IGLOO2 FPGA Low Standby Power - Libero SoC v11.7

DG0564 Demo Guide

Microsemi®

Power Matters.™

Contents

Figures

Tables

1 Preface

1.1 Purpose

This demo is for the IGLOO®2 field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

1.2 Intended Audience

This demo guide is intended for:

- FPGA designers
- System-level designers

1.3 References

The following documents are referred in this demo guide:

- *SmartFusion2 and IGLOO2 Power Estimator User Guide*
- *UG0444: SmartFusion2 SoC and IGLOO2 FPGA Low Power Design User Guide*
- *UG0445: SmartFusion2 SoC and IGLOO2 FPGA Fabric User Guide*

2 IGLOO2 FPGA Low Standby Power - Libero SoC v11.7

2.1 Introduction

Microsemi IGLOO2 FPGAs are designed to meet the demand of low-power FPGAs. The IGLOO2 devices exhibit lower power consumption in static and dynamic modes. This demo guide describes how to implement the standby power mode on the IGLOO2 devices using SmartDesign, and measure the standby power. The design drives the light emitting diodes (LEDs) on the IGLOO2 Evaluation Kit with a pattern based on the state of the switches SW1 and SW3, as shown in Table 1.

Table 1 • **LEDs Pattern**

LED E1, F4, F3, G7 Behavior	Standby Entry (SW1)	Standby Exit (SW3)
LEDs toggle	Released	Released
LEDs on	Depressed and Released	Released
LEDs toggle	Depressed and Released	Depressed

This demo guide describes the following:

- Creating a Libero® System-on-Chip (SoC) project
- Implementing the standby power mode on the IGLOO2 devices using SmartDesign
- Importing a PDC file, running layout, and programming the IGLOO2 silicon
- Measuring the standby power using a standard digital voltmeter (DVM)/multimeter

2.2 Design Requirements

Table 2 shows the design requirements.

Table 2 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
IGLOO2 Evaluation Kit: • 12 V adapter • FlashPro4 programmer	Rev C or later
Desktop or Laptop	Windows 64-bit Operating System
Software Requirements	
Libero SoC	v11.7
FlashPro Programming Software	v11.7

2.3 Demo Design

2.3.1 Introduction

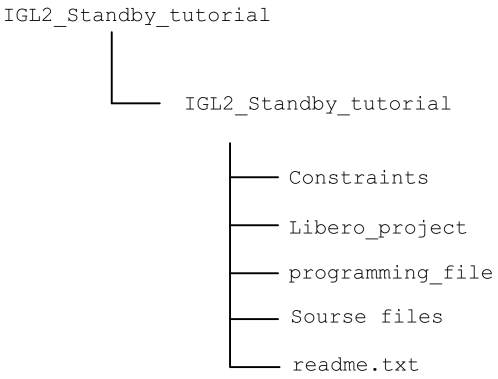
The demo design files are available for download from the following path in the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2gl_dg0564_liberov11p7_df*

The demo design files include:

- Libero SoC project
- Constraint file
- Programming file
- Source files
- Readme file

Figure 1 shows the top-level structure of the design files. For further details, refer to the `Readme.txt` file.

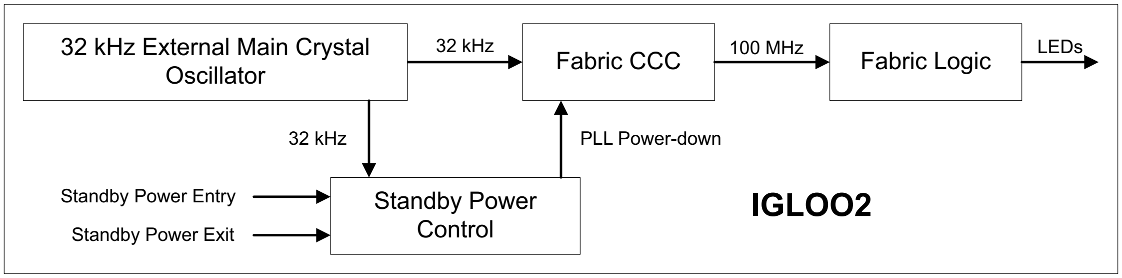
Figure 1 • **Design Files Top-Level Structure**



The design consists a 32 kHz external main crystal oscillator, fabric CCC (FCCC), standby power control logic, and fabric logic block. Figure 2 shows the block diagram of the design.

FCCC is configured to provide a 100 MHz clock to the fabric logic. It is also configured with phase-locked loop (PLL) power-down enabled. The 32 kHz external main crystal oscillator is the reference clock source for FCCC. The lock signal is used as the reset signal to the fabric logic. The standby power control logic consists a clocked S-R latch, which powers down the PLL of FCCC. The fabric logic consists 269 stages of 18-bit loadable up-counters, 13 stages of shift registers, and 11 stages of LSRAM and math blocks. It also consists an LED driver block, which is connected to a set of LEDs to monitor the state of the fabric while entering and exiting the standby power mode.

Figure 2 • **Design Block Diagram**



2.3.2 Extracting the Source Files

Extract the *m2gl_dg0564_liberov11p7_df.zip* file to the *<C:\ or D:\>Microsemi_prj* folder on the PC. Confirm that a folder named *IGL2_Standby_tutorial* containing sub-folders named *Source_files* and *Constraints* are extracted.

2.4 Creating the Design

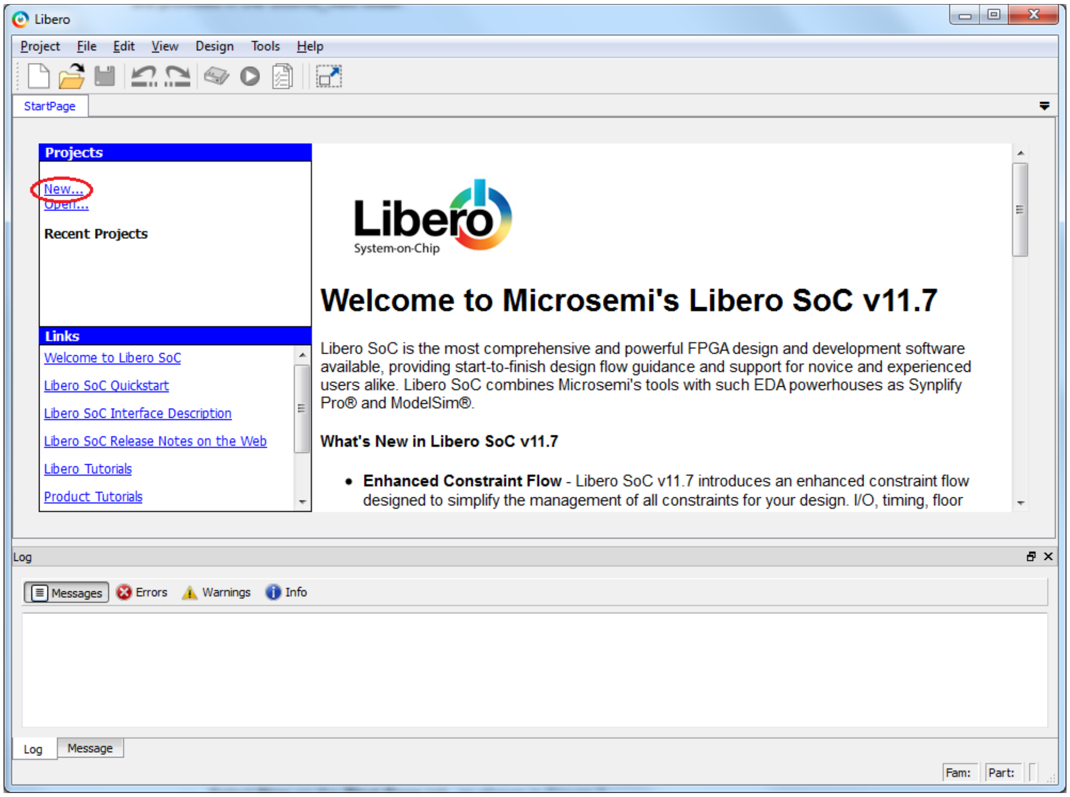
This section describes how to create the standby power mode enabled design using SmartDesign. Some source files are provided in the *Source_files* folder.

2.4.1 Launching Libero SoC

The following steps describe how to launch Libero SoC:

1. Go to **Start > Programs > Microsemi > Libero SoC v11.7 > Libero SoC v11.7,** or double-click the shortcut icon on the PC. This opens the **Libero SoC Project Manager** window, as shown in Figure 3.

Figure 3 • **Libero SoC Project Manager**



Welcome to Microsemi's Libero SoC v11.7

Libero SoC is the most comprehensive and powerful FPGA design and development software available, providing start-to-finish design flow guidance and support for novice and experienced users alike. Libero SoC combines Microsemi's tools with such EDA powerhouses as Synplify Pro® and ModelSim®.

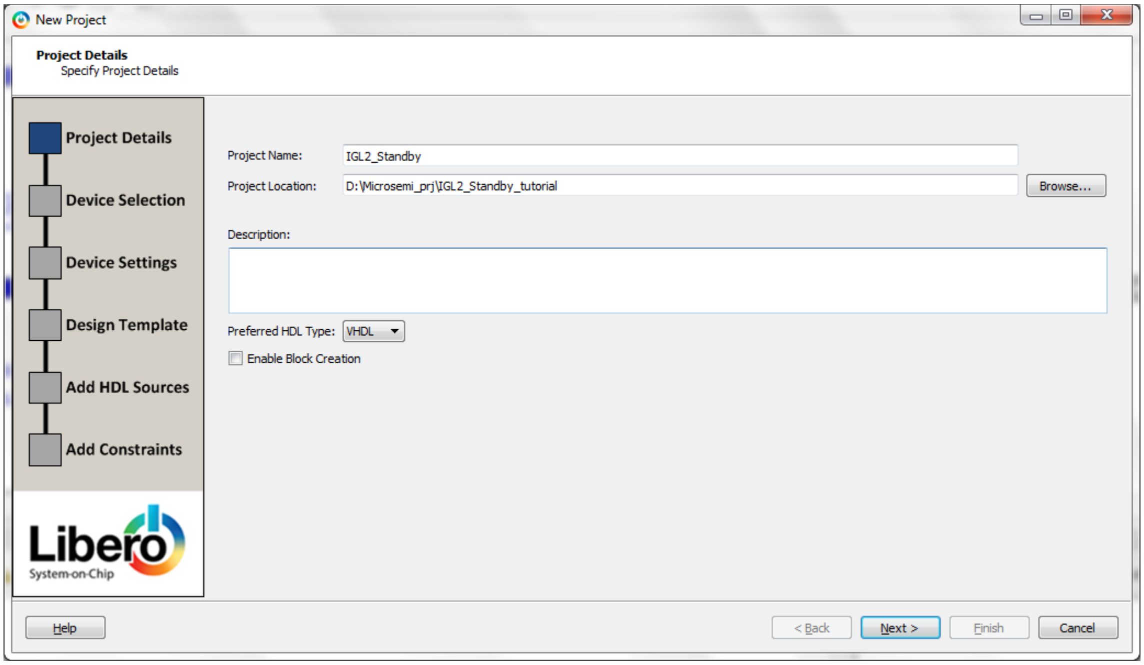
What's New in Libero SoC v11.7

- **Enhanced Constraint Flow** - Libero SoC v11.7 introduces an enhanced constraint flow designed to simplify the management of all constraints for your design. I/O, timing, floor

2. Create a new project using one of the following options:
   - Select **New** on the **Start Page** tab, as shown in Figure 3.
   - In the Libero SoC menu, go to **Project > New Project**.

   This opens the **New Project** window, as shown in Figure 4 on page 9.
3. Enter the following information in the **New Project - Project Details** window, as shown in Figure 4 on page 9:
   - **Project Name**: IGL2_Standby
   - **Project Location**: <C:\ or D:\>Microsemi_prj\IGL2_Standby_tutorial
   - **Preferred HDL type**: VHDL
   - **Enable Block Creation**: Not selected

Figure 4 • **New Project - Project Details**



New Project

Project Details
Specify Project Details

Project Name: IGL2_Standby

Project Location: D:\Microsemi_prj\IGL2_Standby_tutorial

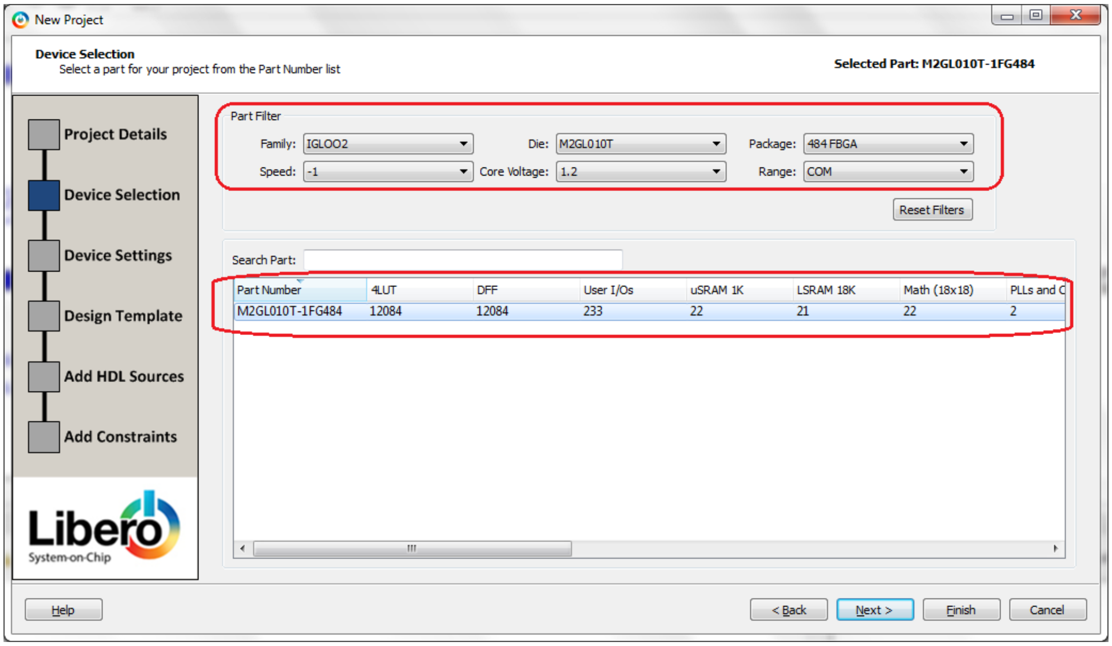
Description:

Preferred HDL Type: VHDL

□ Enable Block Creation

4. Click **Next**. This opens **New Project - Device Selection** window, as shown in Figure 5.
5. Select the following values from the drop-down list, highlighted in Figure 5:
   - **Family**: IGLOO2
   - **Die**: M2GL010T
   - **Package**: 484 FBGA
   - **Speed**: -1
   - **Core Voltage**: 1.2
   - **Range**: COM
6. Select the filtered device **M2GL010T-1FG484**, as shown in Figure 5.

Figure 5 • **New Project - Device Selection**



New Project

Device Selection
Select a part for your project from the Part Number list

Selected Part: M2GL010T-1FG484

Part Filter

Family: IGLOO2
Die: M2GL010T
Package: 484 FBGA
Speed: -1
Core Voltage: 1.2
Range: COM

Search Part:

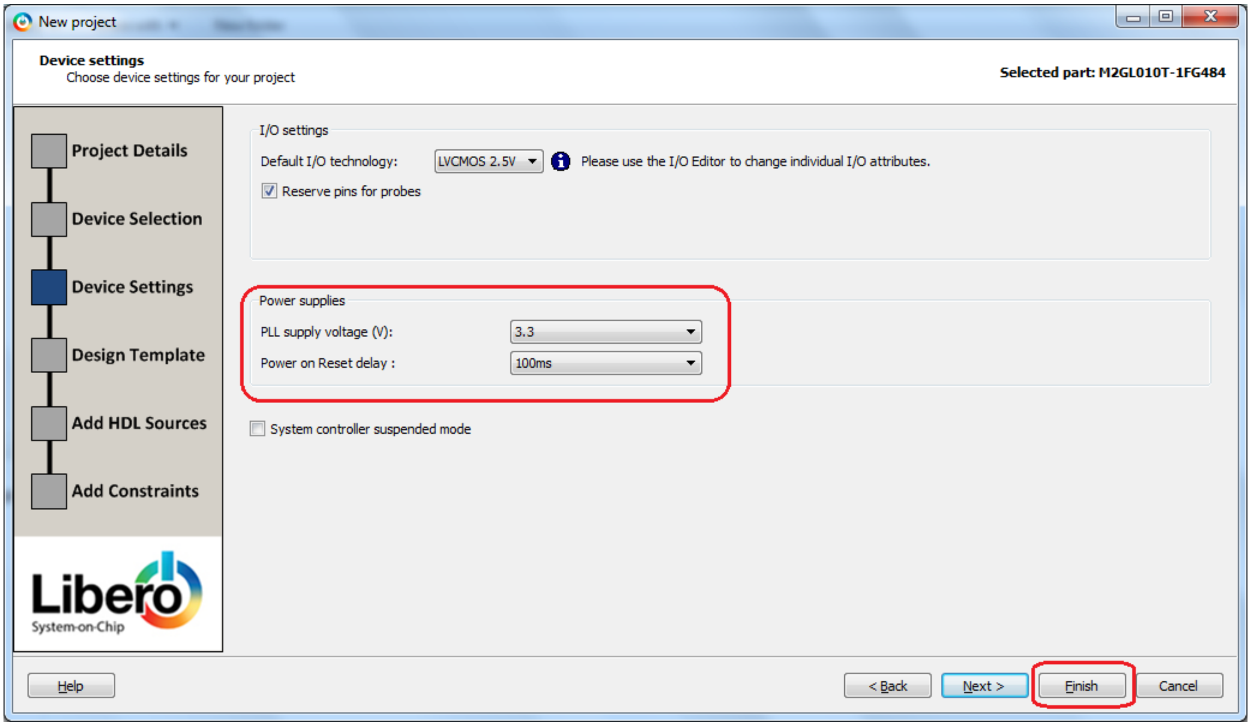
Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and C
M2GL010T-1FG484	12084	12084	233	22	21	22	2

7. Click **Next**. This opens **New Project - Device Settings** window, as shown in Figure 6.
8. Select the following values in the **Power supplies** section from the drop-down list, highlighted in Figure 6:
   - **PLL supply voltage (V)**: 3.3
   - **Power on Reset delay**: 100ms

The PLL analog supply voltage can be either 2.5 V or 3.3 V. The voltage setting in the **New Project - Device Settings** window must match the PLL analog supply voltage on the board to ensure that the PLL works properly. The PLL analog supply voltage is connected to 3.3 V on the IGLOO2 Evaluation Kit.

9. Do not change the default selections. Click **Finish**.

Figure 6 • **New Project - Device Settings**



Device settings
Choose device settings for your project

Selected part: M2GL010T-1FG484

I/O settings

Default I/O technology: LVCMOS 2.5V

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

☑ Reserve pins for probes

Power supplies

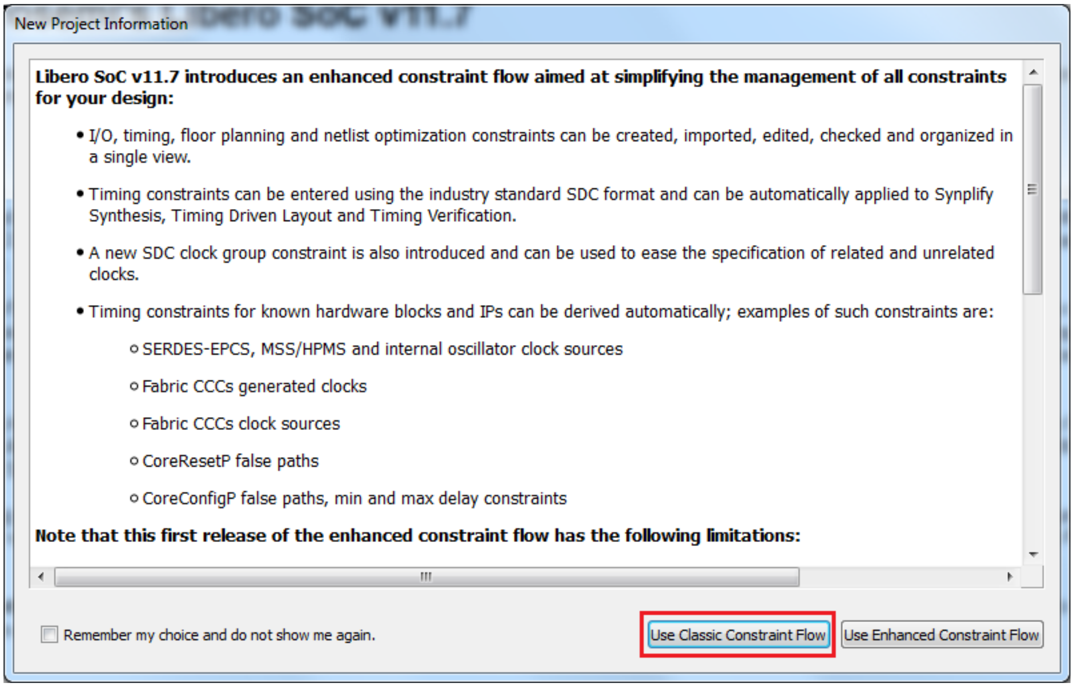
PLL supply voltage (V): 3.3

Power on Reset delay : 100ms

□ System controller suspended mode

10. **New Project Information** window opens, as shown in Figure 7. Click **Use Classic Constraint Flow**.

Figure 7 • **New Project Information**



New Project Information

Libero SoC v11.7 introduces an enhanced constraint flow aimed at simplifying the management of all constraints for your design:

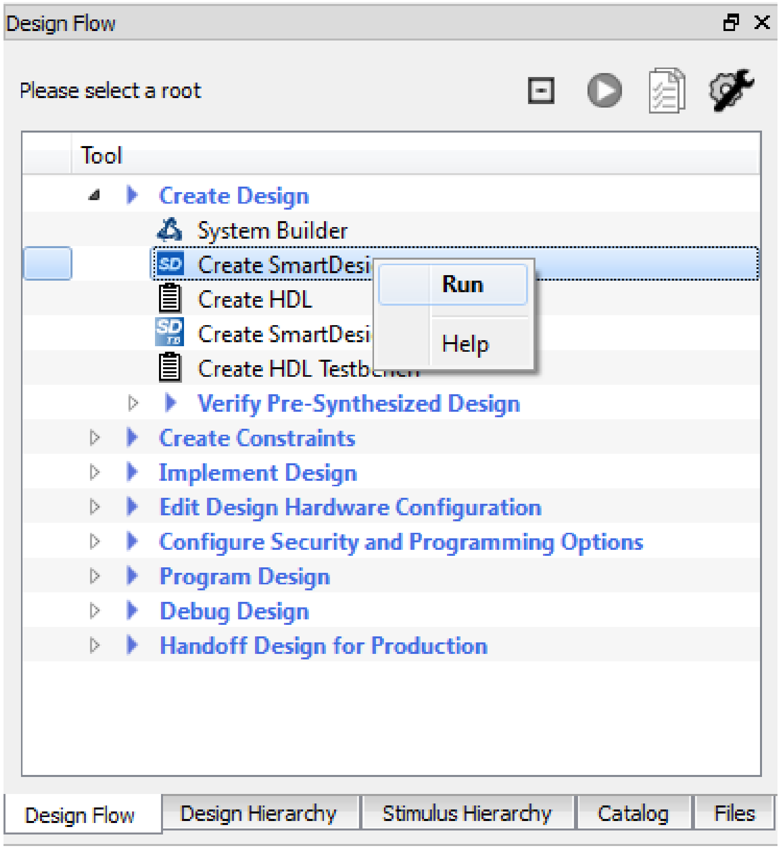
- I/O, timing, floor planning and netlist optimization constraints can be created, imported, edited, checked and organized in a single view.
- Timing constraints can be entered using the industry standard SDC format and can be automatically applied to Synplify Synthesis, Timing Driven Layout and Timing Verification.
- A new SDC clock group constraint is also introduced and can be used to ease the specification of related and unrelated clocks.
- Timing constraints for known hardware blocks and IPs can be derived automatically; examples of such constraints are:
  - SERDES-EPCS, MSS/HPMS and internal oscillator clock sources
  - Fabric CCCs generated clocks
  - Fabric CCCs clock sources
  - CoreResetP false paths
  - CoreConfigP false paths, min and max delay constraints

Note that this first release of the enhanced constraint flow has the following limitations:

☐ Remember my choice and do not show me again.

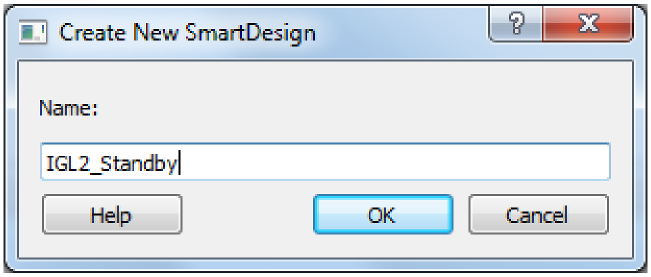
11. In the **Design Flow** window, expand **Create Design**, as shown in Figure 8.
12. Right-click **Create SmartDesign** and click **Run**.

Figure 8 • **Creating SmartDesign**



13. In the **Create New SmartDesign** dialog box, enter the **Name** as **IGL2_Standby** and click **OK**. A new SmartDesign canvas opens.

Figure 9 • **Entering SmartDesign Name**

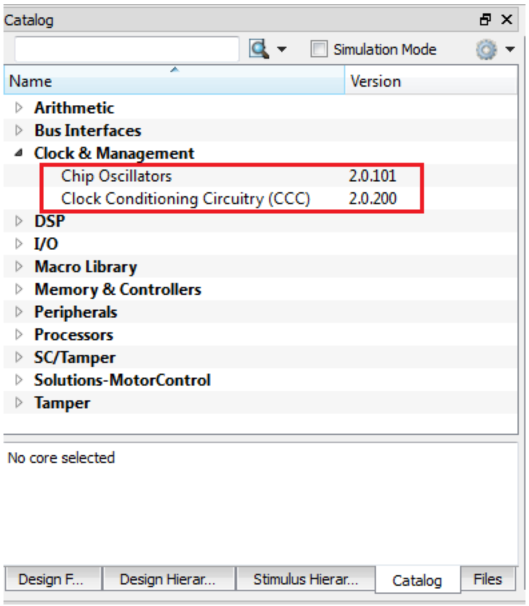


Name: IGL2_Standby

This design uses a fabric CCC to generate a 100 MHz internal clock. The CCC reference clock is the 32 kHz external main crystal oscillator.

14. In the IP **Catalog** tab, expand **Clock & Management**.

Figure 10 • **Clock & Management Category of Libero SoC IP Catalog**



Catalog

□ Simulation Mode

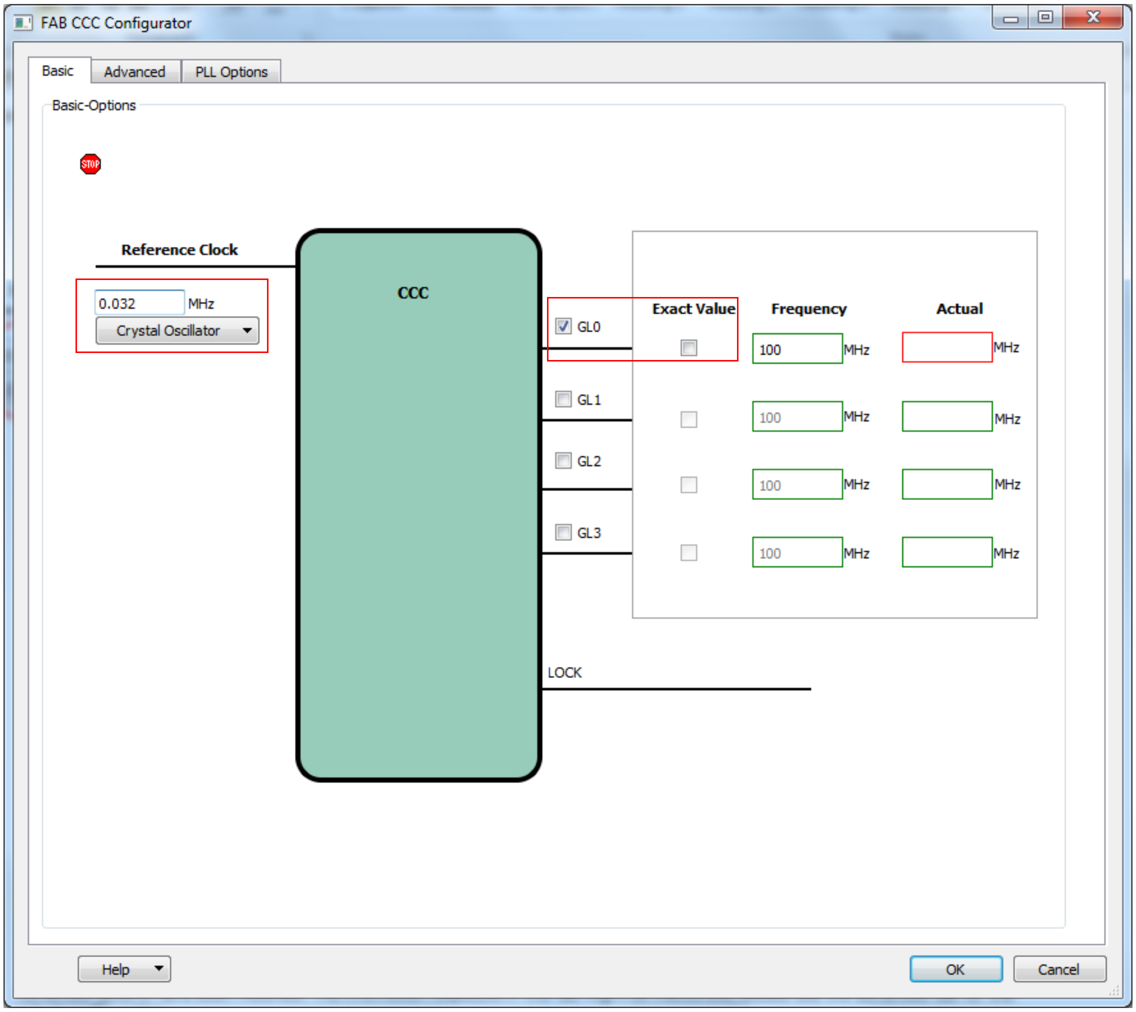
Name	Version
▷ Arithmetic	
▷ Bus Interfaces	
◢ Clock & Management	
Chip Oscillators	2.0.101
Clock Conditioning Circuitry (CCC)	2.0.200
▷ DSP	
▷ I/O	
▷ Macro Library	
▷ Memory & Controllers	
▷ Peripherals	
▷ Processors	
▷ SC/Tamper	
▷ Solutions-MotorControl	
▷ Tamper	

No core selected

15. Drag an instance of the clock conditioning circuitry (CCC) v2.0.200 component into the SmartDesign canvas.

16. Double-click the FCCC_0 component in the SmartDesign canvas and open the **FAB CCC Configurator** window, as shown in Figure 11.
17. Click the **Basic** tab in the **FAB CCC Configurator** window, as shown in Figure 11 and enter the following information:
    - **Reference Clock Frequency**: 0.032 MHz
    - **Reference Clock**: Select **Oscillators > Crystal Oscillator** from the drop-down list
    - **GL0:** Checked; Frequency: 100 MHz

Figure 11 • **Configuring Fabric CCC**

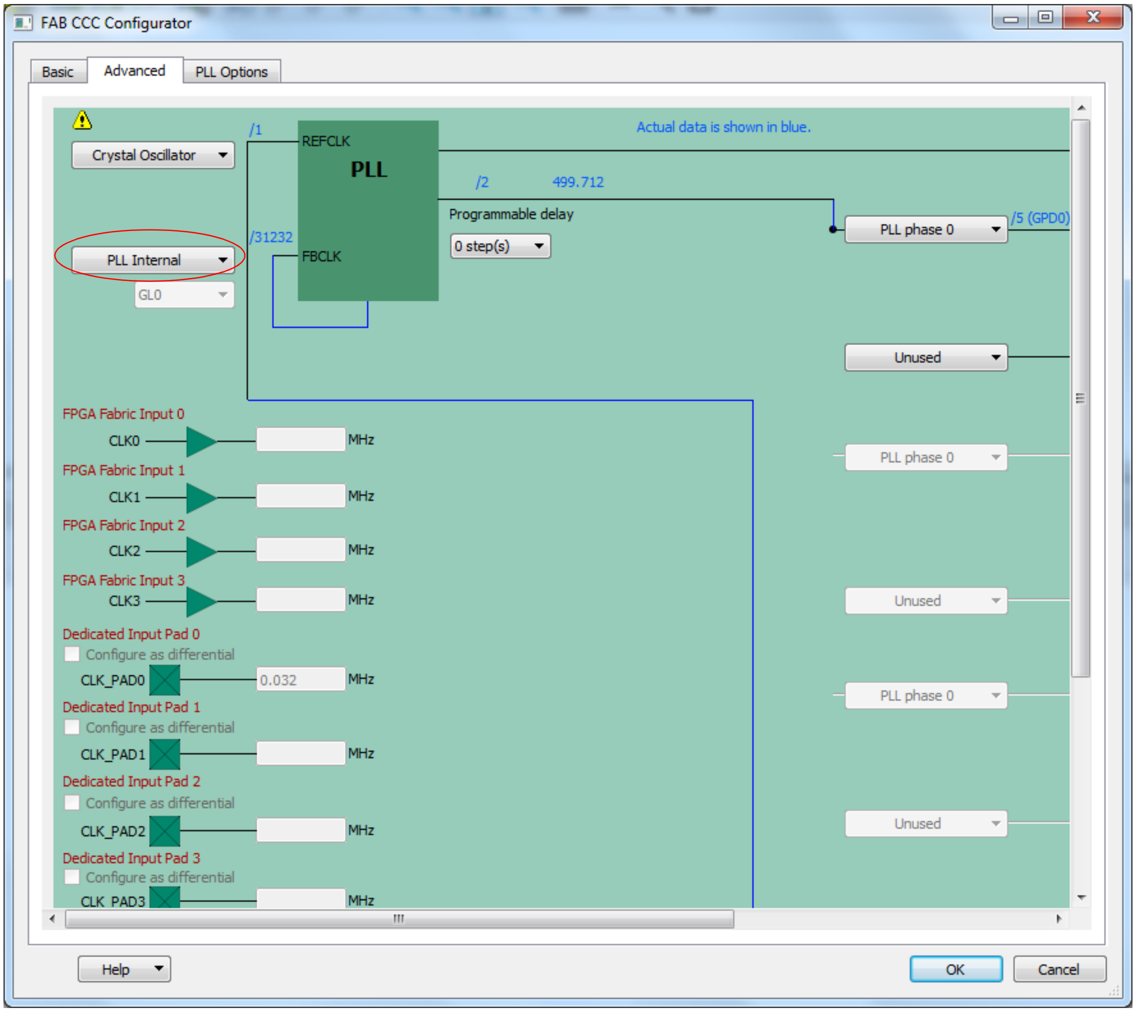


Reference Clock: 0.032 MHz, Crystal Oscillator

	Exact Value	Frequency	Actual
☑ GL0	□	100 MHz	MHz
□ GL1	□	100 MHz	MHz
□ GL2	□	100 MHz	MHz
□ GL3	□	100 MHz	MHz

18. Click the **Advanced** tab in the **FAB CCC Configurator** window and select **Internal > PLL Internal** from the drop-down list as PLL feedback source, as shown in Figure 12.

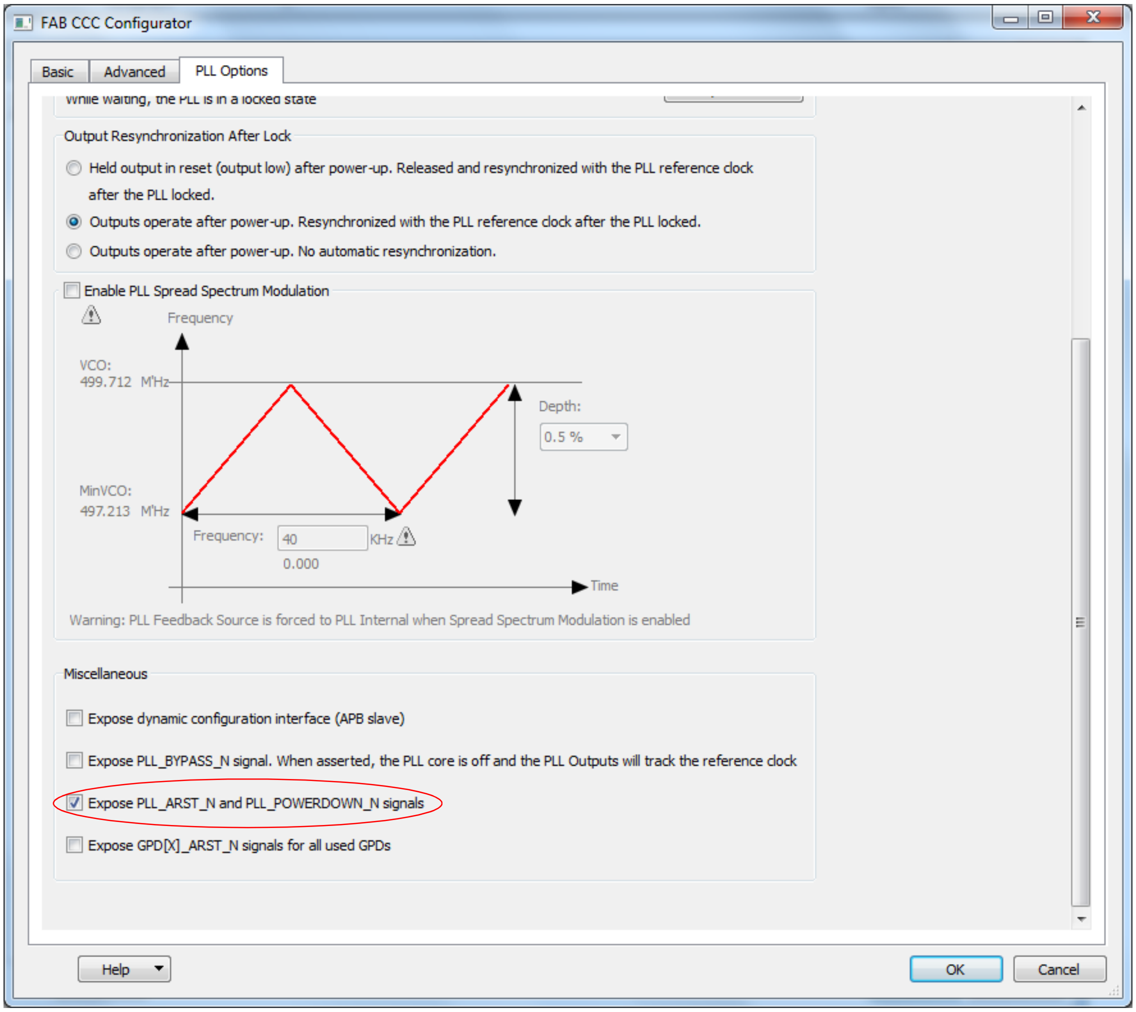
Figure 12 • **Configuring PLL Feedback Source**



Actual data is shown in blue.

19. Click the **PLL Options** tab in the **FAB CCC Configurator** window and select the **Expose PLL_ARST_N and PLL_POWERDOWN_N signals** checkbox, as shown in Figure 13.

Figure 13 • **Configuring PLL Power-down Signal**



FAB CCC Configurator

Basic | Advanced | PLL Options

while waiting, the PLL is in a locked state

Output Resynchronization After Lock

○ Held output in reset (output low) after power-up. Released and resynchronized with the PLL reference clock after the PLL locked.

● Outputs operate after power-up. Resynchronized with the PLL reference clock after the PLL locked.

○ Outputs operate after power-up. No automatic resynchronization.

□ Enable PLL Spread Spectrum Modulation

Warning: PLL Feedback Source is forced to PLL Internal when Spread Spectrum Modulation is enabled

Miscellaneous

□ Expose dynamic configuration interface (APB slave)

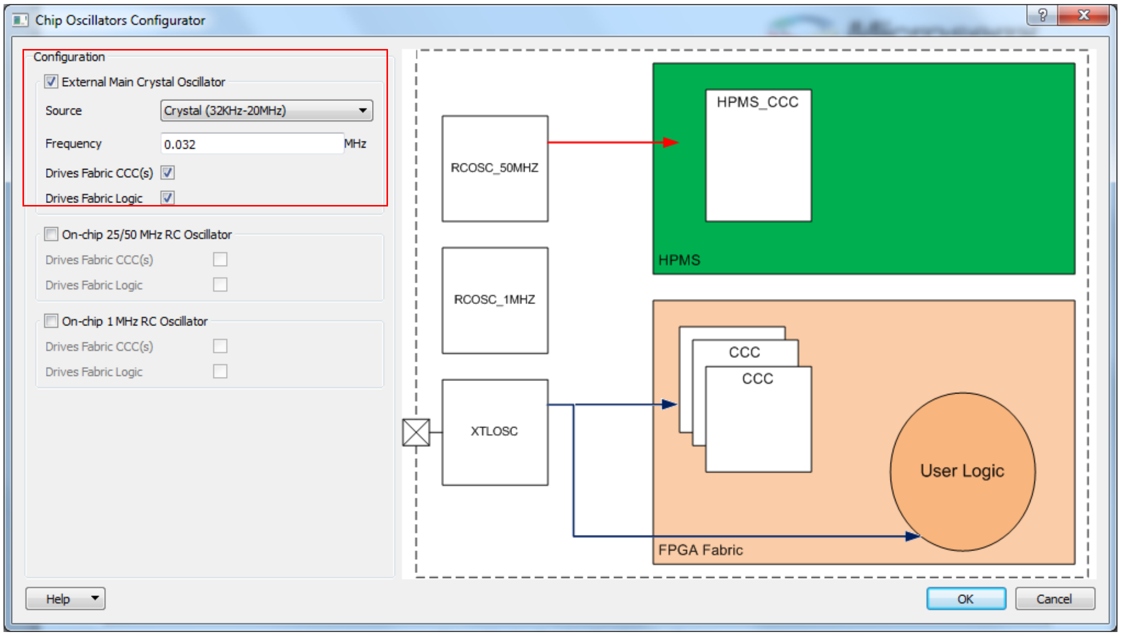
□ Expose PLL_BYPASS_N signal. When asserted, the PLL core is off and the PLL Outputs will track the reference clock

☑ Expose PLL_ARST_N and PLL_POWERDOWN_N signals

□ Expose GPD[X]_ARST_N signals for all used GPDs

20. Click **OK**.
21. Drag an instance of the Chip Oscillators v2.0.101 component from the IP catalog into the SmartDesign canvas.
22. Double-click the OSC_0 component in the SmartDesign canvas and open the **Chip Oscillators Configurator** window, as shown in Figure 14 on page 17.
23. Configure the external main crystal oscillator to drive FCCC and fabric logic.
    Enter the following information, as shown in Figure 14 on page 17:
    - **External Main Crystal Oscillator**: Select
    - **Source**: Select **Crystal (32 KHz - 20 MHz)** from the drop-down list
    - **Frequency**: 0.032 MHz
    - **Drives Fabric CCC(s)**: Select
    - **Drives Fabric Logic**: Select

Figure 14 • **Configuring Chip Oscillators**



Chip Oscillators Configurator

Configuration

☑ External Main Crystal Oscillator

Source: Crystal (32KHz-20MHz)

Frequency: 0.032 MHz

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☑

□ On-chip 25/50 MHz RC Oscillator

Drives Fabric CCC(s) □

Drives Fabric Logic □

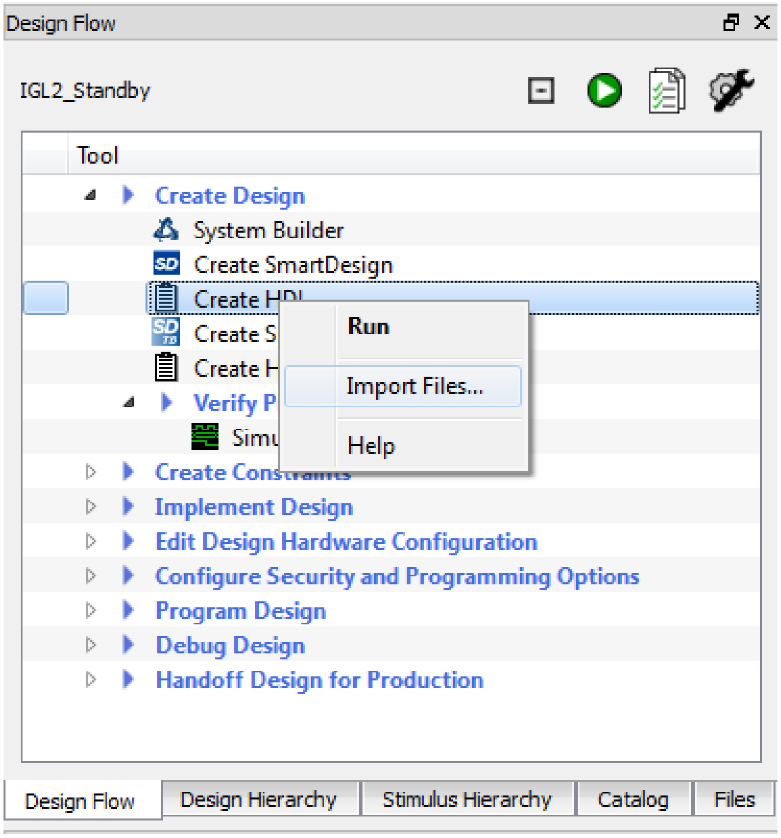
□ On-chip 1 MHz RC Oscillator

Drives Fabric CCC(s) □

Drives Fabric Logic □

24. Click **OK**.
25. Import the VHDL source files into the project by selecting **Create HDL** under **Create Design** in the **Design Flow** tab. Right-click and select **Import Files…**, as shown in Figure 15.

Figure 15 • **Importing HDL Source Files**



Design Flow

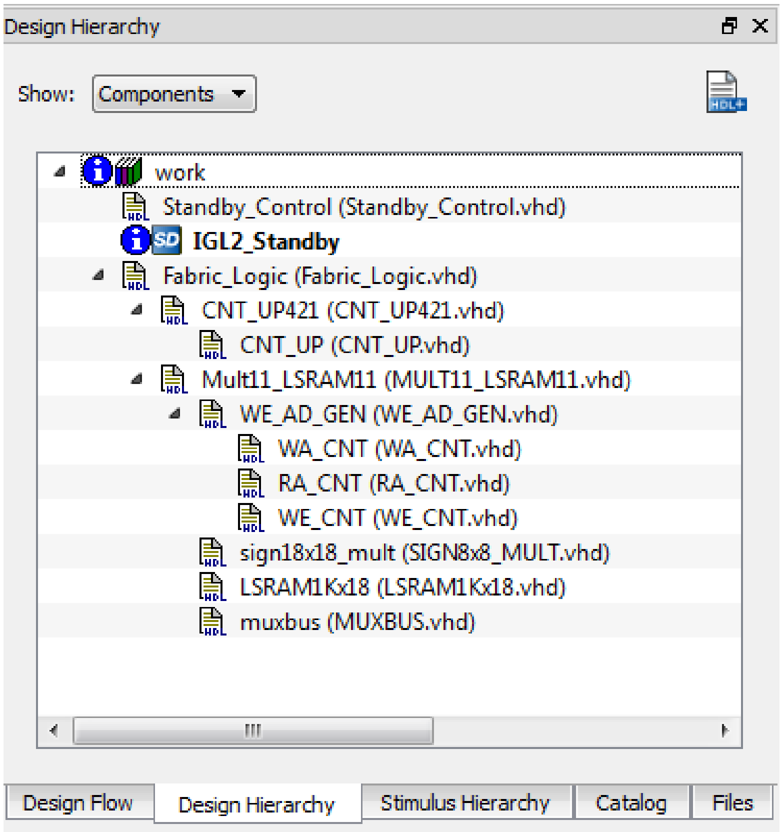
IGL2_Standby

Tool

- Create Design
  - System Builder
  - Create SmartDesign
  - Create HDL
  - Create S
  - Create H
  - Verify P
    - Simu
- Create Constraints
- Implement Design
- Edit Design Hardware Configuration
- Configure Security and Programming Options
- Program Design
- Debug Design
- Handoff Design for Production

26. Browse to *<C:\ or D:\>Microsemi_prj\IGL2_Standby_tutorial\Source_files,* select all `.vhd`, `.v`, and `.h` files, and click **Open**.
    The files are visible in the **Design Hierarchy** tab.

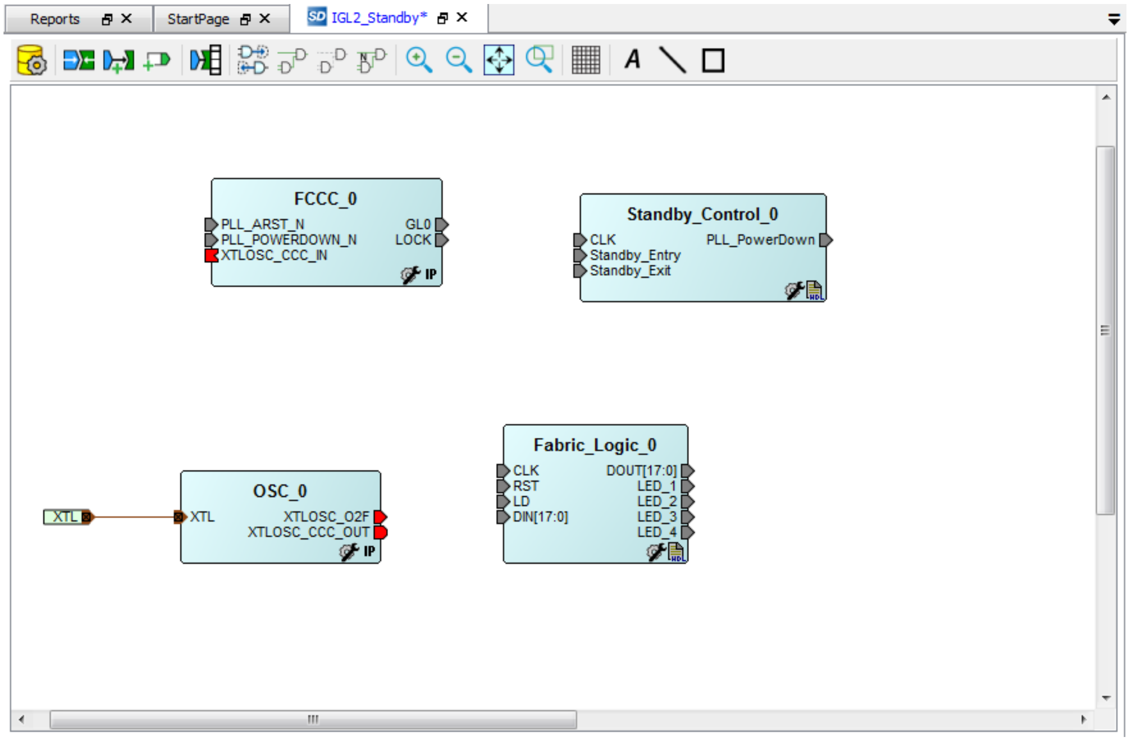
Figure 16 • **Design Hierarchy Tab with Imported Files**



```
work
    Standby_Control (Standby_Control.vhd)
    IGL2_Standby
  Fabric_Logic (Fabric_Logic.vhd)
    CNT_UP421 (CNT_UP421.vhd)
      CNT_UP (CNT_UP.vhd)
    Mult11_LSRAM11 (MULT11_LSRAM11.vhd)
      WE_AD_GEN (WE_AD_GEN.vhd)
        WA_CNT (WA_CNT.vhd)
        RA_CNT (RA_CNT.vhd)
        WE_CNT (WE_CNT.vhd)
      sign18x18_mult (SIGN8x8_MULT.vhd)
      LSRAM1Kx18 (LSRAM1Kx18.vhd)
      muxbus (MUXBUS.vhd)
```

27. Drag the Standby_Control and Fabric_Logic components into the SmartDesign canvas. The SmartDesign resembles Figure 17.
28. Align the components to improve the appearance of the canvas.
    Expand the canvas area by selecting **View > Maximize Work Area**, or click the icon on the tool bar.

Figure 17 • **SmartDesign Canvas after Adding Components**



2.4.2 Connecting Components in the Canvas

SmartDesign in Libero SoC has a connection mode that supports click, drag, and release to connect the components.

Connect the components in the SmartDesign canvas using the following procedure:

1. Select **SmartDesign > Connection Mode** from the Libero SoC menu.
2. Connect the XTLOSC_CCC_OUT port of the OSC_0 component to the XTLOSC_CCC_IN port of the FCCC_0 component as follows:
   a. Click and hold the XTLOSC_CCC_OUT port of the OSC_0 component.

   b. Drag the XTLOSC_CCC_IN port of the FCCC_0 component and release the mouse button to connect.

Note: You can also connect the ports by selecting them using **CTRL** (Ctrl + Click to select a port), right-clicking any of the selected ports, and selecting **Connect**.

3. Connect the other components in the SmartDesign canvas as per Table 3.

Table 3 • **Connections in Canvas**

From	To
OSC_0: XTLOSC_O2F	Standby_Control_0: CLK
Standby_Control_0: PLL_PowerDown	FCCC_0: PLL_ARST_N
	FCCC_0: PLL_POWERDOWN_N
FCCC_0: GL0	Fabric_Logic_0: CLK
FCCC_0: LOCK	Fabric_Logic_0: RST

4. Select **SmartDesign > Connection Mode** from the Libero SoC menu to exit connection mode.

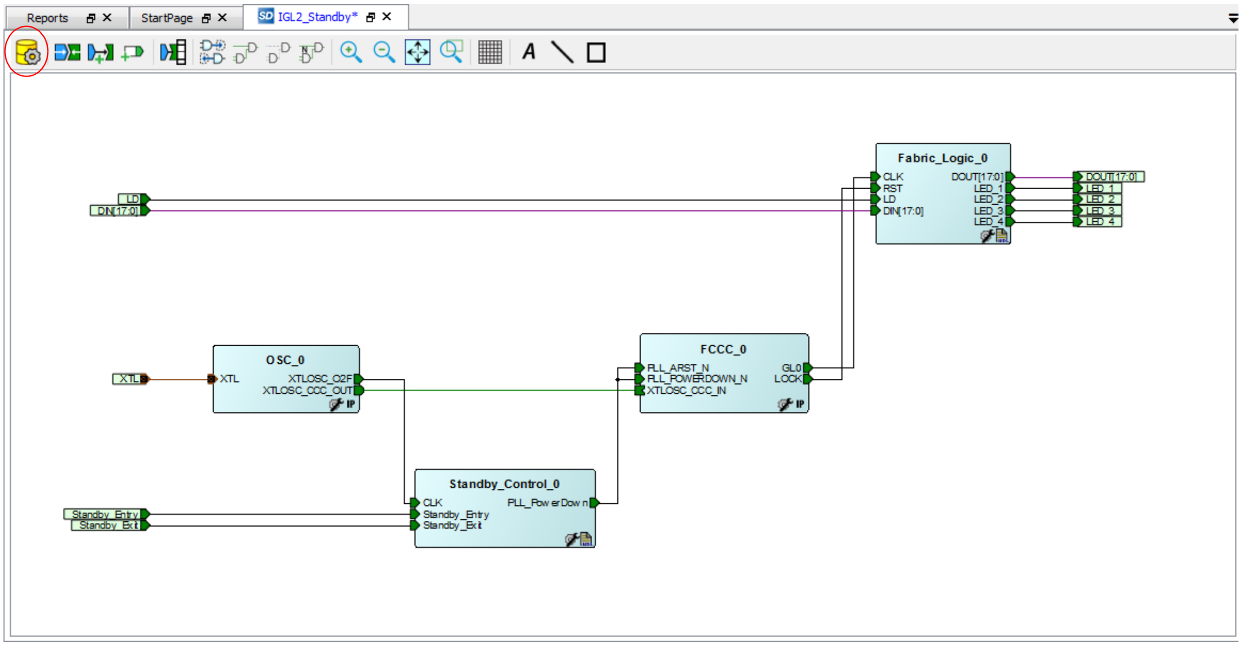
5. Promote the ports shown in Table 4 to the top level. Right-click the port and select **Promote to Top Level**.

Table 4 • **Promote to Top Level**

Ports
Standby_Control_0: Standby_Entry
Standby_Control_0: Standby_Exit
Fabric_Logic_0: LD
Fabric_Logic_0: DIN[17:0]
Fabric_Logic_0: DOUT[17:0]
Fabric_Logic_0: LED_1
Fabric_Logic_0: LED_2
Fabric_Logic_0: LED_3
Fabric_Logic_0: LED_4

The SmartDesign canvas appears, as shown in Figure 18. Drag the components or use the SmartDesign Auto Arrange feature to improve the appearance of the canvas.

Figure 18 • **SmartDesign Canvas after Connections**



6. Go to **File > Save IGL2_Standby**, to save the design.
7. Generate the design by selecting **SmartDesign > Generate Component,** or by clicking the **Generate Component** icon on the SmartDesign toolbar (highlighted in Figure 18).
8. Go to **View > Restore Work Area** to restore the work area, if you expanded the work area earlier.
9. Confirm that the message **IGL2_Standby was generated** appears in the Libero Log window.
10. Go to **File > Close IGL2_Standby** to close the design.

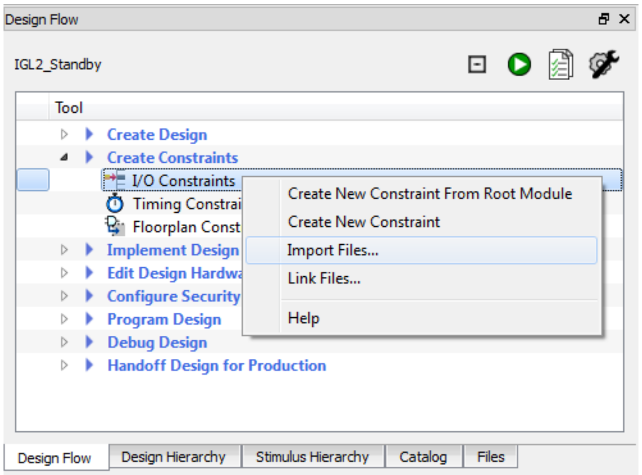
2.5 Importing Physical Constraint files

This section describes how to import a physical design constraint (PDC) file to make I/O attribute and pin assignments for the layout.

The following steps describe how to make I/O assignments:

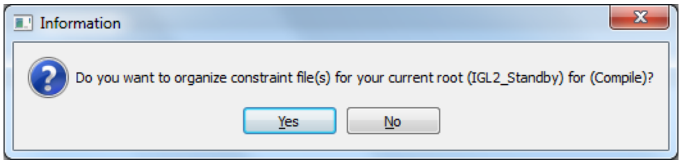
1. Expand **Create Constraints** in the **Design Flow** tab. Right-click **I/O Constraints** and select **Import Files...**.

Figure 19 • **Importing I/O PDC Constraint File**



2. Browse to *<C:\ or D:\>Microsemi_prj\IGL2_Standby_tutorial\Constraints,* select the `IGL2_Standby.pdc` file, and click **Open**.
3. Click **No** in the **Information** dialog box.

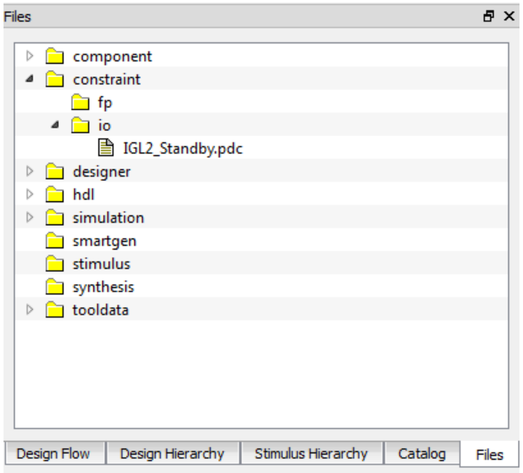
Figure 20 • **Information Dialog Box after Importing PDC Constraint File**



Do you want to organize constraint file(s) for your current root (IGL2_Standby) for (Compile)?

The file is visible in the Libero SoC **Files** tab under **constraint > io**.

Figure 21 • **I/O PDC Constraint File in Libero SoC Project**



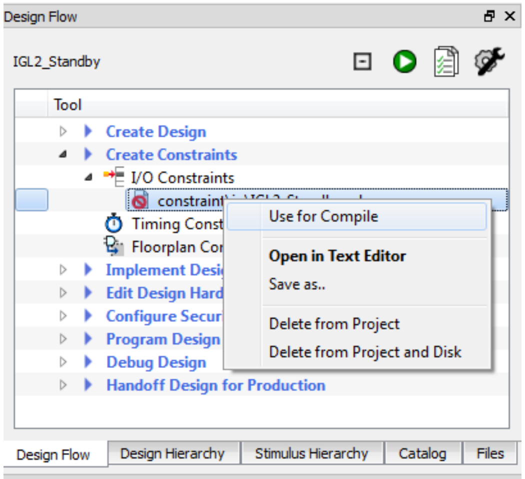
A description of the designer PDC constraints is available in the Libero Help (Go to **Help > Help Topics > Implement Design > Constrain Place and Route > Assigning Design Constraints > Design Constraints Guide > Reference > Constraints by File Format > PDC Command Reference**).

2.6 Synthesis and Layout

Use the push-button flow to synthesize the design with Synplify Pro, run layout, and generate the programming file as mentioned below:

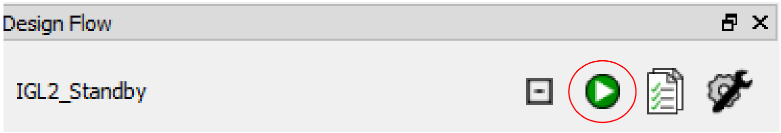
1. Expand **Create Constraints > I/O Constraints** in the Libero SoC **Design Flow** tab. Right-click **IGL2_Standby.pdc** under **Constraints**.
2. Right-click and select **Use for Compile**, as shown in Figure 22. A green tick mark appears on the constraint file indicating that the file will be used.

Figure 22 • **Selecting I/O PDC Constraint File in Design Flow Tab**



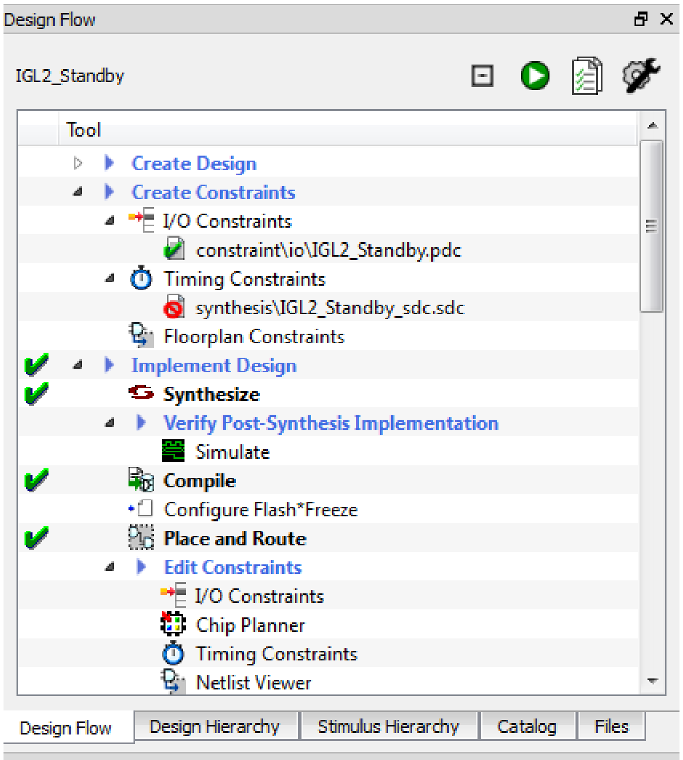
3. Click **Place and Route** icon in the **Design Flow** tab (highlighted in Figure 23), or select **Design > Place and Route** to synthesize the design, and run layout using the I/O constraints that are created.

Figure 23 • **Place and Route**



The design implementation tools run in the batch mode. Successful completion of a design step is indicated by a green tick mark next to **Implement Design** in the **Design Flow** tab, as shown in Figure 24.

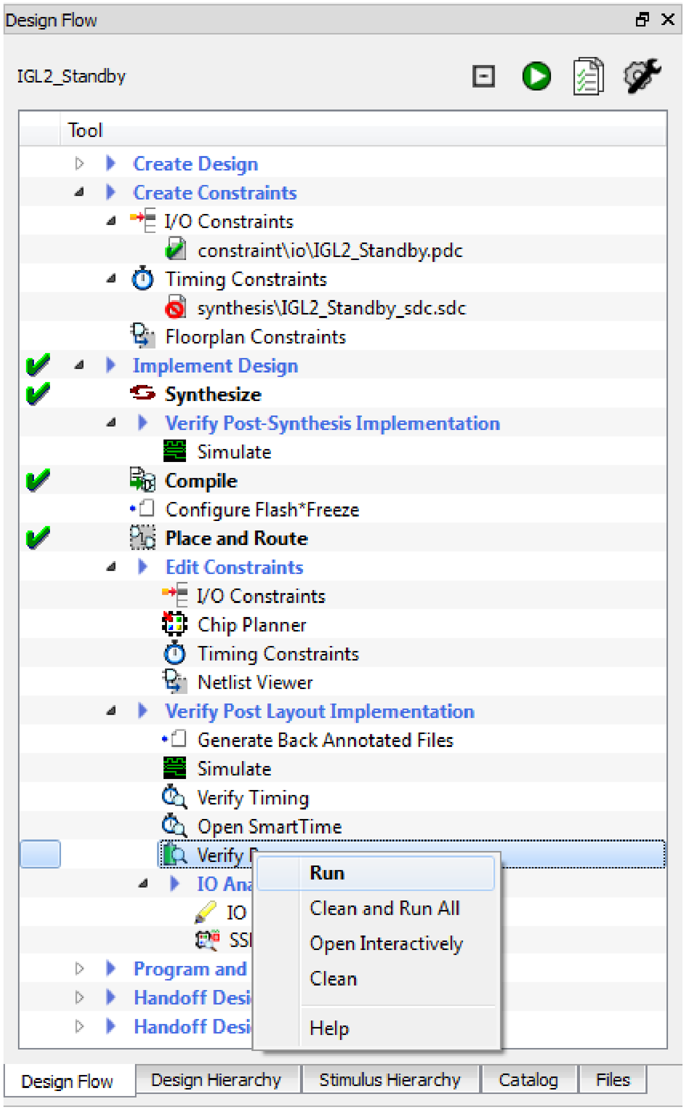
Figure 24 • **Successful Design Implementation**



- Create Design
- Create Constraints
  - I/O Constraints
    - constraint\io\IGL2_Standby.pdc
  - Timing Constraints
    - synthesis\IGL2_Standby_sdc.sdc
  - Floorplan Constraints
- ✓ **Implement Design**
  - ✓ **Synthesize**
  - Verify Post-Synthesis Implementation
    - Simulate
  - ✓ **Compile**
  - Configure Flash*Freeze
  - ✓ **Place and Route**
  - Edit Constraints
    - I/O Constraints
    - Chip Planner
    - Timing Constraints
    - Netlist Viewer

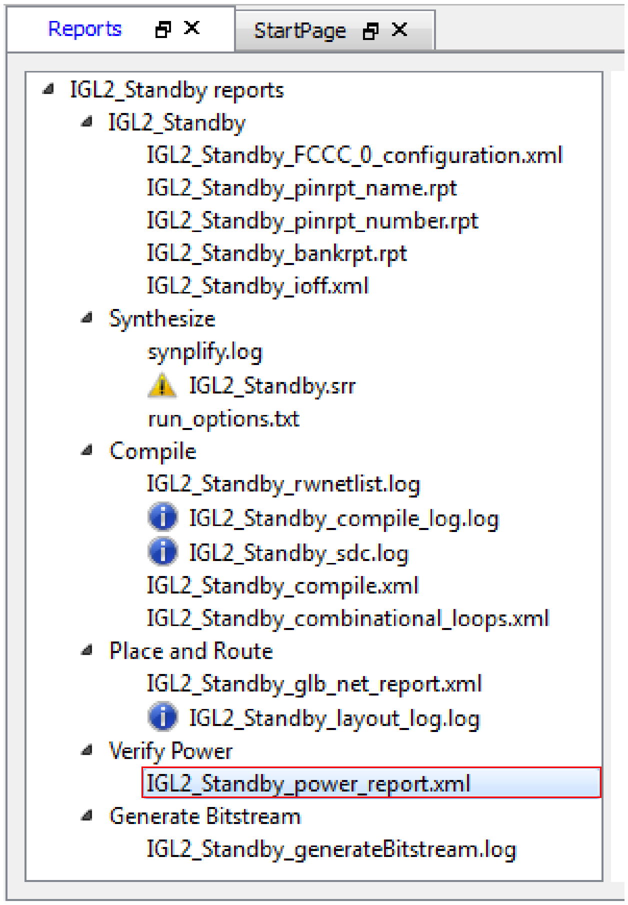
4. Generate a power report by right-clicking **Verify Power** under **Verify Post Layout Implementation** in the **Design Flow** tab and selecting **Run**.

Figure 25 • **Generating Post Layout Power Report**



5. The Reports tab displays reports for the tools used to implement the design. Select `IGL2_Standby_power_report.xml` under Verify Power in the Reports tab to view the power consumption.

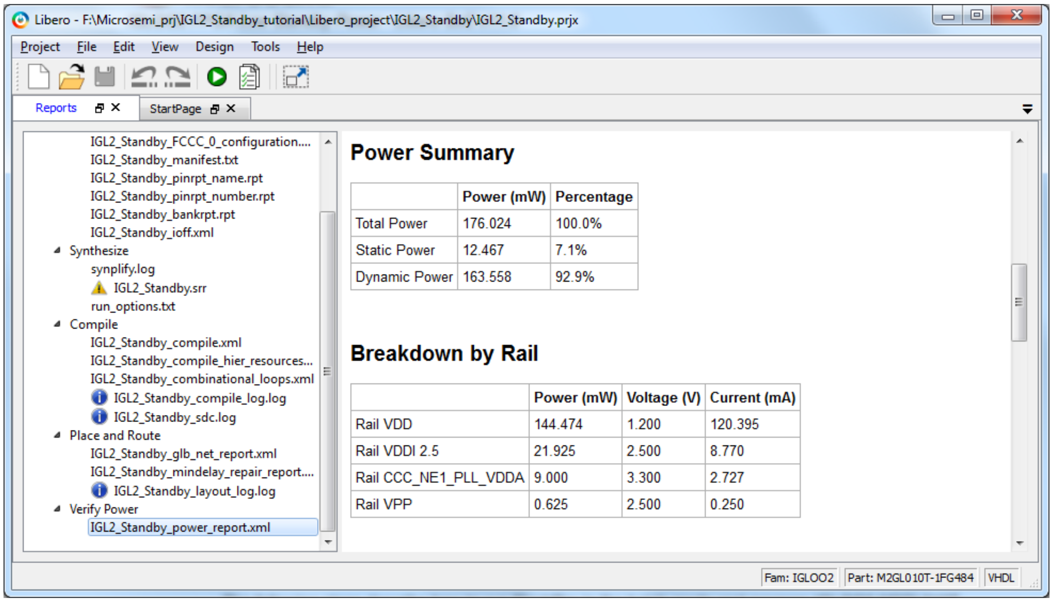
Figure 26 • **Reports Tab after Implementing Design**



- IGL2_Standby reports
  - IGL2_Standby
    - IGL2_Standby_FCCC_0_configuration.xml
    - IGL2_Standby_pinrpt_name.rpt
    - IGL2_Standby_pinrpt_number.rpt
    - IGL2_Standby_bankrpt.rpt
    - IGL2_Standby_ioff.xml
  - Synthesize
    - synplify.log
    - IGL2_Standby.srr
    - run_options.txt
  - Compile
    - IGL2_Standby_rwnetlist.log
    - IGL2_Standby_compile_log.log
    - IGL2_Standby_sdc.log
    - IGL2_Standby_compile.xml
    - IGL2_Standby_combinational_loops.xml
  - Place and Route
    - IGL2_Standby_glb_net_report.xml
    - IGL2_Standby_layout_log.log
  - Verify Power
    - IGL2_Standby_power_report.xml
  - Generate Bitstream
    - IGL2_Standby_generateBitstream.log

The **Reports** tab displays the power report, as shown in Figure 27.

Figure 27 • **Power Report**


Libero - F:\Microsemi_prj\IGL2_Standby_tutorial\Libero_project\IGL2_Standby\IGL2_Standby.prjx
Project
File
Edit
View
Design
Tools
Help
Reports
StartPage
Fam: IGLOO2
Part: M2GL010T-1FG484
VHDL

- IGL2_Standby_FCCC_0_configuration....
- IGL2_Standby_manifest.txt
- IGL2_Standby_pinrpt_name.rpt
- IGL2_Standby_pinrpt_number.rpt
- IGL2_Standby_bankrpt.rpt
- IGL2_Standby_ioff.xml
- Synthesize
  - synplify.log
  - IGL2_Standby.srr
  - run_options.txt
- Compile
  - IGL2_Standby_compile.xml
  - IGL2_Standby_compile_hier_resources...
  - IGL2_Standby_combinational_loops.xml
  - IGL2_Standby_compile_log.log
  - IGL2_Standby_sdc.log
- Place and Route
  - IGL2_Standby_glb_net_report.xml
  - IGL2_Standby_mindelay_repair_report....
  - IGL2_Standby_layout_log.log
- Verify Power
  - IGL2_Standby_power_report.xml

Power Summary

	Power (mW)	Percentage
Total Power	176.024	100.0%
Static Power	12.467	7.1%
Dynamic Power	163.558	92.9%

Breakdown by Rail

	Power (mW)	Voltage (V)	Current (mA)
Rail VDD	144.474	1.200	120.395
Rail VDDI 2.5	21.925	2.500	8.770
Rail CCC_NE1_PLL_VDDA	9.000	3.300	2.727
Rail VPP	0.625	2.500	0.250

2.7 Programming

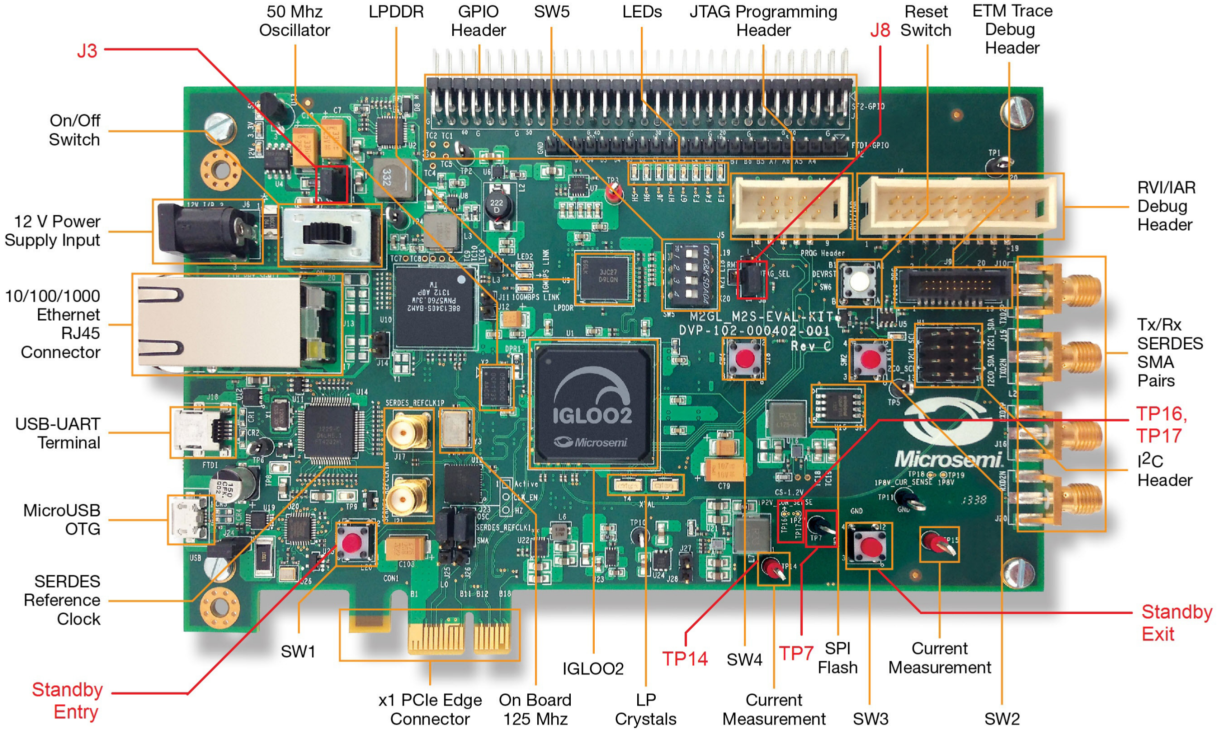
The following steps describe how to run FlashPro in the batch mode and program IGLOO2 M2GL010T on the IGLOO2 Evaluation Kit board:

1. Prior to programming and powering up the IGLOO2 Evaluation Kit board, ensure that the jumpers are positioned, as shown in Table 5.

Table 5 • **Jumper Settings**

Jumper	Location	Setting
J3	Above the On/Off Switch in Figure 28	1-2 installed
J8	Below the JTAG Programming Header (J5) in Figure 28	1-2 installed

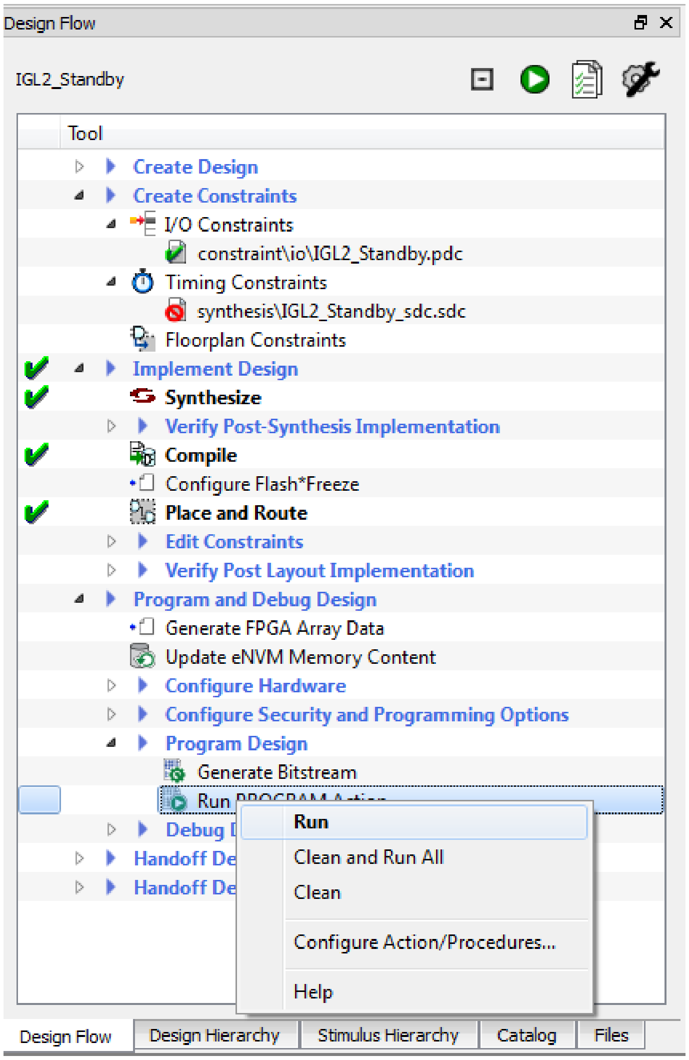
Figure 28 • **IGLOO2 Evaluation Kit**



2. Plug the FlashPro4 ribbon cable into connector J5 (JTAG Programming Header) on the IGLOO2 Evaluation Kit board.
3. Connect FlashPro4 to the USB port of the PC using the mini USB cable.
4. Install the FlashPro4 drivers if prompted. The drivers are located at: *<FlashPro Installation Directory>\Drivers*.
5. Power On the board by plugging in the power cable and switching on the power switch. Three green LEDs on the top left of the board are powered on.

6. In the **Design Flow** tab, expand **Program and Debug Design > Program Design**. Right-click **Run PROGRAM Action** and select **Run** to begin programming.

Figure 29 • **Launching Programming Software from Design Flow Tab**



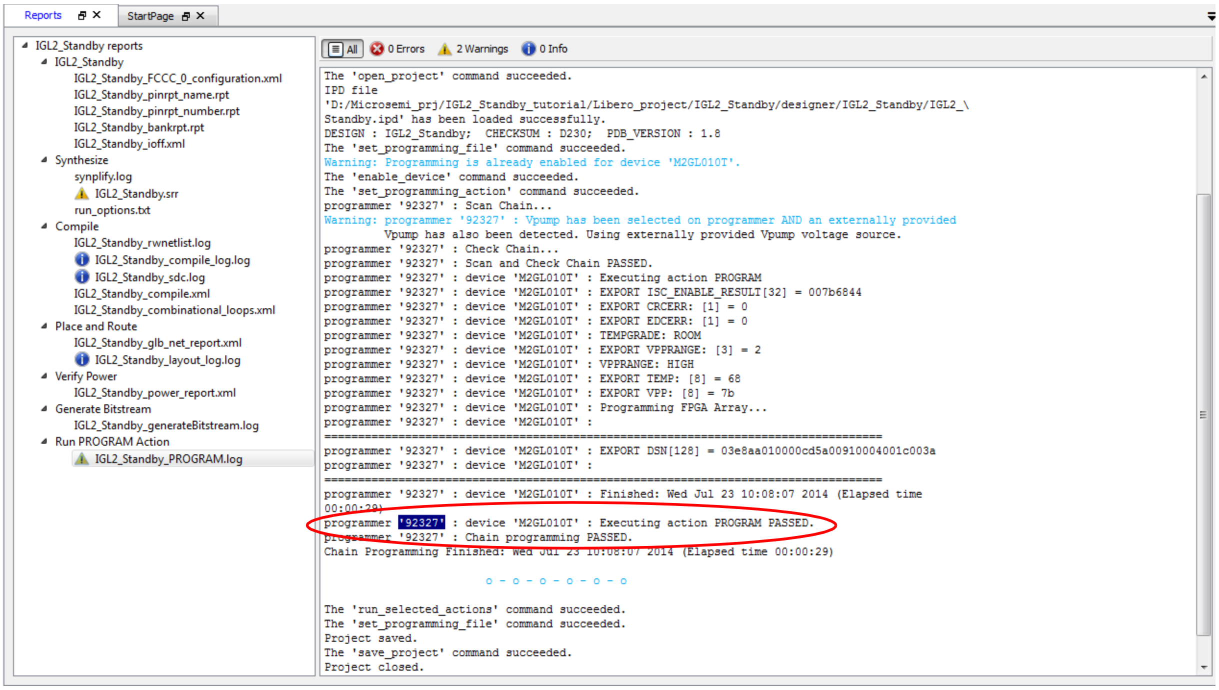
FlashPro runs in the batch mode and programs the device. Programming messages are visible in the Libero SoC log window. Programmer number differs.

Note: Do not interrupt the programming sequence. It may damage the device or programmer.

The following message is displayed in the Reports view under Program Device when the device is programmed successfully, as shown in Figure 30. Programmer number differs:

```
programmer '92327' : device 'M2GL010T' : Executing action PROGRAM PASSED.
```

Figure 30 • **Programming Messages in Libero SoC Log Window**



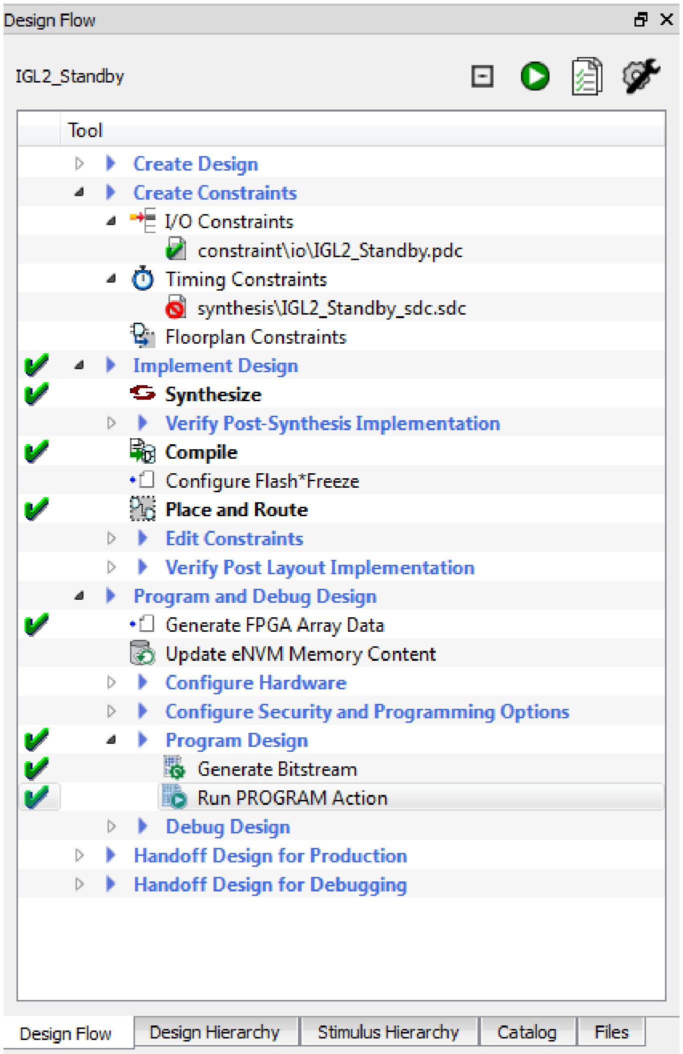
```
The 'open_project' command succeeded.
IPD file
'D:/Microsemi_prj/IGL2_Standby_tutorial/Libero_project/IGL2_Standby/designer/IGL2_Standby/IGL2_\
Standby.ipd' has been loaded successfully.
DESIGN : IGL2_Standby;  CHECKSUM : D230;  PDB_VERSION : 1.8
The 'set_programming_file' command succeeded.
Warning: Programming is already enabled for device 'M2GL010T'.
The 'enable_device' command succeeded.
The 'set_programming_action' command succeeded.
programmer '92327' : Scan Chain...
Warning: programmer '92327' : Vpump has been selected on programmer AND an externally provided
         Vpump has also been detected. Using externally provided Vpump voltage source.
programmer '92327' : Check Chain...
programmer '92327' : Scan and Check Chain PASSED.
programmer '92327' : device 'M2GL010T' : Executing action PROGRAM
programmer '92327' : device 'M2GL010T' : EXPORT ISC_ENABLE_RESULT[32] = 007b6844
programmer '92327' : device 'M2GL010T' : EXPORT CRCERR: [1] = 0
programmer '92327' : device 'M2GL010T' : EXPORT EDCERR: [1] = 0
programmer '92327' : device 'M2GL010T' : TEMPGRADE: ROOM
programmer '92327' : device 'M2GL010T' : EXPORT VPPRANGE: [3] = 2
programmer '92327' : device 'M2GL010T' : VPPRANGE: HIGH
programmer '92327' : device 'M2GL010T' : EXPORT TEMP: [8] = 68
programmer '92327' : device 'M2GL010T' : EXPORT VPP: [8] = 7b
programmer '92327' : device 'M2GL010T' : Programming FPGA Array...
programmer '92327' : device 'M2GL010T' :
==================================================================================
programmer '92327' : device 'M2GL010T' : EXPORT DSN[128] = 03e8aa010000cd5a00910004001c003a
programmer '92327' : device 'M2GL010T' :
==================================================================================
programmer '92327' : device 'M2GL010T' : Finished: Wed Jul 23 10:08:07 2014 (Elapsed time
00:00:29)
programmer '92327' : device 'M2GL010T' : Executing action PROGRAM PASSED.
programmer '92327' : Chain programming PASSED.
Chain Programming Finished: Wed Jul 23 10:08:07 2014 (Elapsed time 00:00:29)

                         o - o - o - o - o - o

The 'run_selected_actions' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
```

7. A green tick mark appears next to **Program Design** in the **Design Flow** tab indicating that programming is completed successfully.

Figure 31 • **Design Flow Tab after Programming**



Design Flow

IGL2_Standby

Tool

- Create Design
- Create Constraints
  - I/O Constraints
    - constraint\io\IGL2_Standby.pdc
  - Timing Constraints
    - synthesis\IGL2_Standby_sdc.sdc
  - Floorplan Constraints
- ✓ Implement Design
  - ✓ Synthesize
  - Verify Post-Synthesis Implementation
  - ✓ Compile
  - Configure Flash*Freeze
  - ✓ Place and Route
  - Edit Constraints
  - Verify Post Layout Implementation
- Program and Debug Design
  - ✓ Generate FPGA Array Data
  - Update eNVM Memory Content
  - Configure Hardware
  - Configure Security and Programming Options
  - ✓ Program Design
    - ✓ Generate Bitstream
    - ✓ Run PROGRAM Action
  - Debug Design
- Handoff Design for Production
- Handoff Design for Debugging

8. Select **Project > Exit** to close Libero SoC. Select **Yes**, if prompted about saving the changes.

2.8 Running the Demo Design

2.8.1 Power Measurement (Normal Operation and Standby)

The IGLOO2 Evaluation Kit board has a voltage measuring circuit that measures the voltage across the VDD (1.2 V) current sense resistor.

The core power can be calculated using the following equations:

$$\text{Core Current (mA)} = \text{Measured Voltage (mV)} \div 5(\text{Scaling Factor})$$

EQ 1

$$\text{Core Power (mW)} = 1.2 \times \text{Core Current}$$

EQ 2

Connect the positive terminal of a standard digital voltmeter (DVM)/multimeter to TP14 and negative terminal to TP7.

Note the digital voltmeter/multimeter reading and calculate the power using the above equations.

2.8.2 Precise Standby Power Measurement

Precise and accurate power measurements can be obtained by measuring voltage across the 1.2 V, 0.05 Ω sense resistor. Test points TP16 and TP17 can be used to directly measure voltage across the 1.2 V sense resistor. Since the current drawn by the device in standby mode is expected to be around or less than 10 mA, the voltage measured across the 0.05 Ω sense resistor is expected to be less than 0.5 mV. A precise digital voltmeter such as Fluke-287 that can measure sub-millivolt readings must be used to read voltage measured across the sense resistor.

Convert the voltage measured across sense resistor to power using the following equation:

$$\text{Power}\ \ (\text{mW})\ = (\text{Voltage}(\text{mV})/0.05) \times 1.2$$

EQ 3

2.8.3 Total Power (Dynamic and Static)

The following steps describe how to calculate total power:

1. Reset the board by pressing and releasing the Reset button (SW6 DEVRST).
2. Observe the pattern of the LEDs E1, F4, F3, and G7 after resetting the board.
3. Measure the power.

Note: If the LEDs are not toggling after reset, the device is in the Standby mode. Press and release the standby exit push button (SW3) and observe the LEDs lighting pattern. When the LEDs start toggling, measure the power.

2.8.4 Standby Power

The following steps describe how to calculate standby power:

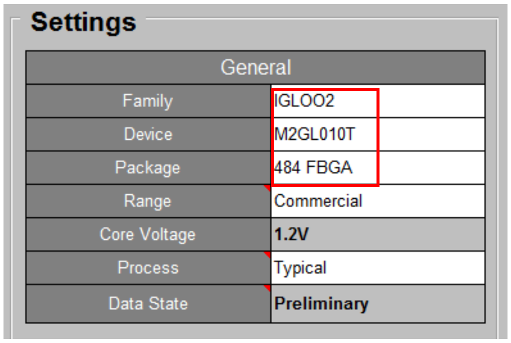
1. Press and release the standby entry push button (SW1) and observe the LEDs lighting pattern. The LEDs stop toggling.
2. Measure the power.
3. Press and release the standby exit push button (SW3).
4. When finished, remove power from the board.

3 Appendix: Power Estimator

The following steps describe how to use Power Estimator and calculate the total power:

1. Download the Power Estimator, *SmartFusion2 and IGLOO2 Power Calculator.*
2. Double-click and open the power estimator spreadsheet.
3. Click the Summary worksheet. The Summary worksheet provides the device settings and the power summary.
4. Change the device settings by entering the following information:
   - **Family**: Select **IGLOO2** from the drop-down list
   - **Device**: Select **M2GL010T** from the drop-down list
   - **Package**: Select **484 FBGA** from the drop-down list

Figure 32 • **Settings Section in the Device Settings and Summary Worksheet**



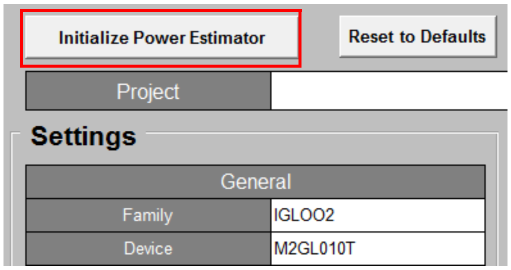
Settings

General	
Family	IGLOO2
Device	M2GL010T
Package	484 FBGA
Range	Commercial
Core Voltage	**1.2V**
Process	Typical
Data State	**Preliminary**

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 information about the design and performs power estimation for the design.

5. Click **Initialize Power Estimator**, as shown in Figure 33. The **Initialize Power Estimator** dialog box opens, as shown in Figure 34 on page 32.

Figure 33 • **Initialize Power Estimator**



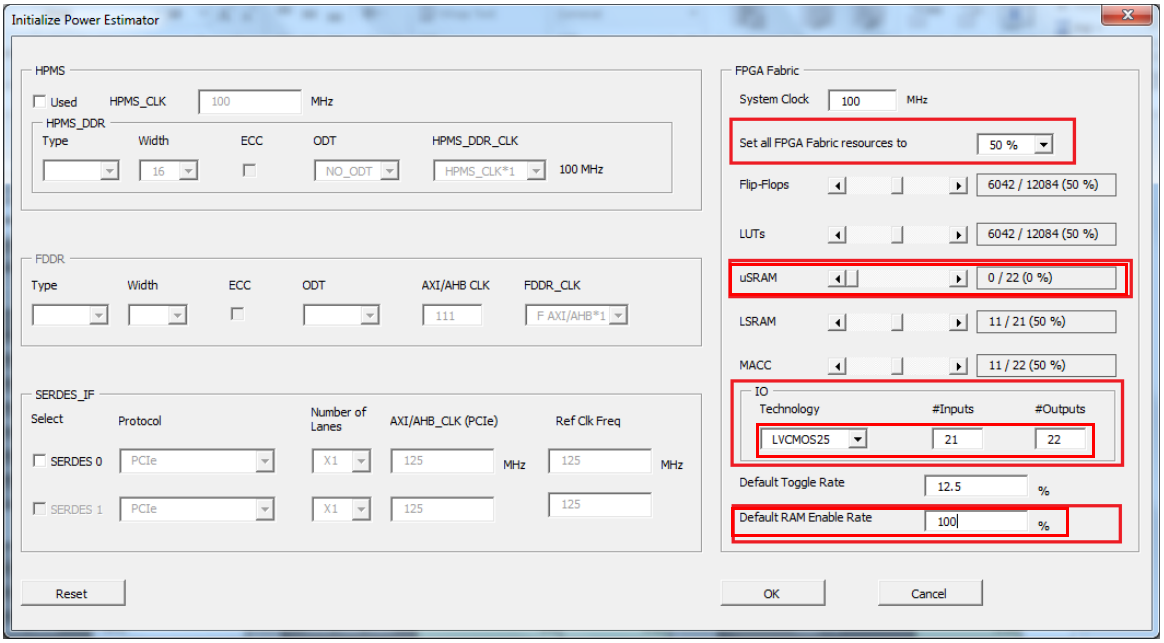
Project	

Settings

General	
Family	IGLOO2
Device	M2GL010T

6. Enter the following information in the **Initialize Power Estimator** dialog box:
   - **Set all FPGA fabric resources to**: 50%
   - **uSRAM**: Move the slider to zero, 0/22 (0%)
   - **IO**:
     - **Technology**: LVCMOS25
     - **#Inputs**: 21
     - **#Outputs**: 22
   - **Default RAM Enable Rate**: 100%

Figure 34 • Initialize Power Estimator Wizard



Initialize Power Estimator

HPMS

□ Used   HPMS_CLK 100 MHz

HPMS_DDR

Type	Width	ECC	ODT	HPMS_DDR_CLK	
	16	□	NO_ODT	HPMS_CLK*1	100 MHz

FDDR

Type	Width	ECC	ODT	AXI/AHB CLK	FDDR_CLK
		□		111	F AXI/AHB*1

SERDES_IF

Select	Protocol	Number of Lanes	AXI/AHB_CLK (PCIe)		Ref Clk Freq	
□ SERDES 0	PCIe	X1	125	MHz	125	MHz
□ SERDES 1	PCIe	X1	125		125	

FPGA Fabric

System Clock 100 MHz

Set all FPGA Fabric resources to 50 %

Resource	Value
Flip-Flops	6042 / 12084 (50 %)
LUTs	6042 / 12084 (50 %)
uSRAM	0 / 22 (0 %)
LSRAM	11 / 21 (50 %)
MACC	11 / 22 (50 %)

IO

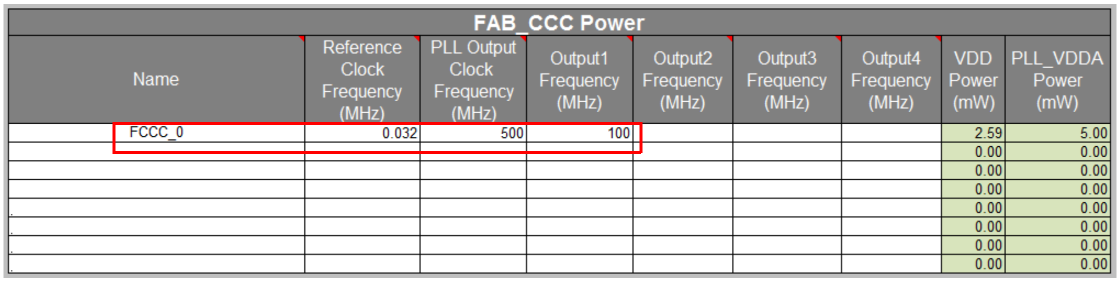
Technology	#Inputs	#Outputs
LVCMOS25	21	22

Default Toggle Rate 12.5 %

Default RAM Enable Rate 100 %

7. Click **OK**. Click **Yes** in the **Reset and set to the values specified** dialog box.
8. Click the **CCC & Oscillator** worksheet and scroll down to the **FAB_CCC Power** section. Enter the following information in the **FAB_CCC Power** table:
   - **Name**: FCC_0
   - **Reference Clock Frequency (MHz)**: 0.032
   - **PLL Output Clock Frequency (MHz)**: 500 MHz
   - **Output1 Frequency (MHz)**:100 MHz

Figure 35 • FAB_CCC Section

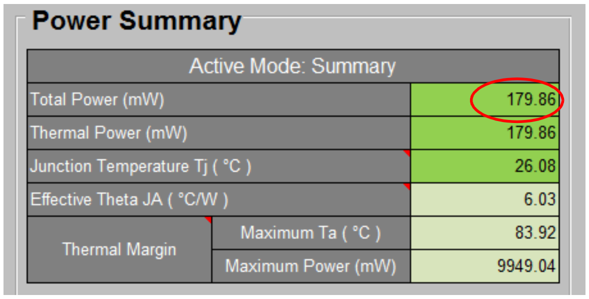


FAB_CCC Power

Name	Reference Clock Frequency (MHz)	PLL Output Clock Frequency (MHz)	Output1 Frequency (MHz)	Output2 Frequency (MHz)	Output3 Frequency (MHz)	Output4 Frequency (MHz)	VDD Power (mW)	PLL_VDDA Power (mW)
FCCC_0	0.032	500	100				2.59	5.00
							0.00	0.00
							0.00	0.00
							0.00	0.00
							0.00	0.00
							0.00	0.00
							0.00	0.00
							0.00	0.00

9. Click the **Summary** worksheet to get the total power. The **Power Summary** section is populated with the Total Active Mode power.

Figure 36 • Power Summary

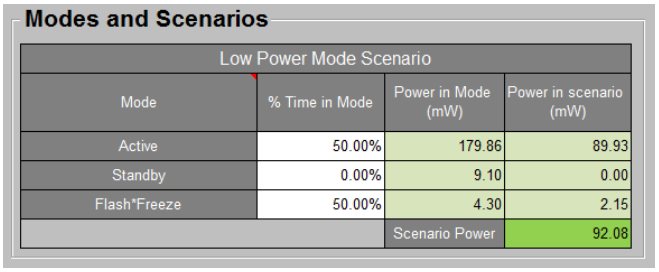


Power Summary

Active Mode: Summary		
Total Power (mW)		179.86
Thermal Power (mW)		179.86
Junction Temperature Tj (°C)		26.08
Effective Theta JA (°C/W)		6.03
Thermal Margin	Maximum Ta (°C)	83.92
	Maximum Power (mW)	9949.04

10. The **Modes and Scenarios** section is populated with the total power in the Active, Standby, and Flash*Freeze modes.

Figure 37 • **Modes and Scenarios**



Modes and Scenarios

Low Power Mode Scenario			
Mode	% Time in Mode	Power in Mode (mW)	Power in scenario (mW)
Active	50.00%	179.86	89.93
Standby	0.00%	9.10	0.00
Flash*Freeze	50.00%	4.30	2.15
		Scenario Power	92.08

11. Close **Power Estimator**.

4 Revision History

The following table shows important changes made in this document for each revision.

Revision	Changes
Revision 5 (March 2016)	Updated the document for Libero SoC v11.7 software release (SAR 76606).
Revision 4 (October 2015)	Updated the document for Libero SoC v11.6 (SAR 71653).
Revision 3 (February 2015)	Updated the document for Libero SoC v11.5 (SAR 64364).
Revision 2 (August 2014)	Updated the document for Libero SoC v11.4.
Revision 1 (October 2013)	Initial release.

5 Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

5.1 Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 408.643.6913

5.2 Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

5.3 Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

5.4 Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/products/fpga-soc/fpga-and-soc.

5.5 Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

5.5.1 Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

5.5.2 My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

5.5.3 Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Visit About Us for sales office listings and corporate contacts.

5.6 ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

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
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace 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, Calif, and has approximately 4,800 employees globally. Learn more at **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.