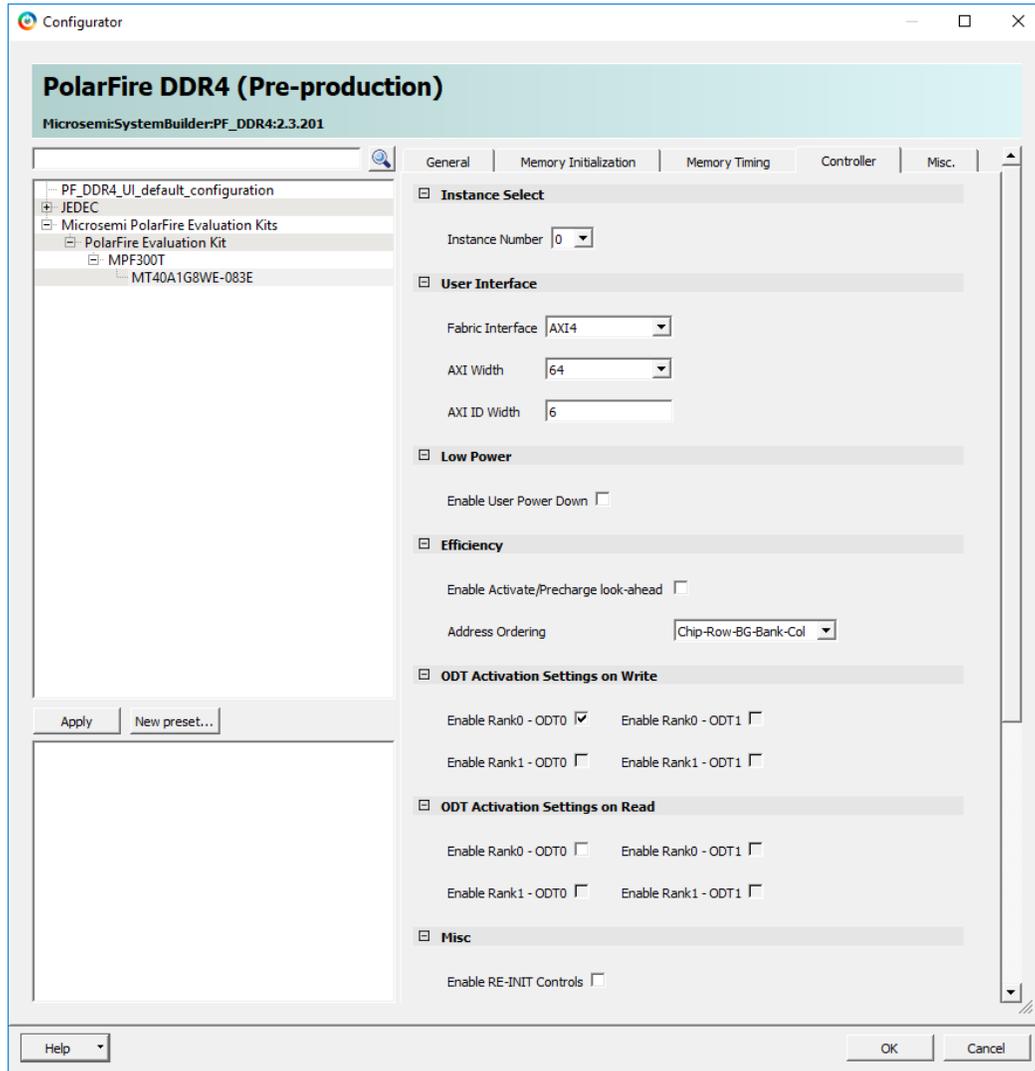
The following figure shows timing configuration settings for the DDR4 memory.

Figure 58 • DDR4 Configurator—Memory Timing



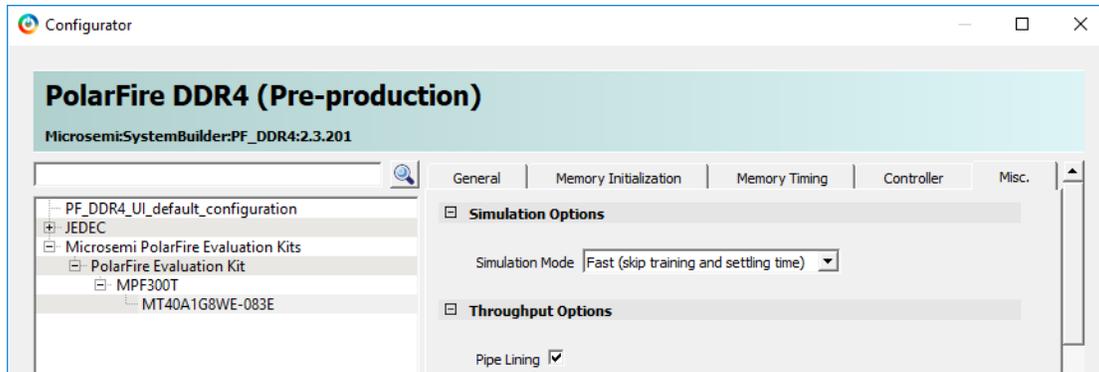
The following figure shows controller configuration settings for the DDR4 memory.

Figure 59 • DDR4 Configurator—Controller



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

Figure 60 • DDR4 Configurator—Misc



7 Appendix 3: Programming the Device Using FlashPro Express

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

```
mpf_dg0756_df\Programming_Job
```

To program the device, perform the following steps:

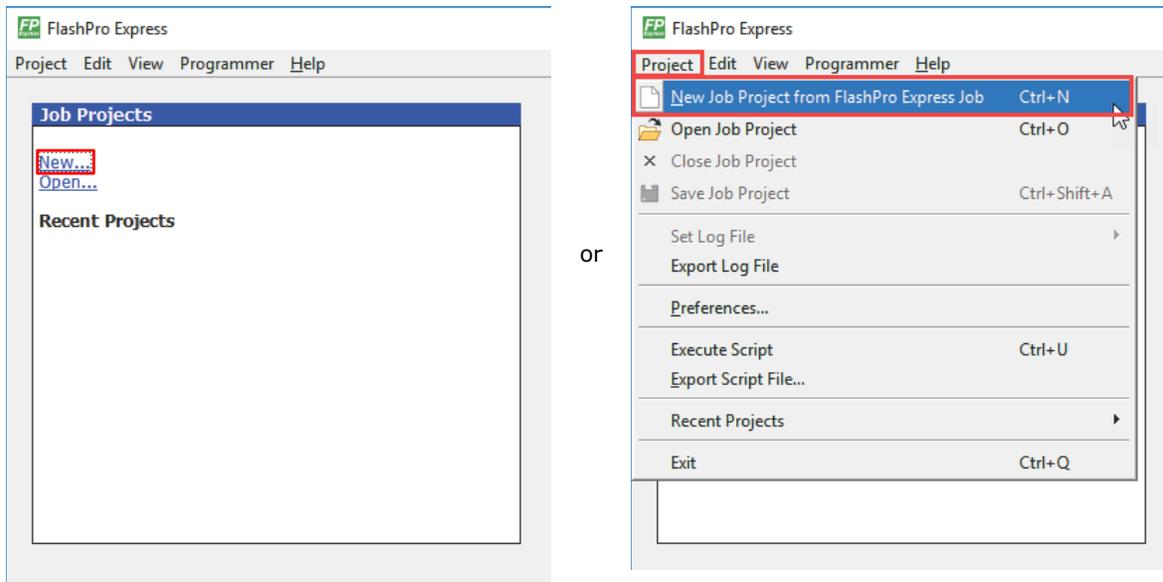
1. Ensure that the jumper settings on the board are the same as listed in Table 4, page 20 (for evaluation) and Table 5, page 21 (for splash).

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

2. Connect the power supply cable to the **J9** connector on the Evaluation board or **J2** connector on the Splash board.
3. Connect the USB cable from the Host PC to the **J5** (FTDI port) on the Evaluation board or **J1** (FTDI port) on the Splash board.
4. Power on the board using the **SW3** slide switch on the Evaluation board or **SW1** slide switch on the Evaluation board.
5. On the host PC, launch the **FlashPro Express** software.
6. To create a new job, click **New** or

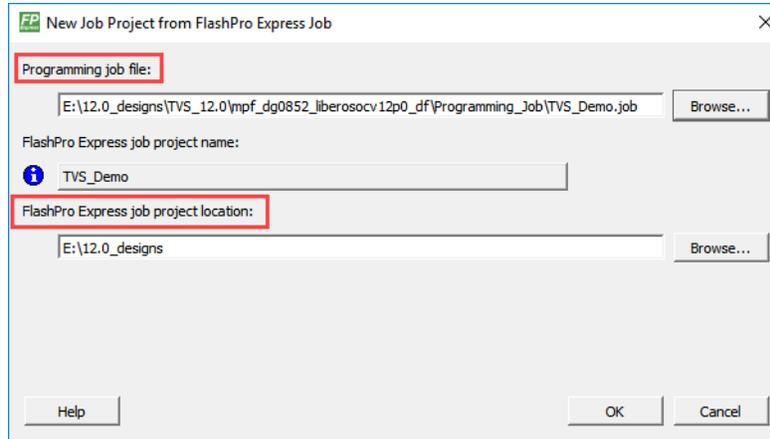
in the **Project** menu, select **New Job Project from FlashPro Express Job** as shown in the following figure.

Figure 61 • FlashPro Express Job Project



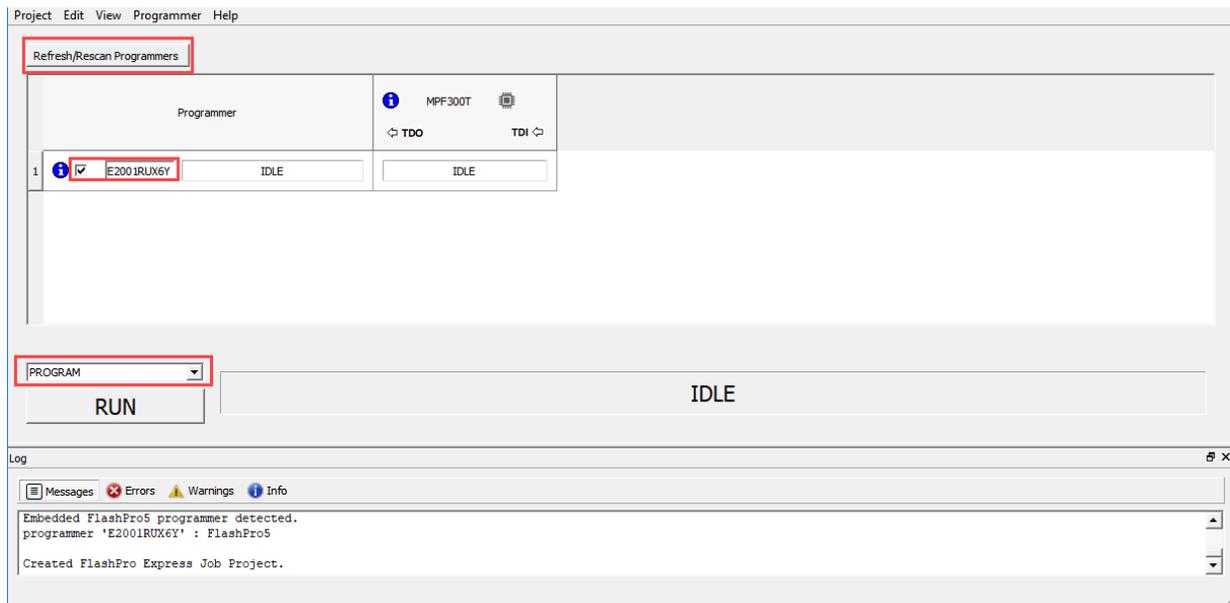
7. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
 - **Programming job file:** Click **Browse**, navigate to the location where the .job file is located, and select the file. The default location is: <download_folder>\mpf_dg0756_df\Programming_Job.
 - **FlashPro Express job project location:** Click **Browse** and navigate to the location where you want to save the project.

Figure 62 • New Job Project from FlashPro Express Job



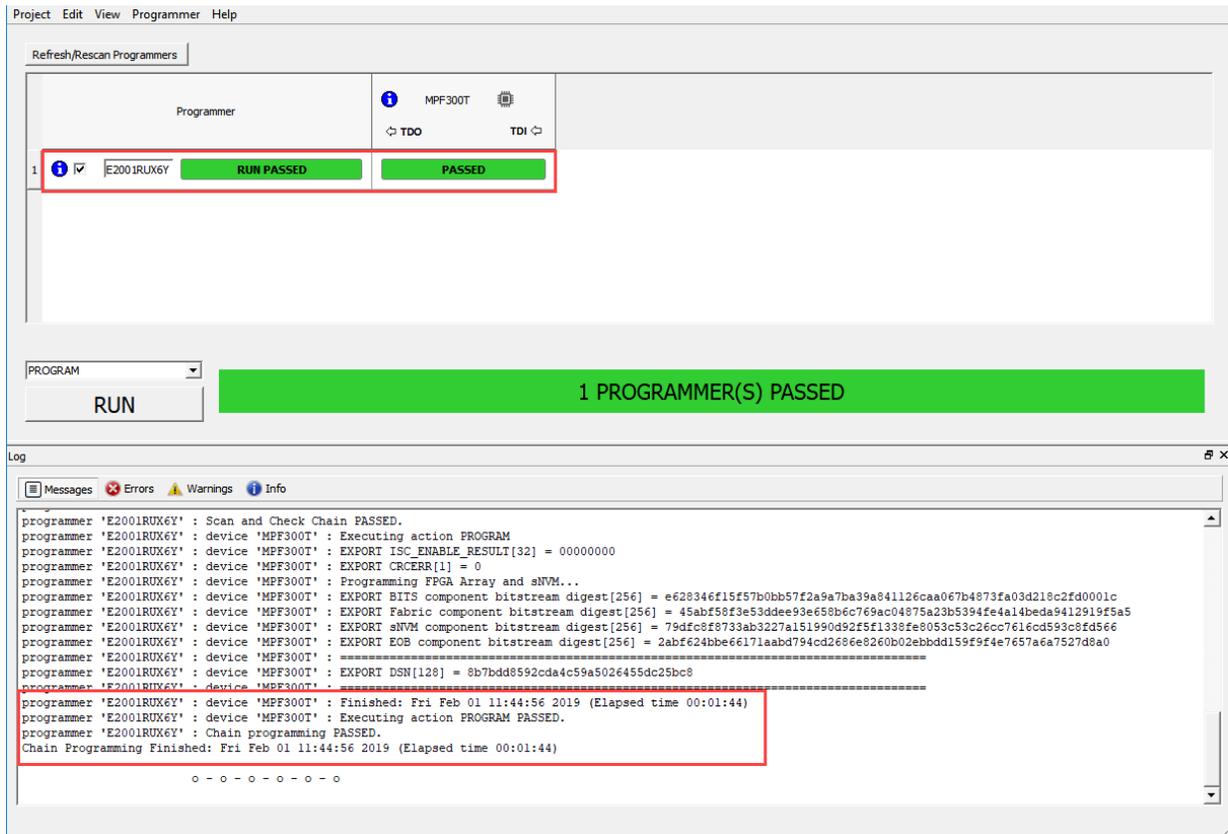
8. Click **OK**. The required programming file is selected and ready to be programmed in the device.
9. The FlashPro Express window appears, as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan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

Figure 63 • Programming the Device



10. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure. See *Running the Demo*, page 22 to run the PCIe EndPoint demo.

Figure 64 • FlashPro Express—RUN PASSED



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

8 Appendix 4: 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:

- `mpf_dg0756_eval_df/TCL_Scripts/readme.txt`
- `mpf_dg0756_splash_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.

9 Appendix 5: 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*.
- For more information about PF_CCC, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.
- For more information about DDR3/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 Evaluation Kit, see *UG0747: PolarFire FPGA Evaluation Kit User Guide*.
- 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*.