## Contents

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

2 PolarFire FPGA Splash Kit Low Power ............................. 2  
  2.1 Design Requirements .................................................. 2  
  2.2 Prerequisites ........................................................... 3  
  2.3 Demo Design ............................................................ 3  
    2.3.1 Design Implementation ......................................... 4  
    2.3.2 IP Configuration ................................................ 5  
  2.4 Clocking Structure ................................................... 11  

3 Libero Design Flow ...................................................... 12  
  3.1 Synthesize ............................................................ 12  
  3.2 Place and Route ....................................................... 13  
    3.2.1 Resource Utilization .............................................. 13  
  3.3 Verify Timing .......................................................... 13  
  3.4 Verify Power ........................................................... 13  
  3.5 Generate FPGA Array Data ........................................... 14  
  3.6 Configure Design Initialization Data and Memories ............... 14  
  3.7 Generate Bitstream ................................................... 15  
  3.8 Run PROGRAM Action ................................................ 15  

4 Programming the Device Using FlashPro .......................... 16  

5 Running the Demo ......................................................... 18  

6 Power Correlation .......................................................... 20  

7 Appendix: Power Estimator ............................................... 21  

8 Appendix: References ..................................................... 24
Figures

Figure 1  Block Diagram  ................................................................. 3
Figure 2  Top-Level SmartDesign Component  .................................... 4
Figure 3  XCVR_Block SmartDesign Component  .................................. 4
Figure 4  PolarFire_Fabric SmartDesign Component  ............................ 5
Figure 5  PF_XCVR_REF_CLK Configurator  ........................................ 6
Figure 6  PF_TX_PLL Configurator  ................................................... 7
Figure 7  PF_XCVR Configurator  ...................................................... 8
Figure 8  PF_INIT_MONITOR Configurator  ......................................... 9
Figure 9  PF_CCC Configurator  ....................................................... 10
Figure 10  Clocking Structure  .......................................................... 11
Figure 11  Libero Design Flow Options  ............................................ 12
Figure 12  Post-Layout Power Summary  ............................................ 14
Figure 13  Post-Layout Transceiver Power  ......................................... 14
Figure 14  Power on the board using SW1 slide switch.Board Setup  ........... 16
Figure 15  RUN PASSED  ............................................................... 17
Figure 16  COM Port Setup  ............................................................. 18
Figure 17  Total Power  .................................................................... 19
Figure 18  Device Settings  ............................................................... 21
Figure 19  Initialize Power Estimator  ................................................. 21
Figure 20  Settings in Initialize Power Estimator Dialog Box  ................... 22
Figure 21  Transceiver Worksheet  ..................................................... 22
Figure 22  Frequency Settings in PLL and DLL Worksheet  ....................... 22
Figure 23  Total Power Summary  ..................................................... 23
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Mapping LEDs to FPGA Component</td>
<td>2</td>
</tr>
<tr>
<td>Table 2</td>
<td>Design Requirements</td>
<td>2</td>
</tr>
<tr>
<td>Table 3</td>
<td>I/O Signals</td>
<td>5</td>
</tr>
<tr>
<td>Table 4</td>
<td>Resource Utilization</td>
<td>13</td>
</tr>
<tr>
<td>Table 5</td>
<td>Jumper Settings</td>
<td>15</td>
</tr>
<tr>
<td>Table 6</td>
<td>Power Correlation</td>
<td>20</td>
</tr>
</tbody>
</table>
1 Revision History

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

1.1 Revision 3.0
The document was updated for Libero® SoC PolarFire v2.2 release.

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

1.3 Revision 1.0
The first publication of this document.
2 PolarFire FPGA Splash Kit Low Power

Microsemi PolarFire® FPGAs are designed to meet the demand for low-power. PolarFire devices exhibit lower-power consumption in static and dynamic modes. This document explains how to use the accompanying demo design to demonstrate low power features of the PolarFire device. The demo design instantiates a 16-bit cascaded counter with 500 µSRAM, 500 LSRAM, and 500 Mathblocks. This demo shows power consumption measurement using Microsemi Power estimator, SmartPower and silicon power on Power Monitor GUI. Microsemi PowerMonitor application is used to measure the real-time power. The demo design also includes a transceiver block.

The following table lists the FPGA component to LED mapping implemented using the demo design.

<table>
<thead>
<tr>
<th>On-board LEDs</th>
<th>FPGA Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 and 2</td>
<td>Fabric</td>
</tr>
<tr>
<td>3 and 4</td>
<td>LSRAM blocks</td>
</tr>
<tr>
<td>5 and 6</td>
<td>Mathblocks</td>
</tr>
<tr>
<td>7 and 8</td>
<td>µSRAM blocks</td>
</tr>
</tbody>
</table>

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

• Using the pre-generated .stp file: To program the device using the .stp file provided along with the demo design, see Programming the Device Using FlashPro, page 16.
• Using Libero SoC PolarFire: To program the device using Libero SoC PolarFire, see Libero Design Flow, page 12.

2.1 Design Requirements

The following table lists the hardware, software requirements to run the demo, and the IP cores used to build the demo design.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>MPF300T SPLASH KIT</td>
<td>Rev 2 or later</td>
</tr>
<tr>
<td>– 12 V, 5 A AC power adapter and cord</td>
<td></td>
</tr>
<tr>
<td>– USB 2.0 A to Mini-B cable for UART and programming</td>
<td></td>
</tr>
<tr>
<td>Host PC</td>
<td>Windows 7, 8.1, or 10</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>Libero SoC PolarFire</td>
<td>v2.2</td>
</tr>
<tr>
<td>FlashPro</td>
<td>v2.2</td>
</tr>
<tr>
<td><strong>IP</strong></td>
<td></td>
</tr>
<tr>
<td>PF_XCVR</td>
<td>1.0.231</td>
</tr>
<tr>
<td>PF_TX_PLL</td>
<td>1.0.112</td>
</tr>
<tr>
<td>PF_XCVR_REF_CLK</td>
<td>1.0.103</td>
</tr>
<tr>
<td>PF_OSC_0</td>
<td>1.0.102</td>
</tr>
<tr>
<td>PF_INIT_MONITOR</td>
<td>2.0.103</td>
</tr>
</tbody>
</table>
2.2 Prerequisites

Before you start:

1. Download the demo design files from the following location:
   http://soc.microsemi.com/download/rsc/?f=mpf_dg0814_liberosocpolarfirev2p2_df

2. Download and install Libero SoC PolarFire v2.2 on the host PC from the following location.
   https://www.microsemi.com/products/fpga-soc/design-resources/design-software/libero-soc-polar-fire#downloads

   Note: The latest versions of ModelSim and Synplify Pro are included in the Libero SoC PolarFire installation package.

3. Download the latest PolarFire power estimator from the following location.

4. Download the Microsemi PowerMonitor application from the following location:
   http://soc.microsemi.com/download/rsc/?f=polarfire_power_monitor

2.3 Demo Design

The block diagram of the transceiver and the low-power design is illustrated in the following figure.

**Figure 1** Block Diagram

In the demo design:

- The DEVICE_INIT_DONE signal of the PF_INIT_MONITOR_0 block is asserted after the device is initialized.
- The DEVICE_INIT_DONE signal and the PLL_LOCK_0 signal of PF_CCC_0 is used to control the reset signal of the FabricLogic_0 and XCVR block.
- OSC_RC160MHz_0 provides 160 MHz clock source to the PF_CCC_0 block.
- The PF_CCC_0 block provides the following fabric clocks:
  - CLK: 100 MHz clock for the fabric
  - CLK1: 100 MHz clock for the µSRAM blocks
  - CLK2: 100 MHz clock for the LSRAM blocks
  - CLK3: 100 MHz clock for Mathblocks
- These separate clocks are provided in the design, to gate clocks to each fabric block if required. The transceiver (PF_XVCR) block instantiates the transceiver in 8b10b mode. This block receives clock
from the REF_CLK signal of PF_XCVR_REF_CLK_0. The PF_TX_PLL_0 block also derives its reference clock from REF_CLK of PF_XCVR_REF_CLK_0.

- The TX and RX lanes of the transceiver are on-board looped back.
- The pattern_gen_0 block is implemented to send data to the transceiver block. The pattern_chk_0 block is implemented to check errors in the data received from the transceiver block.

### 2.3.1 Design Implementation

This section describes the top-level SmartDesign canvas and important I/O signals.

The top-level SmartDesign canvas contains the following blocks:

- XCVR_Block, page 5
- PolarFire_Fabric, page 8

The following figure shows the top-level design component of the PolarFire low-power design.

**Figure 2 • Top-Level SmartDesign Component**

The top component instantiates the XCVR_Block as shown in the following figure.

**Figure 3 • XCVR_Block SmartDesign Component**

The PolarFire_Fabric component is shown in the following figure.
The following table lists the important I/O signals of the design.

### Table 3 • I/O Signals

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PolarFire_Fabric</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED_1, LED_2, LED_3, LED_4, LED_5, LED_6, LED_7, LED_8</td>
<td>Output</td>
<td>Mapped to LEDs 1, 2, 3, 4, 5, 6, 7, and 8 on the board</td>
</tr>
<tr>
<td><strong>PF_XCVR</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REF_CLK_PAD_P and REF_CLK_PAD_N</td>
<td></td>
<td>Indicates that the differential reference clock is generated from the on-board 125 MHz oscillator</td>
</tr>
<tr>
<td>PADs_IN</td>
<td>Input</td>
<td>Transceiver LANE0_RXD_P and LANE0_RXD_N (C20 and C19) on the board</td>
</tr>
<tr>
<td>PADs_OUT</td>
<td>Output</td>
<td>Indicates that LANE0_TXD_P and LANE0_TXD_N (D22 and D21), which are looped back on-board.</td>
</tr>
<tr>
<td>error_o</td>
<td>Output</td>
<td>Asserted when RX and TX data do not match</td>
</tr>
<tr>
<td>lock_o</td>
<td>Output</td>
<td>Asserted when RX and TX data match</td>
</tr>
<tr>
<td>rx_val_o</td>
<td>Output</td>
<td>Indicates that the XCVR has received and validated the 8b/10b idle frames</td>
</tr>
</tbody>
</table>

### 2.3.2 IP Configuration

The demo design consists of the following two top-level blocks:

- **XCVR_Block**, page 5
- **PolarFire_Fabric**, page 8

The following sections describe the user-defined blocks, IP blocks, and their configurations in these top-level blocks.

#### 2.3.2.1 XCVR_Block

The XCVR_Block contains the following modules:

- **PF_XCVR_REF_CLK_0**, page 6
• PF_TX_PLL_0, page 6
• pattern_gen_0, page 7
• PF_XCVR_0, page 7
• pattern_chk_0, page 8

These modules are described in the following sections.

2.3.2.1.1 PF_XCVR_REF_CLK_0

The PF_XCVR_REF_CLK IP block provides the reference clock to PF_TX_PLL. The transceiver clock can be used as a single differential clock or as two single-ended reference clocks. In the demo design, XCVR_REF_CLK is configured as a differential clock. PF_XCVR_REF_CLK IP block is configured as shown in the following figure.

Figure 5 • PF_XCVR_REF_CLK Configurator

2.3.2.1.2 PF_TX_PLL_0

The PF_TX_PLL IP block provides the reference clock to the transceiver lane. This block is configured as shown Figure 6, page 7.
2.3.2.1.3 pattern_gen_0
The pattern_gen_0 user defined block implements a 32-bit incremental counter pattern delimited by K28.5 character for comma alignment. The packet payload is a simple counter pattern.

2.3.2.1.4 PF_XCVR_0
The PF_XCVR_0 IP block is configured in 8b10b mode, 32-bit PCS fabric interface width, and receives 125 MHz reference clock as shown in the following figure.
2.3.2.1.5 pattern_chk_0

The pattern_chk_0 user defined block checks the incoming data for bit errors. The incoming data is the counter pattern which is internally looped back. It reports the following three status signals:

- rx_val_o: This signal is asserted when receiver achieves comma alignment
- lock_o: This signal is asserted when there are no bit errors
- error_o: This signal is asserted when there are bit errors

All the above signals are connected to bread board connector J33 for probing.

2.3.2.2 PolarFire_Fabric

The PolarFire_standby block contains:

- PF_INIT_MONITOR_0, page 9
- OSC_clock_source_0, page 9
- PF_CCC_0, page 9
- Fabric_Logic_0, page 10

These modules are described in the following sections.
2.3.2.2.1 PF_INIT_MONITOR_0

This IP block asserts DEVICE_INIT_DONE signal once the entire device initialization is complete. It is configured as shown in the following figure.

Figure 8 • PF_INIT_MONITOR Configurator

2.3.2.2.2 OSC_clock_source_0

This IP block instantiates the on-chip 160 MHz oscillator that feeds clock to the REF_CLK_0 input of the PF_CCC_0 module and the Standby_Control_0 block.

2.3.2.2.3 PF_CCC_0

The PF_CCC IP block feeds clock signals to the fabric (CLK), µSRAM (CLK1), LSRAM (CLK2), and Mathblocks (CLK3). The PF_CCC_0 is configured to receive an input frequency of 160 MHz as shown in the following figure.
2.3.2.2.4 Fabric Logic 0
The fabric logic block instantiates a counter logic along with 500 µSRAM, 500 LSRAM, and 500 Mathblocks, which utilizes 70% of 4 input LUT and DFF. The width, depth, and number of instantiations of these blocks are parameterized in the `fabric_logic.v` file. The user can increase or decrease these parameters to see how the resource utilization varies. For more information about these blocks, see UG0680: PolarFire FPGA Fabric User Guide.
2.4 Clocking Structure

The following figure shows the clocking structure implemented in the demo design.

*Figure 10 • Clocking Structure*
3 Libero Design Flow

This chapter describes the Libero design flow, which involves the following steps:

- Synthesize, page 12
- Place and Route, page 13
- Verify Timing, page 13
- Verify Power, page 13
- Generate FPGA Array Data, page 14
- Configure Design Initialization Data and Memories, page 14
- Generate Bitstream, page 15
- Run PROGRAM Action, page 15

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

Figure 11 • Libero Design Flow Options

3.1 Synthesize

To synthesize the design:

1. Double-click Synthesize from the Design Flow tab.
   When the synthesis is successful, a green tick mark appears as shown in Figure 11, page 12.
2. Right-click Synthesize and select View Report to view the synthesis report and log files in the Reports tab.
3. View the Top_Level.srr and the Top_Level_compile_netlist.log files for debugging synthesis and compiling respective errors.
3.2 Place and Route

To place and route the design:

1. Double-click **Place and Route** from the **Design Flow** tab.
   When place and route is successful, a green tick mark appears as shown in Figure 11, page 12.

2. Right-click **Place and Route** and select View Report to view the place and route report and log files in the **Reports** tab.

3. View the **Top_Level_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 **Top_Level_layout_log.log** file in the **Reports** tab -> **Place and Route**. The following table 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>185181</td>
<td>299544</td>
<td>61.82</td>
</tr>
<tr>
<td>DFF</td>
<td>205888</td>
<td>299544</td>
<td>68.73</td>
</tr>
<tr>
<td>I/O Register</td>
<td>0</td>
<td>242</td>
<td>0.00</td>
</tr>
<tr>
<td>Logic Element</td>
<td>214482</td>
<td>299544</td>
<td>71.60</td>
</tr>
<tr>
<td>µSRAM</td>
<td>500</td>
<td>2772</td>
<td>18.04</td>
</tr>
<tr>
<td>LSRAM</td>
<td>500</td>
<td>952</td>
<td>52.52</td>
</tr>
<tr>
<td>MACC</td>
<td>500</td>
<td>924</td>
<td>54.11</td>
</tr>
</tbody>
</table>

The fabric resource usage can be viewed in the **Top_Level_compile_netlist_resources.xml** under **Reports** tab -> **Synthesize**.

3.3 Verify Timing

To verify timing:

1. Double-click **Verify Timing** from the **Design Flow** tab.
   When the design successfully meets the timing requirements, a green tick mark appears as shown in Figure 11, page 12.

2. Right-click **Verify Timing** and select View Report to view the verify timing report and log files in the **Reports** tab.

3.4 Verify Power

SmartPower gives post-layout power consumption estimation. Verify Power invokes SmartPower at the back end. The power consumption figures are used to correlate with the real-time power consumption measured using the PowerMonitor application.

To generate power report:

1. Double-click **Verify Power** from the **Design Flow** tab.
   When the Smart Power successfully calculates the power consumption details, a green tick mark appears as shown in Figure 11, page 12.

2. Right-click **Verify Power** and select View Report to view the post layout power consumption report and log files in the **Reports** tab.
The following figures show the design details and power summary.

**Figure 12 • Post-Layout Power Summary**

The following figure shows the transceiver power.

**Figure 13 • Post-Layout Transceiver Power**

### 3.5 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 11, page 12.

### 3.6 Configure Design Initialization Data and Memories

This option is used to create the XCVR initialization client, used in the demo design. When the PolarFire device powers up, the transceiver block is initialized by the initialization client generated during the **Configure Design Initialization Data and Memories** stage in the design flow. For more information, see **UG0725: PolarFire FPGA Device Power-up and Resets User Guide**. The demo design retains these configuration settings. This step is executed successfully when you double-click **Generate Bitstream** option.

The XCVR client generation is successful and a green tick mark appears next to the option as shown in Figure 11, page 12. When the device is programmed, the XCVR initialization client is read from the sNVM.
3.7 **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 11, page 12.
2. Right-click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

3.8 **Run PROGRAM Action**

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

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

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J5, J6, J7, J8, J9</td>
<td>Close pin 2 and 3 for programming the PolarFire FPGA through FTDI</td>
</tr>
<tr>
<td>J11</td>
<td>Close pin 1 and 2 for programming through FTDI Chip</td>
</tr>
<tr>
<td>J10</td>
<td>Open pin 1 and 2 for programming through the FTDI SPI</td>
</tr>
<tr>
<td>J4</td>
<td>Close pin 1 and 2 for manual power switching using SW1</td>
</tr>
<tr>
<td>J3</td>
<td>Open pin 1 and 2 for 1.0 V</td>
</tr>
</tbody>
</table>

2. Connect the power supply cable to the J2 connector on the board.
3. Connect the USB cable from the **Host PC** to the J1 (FTDI port) on the board.
4. Power on the board using the **SW1** slide switch.
5. Double-click **Run PROGRAM Action** from the **Libero > Design Flow** tab.
6. Right-click **Run PROGRAM Action** and select **View Report** to view the corresponding log file in the **Reports** tab.

When the device is successfully programmed, a green tick mark appears as shown in Figure 11, page 12. See, **Running the Demo**, page 18 to run the low power demo.
This chapter describes how to program the PolarFire device with .stp programming file using Flashpro. The .stp file is available at:

```
mpf_dg0814_liberosocpolarfirev2p2_df\Programming_File
```

Follow these steps:

1. Ensure that the jumper settings on the board are same as listed in Table 5, page 15.
2. Connect the power supply cable to the J2 connector on the board.
3. Connect the USB cable from the host PC to J1 (FTDI port) on the board.

![Figure 14](image)

Power on the board using SW1 slide switch. Board Setup

4. On the host PC, start the FlashPro software.
5. Click **New Project** to create a new project.

   In the **New Project** window, do the following, and click OK:

6. Enter a project name.
7. Select **Single device** as the programming mode.
8. Click **Configure Device**.
9. Click **Browse**, and select the `PF_low_power_demo.stp` file from the **Load Programming File** window.
10. Click **Program** to program the device.

   The **Programmer List** window in the FlashPro, shows the Programmer Name, Programmer Type, Port, Programmer Status, and the Programmer Enabled information.
When the device is programmed successfully, a **Run PASSED** status is displayed as shown in the following figure. See, Running the Demo, page 18 to run the low power demo.

**Figure 15 • RUN PASSED**

An image showing the process of programming a device using FlashPro.
5 Running the Demo

This section describes the procedure to put the fabric, µSRAM, LSRAM, and Mathblocks in standby and active modes and to monitor the total static and dynamic power consumed by the device in both the modes. The power is monitored using the Microsemi PowerMonitor GUI application.

PolarFire Evaluation Kit board comes with power monitoring solution, implemented using the on-board SmartFusion A2F 200 device and the PowerMonitor application. The PowerMonitor application connects to the power monitoring program running on the A2F 200 device to measure power. For more information on PowerMonitor, see UG0747: PolarFire FPGA Evaluation Kit User Guide.

Prerequisites for the procedure:
1. The PolarFire Splash board is connected.
2. The PolarFire FPGA is programmed with the Low Power design.

Following are the steps:
1. The LEDs {1, 2}, {3, 4},{5, 6}, and {7,8} blink at different rates. This indicates that the fabric components are in active mode.
2. On the host PC, download the Microsemi PowerMonitor application from the following location: http://soc.microsemi.com/download/rsc/?f=polarfire_power_monitor
3. Follow the instructions in the installation wizard to install the PowerMonitor application.
4. On the host PC desktop, click Start and select PowerMonitor.
5. In the COMPort Setup dialog box, select the highest COM port from the drop-down and click Connect as shown in the following figure.

The PowerMonitor application successfully connects to the board and starts displaying the Core Fabric (VDD) power, Fabric PLL (VDD25) power, Transceiver Core (VDDA) power, and Transceiver PLL (VDDA25) power.
6. The total power consumed by the device is displayed, as shown in the following figure.

Figure 17 • Total Power

7. Close the PowerMonitor application and power-down the board.
This chapter describes the correlation between the power estimated by SmartPower/Power Estimator and real-time power measured using the PowerMonitor.

The real-time transceiver power on the board includes the Transceiver Core power and Transceiver PLL power. The following table lists the power correlation.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Transceiver Power (W)</th>
<th>Total Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Estimator</td>
<td>0.062</td>
<td>1.836</td>
</tr>
<tr>
<td>SmartPower</td>
<td>0.051</td>
<td>1.7260</td>
</tr>
<tr>
<td>Power Monitor (real-time onboard power)</td>
<td>0.049</td>
<td>1.776</td>
</tr>
</tbody>
</table>

For this demo design a 5% difference is seen between the SmartPower total power estimate and PowerMonitor total power.
The following steps describe how to use the PolarFire_Power_Estimator spreadsheet to get the total power and device static power estimates.

Steps to estimate total power and device static power:

1. Download the Power Estimator spreadsheet from the following location.

2. Double-click Power Estimator file to start the power estimator spreadsheet.
   By default, the Summary worksheet opens. Summary worksheet contains the device settings and the power summary.

3. In the Summary worksheet, edit the Settings as shown in the following figure.

   Figure 18 • Device Settings

   The Summary worksheet has an integrated Initialize Power Estimator wizard. This wizard provides an option to select design specific information. Upon running the wizard, it populates the power calculator spreadsheet with design information and performs power estimation for the design.

4. Click Initialize Power Estimator.

   Figure 19 • Initialize Power Estimator

5. In the Initialize Power Estimator dialog box, set the Transceiver, FPGA Fabric, I/O, Default Toggle rate, and the Default RAM Enable rate parameters based on the demo design and then click Reset and Initialize, as shown in the following figure.
Note: For information about the transceiver, RAM and fabric resource utilization, see the Top_Level_compile_netlist_resources.xml file under Reports tab -> Synthesize

Figure 20 • Settings in Initialize Power Estimator Dialog Box

The Clock, Logic, μSRAM, LSRAM, Mathblock, I/O, and the Transceiver worksheets are updated automatically with the Initialize Power Estimator data.

6. In the Transceiver worksheet, set the PCS mode to 8B10B and PCS width to 32-bits based on the demo design specification as shown in the following figure.

Figure 21 • Transceiver Worksheet

7. Set the Reference frequency, Output 0, Output 1, Output 2, Output 3 frequencies, and Mode in the PLL & DLL worksheet, as shown in the following figure.

Figure 22 • Frequency Settings in PLL and DLL Worksheet
8. Switch to the **Summary** worksheet to view the estimated total power. The estimated total power is highlighted in the following figure.

*Figure 23 • Total Power Summary*

<table>
<thead>
<tr>
<th>Summary</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Power (W)</td>
<td>1.836</td>
</tr>
<tr>
<td>Device Static (W)</td>
<td>0.091</td>
</tr>
<tr>
<td>Core Dynamic (W)</td>
<td>1.680</td>
</tr>
<tr>
<td>I/O (W)</td>
<td>0.005</td>
</tr>
<tr>
<td>Transceiver (W)</td>
<td>0.060</td>
</tr>
<tr>
<td>Junction Temperature Tj (°C)</td>
<td>25.00</td>
</tr>
<tr>
<td>Effective Theta JA (°C/W)</td>
<td>N/A</td>
</tr>
<tr>
<td>Thermal Margin</td>
<td></td>
</tr>
<tr>
<td>Maximum Ta (°C)</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum Power (W)</td>
<td>N/A</td>
</tr>
</tbody>
</table>

9. Close the **Power Estimator** wizard.
This section lists documents that provide more information about the tools and IP cores used in the demo design.

- For information about PolarFire transceiver blocks, PF_TX_PLL, and PF_XCVR_REF_CLK, see UG0677: PolarFire FPGA Transceiver User Guide.
- For more information about the PF_INIT_MONITOR IP core, see UG0725: PolarFire FPGA Device Power-Up and Resets User Guide.
- For more information about the PF_CCC IP core, see UG0684: PolarFire FPGA Clocking Resources User Guide.
- For more information about the Power Estimator spreadsheet, see the PolarFire Power Estimator.
- For more information about Libero, ModelSim, and Synplify, see the Microsemi Libero SoC PolarFire webpage.