SmartFusion2 Fabric Lab Guide

Microsemi

Microsemi SoC Products Group, San Jose, CA 95134

© 2015 Microsemi Corporation. All rights reserved.

Printed in the United States of America

Release: January, 2015

Microsemi makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Microsemi assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Microsemi Corporation.

Trademarks

Microsemi, SmartFusion2, IGLOO2 and the associated logos are trademarks or registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Table of Contents

Introduction

This tutorial demonstrates how to implement a basic SmartFusion2 FPGA fabric design using SmartDesign. The design drives the LEDs on the SmartFusion2 target board with different patterns based on the state of the user input switch as shown in the table below. Refer to the Hardware Requirements section on the next page for a list of target boards.

LED Behavior	Reset Switch	User Switch
LEDs off	Depressed	don't care
LEDs toggle	Released	Released
Both LEDs blinking	Released	Depressed

Table 1 – LED behavior

After completing this tutorial you will be familiar with the following:

- Creating a Libero SoC project
- Implementing a SmartFusion2 fabric design with SmartDesign
- Simulating the design
- Importing a PDC file, running layout and programming the SmartFusion2 silicon

Components of SmartFusion2 Device Used

This tutorial uses the SmartFusion2 FPGA fabric, the on-chip 25/50 MHz RC oscillator and the Fabric CCC.

Tutorial Requirements

Software Requirements

This tutorial requires the following software installed on your computer:

- Microsemi Libero SoC v11.5
- ModelSim 10.3c
- Synplify Pro ME I-2014.03M-SP1
- FlashPro v11.5

Hardware Requirements

This tutorial can be used with the following Microsemi SmartFusion2 boards:

- SmartFusion2 Starter kit with M2S050_ES-FGG896 (SF2-STARTER-KIT-ES-2[1])
- SmartFusion2 Starter kit with M2S050-FGG484 (SF2-STARTER-KIT)
- SmartFusion2 Starter Kit with M2S010-FGG484 (SF2-484-STARTER-KIT)
- SmartFusion2 Evaluation Kit with M2S025T-1FGG484 (M2S-EVAL-KIT)
- SmartFusion2 Security Evaluation Kit with M2S090TS-1FGG484 (M2S090S-EVAL-KIT[4])
- SmartFusion2 Development Kit with M2S050T-1FGG896 (SF2-DEV-KIT-PP[2], SF2-DEV-KIT-PP-1[2] or SF2-DEV-KIT[3])
- SmartFusion2 Advanced Development Kit with M2S150T-1FCG1152ES (M2S150-ADV-DEV-KIT-ES[4])
- IGLOO2 Evaluation kit with M2GL010T-1FGG484 (M2GL-EVAL-KIT)

Notes:

1. SF2-STARTER-KIT-ES-2 has been discontinued. Please refer to PDN1403 for more details.
2. The SF2-DEV-KIT-PP and SF2-DEV-KIT-PP-1 part numbers have been discontinued per BPDN1303 and PDN1307.
3. The SF2-DEV-KIT has been discontinued per PDN1406.
4. The SmartFusion2 Security Evaluation Kit and the Advanced Development Kit require a Libero SoC Platinum license. The Libero SoC Gold or Platinum license can be used for the other kits.

Extracting the source files

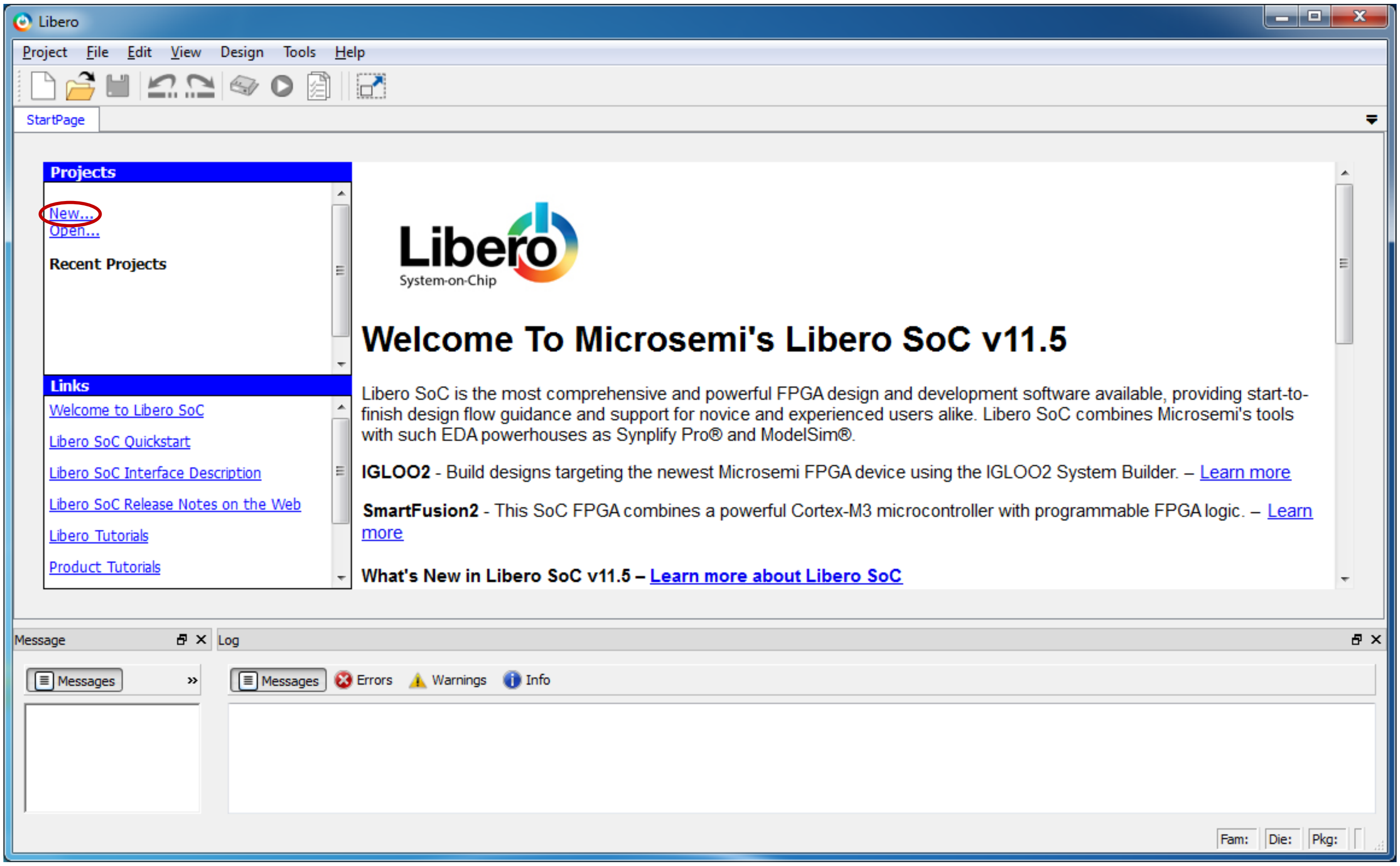
Extract *SF2_Fabric_tutorial.zip* to extract the required lab files to the *<C: or D:>\Microsemiprj* folder on the HDD of your PC. Confirm that a folder named *SF2_Fabric_tutorial* containing a sub-folder named *Source_files* was extracted.

Step 1 – Creating the Design

In this step you will create the fabric design using SmartDesign. Some source files have been provided in the Source_files folder.

Launching Libero SoC and creating a project

1. Click **Start > Programs > Microsemi Libero SoC v11.5 > Libero SoC v11.5**, or click the shortcut on your desktop. The Libero SoC Project Manager will open.



Welcome To Microsemi's Libero SoC v11.5

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®.

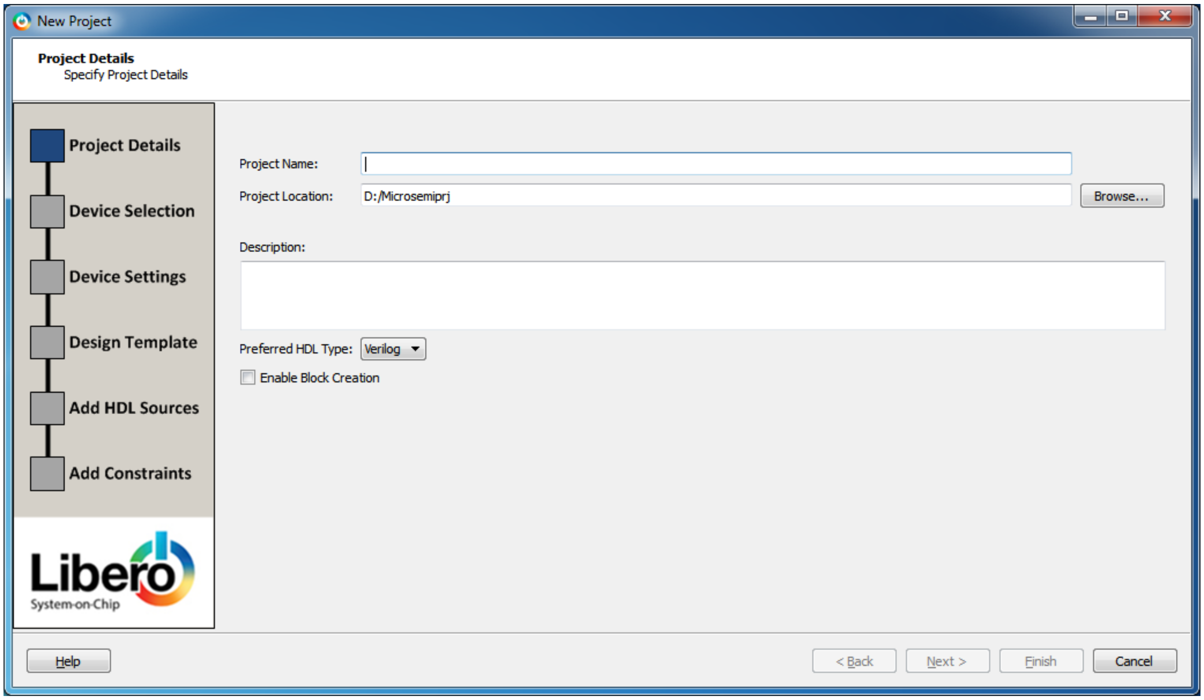
IGLOO2 - Build designs targeting the newest Microsemi FPGA device using the IGLOO2 System Builder. – Learn more

SmartFusion2 - This SoC FPGA combines a powerful Cortex-M3 microcontroller with programmable FPGA logic. – Learn more

What's New in Libero SoC v11.5 – Learn more about Libero SoC

Figure 1 - Libero SoC Project Manager

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



Project Name:

Project Location: D:/Microsemiprj

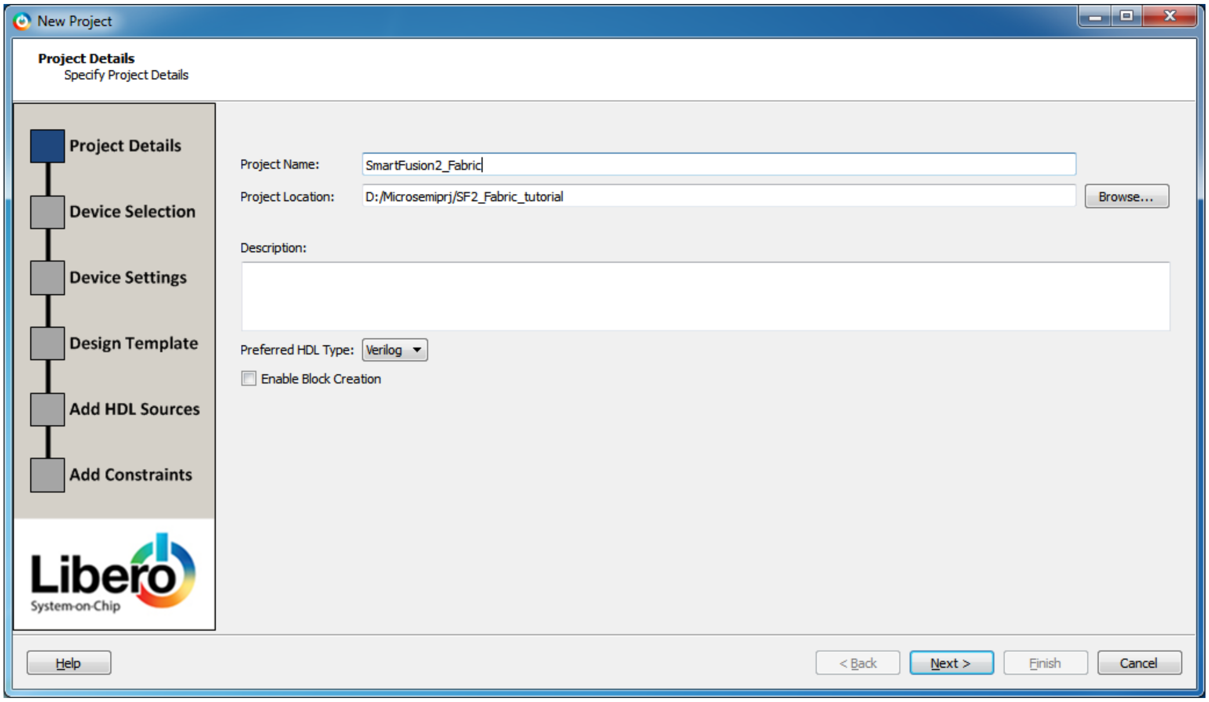
Description:

Preferred HDL Type: Verilog

☐ Enable Block Creation

Figure 2 - Libero SoC New Project dialog box

3. Enter the information shown below in the Project Details page of the New Project dialog box then click **Next**:
   - Project Name: SmartFusion2_Fabric
   - Project Location: <C: or D:>\Microsemiprj\SF2_Fabric_tutorial (depending on where you extracted the source files)
   - Preferred HDL type: Verilog or VHDL
   - Enable Block Creation: un-checked



Project Name: SmartFusion2_Fabric

Project Location: D:/Microsemiprj/SF2_Fabric_tutorial

Description:

Preferred HDL Type: Verilog

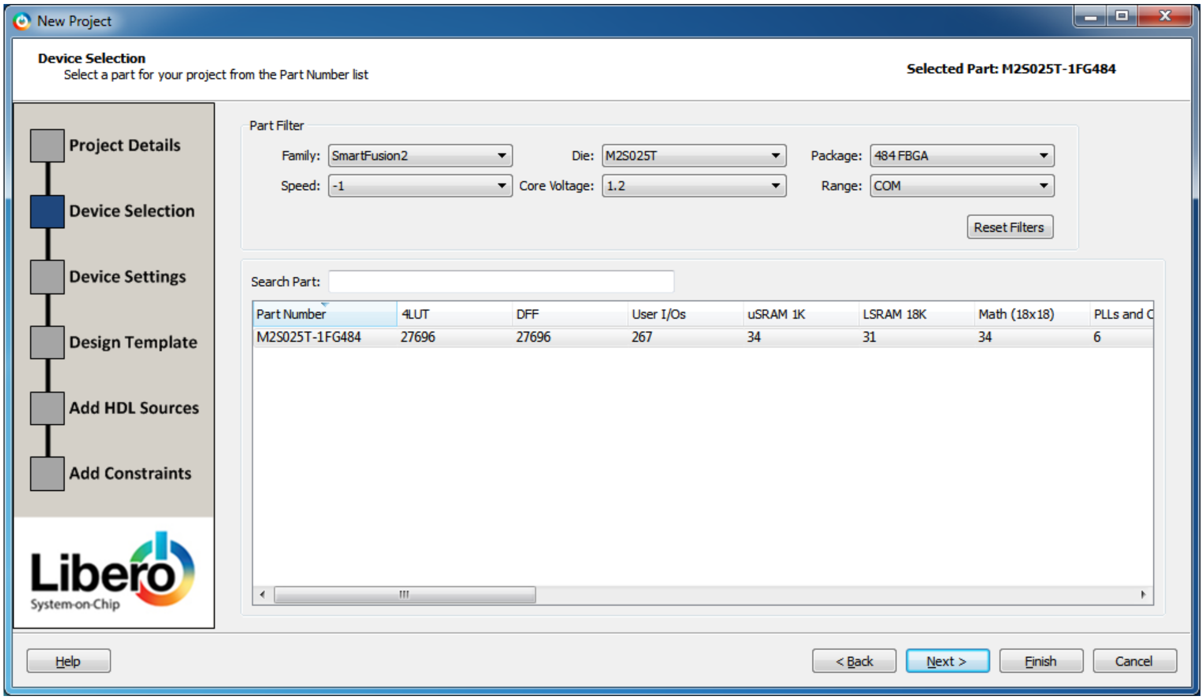
☐ Enable Block Creation

Figure 3 - Project Details

4. Enter the following in the Device Selection page of the New Project dialog box then click **Next**:
   - Family: SmartFusion2
   - Die: Refer to Table 2 below for your specific board
   - Package: Refer to Table 2 below for your specific board
   - Speed: Refer to Table 2 below for your specific board
   - Core Voltage(V): 1.2
   - Range: COM

Board	Die	Package	Speed	PLL Supply Voltage
SF2-STARTER-KIT-ES-2	M2S050T_ES	896 FBGA	STD	3.3
SF2-STARTER-KIT	M2S050	484 FBGA	STD	2.5
SF2-484-STARTER-KIT	M2S010	484 FBGA	STD	2.5
M2S-EVAL-KIT	M2S025T	484 FBGA	-1	3.3
M2S090S-EVAL-KIT	M2S090TS	484 FBGA	-1	3.3
SF2-DEV-KIT-PP	M2S050T	896 FBGA	-1	3.3
SF2-DEV-KIT-PP-1	M2S050T	896 FPGA	-1	3.3
SF2-DEV-KIT	M2S050T	896 FBGA	-1	3.3
M2S150-ADV-DEV-KIT-ES	M2S150T	1152 FC	-1	3.3
M2GL-EVAL-KIT	M2GL010T	484 FBGA	-1	3.3

Table 2 – Die, Package, Speed Grade and PLL settings for the supported target boards


New Project

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

Selected Part: M2S025T-1FG484

Part Filter

Family: SmartFusion2
Die: M2S025T
Package: 484 FBGA
Speed: -1
Core Voltage: 1.2
Range: COM

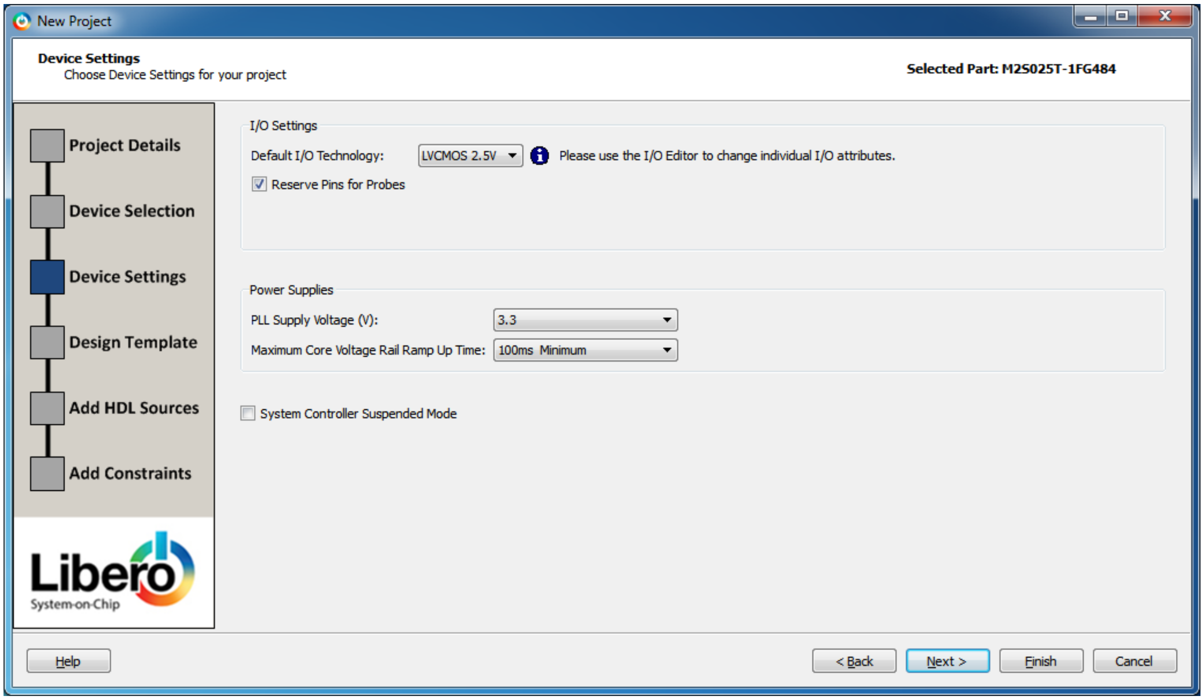
Search Part:

Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and C
M2S025T-1FG484	27696	27696	267	34	31	34	6

Figure 4 - Device selection (settings for M2S-EVAL-KIT board shown)

5. Enter the following in the Device Settings page of the New Project dialog box then click **Next**:
   - Default I/O Technology: LVCMOS 2.5V (default)
   - Reserve Pins for Probes: checked (default)
   - Power Supplies
     - PLL Supply Voltage (V): Refer to Table 2 for your specific board
     - Ramp rate: 100ms Minimum (default)
   - System Controller Suspend Mode: un-checked

The PLL Supply voltage can be either 2.5V or 3.3V. The voltage setting in the New Project dialog box must match the PLL Analog Supply voltage on the board to ensure the PLL works correctly. The PLL Analog Supply voltage is tied to 3.3V on some of the target boards, so the setting must be changed. Refer to Table 2 for the PLL Supply voltage for your target board. Change the supply voltage using the pull-down menu if needed.



New Project

Device Settings
Choose Device Settings for your project

Selected Part: M2S025T-1FG484

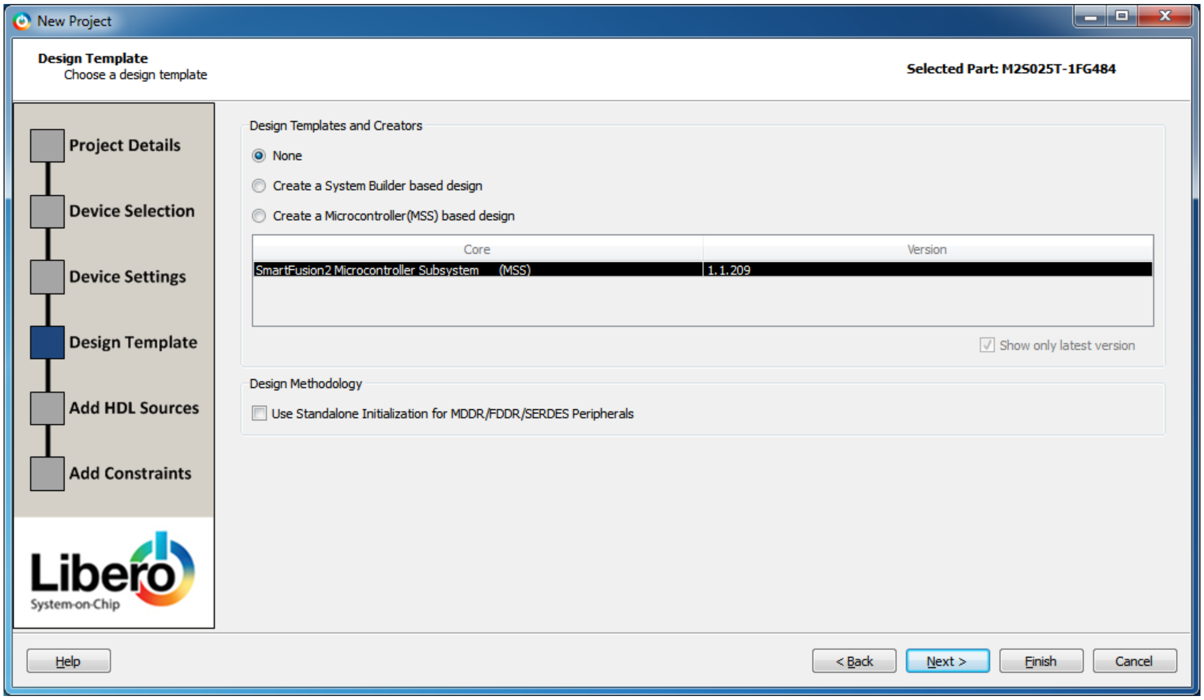
I/O Settings
Default I/O Technology: LVCMOS 2.5V
Please use the I/O Editor to change individual I/O attributes.
☑ Reserve Pins for Probes

Power Supplies
PLL Supply Voltage (V): 3.3
Maximum Core Voltage Rail Ramp Up Time: 100ms Minimum

☐ System Controller Suspended Mode

Figure 5 - Device Settings (settings for M2S-EVAL-KIT board shown)

6. Enter the following in the Design Template page of the New Project dialog box then click **Next**:
   - Design Templates and Creators: None (this design does not use the SmartFusion MSS)
   - Design Methodology:
     - Use Standalone Initialization for MDDR/FDDR/SERDES peripherals: un-checked



New Project

Design Template
Choose a design template

Selected Part: M2S025T-1FG484

Design Templates and Creators
● None
○ Create a System Builder based design
○ Create a Microcontroller(MSS) based design

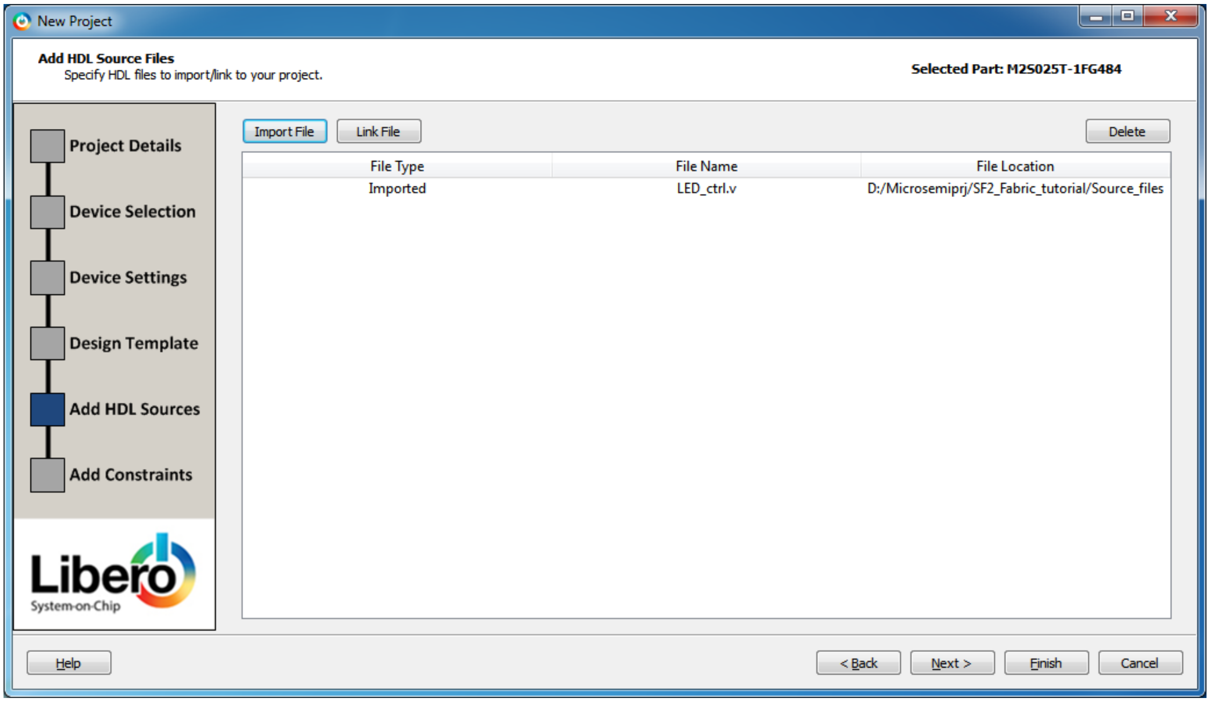
Core	Version
SmartFusion2 Microcontroller Subsystem (MSS)	1.1.209

☑ Show only latest version

Design Methodology
☐ Use Standalone Initialization for MDDR/FDDR/SERDES Peripherals

Figure 6 - Design Template settings

7. The Add HDL Source Files page will be visible. A VHDL and Verilog source file has been provided. Import the files into the project by clicking **Import File**.
8. Enter the following in the Import Files dialog box then click **Open**:
   - Location: <C or D:>\Microsemiprj\SF2_Fabric_tutorial\Source_files
   - File name: LED_ctrl.vhd (for VHDL projects) or LED_ctrl.v (for Verilog projects)
   - Files of type: HDL Source Files (*.vhd *.v *.h)



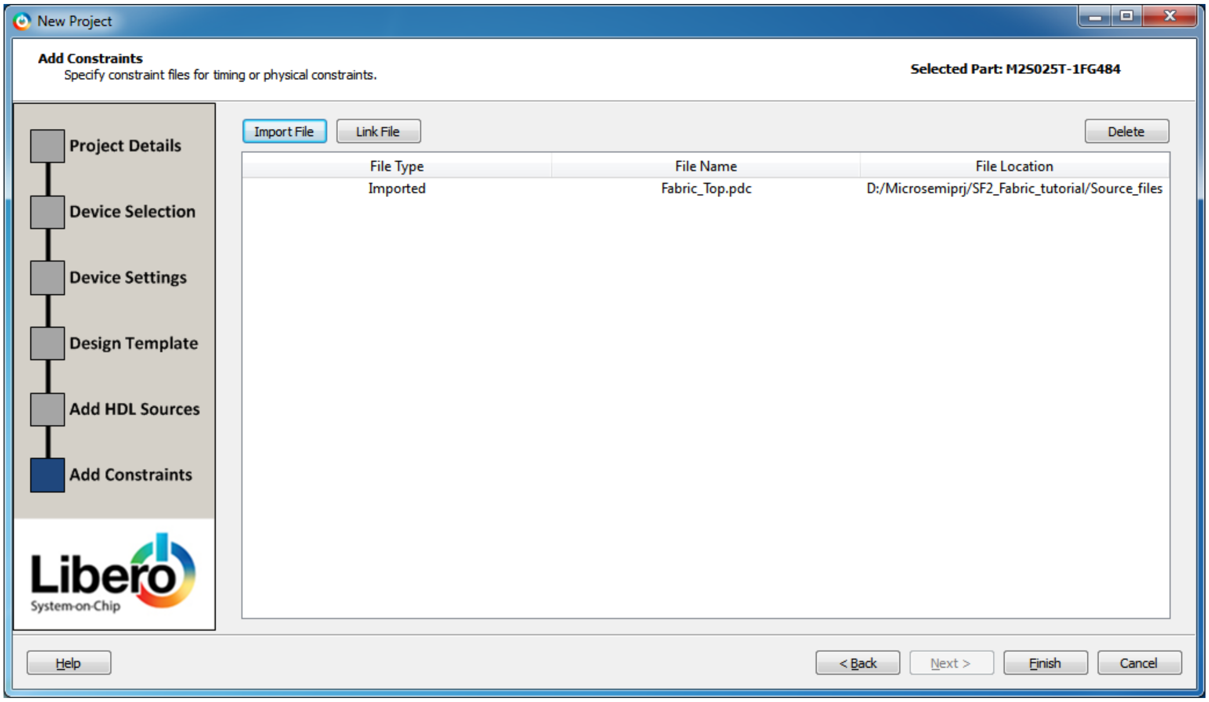
Add HDL Source Files
Specify HDL files to import/link to your project.

Selected Part: M2S025T-1FG484

File Type	File Name	File Location
Imported	LED_ctrl.v	D:/Microsemiprj/SF2_Fabric_tutorial/Source_files

Figure 7 - Imported HDL source files

9. Click **Next**. The Add Constraints page will be visible. An I/O constraint file has been provided. Import the file into the project by clicking **Import File**.
10. Enter the following in the Import Files dialog box then click **Open**:
    - Location: <C or D:>\Microsemiprj\SF2_Fabric_tutorial\Source_files
    - File name: Fabric_Top.pdc
    - Files of type: I/O Constraint Files (*.pdc)



New Project

Add Constraints

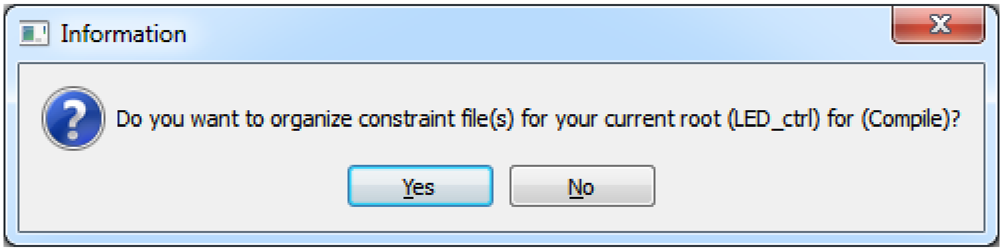
Specify constraint files for timing or physical constraints.

Selected Part: M2S025T-1FG484

File Type	File Name	File Location
Imported	Fabric_Top.pdc	D:/Microsemiprj/SF2_Fabric_tutorial/Source_files

Figure 8 - Imported I/O Constraint file

11. Click **Finish** in the New Project dialog box. Select **No** in the Information dialog box when prompted about organizing the constraint files for compile.



Information

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

Figure 9 - Information dialog box after importing the PDC constraint file

12. The HLD source file will be visible on the Design Hierarchy tab and the Files tab. The I/O constraint file will also be visible on the Files tab.

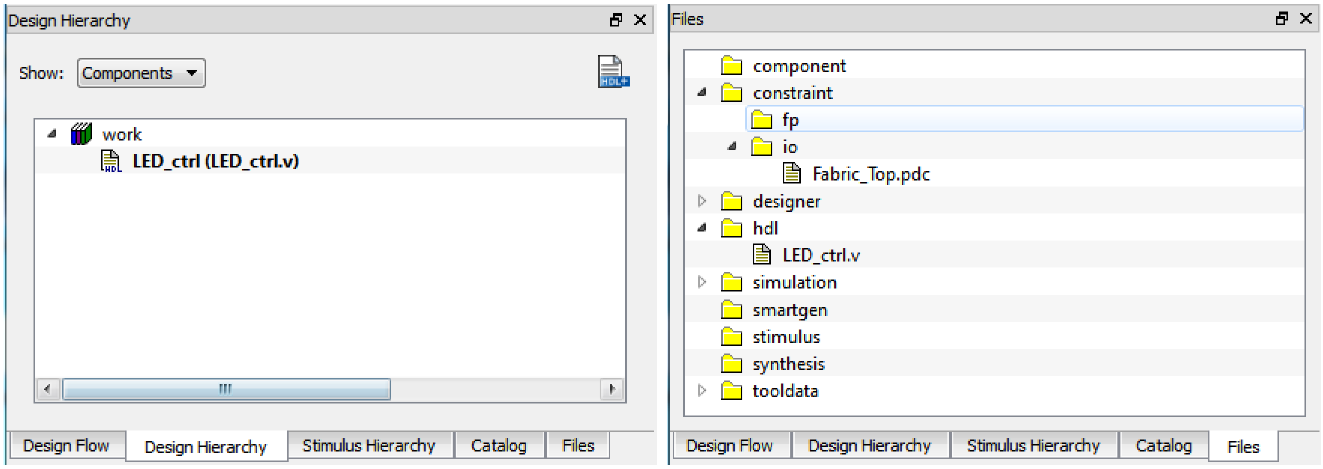


Figure 10 - HDL source file and I/O constraint file in Libero SoC (Verilog shown)

Implementing the design with SmartDesign

7. Open the SmartDesign canvas by selecting **File > New > SmartDesign** from the menu or by double-clicking **Create SmartDesign** under Create Design on the Design Flow tab.

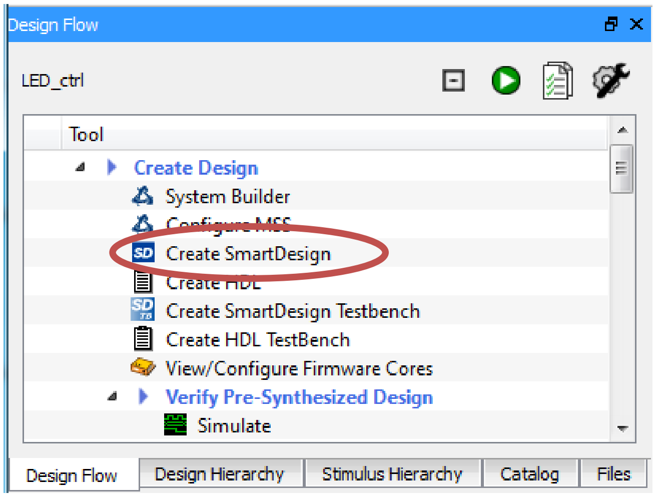
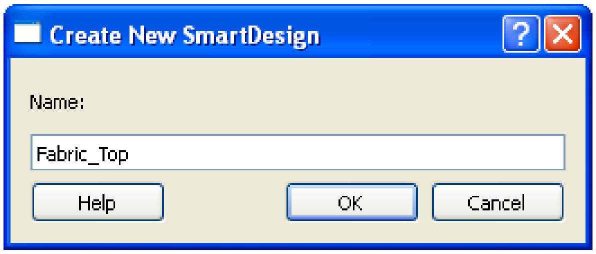


Figure 11 - Opening the SmartDesign canvas

8. Enter *Fabric_Top* in the Create New SmartDesign dialog box then click **OK**. For Verilog designs the name is case sensitive.



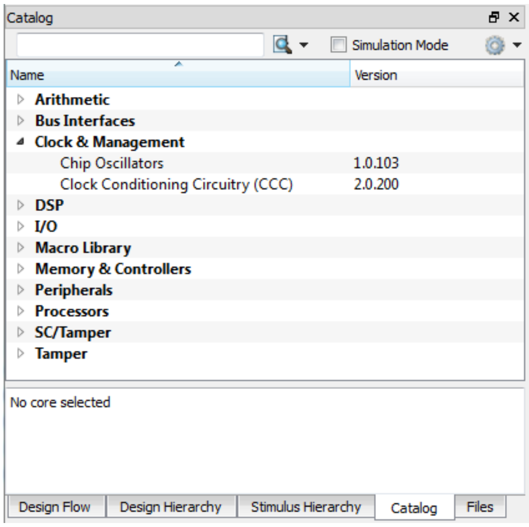
Name: Fabric_Top

Figure 12 - Entering SmartDesign name

9. Drag the LED_ctrl component from the Design Hierarchy tab to the SmartDesign canvas.

This design uses a fabric CCC to generate the 500 kHz internal clock. The CCC reference clock is the 25/50 MHz RC oscillator. In the next steps you will configure the CCC to output a 500 kHz clock and configure the on-chip oscillator.

10. Expand **Clock & Management** in the IP catalog.



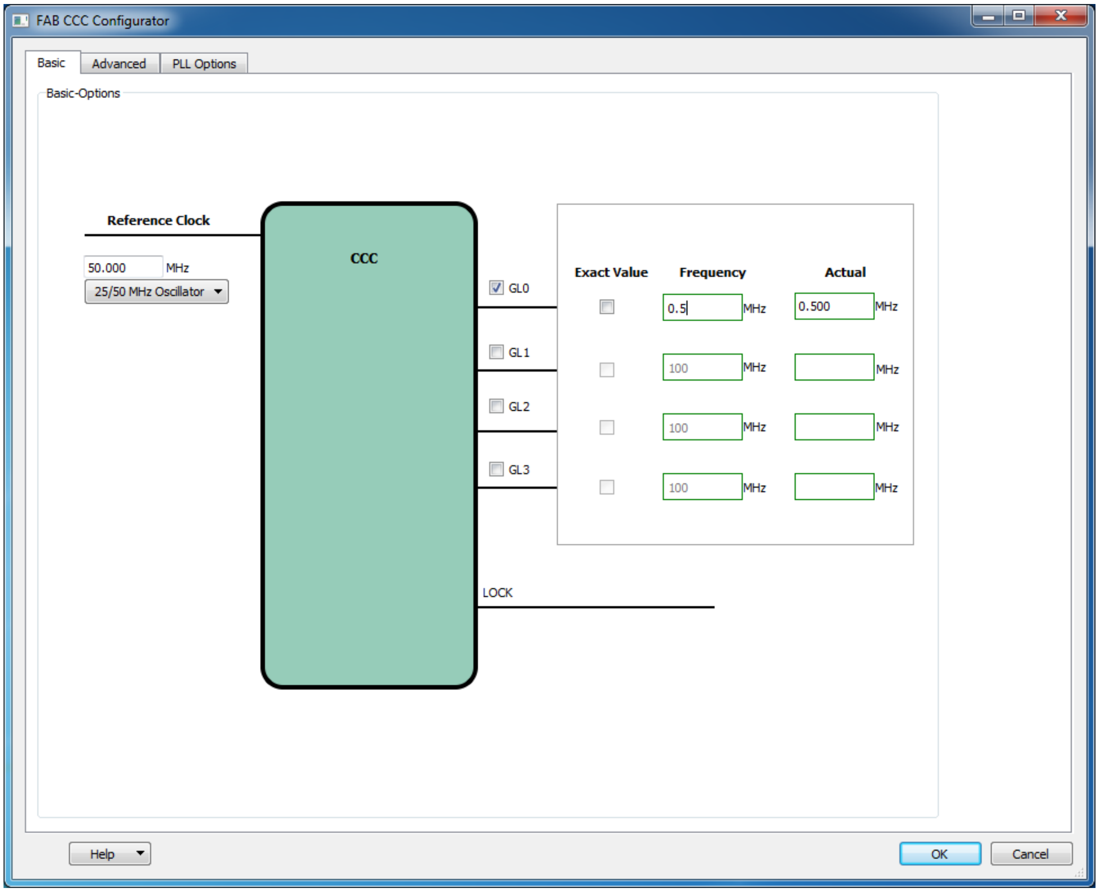
Name	Version
Arithmetic	
Bus Interfaces	
Clock & Management	
Chip Oscillators	1.0.103
Clock Conditioning Circuitry (CCC)	2.0.200
DSP	
I/O	
Macro Library	
Memory & Controllers	
Peripherals	
Processors	
SC/Tamper	
Tamper	

No core selected

Figure 13 - Clock & Management category of the Libero SoC IP Catalog

11. Drag an instance of the Clock Conditioning Circuitry (CCC) v2.0.200 component into the SmartDesign canvas.
12. Double-click the FCCC_0 component in the SmartDesign canvas to open the FAB CCC Configurator.
14. Select the Basic tab in the FAB CCC configurator. Enter the following:

- Reference clock: Select Oscillators > 25/50 MHz Oscillator from the pull-down menu
- GL0: checked; Frequency = 0.5 MHz



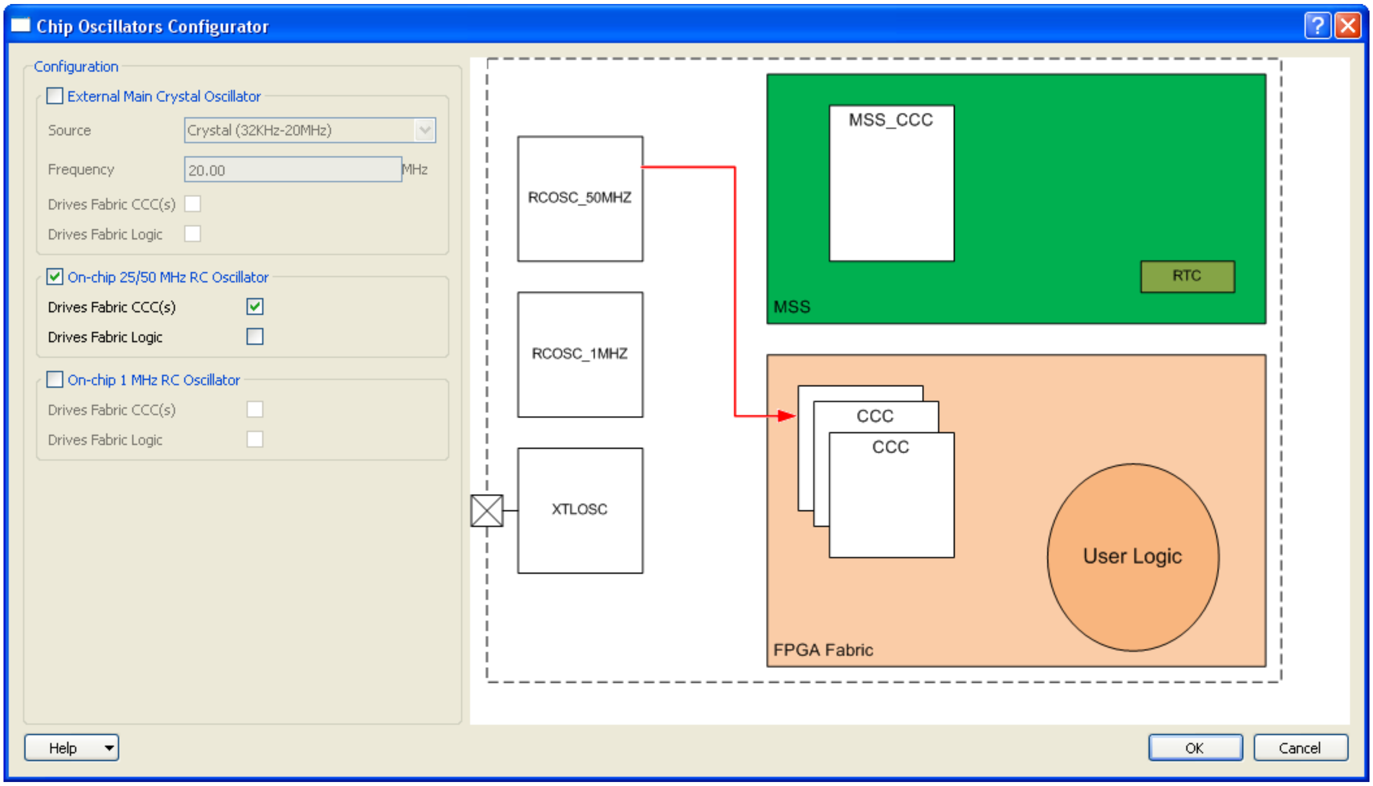
Reference Clock: 50.000 MHz, 25/50 MHz Oscillator

	Exact Value	Frequency	Actual
☑ GL0	□	0.5 MHz	0.500 MHz
□ GL1	□	100 MHz	MHz
□ GL2	□	100 MHz	MHz
□ GL3	□	100 MHz	MHz

Figure 14 - Configuring the fabric CCC

15. Click **OK** to close the FAB CCC Configurator.

16. Drag an instance of the Chip Oscillators v1.0.103 component from the IP catalog into the SmartDesign canvas.
17. Double click the OSC_0 component in the SmartDesign canvas to open the Chip Oscillators configurator.
18. Configure the 25/50 MHz RC oscillator to drive the fabric CCC as follows:
    - Enable the On-chip 25/50 MHz RC Oscillator
    - Select Drives Fabric CCC(s)



Chip Oscillators Configurator

Configuration

☐ External Main Crystal Oscillator

Source: Crystal (32KHz-20MHz)

Frequency: 20.00 MHz

Drives Fabric CCC(s) ☐

Drives Fabric Logic ☐

☑ On-chip 25/50 MHz RC Oscillator

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☐

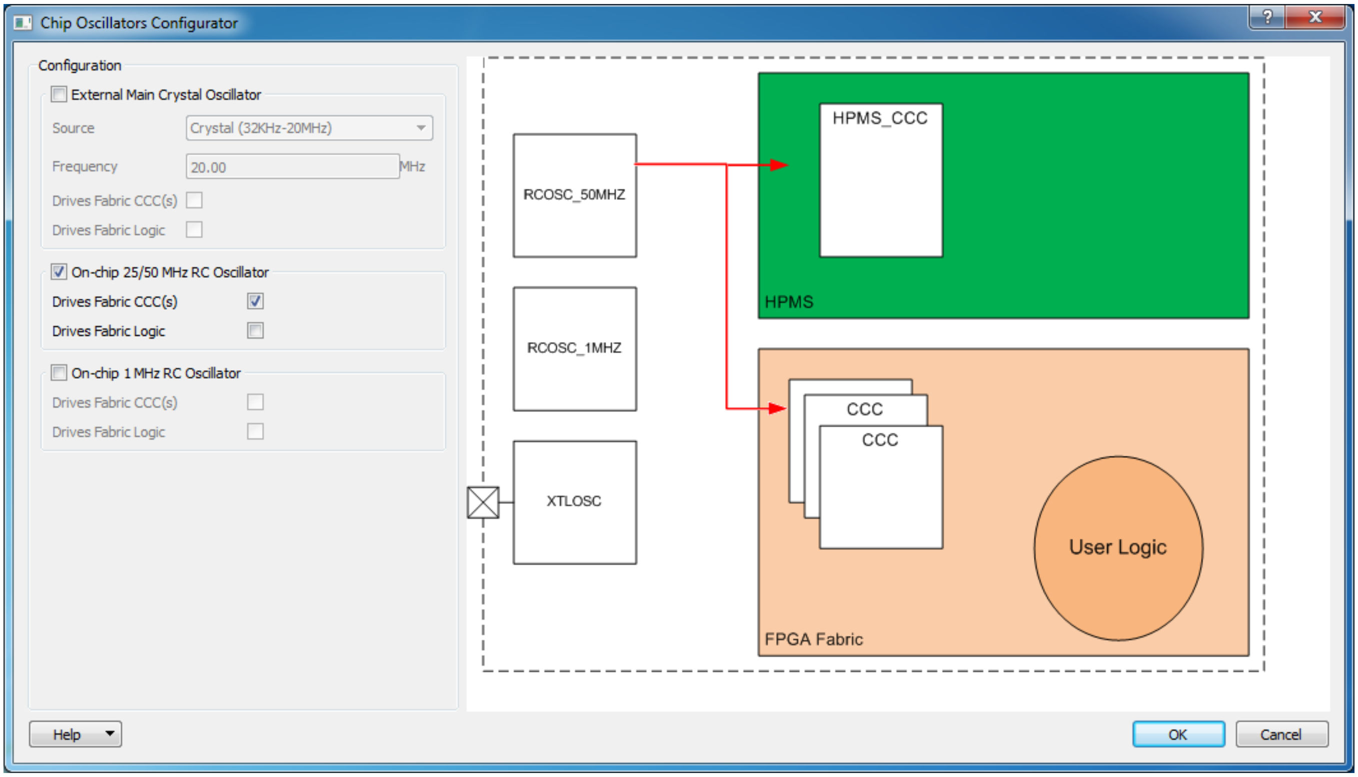
☐ On-chip 1 MHz RC Oscillator

Drives Fabric CCC(s) ☐

Drives Fabric Logic ☐

Figure 15 - Configuring the Chip Oscillators (SmartFusion 2 result)

The Chip Oscillators Configurator will appear as shown below if the target board is the IGLOO2 Evaluation kit.



Chip Oscillators Configurator

Configuration

☐ External Main Crystal Oscillator

Source: Crystal (32KHz-20MHz)

Frequency: 20.00 MHz

Drives Fabric CCC(s) ☐

Drives Fabric Logic ☐

☑ On-chip 25/50 MHz RC Oscillator

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☐

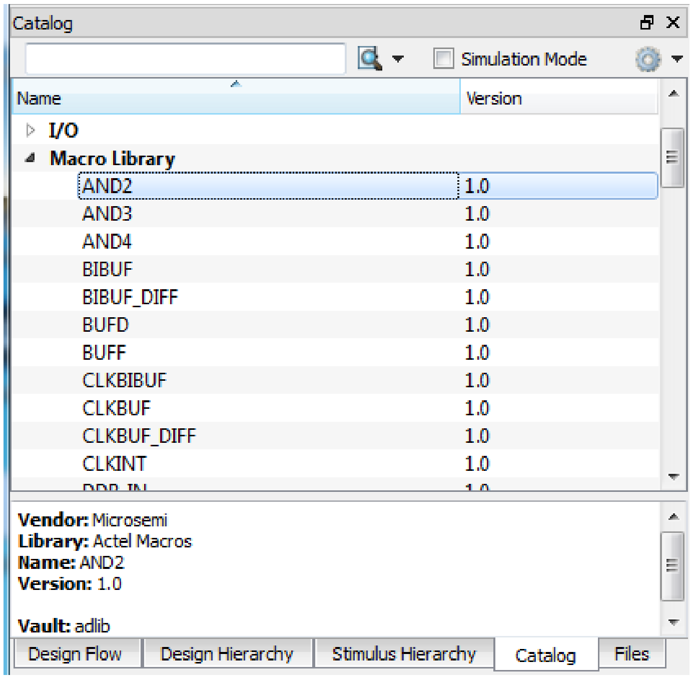
☐ On-chip 1 MHz RC Oscillator

Drives Fabric CCC(s) ☐

Drives Fabric Logic ☐

Figure 16 - Configuring the Chip Oscillators (IGLOO2 result)

19. Click **OK** to close the Chip Oscillators configurator.
20. Expand **Macro Library** in the Libero SoC IP catalog.



Catalog

☐ Simulation Mode

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

Vendor: Microsemi
Library: Actel Macros
Name: AND2
Version: 1.0

Vault: adlib

Figure 17 – Macro Library category of the Libero SoC IP Catalog

21. Drag an instance of AND2 and SYSRESET into the SmartDesign Canvas.

Tip: to make it easier to find the macros listed above, type a few letters of the macro name followed by * in the IP Catalog search field. When finished, change the field to * to display the entire catalog.

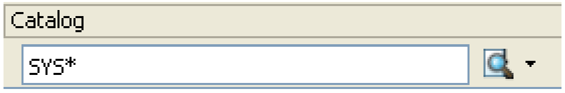


Figure 18 - IP Catalog Search field

22. After adding the components the SmartDesign will resemble the figure below. If needed, drag the components to improve the appearance of the canvas.

    Tip: expand the canvas area by selecting **View > Maximize Work Area** or click the icon on the tool bar ( ).

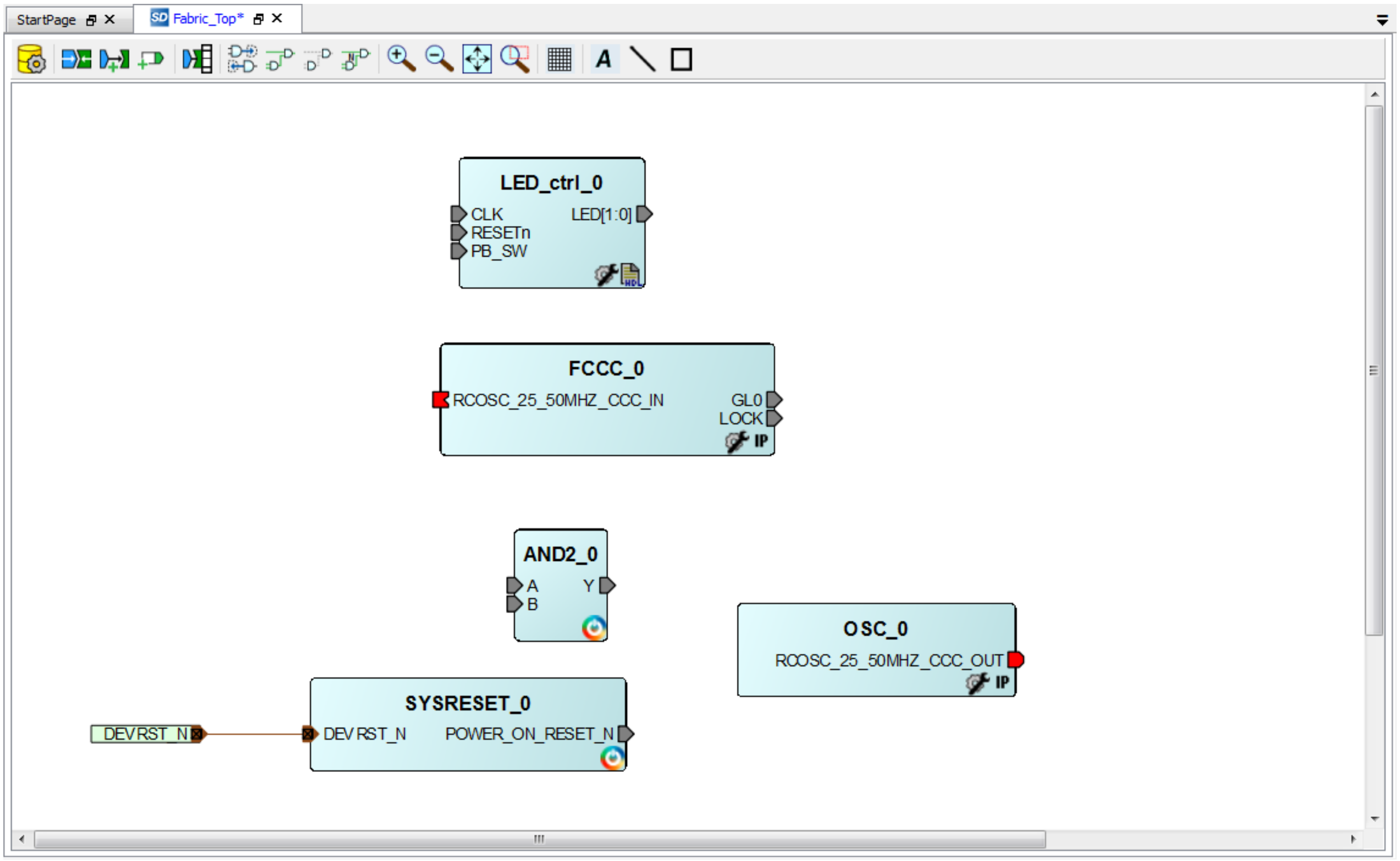


Figure 19 - SmartDesign canvas after adding components

Making connections in the canvas

Next connect the components in the SmartDesign canvas to complete the design. SmartDesign in Libero SoC has a connection mode that supports click, drag and release to make connections.

23. Select **SmartDesign > Connection Mode** from the Libero SoC menu or click the Connection Mode icon ( ).
24. Connect the RCOSC_25_50MHZ_CCC_OUT port of OSC_0 component to the RCOSC_25_50MHZ_CCC_IN port of the FCCC_0 component as follows:
    - Click the RCOSC_25_50MHZ_CCC_OUT port of the OSC_0 component and hold the left mouse button.
    - While holding the left mouse button, drag to the RCOSC_25_50MHZ_CCC_IN port of FCCC_0 component.
    - Release the mouse button to make the connection.
25. Repeat the previous step to make the connections shown it the table below.

From	To
FCCC_0:GL0	LED_ctrl_0:CLK
FCCC_0:LOCK	AND2_0:A
SYSRESET_0:POWER_ON_RESET_N	AND2_0:B
AND2_0:Y	LED_ctrl_0:RESETn

Table 3 – SmartDesign canvas connections

26. Disable the SmartDesign connection mode by selecting **SmartDesign > Connection Mode** from the Libero SoC menu.
27. Promote the following ports to the top level by clicking the port, then right clicking and selecting **Promote to Top Level**:
    - LED_ctrl:PB_SW
    - LED_ctrl:LED[1:0]

28. After making the connections listed above the SmartDesign canvas will appear a shown in the figure below. You can drag the components or use the SmartDesign Auto Arrange feature to improve the appearance of the canvas.

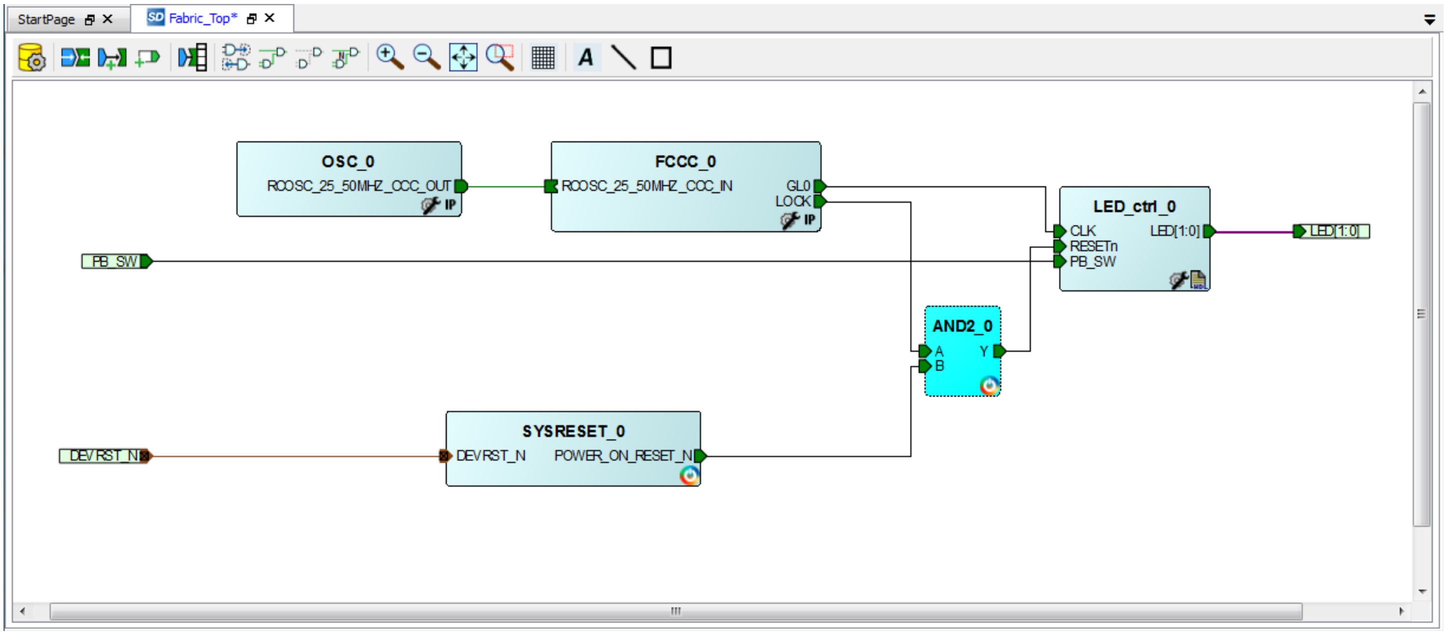


Figure 20 - SmartDesign canvas after making connections

29. Save the design (**File > Save Fabric_Top**).
30. Generate the design by clicking **SmartDesign > Generate Component** or by clicking the Generate Component icon on the SmartDesign toolbar ( ).
31. Restore the work area (**View > Restore Work Area**) if you expanded the work area earlier.
32. Confirm that the message “'Fabric_Top' was successfully generated.” appears in the Libero Log window.
33. Close the design (**File > Close Fabric_Top**).

Step 2 - Simulating the design

The next step is to simulate the design. A testbench, ModelSim macro file and a wave format file have been provided in the source files. The LED_ctrl module/entity contains a 19 bit counter. With the slow clock rate (500 kHz) simulation would take a long time. In order to accelerate the simulation of the design some of the counter bits are forced high in the ModelSim macro file.

1. Confirm that Fabric_Top appears in bold font in the Libero Design Hierarchy window. If it does not, select Fabric_Top, right-click and select **Set As Root**.

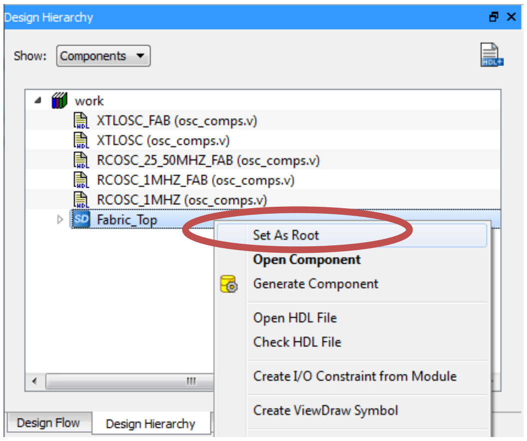


Figure 21 - Setting Fabric_Top as the root level

2. Expand Verify Pre-Synthesized Design in the Design Flow window. Right-click **Simulate** and select **Import Files..** .

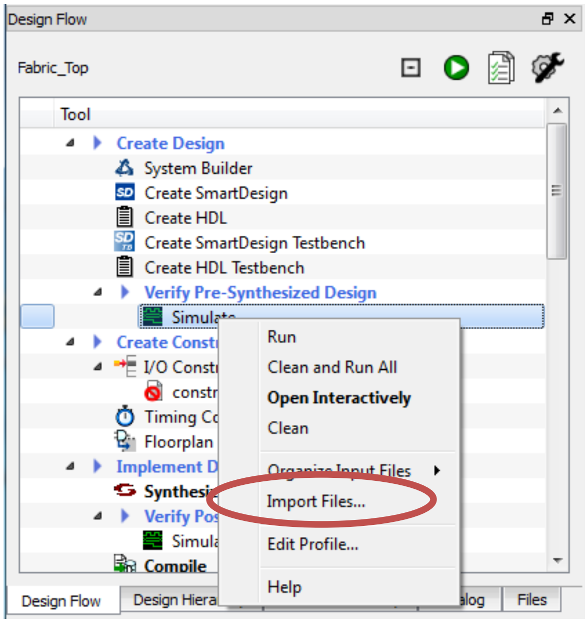
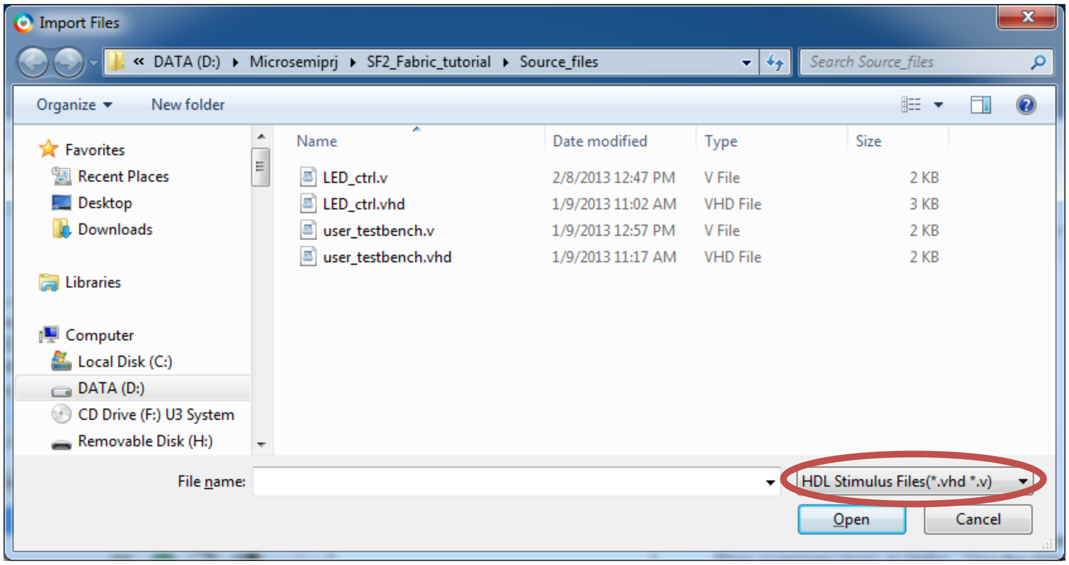


Figure 22 - Importing the testbench

3. Enter the following in the Import Files dialog box then click **Open**:
   - Look in: <C: or D:>\Microsemiprj\SF2_Fabric_tutorial\Source_files
   - Files of type: HDL Stimulus Files (*.vhd *.v)
   - File name: user_testbench.vhd (for VHDL projects) or user_testbench.v (for Verilog projects)



Import Files

« DATA (D:) ▸ Microsemiprj ▸ SF2_Fabric_tutorial ▸ Source_files

Search Source_files

Organize ▾ New folder

Favorites, Recent Places, Desktop, Downloads, Libraries, Computer, Local Disk (C:), DATA (D:), CD Drive (F:) U3 System, Removable Disk (H:)

Name	Date modified	Type	Size
LED_ctrl.v	2/8/2013 12:47 PM	V File	2 KB
LED_ctrl.vhd	1/9/2013 11:02 AM	VHD File	3 KB
user_testbench.v	1/9/2013 12:57 PM	V File	2 KB
user_testbench.vhd	1/9/2013 11:17 AM	VHD File	2 KB

File name:

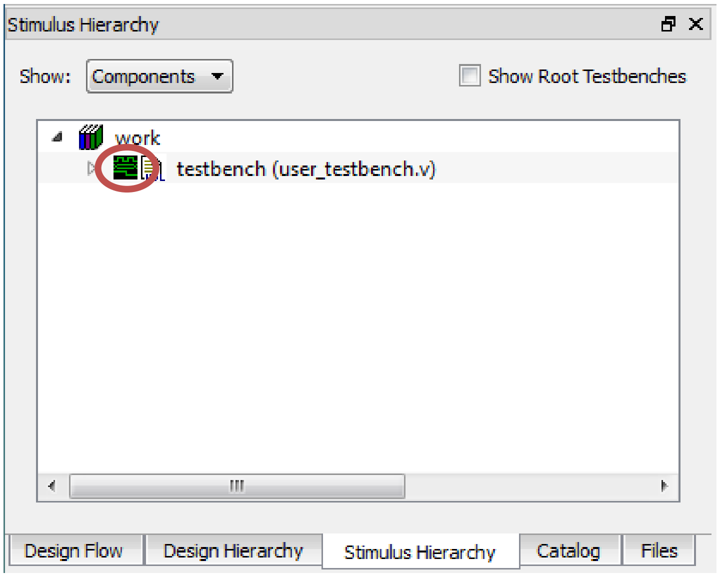
HDL Stimulus Files(*.vhd *.v)

Open Cancel

Figure 23 - Importing the testbench

4. Select the Stimulus Hierarchy tab. The testbench (user_testbench.v(hd)) will be visible. Select the testbench, right-click and select **Set As Active stimulus**. A waveform symbol will indicate the active stimulus (circled below).

   Note: if user_testbench.v(hd) is not visible on the Stimulus Hierarchy tab, the file was not imported as an HDL stimulus file. Re-import the file as described above.



Stimulus Hierarchy

Show: Components ▾ □ Show Root Testbenches

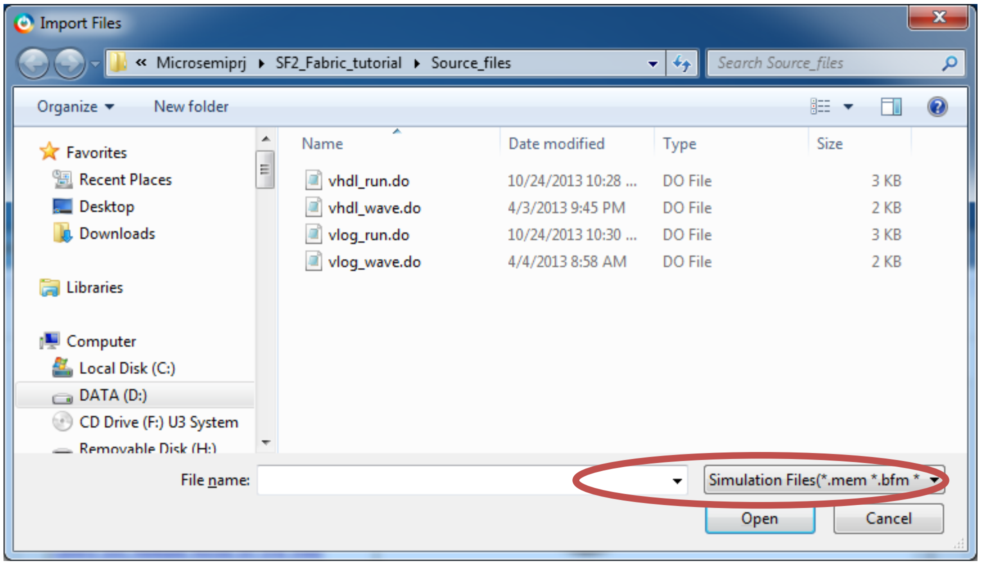
work

testbench (user_testbench.v)

Design Flow | Design Hierarchy | Stimulus Hierarchy | Catalog | Files

Figure 24 - Testbenches in the SmartFusion2_Fabric project (Verilog shown)

5. Import the ModelSim macro file and Wave format file by right-clicking **Simulate** under Verify Pre-Synthesized Design in the Design Flow window and selecting **Import Files..** .
6. Enter the following in the Import Files dialog box then click **Open**:
   - Look in: <C: or D:>\Microsemiprj\SF2_Fabric_tutorial\Source_files
   - Files of type: Simulation Files (*.mem *.bfm *.dat *.txt *.do)
   - File name: Hold the shift key and select vhdl_run.do and vhdl_wave.do (VHDL projects) OR vlog_run.do and vlog_wave.do (Verilog projects)



Name	Date modified	Type	Size
vhdl_run.do	10/24/2013 10:28 ...	DO File	3 KB
vhdl_wave.do	4/3/2013 9:45 PM	DO File	2 KB
vlog_run.do	10/24/2013 10:30 ...	DO File	3 KB
vlog_wave.do	4/4/2013 8:58 AM	DO File	2 KB

Figure 25 - Importing the simulation files

The testbench and simulation files will be visible on the Libero SoC Files tab under Stimulus and Simulation.

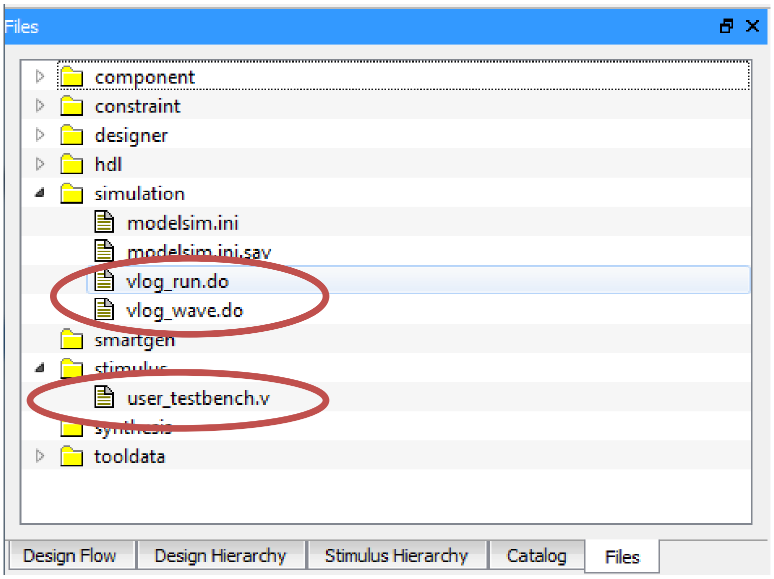


Figure 26 - Testbench and simulation files (Verilog shown)

7. Open the ModelSim macro file (vhdl_run.do or vlog_run.do) in the Libero SoC editor by double-clicking the filename on the Files tab.
8. Locate the variable PROJECT_DIR on line 15 of vhdl_run.do or vlog_run.do and confirm that it matches the location of your Libero SoC fabric tutorial. The location is displayed at the top of the Libero SoC GUI (do not include "SmartFusion2_Fabric.prjx"). Edit the path if necessary.

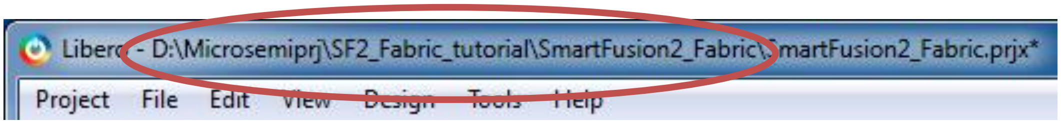
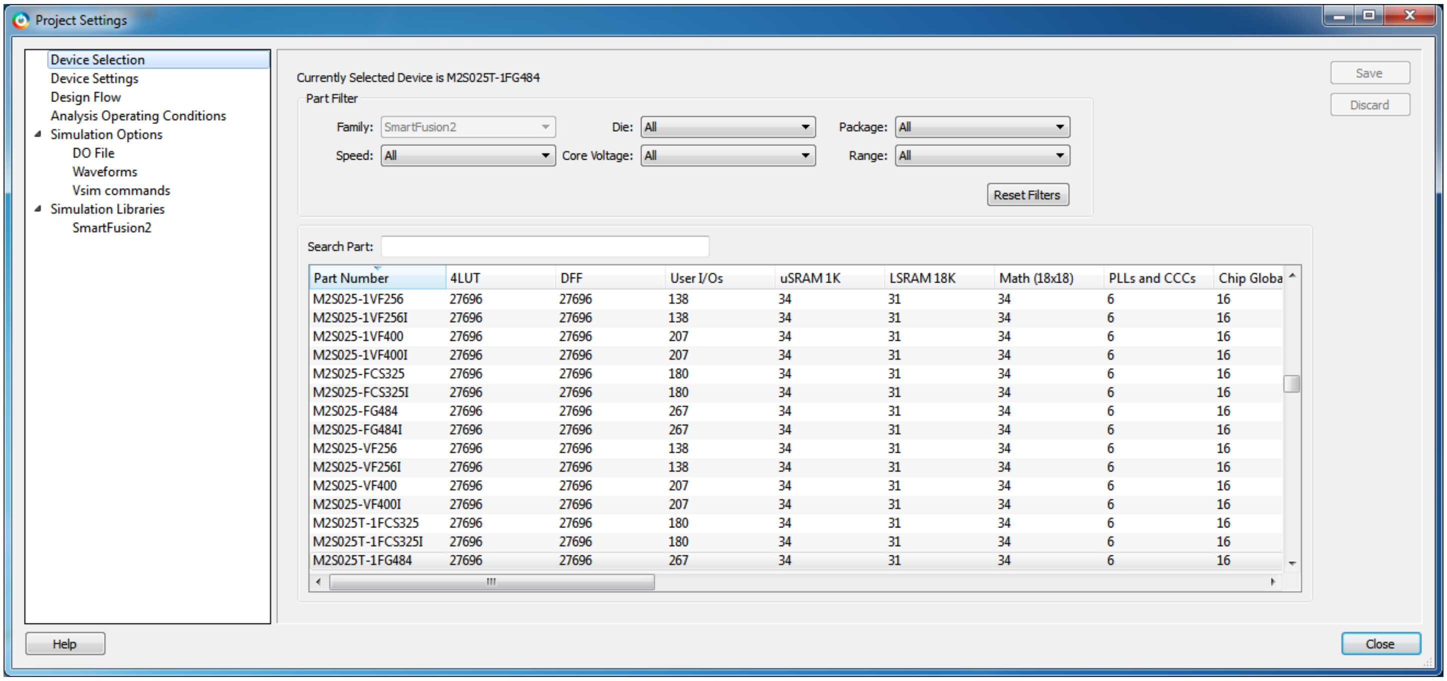

Libero - D:\Microsemiprj\SF2_Fabric_tutorial\SmartFusion2_Fabric\SmartFusion2_Fabric.prjx*

Figure 27 - Location of Libero SoC project

9. Locate the variable INSTALL_DIR on line 16 of vhdl_run.do or vlog_run.do and confirm that it matches the location of your Libero SoC installation. Edit the path if necessary. Ask the lab instructor if you do not know the Libero SoC installation path.

10. If you made changes, save the file (**File > Save vhdl_run.do or File > Save vlog_run.do**).
11. Scroll in the file to become familiar with the commands it contains. The command force –freeze forces some of the counter bits in LED_ctrl high to speed up the simulation.
12. Close the editor (**File > Close vhdl_run.do** or **File > Close vlog_run.do**).
13. Open the Libero SoC project settings (**Project > Project Settings**).



Currently Selected Device is M2S025T-1FG484

Part Filter

Family: SmartFusion2
Die: All
Package: All
Speed: All
Core Voltage: All
Range: All

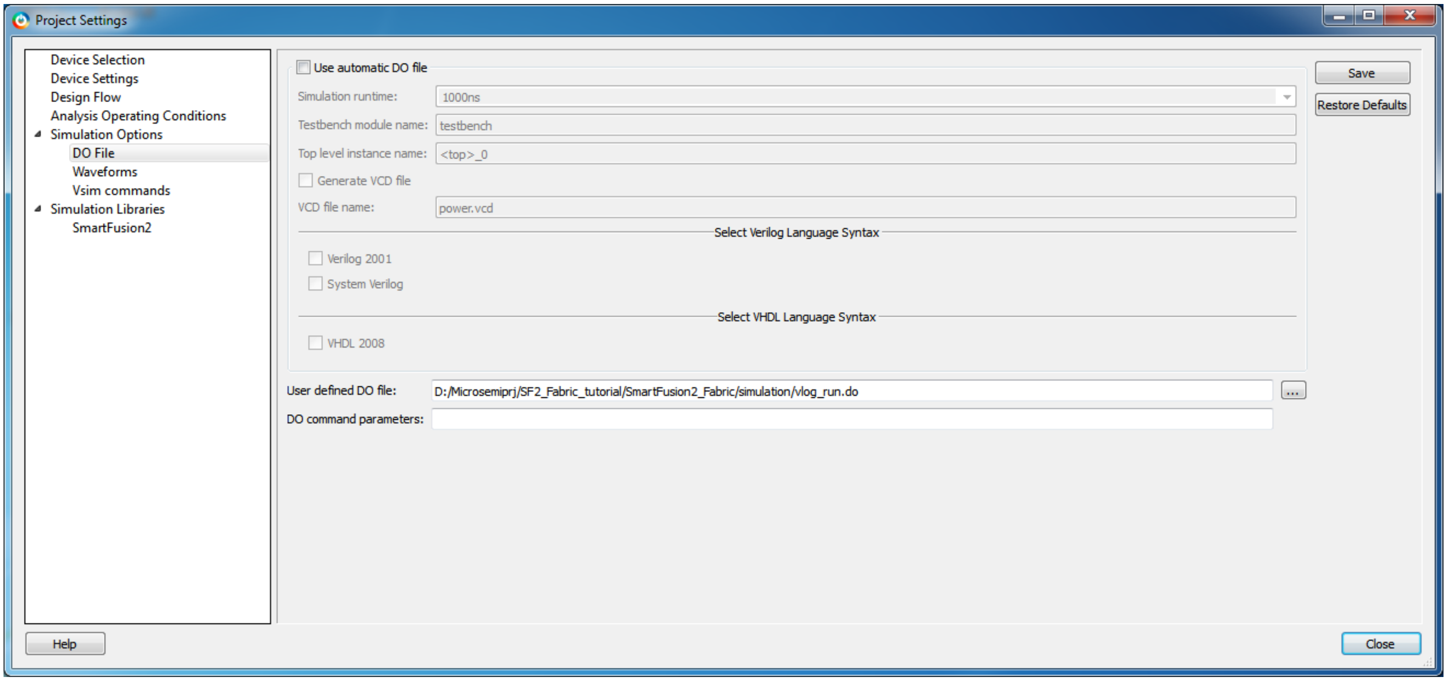
Search Part:

Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and CCCs	Chip Globa
M2S025-1VF256	27696	27696	138	34	31	34	6	16
M2S025-1VF256I	27696	27696	138	34	31	34	6	16
M2S025-1VF400	27696	27696	207	34	31	34	6	16
M2S025-1VF400I	27696	27696	207	34	31	34	6	16
M2S025-FCS325	27696	27696	180	34	31	34	6	16
M2S025-FCS325I	27696	27696	180	34	31	34	6	16
M2S025-FG484	27696	27696	267	34	31	34	6	16
M2S025-FG484I	27696	27696	267	34	31	34	6	16
M2S025-VF256	27696	27696	138	34	31	34	6	16
M2S025-VF256I	27696	27696	138	34	31	34	6	16
M2S025-VF400	27696	27696	207	34	31	34	6	16
M2S025-VF400I	27696	27696	207	34	31	34	6	16
M2S025T-1FCS325	27696	27696	180	34	31	34	6	16
M2S025T-1FCS325I	27696	27696	180	34	31	34	6	16
M2S025T-1FG484	27696	27696	267	34	31	34	6	16

Figure 28 - Libero SoC Project Settings dialog box (settings for M2S-EVAL-KIT board shown)

14. Select Do File under **Simulation Options** in the Project Settings Dialog box.
15. Un-check “Use automatic DO file”.
16. Click the browse button (...)next to “User defined DO file” and enter the following then click **Open**:
    - Look in: <C: or D:>\Microsemiprj\SF2_Fabric_tutorial\SmartFusion2_Fabric\simulation
    - Files of type: *.do
    - File name: vhdl_run.do (VHDL projects) or vlog_run.do (Verilog projects)

    The ModelSim macro file (vhdl_run.do or vlog_run.do) calls the Wave format file (vhdl_wave.do or vlog_wave.do), so there are no settings required for the Wave Format file.



☐ Use automatic DO file

Simulation runtime: 1000ns

Testbench module name: testbench

Top level instance name: <top>_0

☐ Generate VCD file

VCD file name: power.vcd

Select Verilog Language Syntax

☐ Verilog 2001

☐ System Verilog

Select VHDL Language Syntax

☐ VHDL 2008

User defined DO file: D:/Microsemiprj/SF2_Fabric_tutorial/SmartFusion2_Fabric/simulation/vlog_run.do

DO command parameters:

Figure 29 - Simulation options

17. Click **Save** then **Close** to close the Project settings dialog box.
18. Expand Verify Pre-Synthesized Design in the Design Flow window. Right-click **Simulate** and select **Open Interactively** to launch ModelSim in GUI mode.

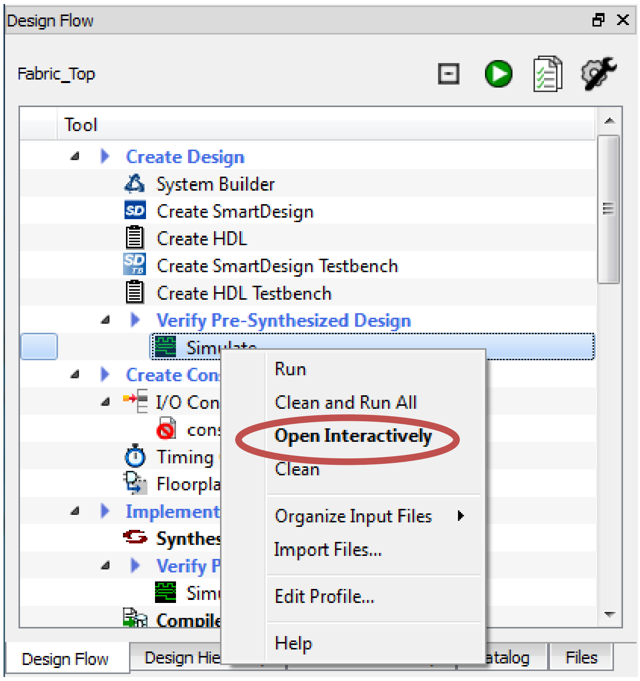
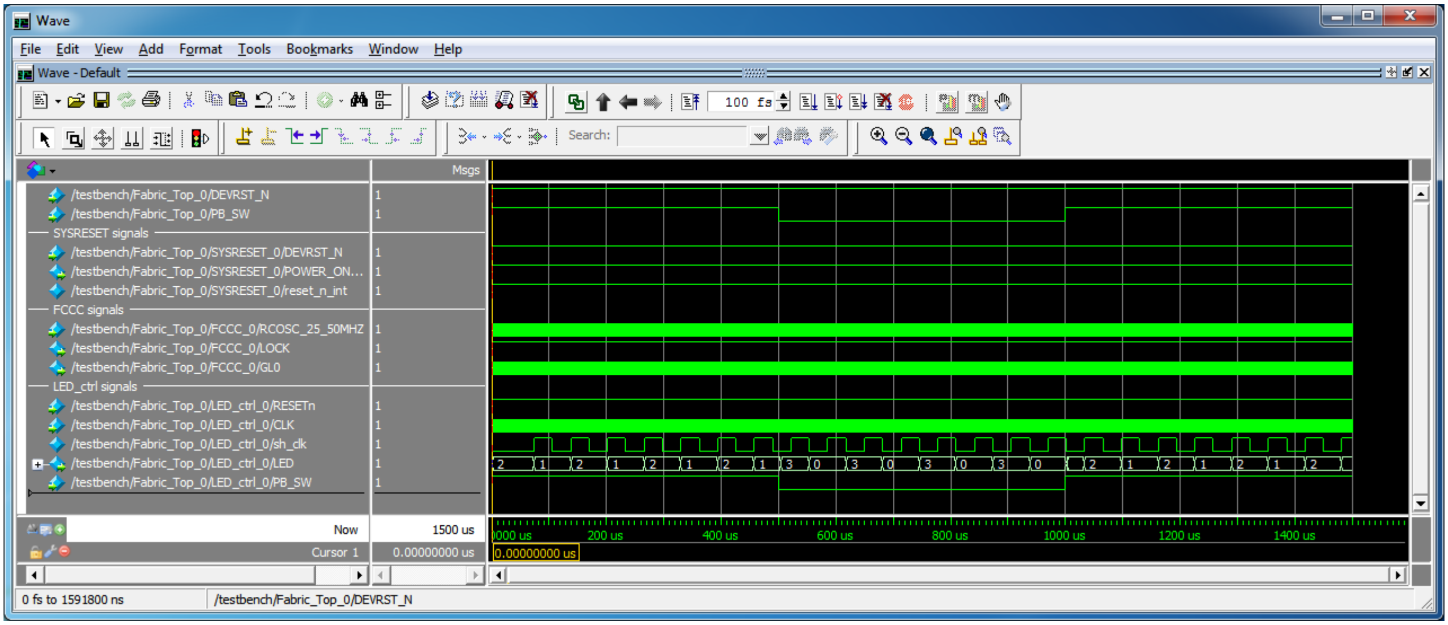


Figure 30 - Launching pre-synthesis simulation

19. The simulation will run for 1.5 ms. When finished, the Wave window should appear as shown below.



```
/testbench/Fabric_Top_0/DEVRST_N                 1
/testbench/Fabric_Top_0/PB_SW                    1
SYSRESET signals
/testbench/Fabric_Top_0/SYSRESET_0/DEVRST_N      1
/testbench/Fabric_Top_0/SYSRESET_0/POWER_ON...   1
/testbench/Fabric_Top_0/SYSRESET_0/reset_n_int   1
FCCC signals
/testbench/Fabric_Top_0/FCCC_0/RCOSC_25_50MHZ    1
/testbench/Fabric_Top_0/FCCC_0/LOCK              1
/testbench/Fabric_Top_0/FCCC_0/GL0               1
LED_ctrl signals
/testbench/Fabric_Top_0/LED_ctrl_0/RESETn        1
/testbench/Fabric_Top_0/LED_ctrl_0/CLK           1
/testbench/Fabric_Top_0/LED_ctrl_0/sh_clk        1
/testbench/Fabric_Top_0/LED_ctrl_0/LED           1
/testbench/Fabric_Top_0/LED_ctrl_0/PB_SW         1

Now         1500 us
Cursor 1    0.00000000 us    0.00000000 us

0 fs to 1591800 ns    /testbench/Fabric_Top_0/DEVRST_N
```

Figure 31 - ModelSim Wave window (Verilog results shown)

20. Scroll in the wave window. The LED port represents the LED driver (0 = LED off; 1 = LED on). For the SmartFusion2 Evaluation kit, 1 = LED off; 0 = LED on. The PB_SW port represents the switch input (0 = switch depressed; 1 = switch released.
21. Confirm that the LED output (LED) matches the description in the table on page 4. Contact the lab instructor if your results do not match the figure above.
22. Close the ModelSim simulator (**File > Quit**). Click **Yes** when asked if you want to quit.

Step 3 – Editing the I/O Physical Constraint file

There are multiple ways to make I/O Assignments. In this lab we will use the I/O Physical Design Constraint (PDC) file that was imported when the project was created. The I/O pdc file must be edited to match the target board.

1. Double-click the PDC file name on the Files tab to open the file in the Libero SoC editor. Scroll in the file to become familiar with the syntax. The constraint set_iobank sets the input voltage of the bank to determine the allowable I/O standards; the constraint set_io sets the pin number and I/O specific attributes. The # symbol is a comment.

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

   For more information regarding pin assignments for the Starter Kit board, see the Starter Kit schematic (som-bsb-ext-1a-schem.pdf) which is available from the SmartFusion2 Starter Kit web page.

2. All the assignments in the PDC file are commented. Uncomment the lines per the table below to match the target board. To un-comment a line, highlight one or more lines by dragging in the file while holding the left mouse button down, then right-click and select **Uncomment Selection**.

Board	Lines to un-comment in PDC file
SF2-STARTER-KIT-ES-2	30 - 39
SF2-STARTER-KIT	45 - 53
SF2-484-STARTER-KIT	
M2S-EVAL-KIT	59 - 63
M2S090S-EVAL-KIT	
SF2-DEV-KIT-PP	68 - 78
SF2-DEV-KIT-PP-1	
SF2-DEV-KIT	
M2S150-ADV-DEV-KIT-ES	82 - 84
M2GL-EVAL-KIT	45 - 53

Table 4 – PDC file edits

3. Save the PDC file after making modifying (**File > Save Fabric_Top.pdc**).
4. Close the PDC file (**File > Close Fabric_Top.pdc**.

Step 4 – Synthesis and Layout

In this step you will use the push-button flow to synthesize the design with Synplify Pro, run layout and generate the programming file.

1. Expand **Create Constraints > I/O Constraints** in the Libero SoC Design Flow window. Right-click Fabric_Top.pdc under Constraints then right-click and select **Use for Compile**. A green check mark will appear next to the constraint file indicating that the file will be used.

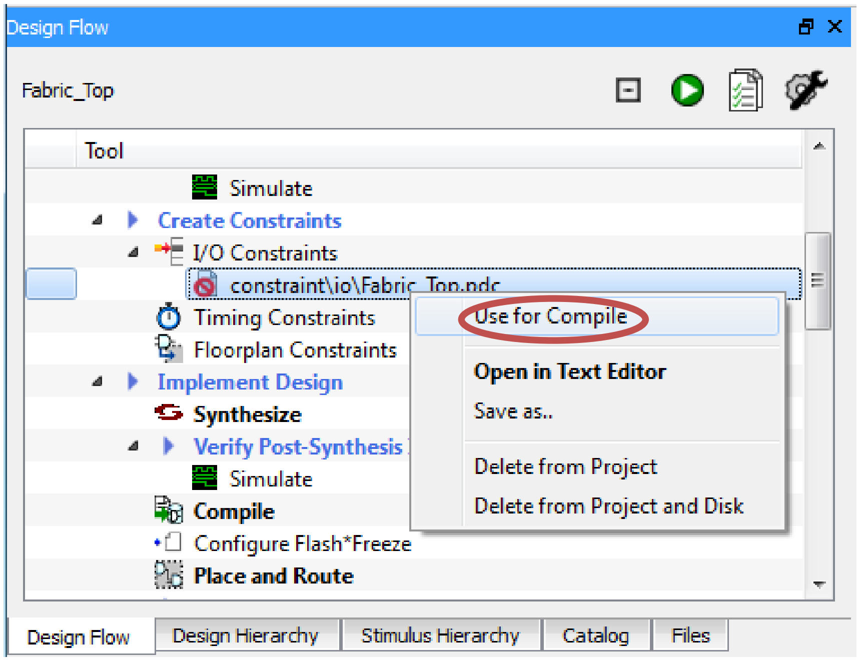


Figure 32 - Selecting the I/O PDC constraint file in the Design Flow window

2. Right-click Compile on the Design Flow tab and select **Configure Options**.

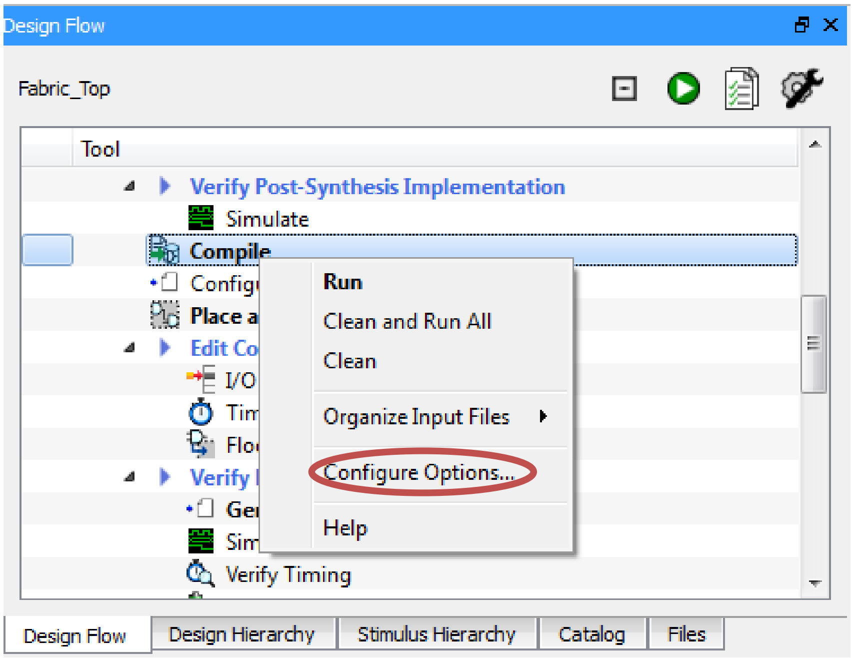
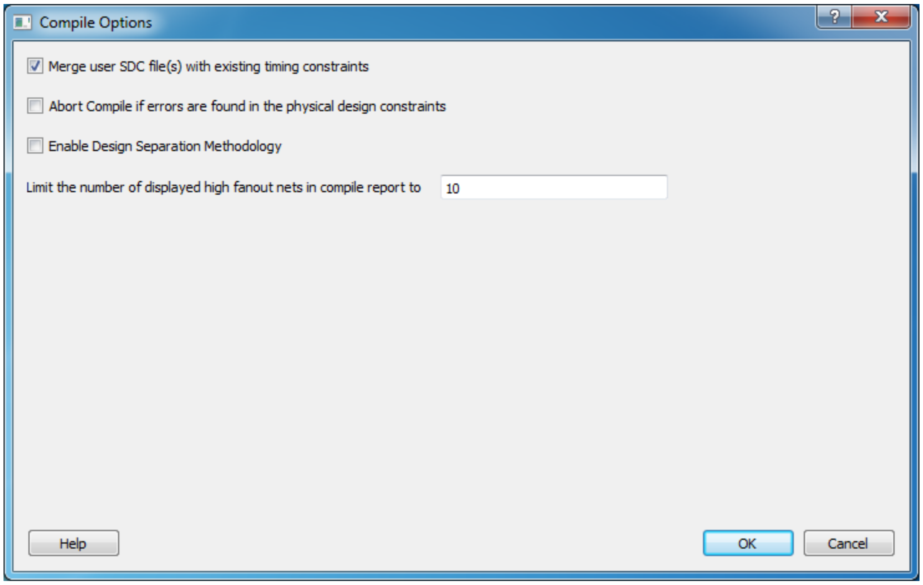


Figure 33 - Opening Compile Options

3. The Compile Options dialog box will open. Un-check "Abort Compile if errors are found in the physical design constraints" and then click **OK**.



☑ Merge user SDC file(s) with existing timing constraints

□ Abort Compile if errors are found in the physical design constraints

□ Enable Design Separation Methodology

Limit the number of displayed high fanout nets in compile report to 10

Figure 34 - Configuring the Compile Options

4. Click the Generate Programming Data icon in the Design Flow window (circled in the figure below) or select **Design > Generate Bitstream** to synthesize the design, run layout using the I/O constraints that were created and generate the programming file.

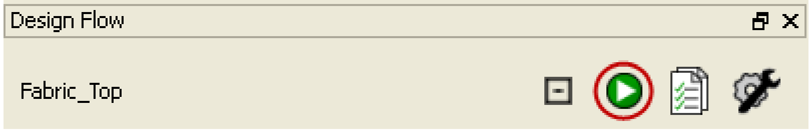
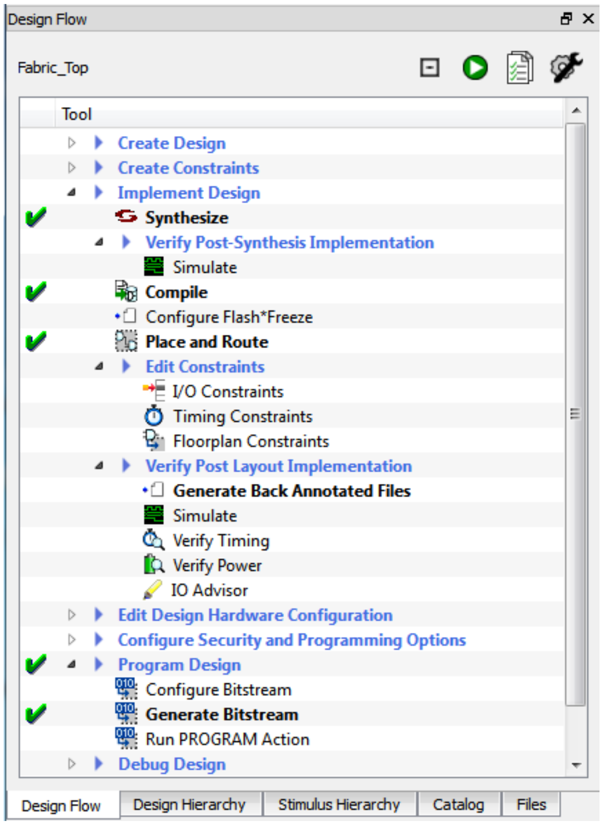


Figure 35 - Generate Programming Data icon

The design implementation tools will run in batch mode. Successful completion of a design step will be indicated by a green check mark next to the Implement Design in the Design Flow window.



Design Flow

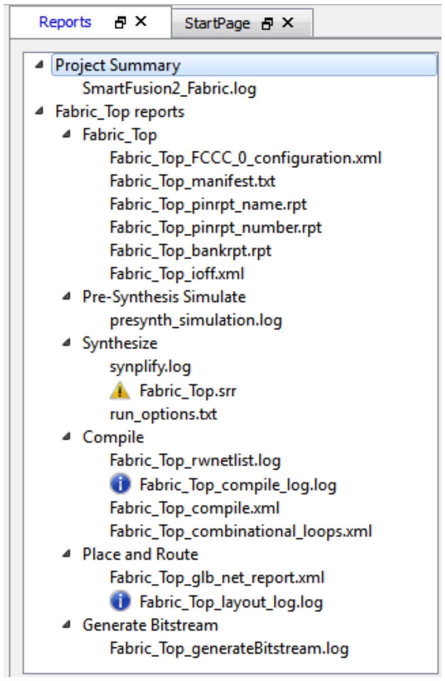
Fabric_Top

Tool

- Create Design
- Create Constraints
- Implement Design
  - ✓ Synthesize
  - Verify Post-Synthesis Implementation
    - Simulate
  - ✓ Compile
  - Configure Flash*Freeze
  - ✓ Place and Route
  - Edit Constraints
    - I/O Constraints
    - Timing Constraints
    - Floorplan Constraints
  - Verify Post Layout Implementation
    - Generate Back Annotated Files
    - Simulate
    - Verify Timing
    - Verify Power
    - IO Advisor
- Edit Design Hardware Configuration
- Configure Security and Programming Options
- ✓ Program Design
  - Configure Bitstream
  - ✓ Generate Bitstream
  - Run PROGRAM Action
- Debug Design

Figure 36 – Successful completion of design implementation

5. The Reports tab will display reports for the tools used to implement the design.



- Project Summary
  - SmartFusion2_Fabric.log
- Fabric_Top reports
  - Fabric_Top
    - Fabric_Top_FCCC_0_configuration.xml
    - Fabric_Top_manifest.txt
    - Fabric_Top_pinrpt_name.rpt
    - Fabric_Top_pinrpt_number.rpt
    - Fabric_Top_bankrpt.rpt
    - Fabric_Top_ioff.xml
  - Pre-Synthesis Simulate
    - presynth_simulation.log
  - Synthesize
    - synplify.log
    - ⚠ Fabric_Top.srr
    - run_options.txt
  - Compile
    - Fabric_Top_rwnetlist.log
    - ⓘ Fabric_Top_compile_log.log
    - Fabric_Top_compile.xml
    - Fabric_Top_combinational_loops.xml
  - Place and Route
    - Fabric_Top_glb_net_report.xml
    - ⓘ Fabric_Top_layout_log.log
  - Generate Bitstream
    - Fabric_Top_generateBitstream.log

Figure 37 - Reports tab after implementing the design

6. Select the Compile report (Fabric_Top_compile.xml) under Compile on the Reports tab to view the resource usage. Record the number of sequential and combinatorial cells used in the design below.

   LUTs (4LUT) ______________________________

   Flip-flops (DFF) ______________________________

7. Generate a timing report by right-clicking Verify Timing under Verify Post Layout Implementation on the Design Flow tab and selecting **Run**.

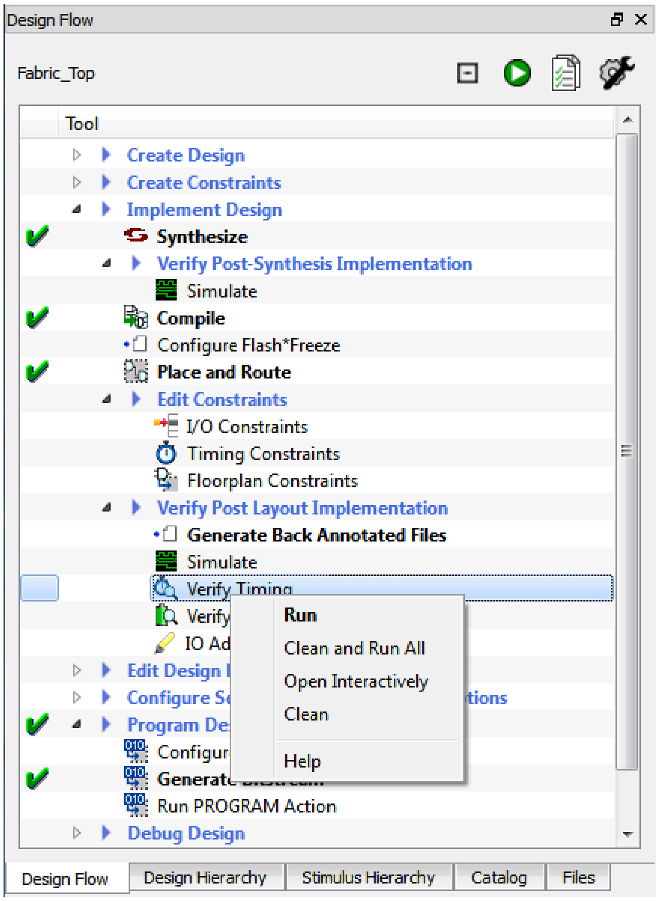
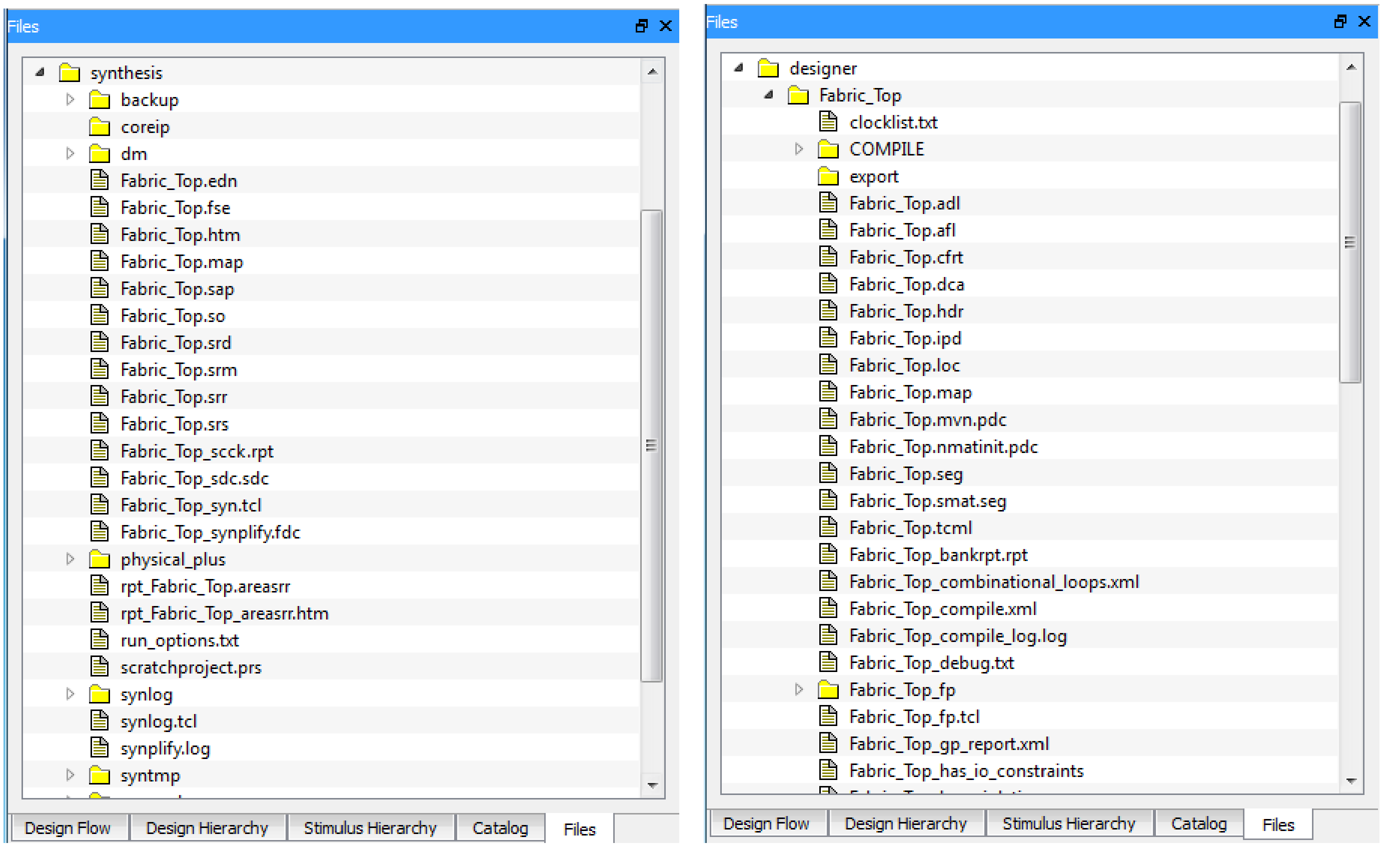


Figure 38 - Generating the post-layout timing report

8. Select the timing report (Fabric_Top_maxdelay_timing_report.xml) under Verify Timing on the Reports tab. Scroll to the timing summary in the Report window. Record the maximum frequency for the clock FCCC_0/CCC_INST/INST_CCC_IP:GL0 below:

   FCCC_0/CCC_INST/INST_CCC_IP:GL0: ___________________________

9. Timing violations can quickly be identified by looking at the timing violations reports (Fabric_Top_maxdleay_timing_violations_report.xml and Fabric_Top_mindleay_timing_violations_report.xml).
10. The output files from Synplify Pro and Designer will be visible on the Libero SoC Files tab.



```
synthesis
    backup
    coreip
    dm
    Fabric_Top.edn
    Fabric_Top.fse
    Fabric_Top.htm
    Fabric_Top.map
    Fabric_Top.sap
    Fabric_Top.so
    Fabric_Top.srd
    Fabric_Top.srm
    Fabric_Top.srr
    Fabric_Top.srs
    Fabric_Top_scck.rpt
    Fabric_Top_sdc.sdc
    Fabric_Top_syn.tcl
    Fabric_Top_synplify.fdc
    physical_plus
    rpt_Fabric_Top.areasrr
    rpt_Fabric_Top_areasrr.htm
    run_options.txt
    scratchproject.prs
    synlog
    synlog.tcl
    synplify.log
    syntmp
```

```
designer
    Fabric_Top
        clocklist.txt
        COMPILE
        export
        Fabric_Top.adl
        Fabric_Top.afl
        Fabric_Top.cfrt
        Fabric_Top.dca
        Fabric_Top.hdr
        Fabric_Top.ipd
        Fabric_Top.loc
        Fabric_Top.map
        Fabric_Top.mvn.pdc
        Fabric_Top.nmatinit.pdc
        Fabric_Top.seg
        Fabric_Top.smat.seg
        Fabric_Top.tcml
        Fabric_Top_bankrpt.rpt
        Fabric_Top_combinational_loops.xml
        Fabric_Top_compile.xml
        Fabric_Top_compile_log.log
        Fabric_Top_debug.txt
        Fabric_Top_fp
        Fabric_Top_fp.tcl
        Fabric_Top_gp_report.xml
        Fabric_Top_has_io_constraints
```

Figure 39 - Synthesis and Designer files on Libero SoC Files tab

Step 5 – Programming

In this step you will run FlashPro in batch mode to program the SmartFusion2 device on the SmartFusion2 target board. Jumper settings for the supported target boards are shown in the tables in Appendix 1 – 4. Prior to programming (and powering up) the SmartFusion2 target board, confirm that the jumpers are positioned as show in the tables and follow the instructions for powering the board.

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

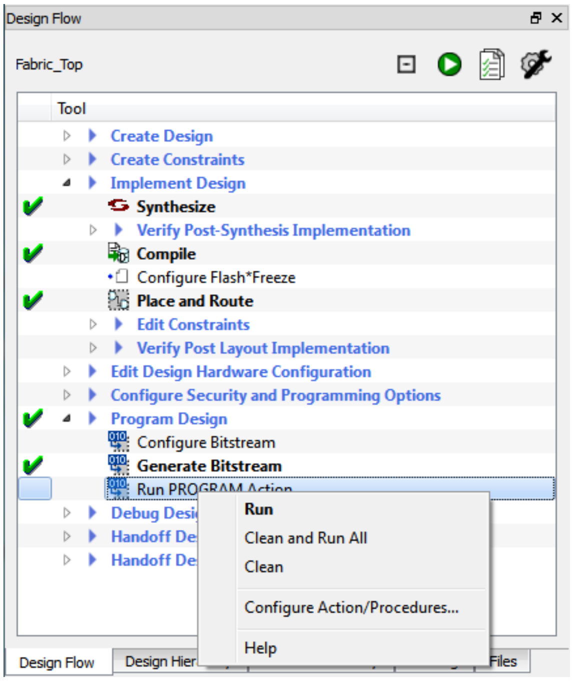


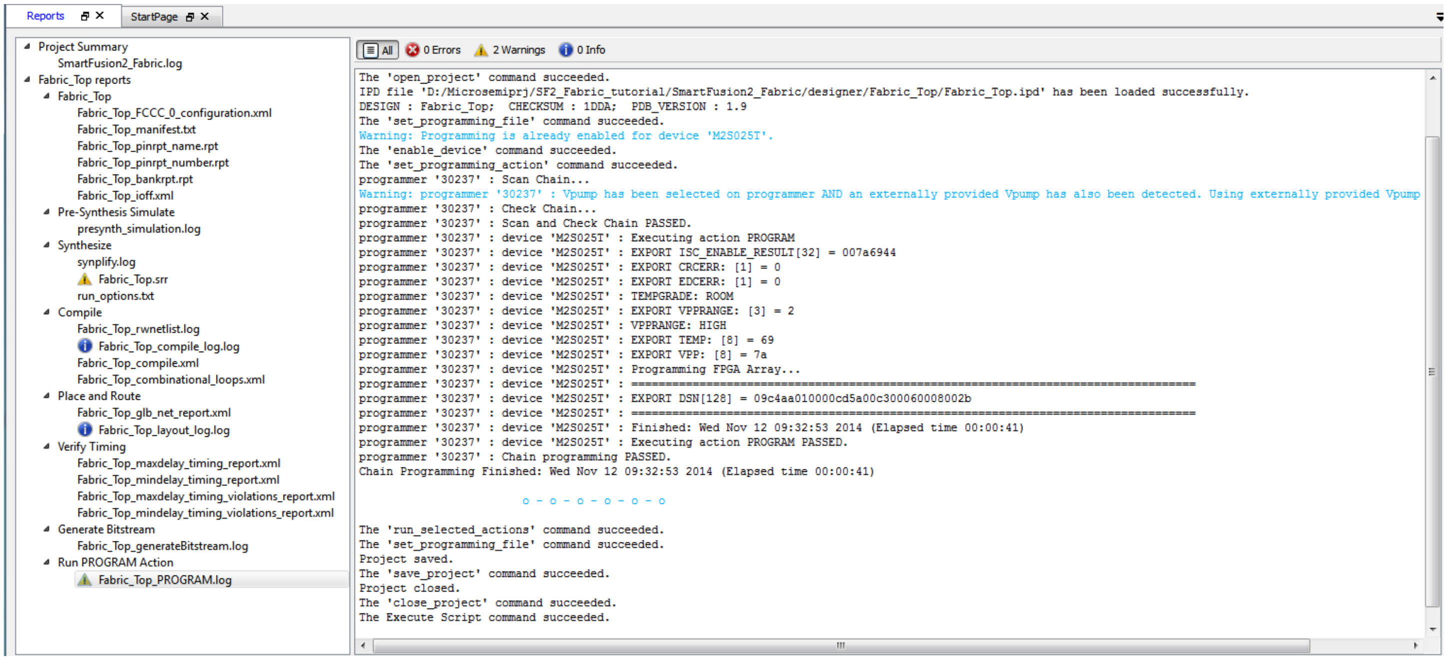
Figure 40 - Launching Programming software from Design Flow window

2. FlashPro will run in batch mode and program the device. Programming messages will be visible in the Libero SoC log window (programmer number and device number may differ).

   Note: Do not interrupt the programming sequence; it may damage the device or the -programmer.

3. The following message should be visible in the Reports view under Program Device when the device is programmed successfully (programmer number will differ; device will differ depending on the target board):

```
programmer '30237' : device 'M2S025T' : Executing action PROGRAM PASSED.
```



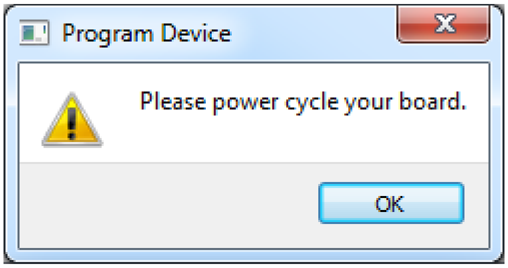
```
The 'open_project' command succeeded.
IPD file 'D:/Microsemiprj/SF2_Fabric_tutorial/SmartFusion2_Fabric/designer/Fabric_Top/Fabric_Top.ipd' has been loaded successfully.
DESIGN : Fabric_Top;  CHECKSUM : 1DDA;  PDB_VERSION : 1.9
The 'set_programming_file' command succeeded.
Warning: Programming is already enabled for device 'M2S025T'.
The 'enable_device' command succeeded.
The 'set_programming_action' command succeeded.
programmer '30237' : Scan Chain...
Warning: programmer '30237' : Vpump has been selected on programmer AND an externally provided Vpump has also been detected. Using externally provided Vpump
programmer '30237' : Check Chain...
programmer '30237' : Scan and Check Chain PASSED.
programmer '30237' : device 'M2S025T' : Executing action PROGRAM
programmer '30237' : device 'M2S025T' : EXPORT ISC_ENABLE_RESULT[32] = 007a6944
programmer '30237' : device 'M2S025T' : EXPORT CRCERR: [1] = 0
programmer '30237' : device 'M2S025T' : EXPORT EDCERR: [1] = 0
programmer '30237' : device 'M2S025T' : TEMPGRADE: ROOM
programmer '30237' : device 'M2S025T' : EXPORT VPPRANGE: [3] = 2
programmer '30237' : device 'M2S025T' : VPPRANGE: HIGH
programmer '30237' : device 'M2S025T' : EXPORT TEMP: [8] = 69
programmer '30237' : device 'M2S025T' : EXPORT VPP: [8] = 7a
programmer '30237' : device 'M2S025T' : Programming FPGA Array...
programmer '30237' : device 'M2S025T' : ===================================================================================
programmer '30237' : device 'M2S025T' : EXPORT DSN[128] = 09c4aa010000cd5a00c300060008002b
programmer '30237' : device 'M2S025T' : ===================================================================================
programmer '30237' : device 'M2S025T' : Finished: Wed Nov 12 09:32:53 2014 (Elapsed time 00:00:41)
programmer '30237' : device 'M2S025T' : Executing action PROGRAM PASSED.
programmer '30237' : Chain programming PASSED.
Chain Programming Finished: Wed Nov 12 09:32:53 2014 (Elapsed time 00:00:41)

                    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.
The 'close_project' command succeeded.
The Execute Script command succeeded.
```

Figure 41 - Programming messages in Libero SoC log window (messages for M2S-EVAL-KIT shown)

4. If you are using the M2S050 ES silicon a dialog box will remind you to power-cycle the board when programming is complete.



Please power cycle your board.

Figure 42 - Reminder to power cycle the board if the target board has ES silicon

5. Click **OK** to close the Program Device dialog box after power cycling the board.

6. A green check mark will appear next to Program Design and Run PROGRAM Action in the Design Flow window to indicate programming completed successfully.

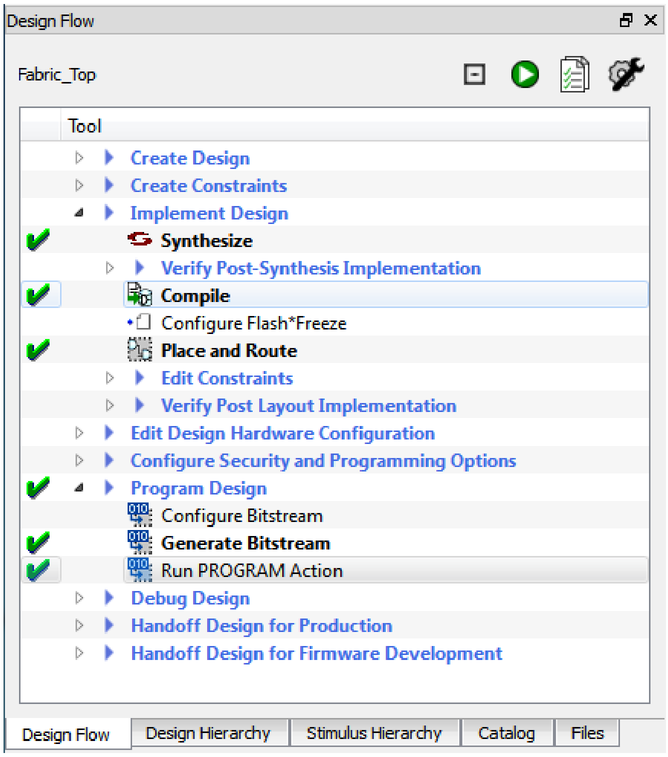


Figure 43 - Design Flow window after programming

7. Close Libero SoC (**Project > Exit**).

Running the Application

8. Reset the board by pressing and releasing the Reset switch. Refer to the table below for switch and LED information for the different target boards.
9. Observe the pattern of the LEDs after resetting the board. Press and hold the User button (refer to table below) and observe the pattern of the LEDs. They should match the description in the table on page 4.
10. When finished, remove power from the board.

Target Board	Reset Switch	User Button	LED[1:0]
Starter kit (All)	SW1	SW2	DS4, DS3
Evaluation kit (All)	SW6	SW1	F4, E1
Development kit	SW9	SW1	LED2, LED1
Advanced Development kit	SW6	SW1	DS1, DS0

Table 5 – Switches and LEDs for target boards

End of SmartFusion2 Fabric lab

Appendix 1 - Jumper settings for SmartFusion2 Starter kit

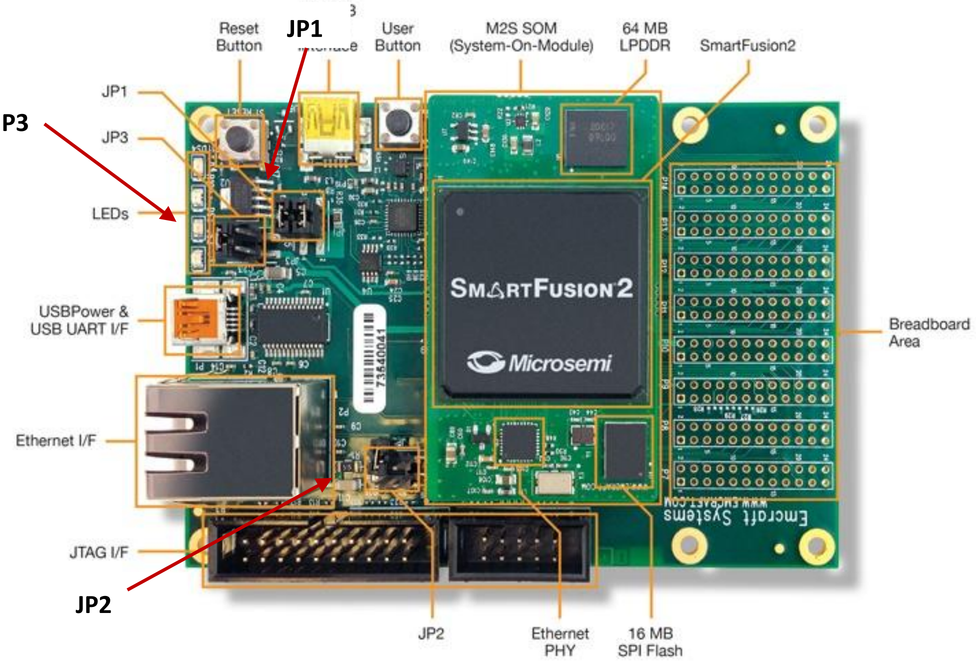


Figure 44 – SmartFusion2 Starter Kit

1. Prior to programming (and powering up) the SmartFusion2 Starter Kit board, confirm that the jumpers are positioned as show in the table below.

Jumper	Location	Function	Setting
JP1	Below the USB OTG connector in picture above	Enable power on the SmartFusion2 SOM (VCC3)	1-2 installed 3-4 open
JP2	To the right of the RJ-45 Ethernet jack in the picture above	Select appropriate JTAG mode and enable power to the SmartFusion2 JTAG connector	1-2 open 3-4 installed
JP3	Above the USB power and UART I/F connector in the picture above	Select the mini-USB port as the power source	1-3 open 2-4 installed

Table 6 – Jumper settings for the SmartFusion2 Starter kit

2. Connect the mini-USB Y-cable into the mini-USB connector labeled USB Power & USB UART I/F (P1) connector on SOM-BSB-EXT board (see figure above). Connect the other end of the cable to a USB port on your PC. As soon as the connection to the PC has been made, the on-board LED DS2 will illuminate, indicating that the board has power.

   A single USB connection provides a 500 mA power to the SF2-STARTER-KIT-ES, which is sufficient for basic functionality. Note however that some advanced operations, such as WiFi connectivity using the USB WiFi module, may require more than 500 mA for reliable operation. Use the second link of the mini-USB Y-cable to connect to the PC for such configurations.

   If prompted, install the FT232R drivers. The drivers can be downloaded from http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files

3. Plug the FlashPro4 ribbon cable into connector P5 on the Starter Kit board.
4. Connect the mini USB cable between the FlashPro4 and the USB port of your PC.

5. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.

Appendix 2 - Jumper settings for SmartFusion2 Evaluation, Security Evaluation and IGLOO2 Evaluation kit

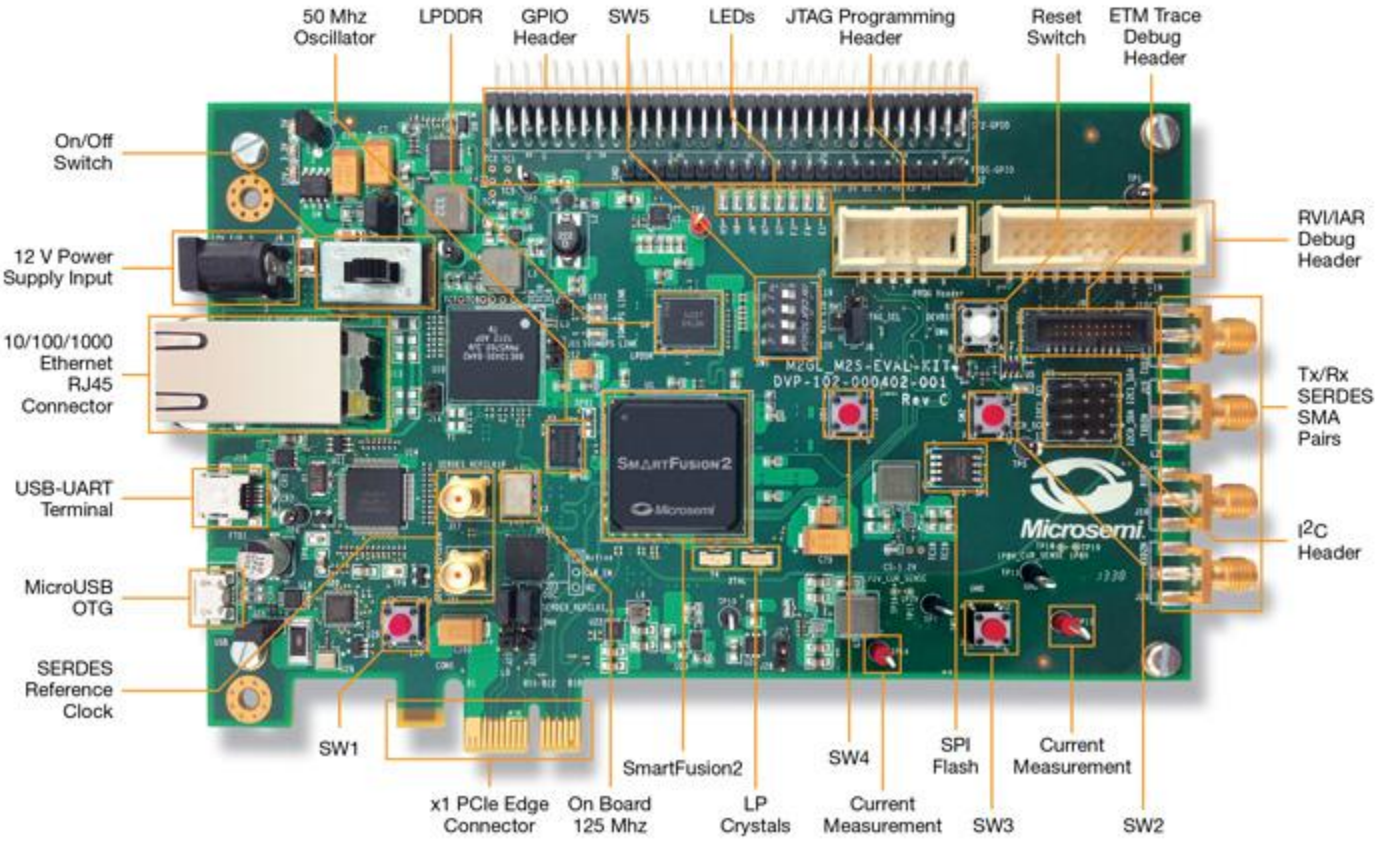


Figure 45 - SmartFusion2 Evaluation kit / Security Evaluation kit

1. Prior to programming (and powering up) the SmartFusion2 Evaluation Kit board, confirm that the jumpers are positioned as show in the table below.

Jumper	Location	Function	Setting
J3	Upper left-hand corner of board near the power switch (SW7)	Jumpers to select either SW2 input or signal ENABLE_FT4232 from FT4232H chip.	1-2 installed
J8	Below JTAG programming header	JTAG selection jumper to select between RVI header and FP4 header for application debug. • Pin 1-2 FP4 for SoftConsole/FlashPro • Pin 2-3 RVI for Keil ULINK™/IAR J-Link® • Pin 2-4 for Toggling JTAG_SEL signal remotely using GPIO capability of FT4232 chip.	1-2 installed
J22	Above PCIe connector	Jumper to select the output enables control for the line side outputs • Pin 1-2 (Line side output enabled) • Pin 2-3 (Line side output disabled)	1-2 installed
J23	Above PCIe connector	• Jumper to select switch-side Mux inputs	1-2 installed

		of A or B to the line side. • Pin 1-2 (Input A to the line side) that is on board 125 MHz differential clock oscillator output will be routed to line side. • Pin 2-3 (Input B to the line side) that is external clock required to source through SMA connectors to the line side.	
J24	Lower left-hand corner of board near the USB connector	Jumper to provide the VBUS supply to USB when using in Host mode.	1-2 installed
H1	Right side of board near SMA connectors	Connect I2C0_SCL to I2C1_SCL Connect I2C0_SDA to I2C1_SDA	6-10 installed 7-11 installed

Table 7 – Jumper settings for the SmartFusion2 Evaluation kit

2. Connect 12 V power supply brick to J6 to power the board.
3. Slide the main power switch SW7 to ON position.
4. Connect the FP4 header to J5.
5. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.

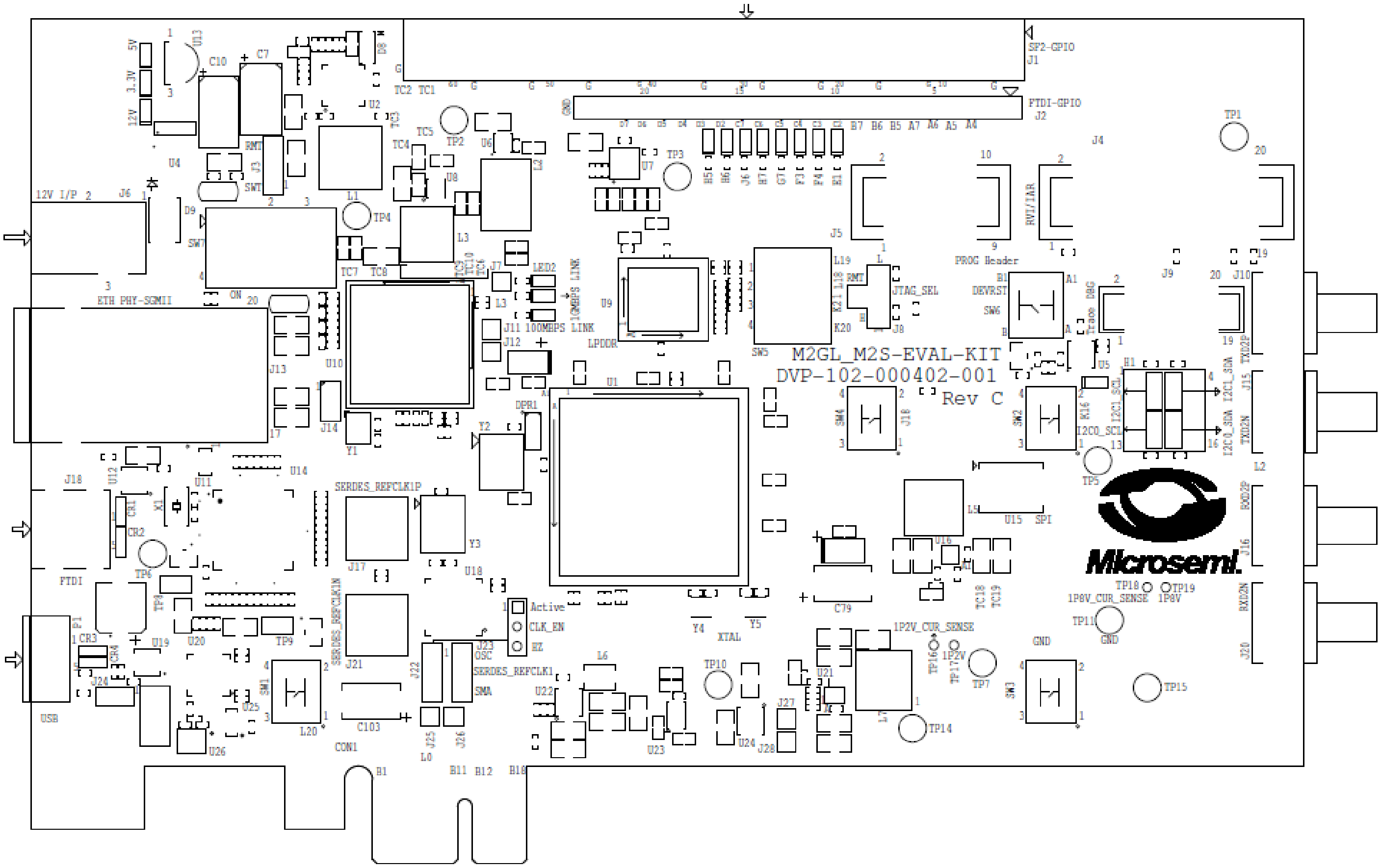


Figure 46 - SmartFusion2 Evaluation kit Silkscreen Top View

Appendix 3 - Jumper settings for SmartFusion2 Development kit

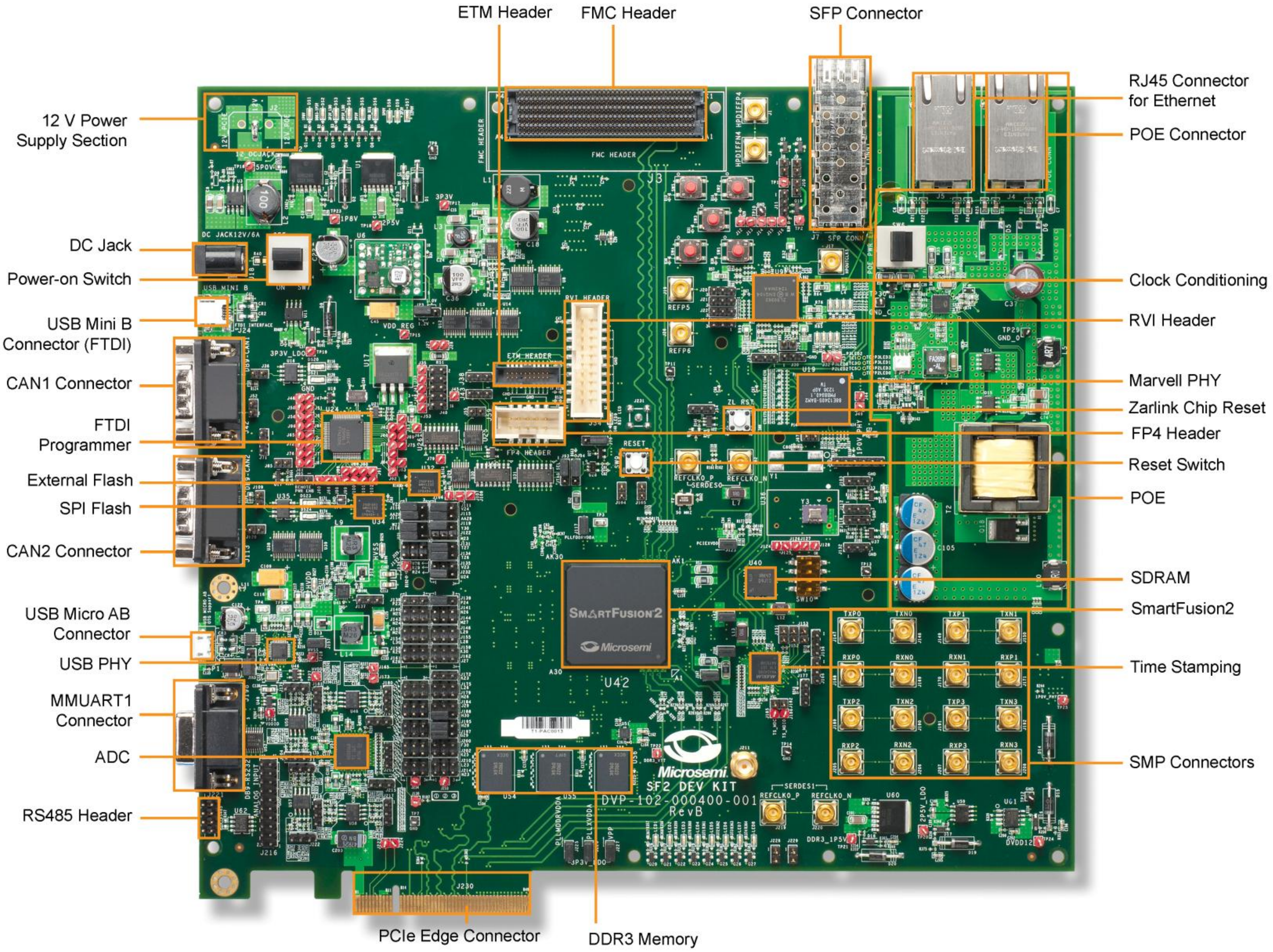


Figure 47 - SmartFusion2 Development kit

1. Prior to programming (and powering up) the SmartFusion2 Development Kit board, confirm that the jumpers are positioned as show in the table below.
2. Connect 12 V power supply brick to J18 to power the board.
3. Slide the main power switch SW7 to ON position.
4. Plug the FlashPro4 ribbon cable into connector J59 on the Development Kit board.
5. Connect the mini USB cable between the FlashPro4 and the USB port of your PC.
6. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.

Jumper	Location	Function	Setting
J2	Upper left-hand corner of the board	Jumper to select the power supply option	1-3 installed (External DC Jack)
J23	Below regulator U6	Jumper to select the core voltage (VDD_REG) to either 1.0 V or 1.2 V	2-3 installed (1.2V core voltage)
J117	Above SmartFusion2 device (U42)	Jumper to connect 3P3V_LDO to PLLFDDRVDDA	1-2 installed
J123	Above U40	Jumper to connect VDD_REG to PCIExVDD	1-2 installed
J142	To the right of the SmartFusion2 device (U42)	Jumper to connect 3P3V_LDO to PLLPCIExVDDA	1-2 installed
J157	Lower left side of board to the left of L14	Jumper to connect 3P3V_LDO to VPPNVMSA0	1-2 installed
J160	To the right of the SmartFusion2 device (U42)	Jumper to connect VDD_REG to PCIExVDDIOx	1-2 installed
J167	To the right of the SmartFusion2 device (U42)	Jumper to connect 2P5V_LDO to PCIExVDDPLL	1-2 installed
J225	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to PLLMDDRVDDA	1-2 installed
J226	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to PLLXVDDA	1-2 installed
J227	Bottom edge near PCIe connector	Jumper to connect 3P3V_LDO to VPP	1-2 installed
J70	Below 20 pin RVI Header	Device reset generation selection • Pin 1-2: Reset depends on 3.3 V rail • Pin 2-3: Reset depends on FLASH_GOLDEN pin	1-2 installed
J93	Below 10 pin FP4 header	JTAG selection jumper to select between RVI header or FP4 header for application debug • Pin 1-2: FP4 for SoftConsole / FlashPro4 • Pin 2-3: RVI for Keil ULINK™/IAR J-Link®	1-2 installed 2-3 open
J94	Below 10 pin FP4 header	Jumper to select the JTAG reset • Pin 1-2: selects reset from FP4 header/RVI header depending on JTAG selection • Pin 2-3: SPI flash programming mode	1-2 installed 2-3 open
J110	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SCK and FMC_V22 • Pin 1-2 SPI flash • Pin 2-3 FMC	open
J118	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SDO and FMC_W27 • Pin 1-2 SPI flash • Pin 2-3 FMC	1-2 installed
J119	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SDI and FMC_Y30 • Pin 1-2 SPI flash • Pin 2-3 FMC	1-2 installed
J121	To the right of the DB9 CAN2 connector	Jumper to select between SPI flash SS and FMC_W28 • Pin 1-2 SPI flash • Pin 2-3 FMC	1-2 installed

J129	To the right of the DB9 CAN2 connector	Jumper to select between FT4232 DD0 and FMC_R29 to SPI_1_SS1 • Pin 1-2 FMC • Pin 2-3 FT4232	2-3 installed
J133	To the right of the DB9 CAN2 connector	Jumper to select between FT4232 DD1 and FMC_R24 to SPI_1_SS2 • Pin 1-2 FMC • Pin 2-3 FT4232	2-3 installed
J139	To the left of the SmartFusion2 device (U42)	Jumper to select between USB reset and FMC_P24 • Pin 1-2 USB • Pin 2-3 FMC	1-2 installed
J163	Near the DB9 RS-232 connector	Jumper to select the USB mode of operation • Pin 1-2 On-The-Go (OTG) mode • Pin 2-3 Either host or device mode	1-2 installed
J164	Near the DB9 RS-232 connector	Jumper to provide the VBUS supply to USB when using in Host mode	open

Table 8 - Jumper settings for the SmartFusion2 Development kit

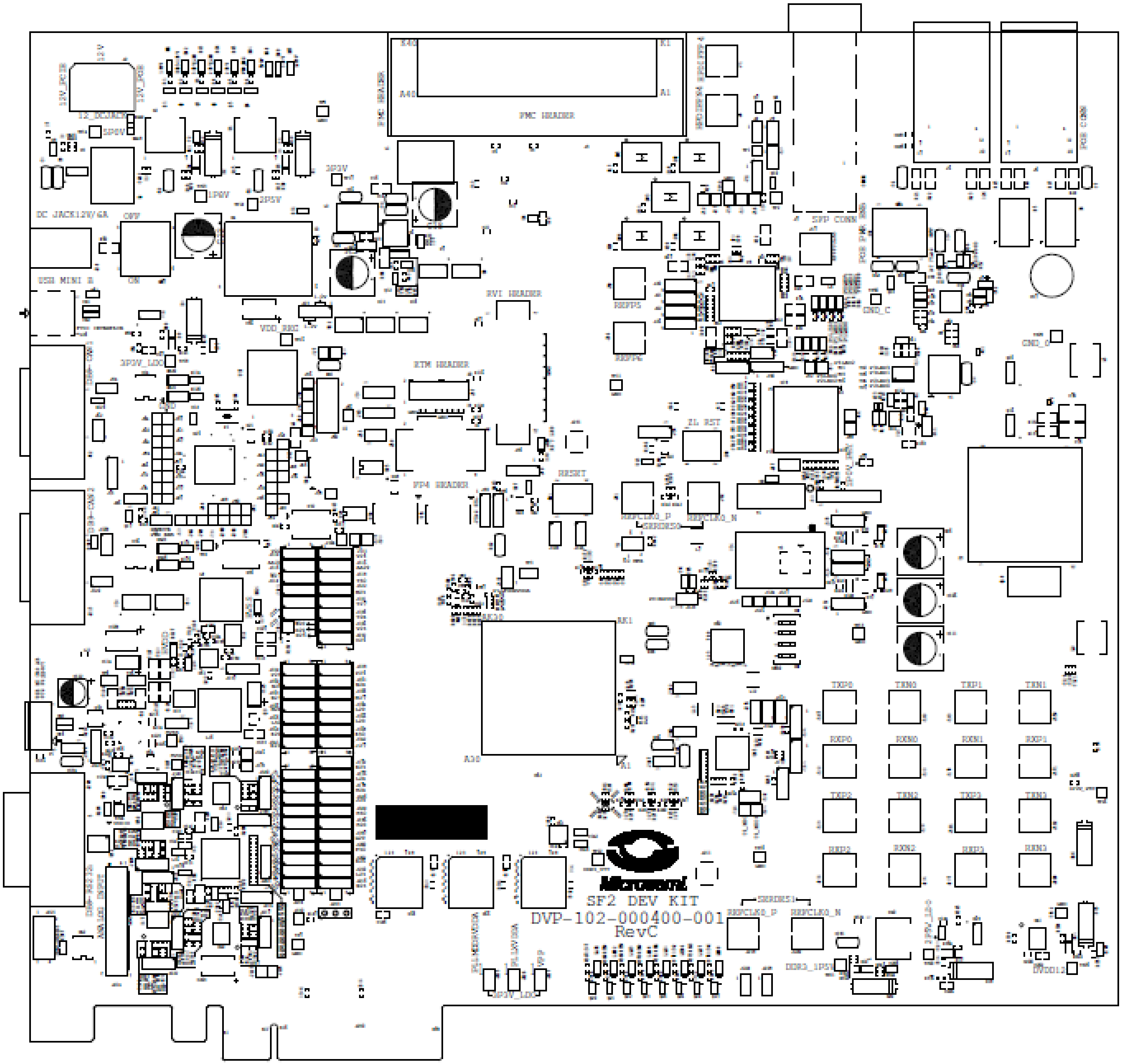


Figure 48 - SmartFusion2 Development kit Silkscreen Top View

Appendix 4 - Jumper settings – SmartFusion2 Advanced Development kit

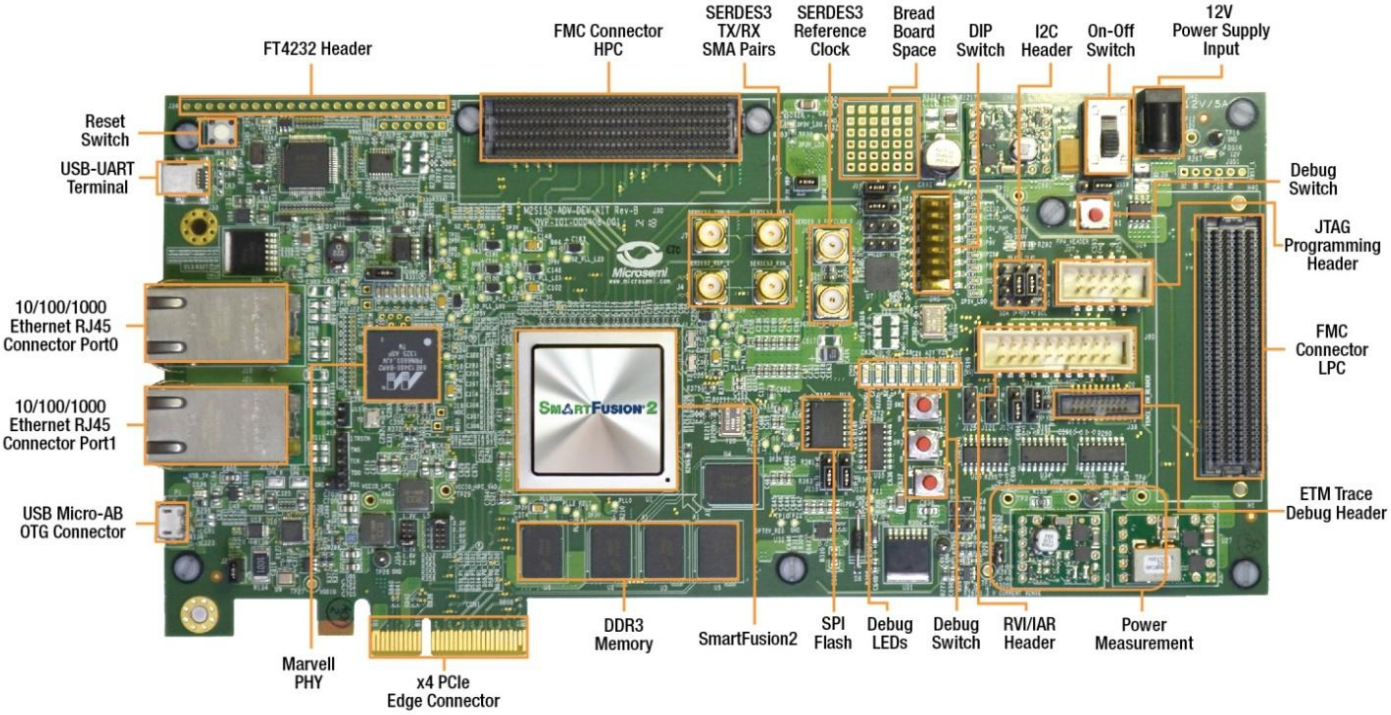


Figure 49 - SmartFusion2 Advanced Development kit

1. Prior to programming (and powering up) the SmartFusion2 Development Kit board, confirm that the jumpers are positioned as show in the table below.
2. Connect 12 V power supply brick to J42 to power the board.
3. Slide the main power switch SW7 to ON position.
4. Connect the FP4 header to J37.
5. Install the FlashPro4 drivers if prompted. The drivers are located in the <*FlashPro Installation Directory*>*Drivers* folder.

Jumper	Location	Function	Setting
J123	Bottom edge of the board to the right of U31.	Jumper to select the core voltage (VDD_REG) to either 1.0V or 1.2V • Pin 1–2 for 1.0V core voltage • Pin 2–3 for 1.2V core voltage	2-3 installed (1.2V core)
J353	Above PCIe connector.	Jumper to select the core voltage (VCCIO_HPC_VADJ) to either 3.3V or 2.5V or 1.8V or 1.5V or 1.2V • Pin 1–2 for 3.3V • Pin 3–4 for 2.5V • Pin 5–6 for 1.8V • Pin 7–8 for 1.5V • Pin 9-10 for 1.2V	1-2 installed
J354	Above PCIe connector.	Jumper to select the core voltage (VCCIO_LPC_VADJ) to 2.5 V or 1.8 V or 1.5 V or 1.2V	1-2 installed

		• Pin 1–2 for 2.5 V • Pin 3–4 for 1.8 V • Pin 5–6 for 1.5 V • Pin 7-8 for 1.2 V	
J116	Below on/off switch (SW7).	Jumpers to select either SW7 input or signal ENABLE_FT4232 from FT4232H chip • Pin 1–2 for SW7 selection • Pin 2-3 for “Enable_FT4232” signal control	1-2 installed
J8	Left of DIP switch (SW5).	Jumper to select the output enables control for the line side outputs • Pin 1-2 (Line side output enabled) • Pin 2-3 (Line side output disabled)	1-2 installed
J11	Left of DIP switch (SW5).	Jumper to select switch-side Mux inputs of A or B to the line side • Pin 1-2 (Input A to the line side) that is on board 125 MHz differential clock oscillator output is routed to line side • Pin 2-3 (Input B to the line side) that is on board 100 MHz differential clock oscillator output is routed to line side	1-2 installed
J32	Left of TRACE_TEM_HEADER (J38).	JTAG selection jumper to select between RVI header or FP4 header for application debug • Pin 1-2 FP4 for Soft Console/Flash Pro • Pin 2-3 RVI for Keil ULINK™/IAR J-Link® • Pin 2-4 for JTAG_SEL pin to DD1 signal of FT4232H chip	1-2 installed
J121	Left of TRACE_TEM_HEADER (J38).	Select FTDI JTAG/ SPI Slave programming • Pin 1-2 FTDI JTAG programming • Pin 2-3 FTDI SPI Slave Programming	1-2 installed
J124	Left of TRACE_TEM_HEADER (J38).	Select JTAG programming via FP4 or FTDI • Pin 1-2 JTAG programming via FTDI • Pin 2-3 JTAG programming via FP4	2-3 installed
J118	Left of push-button switches.	Select programming SPI-0 flash through FTDI SPI-0 (Port-B) or SmartFusion2 SPI-0 • Pin 1-2 Programming SPI-0 flash via SmartFusion2 SPI-0 • Pin 2-3 Programming SPI-0 flash via FTDI SPI-0 (Port-B) and J125 pin 2-3 must be shorted	1-2 installed
J119	Left of push-button switches.	Select programming SPI-1 flash through FTDI SPI (Port-B) or SmartFusion2 SPI-1 • Pin 1-2 Programming SPI-1 flash via SmartFusion2 SPI-1 • Pin 2-3 Programming SPI-1 flash via FTDI SPI (Port-B) and J125 pin 1-2 must be shorted	1-2 installed

Table 9 - Jumper settings for the SmartFusion2 Advanced Development kit

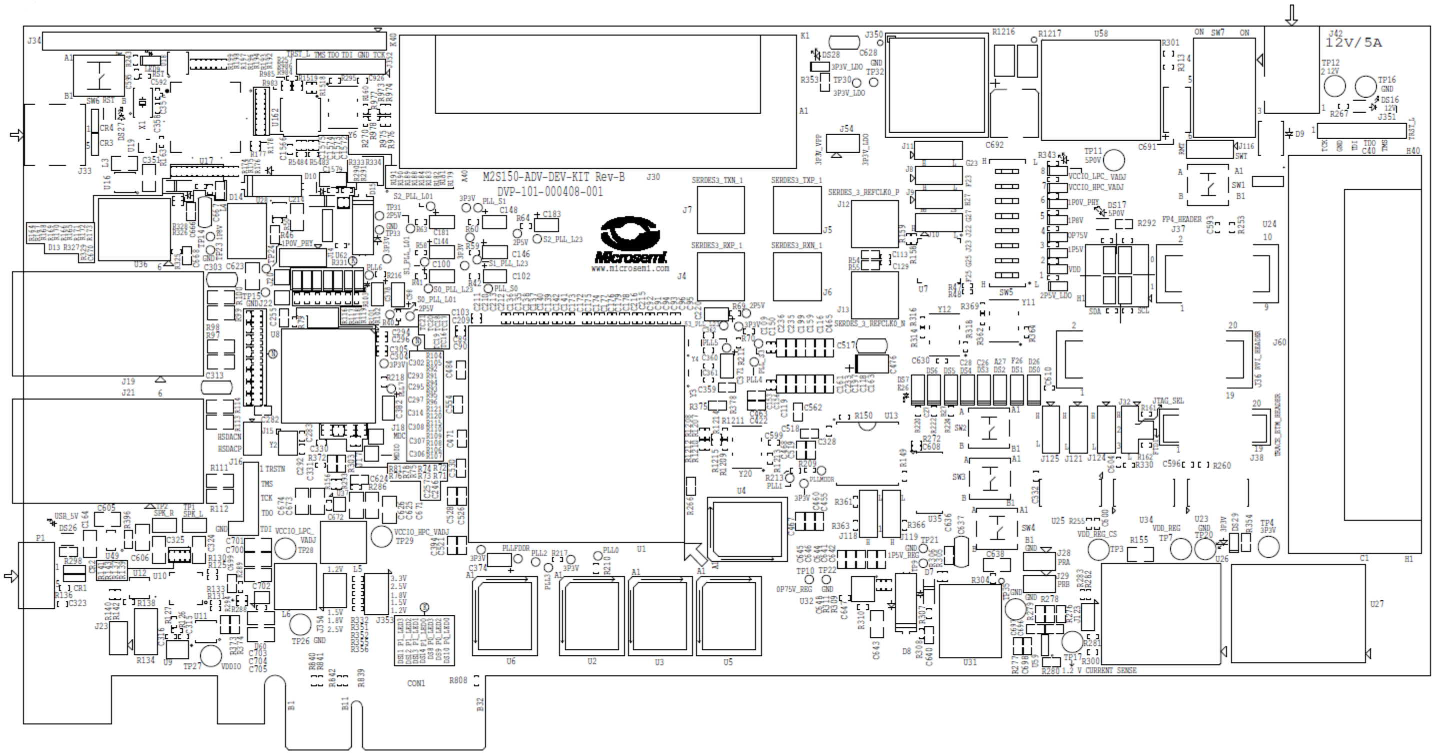


Figure 50 - SmartFusion2 Advanced Development kit Silkscreen Top View

Record answers to questions below

Name: ______________________________

Page 26

Record the number of sequential and combinatorial cells used in the design below.

LUTs (4LUT) ______________________

Flip-flops (DFF) ______________________

HDL Language used: ______________________

Record the maximum frequency for the clock CCC_0/CCC_INST/INST_CCC_IP:GL0 below:

FCCC_0/CCC_INST/INST_CCC_IP:GL0: ______________________

Answers to Questions

Page 28

Record the number of sequential and combinatorial cells used in the design below.

4LUT 45 (Verilog or VHDL)

DFF 24 (Verilog or VHDL)

Record the maximum frequency for the clock CCC_0/CCC_INST/INST_CCC_IP:GL0 below:

FCCC_0/CCC_INST/INST_CCC_IP:GL0: 394.789 MHz

List of Figures