# PolarFire FPGA PCIe Root Port Application Note

## AN4664

## Introduction (Ask a Question)

Microchip PolarFire® FPGAs support 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 PF_PCIE IP configurator in the Libero® SoC software.

The PF_PCIE IP core is compliant with the PCI Express Base Specification, Revision 3.0 with Gen1/2. 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 PolarFire Family PCI Express User Guide.

# Table of Contents

# 1. PolarFire FPGA PCIe Root Port (Ask a Question)

This document describes the Root Port capabilities of the PolarFire FPGA PCIe controller using Mi-V soft processor. The PCIe Root Port capabilities like the enumeration of an Endpoint device, low-speed and high-speed data transfers are demonstrated using the PCIe Root Port Demo GUI application.

The demo design includes a Mi-V soft processor, which initiates PCIe control and data plane functions. For more information about the PCIe Root Port design implementation and the necessary blocks and IP cores instantiated in Libero SoC, see Demo Design.

The demo 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 Setting Up the Demo.
- Using Libero SoC: To program the device using Libero SoC, see Libero Design Flow. Use this option when the demo design is modified.

To run the demo, perform the following steps:

- The Root Port demo design must be programmed on a PolarFire Evaluation board.
- The Endpoint demo design must be programmed on another PolarFire Evaluation board.
- Both the boards must be connected using a PCIe Adapter card.

For more information about setting up the PCIe Root Port demo, see Setting Up the Demo.

## 1.1 Design Requirements (Ask a Question)

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

**Table 1-1.** Design Requirements

| Requirement | Version |
|---|---|
| Operating System | 64-bit Windows® 10 and 11 |
| **Hardware** | |
| Two PolarFire® Evaluation kits (MPF300TS-FCG1152I) | Rev D or later |
| Microchip PCIe Adapter card | PCIe-ROOTPORT-AD |
| **Software** | |
| Libero® SoC[1] | For more information about the software versions used with this reference design, see the `readme.txt` file provided in the design files. |
| SoftConsole® | |
| Microchip FPGA_GUI_Pack[2] | Version 1.0 |

**Important:**

1. Libero® SmartDesign and configuration screenshots shown in this guide are for illustration purpose only. For latest updates, open the Libero design.
2. Microchip FPGA_GUI_Pack includes the necessary run-time engine and drivers for the operation of Microchip FPGA Demo GUI applications.

## 1.2 Prerequisites (Ask a Question)

Before you start, perform the following steps:

1. Download and install Libero® SoC on the host PC from the following location: Libero SoC Documentation.

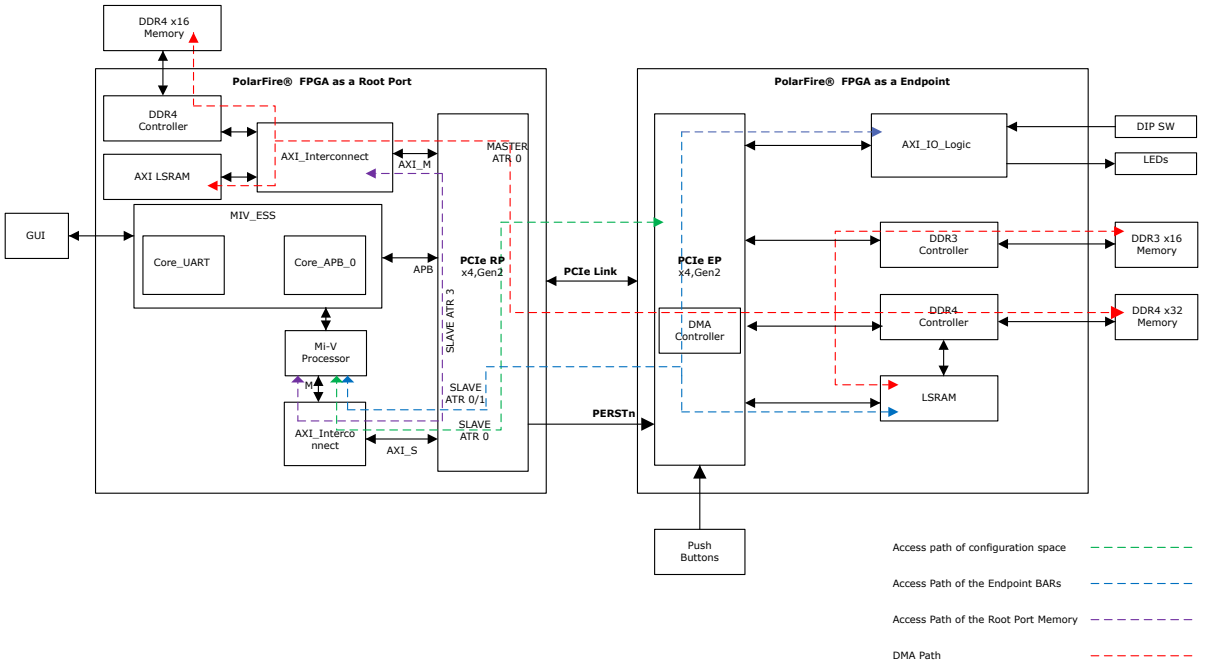
> **Important:** The latest versions of ModelSim®, Synplify Pro®, and FTDI drivers are included in the Libero SoC installation package.

2. Download demo design files from: www.microchip.com/en-us/application-notes/AN4664; and download the PCIe EndPoint design files from: www.microchip.com/en-us/application-notes/AN4597.
3. To create Libero Project, see Appendix 3: Running the TCL Script.
4. Install the PolarFire® PCIe Root Port Demo GUI application by running the `setup.exe` file available in the design files folder:
   `<$Design_Files_Directory>\mpf_an4664_df\GUI_Installer`. At the end of the installation, you may be prompted to download and install the FPGA_GUI_Pack, if it is not already available on your system.
5. Alternatively, you can manually download and install the Microchip FPGA_GUI_Pack.

## 1.3 Demo Design (Ask a Question)

The following figure shows the top-level block diagram of the PCIe Root Port design. The PolarFire PCIe Root port can establish a PCIe link with any PCIe Endpoint or Bridge. The user application enumerates the Endpoint device using the ECAM (enhanced configuration access mechanism) feature. The user application also initiates the AXI transactions from the Root port. These AXI transactions are converted to PCIe Configuration space or memory transactions (CfgWr/CfgRd/MWr/MRd) to Endpoint.

**Figure 1-1.** Block Diagram



As shown in above figure, the following points describe the data flow in the PCIe Root Port design:

1. The MIV_ESS_C0 UART block interfaces with the GUI.
2. The Mi-V soft processor reads/writes data to the MIV_ESS_C0 UART block through the core MIV_ESS_C0 APB.
3. The Mi-V soft processor forwards the corresponding PCIe command to the PF_PCIE_0 block through the PCIe_APB/PCIe_AXI slave interface.
4. The PCIe request and completion TLPs are transmitted and received between the Root Port and the Endpoint through the serial link.
5. Inbound TLPs are reflected as AXI transactions on the AXI_1_master port of PF_PCIE.
6. The Mi-V soft processor uses the PCIe_AXI bus interface to read the data from AXI_1_SLAVE.
7. The Mi-V soft processor uses the core MIV_ESS_C0 APB and writes the data to the MIV_ESS_C0 UART slave interface to forward the data to the GUI.

As shown in above figure, the following points describe the DMA flow from Root Port to Endpoint:

1. The Mi-V soft processor enumerates the Endpoint by accessing the Root Port and the Endpoint configuration space through the SLAVE ATR0 path.

> **Important:** ATRs (address translation registers) perform address translation from PCIe address space to the AXI master. For more information about ATRs, see PolarFire Family PCI Express User Guide.

2. The Mi-V soft processor accesses the Endpoint BAR 0/2 through the SLAVE ATR 0/1 path.
3. The Mi-V soft processor accesses the Root Port LSRAM memory through the SLAVE ATR 3 path.
4. The DMA is performed according to the user selection on the GUI application.

### 1.3.1 Memory and Peripheral Address Map (Ask a Question)

This section lists the memory and peripheral address map of the Root Port demo design.

The address maps of the Mi-V peripherals and main memory are:

- APB interface: 0x60000000 to 0x6FFFFFFF
- AXI interface: 0x70000000 to 7FFFFFFF
- TCM memory interface : 0x80000000 to 0x8FFFFFFF.

The following table lists the address map of the Bus interfaces connecting Mi-V to PF_PCIE.

**Table 1-2.** Mi-V and PF_PCIE Address Maps

| Bus Interface/Component | Description | Memory Map |
|---|---|---|
| PCIe_APB | This bus interface is used to access the PCIe register | 0x63000000 to 0x63FFFFFF |
| MIV_ESS_C0 UART | This block establishes a UART interface to connect the Mi-V processor to the external world | 0x61000000 to 0x61FFFFFF |
| PCIE_1_PERST_OUT_N | This is used to generate the PERSTn signal for the link partner that is connected to the Root port | 0x6300A150 |

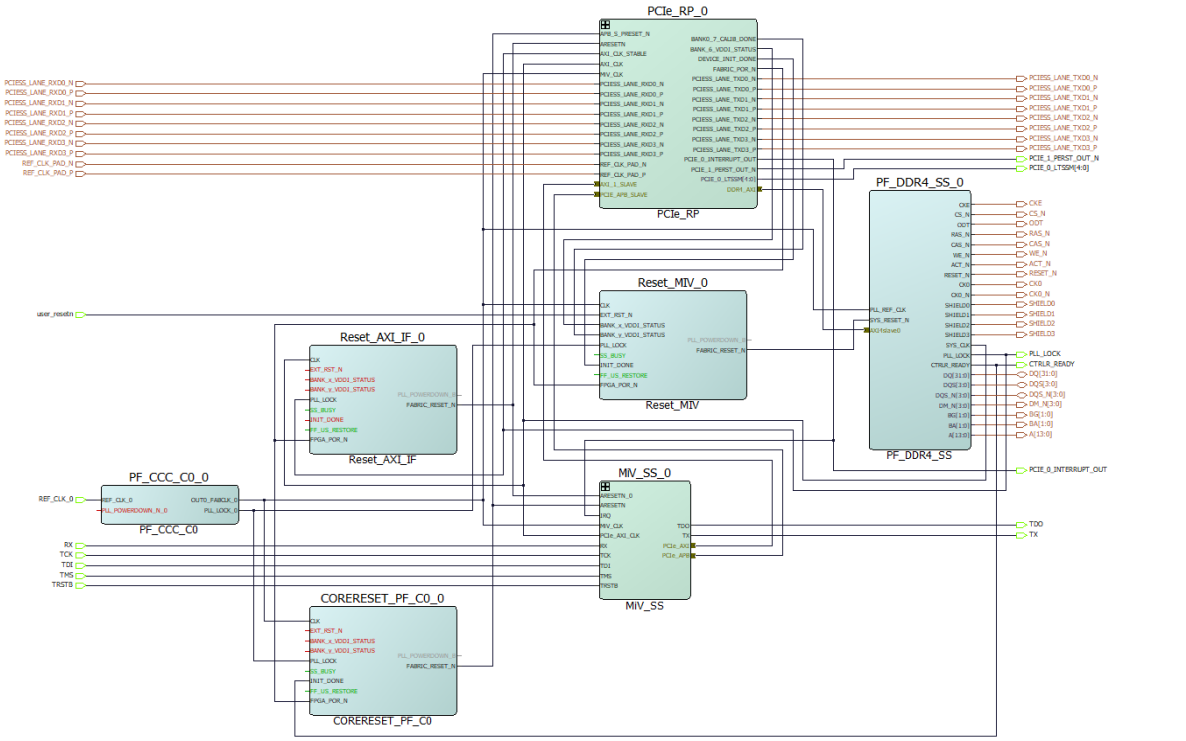
| .....continued | | |
|---|---|---|
| **Bus Interface/Component** | **Description** | **Memory Map** |
| PCIe_AXI | This Bus interface is the PCIe AXI slave for EP configuration or BAR space access | 0x70000000 to 0x7000FFFF—Configuration space (Mi-V configures through PCIe APB)<br>0x71000000 to 0x7100FFFF—EP BAR0 space<br>0x72000000 to 0x7200FFFF—EP BAR2 space<br>0x73000000 to 0x73FFFFFF—RP AXI Master—LSRAM/DDR4 (Mi-V configures through PCIe APB) |
| TCM | This block is the main memory of the Mi-V processor | 0x80000000 to 0x8FFFFFFF |

The PF_PCIE block connects to the DDR4 and LSRAM blocks through the AXI_1_master Bus interface. The address maps of DDR4 and LSRAM are 0x10000000 to 0x1FFFFFFF and 0x00000000 to 0x00000FFF, respectively.

### 1.3.2 Design Implementation (Ask a Question)

The following figure shows the Libero SoC software design implementation of the PCIe Root Port demo design.

**Figure 1-2.** PCIe Root Port Demo Design




The top-level design includes the following SmartDesign components and memory controller subsystems:

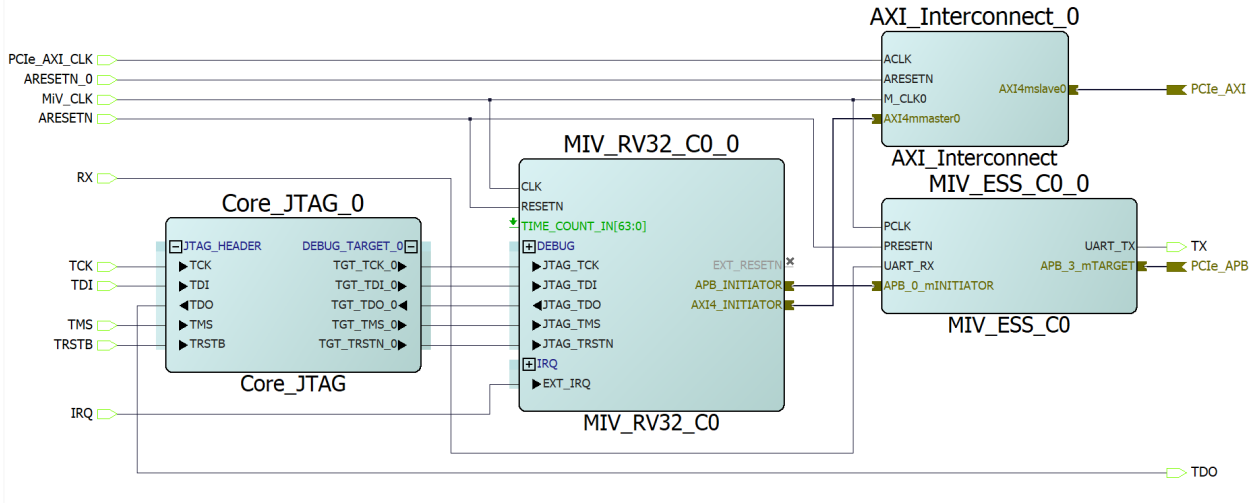
- MIV_SS_0
- PCIe_RP_0
- DDR4

**Important:** PERSTn is a fundamental reset signal defined in both **PCI Express Base Specification** and **PCI Express Card Electromechanical Specification**. It is a reset signal issued by the Root port through PCIe slots to reset the entire PCIe fabric hierarchy. The Root port firmware running on the MI-V processor can assert the PERSTn signal through PCIE_APB_SLAVE interface. When the host is power cycled, the PERSTn signal is asserted by the PCIe_INIT_MONITOR_0 until the PCIe controller in the Root port is initialized.

### 1.3.2.1 Mi-V Subsystem (Ask a Question)

The following figure shows the sub-blocks of MIV_SS_0.

**Figure 1-3.** MIV_SS_0 SmartDesign




#### 1.3.2.1.1 MIV_RV32_C0 (Ask a Question)

The MIV_RV32_C0 (MIV_RV32_C0) is configured with a Reset Vector Address of 0x80000000. After reset, the processor starts executing the instructions from this address. The main memory of the processor address ranges from 0x80000000 to 0x8FFFFFFF.

The AXI_Interconnect_0 and MIV_ESS_C0 block connects the MIV_RV32_C0 block to:

- The PF_PCIE_0 block through PCIe_APB slave interface. The MIV_RV32_C0 block accesses the PCIe control registers through the PCIe_APB slave interface.
- The PF_PCIE_0 block through PCIe_AXI slave interface.
- The MIV_ESS_C0 UART block through a APB slave interface.

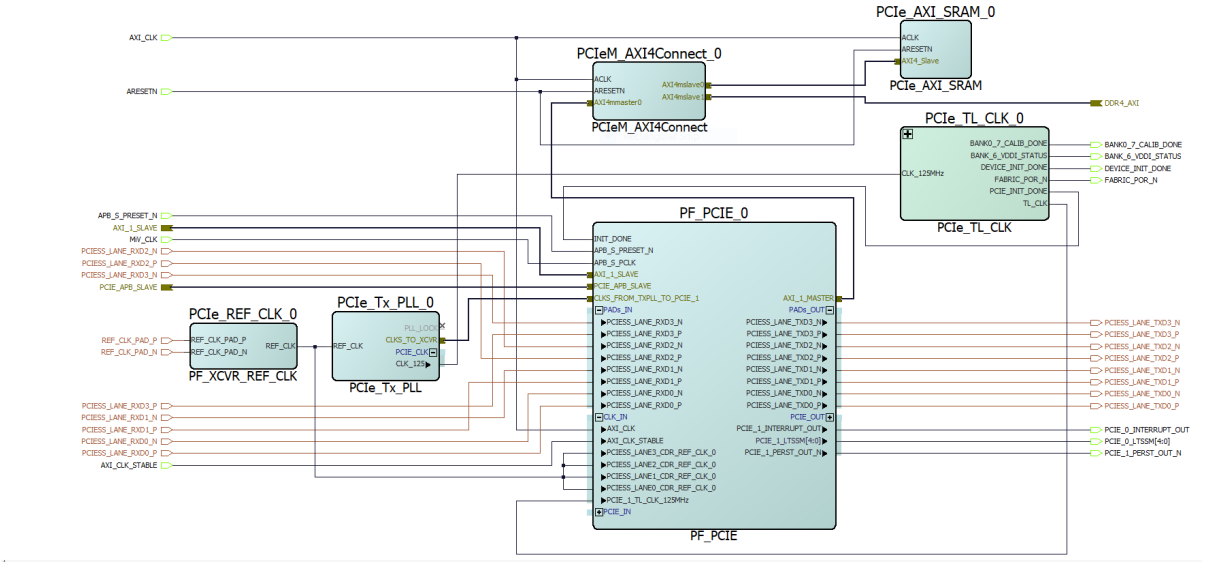
These slaves are connected at the following addresses:

- PCIe_APB slave: 0x63000000
- PCIe_AXI: 0x70000000
- MIV_ESS_C0 UART_ APB slave: 0x61000000

### 1.3.2.2 PCIe Rootport Subsystem (Ask a Question)

The following figure shows the sub-blocks of PCIe_RP_0.

**Figure 1-4.** PCIe_RP SmartDesign




### 1.3.2.2.1 PF_PCIE_0 (Ask a Question)

The PF_PCIE_0 IP block is used to configure the PCIESS as a Root Port (PCIe 1). PCIESS block is configured for × 4 lanes, 5 Gbps data rate, and APB interface for PCIe Controller access.

### 1.3.2.2.2 PCIe_Tx_PLL_0 (Ask a Question)

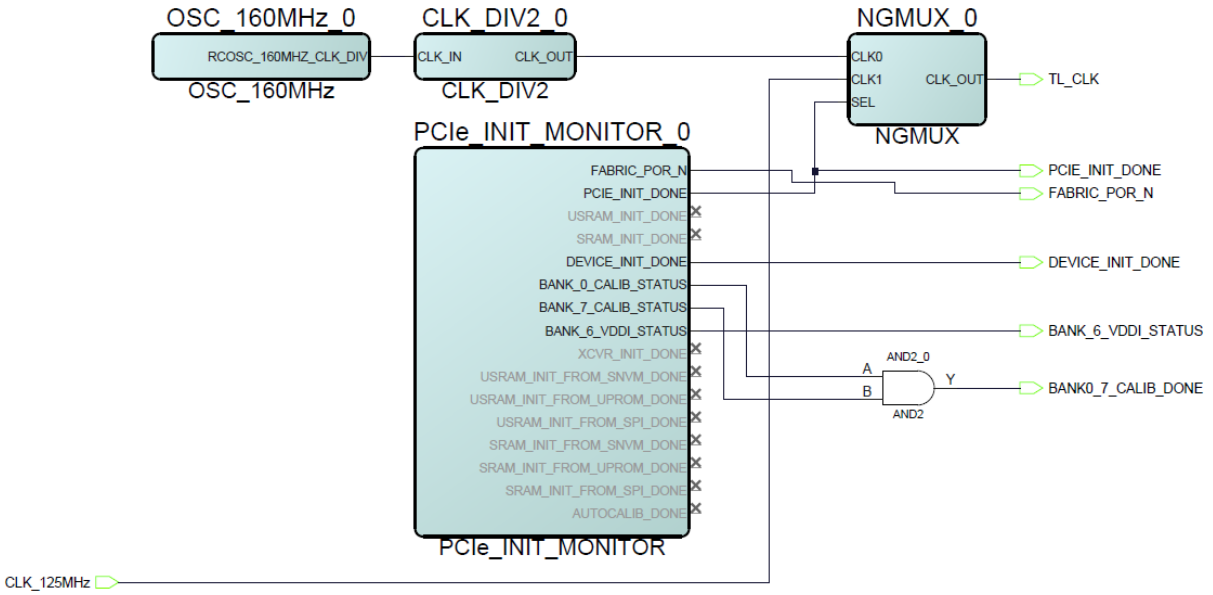
The PCIe_Tx_PLL_0 (Transmit PLL) is configured for a 100 MHz **Reference Clock** and a 5000 Mbps **Desired Output Bit Clock**.

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

### 1.3.2.2.3 PCIe_TL_CLK_0 SmartDesign (Ask a Question)

The following figure shows the PCIe_TL_CLK SmartDesign implementing PCIe TL CLK for PolarFire devices. 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.

**Figure 1-5.** PCIe_TL_CLK_0 SmartDesign




### 1.3.2.2.4 PCIeM_AXI4Connect_0 (Ask a Question)

The PCIeM_AXI4Connect_0 (CoreAXI4Interconnect) bus is configured for a single master and two slaves and used to connect the PF_PCIE_0 with PCIe_AXI_SRAM_0 and DDR4 for DMA operations.

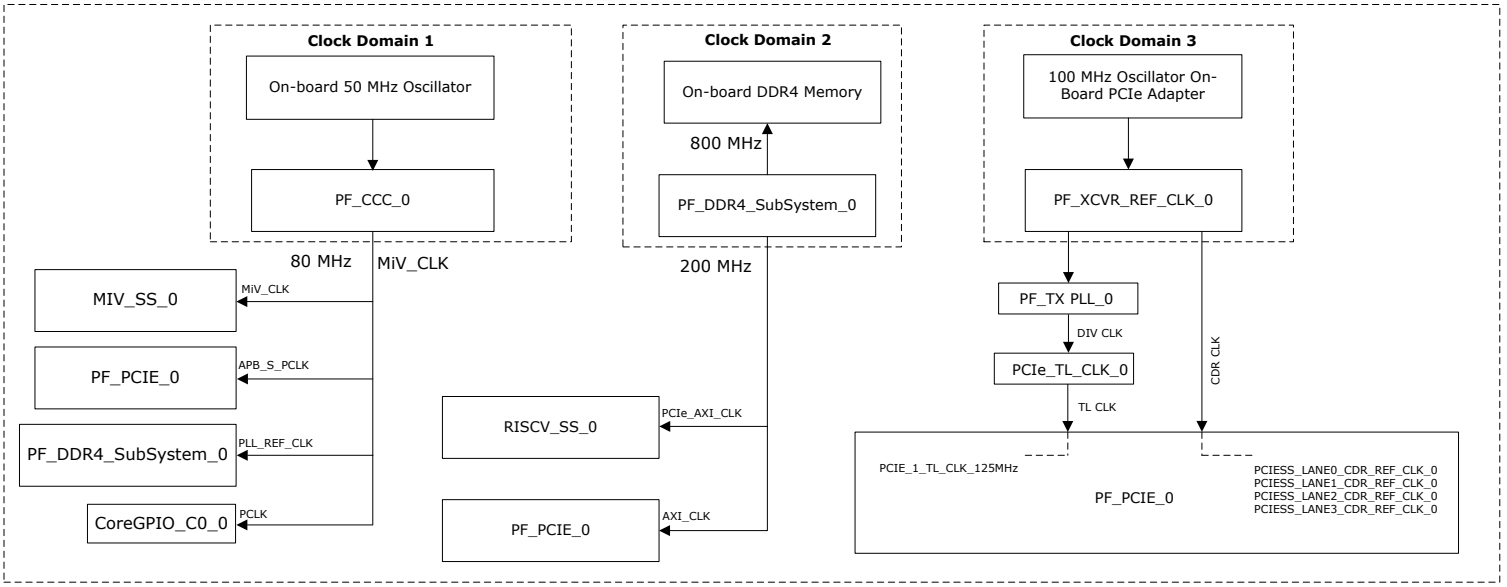
### 1.3.2.3 DDR4 Subsystem (Ask a Question)

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.

## 1.4 Clocking Structure (Ask a Question)

The following figure shows the clocking structure of the demo design.

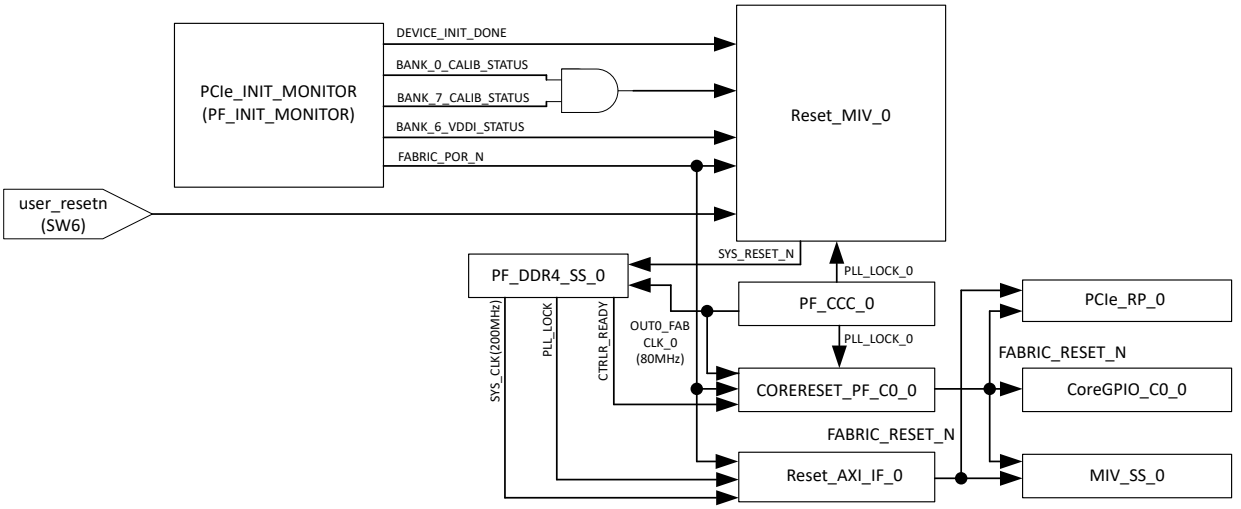
**Figure 1-6.** Clocking Structure



## 1.5 Reset Structure (Ask a Question)

The following figure shows the reset structure of the PCIe Root port demo design.

**Figure 1-7.** Reset Structure




The Reset_AXI_IF_0(CoreReset_PF) block synchronizes “PLL_LOCK” signal of PF_DDR4_SS_0 IP with the DDR4 system clock(200MHz) to generate FABRIC_RESET_N signal, which drives the PCIe_RP_0 and MIV_SS_0 blocks.

The Reset_MIV_0(CoreReset_PF) block synchronizes the external user_resetn (SW6 on the PolarFire Evaluation board) and DEVICE_INIT_DONE(PF_INIT_MONITOR) together with the RISCV system clock (80 MHz) to generate the SYS_RESET_N, which drives the PF_DDR4_SS block.

The CORERESET_PF_C0_0(CoreReset_PF) block synchronizes “CTRLR_READY” signal of PF_DDR4_SS_0 IP with the RISCV system clock (80 MHz) to generate FABRIC_RESET_N signal, which drives the PCIe_RP_0, MIV_SS_0.

For more information about device initialization, see PolarFire Family Power-Up and Resets User Guide.

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

# 2. Libero Design Flow (Ask a Question)

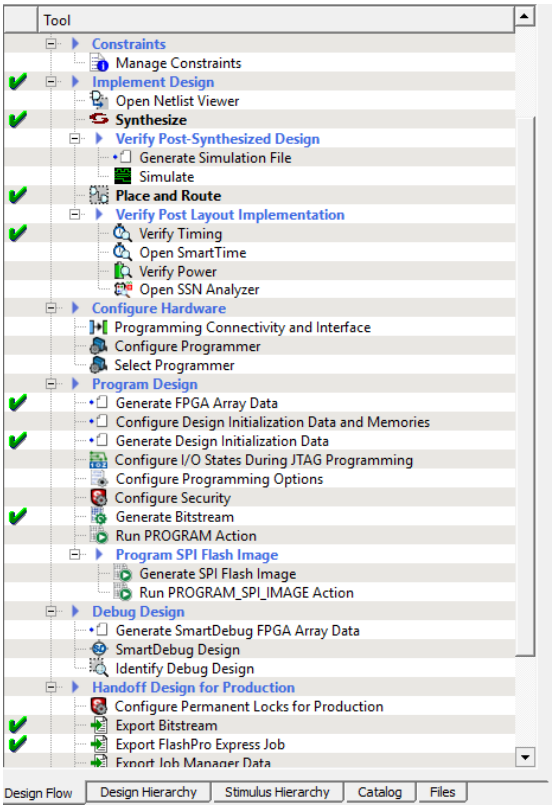
This chapter describes the Libero design flow of the demo design. The Libero design flow involves the following steps:

1. Synthesize
2. Place and route
3. Verify Timing
4. Configure Design Initialization Data and Memories
5. Generate Bitstream
6. Run PROGRAM Action

**Important:** To initialize the TCM in PolarFire using the system controller, a local parameter **l_cfg_hard_tcm0_en**, in the `miv_rv32_subsys_pkg.v` file must be changed to 1'b1 prior to synthesis. See the TCM section in the *MIV_RV32 Handbook*. This user guide can be downloaded from the Libero SoC Catalog.

The following figure shows these options in the Design Flow tab.

**Figure 2-1.** Libero Design Flow Options




## 2.1 Synthesize (Ask a Question)

To synthesize the design, perform the following steps:

1. From the **Design Flow** window, double click **Synthesize**.
   When the synthesis is successful, a green tick mark appears.

2. Right click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab.

We recommend viewing the `top.srr` and the `top_compile_netlist.log` files for debugging synthesis and compile errors.

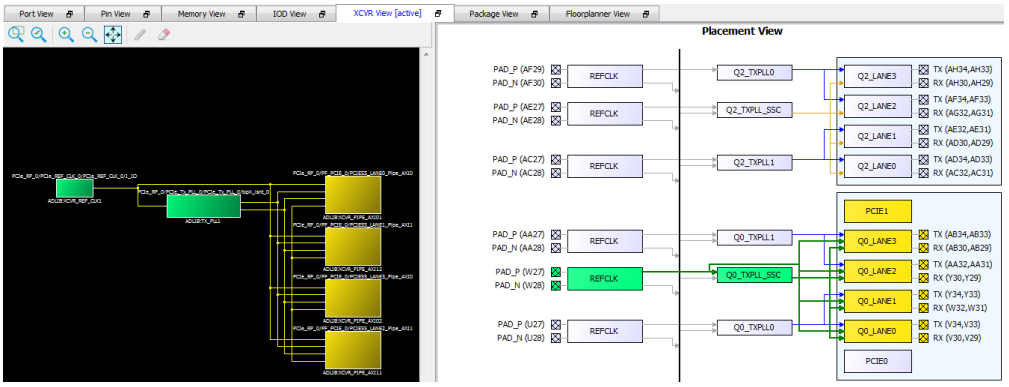
## 2.2 Place and Route (Ask a Question)

To place and route the design, the Transmit PLL (TX_PLL), XCVR_REF_CLK, PF_XCVR TX and RX lane, and the PF_DDR4_SS_0 must be placed using the **I/O Editor**.

To place and route the design, perform the following steps:

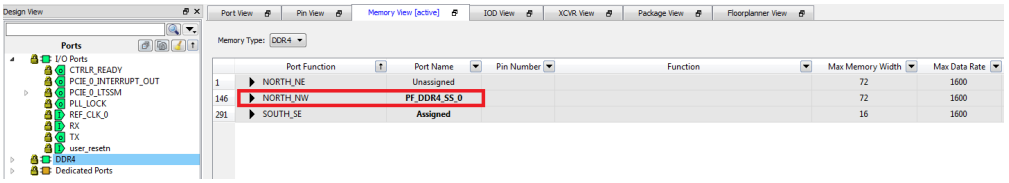
1. From the **Constraints Manager** window, place the Transmit PLL, XCVR_REF_CLK, and PF_XCVR TX and RX lanes using **I/O Editor**. See the following figure.

**Figure 2-2.** I/O Editor—XCVR View




2. Place the PF_DDR4_SS_0 at NORTH_NW location, see the following figure.

**Figure 2-3.** PF_DDR4_SubSystem_0 Placement




3. From the **Design Flow** window, double-click **Place and Route**.
   When place and route is successful, a green tick mark appears, as shown in Figure 2-1.
4. Right-click **Place and Route** and select **View Report** to view the place and route report and log files in the **Reports** tab.

We recommend viewing the `top_place_and_route_constraint_coverage.xml` file for place and route constraint coverage.

### 2.2.1 Resource Utilization (Ask a Question)

The resource utilization report is written to the `top_layout_log.log` file in the **Reports tab** > **RP_Demo_Top report** > **Place and Route**. Table 2-1 lists the resource utilization of the design after place and route. These values may vary slightly for different Libero runs, settings, and seed values.

**Table 2-1.** Resource Utilization

| Type | Used | Total | Percentage |
|---|---|---|---|
| 4LUT | 31943 | 299544 | 10.66 |
| DFF | 23193 | 299544 | 7.74 |

.....continued

| Type | Used | Total | Percentage |
|---|---|---|---|
| I/O Register | 0 | 1536 | 0 |

## 2.3 Verify Timing (Ask a Question)

To verify timing, perform the following steps:

1. From the **Design Flow** window, double click **Verify Timing**.
2. When the design successfully meets the timing requirements, a green tick mark appears, as shown in Figure 2-1.
3. Right click **Verify Timing** and select **View Report** to view the verify timing report and log files in the **Reports** tab.

## 2.4 Generate FPGA Array Data (Ask a Question)

To generate FPGA array data, In the Design Flow window, double click **Generate FPGA Array Data**.

A green tick mark is displayed after the successful generation of the FPGA array data, as shown in Figure 2-1.

## 2.5 Configure Design Initialization Data and Memories (Ask a Question)

The Configure Design Initialization Data and Memories step generates an TCM initialization client and adds it to sNVM, µPROM, or an external SPI flash, based on the type of non-volatile memory selected. In this tutorial, the TCM is initialized from µPROM.

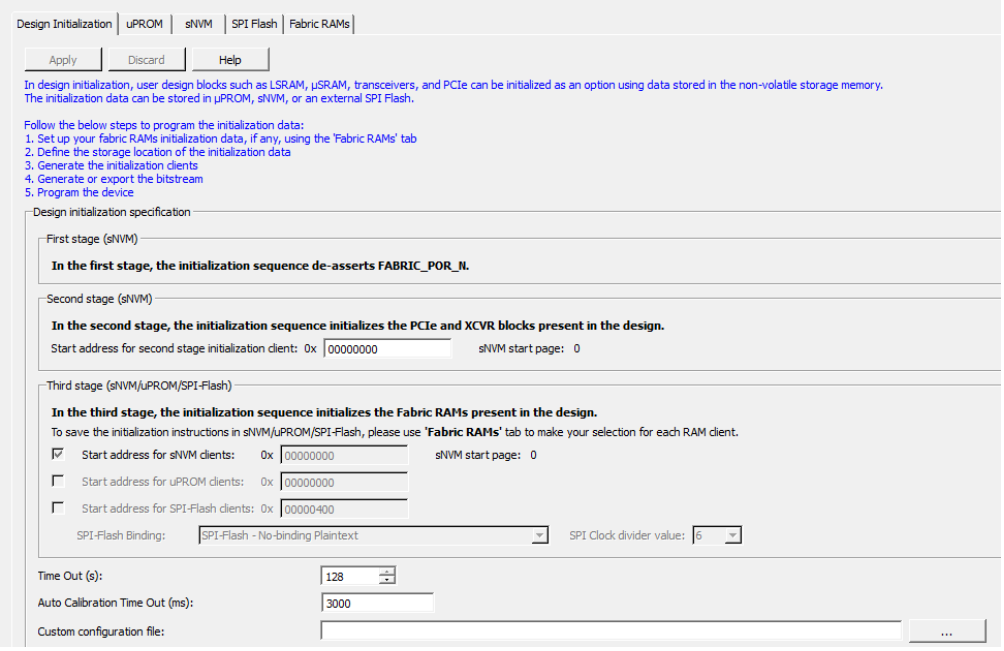
This process requires the user application executable file (hex file) as input to initialize the TCM after device power-up. The hex file is provided with the design files.

To select the non-volatile memory and generate the initialization client, perform the following steps:

1. On the **Design Flow** tab, double-click **Configure Design Initialization Data and Memories**. The **Design and Memory Initialization** window opens.
2. Under **Third stage (uPROM/sNVM/SPI-Flash)**, select **µPROM**, as shown in the following figure. In the Third Stage pane, select uPROM as the non-volatile memory, and retain the default start address (0x00000000).

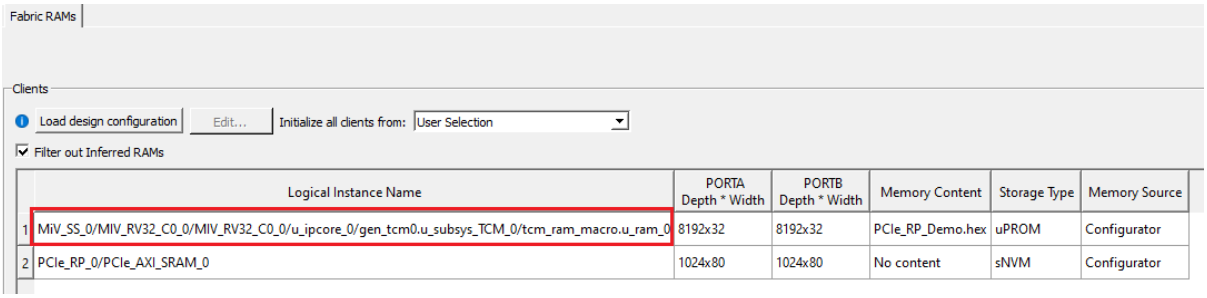
> **Important:** The default start address 0x00000000 is retained because there are no other initialization clients specified in µPROM.

**Figure 2-4.** Design and Memory Initialization Window



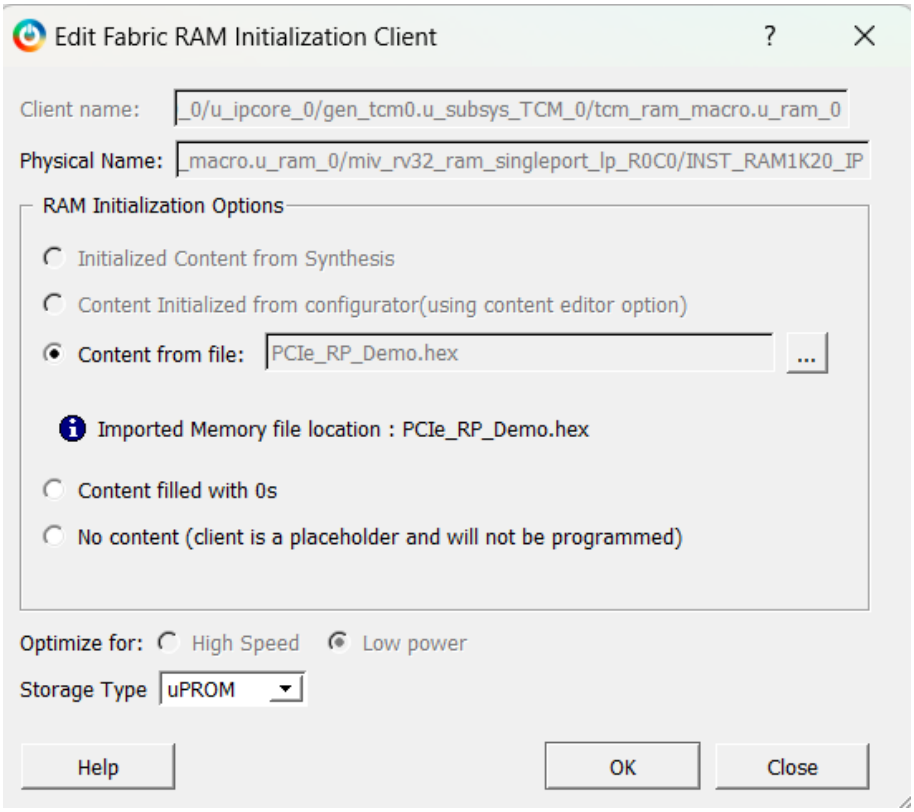

3. On the **Fabric RAMs** tab, select `MiV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0` from the list of logical instances, and click **Edit**, as shown in the following figure. The `MiV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0` instance is the Mi-V processor's main memory. The System Controller initializes this instance with the imported client at power-up.

**Figure 2-5.** Fabric RAMs Tab




4. In the **Edit Fabric RAM Initialization Client** dialog box, click the **Import** button next to **Content from file**, as shown in the following figure.

**Figure 2-6.** Edit Fabric RAM Initialization Client Dialog Box

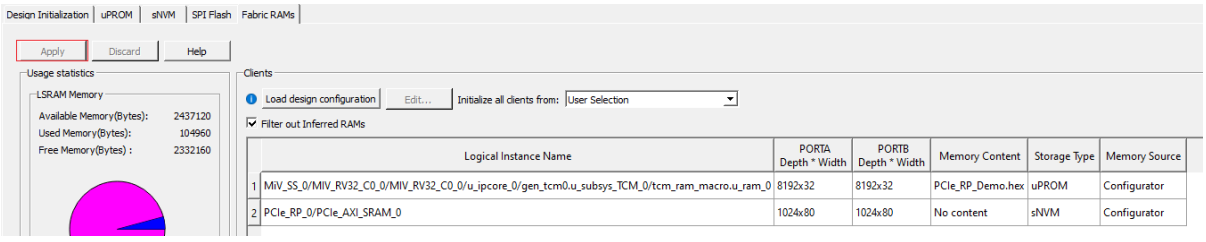



In the following dialog box, browse the `mpf_an4664_df/HW/src/softconsole/PCIe_RP_Demo.hex` file and double-click it.

**Important:** If any changes are applied to the Mi-V application code, rebuild the SoftConsole project in the release mode.

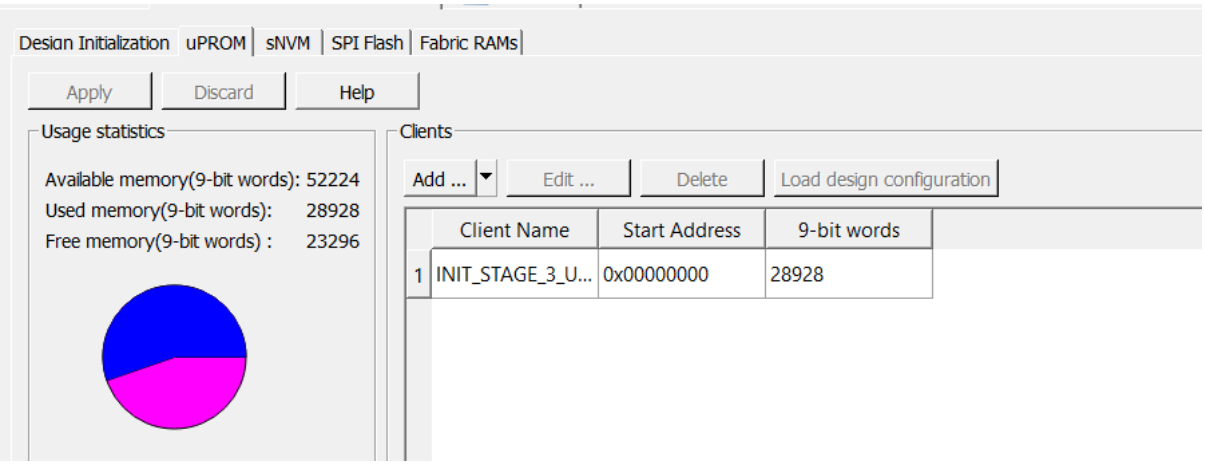
5. In the **Edit Fabric RAM Initialization Client** window, click **OK**.
6. On the **Fabric RAMs** tab, click **Apply**, as shown in the following figure.

**Figure 2-7.** Applying Fabric RAM Content




7. The initialization client for `MiV_SS_0/MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram_macro.u_ram_0` instance is generated and stored in µPROM. This step must be verified by viewing the third stage client created in the µPROM tab, as shown in the following figure.

**Figure 2-8.** Third Stage INIT Client




By default, the first and second stage clients are generated and stored in sNVM.

## 2.6 Generate Bitstream (Ask a Question)

To generate the bitstream, perform the following steps:

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

## 2.7 Run PROGRAM Action (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 following Jumper Settings are set on the board, which will be used as the Root Port device.

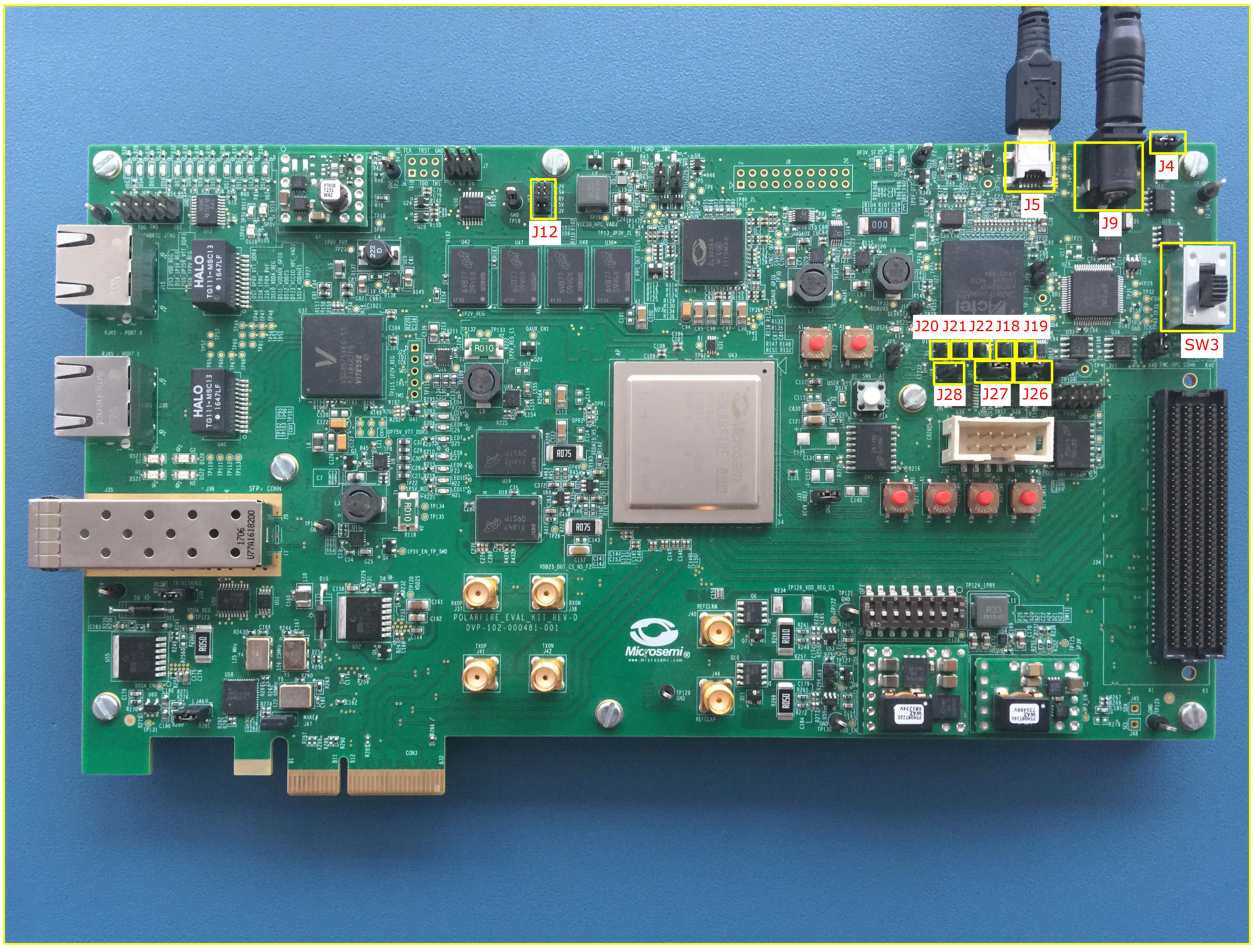
**Table 2-2.** Jumper Settings

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

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.
5. On the Libero in the **Design Flow** tab, double click **Run PROGRAM Action**.

When the device is programmed successfully, a green tick mark appears as shown Figure 2-1. The device is successfully programmed, see Setting Up the Demo.

**Figure 2-9.** Board Setup—Evaluation Kit



# 3. Setting Up the Demo (Ask a Question)

To set up the demo, perform the following steps.

1. Programming the PolarFire devices on the two evaluation boards
2. Connecting the two PolarFire Evaluation boards though the PCIe Adapter card

Throughout this chapter, the two boards are referred using the following labels for simplicity:

- Board A—board running the Root Port design
- Board B—board running the Endpoint design

## 3.1 Connecting the Two Boards (Ask a Question)

This section describes how to connect the two boards though the Microchip PCIe Adapter Card.

To connect the boards, perform the following steps:

1. Ensure that the pins 1 and 2 of the J1 jumper on the PCIe adapter card are closed.
2. Ensure that the pins 1 and 3 of the J2 jumper on the PCIe adapter card are open.
3. Connect CON1 of the adapter card to CON3 (PCIe slot) of Board A.
4. Connect CON2 of the adapter card to CON3 (PCIe slot) of Board B.
5. Connect the USB cable from the Host PC to J5 (FTDI port) on Board A.
6. Connect the USB cable from the Host PC to J5 (FTDI port) on Board B.
7. Connect the power supply cable to the J3 connector of the PCIe adapter card.
8. Power-on Board A and B using the SW3 slide switch.
9. Power-up the PCIe adapter card using the SW1 slide switch.

Board A and Board B power-up using the PCIe adapter card. The following figure shows a demo setup after the two boards are successfully connected.

**Figure 3-1.** Demo Setup

# 4. Running the Demo (Ask a Question)

This section describes how to install and use the GUI to run the PCIe Root Port demo. This section is divided into the following subsections:

- Installing the GUI
- Viewing the Enumeration Data
- Running the Control Plane Commands
- Running the Data Plane Commands

## 4.1 Installation of the GUI (Ask a Question)

To install the GUI, perform the following steps:

1. Extract the contents of the `mpf_an4664_df.zip` file. From the `mpf_an4664_df\GUI_Installer` folder, double-click the `setup.exe` file.
2. Follow the instructions displayed on the installation wizard.

After successful installation, PCIe_Root_Port_GUI appears on the **Start** menu of the host PC desktop. If the GUI application is not yet installed, see Prerequisites section.

## 4.2 Viewing the Enumeration Data (Ask a Question)

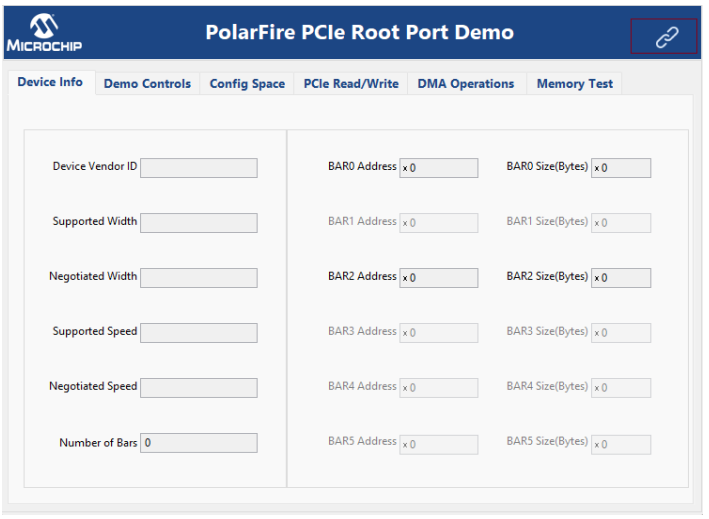
Before you begin, ensure that:

1. The PolarFire FPGA on one board is programmed with the PCIe Root Port design and the PolarFire FPGA on the other board is programmed with the PCIe Endpoint design.
2. The two boards are connected through a Microchip PCIe adapter card and powered-up.
3. LEDs 9, 10, and 11 glow on the Root Port board. This indicates that the PCIe link is up. Otherwise, power-cycle the boards again.

To start the GUI and view the enumeration data, perform the following steps:

1. From the task bar, click the Start button and select PCIe_Root_Port_GUI.
2. Click **Connect** to connect the GUI to the Root Port board, as shown in the following figure.

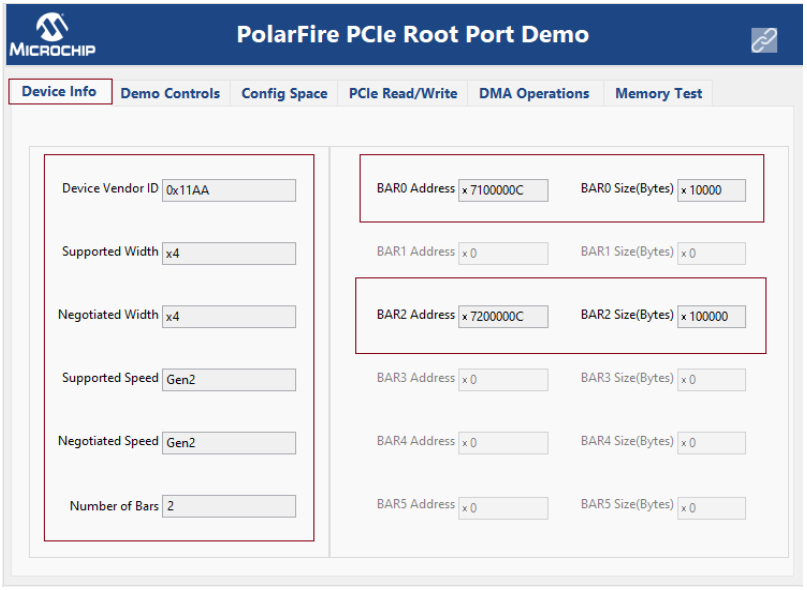
**Figure 4-1.** PCIe Root Port GUI




The GUI starts detecting the UART COM Port of the Root Port device.

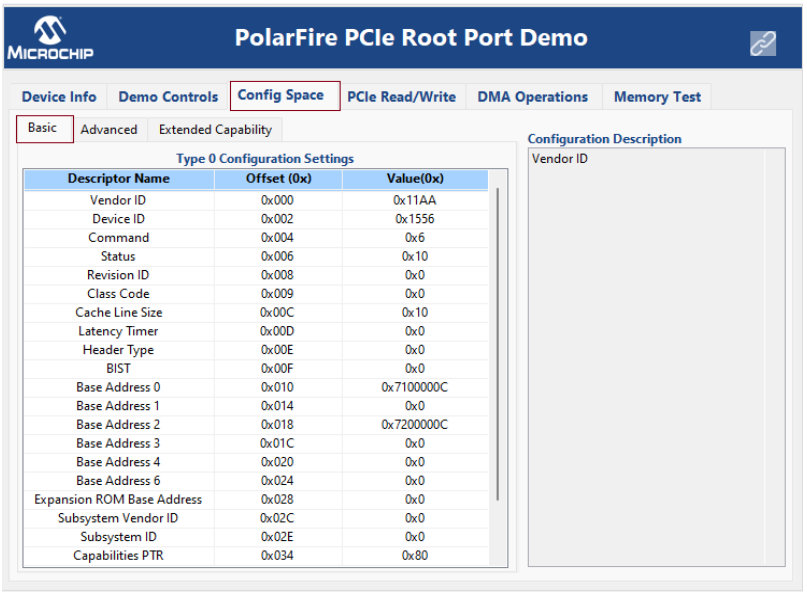
3. After successfully connecting to the COM port, the Mi-V soft processor enumerates the PCIe EP device and sends the configuration space data to the GUI.
4. Click **Device Info** tab to view the Endpoint device information. See the following figure.

**Figure 4-2.** Endpoint Device Information



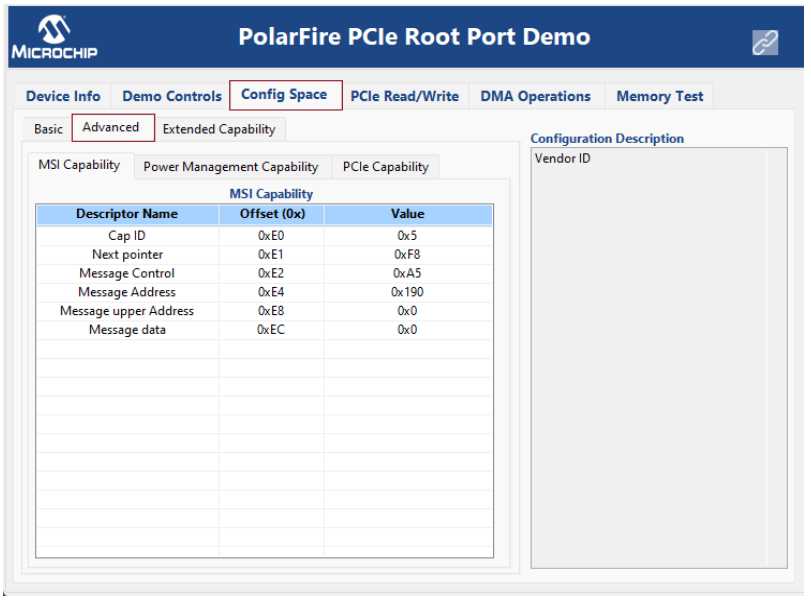

5. Click **Config Space** tab to view the basic Type 0 Configuration Settings of the Endpoint.

**Figure 4-3.** Endpoint Config Space—Basic




6. Click **Advanced** tab to view the MSI Capabilities of the Endpoint, as shown in the following figure.

**Figure 4-4.** Endpoint Config Space—Advanced




7. Similarly, click **Power Management Capability** and **PCIe Capability** tabs to view the relevant data.

## 4.3 Running the Control Plane Commands (Ask a Question)

In this demo, the Root Port initiates the following control plane operations:

- Control Endpoint LEDs
- Read DIP SW Status
- Read MSI count values
- BAR2 Memory read/write commands

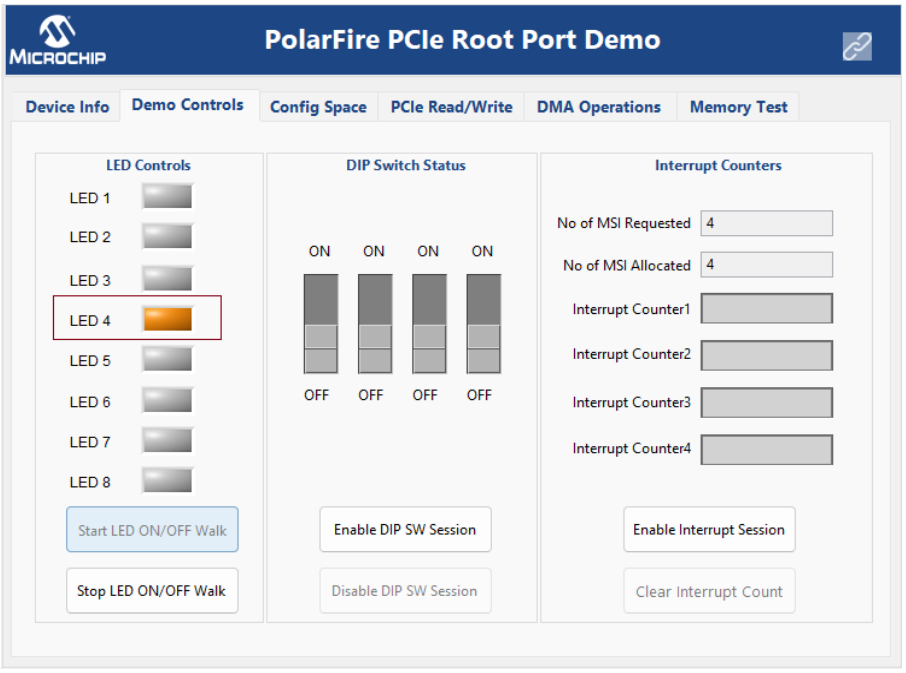
### 4.3.1 Controlling Endpoint LEDs (Ask a Question)

Root Port can initiate the Endpoint LED glowing and walk through.

To issue LED Commands, perform the following steps:

1. Click **Demo Controls** tab.
2. Select any single LED. For example, select LED3, as shown in the following figure.

**Figure 4-5.** Single LED Control

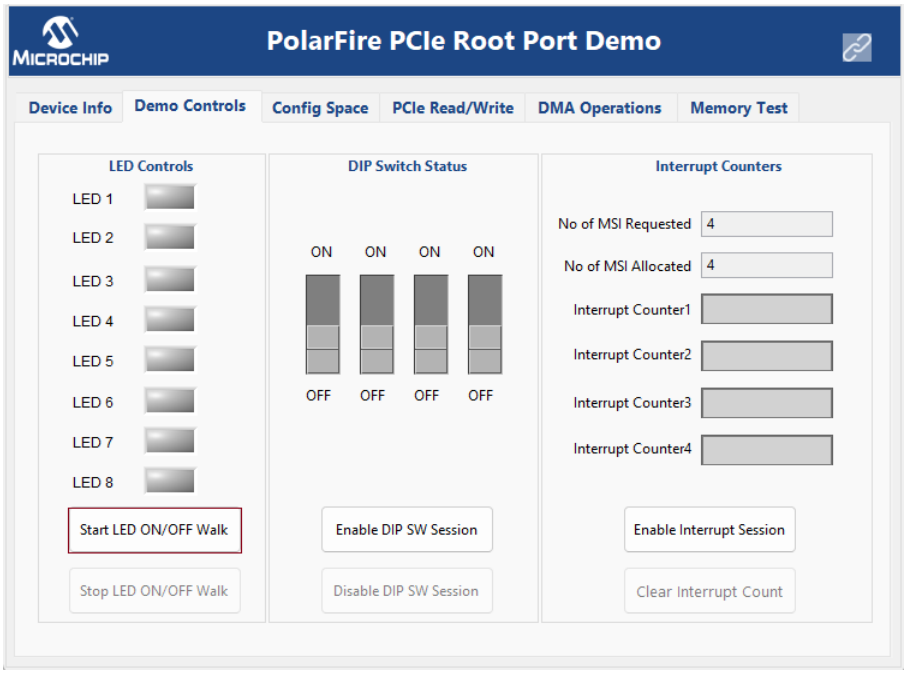



The GUI initiates the LED glow request to the RISC-V processor, which passes this request to the PF_PCIE_0 block. PF_PCIE_0 sends the BAR2 MWr packet to the Endpoint.

As a result, LED3 on the Endpoint board glows.

3. Click **Start LED ON/OFF Walk** button.

**Figure 4-6.** LED ON/OFF Walk

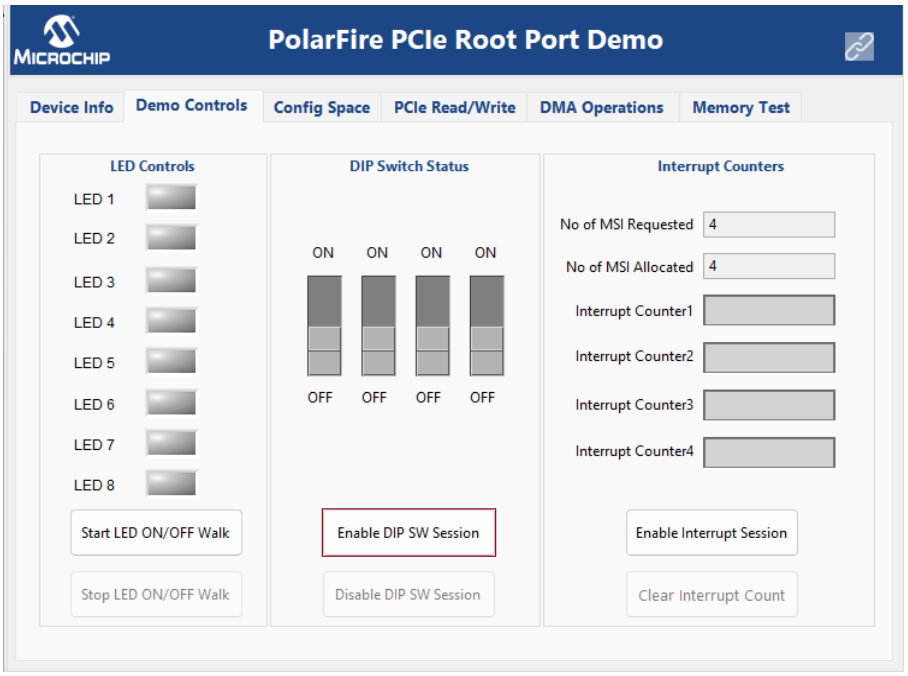



The GUI initiates the LED ON/OFF walk request. As a result, LED ON/OFF is executed from the first to the last LED and in the reverse order.

### 4.3.2 Reading Endpoint DIP SW Status (Ask a Question)

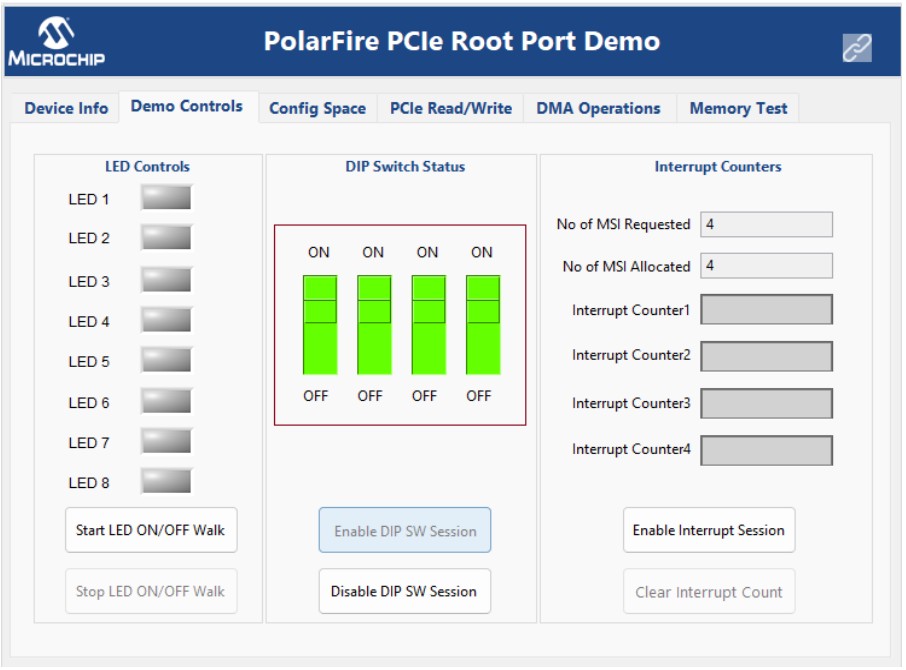
To read the DIP SW Status, click **Enable DIP SW Session** button, as shown in the following figure.

**Figure 4-7.** DIP SW Status Option




The GUI initiates the DIP switch status read request. As a result, the DIP SW status on the Endpoint board is displayed, as shown in the following figure. Change the DIP SW positions on the Endpoint board and observe the same in GUI.

**Figure 4-8.** Endpoint DIP SW Status



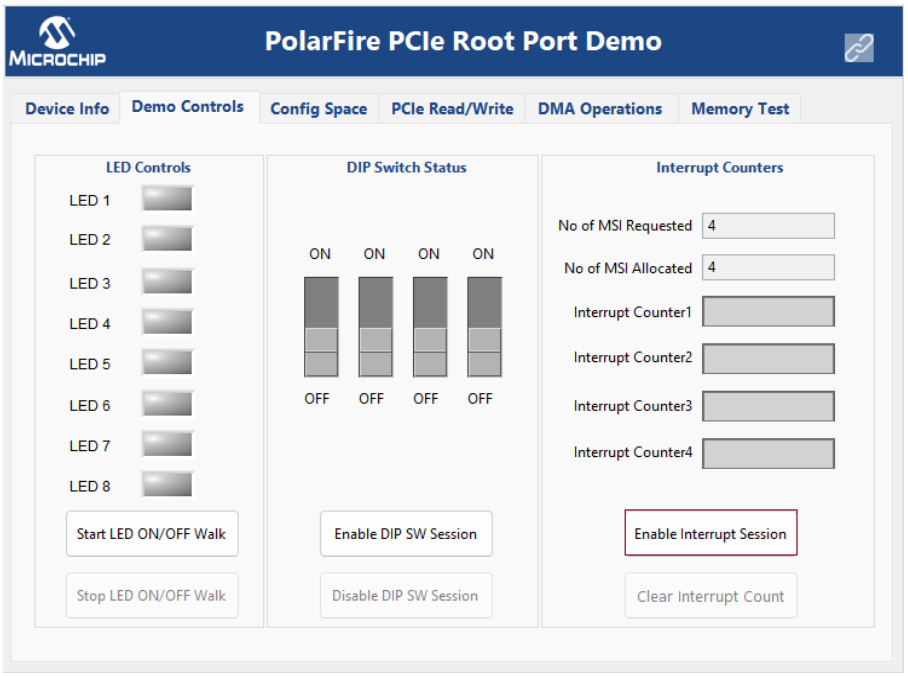
### 4.3.3 Reading MSI Count Values (Ask a Question)

In the demo, Root Port can read the MSI count values for push-button interrupts on the Endpoint board.

To read the MSI count values:

1. Click **Enable Interrupt Session** button, as shown in the following figure.

**Figure 4-9.** Enable Interrupt Session Option



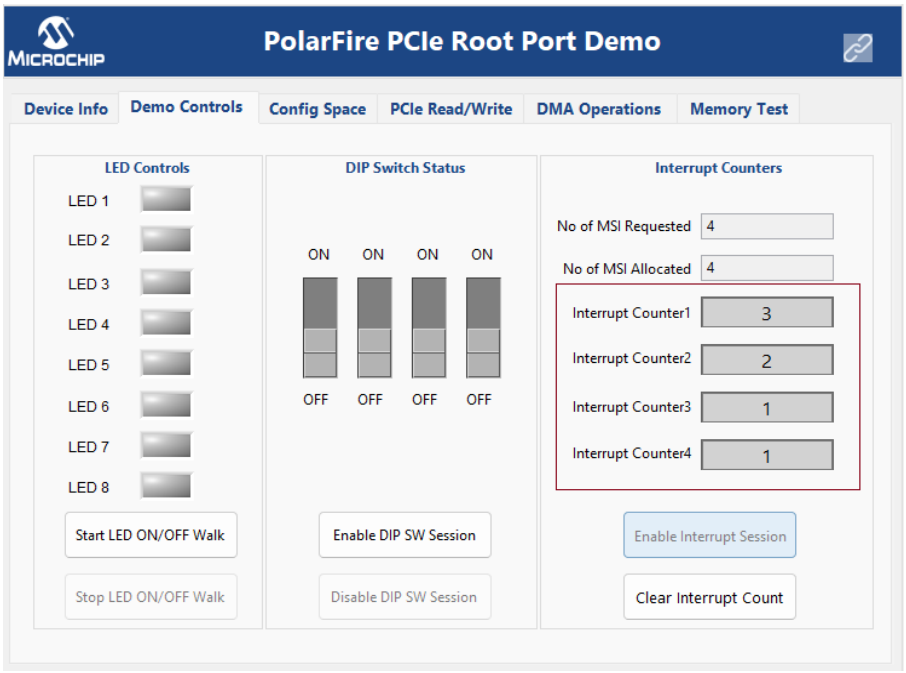

When the Interrupt session is enabled, the GUI sends the Enable Interrupt session request to the RISC-V processor. PF_PCIE_0 receives the number of MSI requested by the Endpoint. The following table lists the four types of MSI allocated by Root port in the reference design.

**Table 4-1.** Allocated MSIs

| MSI Number | Interrupt Type on the Endpoint Board | Mapped Interrupt Counter on the GUI |
|---|---|---|
| 1 | SW10 | Interrupt Counter1 |
| 2 | SW9 | Interrupt Counter2 |
| 3 | SW8 | Interrupt Counter3 |
| 4 | SW7 | Interrupt Counter4 |

2. Press switch and observe interrupt count.

**Figure 4-10.** Interrupt Counter4




3. Click **Clear Interrupt Count** button to clear all of the Interrupt counters on the GUI.

### 4.3.4 Running BAR2 Memory Read/Write Commands (Ask a Question)

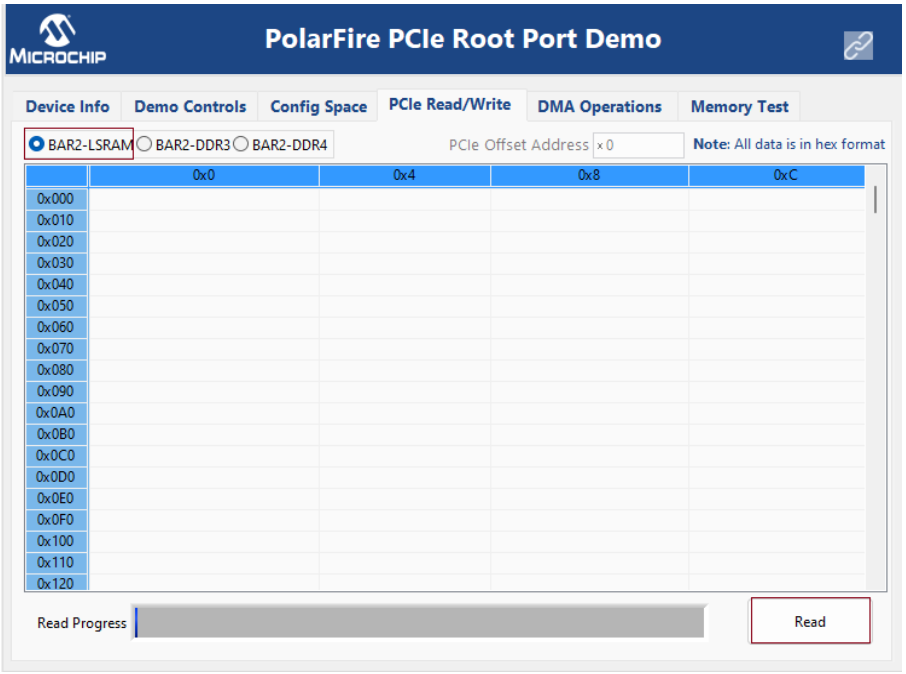
In the demo, the Root port can initiate BAR2 memory read/write commands for reading/writing to Endpoint LSRAM/DDR3/DDR4.

The **PCIe Read/Write** tab on the GUI is used to initiate these commands. The Endpoint LSRAM/DDR3/DDR4 memory is first read, and then a value can be entered in a specific location to initiate the write command.

To run BAR2 read/write, perform the following steps:

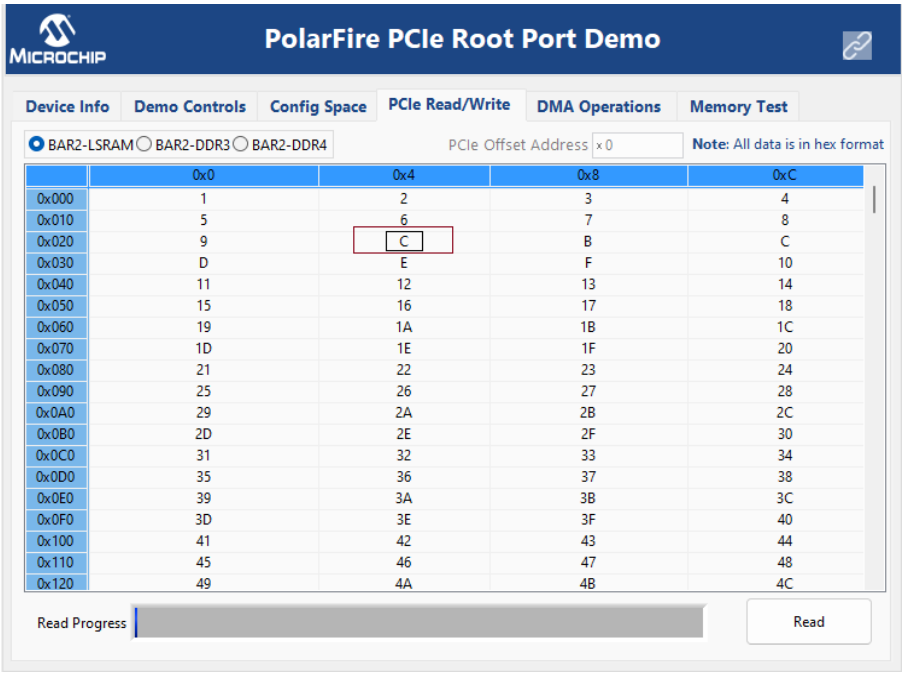
1. Select **BAR2-LSRAM** option and click **Read** button, as shown in the following figure.

**Figure 4-11.** BAR2—LSRAM Read Option



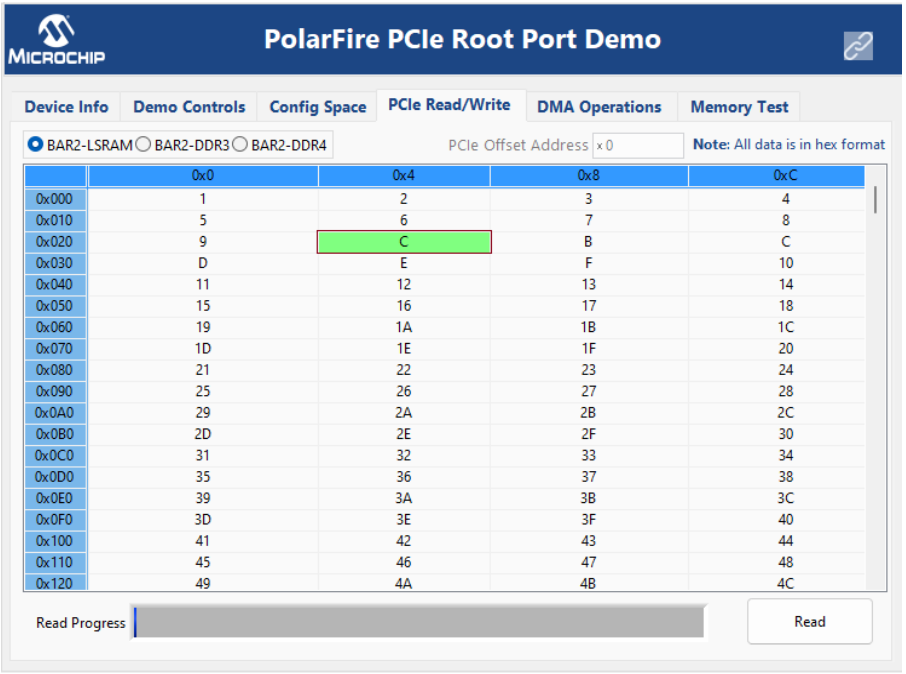

2. Select any memory location and edit the value of that location. For example, see the following figure.

**Figure 4-12.** BAR2—LSRAM Write




3. The edited memory location turns green and the value entered is written to the Endpoint LSRAM memory location, as shown in the following figure.

**Figure 4-13.** BAR2—LSRAM Write Successful




4. Similarly change any other memory location also.
5. Click **Read** button to check whether the memory locations contain the latest values or not.
6. Similarly, run the BAR2-DDR3 and BAR2-DDR4 memory read/write.

## 4.4 Running the Data Plane Commands (Ask a Question)

In the demo, the Root port initiates the Endpoint DMA engines to perform the following data plane commands:

- Running DMA operations
- Running memory test

### 4.4.1 Running DMA Operations (Ask a Question)

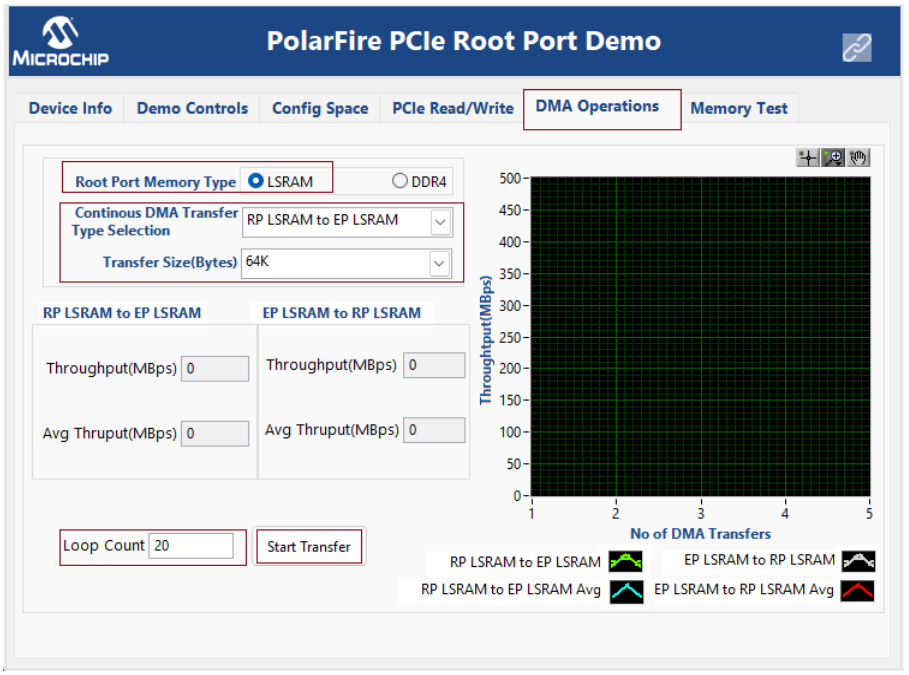
When the Root Port initiates the DMA operation, the Mi-V soft processor activates the Endpoint DMA registers through BAR0. The Endpoint DMA engines can perform the following DMA operations:

- Root Port LSRAM/DDR4 to Endpoint LSRAM\DDR3\DDR4
- Endpoint LSRAM\DDR3\DDR4 to Root Port LSRAM\DDR4

To run the DMA operations, perform the following steps:

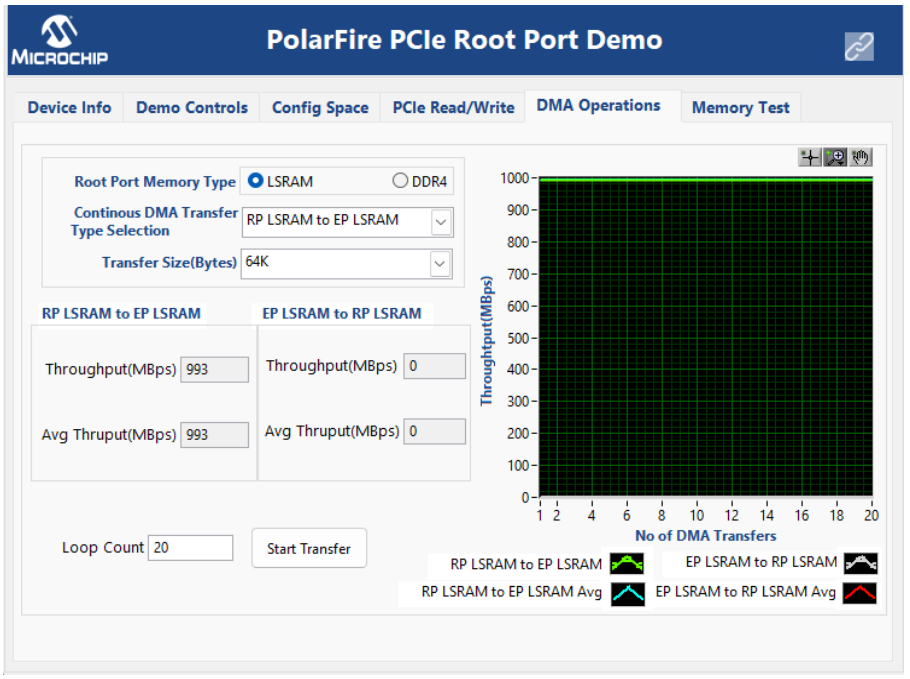
1. Click **DMA Operations** tab, as shown in Figure 4-14.
2. Do the following:
   - Select the **RP LSRAM -> EP LSRAM** from the drop-down list.
   - Select 64K from the Transfer Size (Bytes) drop-down.
   - Set the **Loop Count** to 20 (Transfer Size and Loop Count are common parameters that can be adjusted).
   - Click **Start transfer**.

**Figure 4-14.** Initiating RP LSRAM to EP LSRAM DMA




The GUI displays the corresponding throughput details and graph, as shown in the following figure.

**Figure 4-15.** RP LSRAM to EP LSRAM Throughput




3. Do the following to initiate another DMA transaction:
   - Select **Both RP LSRAM <-> EP LSRAM** from the drop-down list.
   - Select 64K from the **Transfer Size (Bytes)** drop-down.
   - Set the **Loop Count** to 20 (Transfer Size and Loop Count are common parameters that can be adjusted).

- Click **Start Transfer**.

4. Similarly, select the RP LSRAM to EP DDR3 and RP LSRAM to EP DDR4 from the drop-down and observe the throughputs.
5. Select DDR4 as the Root Port Memory Type and perform DMA operations by selecting the Endpoint destination memory type.

### 4.4.2 Running Memory Test (Ask a Question)

The **Memory Test** tab provides the memory test feature, which is also a DMA operation. The **Memory Test** tab enables DMA transactions between Root Port and Endpoint memory type (LSRAM, DDR3, and DDR4). This feature provides data pattern options with which the Root Port memory is initialized and then DMA operation is performed.

In memory testing, the user application performs the following sequence of operations:

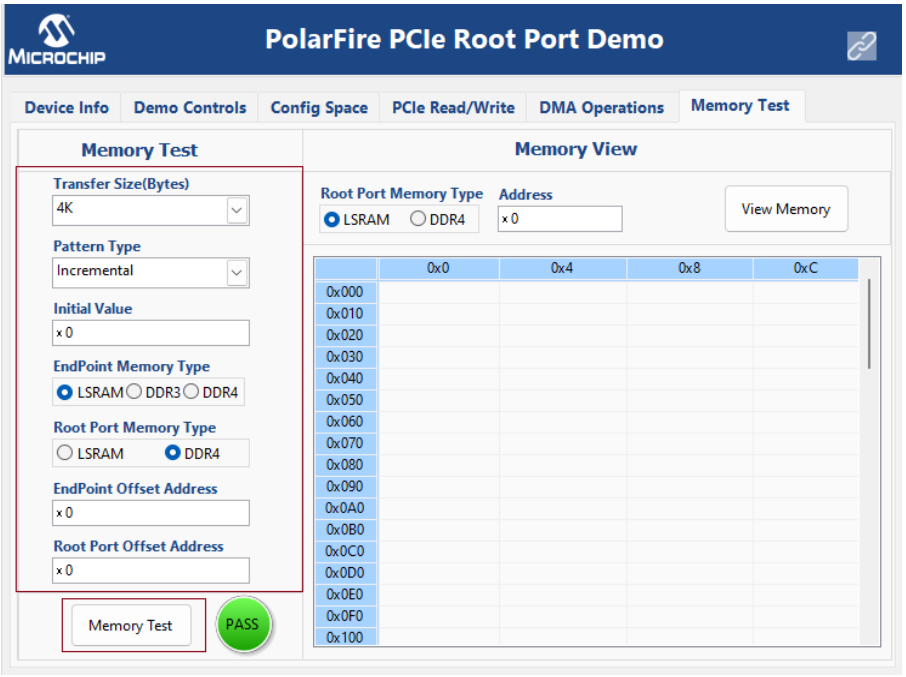
1. Initializes the Root Port memory with the specified data pattern
2. Performs the DMA from Root Port memory to Endpoint memory
3. Erases the data pattern in the Root Port memory
4. Performs the DMA from Endpoint memory to Root Port memory
5. Compares the data in Root Port memory with the selected data pattern

To run the memory test, perform the following steps:

1. Select the DMA parameters like Transfer Size (Bytes), Pattern Type, Endpoint Memory Type, RootPort Memory Type, EndPoint Offset Address, and RootPort Offset Address, as shown in the following figure.

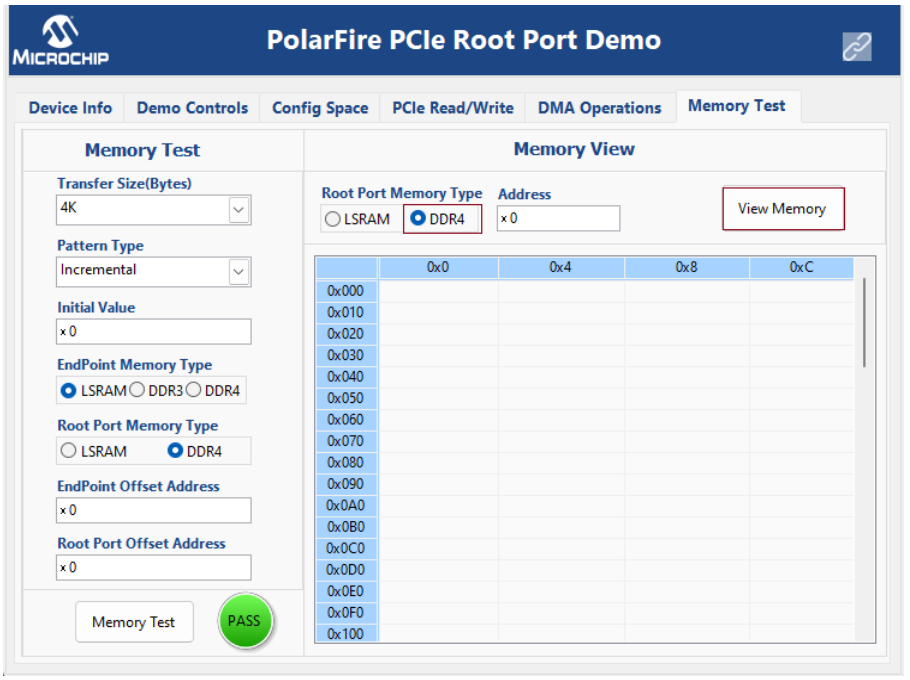
**Important:** The Root Port slave ATR3 is configured for 1 MB. Therefore, the maximum Endpoint offset address is F80000 and the maximum Root Port address is 0x80000.

**Figure 4-16.** Memory Test Feature



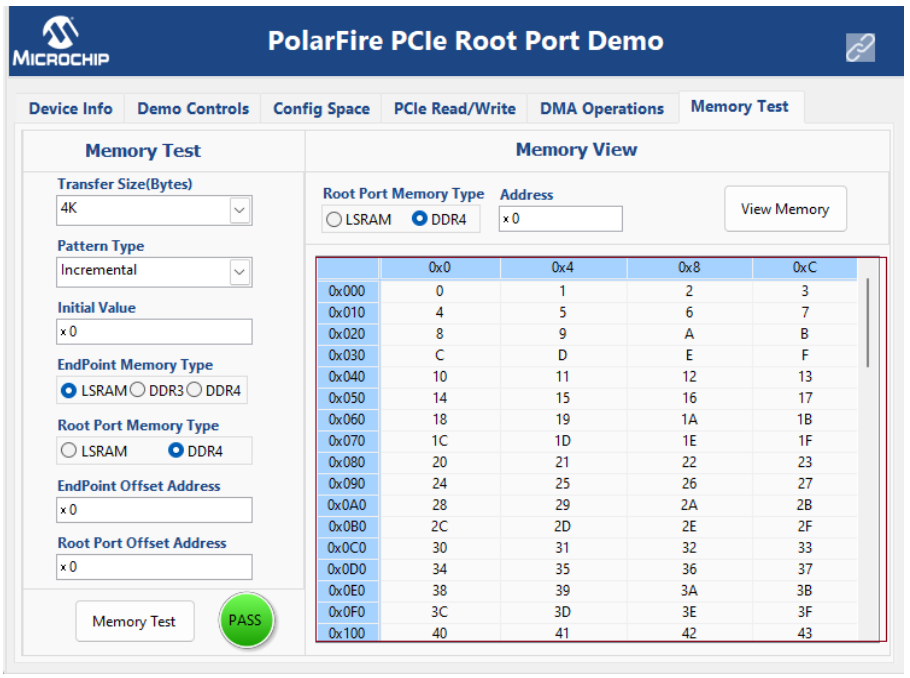
2. Click **Memory Test**.
3. Select option **DDR4** from the Root port memory type and click **View Memory**, as shown in the following figure to read the Root port DDR4.

**Figure 4-17.** The View Memory Option



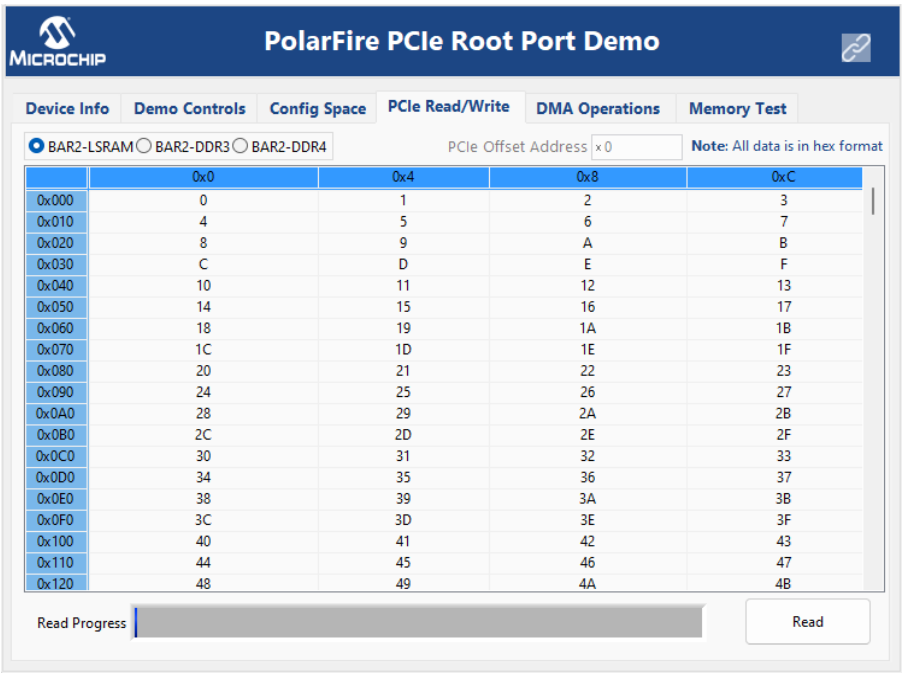

4. The GUI displays the data pattern written to the Root port DDR4, as shown in the following figure.

**Figure 4-18.** Root port DDR4 Memory Content




5. Select the **PCIe Read/Write** tab and click **Read** to view the data pattern written to the Endpoint LSRAM.

**Figure 4-19.** Endpoint LSRAM Memory Content




## 4.5 PolarFire DMA Throughput Summary (Ask a Question)

The following table lists the throughput values observed during the continuous DMA mode.

**Table 4-2.** Throughput Summary

| DMA Transfer Type | DMA Size | Throughput (MBps) | Throughput Average (MBps) |
|---|---|---|---|
| RP LSRAM to EP LSRAM | 512K | 1022 | 1022 |
| EP LSRAM to RP LSRAM | 512K | 779 | 779 |
| RP LSRAM to EP DDR3 | 512K | 773 | 773 |
| EP DDR3 to RP LSRAM | 512K | 328 | 328 |
| RP LSRAM to EP DDR4 | 512K | 998 | 998 |
| EP DDR4 to RP LSRAM | 512K | 391 | 391 |
| RP DDR4 to EP LSRAM | 512K | 540 | 540 |
| EP LSRAM to RP DDR4 | 512K | 779 | 779 |
| RP DDR4 to EP DDR3 | 512K | 540 | 540 |
| EP DDR3 to RP DDR4 | 512K | 328 | 328 |
| RP DDR4 to EP DDR4 | 512K | 540 | 540 |
| EP DDR4 to RP DDR4 | 512K | 391 | 391 |

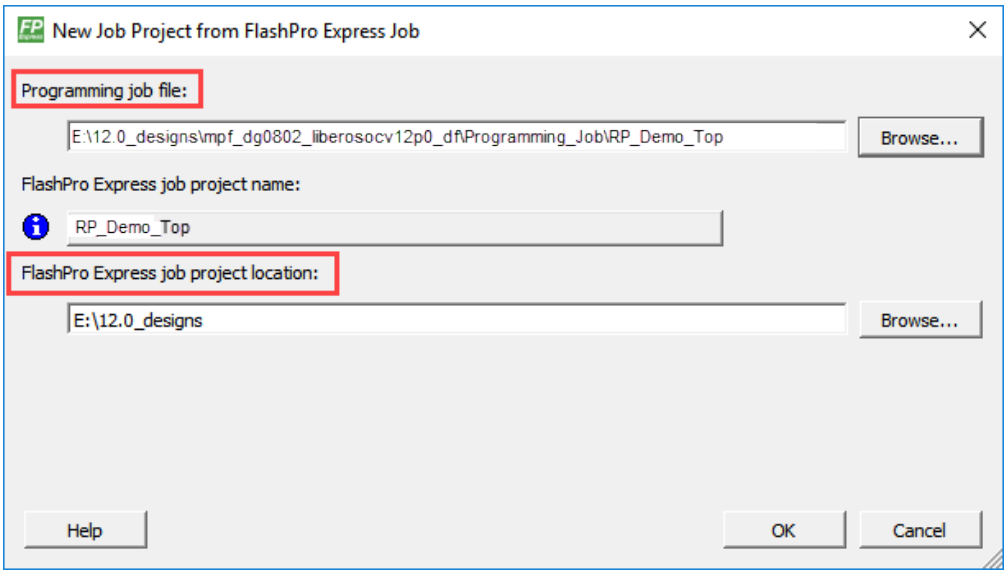
# 5. Appendix 1: Programming the Devices Using FlashPro Express (Ask a Question)

The Root Port design must be programmed on Board A and the Endpoint design must be programmed on Board B.

To program, perform the following steps:

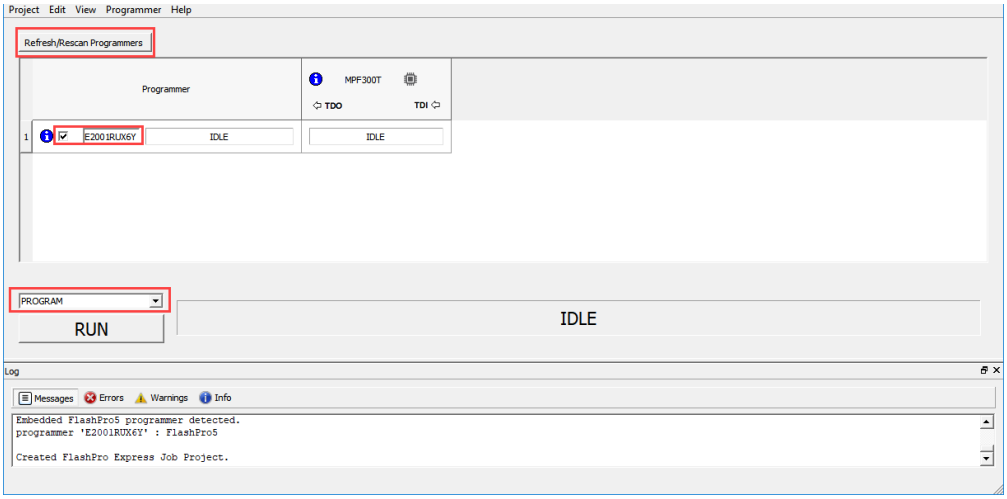
1. Take Board A and ensure that the Jumper Settings are set as listed in Table 2-2.
2. Connect the power supply cable to the J9 connector on Board A.
3. Connect the USB cable from the Host PC to J5 (FTDI port) on Board A.
4. Power-up Board A using the SW3 slide switch.
5. On the host PC, launch the **FlashPro Express** software.
6. To create a new job, click **New** or in the **Project** menu, and 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**, and navigate to the location where the `.job` file is located and select the file.
     The default location is: `<download_folder>\mpf_an4664_df\Programming_files`.
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

**Figure 5-1.** New Job Project from FlashPro Express Job



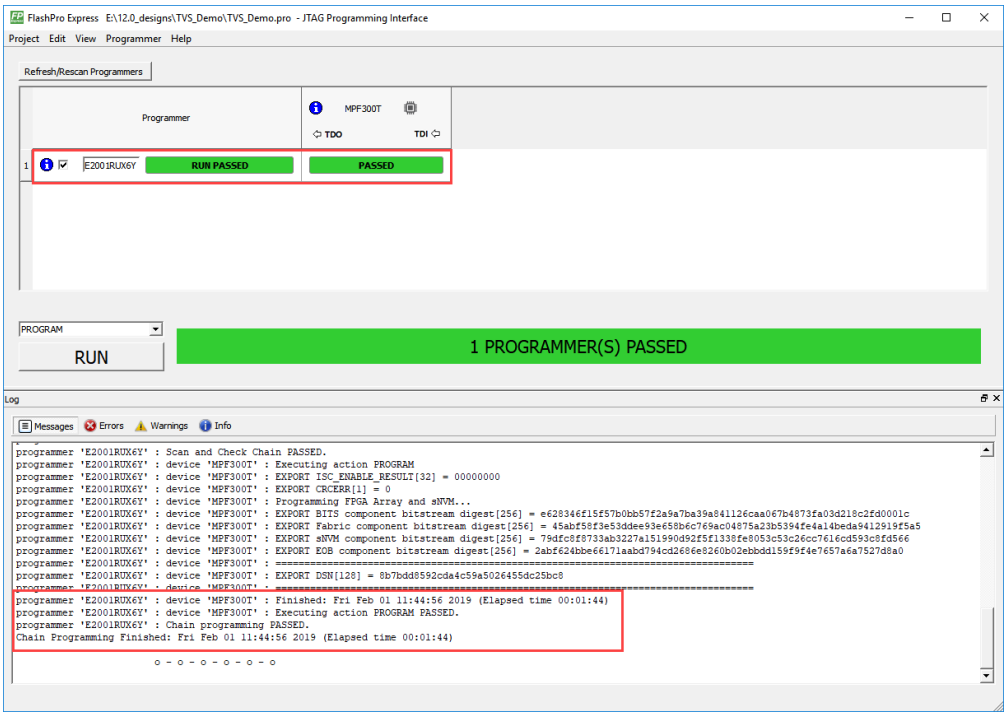

8. Click **OK**. The required programming file is selected and ready to be programmed in the device.
9. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, then confirm the board connections and click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

**Figure 5-2.** Programming the Device




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

**Figure 5-3.** FlashPro Express—RUN PASSED

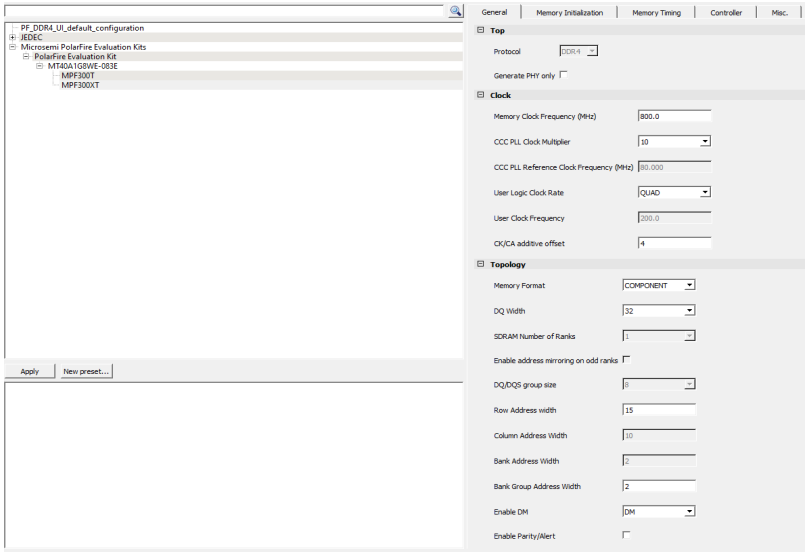



11. Close **FlashPro Express** or in the **Project** tab, click **Exit**.
    Root port design is successfully programmed on Board A.
12. Similarly, program Board B with the Endpoint design. Browse Programming file from `mpf_an4597_df\Programming_files\top_Eval_Kit.job` location.

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

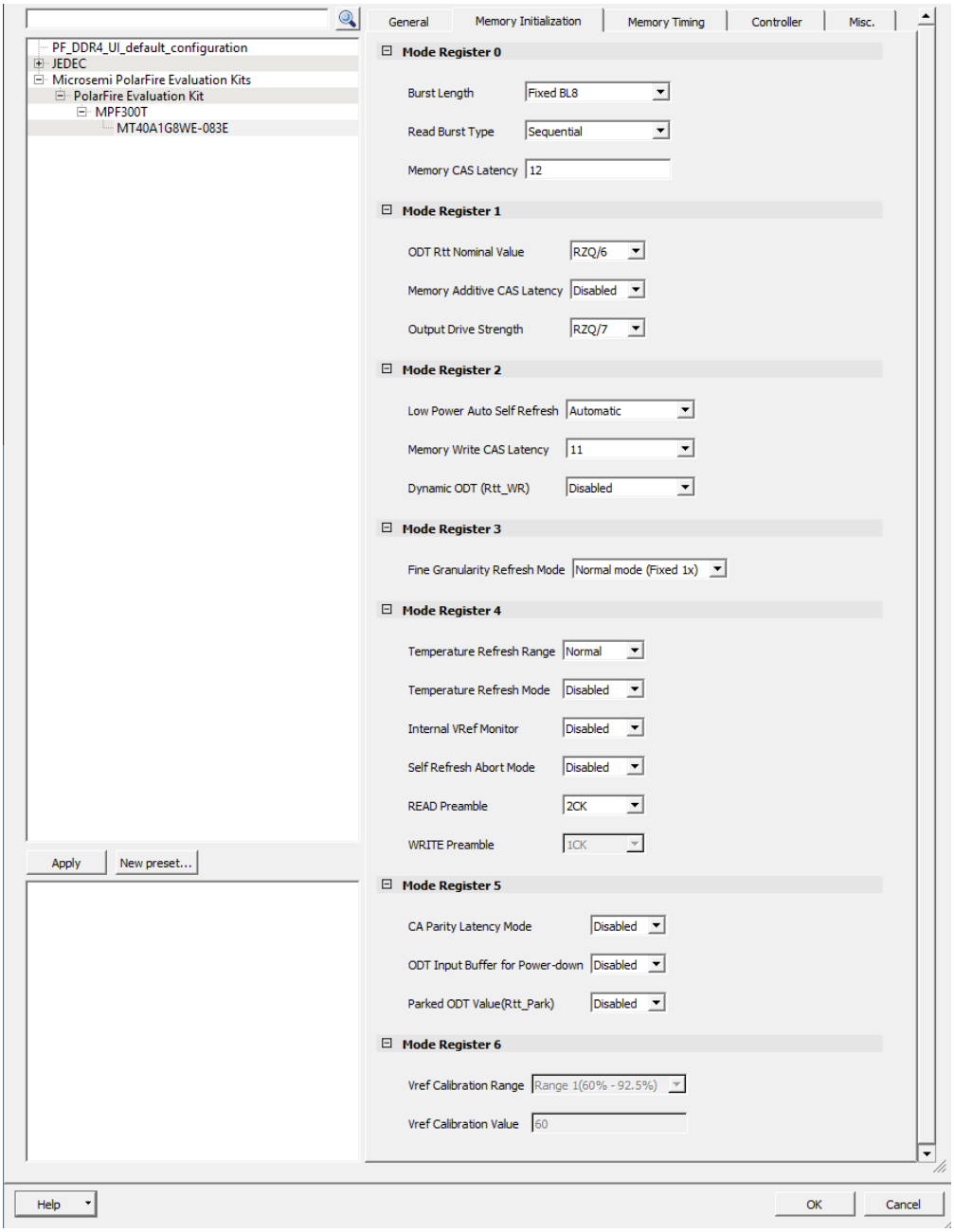
The DDR4 subsystem is configured to access the 32-bit DDR4 memory through an AXI4 64-bit interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4 memory on the PolarFire Evaluation kit. The following figure shows the general configuration settings for the DDR4 memory.

**Figure 6-1.** PF_DDR4 Configurator—General



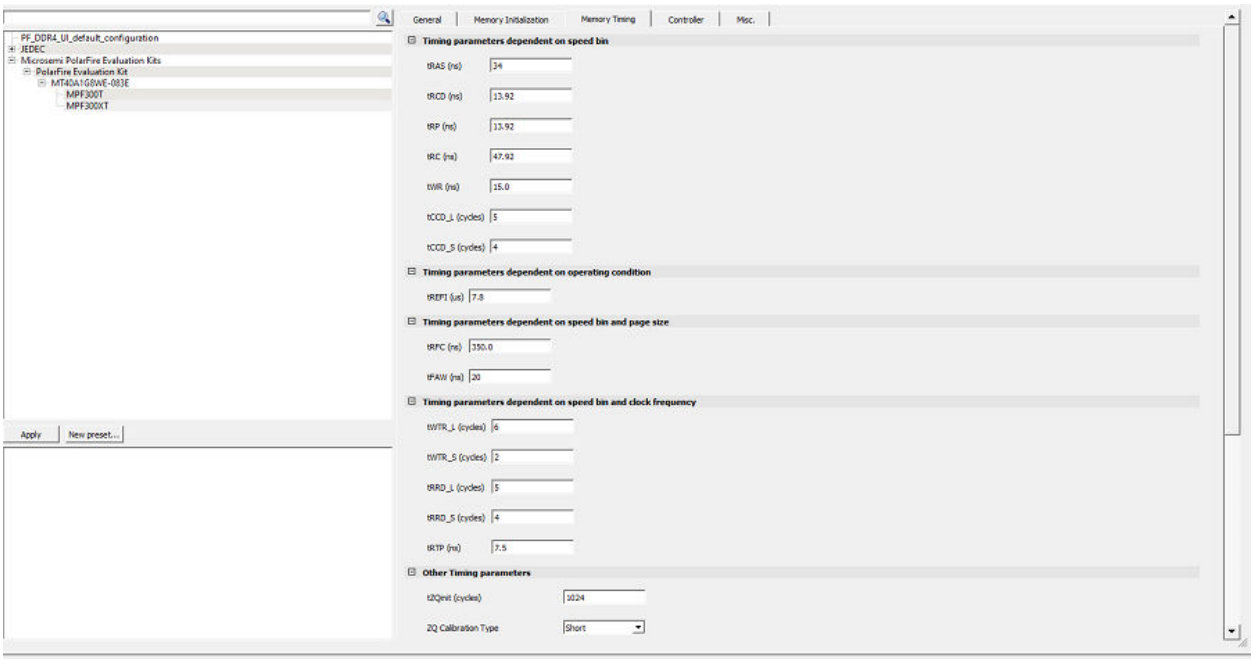

The following figure shows the initialization configuration settings for the DDR4 memory.

**Figure 6-2.** PF_DDR4 Configurator—Memory Initialization



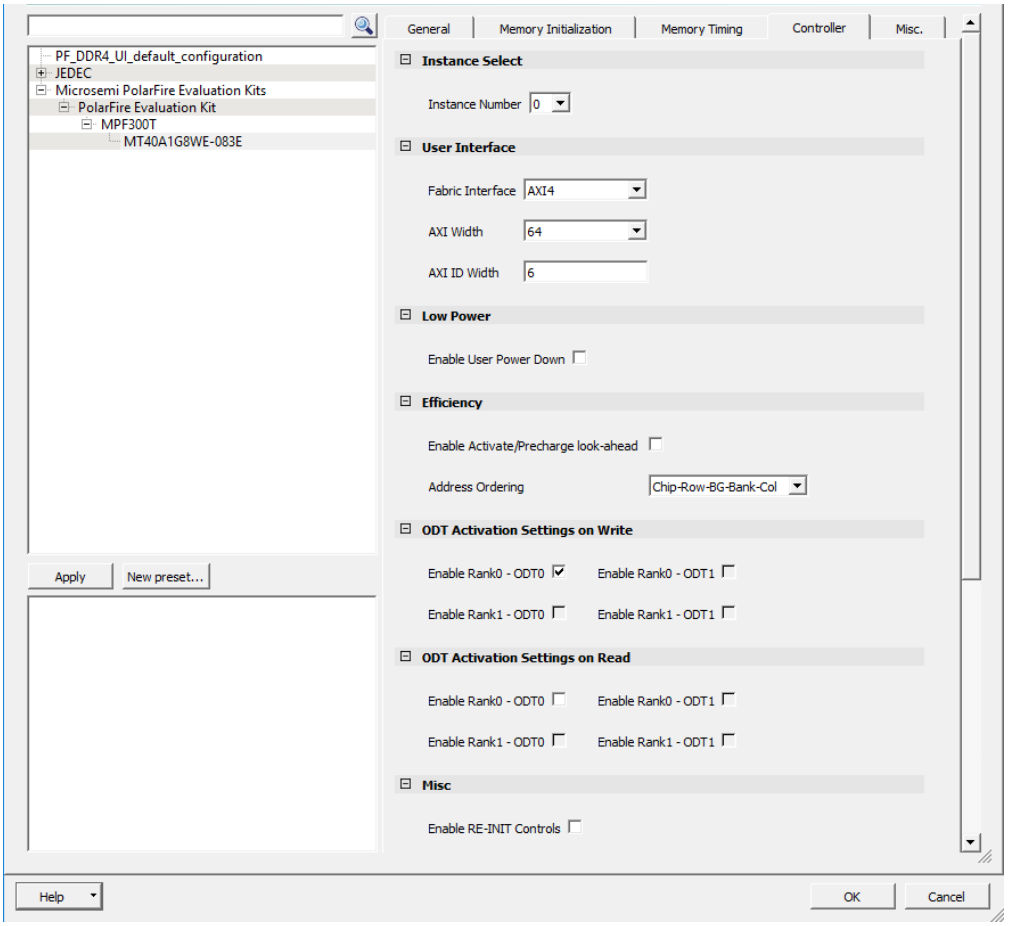

The following figure shows the memory initialization configuration settings for the DDR memory.

**Figure 6-3.** PF_DDR4 Configurator—Memory Timing Initialization




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

**Figure 6-4.** PF_DDR4 Configurator—Controller



# 7. Appendix 3: 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 reproduced 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 `mpf_an4664_df\HW\TCL_Script_readme.txt`. Contact Technical Support for any queries encountered when running the TCL script.

# 8. Appendix 4: 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 DDR3 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 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.

# 9. 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 most current publication.

**Table 9-1.** Revision History

| Revision | Date | Description |
|---|---|---|
| C | 01/2025 | The following is a summary of the changes made in C revision:<br>• Updated the document for Libero® v2024.2.<br>• Updated the .job filepath and TCL script filepath throughout the document.<br>• Added Microchip FPGA_GUI_Pack in Design Requirements section.<br>• Updated Prerequisites section.<br>• Updated Figure 1-2 in the Design Implementation section.<br>• Updated Table 2-1 in the Resource Utilization section.<br>• Updated GUI images in the Running the Demo section.<br>• Updated Figure 6-1 in the Appendix 2: DDR4 Configuration section. |
| B | 06/2024 | The following is a summary of the changes made in B revision:<br>• Updated Figure 1-1.<br>• Updated Table 1-2.<br>• Updated Figure 1-3.<br>• Updated Figure 1-4.<br>• Updated Figure 1-6 and Figure 1-7.<br>• Updated .job file and programming file location in Appendix 1: Programming the Devices Using FlashPro Express.<br>• Updated Tcl scripts file location in Appendix 3: Running the TCL Script. |
| A | 08/2022 | The following is a summary of the changes made in A revision:<br>• The document was migrated to the Microchip template.<br>• The document number was updated to DS00004664A from 50200802.<br>• The document ID was updated to AN4664 from DG0802.<br>• Added a new line in Prerequisites.<br>• Updated Figure 1-1.<br>• Updated Table 1-2.<br>• Updated Figure 1-2.<br>• Updated Figure 1-3.<br>• Updated Figure 1-4.<br>• Updated Figure 1-6 and Figure 1-7.<br>• Replaced GUI installation link in Installation of the GUI.<br>• Updated .job file and programming file location in Appendix 1: Programming the Devices Using FlashPro Express.<br>• Updated Tcl scripts file location in Appendix 3: Running the TCL Script. |

| .....continued | | |
|---|---|---|
| **Revision** | **Date** | **Description** |
| 7.0 | — | The following is a summary of the changes made in this revision:<br>• Updated the document for Libero SoC v2021.2.<br>• Added support for PERSTn signal generation.<br>• Replaced Figure 1-1 through Figure 1-7.<br>• Updated Table 1-2 and Table 2-1.<br>• Updated section Design Implementation.<br>• **AHBtoAXIAPB** subsystem is removed from the MiV subsystem.<br>• **AXI_Interconnect_0** and **Core_APB_0** were added in the MiV subsystem.<br>• Enabled AXI initiator interface and APB initiator interface on MiV core.<br>• Updated the reset structure. |
| 6.0 | — | The following is a summary of the changes made in this revision:<br>• Updated the document for Libero SoC v12.6.<br>• Removed the references to Libero version numbers. |
| 5.0 | — | The document was updated for Libero SoC v12.0. |
| 4.0 | — | The document was updated for Libero SoC PolarFire v2.3. |
| 3.0 | — | The document was updated for Libero SoC PolarFire v2.2. |
| 2.0 | — | The document was updated for Libero SoC PolarFire v2.1. |
| 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: 979-8-3371-0416-4

## 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.