DG0774
Demo Guide
PolarFire FPGA Low Power

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) 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. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200774. 3.0 6/18

Contents

Figures

Tables

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

This document was updated for Libero® SoC PolarFire v2.2 release.

1.2 Revision 2.0

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

1.3 Revision 1.0

Revision 1.0 was the first publication of this document.

2 PolarFire FPGA 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 Polarfire by instantiating 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 1 • **Mapping LEDs to FPGA Component**

On-board LEDs	FPGA Component
4 and 5	Fabric
6 and 7	LSRAM blocks
8 and 9	Mathblocks
10 and 11	µSRAM blocks

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 2 • **Design Requirements**

Requirement	Version
Hardware	
MPF300-EVAL-KIT – 12 V, 5 A AC power adapter and cord – USB 2.0 A to Mini-B cable for UART and programming	Rev D or later
2 SMA-to-SMA cables (not provided with the kit)	
Host PC	Windows 7, 8.1, or 10
Software	
Libero SoC PolarFire	v2.2
FlashPro	v2.2
IP	
PF_XCVR	1.0.231
PF_TX_PLL	1.0.112
PF_XCVR_REF_CLK	1.0.103

Table 2 • **Design Requirements**

PF_OSC_0	1.0.102
PF_INIT_MONITOR	2.0.103
PF_CCC	1.0.113
CORERESET_PF	2.1.100

2.2 Prerequisites

Before you start:

1. Download the demo design files from the following location:
   http://soc.microsemi.com/download/rsc/?f=mpf_dg0774_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-polarfire#downloads

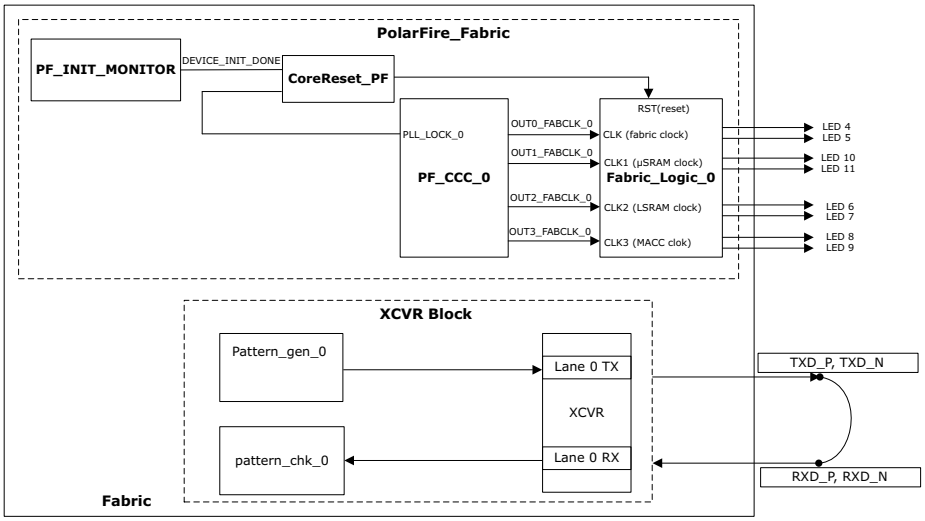
Note: The latest versions of ModelSim and Synplify Pro are included in the Libero SoC PolarFire installation package.

3. Download the latest PolarFire power estimator from the following location.
   https://www.microsemi.com/document-portal/doc_download/136554-polarfire-power-estimator
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.
- CoreReset_PF IP core is used to control reset signal of the Fabric_Logic_0 and XCVR block. For more information on CoreReset_PF IP core, refer CoreReset_PF Handbook from the Libero catalog.
- OSC_RC200MHz_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 looped back using external SMA to SMA loopback cables.
- 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 by the transceiver block.

2.3.1 Design Implementation

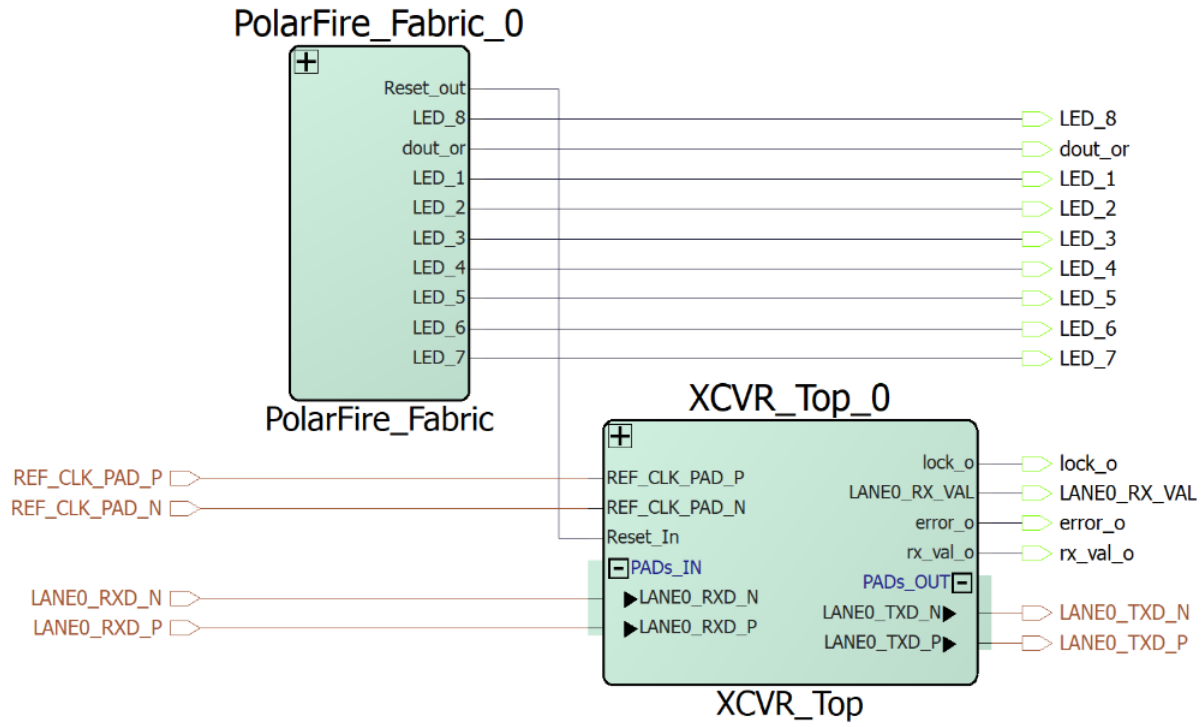
This section describes the top-level SmartDesign canvas and important I/O signals.

The top-level SmartDesign canvas contains the following blocks:

- XCVR_Block, page 6
- PolarFire_Fabric, page 8

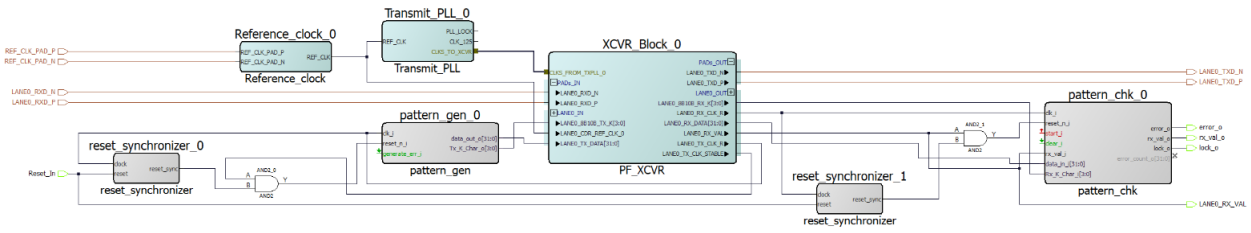
Figure 2, page 4 shows the top-level design component of the PolarFire low-power design.

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



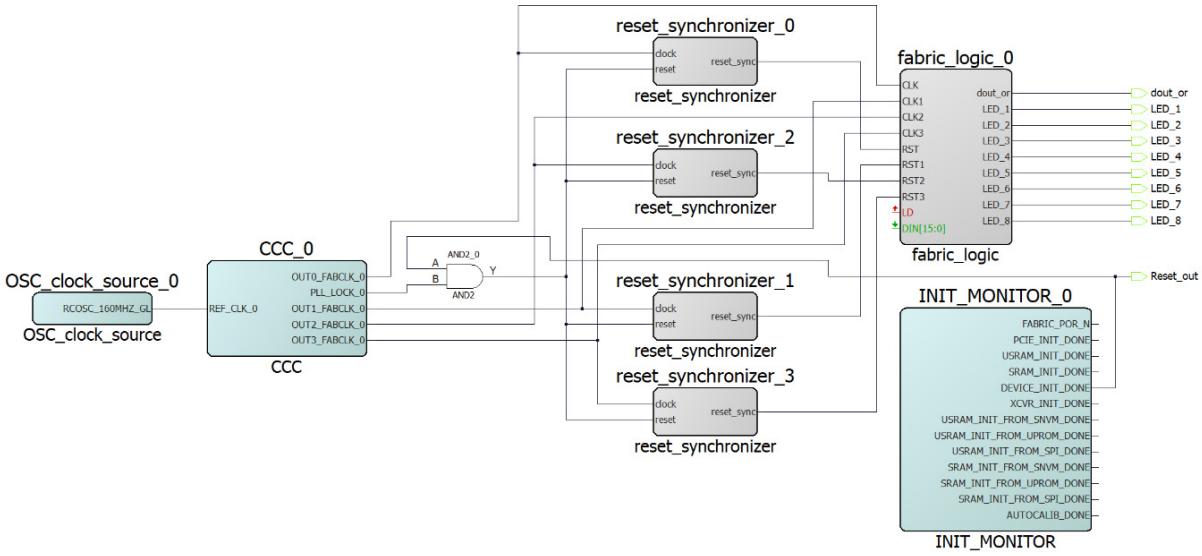
The top component instantiates the XCVR_Block as shown in the Figure 3, page 4.

Figure 3 • **XCVR_Block SmartDesign Component**



The PolarFire_Fabric component is shown in the Figure 4, page 5.

Figure 4 • **PolarFire_Fabric SmartDesign Component**



The following table lists the important I/O signals of the design.

Table 3 • **I/O Signals**

Signals	Type	Description
PolarFire_Fabric		
LED_1, LED_2, LED_3, LED_4, LED_5, LED_6, LED_7, LED_8	Output	Mapped to LEDs 4, 5, 6, 7, 8, 9, 10, and 11 on the board
PF_XCVR		
REF_CLK_PAD_P and REF_CLK_PAD_N		Indicates that the differential reference clock is generated from the on-board 156.25 MHz oscillator
PADs_IN	Input	Transceiver LANE0_RXD_P and LANE0_RXD_N (J37 and J38) on the board
PADs_OUT	Output	Indicates that LANE0_TXD_P and LANE0_TXD_N (J41 and J42), which are looped back to J37 and J38 using SMA cables
error_o	Output	Asserted when RX and TX data do not match
lock_o	Output	Asserted when RX and TX data match
rx_val_o	Output	Indicates that the XCVR has received and validated the 8b/10b idle frames

2.3.2 IP Configuration

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

- XCVR_Block, page 6
- 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:

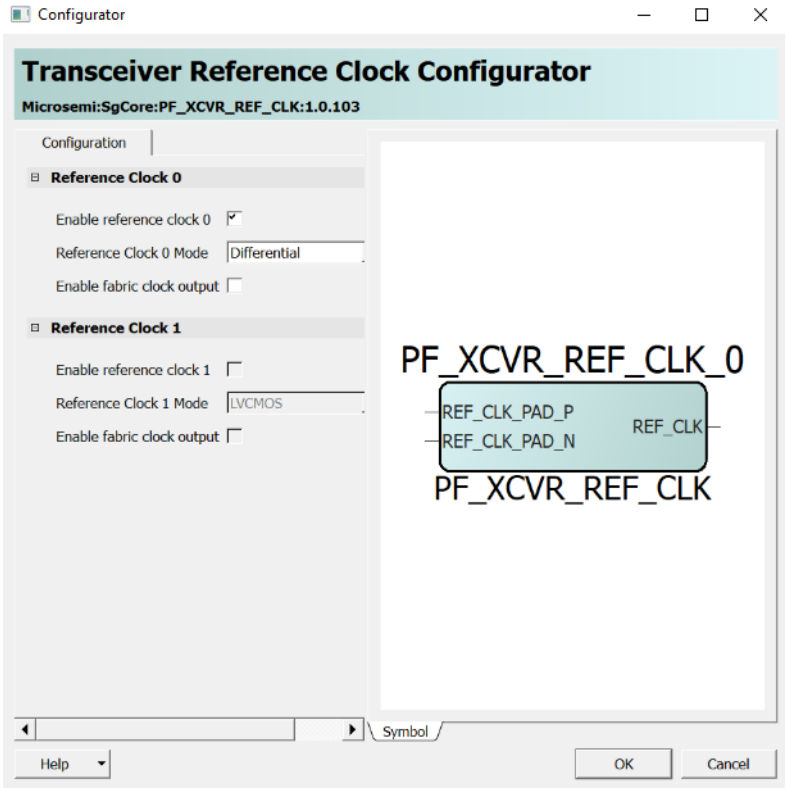
- PF_XCVR_REF_CLK_0, page 6
- PF_TX_PLL_0, page 7
- pattern_gen_0, page 7
- PF_XCVR_0, page 7
- pattern_chk_0, page 8

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.

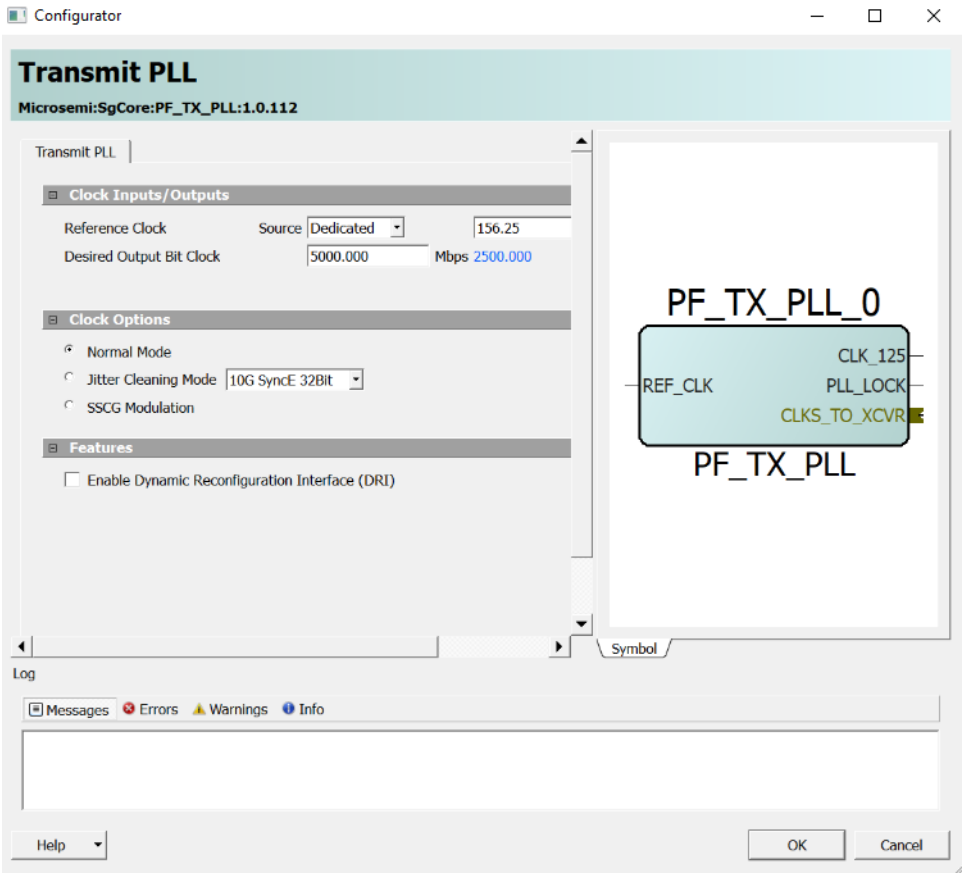
Figure 5 • **PF_XCVR_REF_CLK Configurator**



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.

Figure 6 • **PF_TX_PLL Configurator**



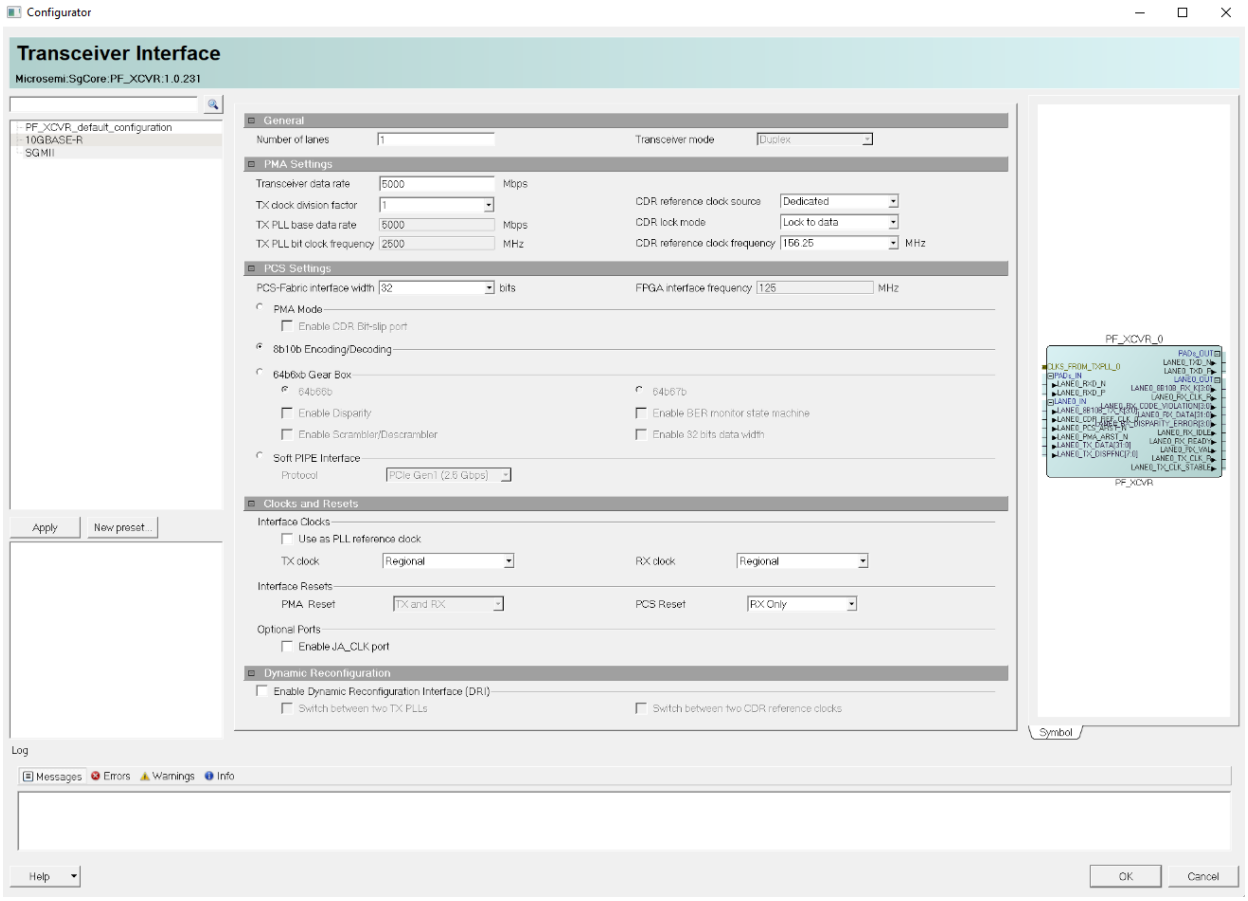
2.3.2.1.3 pattern_gen_0

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 156.25 MHz reference clock as shown Figure 7, page 8.

Figure 7 • **PF_XCVR Configurator**



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 looped back using external SMA to SMA loopback cables. 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 J8 for probing.

2.3.2.2 PolarFire_Fabric

The PolarFire_standby block contains:

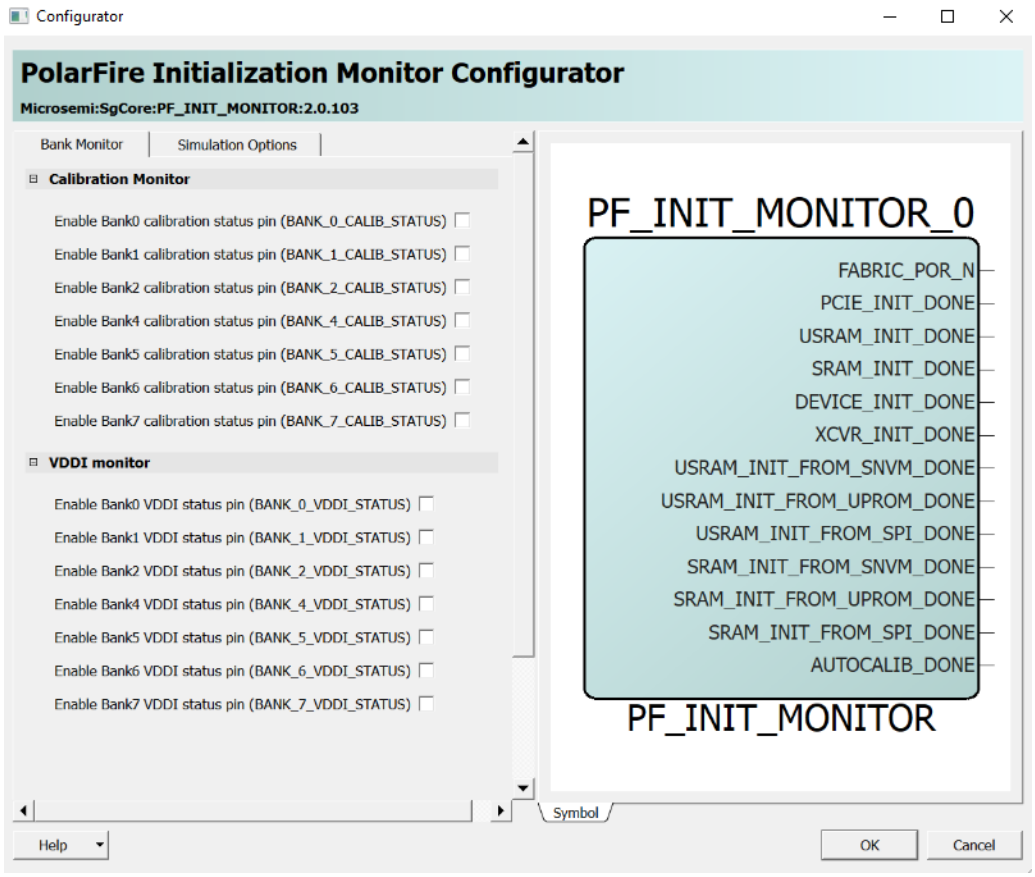
- PF_INIT_MONITOR_0, page 9
- OSC_clock_source_0, page 9
- PF_CCC_0, page 10
- Fabric_Logic_0, page 10

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**



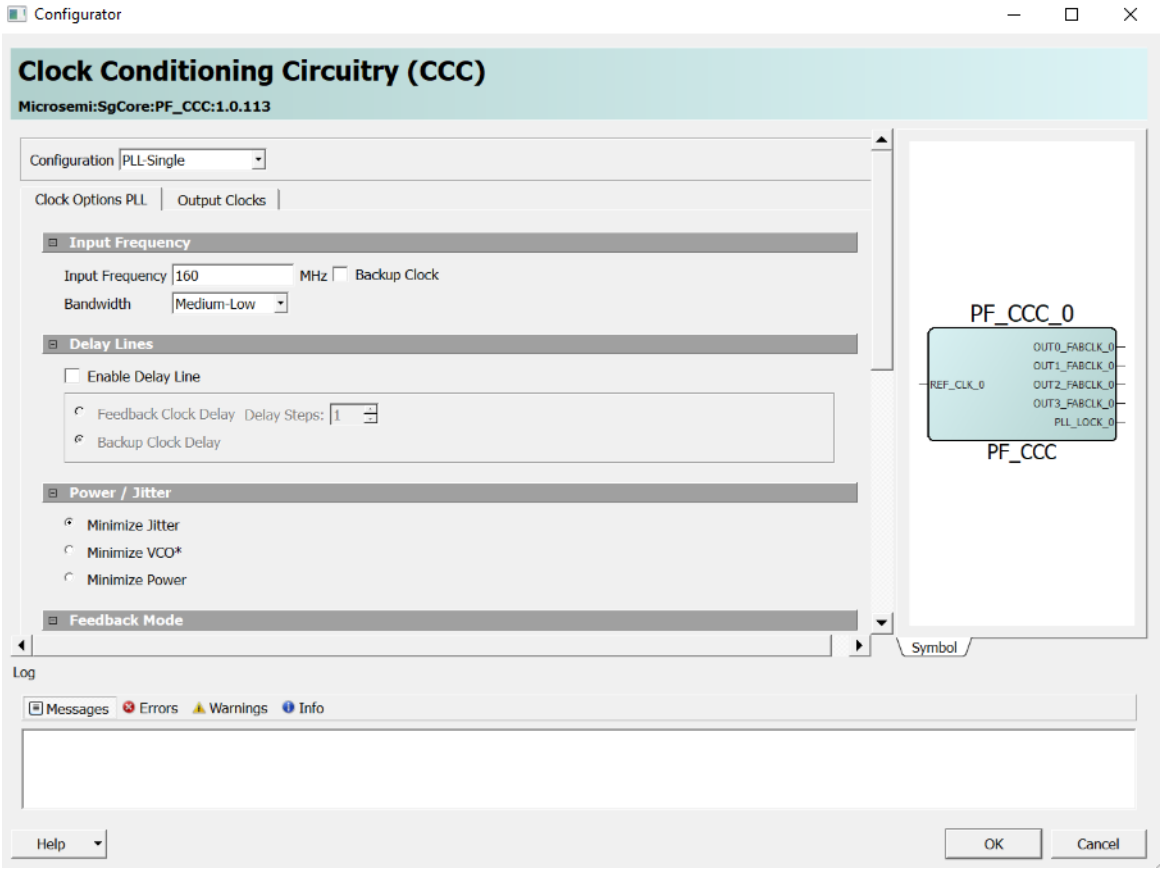
2.3.2.2.2 OSC_clock_source_0

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.

Figure 9 • **PF_CCC Configurator**



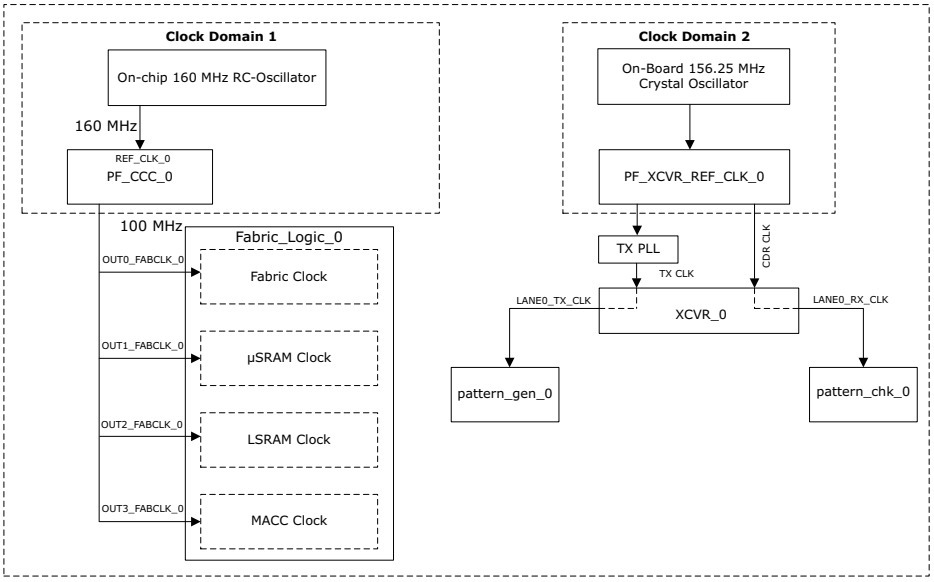
2.3.2.2.4 Fabric_Logic_0

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**



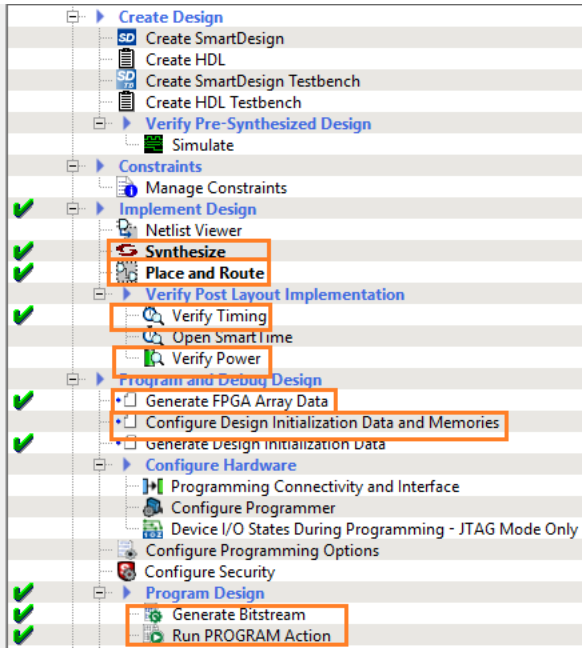
3 Libero Design Flow

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 4 • **Resource Utilization**

Type	Used	Total	Percentage
4LUT	185120	299544	61.80
DFF	205887	299544	68.73
I/O Register	0	1536	0.00
Logic Element	214413	299544	71.58
µSRAM	500	2772	18.04
LSRAM	500	952	52.52
MACC	500	924	54.11

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**

Design:	Top_Level
Family:	PolarFire
Die:	MPF300TS_ES
Package:	FCG1152
Temperature Range:	EXT
Voltage Range:	EXT
Operating Conditions:	Typical
Operating Mode:	Active
Process:	Typical
Data Source:	Advanced

Power Summary

	Power (mW)	Percentage
Total Power	1715.933	100.0%
Static Power	100.538	5.9%
Dynamic Power	1615.396	94.1%

The following figure shows the transceiver power.

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

Breakdown by Type

	Power (mW)	Percentage
Type Net	319.781	18.6%
Type Gate	507.413	29.6%
Type I/O	6.252	0.4%
Type Memory	470.164	27.4%
Type Core Static	100.088	5.8%
Type Other Rails Static	0.450	0.0%
Type Built-in Blocks	259.791	15.1%
Type SERDES	51.994	3.0%

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 5 • **Jumper Settings**

Jumper	Description	Default
J18, J19, J20, J21, J22	Close pin 2 and 3 for programming the PolarFire FPGA through FTDI	Closed
J28	Close pin 2 and 3 for programming through the on-board FlashPro5	Open
J26	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J27	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J4	Close pin 1 and 2 for manual power switching using SW3	Closed
J12	Close pin 3 and 4 for 2.5 V	Closed
J46	Close pin 1 and 2 for setting the Reference Clock to 125 MHz on board oscillator	Closed

2. Connect the power supply cable to the **J9** connector on the board.
3. Connect the USB cable from the **Host PC** to the **J5** (FTDI port) on the board.
4. Power on the board using the **SW3** slide switch.
5. Connect TXN to RXN and TXP to RXP using the 2 SMA-to-SMA cables as shown in (board setup).
6. Double-click **Run PROGRAM Action** from the **Libero > Design Flow** tab.
7. 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.

4 Programming the Device Using FlashPro

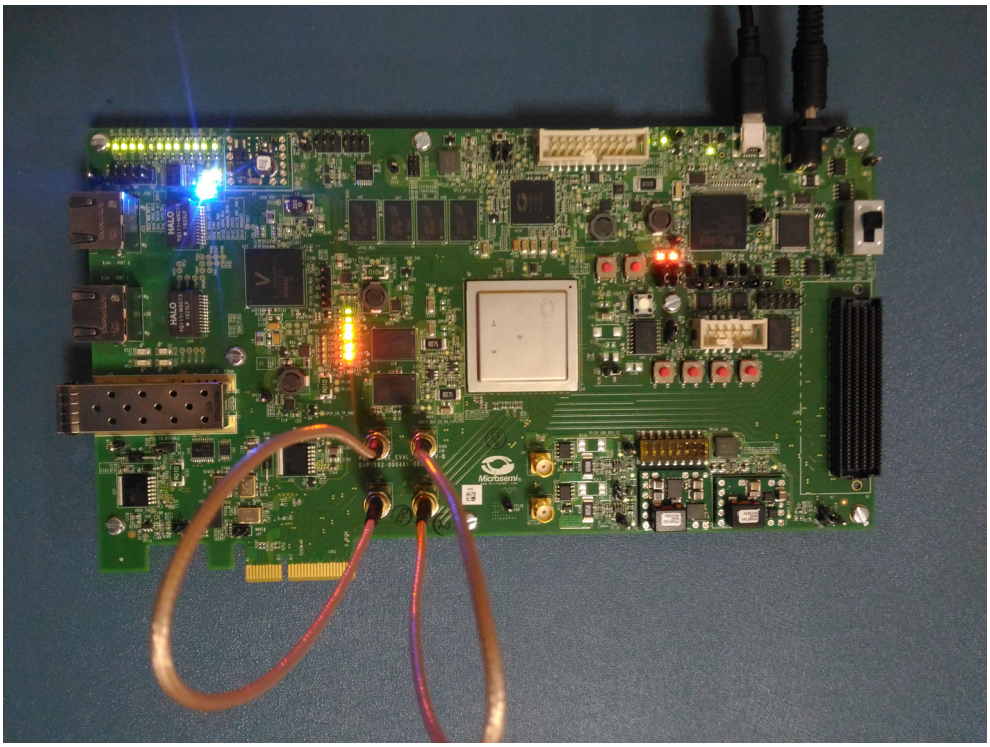
This chapter describes how to program the PolarFire device with `.stp` programming file using Flashpro. The `.stp` file is available at:

`mpf_dg0774_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 **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 **SW3** slide switch.
5. Connect **TXN** to **RXN** and **TXP** to **RXP** using the 2 SMA-to-SMA cables as shown in the following figure. The following figure shows the board setup.

Figure 14 • **Board Setup**

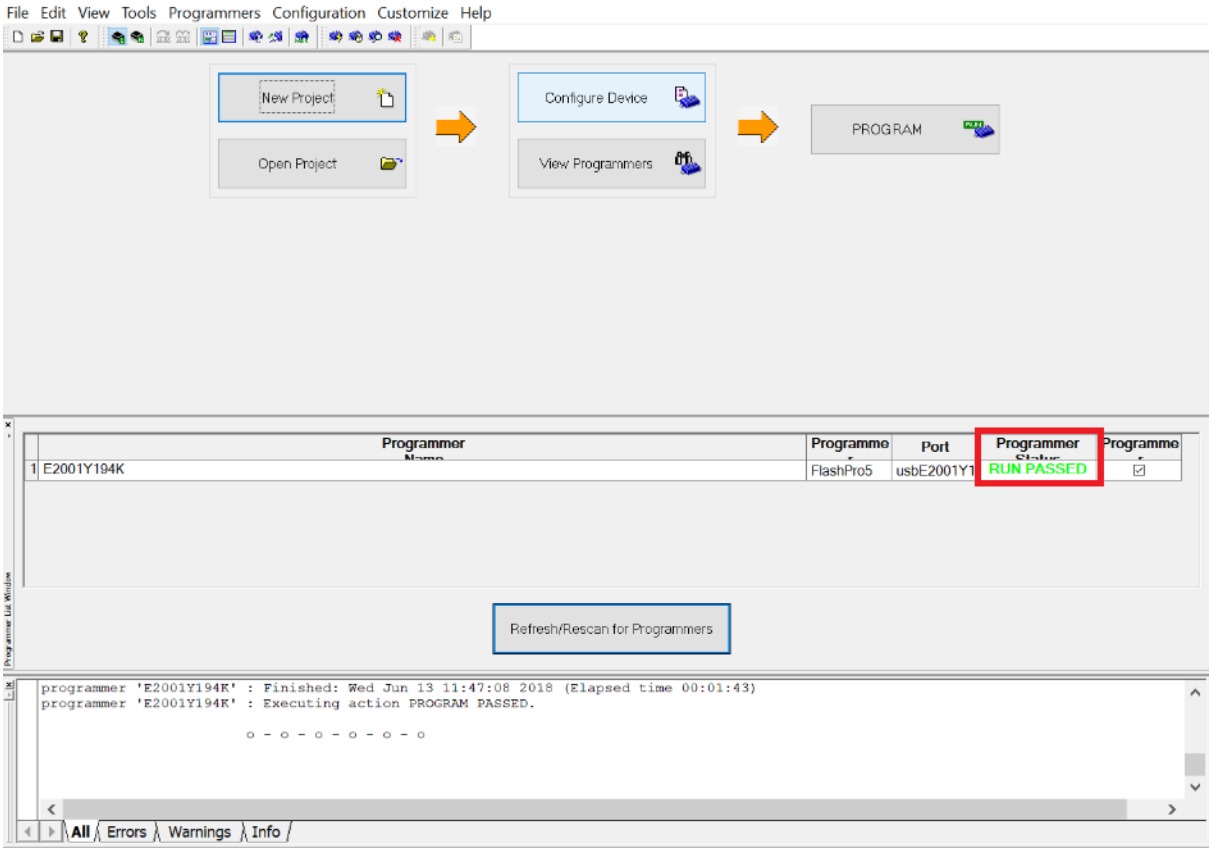


6. On the host PC, start the FlashPro software.
7. Click **New Project** to create a new project.
   In the **New Project** window, do the following, and click **OK**:
8. Enter a project name.
9. Select **Single device** as the programming mode.
10. Click **Configure Device**.
11. Click **Browse**, and select the `PolarFire_Low_Power_Demo.stp` file from the **Load Programming File** window.
12. 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.

Note: When the device is programmed, the LEDs 4,5,6,7,8,9,10,11 blink.

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**



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*.

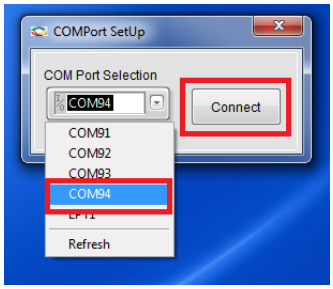
Prerequisites for the procedure:

1. The PolarFire Evaluation board is connected.
2. The PolarFire FPGA is programmed with the Low Power design.

Following are the steps:

1. The LEDs {4, 5}, {6, 7},{8, 9}, and {10,11} 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.

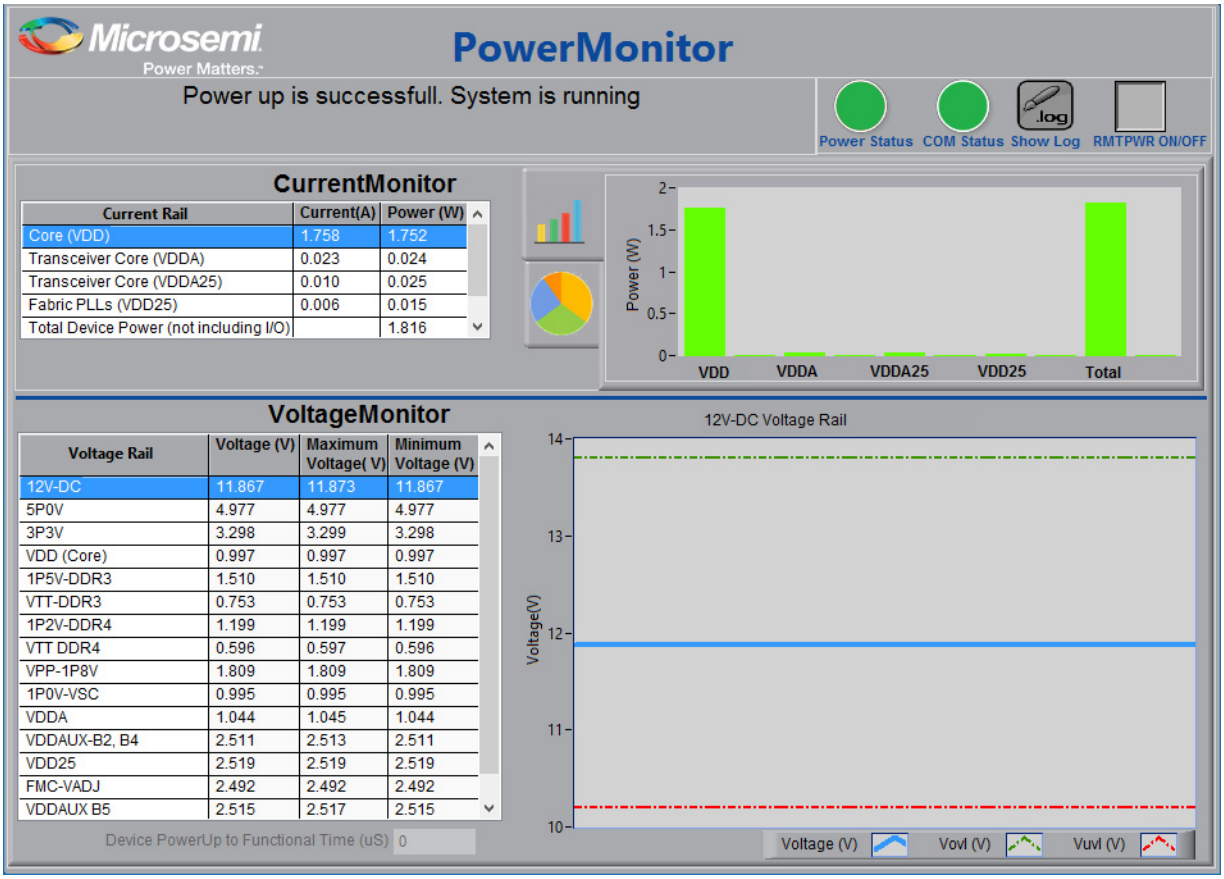
Figure 16 • **COM Port Setup**



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.

6 Power Correlation

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 6 • **Power Correlation**

Tool	Transceiver Power(W)	Total Power (W)
Power Estimator	0.079	1.853
SmartPower	0.051	1.715
Power Monitor (real-time onboard power)	0.049	1.816

For this demo design, ~11% difference is seen between the SmartPower total power estimate and PowerMonitor total power.

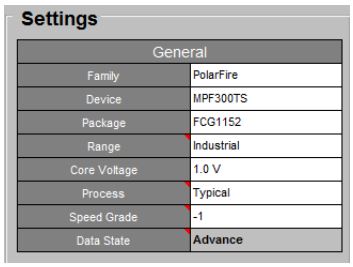
7 Appendix: Power Estimator

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.
   https://www.microsemi.com/document-portal/doc_download/136554-polarfire-power-estimator
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.

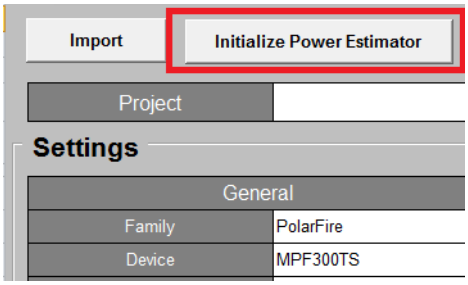
Figure 18 • **Device Settings**



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**.

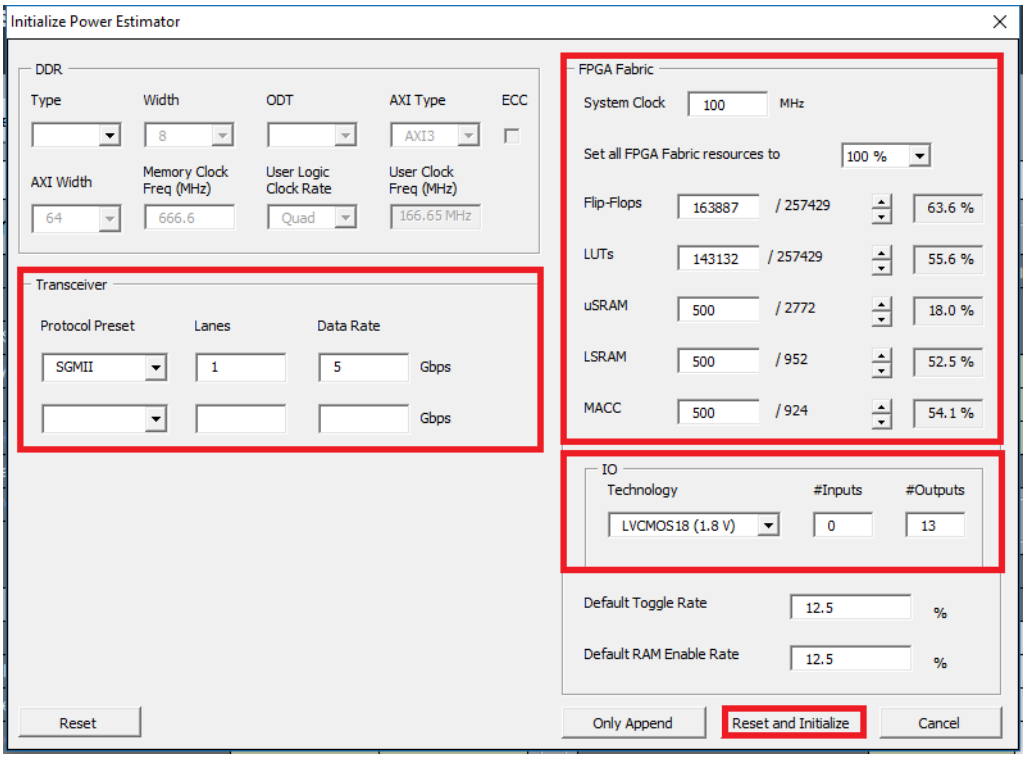
Figure 19 • **Initialize Power Estimator**



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**

				PMA						PCS		
Name	Protocol Preset	Number of Lanes	Operational Mode	Data Rate (Gbps)	PLL Used	DFE Enable	Eye Monitor Enable	CTLE Drive	TX Amplitude (mV)	Mode	Width	Hard PCIe
SGMII/Init_PE_1	SGMII	1	Duplex	5	Q1_TXPLL0	No	No	3	771.3	8b/10b	32	No
			Duplex		Q0_TXPLL0	No	No	3	270	8b/10b	32	No

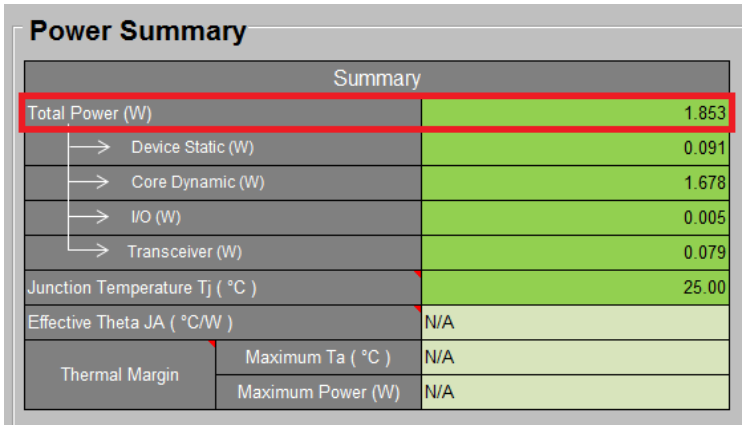
7. Set the **Reference frequency, Output 0, Output 1, Output 2, Output 3** frequencies, and **Mode** in the PLL & DLL worksheet, as shown in the following figure.

Figure 22 • **Frequency Settings in PLL and DLL Worksheet**

PLL Power								
Name	Reference Clock Frequency (MHz)	Output 0 Frequency (MHz)	Output 1 Frequency (MHz)	Output 2 Frequency (MHz)	Output 3 Frequency (MHz)	Mode	VDD Power (W)	VDD25 Power (W)
	160	100	100	100	100	Low Jitter	0.008	0.008
						Low Power	0.000	0.000
						Low Power	0.000	0.000

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**



9. Close the **Power Estimator** wizard**.**

8 Appendix: References

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.*