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

About Microsemi

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

1 Revision History ............................................................................................................. 1
  1.1 Revision 4.0 ............................................................................................................. 1
  1.2 Revision 3.0 ............................................................................................................. 1
  1.3 Revision 2.0 ............................................................................................................. 1
  1.4 Revision 1.0 ............................................................................................................. 1

2 PolarFire FPGA PCIe Root Port ..................................................................................... 2
  2.1 Design Requirements ............................................................................................. 2
  2.2 Prerequisites .......................................................................................................... 3
  2.3 Demo Design ......................................................................................................... 3
    2.3.1 Memory and Peripheral Address Map ............................................................... 4
    2.3.2 Design Implementation ................................................................................. 5
  2.4 Clocking Structure ................................................................................................. 8
  2.5 Reset Structure ..................................................................................................... 9

3 Libero Design Flow ..................................................................................................... 10
  3.1 Synthesize ............................................................................................................. 10
  3.2 Place and Route .................................................................................................... 11
    3.2.1 Resource Utilization ...................................................................................... 11
  3.3 Verify Timing ....................................................................................................... 12
  3.4 Generate FPGA Array Data ................................................................................. 12
  3.5 Configure Design Initialization Data and Memories ............................................ 12
  3.6 Generate Bitstream .............................................................................................. 15
  3.7 Run PROGRAM Action ......................................................................................... 15

4 Setting Up the Demo .................................................................................................. 16
  4.1 Programming the PolarFire Devices .................................................................... 16
  4.2 Connecting the Two Boards ................................................................................ 17

5 Running the Demo ..................................................................................................... 18
  5.1 Installing the GUI .................................................................................................. 18
  5.2 Viewing the Enumeration Data ............................................................................ 18
  5.3 Running the Control Plane Commands ............................................................... 20
    5.3.1 Controlling Endpoint LEDs ........................................................................... 20
    5.3.2 Reading Endpoint DIP SW Status ................................................................ 21
    5.3.3 Reading MSI Count Values .......................................................................... 22
    5.3.4 Running BAR2 Memory Read/Write Commands .......................................... 23
  5.4 Running the Data Plane Commands ..................................................................... 25
    5.4.1 Running DMA Operations ............................................................................ 25
    5.4.2 Running Memory Test ................................................................................... 26
  5.5 PolarFire DMA Throughput Summary .................................................................. 29

6 Appendix: DDR4 Configuration ..................................................................................... 30

7 Appendix: References .................................................................................................. 33
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Block Diagram</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>PCIe Root Port demo design</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3</td>
<td>MIV_SS_0 SmartDesign</td>
<td>5</td>
</tr>
<tr>
<td>Figure 4</td>
<td>AHBtoAXIAPB_0 SmartDesign</td>
<td>6</td>
</tr>
<tr>
<td>Figure 5</td>
<td>PCIe_RP SmartDesign</td>
<td>7</td>
</tr>
<tr>
<td>Figure 6</td>
<td>PCIe_TL_CLK_0 SmartDesign</td>
<td>7</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Clocking Structure</td>
<td>8</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Reset Structure</td>
<td>9</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Libero Design Flow Options</td>
<td>10</td>
</tr>
<tr>
<td>Figure 10</td>
<td>I/O Editor—XCVR View</td>
<td>11</td>
</tr>
<tr>
<td>Figure 11</td>
<td>PF_DDR3_SubSystem_0 Placement</td>
<td>11</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Design and Memory Initialization Window</td>
<td>12</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Fabric RAMs Tab</td>
<td>13</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Edit Fabric RAM Initialization Client Dialog Box</td>
<td>13</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Applying Fabric RAM Content</td>
<td>14</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Third Stage INIT Client</td>
<td>14</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Program Action Completed</td>
<td>16</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Demo Setup</td>
<td>17</td>
</tr>
<tr>
<td>Figure 19</td>
<td>PCIe Root Port GUI</td>
<td>18</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Endpoint Device Information</td>
<td>19</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Endpoint Config Space-Basic</td>
<td>19</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Endpoint Config Space-Advanced</td>
<td>20</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Single LED Control</td>
<td>20</td>
</tr>
<tr>
<td>Figure 24</td>
<td>LED ON/OFF Walk</td>
<td>21</td>
</tr>
<tr>
<td>Figure 25</td>
<td>DIP SW Status Option</td>
<td>21</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Endpoint DIP SW Status</td>
<td>22</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Enable Interrupt Session Option</td>
<td>22</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Interrupt Counter4</td>
<td>23</td>
</tr>
<tr>
<td>Figure 29</td>
<td>BAR2-LSRAM Read Option</td>
<td>23</td>
</tr>
<tr>
<td>Figure 30</td>
<td>BAR2-LSRAM Write</td>
<td>24</td>
</tr>
<tr>
<td>Figure 31</td>
<td>BAR2-LSRAM Write Successful</td>
<td>24</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Initiating RP LSRAM to EP LSRAM DMA</td>
<td>25</td>
</tr>
<tr>
<td>Figure 33</td>
<td>RP LSRAM to EP LSRAM Throughput</td>
<td>26</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Memory Test Feature</td>
<td>27</td>
</tr>
<tr>
<td>Figure 35</td>
<td>The View Memory Option</td>
<td>27</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Root port DDR4 Memory Content</td>
<td>28</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Endpoint LSRAM Memory Content</td>
<td>28</td>
</tr>
<tr>
<td>Figure 38</td>
<td>PF_DDR4 Configurator—General</td>
<td>30</td>
</tr>
<tr>
<td>Figure 39</td>
<td>PF_DDR4 Configurator—Memory Initialization</td>
<td>31</td>
</tr>
<tr>
<td>Figure 40</td>
<td>PF_DDR4 Configurator—Controller</td>
<td>32</td>
</tr>
</tbody>
</table>
Tables

Table 1  Design Requirements ................................................................. 2
Table 2  Mi-V and PF_PCIE Address Maps .............................................. 4
Table 3  Resource Utilization ................................................................. 11
Table 4  Jumper Settings ........................................................................... 15
Table 5  Allocated MSIs ........................................................................... 23
Table 6  Throughput Summary ................................................................. 29
1 Revision History

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

1.1 Revision 4.0
The document was updated for Libero SoC PolarFire v2.3.

1.2 Revision 3.0
The document was updated for Libero SoC PolarFire v2.2.

1.3 Revision 2.0
The document was updated for Libero SoC PolarFire v2.1.

1.4 Revision 1.0
The first publication of this document.
2 PolarFire FPGA PCIe Root Port

Microsemi 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® System-on-Chip (SoC) PolarFire 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 UG0685: PolarFire FPGA PCI Express User Guide.

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 PolarFire, see Demo Design, page 3.

The demo design can be programmed using any of the following options:

- Using the stp file: To program the device using the stp file provided along with the design files, see Setting Up the Demo, page 16.
- Using Libero SoC PolarFire: To program the device using Libero SoC PolarFire, see Libero Design Flow, page 10. Use this option when the demo design is modified.

To run the demo:

- 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, page 16.

2.1 Design Requirements

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

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>64-bit Windows 7 or 10</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>Two PolarFire Evaluation Kits (ES)</td>
<td>Rev C or later</td>
</tr>
<tr>
<td>Each kit contains:</td>
<td></td>
</tr>
<tr>
<td>– PolarFire Evaluation Board</td>
<td></td>
</tr>
<tr>
<td>– 12 V, 5 A AC power adapter and cord</td>
<td></td>
</tr>
<tr>
<td>– USB 2.0 A-male to mini-B cable programming</td>
<td></td>
</tr>
<tr>
<td>Microsemi PCIe Adapter Card</td>
<td>PCIE-ROOTPORT-AD</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>Libero SoC PolarFire</td>
<td>v2.3</td>
</tr>
<tr>
<td>SoftConsole</td>
<td>v5.3</td>
</tr>
<tr>
<td>ModelSim</td>
<td>10.5c Pro</td>
</tr>
<tr>
<td>Synplify Pro</td>
<td>L-2017.09M-SP1-1</td>
</tr>
</tbody>
</table>
2.2 Prerequisites

Before you start:

1. Download the design files from the following link:
   - http://soc.microsemi.com/download/rsc/?f=mpf_dg0802_liberosocpolarfirev2p3_df
   - http://soc.microsemi.com/download/rsc/?f=mpf_dg0756_liberosocpolarfirev2p3_df

2. Download and install Libero SoC PolarFire on the host PC.
   - https://www.microsemi.com/products/fpga-soc/design-resources/design-software/libero-soc-polarfire#downloads
   
   The latest versions of ModelSim, Synplify Pro, and FTDI drivers are included in the Libero SoC PolarFire installation package.

2.3 Demo Design

The top-level block diagram of the PCIe Root Port design is shown in the following figure. The PolarFire PCIe Root Port can establish 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 • Block Diagram
As shown in Figure 1, page 3, the following points describe the data flow in the PCIe Root Port design:

1. The CoreUART block interfaces with the GUI.
2. The Mi-V soft processor reads/writes data to the Core_UART_0 block via the AHBtoAPB_0 bridge.
3. The Mi-V soft processor forwards the corresponding PCIe command to the PF_PCIE_0 block via 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 via 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 AHBtoAPB_0 bridge and writes the data to UART_APB slave interface to forward the data to the GUI.

As shown in Figure 1, page 3, 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 via the SLAVE ATR0 path.
2. The Mi-V soft processor accesses the Endpoint BAR 0/2 via the SLAVE ATR 0/1 path.
3. The Mi-V soft processor accesses the Root Port LSRAM memory via the SLAVE ATR 3 path.
4. The DMA is performed according to the user selection on the GUI application.

### 2.3.1 Memory and Peripheral Address Map

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

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

- **AHB MMIO Interface**: 0x60000000 to 0x7FFFFFFF
- **AHB MEM Interface**: 0x80000000 to 0x8FFFFFFF

The address map of the bus interfaces connecting Mi-V to PF_PCIE are listed in the following table.

<table>
<thead>
<tr>
<th>Bus Interface/Component</th>
<th>Description</th>
<th>Memory Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe_APB</td>
<td>This bus interface is used access the PCIe register</td>
<td>0x63000000 to 0x63FFFFFF</td>
</tr>
<tr>
<td>CoreUARTapb</td>
<td>This block establishes a UART interface to connect Mi-V processor to the external world</td>
<td>0x64000000 to 0x64FFFFFF</td>
</tr>
<tr>
<td>PCIe_AXI</td>
<td>This bus interface is the PCIe AXI slave for EP configuration or BAR space access</td>
<td>0x70000000 to 0x7000FFFF – Configuration space (Mi-V configures through PCIe APB) 0x71000000 to 0x7100FFFF – EP BAR0 space 0x72000000 to 0x7200FFFF – EP BAR2 space 0x73000000 to 0x73FFFFFF – RP AXI Master - LSRAM/DDR4 (Mi-V configures through PCIe APB)</td>
</tr>
<tr>
<td>SRAM_AHB</td>
<td>This block is the main memory of the Mi-V processor</td>
<td>0x80000000 to 0x8FFFFFFF</td>
</tr>
</tbody>
</table>

The PF_PCIE block connects to the DDR4 and LSRAM blocks via the AXI_1_MASTER bus interface. The address maps of DDR4 and LSRAM are 0x10000000 to 0x1FFFFFFF and 0x00000000 to 0x00000FFF respectively.
2.3.2 Design Implementation

Figure 2, page 5 shows the Libero SoC PolarFire software design implementation of the PCIe Root Port demo design.

Figure 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

2.3.2.1 Mi-V Subsystem

The sub-blocks of MIV_SS_0 are shown in the following figure.

Figure 3 • MIV_SS_0 SmartDesign
2.3.2.1.1 **Mi_V_Processor_0**
The Mi_V_Processor_0 (MIV_RV32IMA_L1_AHB) 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 is connected to Mi-V AHB memory interface whose memory mapped address ranges from 0x80000000 to 0x8FFFFFFF.

2.3.2.1.2 **Core_AHBLite_0**
The Core_AHBLite_0 bus is used to connect the fabric LSRAM to the Mi_V_Processor_0 block. The Core_AHBLite_0 configuration sets the slave address map to 0x80000000-0x8FFFFFFF. The LSRAM must be connected on this slave interface so that the Mi-V processor can access the LSRAM.

2.3.2.1.3 **SRAM_AHB_0**
The PF_SRAM_AHB_AXI IP is configured to access the 32 KB fabric memory (LSRAM) using the AHB interface. The *Initialize RAM at Power Up* is enabled to initialize the block with the user application (PCIe_RP_Demo.hex).

2.3.2.2 **AHBtoAXIAPB_0 SmartDesign**
The AHBtoAXIAPB_0 SmartDesign implements AHB to AXI and APB using the IP cores shown in Figure 4, page 6.

*Figure 4 • AHBtoAXIAPB_0 SmartDesign*

The AHBtoAXIAPB_0 block connects the Mi_V_Processor_0 block to:

- The PF_PCIE_0 block via PCIe_APB slave interface. The MIV_SS_0 block accesses the PCIe control registers through the PCIe_APB slave interface.
- The PF_PCIE_0 block via PCIe_AXI slave interface.
- The Core_UART_0 block via the UART_APB slave interface of Core_UART_0.

These slaves are connected at the following addresses:

- PCIe_APB slave: 0x63000000
- PCIe_AXI: 0x70000000
- UART_APB slave: 0x64000000
2.3.2.3 PCIe Rootport Subsystem

The sub-blocks of PCIe_RP_0 are shown in the following figure.

Figure 5 • PCIe_RP SmartDesign

2.3.2.3.1 PF_PCIE_0

The PF_PCIE_0 IP block is used to configure the PCIe subsystem (PCIESS) as a Root Port (PCIe 1). PCIESS block is configured for x4 lanes, 5 Gbps data rate, and APB interface for PCIe Controller access.

2.3.2.3.2 PCIe_Tx_PLL_0

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 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.

2.3.2.3.3 PCIe_TL_CLK_0 SmartDesign

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

Figure 6 • PCIe_TL_CLK_0 SmartDesign
2.3.2.3.4 PCIeM_AXI4Connect_0

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

2.3.2.4 DDR4 Subsystem

The DDR4 subsystem is configured to access the 16-bit DDR4 memory through an AXI4 64-bit
interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4
memory on PolarFire Evaluation kit.

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

2.4 Clocking Structure

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

Figure 7 • Clocking Structure
2.5 Reset Structure

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

Figure 8 • Reset Structure

The Reset_AXI_IF_0 (CoreReset_PF) block synchronizes the external user_resetn (SW6 on PolarFire Evaluation board) with DDR4 system clock (200 MHz) to generate the FABRIC_RESET_N signal, which drives the PCIe_RP_0 block.

The Reset_MIV_0 (CoreReset_PF) block synchronizes the external user_resetn (SW6 on PolarFire Evaluation board) with the RISCV system clock (100 MHz) to generate the FABRIC_RESET_N, which drives the MIV_SS_0 block.

The CoreReset_PF block uses the DEVICE_INIT_DONE signal, which is asserted when the device initialization is complete. For more information about device initialization, see UG0725: PolarFire FPGA Device Power-Up and Resets User Guide.

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

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

- Synthesize
- Place and route
- Verify Timing
- Configure Design Initialization Data and Memories
- Generate Bitstream
- Run PROGRAM Action

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

**Figure 9 • Libero Design Flow Options**

### 3.1 Synthesize

To synthesize the design:

1. From the Design Flow window, double-click **Synthesize**.

When the synthesis is successful, a green tick mark appears as shown in Figure 9, page 10.

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

We recommend viewing the `RP_Demo_Top.srr` and the `RP_Demo_Top_compile_netlist.log` files for debugging synthesis and compile errors.
3.2 Place and Route

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

To place and route the design:

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

   ![I/O Editor—XCVR View](image)

2. Place the PF_DDR4_SS_0 at NORTH_NW location as shown in the following figure.

   ![PF_DDR3_SubSystem_0 Placement](image)

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 9, page 10.

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 RP_Demo_Top_place_and_route_constraint_coverage.xml file for place and route constraint coverage.

3.2.1 Resource Utilization

The resource utilization report is written to the RP_Demo_Top_layout_log.log file in the Reports tab -> RP_Demo_Top report -> Place and Route. Table 3, page 11 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>
<thead>
<tr>
<th>Type</th>
<th>Used</th>
<th>Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4LUT</td>
<td>29744</td>
<td>299544</td>
<td>9.93</td>
</tr>
<tr>
<td>DFF</td>
<td>21536</td>
<td>299544</td>
<td>7.19</td>
</tr>
<tr>
<td>I/O Register</td>
<td>0</td>
<td>1536</td>
<td>0</td>
</tr>
</tbody>
</table>
3.3 Verify Timing

To verify timing:
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 9, page 10.
3. Right-click Verify Timing and select View Report to view the verify timing report and log files in the Reports tab.

3.4 Generate FPGA Array Data

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

A green tick mark is displayed after the successful generation of the FPGA array data as shown in Figure 9, page 10.

3.5 Configure Design Initialization Data and Memories

The Configure Design Initialization Data and Memories step generates an LSRAM 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 SRAM_AHB_0 block is initialized from μPROM.

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

Note: The LSRAM IP block can be initialized at this stage or during the LSRAM component creation using the PF_SRAM_AHB_AXI Configurator.

To select the non-volatile memory and generate the initialization client:
2. Under Third stage (uPROM/sNVM/SPI-Flash), select μPROM, as shown in Figure 12, page 12. In the Third Stage pane, select uPROM as the non-volatile memory, retain the default start address (0x00000000).

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

Figure 12 • Design and Memory Initialization Window
3. On the **Fabric RAMs** tab, select RP_Demo_Top/MIV_SS_0/SRAM_AHB from the list of logical instances, and click **Edit**, as shown in Figure 13, page 13. The RP_Demo_Top/MIV_SS_0/SRAM_AHB instance is the Mi-V processor's main memory. The System Controller initializes this instance with the imported client at power-up.

![Figure 13 • Fabric RAMs Tab](image)

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 14 • Edit Fabric RAM Initialization Client Dialog Box](image)

**Note:** In the following dialog box, browse the `mpf_dg0802_liberosocpolarfirev2p3_df\liberoProject\PCIe_RP_Demo.hex` file and double-click it.

**Note:** 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 15 • Applying Fabric RAM Content*

7. The initialization client for RP_Demo_Top\MIV_SS_0\SRAM_AHB instance is generated and stored in µPROM. This step can be verified by viewing the third stage client created in the µPROM tab as shown in the following figure.

*Figure 16 • Third Stage INIT Client*

The first and second stage clients are generated and stored in sNVM by default.
3.6 Generate Bitstream

To generate the bitstream:

1. Double-click Generate Bitstream from the Design Flow tab. When the bitstream is successfully generated, a green tick mark appears as shown in Figure 9, page 10.
2. Right-click Generate Bitstream and select View Report to view the corresponding log file in the Reports tab.

3.7 Run PROGRAM Action

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

1. Ensure that the following Jumper Settings are set on the board, which will be used as the Root Port device.

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>J18, J19, J20, J21, J22</td>
<td>Short pin 2 and 3 for programming the PolarFire FPGA through FTDI</td>
<td>Closed</td>
</tr>
<tr>
<td>J28</td>
<td>Short pin 1 and 2 for programming through the on-board FlashPro5</td>
<td>Open</td>
</tr>
<tr>
<td>J26</td>
<td>Short pin 1 and 2 for programming through the FTDI SPI</td>
<td>Closed</td>
</tr>
<tr>
<td>J4</td>
<td>Short pin 1 and 2 for manual power switching using SW3</td>
<td>Closed</td>
</tr>
<tr>
<td>J12</td>
<td>Short pin 3 and 4 for 2.5 V</td>
<td>Closed</td>
</tr>
</tbody>
</table>

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.

When the device is programmed successfully, a green tick mark appears as shown Figure 9, page 10. The device is successfully programmed, see Setting Up the Demo, page 16.
4 Setting Up the Demo

Setting up the demo involves 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

**Note:** If Board A was programmed using Libero, use the following section to program the Endpoint design on Board B.

4.1 Programming the PolarFire Devices

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

To program:

1. Take Board A and ensure that the Jumper Settings are set as listed in Table 4, page 15.
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, start the FlashPro software.
6. Click New Project to create a new project.
7. In the New Project window, do the following, and click OK:
   - Enter a project name.
   - Select Single device as the programming mode.
8. Click Configure Device.
9. Browse the `RP_Demo_Top.stp` file from the `mpf_dg0802_liberosocpolarfirev2p3_df\ProgrammingFile` location from the Load Programming File window.
10. Click **Program** to program the device.
11. The Programmer List Window shows the Programmer Name, Programmer Type, Port, Programmer Status, and the Programmer Enabled information. When the device is programmed successfully, a Run Program PASSED status is displayed as shown in the following figure.

![Figure 17 • Program Action Completed](image)

Root port design is successfully programmed on Board A.

4.2 Connecting the Two Boards

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

To connect the boards:

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. After successfully connecting the two boards, the demo setup looks like the following figure:

Figure 18 • Demo Setup
5 Running the Demo

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

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

5.1 Installing the GUI

To install the GUI:

1. Extract the contents of the mpf_dg0802_liberosocpolarfirev2p3_df.zip file. From the mpf_dg0802_liberosocpolarfirev2p3_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.

5.2 Viewing the Enumeration Data

Before you start, 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 Microsemi PCIe adapter card and powered-up.
3. LED 9, 10, and 11 are glowing 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:

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 19 • 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 the Device Info tab to view the Endpoint device information as shown in the following figure.

Figure 20 • Endpoint Device Information

![Figure 20](endpoint_device_info.png)

5. Click the Config Space tab to view the basic Type 0 Configuration Settings of the Endpoint as shown in the following figure.

Figure 21 • Endpoint Config Space-Basic

![Figure 21](endpoint_config_space.png)
6. Click the **Advanced** tab to view the MSI Capabilities of the Endpoint as shown in the following figure.

*Figure 22 • Endpoint Config Space-Advanced*

![Endpoint Config Space-Advanced](image)

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

### 5.3 Running the Control Plane Commands

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

#### 5.3.1 Controlling Endpoint LEDs

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

To issue LED Commands:

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

*Figure 23 • Single LED Control*

![Single LED Control](image)

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, LED6 on the Endpoint board glows.

3. Click the Start LED ON/OFF Walk button as shown in Figure 24, page 21.

Figure 24 • 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.

5.3.2 Reading Endpoint DIP SW Status

To read the DIP SW Status:

1. Click the Enable DIP SW Session button as shown in the following figure.

Figure 25 • 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 Figure 26, page 22. Change the DIP SW positions on the Endpoint board and observe the same in GUI.

**Figure 26 • Endpoint DIP SW Status**

5.3.3 Reading MSI Count Values

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 the Enable Interrupt Session button as shown in Figure 27, page 22.

**Figure 27 • 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. In the reference design, the Root Port allocates 4 types of MSI as shown in the following table.

<table>
<thead>
<tr>
<th>MSI Number</th>
<th>Interrupt Type on the Endpoint Board</th>
<th>Mapped Interrupt Counter on the GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SW10</td>
<td>Interrupt Counter1</td>
</tr>
<tr>
<td>2</td>
<td>SW9</td>
<td>Interrupt Counter2</td>
</tr>
<tr>
<td>3</td>
<td>SW8</td>
<td>Interrupt Counter3</td>
</tr>
<tr>
<td>4</td>
<td>SW7</td>
<td>Interrupt Counter4</td>
</tr>
</tbody>
</table>

2. Press the switch and observe interrupt count.

Figure 28 • Interrupt Counter4

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

5.3.4 Running BAR2 Memory Read/Write Commands

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 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:

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

Figure 29 • BAR2-LSRAM Read Option
2. Select any memory location and edit the value of that location. See the following figure for example.

*Figure 30 • BAR2-LSRAM Write*

![BAR2-LSRAM Write](image)

3. The edited memory location turns green and the value entered is written to the Endpoint LSRAM memory location as shown in *Figure 31*, page 24.

*Figure 31 • BAR2-LSRAM Write Successful*

![BAR2-LSRAM Write Successful](image)

4. Similarly change any other memory location also.
5. Click the 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.
5.4 **Running the Data Plane Commands**

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

- Running DMA operations
- Running memory test

5.4.1 **Running DMA Operations**

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:

1. Click the DMA Operations tab as shown in Figure 32, page 25.
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
   - Click **Start transfer**.

*Figure 32 • Initiating RP LSRAM to EP LSRAM DMA*
The GUI displays the corresponding throughput details and graph as shown in the following figure.

Figure 33 • 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.
   - 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.

5.4.2 Running Memory Test

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:

1. Select the DMA parameters like Transfer Size, Pattern Type, Endpoint and Root Port memory Type, Root Port and Endpoint Offset Address as shown in the following figure.

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

   ![Figure 34 • Memory Test Feature](image)

2. Click Memory Test.

3. Select DDR4 as the Root Port memory type and click View Memory as shown in the following figure to read the Root Port DDR4.

   ![Figure 35 • The View Memory Option](image)

4. The GUI displays the data pattern written to the Root port DDR4 as shown in Figure 36, page 28.
Running the Demo

**Figure 36 • 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 37 • Endpoint LSRAM Memory Content**
5.5 **PolarFire DMA Throughput Summary**

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

*Table 6 - Throughput Summary*

<table>
<thead>
<tr>
<th>DMA Transfer Type</th>
<th>DMA Size</th>
<th>Throughput (MBps)</th>
<th>Throughput Average (MBps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP LSRAM to EP LSRAM</td>
<td>512 K</td>
<td>1017</td>
<td>1017</td>
</tr>
<tr>
<td>EP LSRAM to RP LSRAM</td>
<td>512 K</td>
<td>925</td>
<td>925</td>
</tr>
<tr>
<td>RP LSRAM to EP DDR3</td>
<td>512 K</td>
<td>496</td>
<td>496</td>
</tr>
<tr>
<td>EP DDR3 to RP LSRAM</td>
<td>512 K</td>
<td>296</td>
<td>296</td>
</tr>
<tr>
<td>RP LSRAM to EP DDR4</td>
<td>512 K</td>
<td>1005</td>
<td>1005</td>
</tr>
<tr>
<td>EP DDR4 to RP LSRAM</td>
<td>512 K</td>
<td>463</td>
<td>463</td>
</tr>
<tr>
<td>RP DDR4 to EP LSRAM</td>
<td>512 K</td>
<td>556</td>
<td>556</td>
</tr>
<tr>
<td>EP LSRAM to RP DDR4</td>
<td>512 K</td>
<td>925</td>
<td>925</td>
</tr>
<tr>
<td>RP DDR4 to EP DDR3</td>
<td>512 K</td>
<td>493</td>
<td>493</td>
</tr>
<tr>
<td>EP DDR3 to RP DDR4</td>
<td>512 K</td>
<td>295</td>
<td>295</td>
</tr>
<tr>
<td>RP DDR4 to EP DDR4</td>
<td>512 K</td>
<td>556</td>
<td>556</td>
</tr>
<tr>
<td>EP DDR4 to RP DDR4</td>
<td>512 K</td>
<td>462</td>
<td>462</td>
</tr>
</tbody>
</table>
The DDR4 subsystem is configured to access the 16-bit DDR4 memory through an AXI4 64-bit interface. The DDR4 memory initialization and timing parameters are configured as per the DDR4 memory on PolarFire Evaluation kit. The following figure shows the general configuration settings for the DDR4 memory.

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

*Figure 38 • PF_DDR4 Configurator—General*
The following figure shows the initialization configuration settings for the DDR4 memory.

*Figure 39 • PF_DDR4 Configurator—Memory Initialization*
The following figure shows the controller configuration settings for the DDR4 memory.

Figure 40 • PF_DDR4 Configurator—Controller
Appendix: References

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

- For more information about PolarFire transceiver blocks, PF_TX_PLL, and PF_XCVR_REF_CLK, see UG0677: PolarFire FPGA Transceiver User Guide.
- For more information about PF_PCIE, see UG0685: PolarFire FPGA PCI Express User Guide.
- For more information about PF_CCC, see UG0684: PolarFire FPGA Clocking Resources User Guide.
- For more information about DDR3 memory, see UG0676: PolarFire FPGA DDR Memory Controller User Guide.
- For more information about Libero, ModelSim, and Synplify, see the Microsemi Libero SoC PolarFire web page.
- For more information about PolarFire FPGA Evaluation Kit, see UG0747: PolarFire FPGA Evaluation Kit User Guide.
- For more information about CoreAHBLite, see CoreAHBLite Handbook.
- For more information about CoreAHBtoAPB3, see CoreAHBtoAPB3 Handbook.
- For more information about CoreUART, see CoreUART Handbook.