PolarFire FPGA PCIe EndPoint DDR3L DDR4 Memory Controller Data Plane Application Note

AN4597

Introduction (Ask a Question)

Microchip 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) and Advanced eXtensible Interface 4 (AXI4) access to the PCIe space, and the PCIe access to the memory-mapped AXI4 space. For more information, see PolarFire Family 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 PolarFire Family Memory Controller User Guide.

Table of Contents

1. PolarFire FPGA PCIe EndPoint, DDR3L, and DDR4 Memory Controller Data Plane (Ask a Question)

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 DDR3L 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 DDR3L/DDR4/LSRAM.
- The CoreAXI4DMAController performs data transfers between DDR3L/DDR4 memory and LSRAM using the CoreAXI4DMAController.

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, DDR3L, DDR4 and PCIe.

This design can be used with a host PC running either Windows® 10.0 or Linux® 6.8 (Ubuntu).

The Windows® 10 kernel-mode PCIe device driver, developed using the Windows Driver Kit (WDK) platform, interacts with the PolarFire PCIe EndPoint from the host PC. A Graphic User Interface (GUI) application that runs on the host PC is provided to set up and initiate the DMA transactions between the host PC memory, DDR3L, 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 DDR3L/DDR4 and LSRAM through UART IF. If the host PC PCIe slot is not available, the DMA between DDR3L/DDR4 and LSRAM is exercised through UART IF.

The PCIe EndPoint reference design can be programmed using the job file: To program the device using the job file provided along with the design files, see Appendix 4: Programming the Device Using FlashPro Express.

Important: 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.

1.1 Design Requirements (Ask a Question)

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

Table 1-1. Design Requirements

Requirement	Version
Operating System	64-bit Windows® 10 Linux® Ubuntu Kernel version 6.8
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	

Table 1-1. Design Requirements (continued)

Requirement	Version
Libero® SoC	See the `readme.txt` file provided in the design files for the software versions used with this reference design.

Important: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

1.2 Prerequisites (Ask a Question)

Before you begin, perform the following steps:

- For demo design files, download the link from AN4597: PolarFire FPGA PCIe Endpoint DDR3L and DDR4 Memory Controller Data Plane Application Note.
- Download and install Libero SoC on the host PC from the following location: Libero SoC Documentation.

1.3 Demo Design (Ask a Question)

Any external PCIe root-port or bridge can establish a PCIe link with the PolarFire FPGA PCIe EndPoint and access the control registers, DDR3L, 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 initiator 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 (Windows 10 and Linux Kernel version 6.8). 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 the following two independent DMA Engines:

- DMA Engine0: Performs DMA from host PC memory to DDR3L/DDR4/LSRAM
- DMA Engine1: Performs DMA from DDR3L/DDR4/LSRAM to host PC memory

Important: 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.

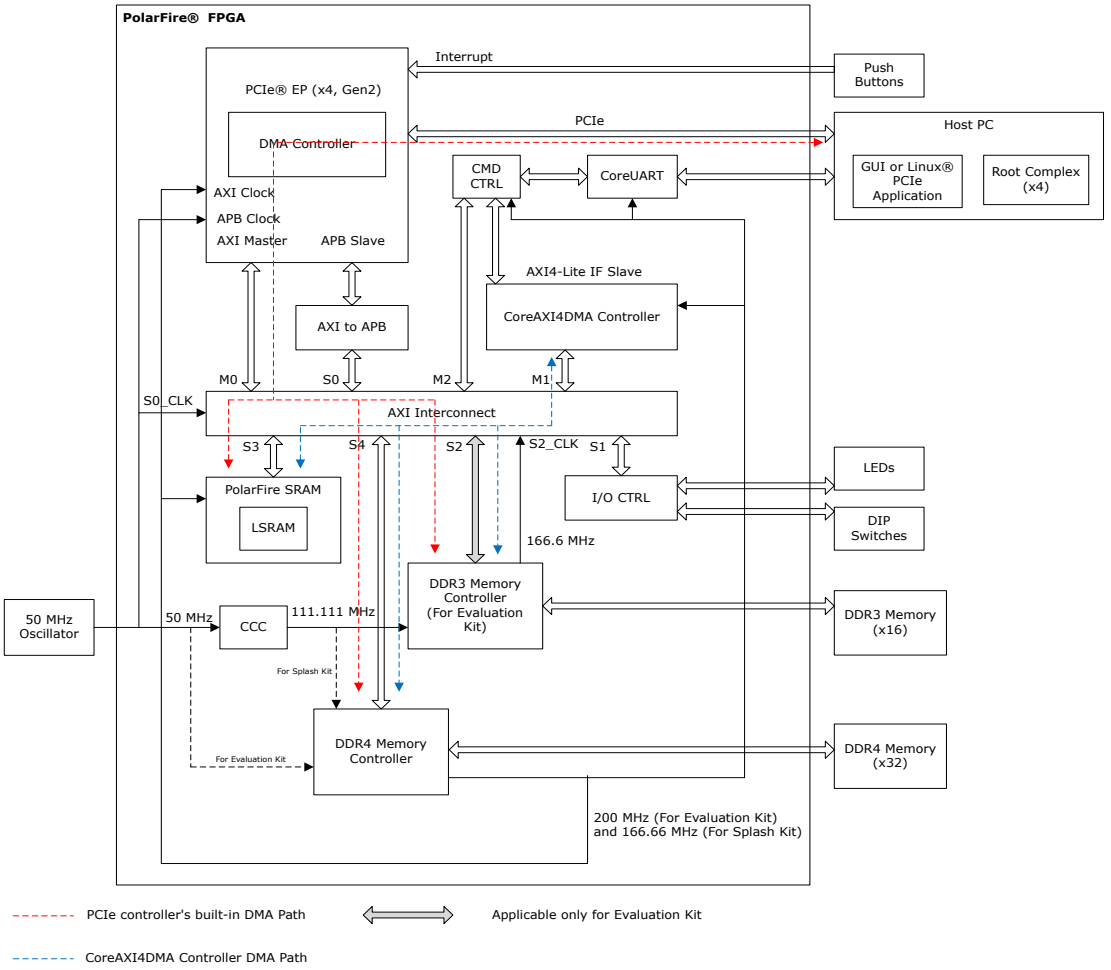
The PCIe demo application initiates CoreAXI4DMA controller IP to perform the DMA between DDR3L memory and LSRAM. The following are the two channels of the CoreAXI4DMA controller IP:

- Channel0: Performs DMA from—DDR3L to DDR4, DDR3L to LSRAM and DDR4 to LSRAM
- Channel1: Performs DMA from—DDR4 to DDR3L, LSRAM to DDR3L 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 or Splash kit is not connected to the host PC PCIe slot.

The following figure shows the top-level block diagram of the PCIe EndPoint demo design.

Figure 1-1. PCIe Demo Design Top-Level Block Diagram



1.3.1 Design Data Flow (Ask a Question)

The demo design performs the following control plane operations:

- LED Blink: Host PC driver performs BAR2 MWr operation 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 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 DDR3L/DDR4/ LSRAM base address. It performs BAR2 memory read/write transactions to DDR3L/DDR4/LSRAM memories.

The demo design supports the following types of DMA operations.

- Continuous DMA operations
- SDGMA Operations
- Core DMA Operations

1.3.1.1 Continuous DMA Operations (Ask a Question)

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.

1.3.1.1.1 DMA0—Host PC Memory to DDR3L/DDR4/LSRAM (Ask a Question)

PCIe DMA Engine0 performs continuous DMA from host PC memory to DDR3L/DDR4/LSRAM memories as described in the following steps:

1. The PolarFire_PCIe_GUI application sets 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 MRd 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 DDR3L/DDR4/LSRAM memories using PCIe AXI initiator interface.
6. The DMA controller repeats this process (from steps 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.

1.3.1.1.2 DMA1—DDR3L/DDR4/LSRAM to Host PC Memory (Ask a Question)

PCIe DMA Engine1 performs continuous DMA from DDR3L/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 DDR3L/DDR4/LSRAM memories.
3. The DMA controller initiates write transaction to PCIe core with the read data. The PCIe core sends a 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.

1.3.1.2 SGDMA Operations (Ask a Question)

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

1.3.1.2.1 Host PC Memory to DDR3L/DDR4 (Ask a Question)

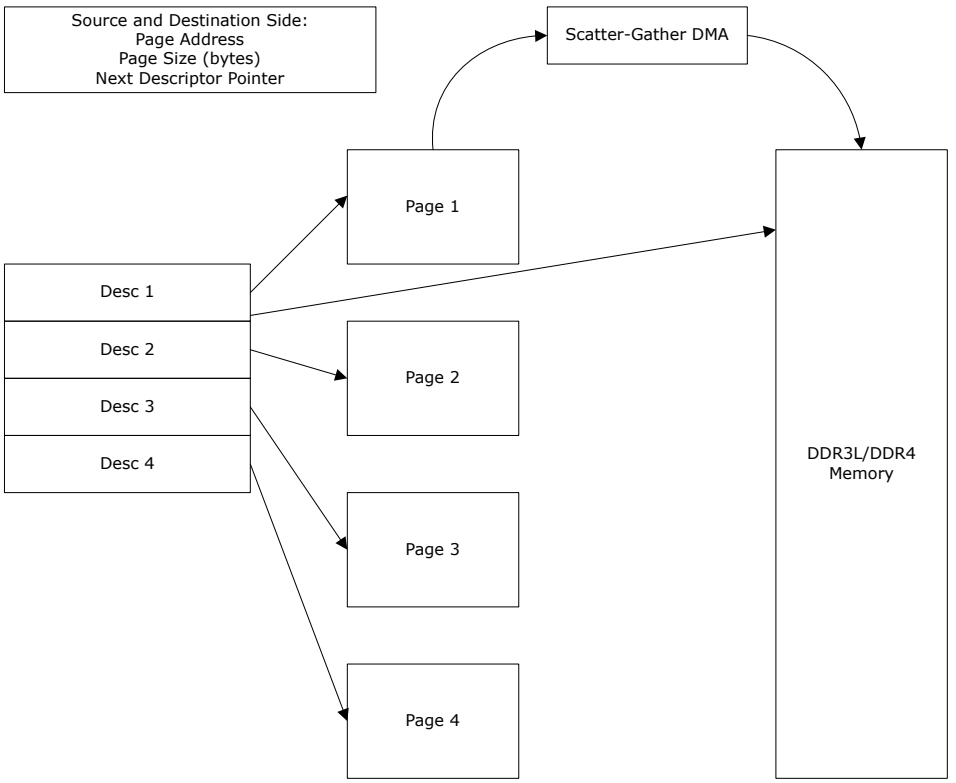
PCIe DMA Engine0 performs DMA from host PC memory to DDR3L/DDR4 memories as shown in the following figure.

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 DDR3L/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 MRd 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 MRd TLP to the host PC. The host PC returns a CplD TLP to the PCIe link.
7. This return data is written to the DDR3L/DDR4 memories using PCIe AXI initiator interface.
8. The DMA controller repeats this process (from steps 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 1-2. DMA0—Example of SG DMA Operation



1.3.1.2.2 DDR3L/DDR4 to Host PC Memory (Ask a Question)

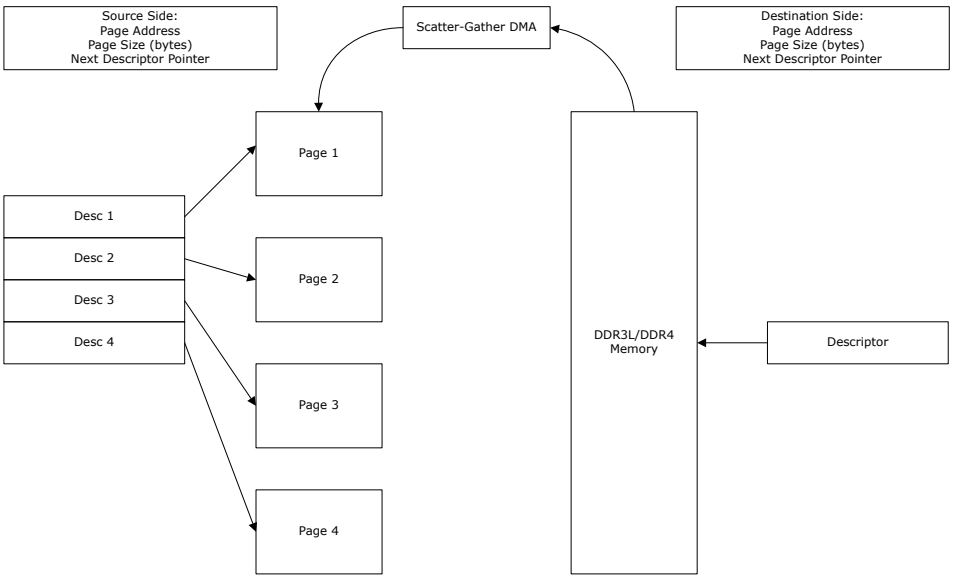
The PCIe DMA Engine1 performs DMA from DDR3L/DDR4 memories to host PC memory (Figure 1-3).

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 DDR3L/DDR4 memory is treated as the continuous memory. Single buffer descriptor is created in LSRAM with the base address of DDR3L/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 MRd TLP to the host PC. The host PC returns a 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 DDR3L/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 MWr TLP to the host PC.
9. The DMA controller repeats this process (from steps 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.

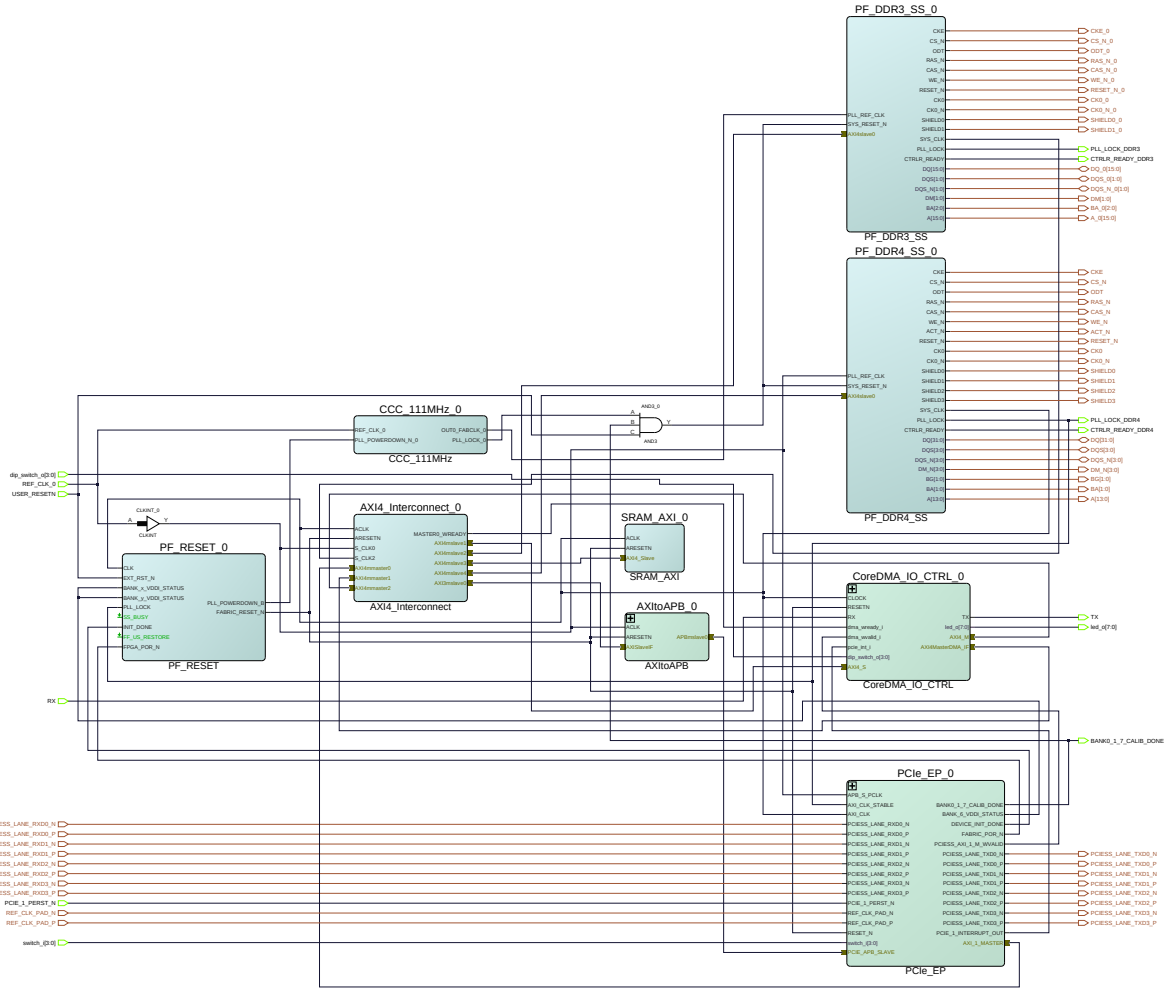
Figure 1-3. DMA1—Example of SG DMA Operation



1.3.2 Design Implementation (Ask a Question)

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

Figure 1-4. 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
- DDR3L subsystem
- DDR4 subsystem
- AXI LSRAM
- AXI4Interconnect IP

1.3.2.1 PCIe EP Subsystem (Ask a Question)

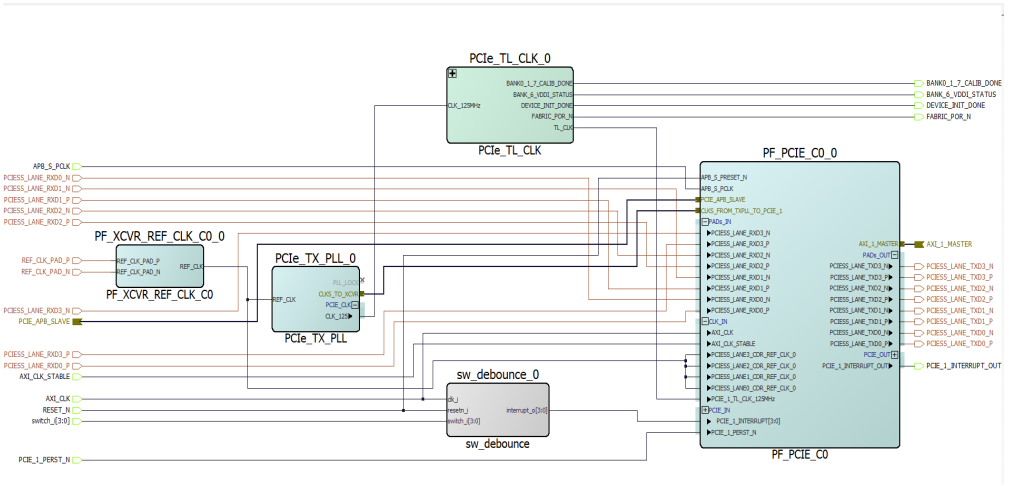
The PCIe_EP SmartDesign implements PCIe EndPoint and its clocking scheme, as shown in the following figure. It also includes the sw_debounce module, which is used to suppress bounces from on-board push buttons and to generate a pulse to the PCIe controller interrupt line.

The PCIE core is configured as an EndPoint with maximum link speed and maximum link width—Gen2 (5.0 Gbps) link speed and × 4 link width. The **Simulation Level** in the configurator is set to **BFM** to simulate the design using PCIe BFM script. The PCIe fabric interface is always the same regardless of the link width or lane rate. The APB interface is enabled to access the PCIe DMA and Address translation registers.

The following two BARs are configured in 64-bit:

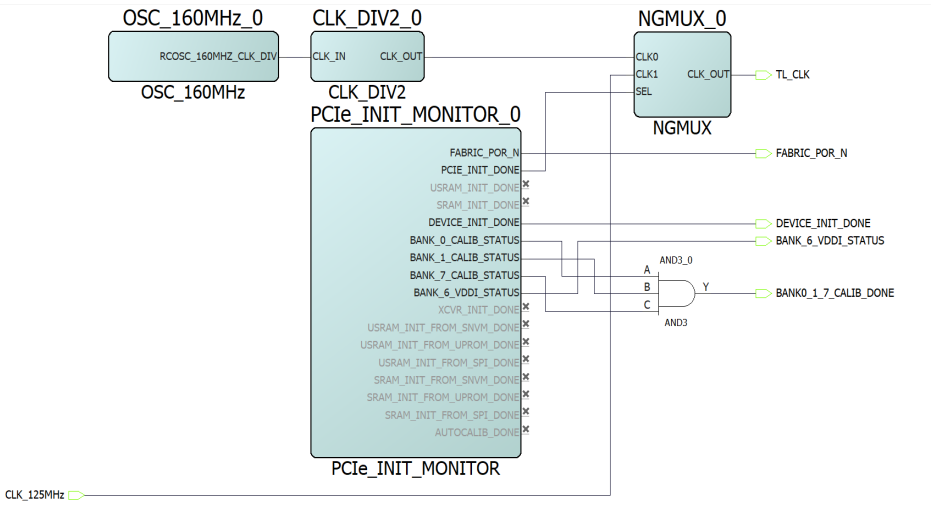
- BAR0: Accesses the PCIe DMA, address translation and interrupt registers through the PCIe controller's APB interface. The address translation register associated with BAR0 is configured to translate the BAR0 address to the PCIe APB IF base address (0x0300_0000).
- BAR2: Accesses the fabric control registers and AXI LSRAM, DDR3L 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, DDR3L and DDR4 memories, the driver on the host PC configures the BAR2 address translation register (TRSL_ADDR) to LSRAM (0x3000_0000)/DDR3L (0x2000_0000)/DDR4 (0x4000_0000) memory base address using the PCIe APB IF through BAR0.

Figure 1-5. PCIe_EP SmartDesign



The PCIe_TL_CLK SmartDesign implements PCIe TL CLK for PolarFire devices, see Figure 1-6. PCIe TL CLK must be connected to CLK_125 MHz 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_125 MHz after PCIe initialization. The BANK 0, BANK 1 and BANK 7 calibration status signals of PF Initialization Monitor IP is used to generate CALIB_DONE signal, which is used for DDR3L/DDR4 reset.

Figure 1-6. PCIe_TL_CLK SmartDesign



1.3.2.2 CoreDMA and UART Subsystem (Ask a Question)

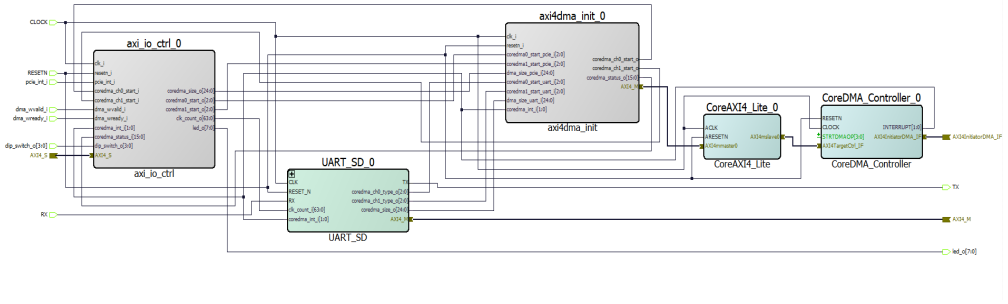
The CoreDMA_IO_CTRL SmartDesign implements fabric registers, CoreDMA4DMA IP initialization and UART_SD. See the following figure.

The axi4dma_init logic initiates the CoreDMA through the AXI4Lite interface to perform the DMA as per commands from GUI. The 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—DDR3L to DDR4, DDR3L to LSRAM and DDR4 to LSRAM DMA and descriptor1 is used for—DDR4 to DDR3L, LSRAM to DDR3L and LSRAM to DDR4 DMA.

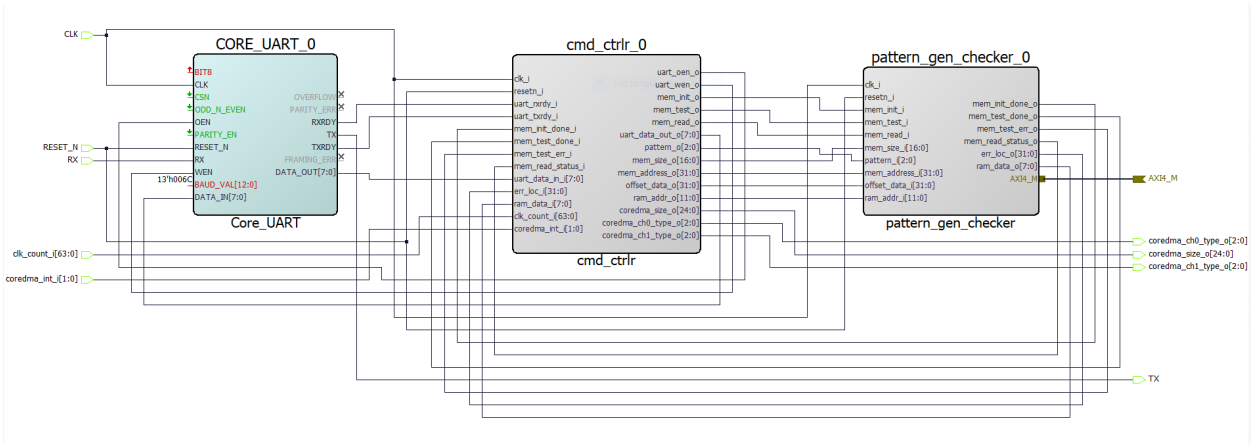
The following figure shows the CoreDMA_IO_CTRL SmartDesign.

Figure 1-7. CoreDMA_IO_CTRL SmartDesign



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

Figure 1-8. UART SmartDesign



1.3.2.3 Memory Controller Subsystem (Ask a Question)

This section describes the various memory controller subsystem like DDR3L, DDR4 and AXI LSRAM.

1.3.2.3.1 DDR3L (Ask a Question)

The DDR3L subsystem is configured to access the 16-bit DDR3L memory through an AXI4 interface. The "PolarFire® Evaluation kit DDR3L memory" preset is applied to configure all of the memory initialization and timing parameters in the DDR3L configurator.

Important:

- DDR3L is applicable only for Evaluation kit demo design.
- For more information about Rev E or later kit DDR3L Configurations (MT41K512M8DA-107: P), see *Appendix 3 - DDR3 Configuration* of the AN4997: PolarFire FPGA Building a Mi-V Processor Subsystem.

1.3.2.3.2 DDR4 (Ask a Question)

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.

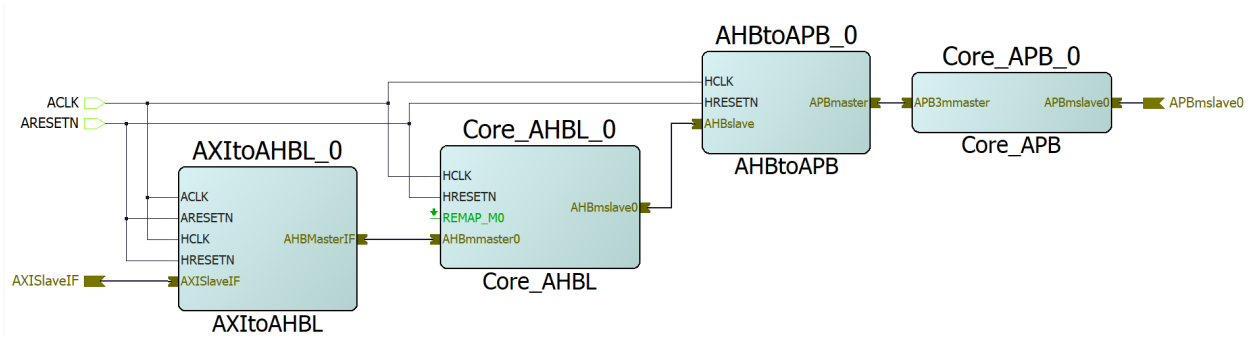
1.3.2.3.3 AXI LSRAM (Ask a Question)

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.

1.3.2.4 AXI to APB SmartDesign (Ask a Question)

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

Figure 1-9. AXI_to_APB SmartDesign



1.3.2.5 CoreAXI4Interconnect IP (Ask a Question)

The CoreAXI4Interconnect IP is configured for the following initiator and target ports:

- Initiator0: PCIe
- Initiator1: CoreAXI4DMAController IP
- Initiator2: Pattern generator and checker logic (pattern_gen_checker block)
- Target0: AXItoAPB bridge (0x0000_0000 to 0x0FFF_FFFF)
- Target1: AXI Target Fabric Registers (0x1000_0000 to 0x1FFF_FFFF)
- Target2: DDR3L Subsystem (0x2000_0000 to 0x2FFF_FFFF) (not enabled for Splash kit)
- Target3: AXI4 LSRAM (0x3000_0000 to 0x3FFF_FFFF)
- Target4: DDR4 Subsystem (0x4000_0000 to 0x4FFF_FFFF)

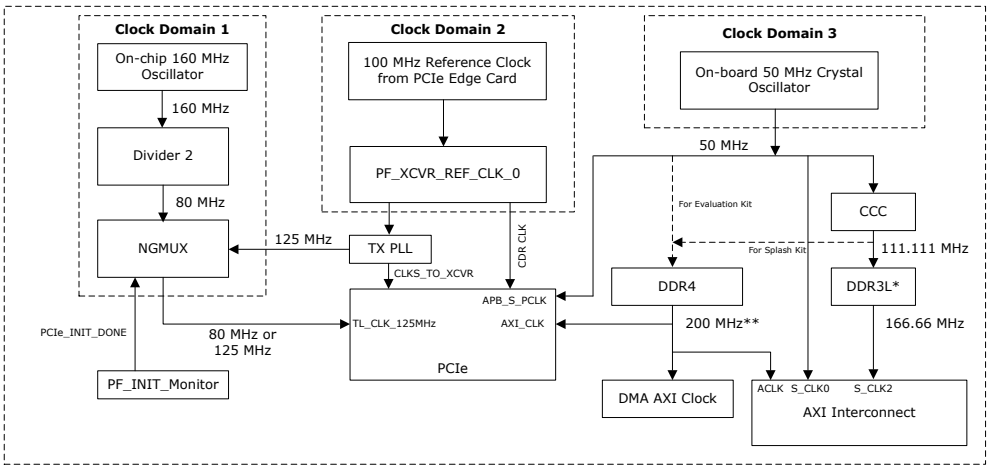
Slave0 is configured to convert AXI4 transactions to AXI3 transactions.

1.4 Clocking Structure (Ask a Question)

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. DDR3L subsystem generates 166.66 MHz clock and is connected to AXI interconnect target2 CDC interface.

Figure 1-10. Clocking Structure



*: DDR3L is applicable only for Evaluation demo design.
On Splash Kit, 166.66 MHz is used

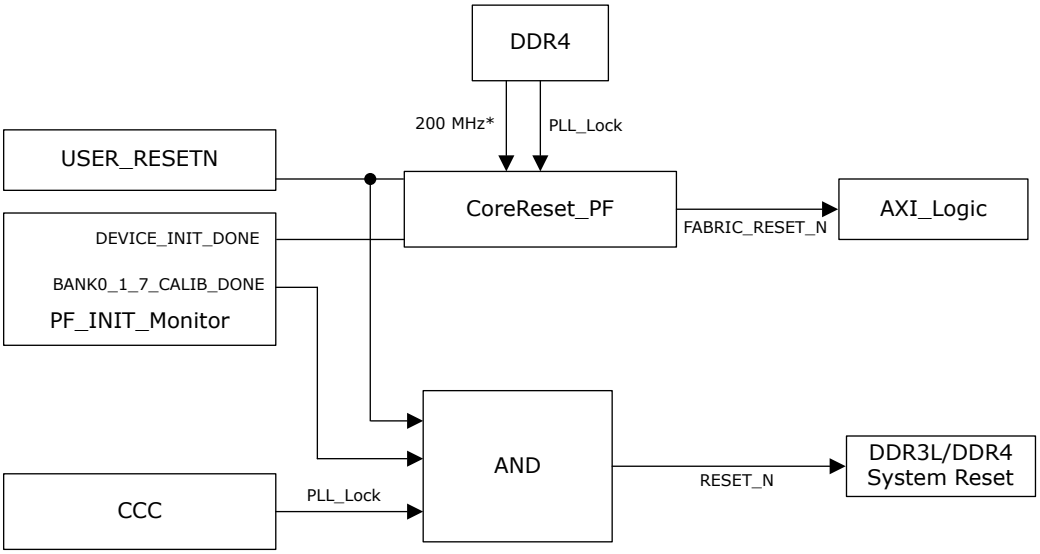
1.5 Reset Structure (Ask a Question)

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 PolarFire Family Power-Up and Resets User Guide.

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

The DDR3L/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 1-11. Reset Structure



*: On Splash Kit, 166.66 MHz is used.

1.6 Throughput Measurement (Ask a Question)

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.

1.7 Simulating the Design (Ask a Question)

Before you begin, perform the following steps:

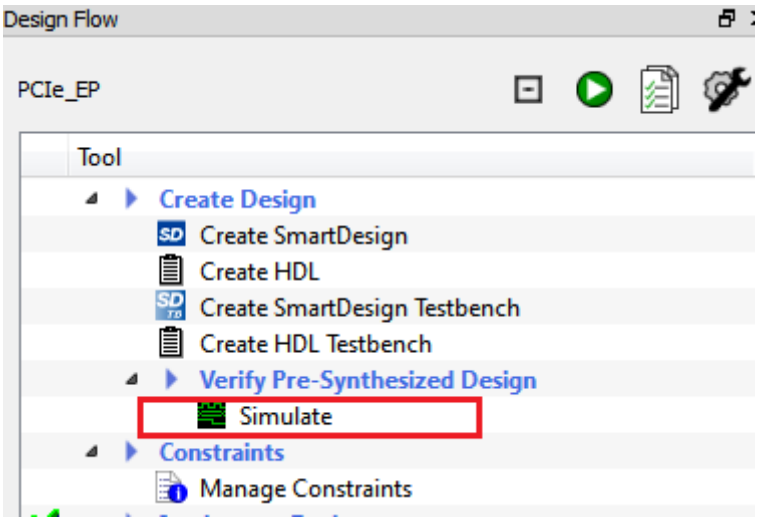
1. Start Libero SoC in the **Project** menu, and click **Open Project**.
2. Browse `mpf_an4597_df/HW/Eval_Kit` or `mpf_an4597_df/HW/Splash_Kit`. For simulation, create the Libero Project using provided TCL scripts. To create the Libero Project, see Appendix 4 : Running the TCL script section.

The PCIe BFM performs 1 KB DMA operations between PCIe and DDR3L, 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 PolarFire Family PCI Express User Guide. The micron DDR3L and DDR4 memory models are instantiated in the testbench for simulating DDR3L and DDR4 memory controllers.

Important: 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. The ModelSim tool takes 10 to 15 minutes to complete the simulation, see the following figure.

Figure 1-12. Simulating the Design



1.7.1 Simulation Flow (Ask a Question)

The following steps describe the PCIe BFM simulation flow.

1. At the start, the `NSYSREST` signal, reset all the components.
2. DDR3L and DDR4 memory controllers initializes the DDR3L/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 initiator interface initiates write and read burst transactions to SRAM_AXI_0, DDR3L and DDR4 through CoreAXI4Interconnect as per the `.bfm` script.
5. After 18 µs, the simulation completes. **PCIE1 BFM Simulation Complete – 282 Instructions – NO ERRORS** message is displayed for **Evaluation kit**.
6. After 40 µs, 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

Important:

- While running the scripts, simulation is running in batch mode by default which takes more time for the design to complete the build and compile flow. If users do not want to run the simulation, they can comment the simulation flow command in the `script.tcl`.
- For evaluation kit simulations, users might get errors in the Simulation window and user can ignore them.

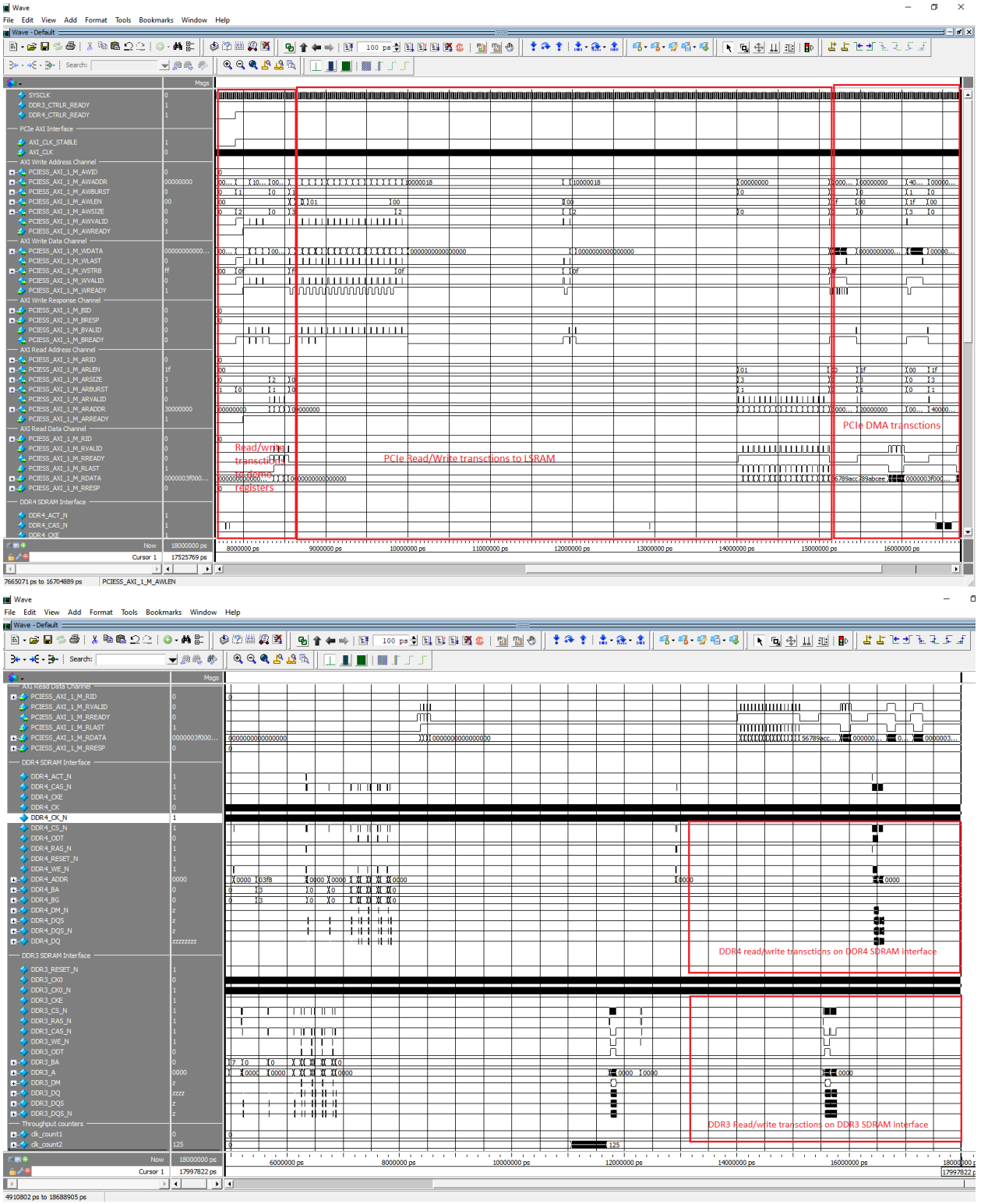
The ModelSim transcript window displays the BFM commands execution messages.

Figure 1-13. Simulation Transcript Window

```
# SFM: Data Read 300000a8 0000002b0000002c at 17686.250000ns
# SFM: Data Read 300000b0 0000002d0000002e at 17691.250000ns
# SFM: Data Read 300000b8 0000002f00000030 at 17696.250000ns
# SFM: Data Read 300000c0 0000003100000032 at 17701.250000ns
# SFM: Data Read 300000c8 0000003300000034 at 17706.250000ns
# SFM: Data Read 300000d0 0000003500000036 at 17711.250000ns
# SFM: Data Read 300000d8 0000003700000038 at 17716.250000ns
# SFM: Data Read 300000e0 000000390000003a at 17721.250000ns
# ----------------------------------------------------------------------
# ----DMA TRANSFER DONE(FROM FABRIC ADDRESS SPACE TO PCIe ADDRESS SPACE)----
# ----------------------------------------------------------------------
# SFM: Data Read 300000e8 0000003b0000003c at 17726.250000ns
# BFM:204:wait 1  starting at 17731 ns
# SFM: Data Read 300000f0 0000003d0000003e at 17731.250000ns
# BFM: ******************************End of PCIe BFM Simulation**********************************
# BFM:207:return
# SFM: Data Read 300000f8 0000003f00000040 at 17736.250000ns
##########################################################################
#
# PCIE1 BFM Simulation Complete - 282 Instructions - NO ERRORS
#
##########################################################################
#
```

The following figure shows the actual **Waveform** window displaying the sequence of data being written and read using BFM.

Figure 1-14. Simulation Waveform Window



2. Run PROGRAM Action for Evaluation Kit (Ask a Question)

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

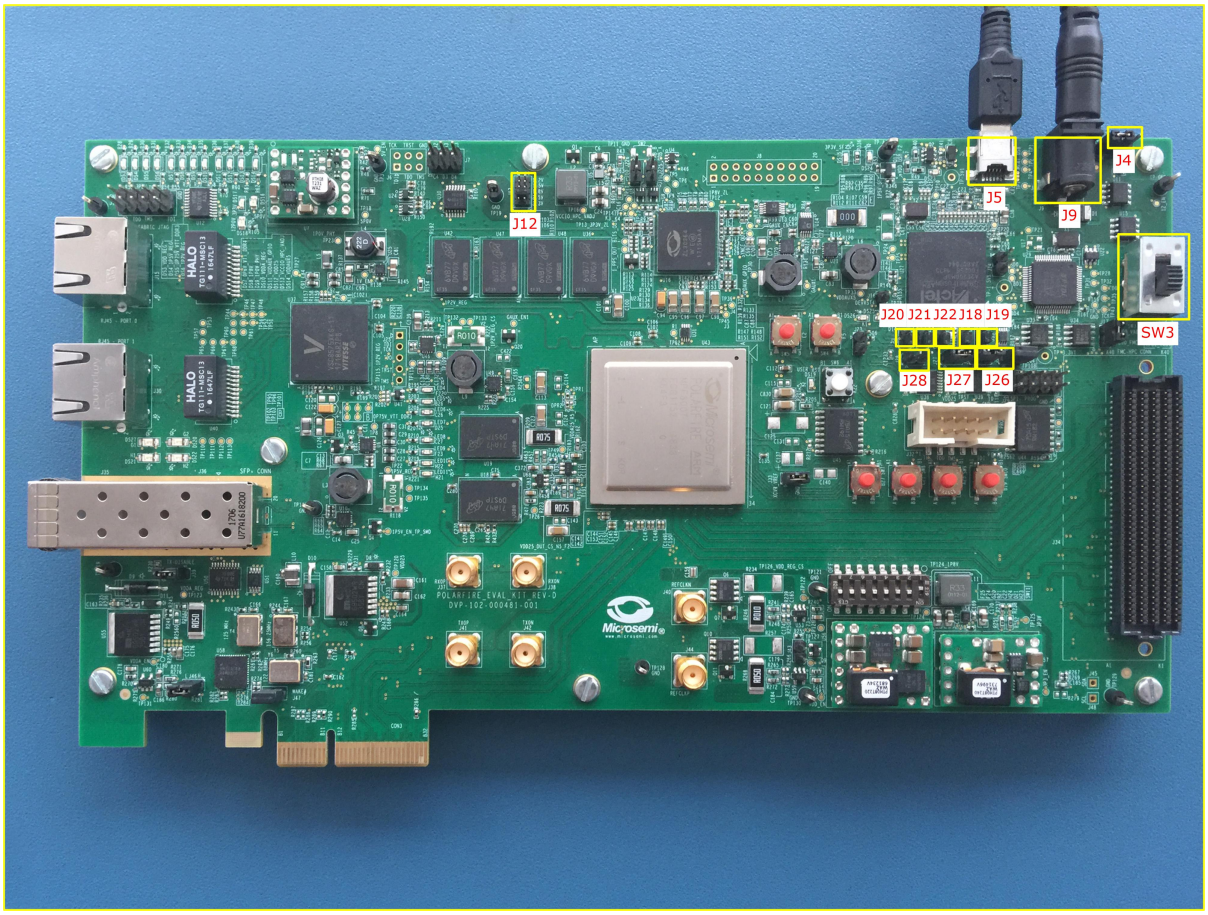
1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Table 2-1. 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 2.5V

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 2-1. Evaluation Kit Board Setup



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

3. Run PROGRAM Action for Splash Kit (Ask a Question)

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

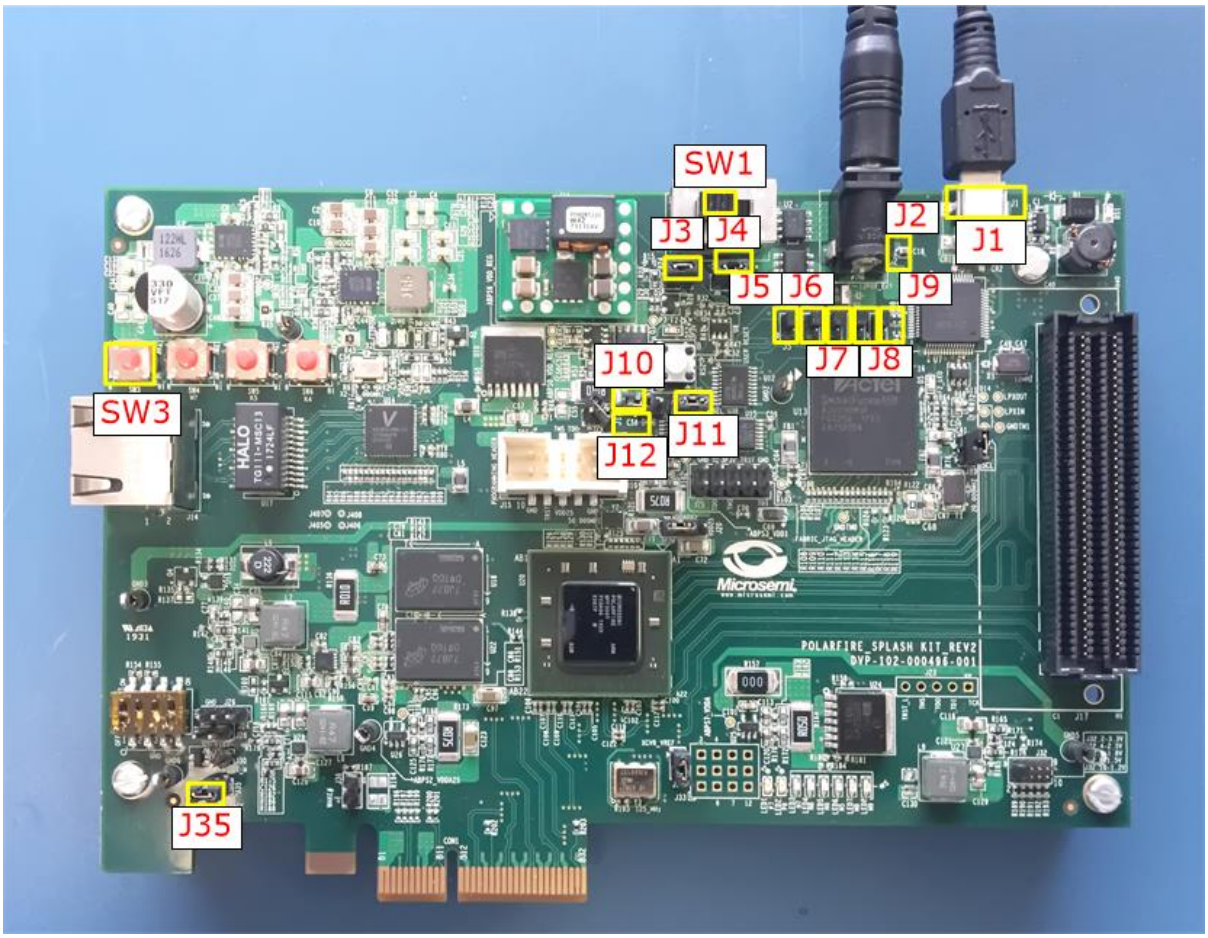
1. Ensure that the jumper settings on the board are the same as those listed in the following table.

Table 3-1. 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 1.0V

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 3-1. 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 to run the PCIe EndPoint demo.

Figure 3-2. Programming the Device

4. Running the Demo (Ask a Question)

This section describes how to install and use the PCIe Demo application. The PolarFire PCIe demo application is a simple 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.

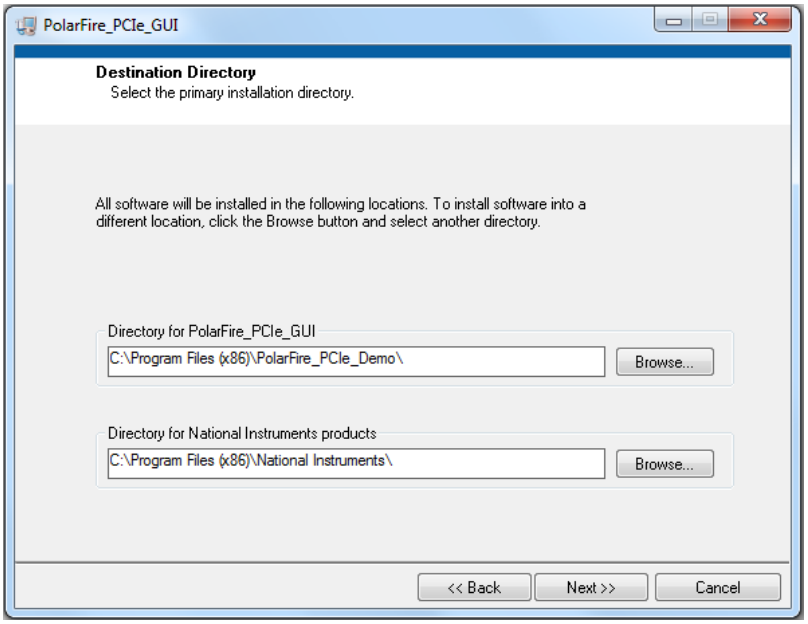
This section also describes how to connect the kit to the Host PC PCIe slot. If the host PC PCIe slot is not available, the DMA between DDR3L/DDR4 and LSRAM can be exercised through UART IF.

4.1 Installing PCIe Demo Application (Ask a Question)

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

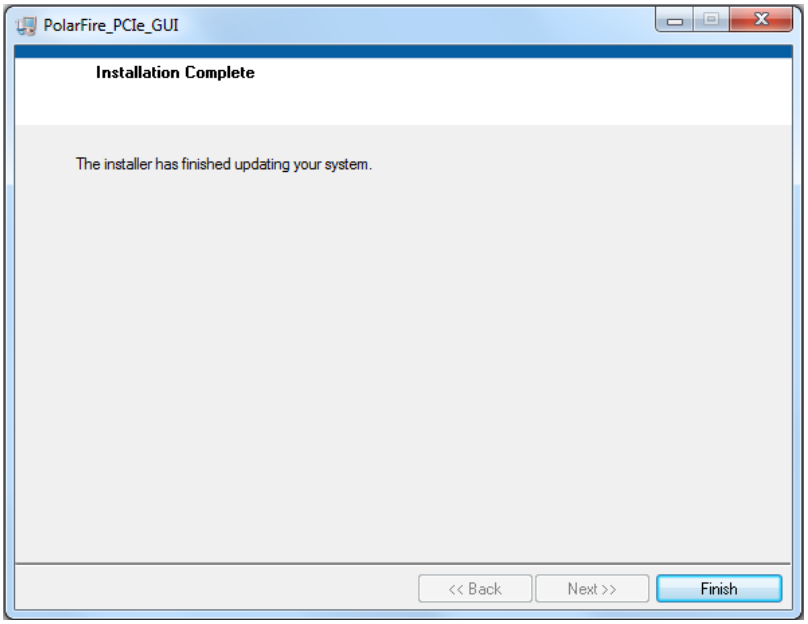
1. Install the GUI_Installer (`setup.exe`) from the following design files folder: `mpf_an4597_df\GUI_Installer`
2. Double click the `setup.exe` in the provided GUI installation: `(GUI_Installer\setup.exe).`
3. Apply default options, as shown in the following figure.

Figure 4-1. Installing PCIe Demo Application



4. To start the installation, click **Next**.
5. To complete the installation, click **Finish**.

Figure 4-2. Successful Installation of PCIe Demo Application



4.2 Running the Demo Through PCIe (Ask a Question)

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 (Ask a Question)

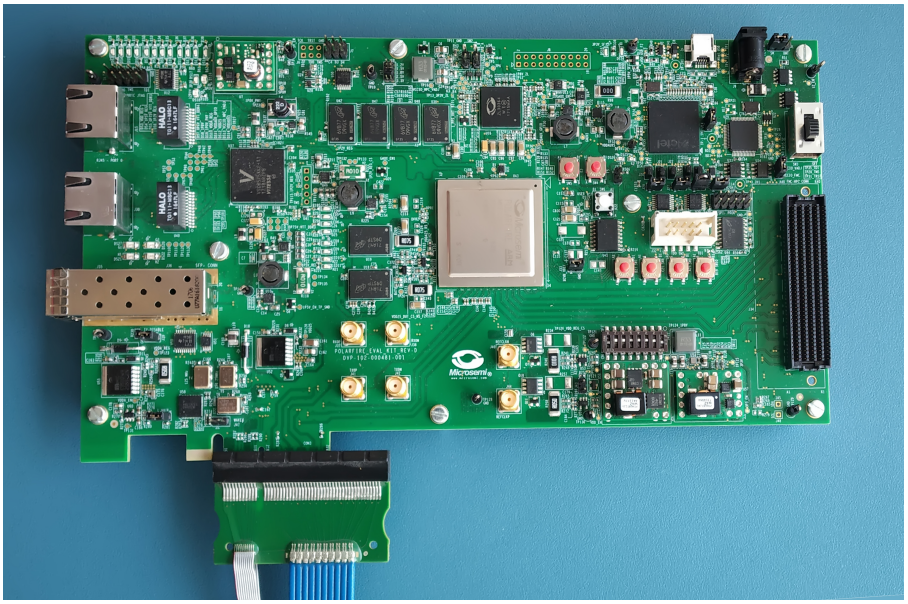
1. After successful programming, power OFF the PolarFire Evaluation/Splash board and shut down the host PC.
2. Connect the CON3-PCIe Edge connector of the PolarFire Evaluation/Splash board to the host PC's PCIe slot through the PCI Edge card ribbon cable.
   This demo is designed to work with any PCIe Gen2 compliant slot. If the host PC does not support Gen2 compliant slot, the demo switches to Gen1 mode.

Important:

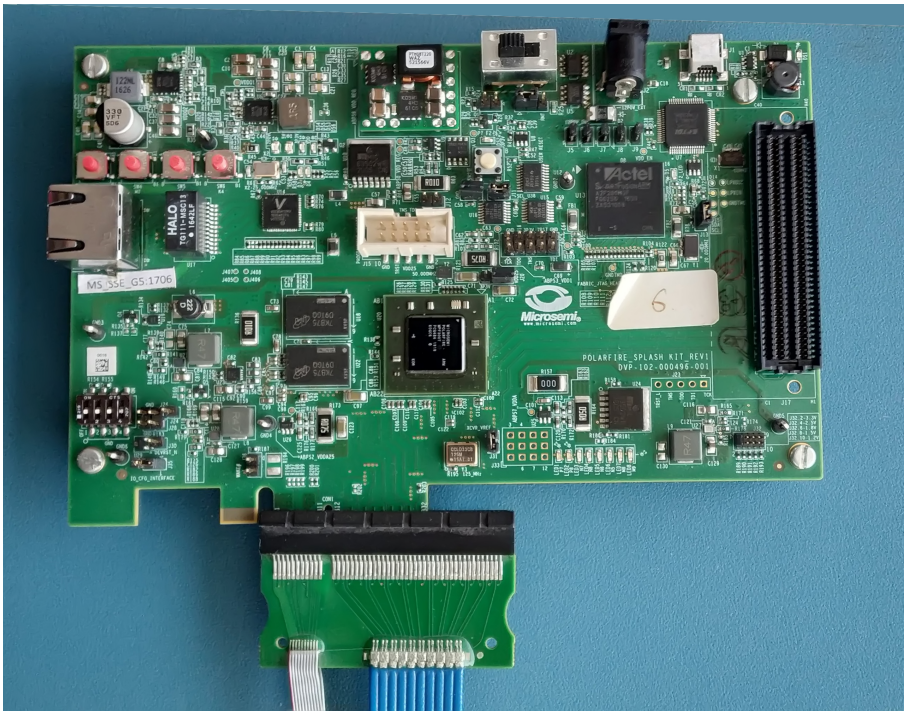
- Power OFF the host PC while inserting the PCIe Edge connector. If it is not powered OFF, the PCIe device detection and the selection of Gen1 or Gen2 mode might fail. The device detection and selection depend on the host PC PCIe configuration.
- After connecting the board to the host PC, the host PC 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 4-3. PolarFire Evaluation Kit Setup for Host PC

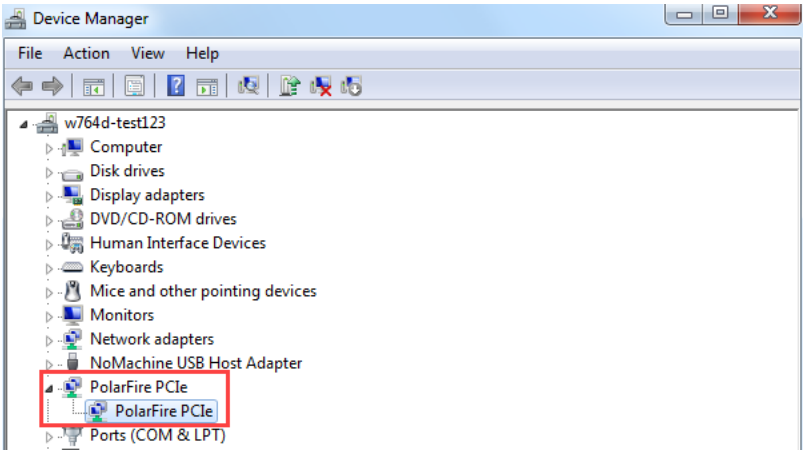


3. Power-on the power supply switch SW3.
   The following figure shows the board setup for the host PC in which the PolarFire Splash Kit is connected to the host PC PCIe slot.



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 4-4. Device Manager



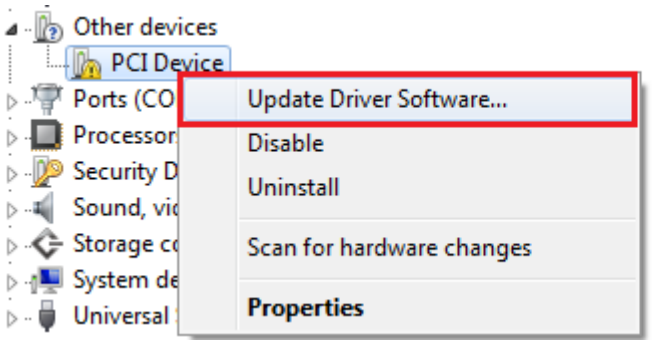
Important: If the device is still not detected, check if the BIOS version in the host PC is the latest and if PCI is enabled in the host PC BIOS.

4.2.2 Driver Installation (Ask a Question)

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

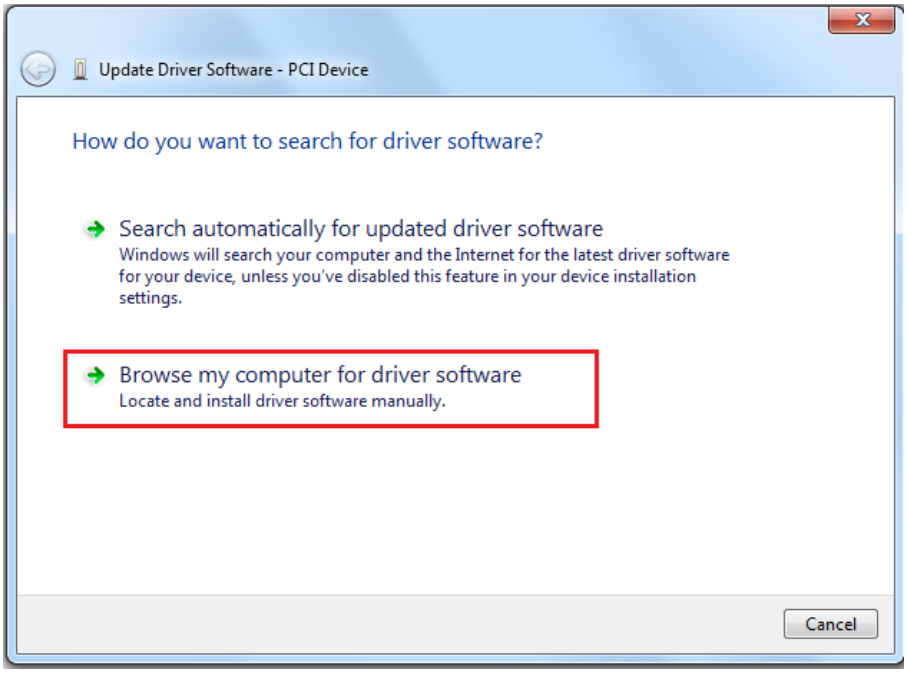
1. In the **Device Manager**, right-click **PCI Device** and select **Update Driver Software...**, as shown in the following figure. To install the drivers, administrative rights are required.

Figure 4-5. Update Driver Software



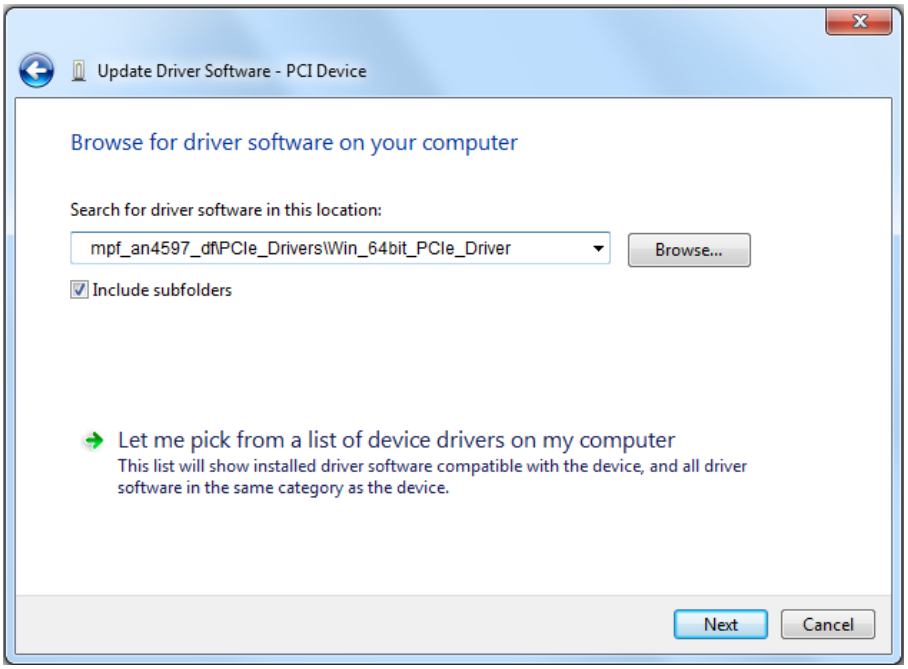
2. In the **Update Driver Software - PCIe Device** window, select **Browse my computer for driver software**, as shown in the following figure.

Figure 4-6. Browse for Driver Software

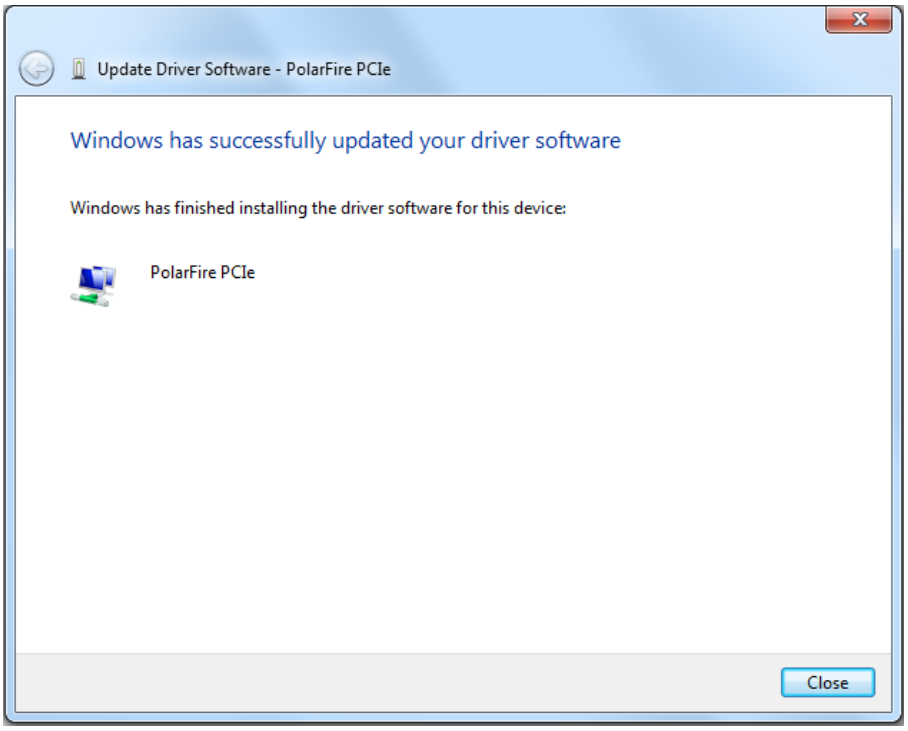


3. Unzip the file and browse the drivers folder: `mpf_an4597_df\PCIe_Drivers\Win_64bit_PCIe_Driver` and click **Next**, as shown in the following figure.

Figure 4-7. Browse for Driver Software Continued



4. **Figure 4-8.** Successful Driver Installation

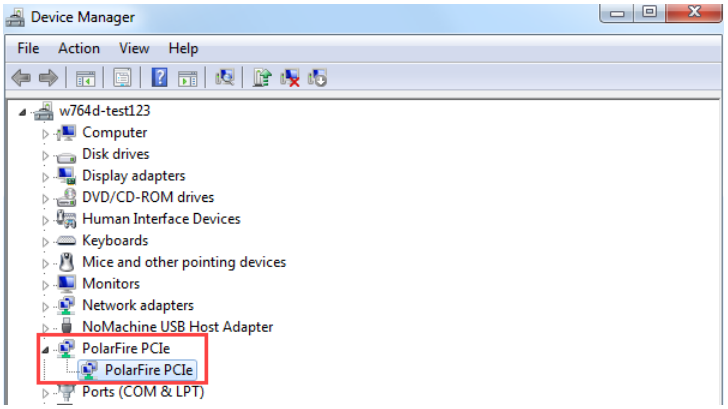


4.2.3 Running the PCIe Demo Application (Ask a Question)

To run the demo design, perform the following steps:

1. To expand the **PolarFire PCIe** device, click **PolarFire PCIe** option in the host PC **Device Manager**.

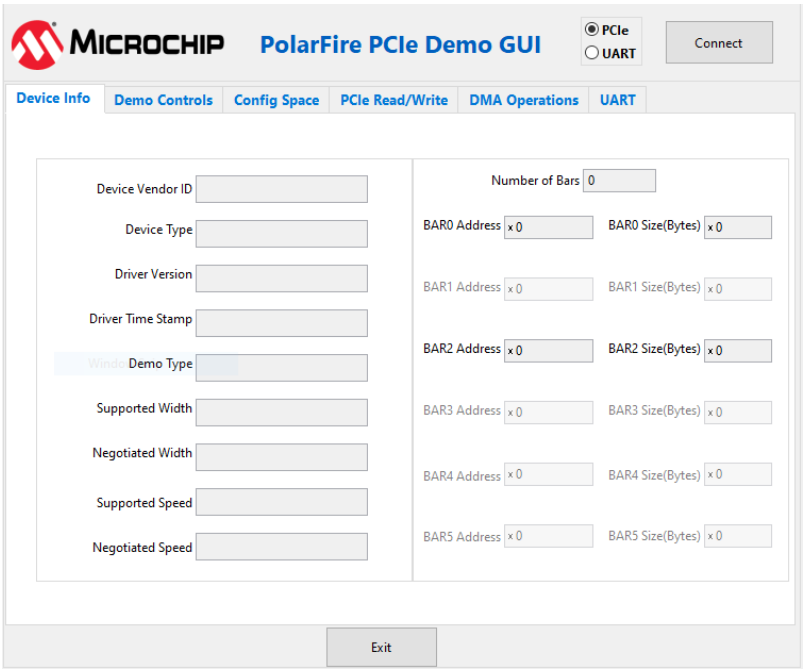
Figure 4-9. Device Manager—PCIe Device Detection



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

2. Navigate to **All Programs** > **PolarFire_PCIe_GUI** > **PolarFire_PCIe_GUI**.

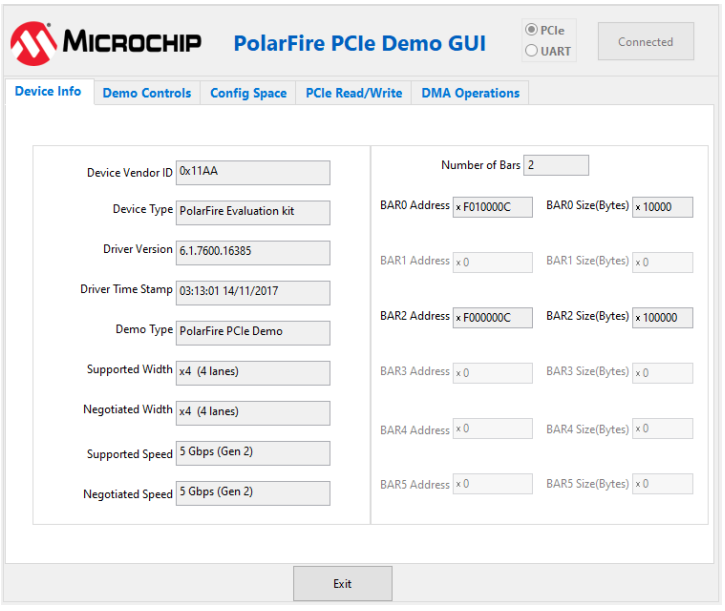
Figure 4-10. PCIe EndPoint Demo Application



The **PolarFire PCIe Demo** window is displayed.

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 Width, Negotiated Width, Supported Speed, Negotiated Speed, Number of Bars and BAR Address.

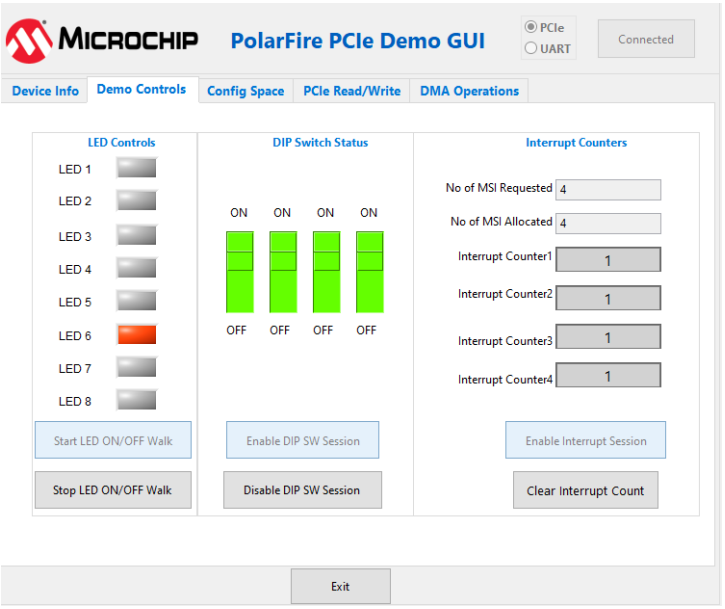
Figure 4-11. Device Info Tab



4. To display the **LED Controls**, **DIP Switch Status** and **Interrupt Counters**, click **Demo Controls** tab.
5. To view the controlling LEDs (observe LED4 to LED11 on the PolarFire Evaluation Kit), click **Start LED ON/OFF Walk**, **Enable DIP SW Session** and **Enable Interrupt Session**. This allows you

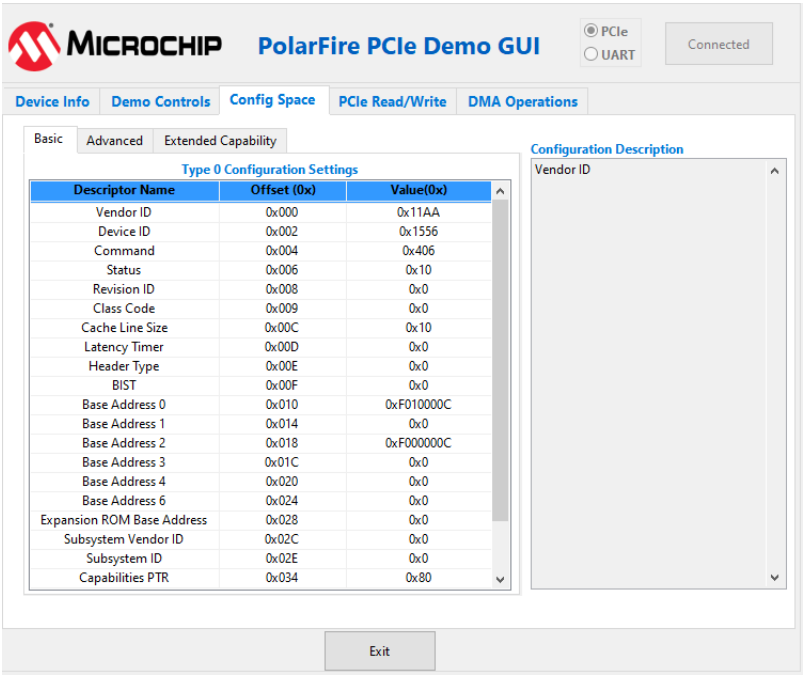
to check the status of the DIP switch (ON/OFF the DIP1 to DIP4 on the PolarFire Evaluation Kit/Splash Kit) and monitor the interrupts (press SW7 to SW10 on the PolarFire Evaluation Kit and SW3 to SW6 on the PolarFire Splash Kit to generate interrupts) simultaneously, as shown in the following figure.

Figure 4-12. Demo Controls



6. To view the details about the PCIe configuration space, click **Config Space** tab, as shown in the following figure.

Figure 4-13. Config Space Tab

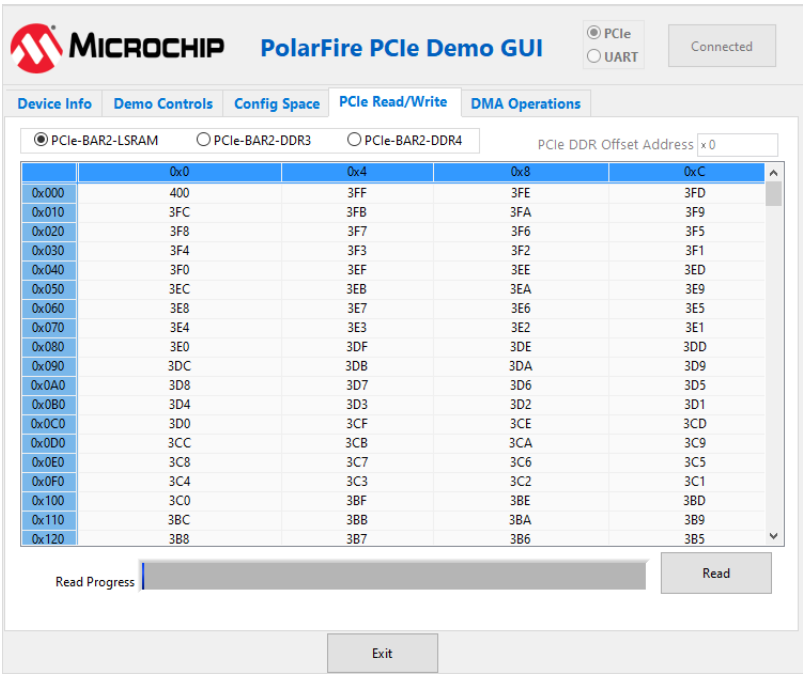


7. To perform read and write operations to DDR/LSRAM using BAR2 space, click **PCIe Read/Write** tab.

8. Select **LSRAM/DDR3L/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.

Important: PCIe BAR2-DDR3L is applicable only for Evaluation kit.

Figure 4-14. PCIe BAR2 Memory Access—LSRAM



9. For different DMA operations such as DDR and LSRAM, click **DMA Operations** tab.

Important: DDR3L DMA options are not applicable for Splash Kit demo.

4.2.3.1 Continuous DMA—Operations (Ask a Question)

The following instructions describe running DMA operations between PC and DDR3L, PC and DDR4, PC and LSRAM:

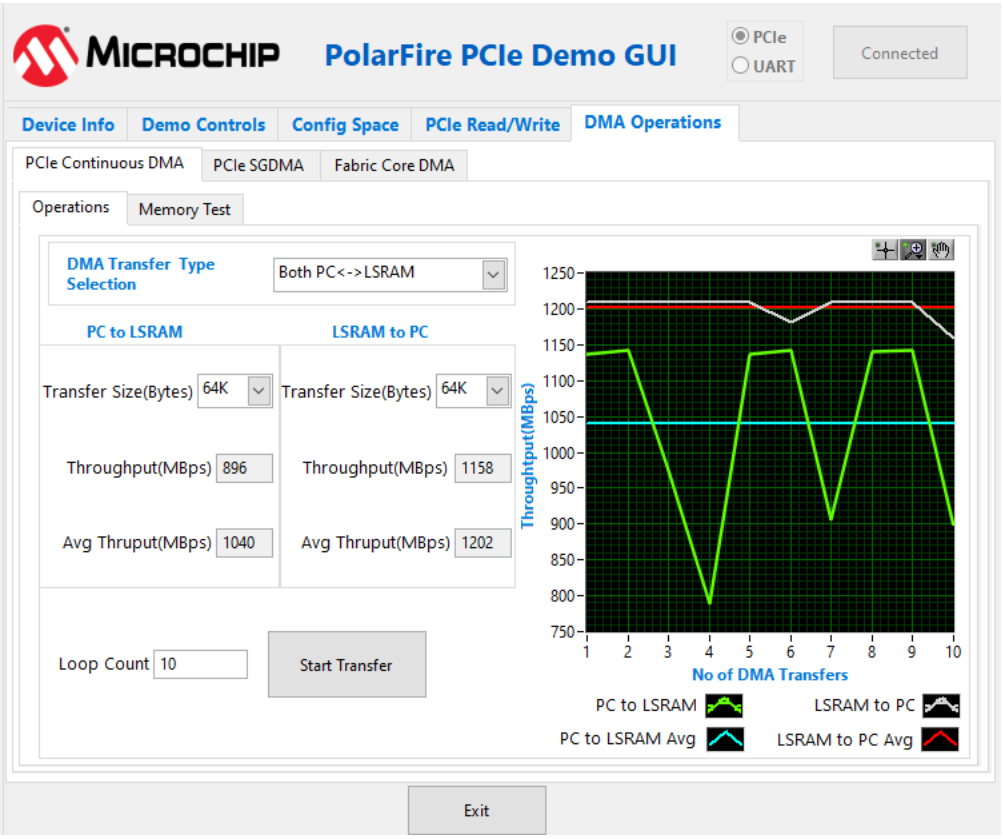
1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
   - **PC->DDR3L—**to transfer the data from host PC to PolarFire DDR3L memory
   - **DDR3L->PC—**to transfer the data from PolarFire DDR3L memory to host PC
   - **Both- PC<->DDR3L—**to transfer the data from host PC to and from PolarFire DDR3L 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

2. 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.
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.

Important: The AXI LSRAM in the design is configured for 4 KB. This 4 KB is overwritten 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 4-15. Continuous DMA Operations with DMA Transfer Type Selection as Both PC and LSRAM



4.2.3.2 Continuous DMA—Memory Test (Ask a Question)

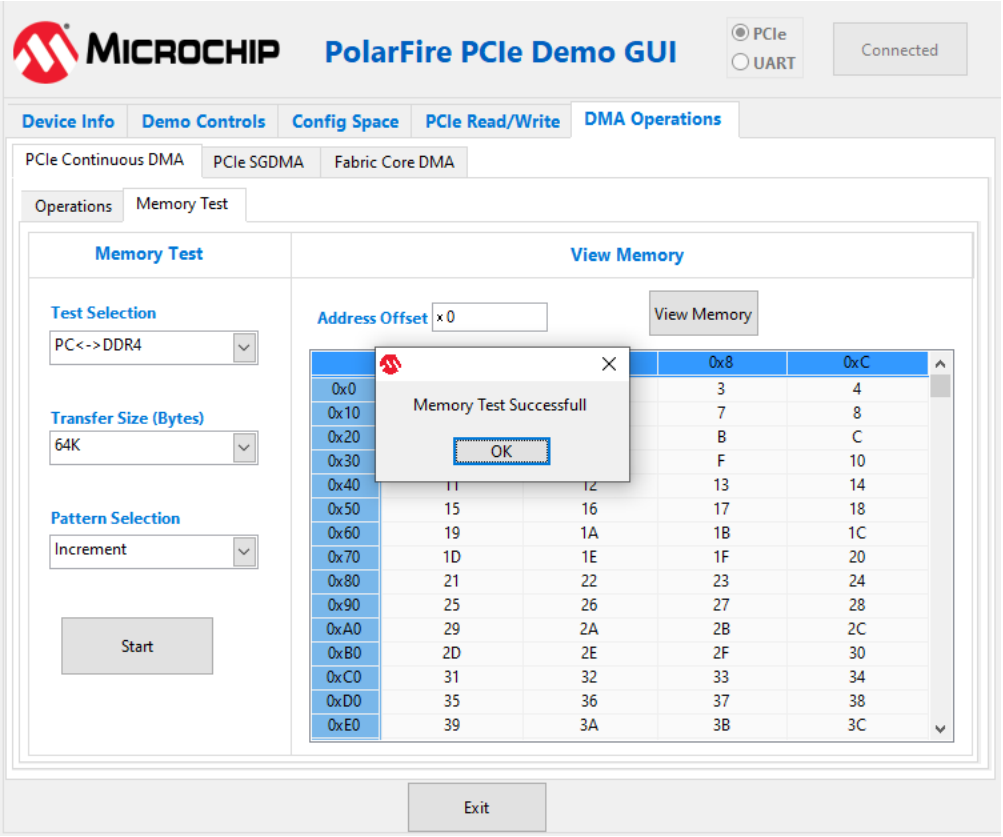
The following instructions describe running **Memory Test** between PC and DDR3L/DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
   - **PC<->DDR3L—**to transfer the data from host PC to and from PolarFire DDR3L 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

The following figure shows the **Memory Test Successful** window.

Figure 4-16. Continuous DMA Memory Test—Memory Test Successful



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

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

4.2.3.3 SGDMA—Operations (Ask a Question)

The following instructions describe running SGDMA operations between PC and DDR3L, PC and DDR4:

1. Select one of the following options from the **DMA Transfer Type** Selection drop-down list:
   - **PC -> DDR3L—**to transfer the data from host PC to PolarFire DDR3L memory
   - **DDR3L-> PC—**to transfer the data from PolarFire DDR3L memory to host PC

- **Both PC <->DDR3L—**to transfer the data from host PC to and from PolarFire DDR3L 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 1 MB) 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 (Ask a Question)

The following instructions describe running **Memory Test** between PC and DDR3L/DDR4/LSRAM:

1. Select one of the following options from the **Test Selection** drop-down list:
   - **PC<->DDR3L—**to transfer the data from host PC to and from PolarFire DDR3L 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 (Ask a Question)

The following instructions describe running **DMA operations** between LSRAM and DDR3L, LSRAM and DDR4, DDR3L and DDR4:

1. Select one of the following options from the **DMA Transfer Type Selection** drop-down list:
   - **LSRAM -> DDR3L—**to transfer the data from LSRAM to PolarFire DDR3L memory
   - **DDR3L-> LSRAM—**to transfer the data from PolarFire DDR3L memory to LSRAM
   - **Both LSRAM <->DDR3L—**to transfer the data from LSRAM to and from PolarFire DDR3L 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 -> DDR3L—**to transfer the data from DDR4 to DDR3L memory
   - **DDR3L -> DDR4—**to transfer the data from DDR3L to DDR4 memory
   - **Both DDR4 <-> DDR3L—**to transfer the data from DDR4 to and from DDR3L 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.

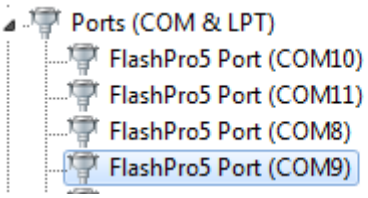
> **Important:** The AXI LSRAM in the design is configured for 4 KB. This 4 KB is overwritten if more than 4 KB of DMA operation is performed on LSRAM. This option is provided to exercise the throughputs with larger DMA size.

5. Click **Exit**.

4.3 Running the Demo Through UART (Ask a Question)

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 4-17. 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.
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 4-18.

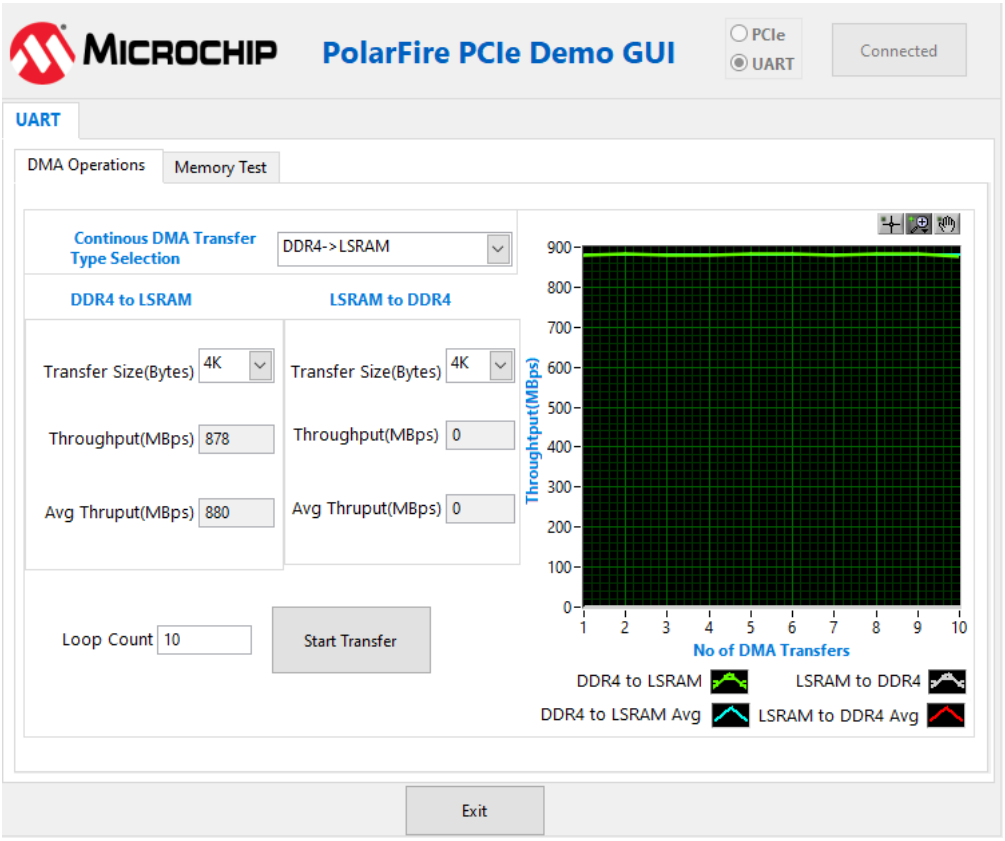
4.3.1 UART—DMA Operations (Ask a Question)

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:
   - **DDR3L -> LSRAM**: To transfer the data from DDR3L to PolarFire LSRAM memory.
   - **LSRAM -> DDR3L**: To transfer the data from PolarFire LSRAM memory to DDR3L.
   - **Both DDR3L <->LSRAM**: To transfer the data from DDR3L 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 -> DDR3L:** To transfer the data from DDR4 to DDR3L memory
   - **DDR3L -> DDR4:** To transfer the data from DDR3L to DDR4 memory
   - **Both DDR4 <-> DDR3L:** To transfer the data from DDR4 to and from DDR3L memory
   - **Both DDR3L<->DDR4:** To transfer the data from DDR3L 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 4-18. UART—DMA Operations



Important: 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.

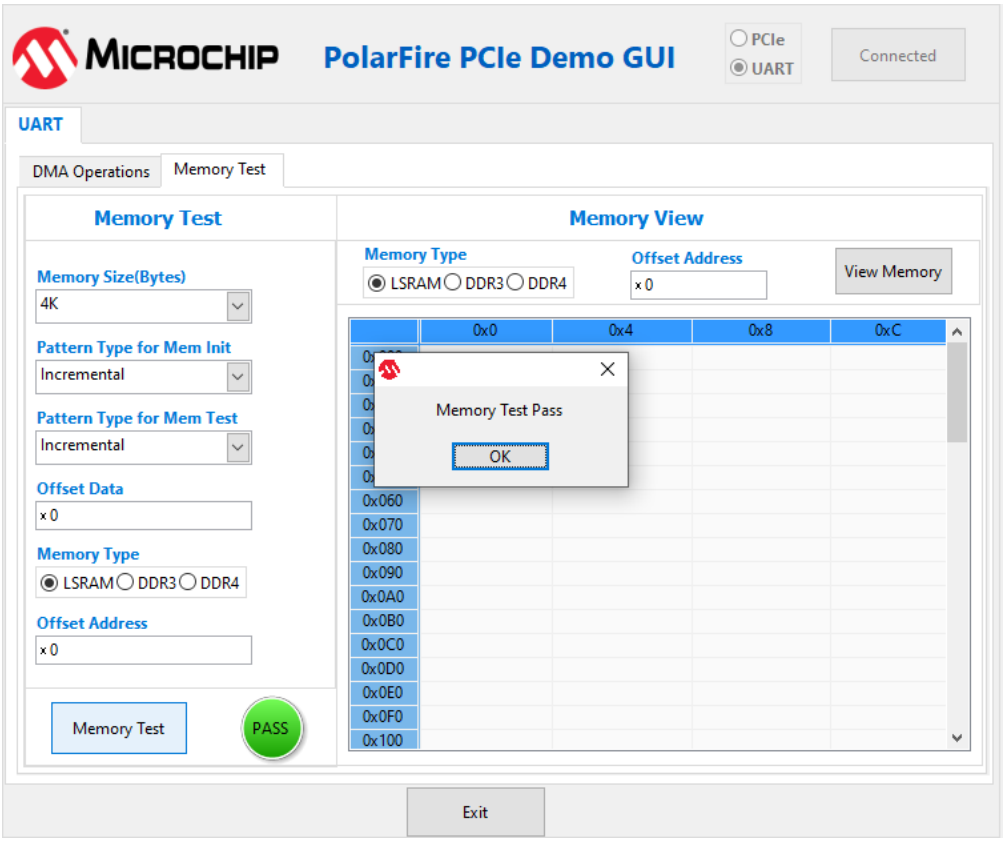
4.3.1.1 UART—Memory Test (Ask a Question)

The following instructions describe running **Memory Test** between PC and DDR3L/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 **Pattern Type for Mem Init** and **Pattern Type for Mem Test** should be same.
3. Click **Memory Test**.
   - GUI sends command to fabric logic to initiate the LSRAM/DDR3L/DDR4 memory
   - GUI sends command to fabric logic to read and compare LSRAM/DDR3L/DDR4 memory

   The following figure shows the **UART—Memory Test** tab.

Figure 4-19. UART—Memory Test



Important: Change the **Offset Address** and click **View Memory** to read the RAM memory content. Ensure that the **Pattern Type for Mem Init** and **Pattern Type for Mem Test** should be same as shown in the preceding image.

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 (Ask a Question)

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 Ubuntu Linux Host PC.
3. Ubuntu Linux Kernel detects the PolarFire Evaluation Kit board or PolarFire Splash Kit PCIe end point as Actel DevicePCIe end point as Actel Device.
4. On Linux Command Prompt, use the lspci command to display the PCIe information.

```
# lspci
```

Figure 4-20. PCIe Device Detection

```
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_app$ lspci
00:00.0 Host bridge: Intel Corporation Device a703 (rev 01)
00:01.0 PCI bridge: Intel Corporation Device a70d (rev 01)
00:02.0 VGA compatible controller: Intel Corporation Device a780 (rev 04)
00:04.0 Signal processing controller: Intel Corporation Device a71d (rev 01)
00:06.0 PCI bridge: Intel Corporation Device a74d (rev 01)
00:14.0 USB controller: Intel Corporation Device 7ae0 (rev 11)
00:14.2 RAM memory: Intel Corporation Device 7aa7 (rev 11)
00:16.0 Communication controller: Intel Corporation Device 7ae8 (rev 11)
00:16.3 Serial controller: Intel Corporation Device 7aeb (rev 11)
00:17.0 SATA controller: Intel Corporation Device 7ae2 (rev 11)
00:1f.0 ISA bridge: Intel Corporation Device 7a83 (rev 11)
00:1f.3 Audio device: Intel Corporation Device 7ad0 (rev 11)
00:1f.4 SMBus: Intel Corporation Device 7aa3 (rev 11)
00:1f.5 Serial bus controller: Intel Corporation Device 7aa4 (rev 11)
00:1f.6 Ethernet controller: Intel Corporation Ethernet Connection (17) I219-LM (rev 11)
01:00.0 Non-VGA unclassified device: Actel Device 1556
```

4.3.2.1 Drivers Installation (Ask a Question)

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

1. Change to `PCIe_Drivers` directory by using the following command:
   ```
   #cd
   <source_code_directory>\mpf_an4597_df\PCIe_Drivers\Linux_PCIe_Drivers_application\linux_pcie_dma\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.
   ```
   sudo insmod mpci.ko
   ```

Important: Root privileges are required to execute this command.

The following figure shows the PCIe device driver installation.

Figure 4-21. PCIe Device Driver Installation

```
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_driver_updated_6.8$ make
#make -W -Wstrict-prototypes -Wmissing-prototypes -C  /lib/modules/6.8.0-49-generic/build SUBDIRS=/home/chandra/work/cardtest/linux_pcie_driver_updated_6.8 modules
#@echo /lib/modules/6.8.0-49-generic/build
Building now


#make -W -Wstrict-prototypes -Wmissing-prototypes -C  /lib/modules/6.8.0-49-generic/build SUBDIRS=/home/chandra/work/cardtest/linux_pcie_driver_updated_6.8 modules
make  -W -Wstrict-prototypes -Wmissing-prototypes -C  /lib/modules/6.8.0-49-generic/build M=/home/chandra/work/cardtest/linux_pcie_driver_updated_6.8 modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-49-generic'
warning: the compiler differs from the one used to build the kernel
  The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
  You are using:           gcc-12 (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
  CC [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpcie.o
  CC [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mdma.o
  CC [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/misrdpc.o
  LD [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpci.o
  MODPOST /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/Module.symvers
  CC [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpci.mod.o
  LD [M]  /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpci.ko
  BTF [M] /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpci.ko
Skipping BTF generation for /home/chandra/work/cardtest/linux_pcie_driver_updated_6.8/mpci.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-49-generic'
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_driver_updated_6.8$ sudo insmod mpci.ko
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_driver_updated_6.8$ ls /dev/MS_PCI_DEV
/dev/MS_PCI_DEV
```

4.3.2.1.1 Linux PCIe Application Compilation (Ask a Question)

1. Compile the Linux user space application as follows:
   ```
   #cd
   <source_code_directory>\mpf_an4597_df\PCIe_Drivers\Linux_PCIe_Drivers_application\linux_pcie_app
   # make clean
   # 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:
   `#sudo ./pcie_app`
   The following figure shows the displayed **Help** menu.

Figure 4-22. Linux PCIe Application Utility

```
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_app$ ls
Makefile  pcie_appln_dma.c  pcie_appln.h  pcie_appln_main.c
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_app$ make
gcc  -c pcie_appln_dma.c -o pcie_appln_dma.o
gcc  -c pcie_appln_main.c -o pcie_appln_main.o
gcc  pcie_appln_dma.o  pcie_appln_main.o -Wall -lm -o pcie_app
chandra@hyd-ld-i52899a:~/work/cardtest/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
chandra@hyd-ld-i52899a:~/work/cardtest/linux_pcie_app$ sudo ./pcie_app
Press CTRL+C to exit the program...
Welcome to the PCI Demo Application
-------------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
```

4.3.2.1.2 Device Information (Ask a Question)

Enter **1** to get device information, as shown in the following figure.

Figure 4-23. Device Information

```
Enter your choice: 1
Input is taken from user
-------------------------------------------
Demo_type = PolarFire PCIe Demo
Device_status = Microsemi Device Detected
Device_type = PolarFire Evaluation kit
Number of BARs enabled = 2
Bar0_add      = 0x110000c
Bar0_size     = 0x10000
Bar2_add      = 0x100000c
Bar2_size     = 0x100000

Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
```

Important: 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 (Ask a Question)

Enter **2** to blink LEDs, as shown in the following figure.

Figure 4-24. Blink LEDs

```
-----------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 2
Input is taken from user
-----------------------------------------
Enter LED data (Numeric format)
Example: 255 to blink all LEDs (1111 1111 in binary)
         85 to blink alternate LEDs (0101 0101 in binary)
-----------------------------------------
255
Success: LEDs configured to blink as per the provided pattern.
```

4.3.2.1.4 DIP Switch Status (Ask a Question)

Enter **3** to get dip switch status, as shown in the following figure.

Figure 4-25. Dip Switch Status

```
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 3
Input is taken from user
Reading DIP Switch Status...
Success: DIP switch status read successfully.

DIP Switch Status:
  SW1: ON
  SW2: OFF
  SW3: OFF
  SW4: OFF
```

4.3.2.1.5 For Interrupt (Ask a Question)

Enter **4,** then enter **1** for interrupt count or **2** for clear, as shown in the following figures.

Figure 4-26. Adding Interrupt

```
Welcome to the PCI Demo Application
-------------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 4
Input is taken from user
-----------------------------------------------
Select an option:
  1. Display ISR Counters
  2. Clear ISR Counters
-----------------------------------------------
1
ISR Counters:
  Counter 1: 1
  Counter 2: 1
  Counter 3: 1
  Counter 4: 0
```

Figure 4-27. Clearing Interrupt

```
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 4
Input is taken from user
-----------------------------------------------
Select an option:
  1. Display ISR Counters
  2. Clear ISR Counters
-----------------------------------------------
2
ISR Counters have been cleared successfully.

Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 4
Input is taken from user
-----------------------------------------------
Select an option:
  1. Display ISR Counters
  2. Clear ISR Counters
-----------------------------------------------
1
ISR Counters:
  Counter 1: 0
  Counter 2: 0
  Counter 3: 0
  Counter 4: 0
```

4.3.2.1.6 For Read/Write to Bar Space (Ask a Question)

Enter **5** to read or write to bar space, as shown in the following figure.

Figure 4-28. Read/Write to BAR Space

```
Welcome to the PCI Demo Application
-------------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 5
Input is taken from user
-------------------------------------------
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 in the hexadecimal format (example: 0x10)
0x10
Provide the data to write in the hexadecimal format (example: 0xa0)
0xa0
Write successful.


Operation completed successfully.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 5
Input is taken from user
-------------------------------------------
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 in the hexadecimal format (example: 0x10)
0x10
Read value = 0xa0


Operation completed successfully.
Select an option from the menu below:
```

4.3.2.1.7 For PCIe ConfigSpace (Ask a Question)

Enter **6,** then enter **2** for full PCIe configuration read, as shown in the following figure.

Figure 4-29. For PCIe ConfigSpace

```
Welcome to the PCI Demo Application
-------------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCI Demo Application
-------------------------------------
Enter your choice: 6
Input is taken from user
-------------------------------------------
1. Read from PCIe ConfigSpace offset
2. Read entire PCIe ConfigSpace
Enter your choice: 2
Vendor ID: 0x11aa
Device ID: 0x1556
Cmd Reg: 0x406
Stat Reg: 0x10
Revision ID: 0x0
Class Prog: 0x0
Device Class: 0x0
Cache Line Size: 0x0
Latency Timer: 0x0
Header Type: 0x0
BIST: 0x0
BAR0: Addr:0x110000c
BAR1: Addr:0x60
BAR2: Addr:0x100000c
BAR3: Addr:0x60
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: 0xff
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 (Ask a Question)

Enter **7** for PCIe DMA operation.

- Maximum transfer length is 4k < transfer length < 1 MB.
- For DDR3L/4 maximum transfer length is 4k < transfer length < 1 MB.
- For LSRAM maximum transfer length is 4k < transfer length < 1 MB.

Figure 4-30. For DMA Operation

```
Welcome to the PCI Demo Application
-----------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCIe Demo Application
-------------------------------------
Enter your choice: 7
Input is taken from user
-------------------------------------------------------

Read/write to memory using DMA
-------------------------------------------------------------
1. Continuous Dma Write to DDR3
2. Continuous Dma Read from DDR3
3. Continuous Dma write/read DDR3
4. Continuous Dma Write to DDR4
5. Continuous Dma Read from DDR4
6. Continuous Dma write/read DDR4
7. Continuous Dma Write to LSRAM
8. Continuous Dma Read from LSRAM
9. Continuous Dma write/read LSRAM
10. SG Dma Write to DDR3
11. SG Dma Read from DDR3
12. SG Dma write/read DDR3
13. SG Dma Write to DDR4
14. SG Dma Read from DDR4
15. SG Dma write/read DDR4
15
Enter the Tx size (in bytes) between 4 bytes to 1MB  (example: 4096 ):
-----------------------------------------------------------------
1048576
Enter the Rx size (in bytes) between 4 bytes   and 1MB  (example 4096):
-----------------------------------------------------------------
1048576
Enter the data pattern 1.inc 2. dec 3.rand 4.zero's 5.one's 6. AAA 7. 555
-------------------------------------------------------------------------------
7
Write throughput = 915 MBPS
Read throughput  = 524 MBPS
Write Buffer Descriptor count = 16
Read Buffer Descriptor count = 16
```

The following figure shows how to exit from the PCIe Demo Application.

Figure 4-31. Exit from the PCIe Demo Application

```
Welcome to the PCI Demo Application
-----------------------------------------
Initialization successful: PCI device [/dev/MS_PCI_DEV] is ready for use.
Select an option from the menu below:
-------------------------------------
1. Display device information
2. Blink LEDs
3. Show DIP switch status
4. Handle Interrupt
5. Read/Write to BAR space
6. Show PCIe configuration space information
7. Perform PCIe DMA operation
8. Exit from the PCIe Demo Application
-------------------------------------
Enter your choice: 8
Input is taken from user
Exiting from the PCI Demo Application
PCIe Device [/dev/MS PCI DEV] closed
```

4.4 Throughput Summary of Evaluation Kit (Ask a Question)

The following tables list the throughput values observed.

Table 4-1. PolarFire Throughput Summary—PCIe Continuous DMA Mode

DMA Transfer Type	DMA Size	Throughput (Mbps)	Average Throughput (Mbps)
PC to LSRAM	64K	1091	1069
LSRAM to PC		1104	1147
Both PC to and from LSRAM		1124/1154	1093/1149
PC to DDR4	64K	998	970
DDR4 to PC		523[1]	523
Both PC to and from DDR4		998/523	972/523
PC to DDR3L	64K	468	460
DDR3L to PC		327[1]	327
Both PC to and from DDR3L		468/327	465/327

Important:
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 DDR3L/ DDR4.

Table 4-2. 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 DDR3L	1 MB	469	472
DDR3L to PC		325[1]	325
Both PC to and from DDR3L		473/325	473/325

Important:
1. The PCIe DMA performs maximum of 32 beat AXI burst transactions (not maximum of 256 beat of AXI4), which causes low read performance of DDR3L/ DDR4.

Table 4-3. 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 DDR3L	1 MB	574	574
DDR3L to LSRAM		553	553
Both LSRAM to and from DDR3L		574/554	574/554
DDR4 to DDR3L	1 MB	574	574
DDR3L to DDR4		553	553
Both DDR4 to and from DDR3L		574/553	574/553

Important: DDR3L throughput is less due to the AXI interconnect CDC path limitation.

4.5 Throughput Summary of Splash Kit (Ask a Question)

The following table lists the throughput values observed.

Table 4-4. PolarFire Throughput Summary—PCIe Continuous DMA Mode

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

Important:

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

Table 4-5. 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

Important:

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 4-6. 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: DDR3L and DDR4 Power Measurement (Ask a Question)

5.1 DDR3L (Ask a Question)

The PolarFire Evaluation Kit board has a current sense resistor (R118) for 1.5V power rail. Measure the voltage across the R118 using test points TP134 and TP135 and use the following equations to get the DDR3L power. This measurement includes PolarFire DDR3L IO power consumption and actual Micron DDR3L memory power consumption.

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

Power (mW) = Current × Voltage

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

Current (mA) = 6.3 ÷ 0.01 = 630 mA
Power = 630 × 1.5 = 945 MW

5.2 DDR4 (Ask a Question)

The PolarFire Evaluation Kit board has a current sense resistor (R222) for 1.2V 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.

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

Power (mW) = Current × Voltage

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

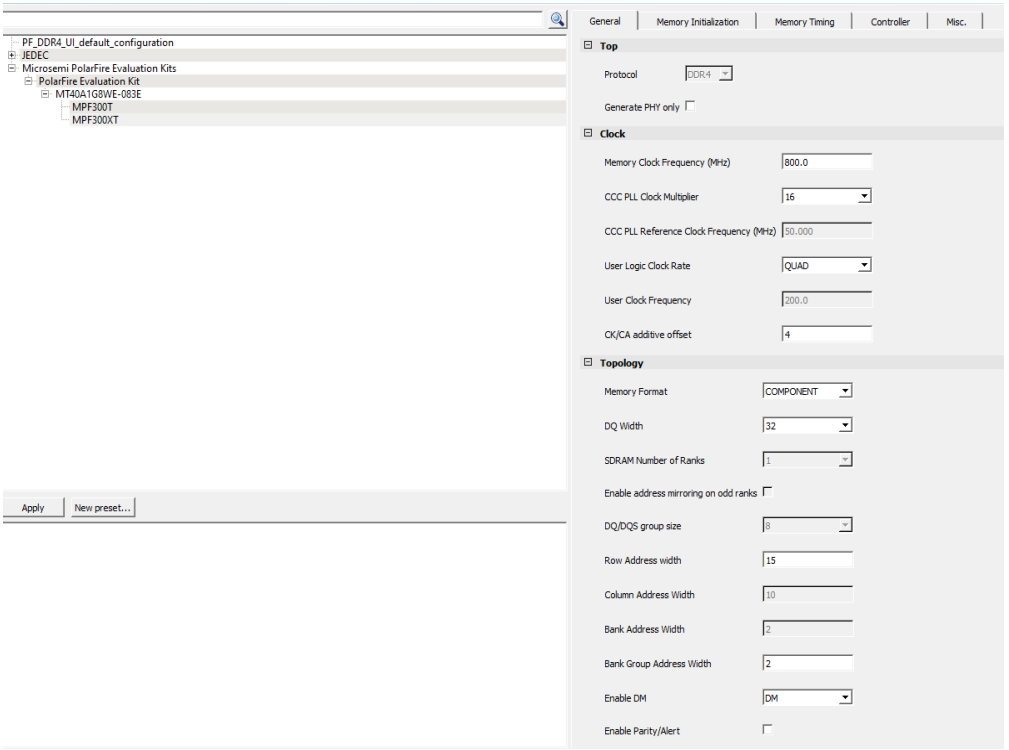
Current (mA) = 2.4 ÷ 0.01 = 240 mA
Power = 240 × 1.2 = 288 MW

6. Appendix 2: DDR4 Configuration (Ask a Question)

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.

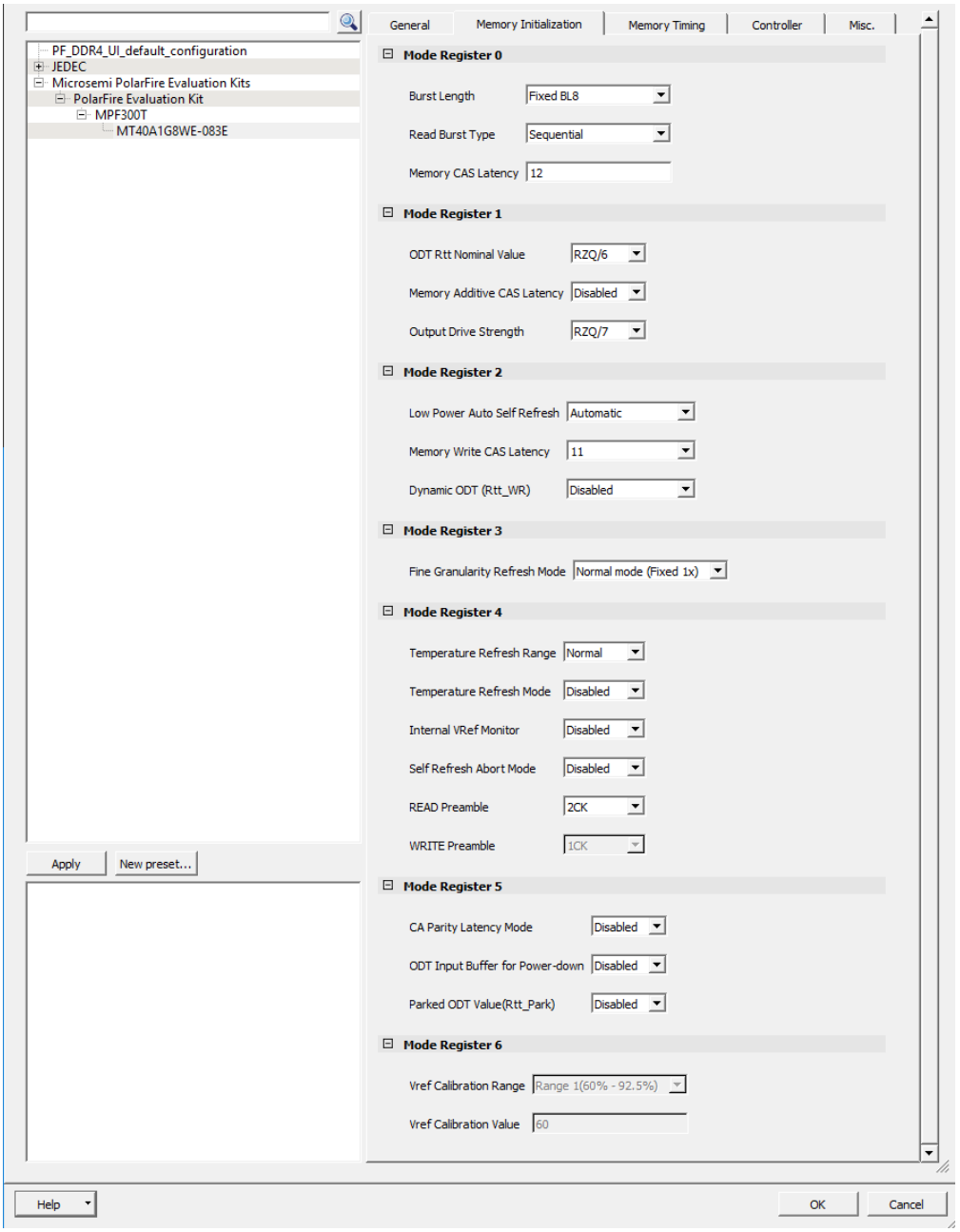
The following figure shows the general configuration settings for the DDR4 memory.

Figure 6-1. DDR4 Configurator



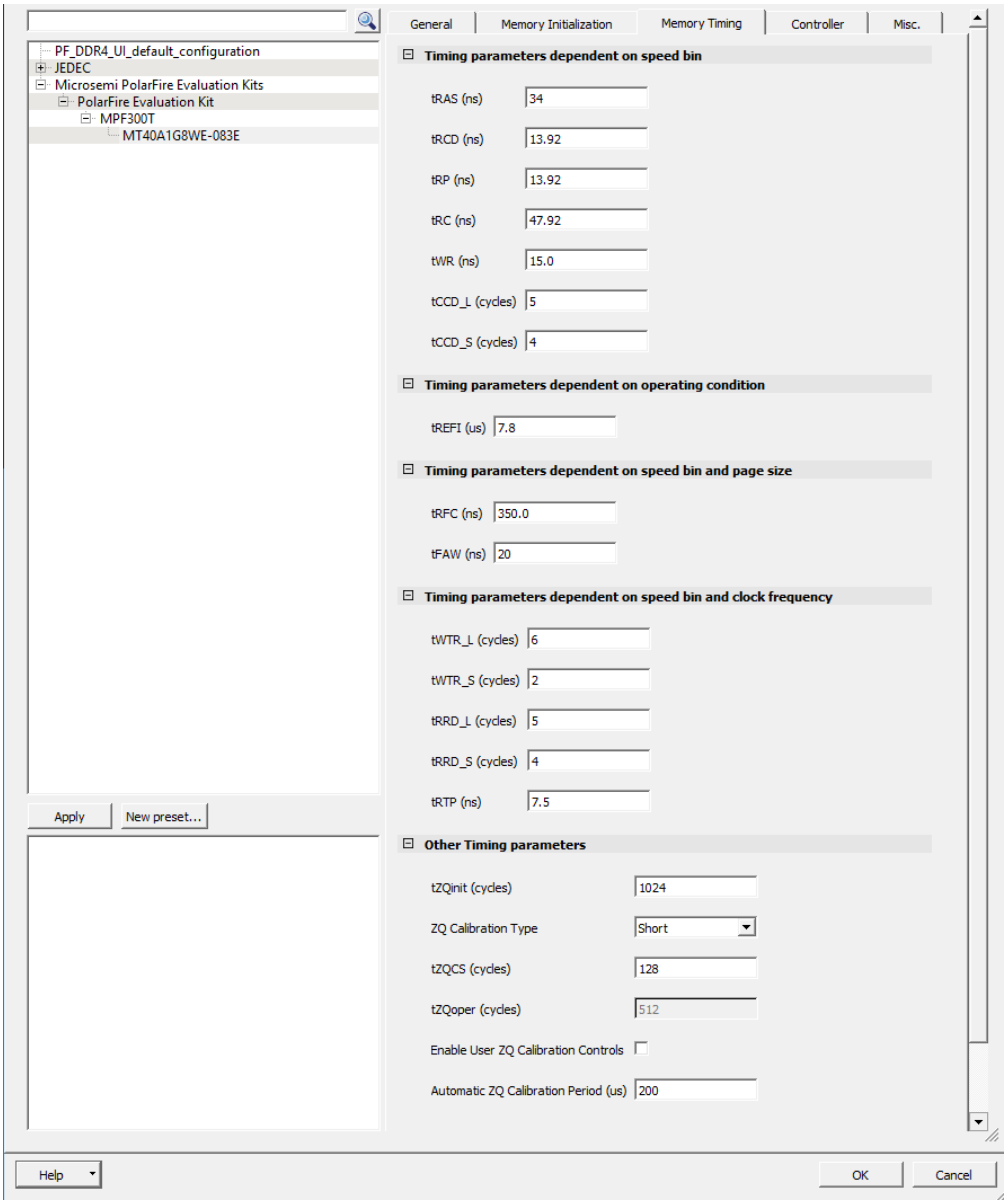
The following figure shows the initialization configuration settings for the DDR4 memory.

Figure 6-2. DDR4 Configurator—Memory Initialization



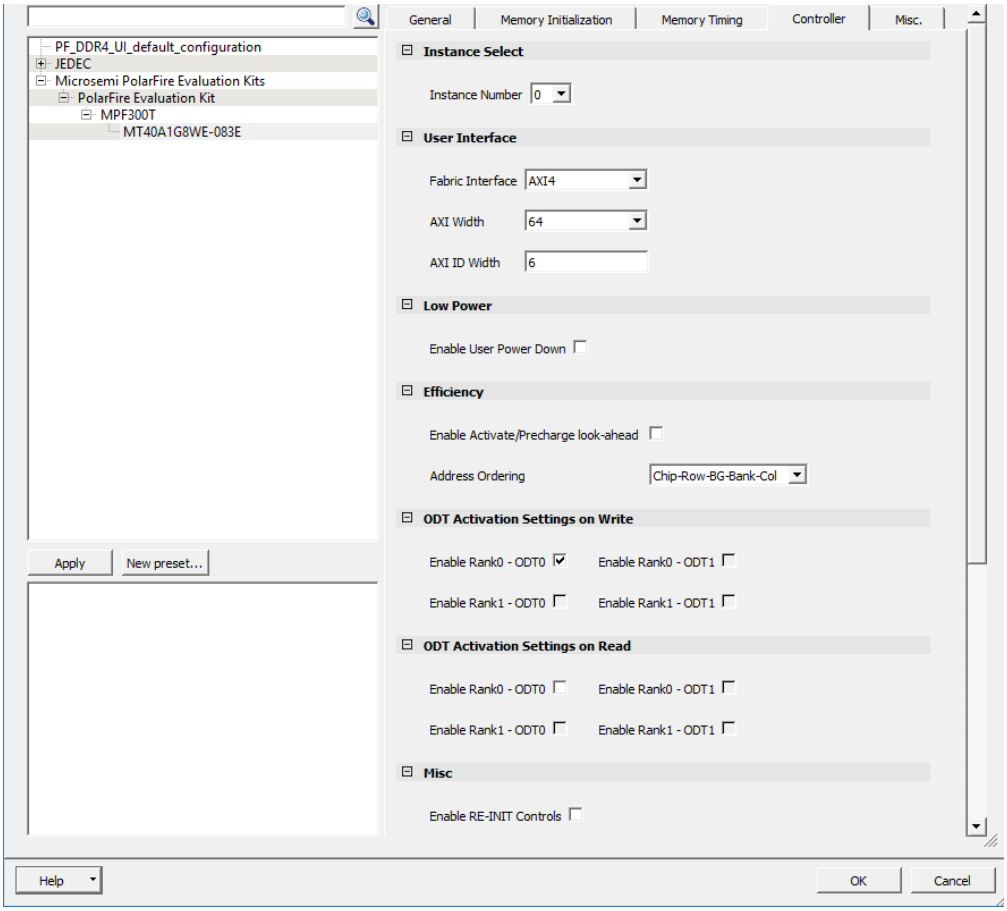
The following figure shows the timing configuration settings for the DDR4 memory.

Figure 6-3. DDR4 Configurator—Memory Timing



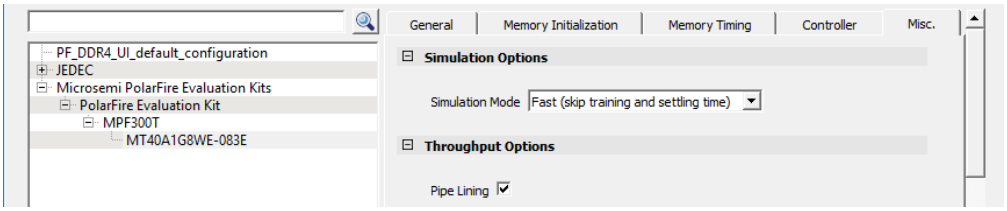
The following figure shows controller configuration settings for the DDR4 memory.

Figure 6-4. DDR4 Configurator—Controller



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

Figure 6-5. DDR4 Configurator—Misc



7. Appendix 3: DDR3L Configuration (Ask a Question)

For information about Rev E or later Kit DDR3L configurations, see *Appendix 3 - DDR3 Configuration* of AN4997: PolarFire FPGA Building a Mi-V Processor Subsystem.

8. Appendix 4: Programming the Device Using FlashPro Express (Ask a Question)

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_an4597_df\Programming_files`

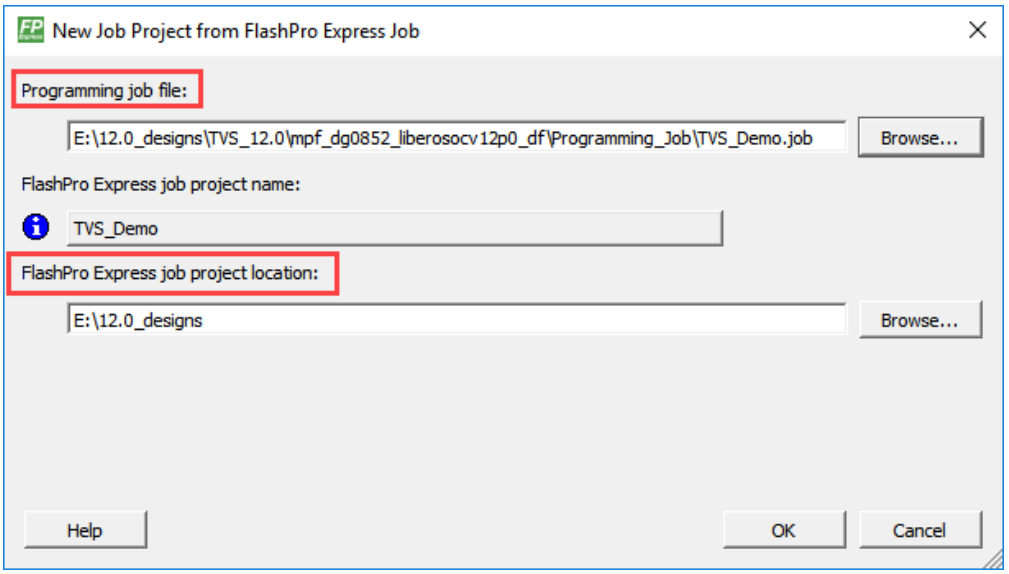
To program the device, perform the following steps:

1. Ensure that the jumper settings on the board are the same as listed in Table 2-1 (for evaluation) and Table 3-1 (for splash).

   Important: 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**.
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_an4597_df\Programming_files`.
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

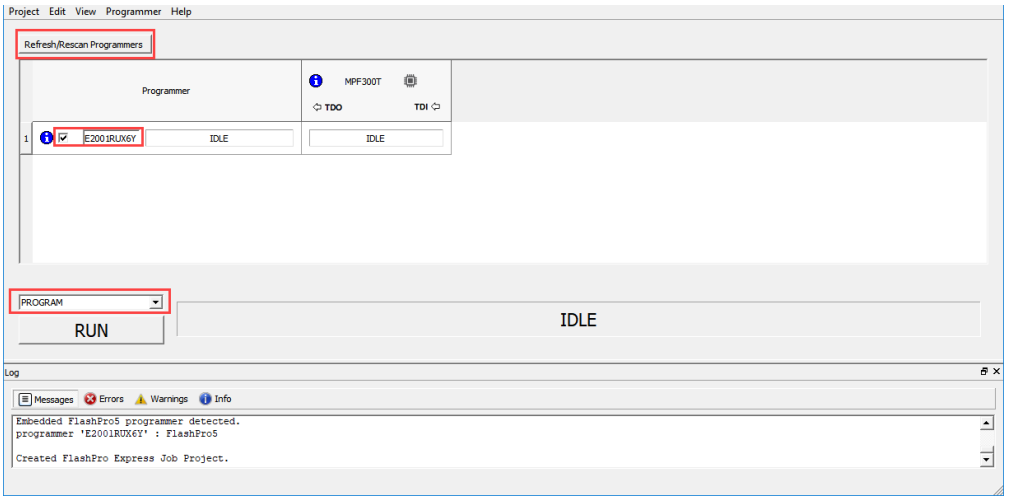
Figure 8-1. New Job Project from FlashPro Express Job



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

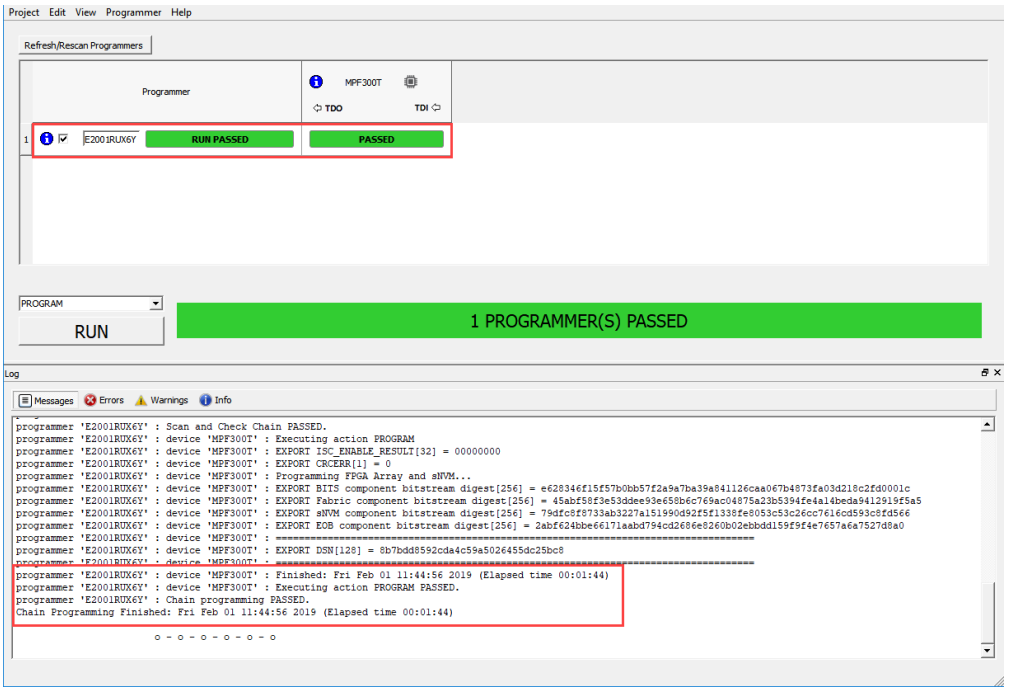
9. The FlashPro Express window appears, as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan Programmers**.

Figure 8-2. Programming the Device



10. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure. To run the PCIe EndPoint demo, see Running the Demo section.

Figure 8-3. FlashPro Express—RUN PASSED



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

9. Appendix 5: Running the TCL Script (Ask a Question)

TCL scripts are provided in the design files folder under directory `HW`. If required, the design flow can be regenerated from Design Implementation till generation of job file.

To run the TCL, perform the following steps:

1. Launch the Libero software.
2. Select **Project** > **Execute Script....**
3. Click **Browse** and select `script.tcl` from the downloaded `HW` directory.
4. Click **Run**.

After successful execution of TCL script, Libero project is created within `HW` directory. For more information about TCL scripts, see the following:

- `mpf_an4597_df/HW/Eval_Kit/TCL_Script_readme.txt`
- `mpf_an4597_df/HW/Splash_Kit/TCL_Script_readme.txt`

For more details on TCL commands, see Tcl Commands Reference Guide. Contact Technical Support for any queries encountered while running the TCL script.

10. Appendix 6: References (Ask a Question)

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 PolarFire Family Transceiver User Guide.
- For more information about PF_PCIE, see PolarFire Family PCI Express User Guide.
- Fore more information about PF_CCC, see PolarFire Family Clocking Resources User Guide.
- Fore more information about DDR3L/DDR4 memory, see PolarFire Family Memory Controller User Guide.
- For more information about Libero, ModelSim and Synplify, see the Libero SoC Documentation 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. This user guide can be downloaded from the Libero SoC Catalog.
- For more information about CoreAHBtoAPB3, see CoreAHBtoAPB3 Handbook. This user guide can be downloaded from the Libero SoC Catalog.
- For more information about CoreUART, see CoreUART User Guide. This user guide can be downloaded from the Libero SoC Catalog.

11. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the current publication.

Table 11-1. Revision History

Revision	Date	Description
C	02/2025	The following is the list of changes in revision C of the document: • The drivers have been updated to enhance compatibility with Windows® 10 and Linux® Kernel 6.8. • Updated the document for Libero® v2024.2. • Updated the `.job` filepath and TCL script filepath throughout the document. • Updated Figure 1-1 in the Demo Design section. • Updated Figure 1-5, Figure 1-6 and Figure 1-7 in the Design Implementation section. • Updated all GUI figures in the Running the PCIe Demo Application section. • Updated Figure 6-1 in the Appendix 2: DDR4 Configuration section.
B	02/2024	The following is the list of changes in revision B of the document: • Added a note to the DDR3L section. • Updated Appendix 3: DDR3L Configuration section.
A	08/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated to DS00004597A from 50200756. • The document ID was updated to AN4597 from DG0756. • Replaced all Microsemi links with Microchip links. • Renamed the DDR3 to DDR3L throughout the document. • Updated Figure 1-4. • Updated Figure 1-5 and Figure 1-6. • Updated Figure 1-7 and Figure 1-8. • Updated Figure 1-9. • Updated content of Simulating the Design. • Updated design files location in Installing PCIe Demo Application. • Updated Figure 4-10, Figure 4-11, Figure 4-12, Figure 4-13 and Figure 4-14. • Updated Figure 4-15, Figure 4-16, Figure 4-18 and Figure 4-19. • Updated `.job` file location in Appendix 4: Programming the Device Using FlashPro Express. • Updated TCL scripts file location Appendix 5: Running the TCL Script. • Added a new section Appendix 3: DDR3L Configuration.
10.0	—	Added Appendix 5: Running the TCL Script.
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.
8.0	—	The document was updated for Libero SoC v12.0 release.
7.0	—	Merged Splash kit related content and updated the document for Libero SoC PolarFire v2.3 release.
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: DDR3L and DDR4 Power Measurement.

Table 11-1. Revision History (continued)

Revision	Date	Description
5.0	—	The document was updated for Libero SoC PolarFire v2.1 release.
4.0	—	The document was updated to include features and enhancements introduced in the Libero SoC PolarFire v2.0 release.
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 Appendix 4: Programming the Device Using FlashPro Express and Running the Demo. • List of reference was added. For more information, see Appendix 6: References.
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.
1.0	—	The first publication of this document.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-0700-4

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.