TU0844
Tutorial
Libero SoC PolarFire v2021.2 Design Flow

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

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 3.0 1

1.2 Revision 2.0 1

1.3 Revision 1.0 1

2 Introduction 2

2.1 Prerequisites 2

3 Running the Tutorial 3

3.1 Step 1—Creating a Libero SoC PolarFire Project 3

3.2 Step 2—Creating Your Design 6

3.3 Step 3—Synthesis and Pin Assignment 9

3.4 Step 4—Programming the Device 13

3.5 Step 5—Running the Design 17

3.6 Step 6—Design Iterations in the Libero SoC PolarFire 18

4 References 19

Figures

Figure 1 Libero SoC PolarFire Project Manager 3
Figure 2 New Project Window 4
Figure 3 Project Details 4
Figure 4 Device Selection 5
Figure 5 Device Settings 6
Figure 6 Design Creation 6
Figure 7 Create New SmartDesign 7
Figure 8 Catalog Tab 7
Figure 9 AND2 Gate 7
Figure 10 AND2 Connection 8
Figure 11 Success Message in Libero SoC Log Tab 8
Figure 12 Design Hierarchy 8
Figure 13 Synthesize in Design Flow 9
Figure 14 Libero SoC Constraints Manager 10
Figure 15 I/O Editor 10
Figure 16 I/O Standard and Pin Number Assignments 11
Figure 17 Push-Button and Pin Connections 11
Figure 18 PolarFire Evaluation Kit User Defined LEDs 12
Figure 19 Running the Design 12
Figure 20 PolarFire Evaluation Kit Board 13
Figure 21 Run Program Action 15
Figure 22 Reports Tab 16
Figure 23 Design Flow Green Tick Marks 16
Figure 24 Design Iterations 18
Figure 25 Ports Inversion 18

Tables

Table 1 Prerequisites for the Tutorial . . . 2
Table 2 PolarFire Evaluation Kit - Push-Button Switches . . . 11
Table 3 PolarFire Evaluation Kit - User LEDs . . . 12
Table 4 Jumper Settings . . . 14

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 3.0

The following is a summary of the changes in revision 3.0 of this document.

- Updated the document for Libero SoC v2021.2.
- Updated Figure 1, page 3.

1.2 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- Updated Table 1, page 2
- Updated section Step 1—Creating a Libero SoC PolarFire Project, page 3
- Updated section Step 3—Synthesis and Pin Assignment, page 9
- Replaced Figure 17, page 11, Figure 18, page 12, and Figure 20, page 13

1.3 Revision 1.0

Revision 1.0 was published in September 2018. It was the first publication of this document.

2 Introduction

This tutorial demonstrates how to implement basic PolarFire® Field Programmable Gate Array (FPGA) designs leveraging the Libero® System-on-Chip (SoC) PolarFire design tool. Each section builds complexity and introduces new features on top of the previous section. After completing this tutorial, you will be familiar with the following steps.

- Step 1—Creating a Libero SoC PolarFire Project, page 3
- Step 2—Creating Your Design, page 6
- Step 3—Synthesis and Pin Assignment, page 9
- Step 4—Programming the Device, page 13
- Step 5—Running the Design, page 17
- Step 6—Design Iterations in the Libero SoC PolarFire, page 18

2.1 Prerequisites

The following table lists the prerequisites for the tutorial.

Table 1 • **Prerequisites for the Tutorial**

Type	Details
Software requirements	Libero SoC PolarFire v2021.2
Hardware requirements	PolarFire Evaluation Kit (MPF300-EVAL-KIT)
Source files	Not required. The design will be created from scratch

3 Running the Tutorial

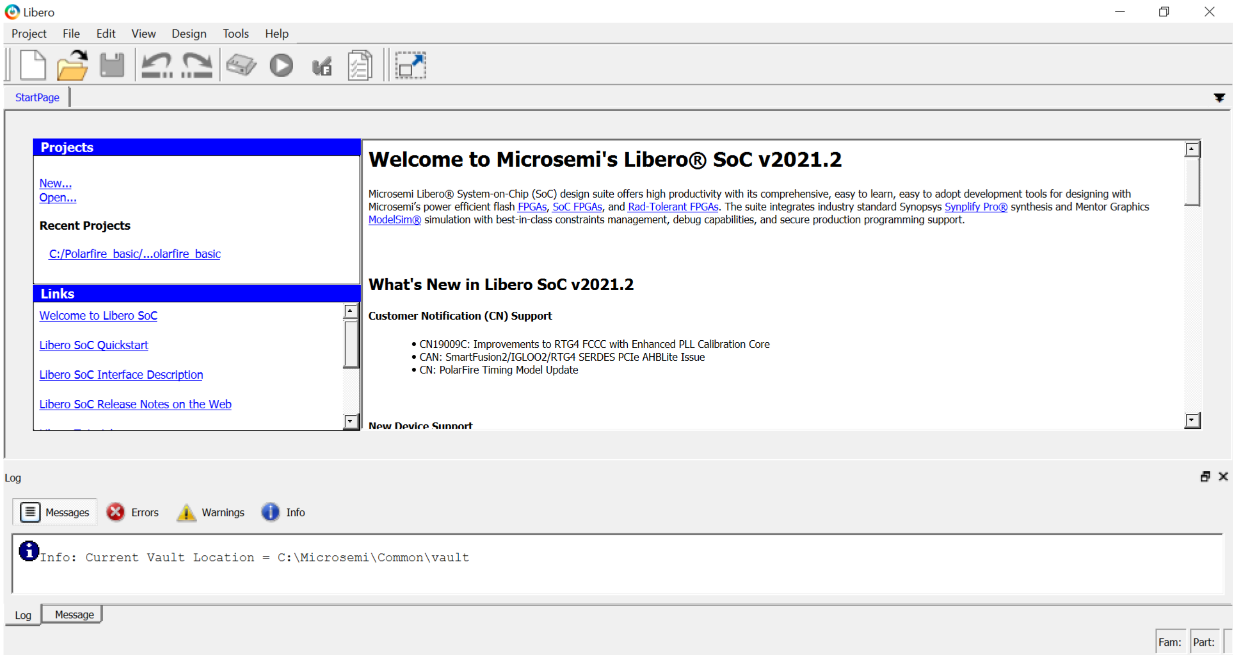
The following sections describe the step-by-step procedure to be followed to run this tutorial.

3.1 Step 1—Creating a Libero SoC PolarFire Project

In this step, you create a Libero SoC project.

1. Click **Start > Programs > Microsemi Libero SoC PolarFire v2021.2 > Libero SoC PolarFire v2021.2**, or click the shortcut on your desktop. The Libero SoC PolarFire Project Manager will open, as shown in the following figure.

Figure 1 • **Libero SoC PolarFire Project Manager**



Welcome to Microsemi's Libero® SoC v2021.2

Microsemi Libero® System-on-Chip (SoC) design suite offers high productivity with its comprehensive, easy to learn, easy to adopt development tools for designing with Microsemi's power efficient flash FPGAs, SoC FPGAs, and Rad-Tolerant FPGAs. The suite integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim® simulation with best-in-class constraints management, debug capabilities, and secure production programming support.

What's New in Libero SoC v2021.2

Customer Notification (CN) Support

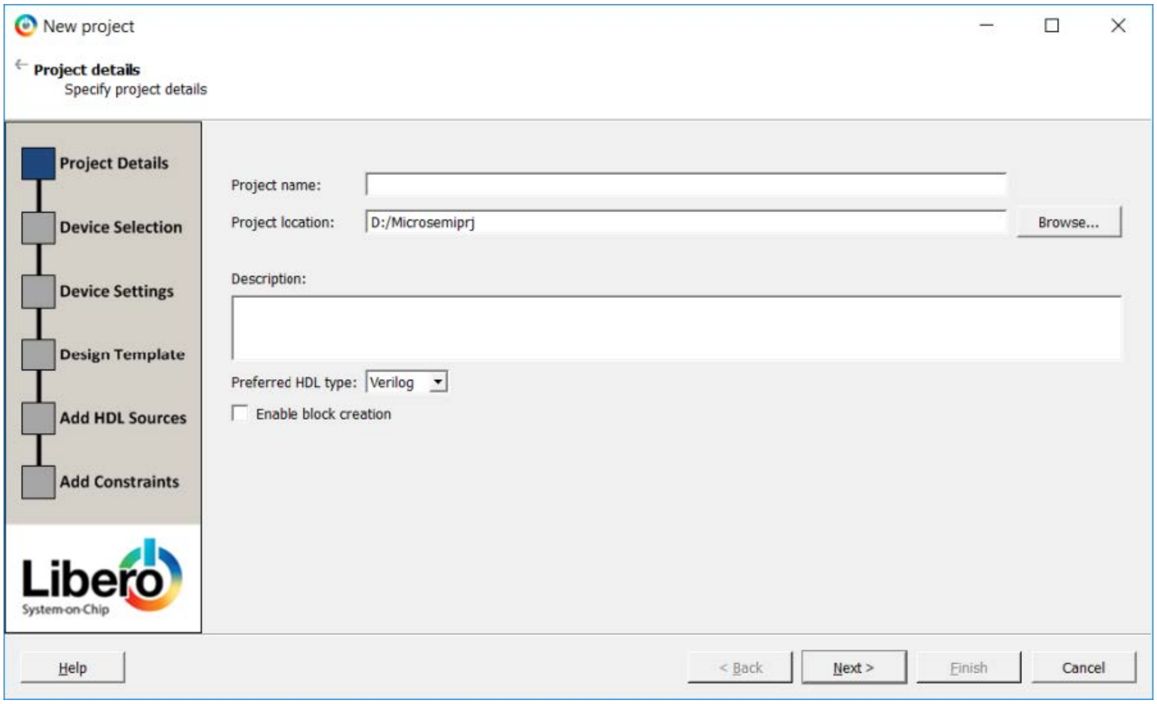
- CN19009C: Improvements to RTG4 FCCC with Enhanced PLL Calibration Core
- CAN: SmartFusion2/IGLOO2/RTG4 SERDES PCIe AHBLite Issue
- CN: PolarFire Timing Model Update

New Device Support

```
Info: Current Vault Location = C:\Microsemi\Common\vault
```

2. Create a new project by selecting **New** on the **Start Page** tab (shown in the preceding figure), or by clicking **Project > New Project** from the Libero SoC menu. The **New Project** wizard will open.

Figure 2 • **New Project Window**



Project name:

Project location: D:/Microsemiprj

Description:

Preferred HDL type: Verilog

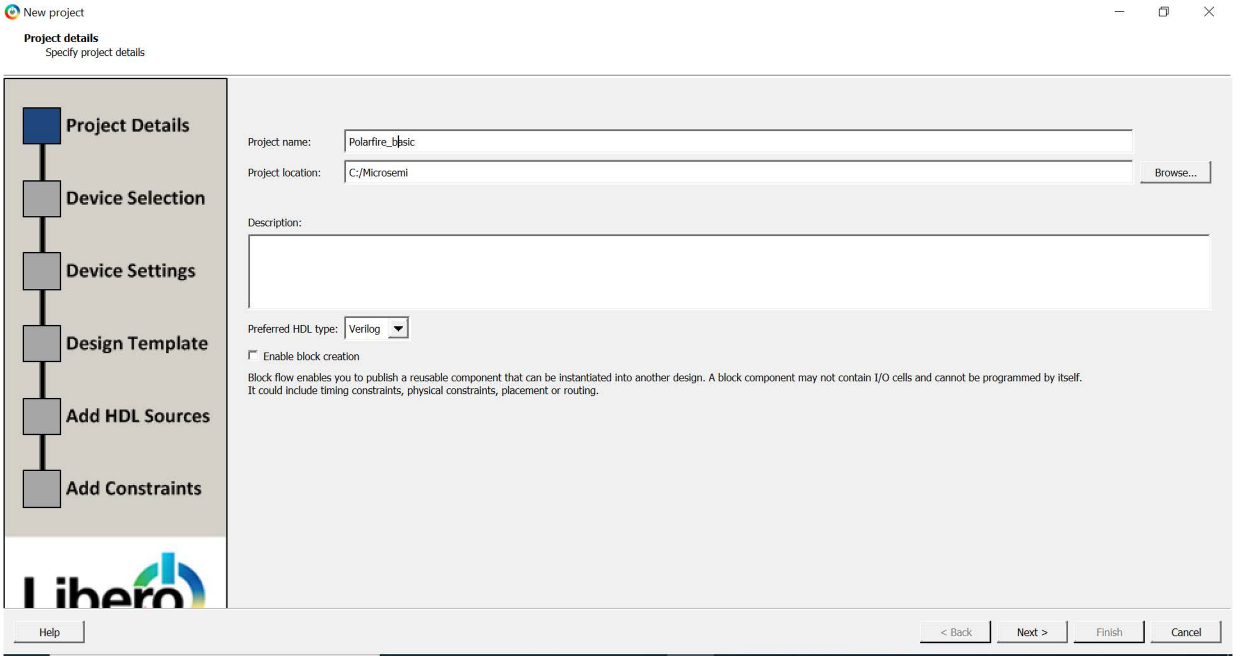
☐ Enable block creation

3. Enter the information given in the following list, in the **Project Details** page of the **New Project** dialog box and click **Next**.

- Project name: PolarFire_basic
- Project location: `<C: or D:>/Microsemiprj` (or the path to where the files were extracted).
- Preferred HDL type: Select either **VHDL** or **Verilog** based on your HDL preferences.

Note: Project name is case sensitive for Verilog design.

Figure 3 • **Project Details**



Project name: Polarfire_basic

Project location: C:/Microsemi

Description:

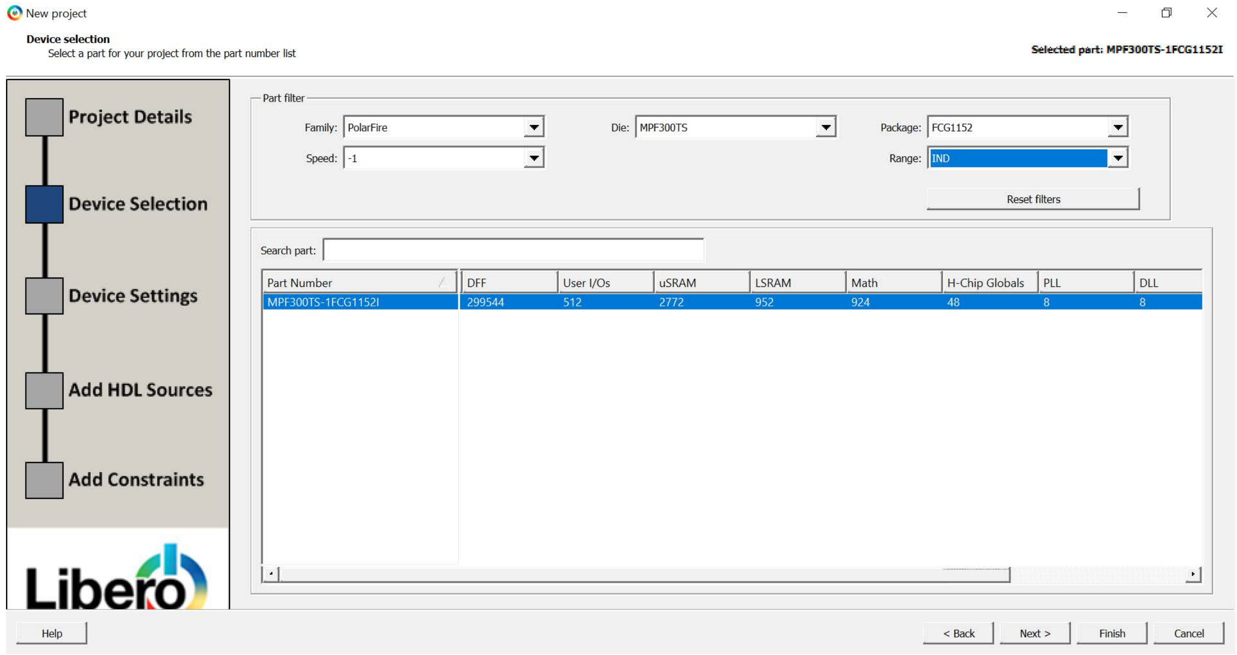
Preferred HDL type: Verilog

☐ Enable block creation

Block flow enables you to publish a reusable component that can be instantiated into another design. A block component may not contain I/O cells and cannot be programmed by itself. It could include timing constraints, physical constraints, placement or routing.

4. Enter the following details in the **Device Selection** page of the **New Project** dialog box and click **Next**.
- Family: PolarFire
- Die: MPF300TS
- Package: FCG1152
- Speed: –1
- Range: IND

Figure 4 • **Device Selection**



New project

Device selection

Select a part for your project from the part number list

Selected part: MPF300TS-1FCG1152I

Part filter

Family: PolarFire

Die: MPF300TS

Package: FCG1152

Speed: -1

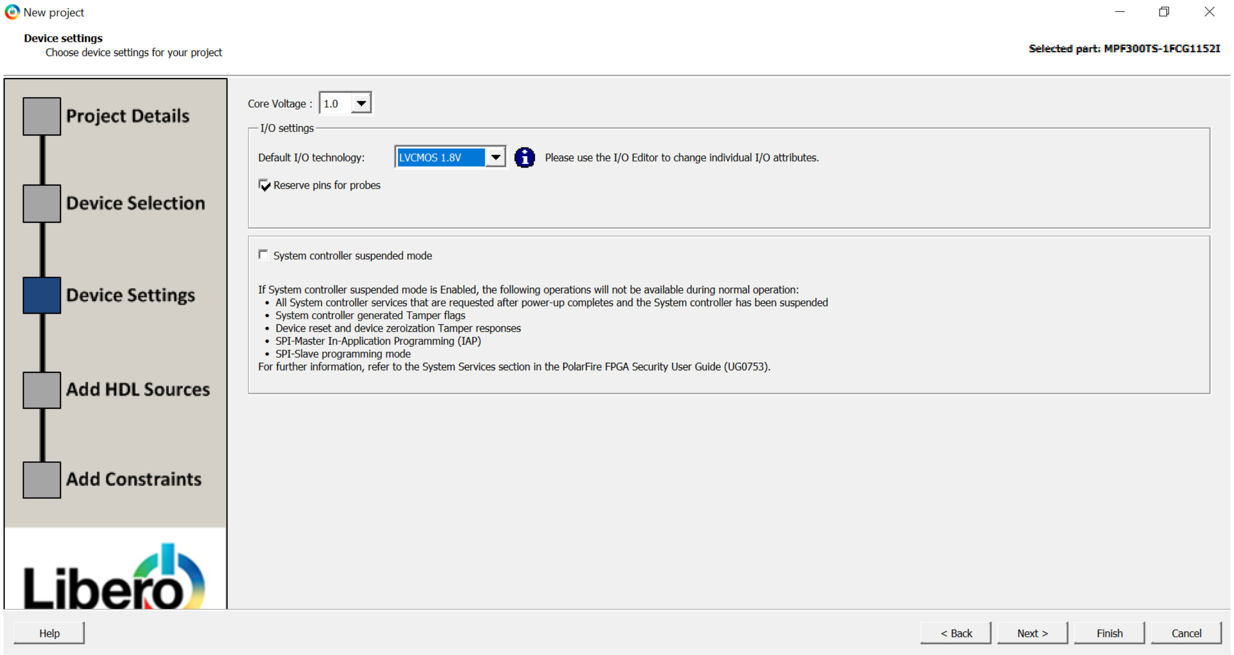
Range: IND

Search part:

Part Number	DFF	User I/Os	uSRAM	LSRAM	Math	H-Chip Globals	PLL	DLL
MPF300TS-1FCG1152I	299544	512	2772	952	924	48	8	8

5. Enter the following details in the **Device settings** page of the **New Project** dialog box and click **Finish**.
- Core Voltage: 1.0 (default)
- I/O settings
  - Default I/O technology: LVCMOS 1.8 V
  - Reserve pins for probes: Checked (default)
- System controller suspended mode: Un-checked (default)

Figure 5 • **Device Settings**



New project

Device settings
Choose device settings for your project

Selected part: MPF300TS-1FCG1152I

Core Voltage : 1.0

I/O settings

Default I/O technology: LVCMOS 1.8V

Please use the I/O Editor to change individual I/O attributes.

☑ Reserve pins for probes

□ System controller suspended mode

If System controller suspended mode is Enabled, the following operations will not be available during normal operation:

- All System controller services that are requested after power-up completes and the System controller has been suspended
- System controller generated Tamper flags
- Device reset and device zeroization Tamper responses
- SPI-Master In-Application Programming (IAP)
- SPI-Slave programming mode

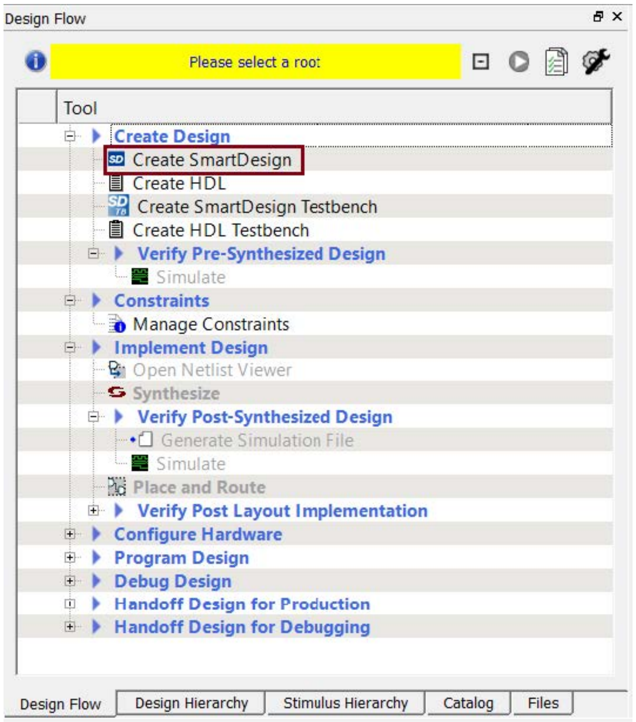
For further information, refer to the System Services section in the PolarFire FPGA Security User Guide (UG0753).

3.2 Step 2—Creating Your Design

The following steps describe how to create the design.

1. The **Design Flow** window is used to manage most of the flow. This window has multiple tabs. You can switch between these tabs by clicking the tab names at the bottom, as shown in the following figure.

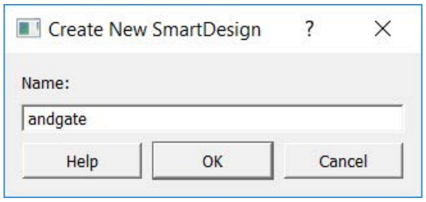
Figure 6 • **Design Creation**



Please select a roo:

2. In the **Design Flow** window, double click **Create SmartDesign** and name your design “**andgate**,” then click **OK**. This opens the **SmartDesign Canvas**.

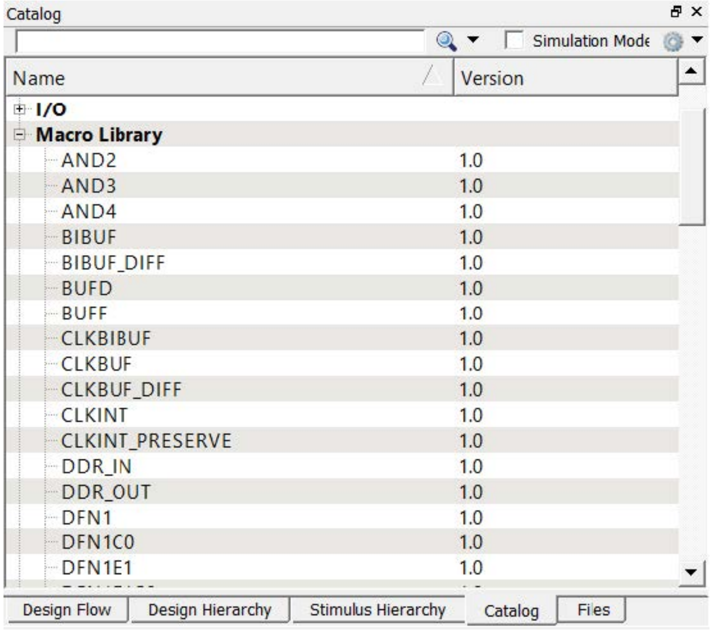
Figure 7 • **Create New SmartDesign**



Name: andgate

3. Select the **Catalog** tab in the **Design Flow** window and expand the **Macro Library** section, as shown in the following figure.

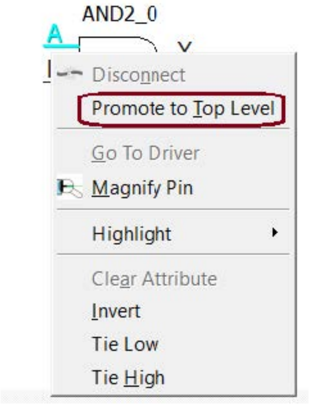
Figure 8 • **Catalog Tab**



Name	Version
I/O	
Macro Library	
AND2	1.0
AND3	1.0
AND4	1.0
BIBUF	1.0
BIBUF_DIFF	1.0
BUFD	1.0
BUFF	1.0
CLKBIBUF	1.0
CLKBUF	1.0
CLKBUF_DIFF	1.0
CLKINT	1.0
CLKINT_PRESERVE	1.0
DDR_IN	1.0
DDR_OUT	1.0
DFN1	1.0
DFN1C0	1.0
DFN1E1	1.0

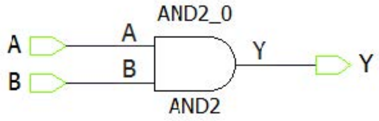
4. Drag the **AND2** gate from the **Catalog** onto the blank **SmartDesign Canvas** on the right.
5. Use the mouse to select **A** and then right click to bring up the menu. From the menu, select **Promote to Top Level**, as shown in the following figure. This creates an external connection on the device that is connected to a switch. Repeat the same steps for B and Y.

Figure 9 • **AND2 Gate**



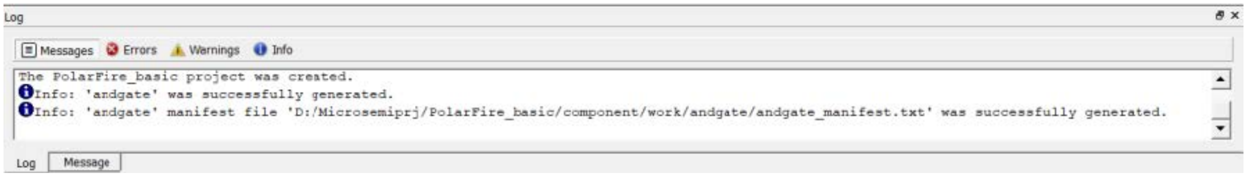
6. When you have completed this step, you should have the view shown in the following figure.

Figure 10 • **AND2 Connection**



7. Click the **Generate Component** button on the top of the canvas to generate the design.
8. The message, **“'andgate' was successfully generated.”** is displayed in the Libero SoC Log tab, as shown in the following figure.

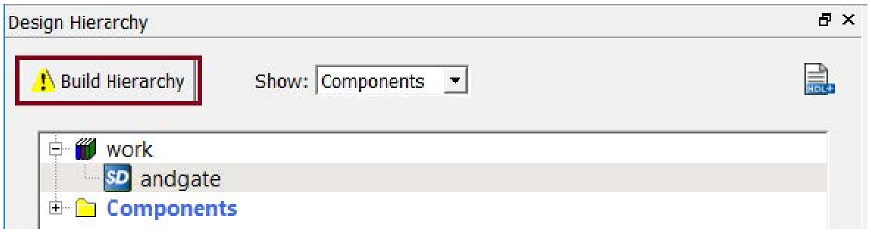
Figure 11 • **Success Message in Libero SoC Log Tab**



```
The PolarFire_basic project was created.
Info: 'andgate' was successfully generated.
Info: 'andgate' manifest file 'D:/Microsemiprj/PolarFire_basic/component/work/andgate/andgate_manifest.txt' was successfully generated.
```

9. Select the **Design Hierarchy** window, set andgate as root and click **Build Hierarchy** as shown in the following figure.

Figure 12 • **Design Hierarchy**



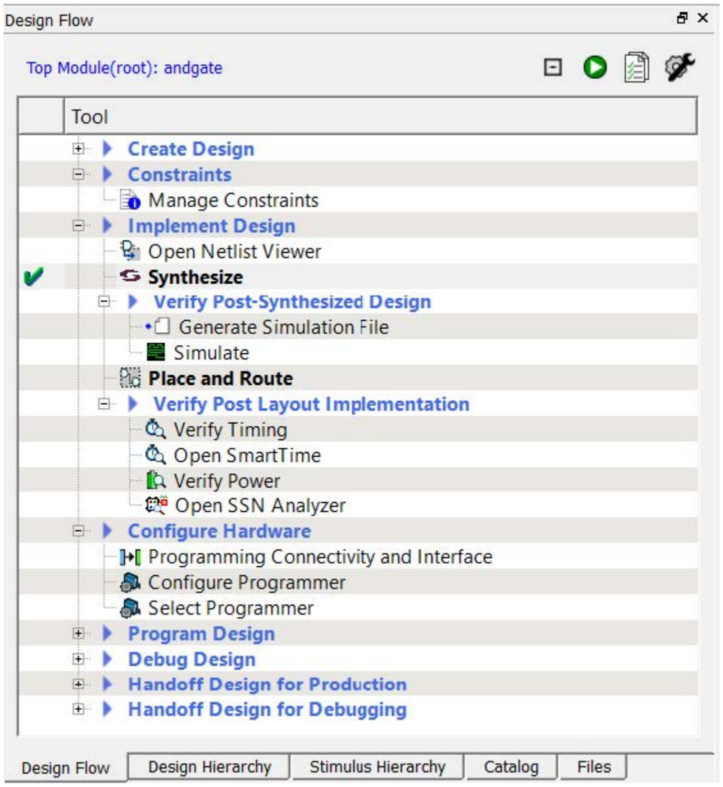
10. **“andgate”** will appear in bold font on the **Design Hierarchy** tab, indicating that it is the root level.

3.3 Step 3—Synthesis and Pin Assignment

The following steps describe the synthesis process and pin assignment.

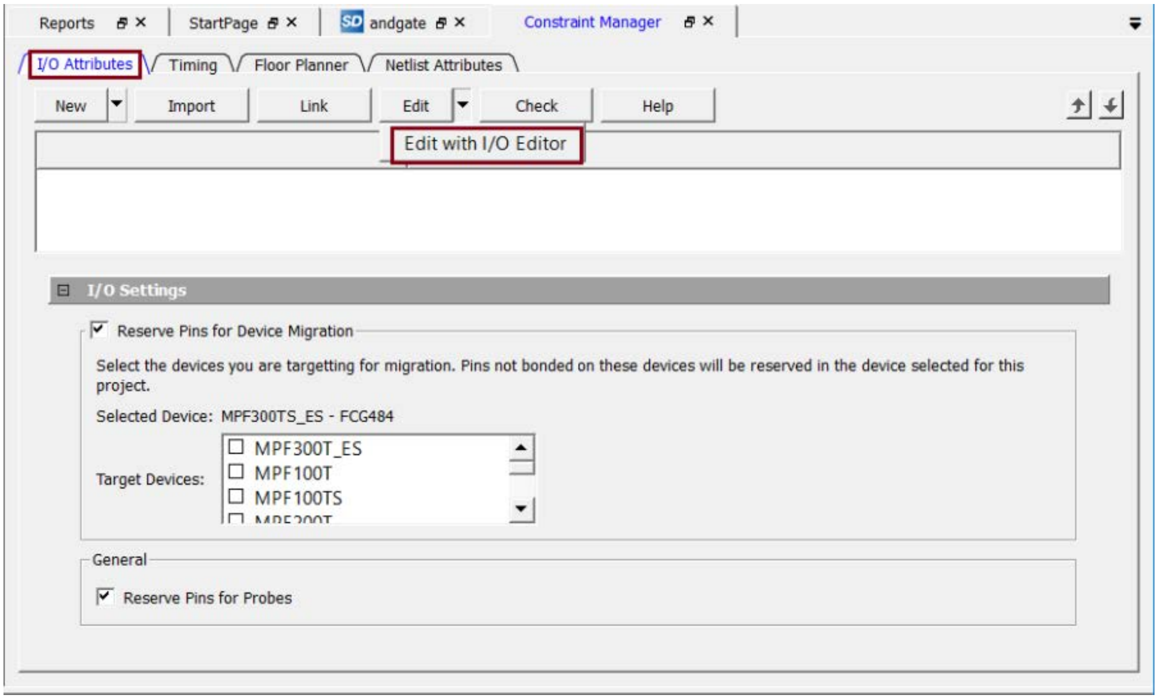
1. Double-click **Synthesize** under **Implement Design** in the **Design Flow** window, as shown in the following figure, to synthesize the design with Synplify Pro. Synplify Pro adds I/O pads to the design. A green check mark will appear in the **Design Flow** window to indicate synthesis was successful.

Figure 13 • **Synthesize in Design Flow**



2. This is a very simple design, hence, the only other step required is, configuration of the I/Os. Double-click **Manage Constraints** in the **Design Flow** window to open the **Libero SoC Constraints Manager**.

Figure 14 • **Libero SoC Constraints Manager**



☑ Reserve Pins for Device Migration

Select the devices you are targetting for migration. Pins not bonded on these devices will be reserved in the device selected for this project.

Selected Device: MPF300TS_ES - FCG484

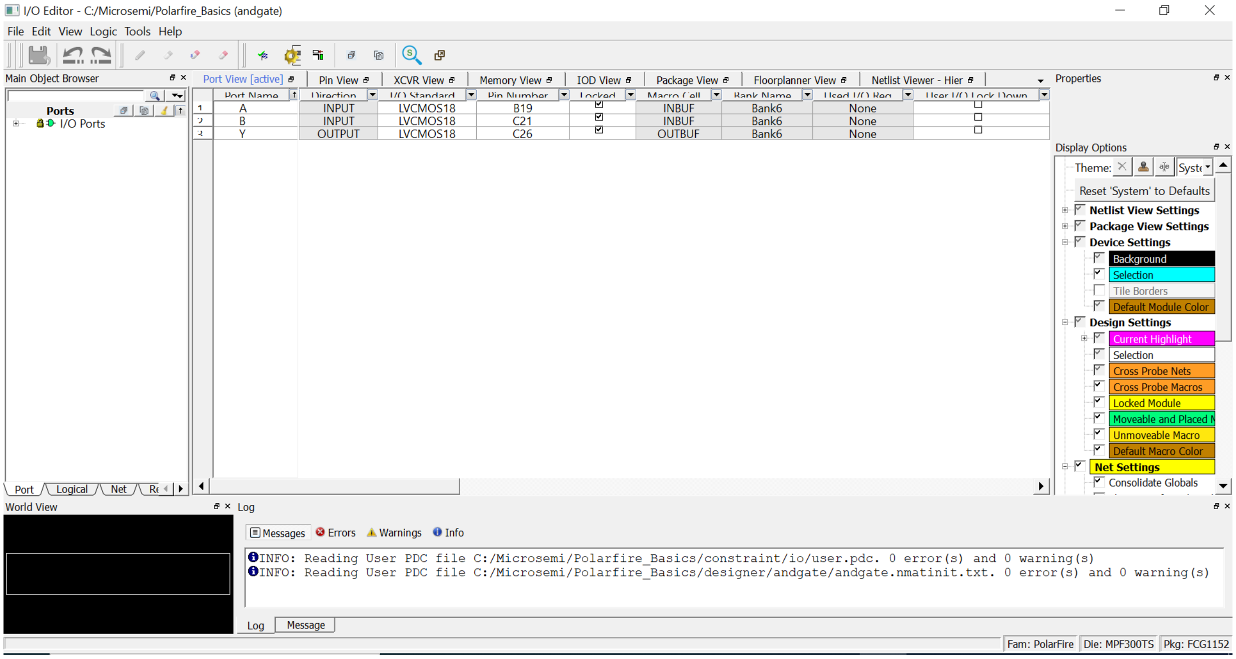
Target Devices:
☐ MPF300T_ES
☐ MPF100T
☐ MPF100TS
☐ MPF200T

General

☑ Reserve Pins for Probes

3. Confirm that the **I/O Attributes** tab is selected. Use the pull-down menu on the **Edit** button to select **Edit with I/O Editor** (highlighted in the preceding figure) to open the **I/O Editor**.
4. Select the **Port View** tab of the **I/O Editor**, as shown in the following figure.

Figure 15 • **I/O Editor**



	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name	Used I/O Reg	User I/O Lock Down
1	A	INPUT	LVCMOS18	B19	☑	INBUF	Bank6	None	☐
2	B	INPUT	LVCMOS18	C21	☑	INBUF	Bank6	None	☐
3	Y	OUTPUT	LVCMOS18	C26	☑	OUTBUF	Bank6	None	☐

```
INFO: Reading User PDC file C:/Microsemi/Polarfire_Basics/constraint/io/user.pdc. 0 error(s) and 0 warning(s)
INFO: Reading User PDC file C:/Microsemi/Polarfire_Basics/designer/andgate/andgate.nmatinit.txt. 0 error(s) and 0 warning(s)
```

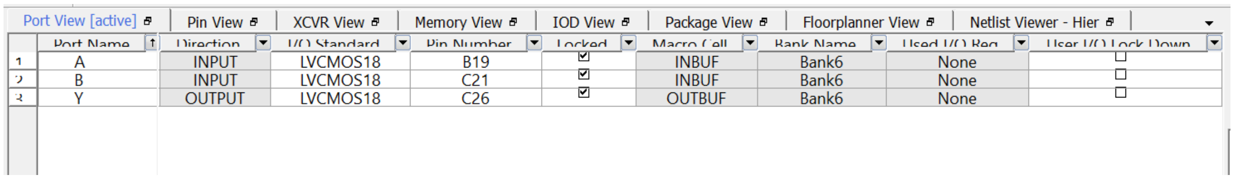
Fam: PolarFire | Die: MPF300TS | Pkg: FCG1152

5. Use this window to set the **I/O Standard** and **Pin Number**. Confirm that LVCMOS18 is the I/O standard for each pin, as shown in the following figure.
6. Click **Unassigned** in the **Pin Number** column to reveal available I/O pins. Make the following assignments. You can either type the pin number in the field or use the pull-down menu to select the pin.

- A—B19
- B—C21
- Y—C26

7. Make sure the connections match the following screenshot, then click on **File > Commit** and close the window by clicking **File > Exit**.

Figure 16 • I/O Standard and Pin Number Assignments



	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name	Used I/O Reg	User I/O Lock Down
1	A	INPUT	LVCMOS18	B19	☑	INBUF	Bank6	None	☐
2	B	INPUT	LVCMOS18	C21	☑	INBUF	Bank6	None	☐
3	Y	OUTPUT	LVCMOS18	C26	☑	OUTBUF	Bank6	None	☐

Push-Button Switches

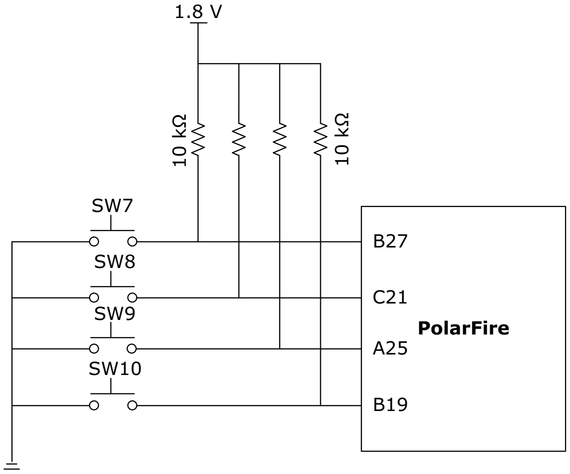
The PolarFire evaluation board comes with four debug push-button switches connected to the PolarFire FPGA. The following table lists the on-board push-button switches.

Table 2 • PolarFire Evaluation Kit - Push-Button Switches

PolarFire Evaluation Board Pin	PolarFire FPGA Pin Number	PolarFire FPGA Pin Name	Bank
SW10	B19	HSIO46PB6	Bank - 6
SW9	C21	HSIO49PB6	Bank - 6
SW8	A25	HSIO68PB6/DQS/CCC_SE_PLL1_OUT0	Bank - 6
SW7	B27	HSIO69NB6	Bank - 6

The following figure shows push-button and pin connections.

Figure 17 • Push-Button and Pin Connections



User LEDs

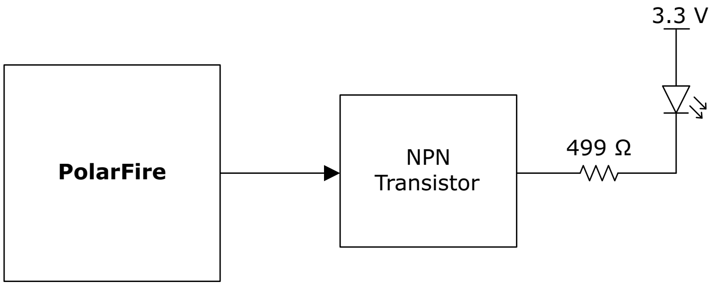
The board provides user access to eight active-High LEDs that are connected to the PolarFire device for debugging applications. The following table lists the onboard user LEDs.

Table 3 • **PolarFire Evaluation Kit - User LEDs**

PolarFire Evaluation Board Pin	PolarFire FPGA Pin Number	Number PolarFire Pin Name	Bank
LED7	D25	HSIO70PB6/CCC_SE_PLL1_OUT1	Bank - 6
LED6	C26	HSIO70NB6	Bank - 6
LED5	B26	HSIO71NB6	Bank - 6
LED4	F22	HSIO64NB6	Bank - 6
LED11	H21	HSIO36NB6	Bank - 6
LED10	H22	HSIO60NB6	Bank - 6
LED9	F23	HSIO62PB6/DQS/CCC_SE_PLL0_OUT0	Bank - 6
LED8	C27	HSIO71PB6/CCC_SE_CLKIN_S_15	Bank - 6

The following figure shows how the LEDs are connected to the PolarFIre FPGA. When the PolarFire output goes high (logic 1), the LED turns on. When the output goes low (logic 0), the LED turns off.

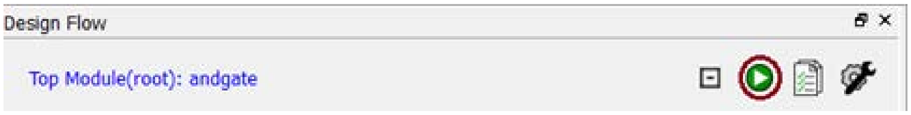
Figure 18 • **PolarFire Evaluation Kit User Defined LEDs**



Note: For more information, see the *PolarFire Evaluation Board Schematic* document provided separately.

8. If you click the **green arrow button** at the top of the **Design Flow** window, circled in the following figure, you can now run the design all the way through place and route.

Figure 19 • **Running the Design**

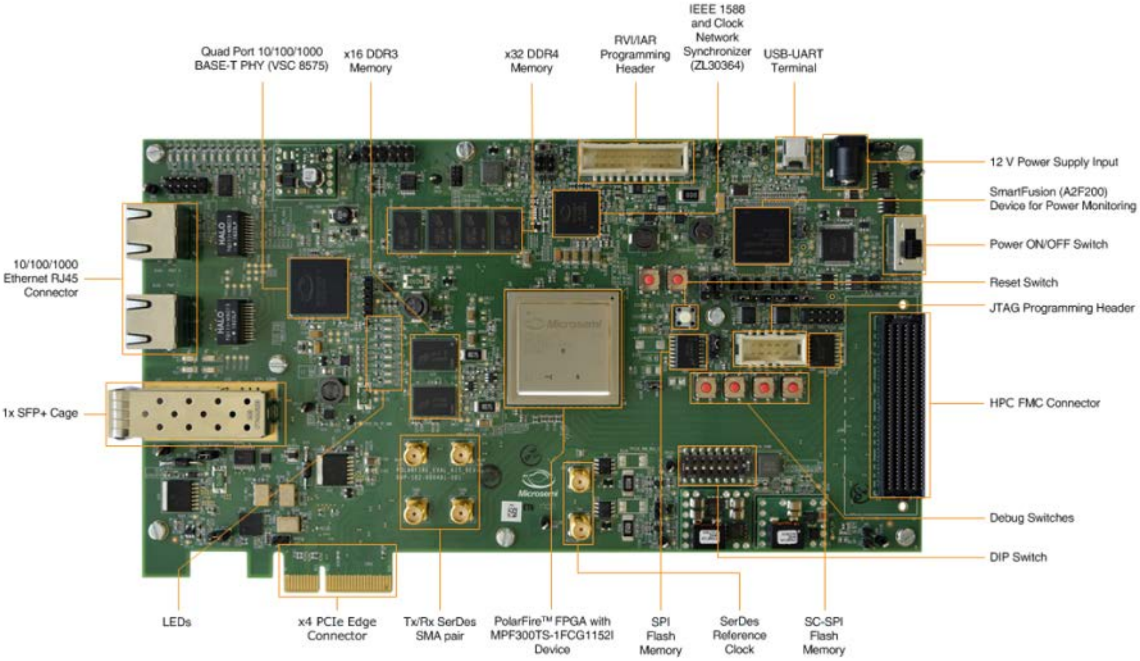


Note: Double click **Verify Timing** to perform timing analysis for sequential circuits and to analyze the generated timing report.

3.4 Step 4—Programming the Device

The following steps describe how to run the FlashPro in batch mode to program the PolarFire MPG300 on the PolarFire evaluation board, as shown in the following figure.

Figure 20 • **PolarFire Evaluation Kit Board**



1. Prior to programming (and powering up) the PolarFire evaluation board, confirm that the jumpers are positioned as listed in the following table.

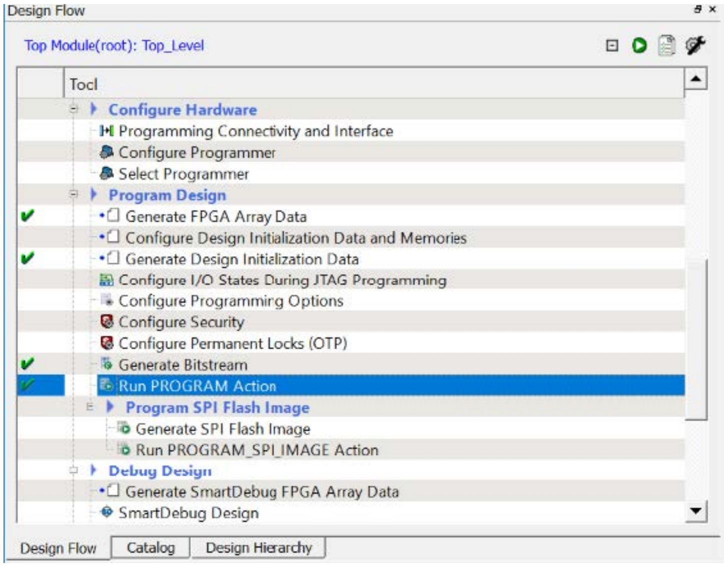
Table 4 • **Jumper Settings**

Jumper	Description	Pin	Default Setting
J18, J19, J20, J21, J22	Jumpers to select the PolarFire JTAG or A2F JTAG	Close pin 1 and 2 for programming the power sequence and monitoring chip through the FTDI	Open
		Close pin 2 and 3 for programming the PolarFire FPGA through FTDI User must always retain the default jumper setting	Closed
J46	Jumper to select switch-side MUX inputs of A or B to the line side	Close pin 1 and 2 (Input A to the line side) for routing the on-board 122.88 MHz differential clock oscillator output to the line side	Open
		Close pin 1 and 2 (Input B to the line side) for routing the on-board 125 MHz differential clock oscillator output to the line side	Closed
J28	Jumper to select the external JTAG or the onboard FlashPro5 for programming the PolarFire device	Close pin 1 and 2 for programming through the on-board FlashPro5	Closed
J26	Jumper to select the FTDI SPI or SC_SPI header	Close pin 1 and 2 for programming through the SC_SPI Header	Open
J27	Jumper to select between FTDI SPI or External SPI Flash to program the device	Close pin 1 and 2 for programming through the External SPI flash	Open
J23	Jumper to define the SPI interface mode	Close pin 1 and 2 to define the SPI Slave mode	Open
J4	Jumper to select the SW3 input or the ENABLE_FT4 232 signal from the FT4232H chip	Close pin 1 and 2 for manual power switching using SW3	Closed
		Close pin 2 and 3 for remote power switching using the GPIO capability of the FT4232 chip	Open
J12	Jumper to select the PolarFire VCCIO voltage (VCCIO_HPC_VADJ) to 1.2V, 1.5V, 1.8V, 2.5V, or 3.3V	Close pin 1 and 2 for 3.3 V Close pin 3 and 4 for 2.5 V Close pin 5 and 6 for 1.8 V Close pin 7 and 8 for 1.5 V Close pin 9 and 10 for 1.2 V	Open Closed Open Open Open
J43	Jumper to select the VDD voltage	Close pin 1 and 2 for 1.05 V	Open

2. Connect 12 V power supply brick to the J2 connector.
3. Connect a USB cable between the J1 mini USB connector and the host PC.
4. Slide the main power switch SW1 to on.
5. Install the FlashPro4 drivers if prompted. The drivers are located in the `<Libero SoC PolarFire v2021.2 Installation Directory>\drivers` folder.

6. Expand **Program Design** in the **Design Flow** window. Right-click **Run PROGRAM Action** and select **Run** to generate the programming file and begin programming.

Figure 21 • **Run Program Action**


Design Flow
Top Module(root): Top_Level
Tool
Configure Hardware
Programming Connectivity and Interface
Configure Programmer
Select Programmer
Program Design
Generate FPGA Array Data
Configure Design Initialization Data and Memories
Generate Design Initialization Data
Configure I/O States During JTAG Programming
Configure Programming Options
Configure Security
Configure Permanent Locks (OTP)
Generate Bitstream
Run PROGRAM Action
Program SPI Flash Image
Generate SPI Flash Image
Run PROGRAM_SPI_IMAGE Action
Debug Design
Generate SmartDebug FPGA Array Data
SmartDebug Design
Design Flow
Catalog
Design Hierarchy

7. FlashPro runs in batch mode and programs the device. Programming messages will be visible in the **Libero SoC log** window (programmer number will differ).

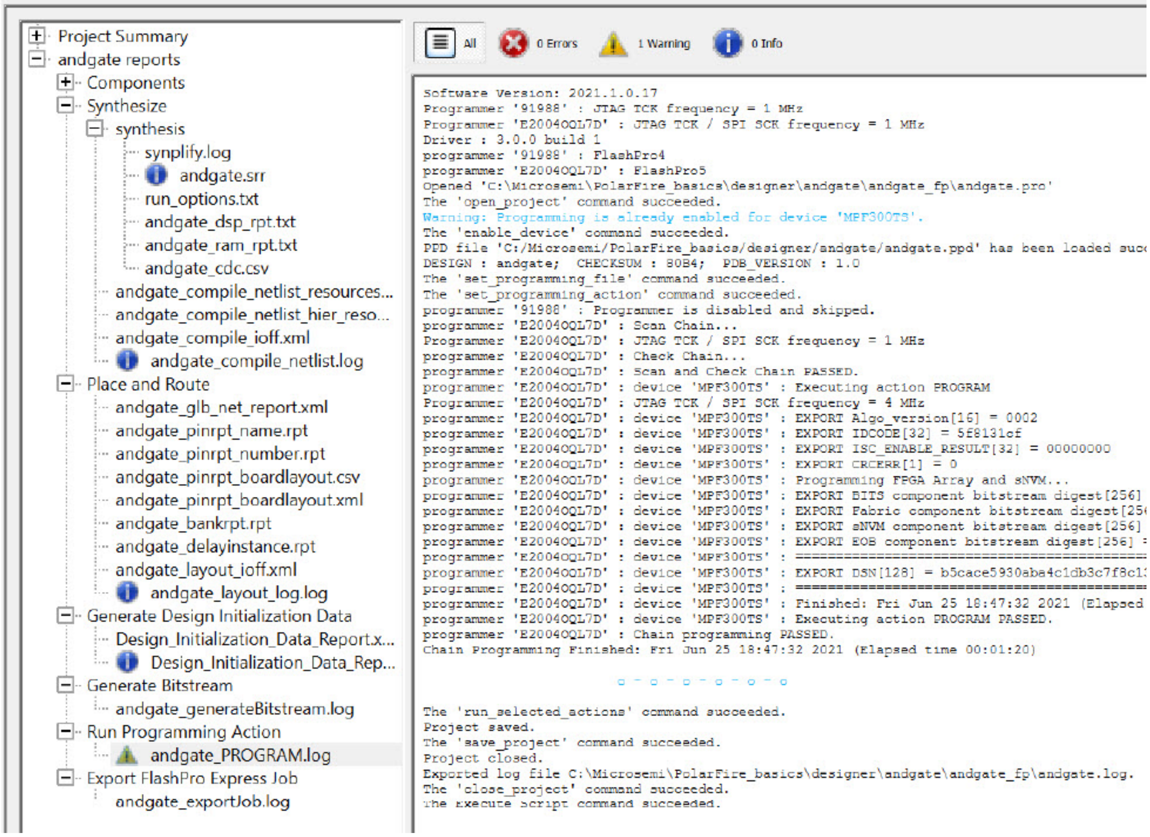
Note: Do not interrupt the programming sequence.

8. The following message should be visible in the **Reports** view under **Run PROGRAM Action**, when the device is programmed successfully (programmer number will differ).

```
programmer 'E2001JZK2X': device 'MPF300T_ES': Executing action PROGRAM
PASSED.
```

```
programmer 'E2004OQL7D': device 'MPF300TS': Executing action PROGRAM
PASSED.
```

Figure 22 • **Reports Tab**


Project Summary
andgate reports
Components
Synthesize
synthesis
synplify.log
andgate.srr
run_options.txt
andgate_dsp_rpt.txt
andgate_ram_rpt.txt
andgate_cdc.csv
andgate_compile_netlist_resources...
andgate_compile_netlist_hier_reso...
andgate_compile_ioff.xml
andgate_compile_netlist.log
Place and Route
andgate_glb_net_report.xml
andgate_pinrpt_name.rpt
andgate_pinrpt_number.rpt
andgate_pinrpt_boardlayout.csv
andgate_pinrpt_boardlayout.xml
andgate_bankrpt.rpt
andgate_delayinstance.rpt
andgate_layout_ioff.xml
andgate_layout_log.log
Generate Design Initialization Data
Design_Initialization_Data_Report.x...
Design_Initialization_Data_Rep...
Generate Bitstream
andgate_generateBitstream.log
Run Programming Action
andgate_PROGRAM.log
Export FlashPro Express Job
andgate_exportJob.log
All
0 Errors
1 Warning
0 Info

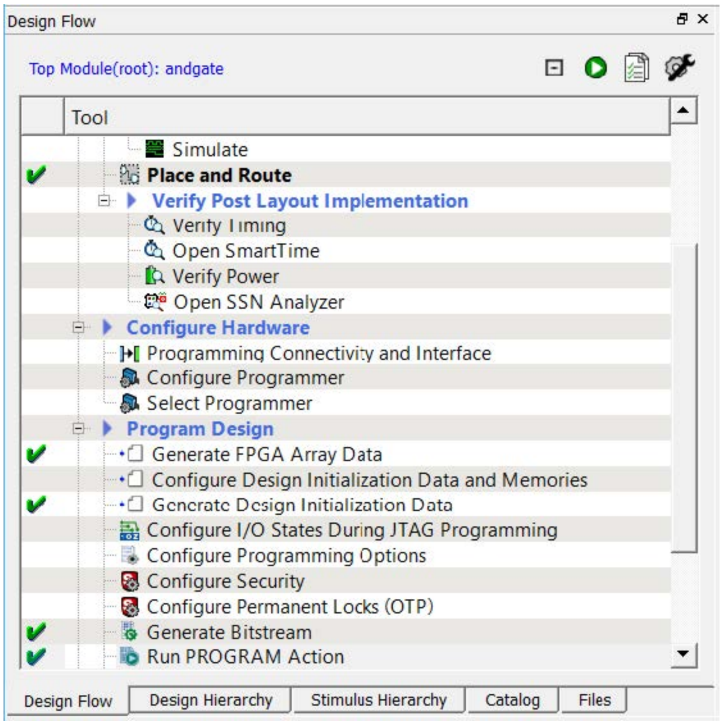
```
Software Version: 2021.1.0.17
Programmer '91988' : JTAG TCK frequency = 1 MHz
Programmer 'E2004OQL7D' : JTAG TCK / SPI SCK frequency = 1 MHz
Driver : 3.0.0 build 1
programmer '91988' : FlashPro4
programmer 'E2004OQL7D' : FlashPro5
Opened 'C:\Microsemi\PolarFire_basics\designer\andgate\andgate_fp\andgate.pro'
The 'open_project' command succeeded.
Warning: Programming is already enabled for device 'MPF300TS'.
The 'enable_device' command succeeded.
PPD file 'C:/Microsemi/PolarFire_basics/designer/andgate/andgate.ppd' has been loaded suc
DESIGN : andgate;  CHECKSUM : 80B4;  PDB_VERSION : 1.0
The 'set_programming_file' command succeeded.
The 'set_programming_action' command succeeded.
programmer '91988' : Programmer is disabled and skipped.
programmer 'E2004OQL7D' : Scan Chain...
Programmer 'E2004OQL7D' : JTAG TCK / SPI SCK frequency = 1 MHz
programmer 'E2004OQL7D' : Check Chain...
programmer 'E2004OQL7D' : Scan and Check Chain PASSED.
programmer 'E2004OQL7D' : device 'MPF300TS' : Executing action PROGRAM
Programmer 'E2004OQL7D' : JTAG TCK / SPI SCK frequency = 4 MHz
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT Algo_version[16] = 0002
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT IDCODE[32] = 5f8131cf
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT ISC_ENABLE_RESULT[32] = 00000000
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT CRCERR[1] = 0
programmer 'E2004OQL7D' : device 'MPF300TS' : Programming FPGA Array and sNVM...
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT BITS component bitstream digest[256]
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT Fabric component bitstream digest[25
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT sNVM component bitstream digest[256]
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT EOB component bitstream digest[256] =
programmer 'E2004OQL7D' : device 'MPF300TS' : ==========================================
programmer 'E2004OQL7D' : device 'MPF300TS' : EXPORT DSN[128] = b5cace5930aba4c1db3c7f8c1
programmer 'E2004OQL7D' : device 'MPF300TS' : ==========================================
programmer 'E2004OQL7D' : device 'MPF300TS' : Finished: Fri Jun 25 18:47:32 2021 (Elapsed
programmer 'E2004OQL7D' : device 'MPF300TS' : Executing action PROGRAM PASSED.
programmer 'E2004OQL7D' : Chain programming PASSED.
Chain Programming Finished: Fri Jun 25 18:47:32 2021 (Elapsed time 00:01:20)

                   o - o - o - o - o - o

The 'run_selected_actions' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
Exported log file C:\Microsemi\PolarFire_basics\designer\andgate\andgate_fp\andgate.log.
The 'close_project' command succeeded.
The Execute Script command succeeded.
```

A green tick mark will appear next to **Program Design** and **Run PROGRAM Action** in the **Design Flow** window to indicate successful programming as shown in the following figure.

Figure 23 • **Design Flow Green Tick Marks**


Design Flow
Top Module(root): andgate
Tool
Simulate
✓ Place and Route
Verify Post Layout Implementation
Verify Timing
Open SmartTime
Verify Power
Open SSN Analyzer
Configure Hardware
Programming Connectivity and Interface
Configure Programmer
Select Programmer
Program Design
✓ Generate FPGA Array Data
Configure Design Initialization Data and Memories
✓ Generate Design Initialization Data
Configure I/O States During JTAG Programming
Configure Programming Options
Configure Security
Configure Permanent Locks (OTP)
✓ Generate Bitstream
✓ Run PROGRAM Action
Design Flow
Design Hierarchy
Stimulus Hierarchy
Catalog
Files

9. Click **Project > Save** from the Libero menu to save your completed project.

Note: Do not close the window.

3.5 Step 5—Running the Design

This is a very simple design designed to show an AND gate. The following steps are performed to run the design.

- Observe LED1 with switch 5 and switch 6 released—it will illuminate.
- Press switch 5 and observe the behavior of LED1—it will turn off.
- Press switch 6 and observe the behavior of LED1—it will turn off.
- Press both switches and LED1 will turn off.

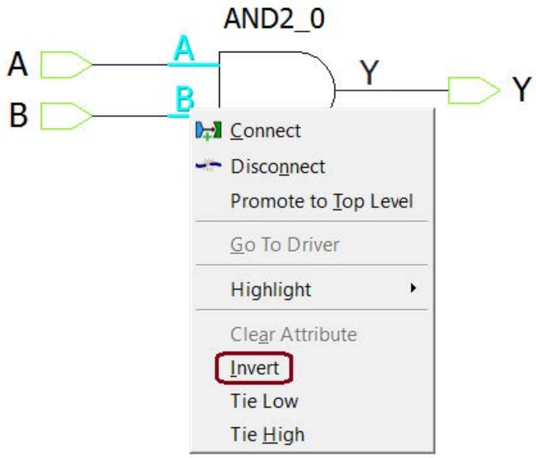
For and AND gate, users would expect the LED to illuminate when both switches are pressed, but in this application, releasing the switch generates a logic one, which illuminates the LED. To correct this, we need to invert the inputs to the AND gate. For more information, see **Figure 16• Switches Interface** in *UG0747: PolarFire FPGA Evaluation Kit User Guide*.

3.6 Step 6—Design Iterations in the Libero SoC PolarFire

The following steps describe the design iteration in the Libero SoC PolarFire v2021.2.

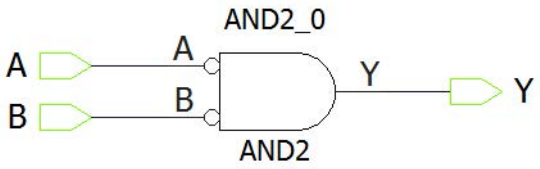
1. Reopen the **SmartDesign** Canvas again, look for the SD andgate tab.
2. Hold the **CTRL** key and select input ports **A** and **B** of AND2_0, then right-click, to bring up the menu. Select **Invert** to invert the signals, as shown in the following figure. This is built into the SmartDesign capabilities to invert or tie off any I/O.

Figure 24 • **Design Iterations**



3. The ports will be inverted.

Figure 25 • **Ports Inversion**



4. Generate the Component . It resets the rest of the compilation functions, but keeps your I/O constraints.
5. Right-click **Run PROGRAM Action** on the **Design flow** window, then select **Update and Run** to run all the steps in between and reprogram the device in one click.
6. Check if the **Warning** dialog box about running in non-timing-driven mode is displayed behind your active window.
7. Observe the correct behavior on the board by pressing both switches simultaneously to illuminate LED1.

This step completes the tutorial of the Libero SoC PolarFire v2021.2 design flow.

4 References

The following documents are referred for this tutorial.

- *UG0758: PolarFire FPGA Design Flow User Guide*
- *UG0779: SmartDesign User Guide for PolarFire*
- *UG0750: PolarFire FPGA I/O Editor User Guide*
- *PolarFire FPGA Tcl Commands Reference Guide*
- *UG0776: PolarFire FPGA Design Constraints User Guide*
- *UG0769: Timing Constraints Editor User Guide for PolarFire*
- *UG0738: Netlist Viewer Interface User Guide for PolarFire*
- *UG0821: PolarFire FPGA Chip Planner User Guide*
- *SmartPower User Guide for Libero SoC PolarFire*
- *UG0773: PolarFire FPGA SmartDebug User Guide*