Interfacing User Logic with the Microcontroller Subsystem

Libero SoC Design Flow Tutorial User's Guide

Microsemi

Table of Contents

Interfacing User Logic with the Microcontroller Subsystem

Introduction

This tutorial shows how to interface and handle communication between user logic in the field programmable gate array (FPGA) fabric and the SmartFusion®2 microcontroller subsystem (MSS). It also explains the Microsemi® Libero® System-on-Chip (SoC) design software tool flow for designing applications for the SmartFusion2 system-on-chip (SoC) FPGA family of devices.

A SmartFusion2 SoC FPGA device has two fabric interface controllers (FIC_0 and FIC_1) as a part of the MSS. These FIC blocks provide a means of interfacing from the SmartFusion2 SoC FPGA MSS AHB-Lite (AHBL) bus to user masters or user slaves in the FPGA fabric. Each FIC block performs an AHBL to AHBL or AHBL to APB3 bridging function between the AHB Bus Matrix and AHBL or APB3 bus in the FPGA fabric. Each FIC block provides two bus interfaces between the MSS and FPGA fabric. The first one is mastered by the MSS and has slaves in the FPGA fabric; the second one has a master in the fabric and slaves in the MSS. The bus interfaces to the FPGA fabric can be either 32-bit AHBL or 32-bit APB type. The FIC block provides registered bridging between the MSS AHBL interface and the FPGA fabric AHBL/APB circuitry to run at frequency ratios of 1:1, 2:1, 4:1, 8:1, 16:1, or 32:1. In AHB-Lite configuration, a bypass mode is provided, in which signals to and from the fabric are not registered and hence requires fewer clock cycles to complete each transaction. SmartFusion2 SoC FPGA FIC has six memory regions. You can allocate a memory region to a particular FIC that is either to FIC_0 or FIC_1. Each memory region has a predefined memory map. Refer to the Fabric Interface Controller chapter of the *SmartFusion2 Microcontroller Subsystem User Guide* for more information on FIC blocks.

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

1. Creating a project for a SmartFusion2 SoC FPGA using the Microsemi Libero SoC toolset.
2. Using SmartFusion2 SoC FPGA System Builder to Configure MSS and generate System Builder Component.
3. Configuring fabric interface controllers (FIC_0 and FIC_1) to interface user logic in the fabric with the MSS.
4. Using on-chip oscillators and fabric CCC (FAB_CCC) for generating system clocks.
5. Writing a simple bus functional model (BFM) script for simulating the design.
6. Verifying the design by running BFM commands.
7. Generating the programming file to program the SmartFusion2 device.
8. Opening the project in SoftConsole from Libero SoC and writing the application code.
9. Validating the application design on SmartFusion2 Board.

Requirements and Details

Table 1 • **Reference Design Requirements and Details**

Reference Design Requirements and Details	Description
Hardware Requirements	
SmartFusion2 Development Kit board or SmartFusion2 Starter Kit board - SF2-STARTER-KIT ES- 2. FlashPro4 programmer	Rev C or later
USB Cables	-
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	11.3
SoftConsole	3.4
Host PC Drivers	USB Drivers

Project Files

You can download the associated project files for this tutorial from the Microsemi website: www.microsemi.com/soc/download/rsc/?f=SmartFusion2_FIC_Tutorial_DF

The project files include the following:

- Source
- Solution
- Programming file
- Readme file

Refer to the Readme.txt file provided in the design files for the complete directory structure.

Design Description

The design uses the SmartFusion2 SoC FPGA MSS block, one CCC block, on-chip 25/50 MHz RC oscillator and two different slaves in the FPGA fabric. The MSS is configured with FIC_0 and FIC_1 enabled, FIC_0 is configured for the AHBL master interface and FIC_1 is configured for the APB3 master interface. Choosing this configuration allows the application to access two different types of FPGA fabric peripheral slaves from two different masters in the MSS. The slaves in the FPGA fabric are CoreAHBLSRAM and CoreGPIO. CoreAHBLSRAM is connected to FIC_0 through an AHBL bus interface and CoreGPIO is connected to FIC_1 through an APB3 bus interface. Figure 1 shows the block diagram of the design. The ARM® Cortex™-M3 processor or any other MSS master can access these slaves via the FIC blocks. In this design, using BFM models, you will verify the bus read and writes to the fabric peripherals from the MSS side. Using a BFM script, you will perform reads and writes to the CoreAHBLSRAM memory, configure the CoreGPIO block, and set GPIO outputs. In this design, you will validate the bus read and writes to the CoreAHBLSRAM, and setting the GPIO to blink the LEDs on the SmartFusion2 Development Kit Board and SmartFusion2 Starter Kit Board.

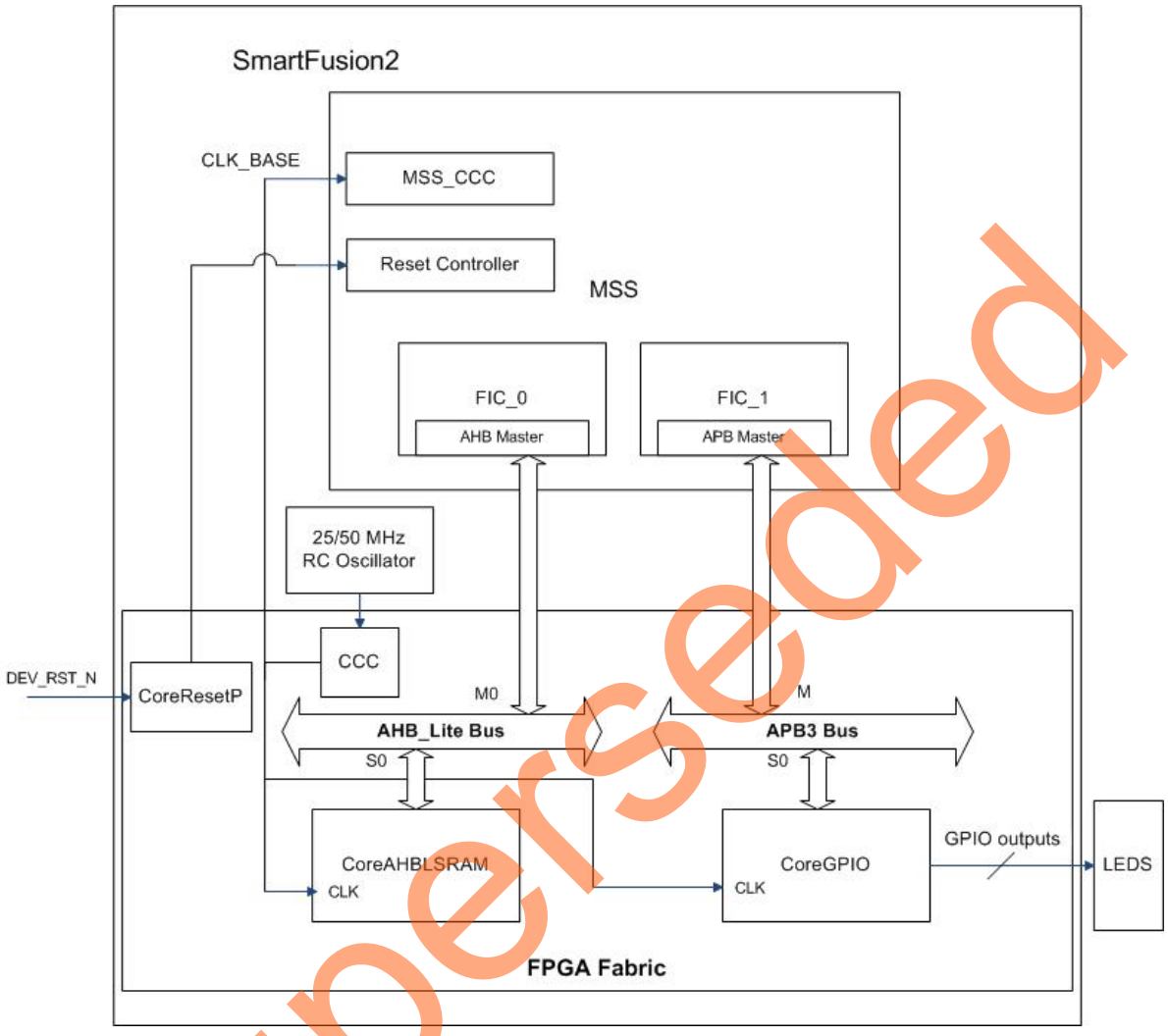

***Figure 1* • Block Diagram of the Design**

Design Steps

The major steps to run this tutorial are as follows:

1. Creating a new Libero SoC project for SmartFusion2 SoC FPGA.
2. Using SmartFusion2 SoC FPGA System Builder to configure the FIC blocks, and clock.
3. Writing user BFM script to simulate a design.
4. Simulating the design using BFM Models and ModelSim.
5. Generate a programming file to program the SmartFusion2 SoC device.
6. Open the software project in SoftConsole and write the application program.
7. Run the design on the SmartFusion2 Development Kit Board or SmartFusion2 Starter Kit Board.

Step 1: Creating a New Libero SoC Project

1. Open Libero SoC design software (**Start > Programs > Microsemi Libero SoC 11.3 > Libero SoC 11.3**) or click the Libero SoC shortcut available on your desktop. The version number of the Libero SoC design software depends on the version that is installed on your PC. You can use either v11.3 or latest.
2. Select **New Project** from the **Project** menu. Enter the information shown below in the New Project wizard dialog box.
    - Project Name: SmartFusion2_FIC_Tutorial
    - Location: Select an appropriate location (for example, D:/Microsemi_prj)
    - Preferred HDL type: Verilog
    - Family: SmartFusion2
    - Die: M2S050
    - Package: 896 FBGA
    - Speed: -1
    - Core Voltage(V): 1.2
    - Operating Conditions: COM
    - PLL Supply Voltage(V): 2.5V
    - Design Templates and Creators: Select **Use Design Tool** and select **Use System Builder** under core section.

Note: For SmartFusion2 Starter Kit (SF2-STARTER-KIT-ES-2): Die: M2S050T_ES

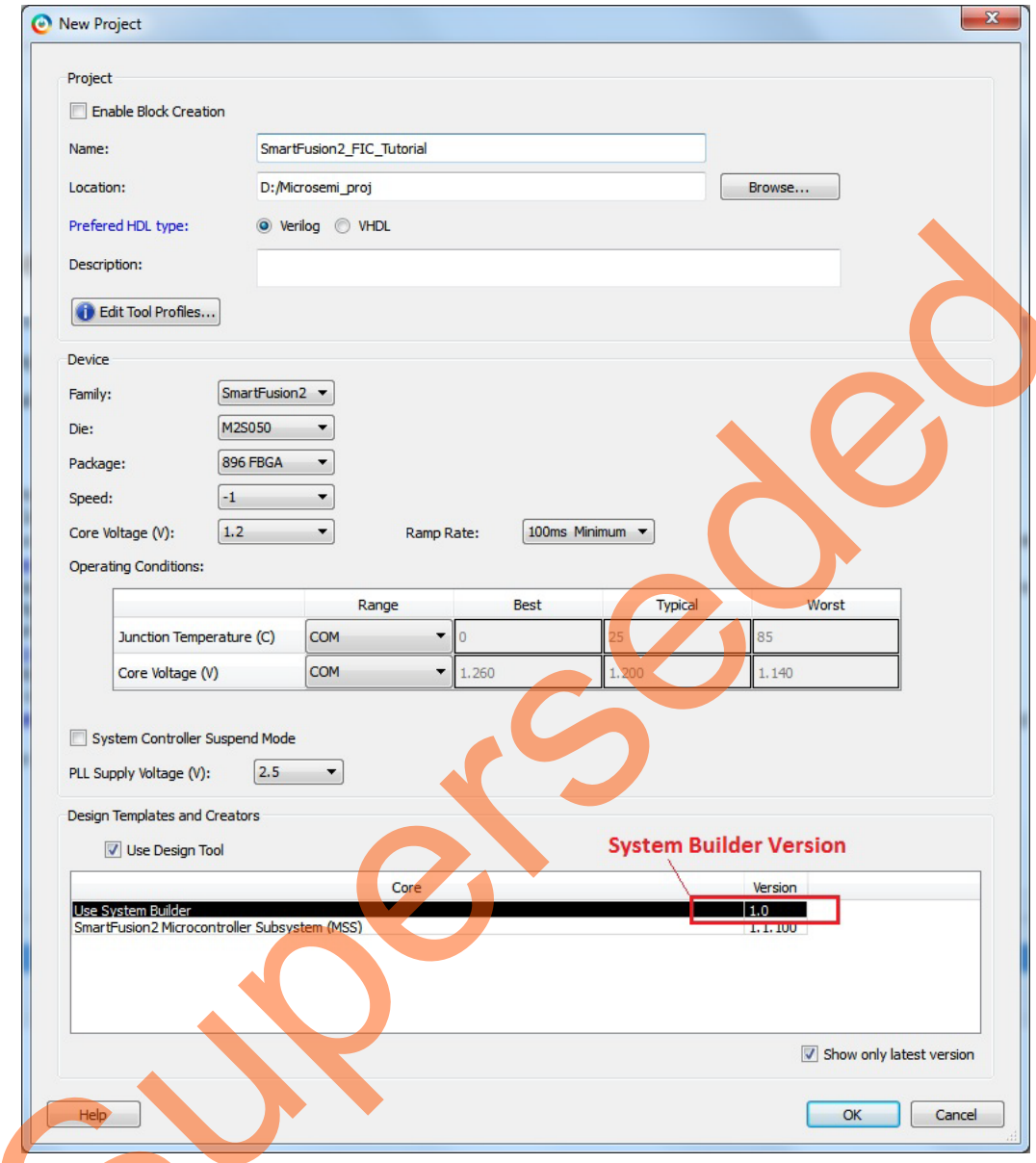

Figure 2 • **New Project Wizard Dialog Box**

3. Select **Use System Builder** in the Design Templates and Creators of the **New Project** window.
4. Click **OK** on the **New Project** window.

5. Since you selected **Use System Builder**, as shown in Figure 2, then **Enter a name for your system** dialog box is displayed, as shown in Figure 3.

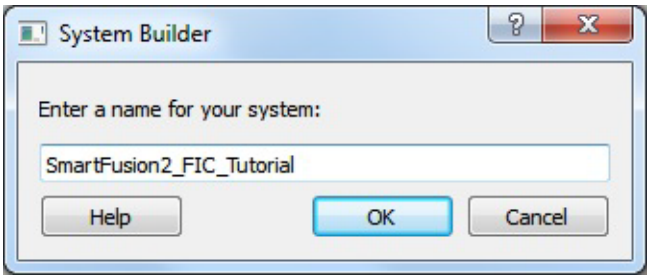

***Figure 3* • Libero SoC Project Window**

6. Enter **SmartFusion2_FIC_Tutorial** as the name of the system and click **OK**. The **System Builder** window is displayed, as shown in Figure 4.

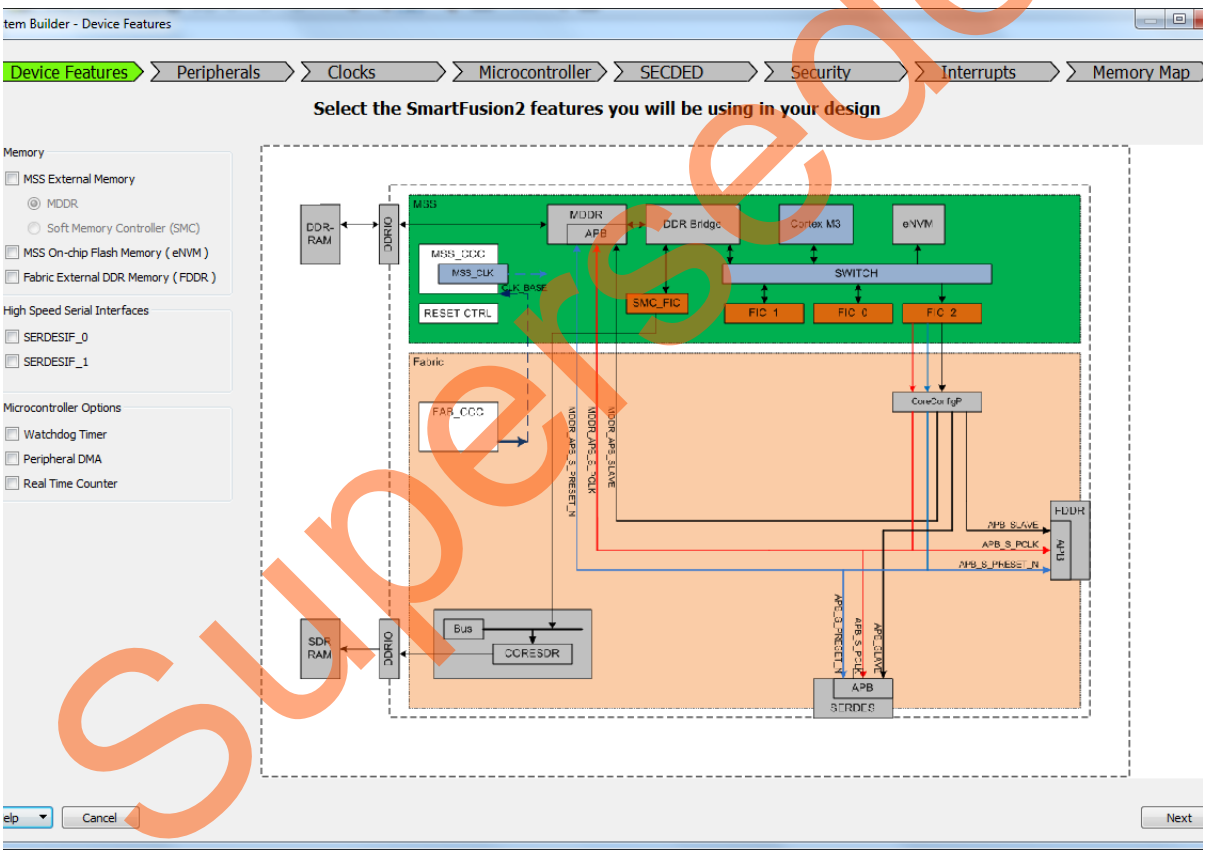

***Figure 4* • SmartFusion2 SoC FPGA System Builder Device Features**

7. Select **Next**. **System Builder- Peripherals** page is displayed, as shown in Figure 5.

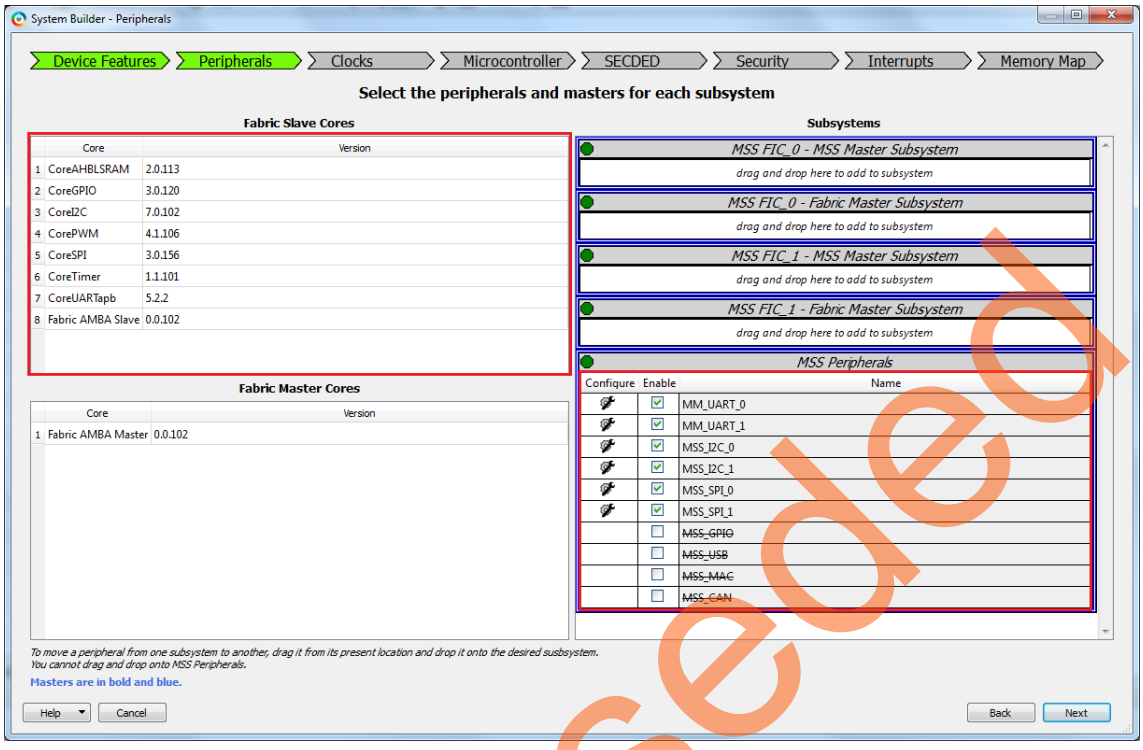

***Figure 5* • SmartFusion2 System Builder Peripherals**

This tutorial will use MSS peripherals; therefore select **MMUART** and uncheck all other peripherals.

This tutorial uses CoreAHBLSRAM and CoreAPB IPs; drag and drop CoreAHBLSRAM to MSS_FIC_0 - MSS Master Subsystem and drag and drop CoreGPIO to MSS_FIC_1 - MSS Master Subsystem as shown in Figure 6.

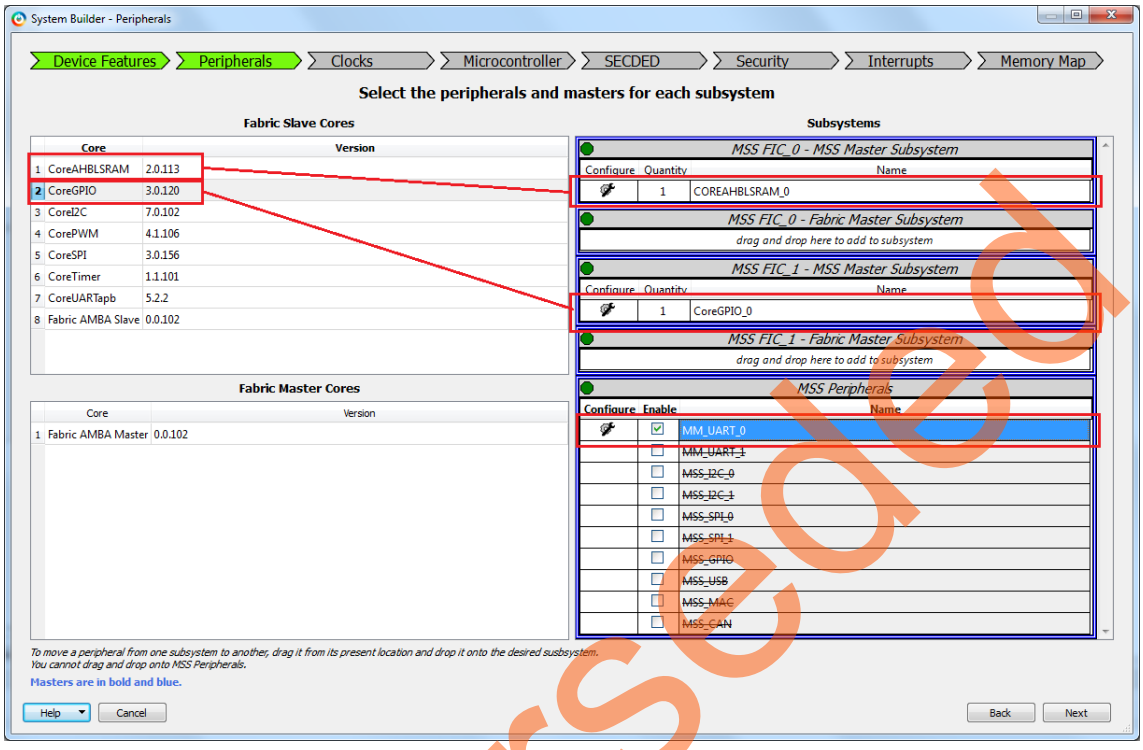

***Figure 6* • SmartFusion2 System Builder MSS Peripherals**

8. Configure COREAHBLSRAM_0, by clicking Configure icon as shown in Figure 7.

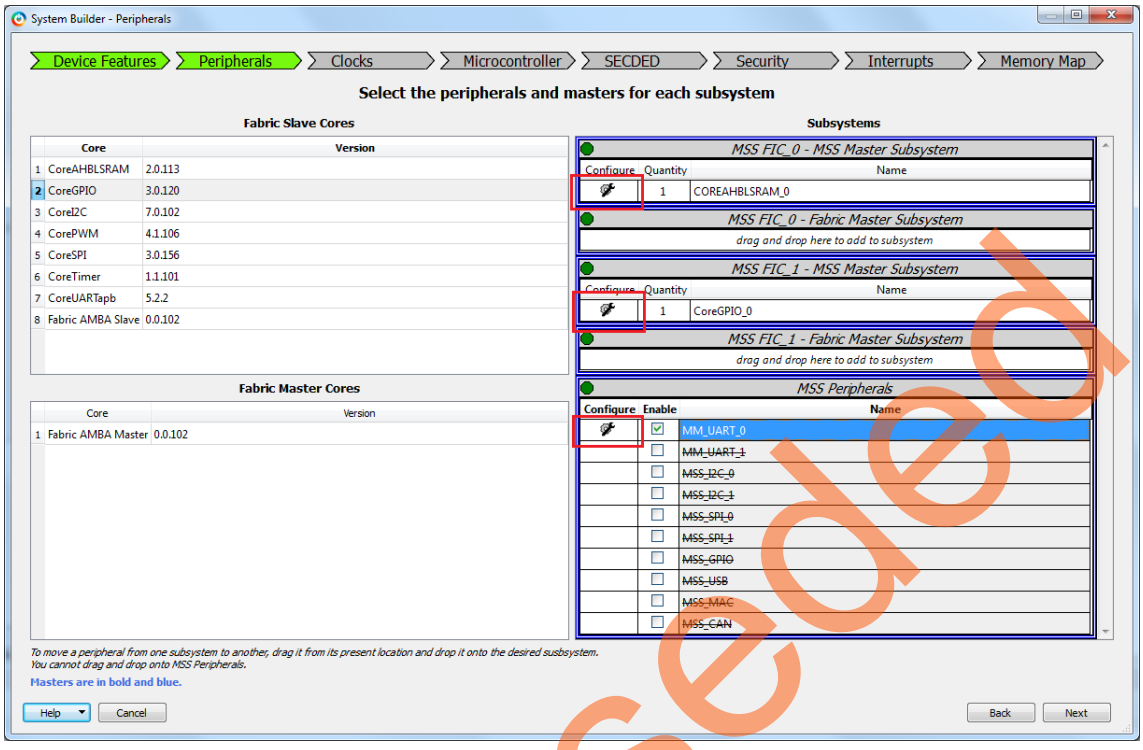

Figure 7 • **CoreAHBLSRAM Configuration**

Use the settings as shown in Figure 8.

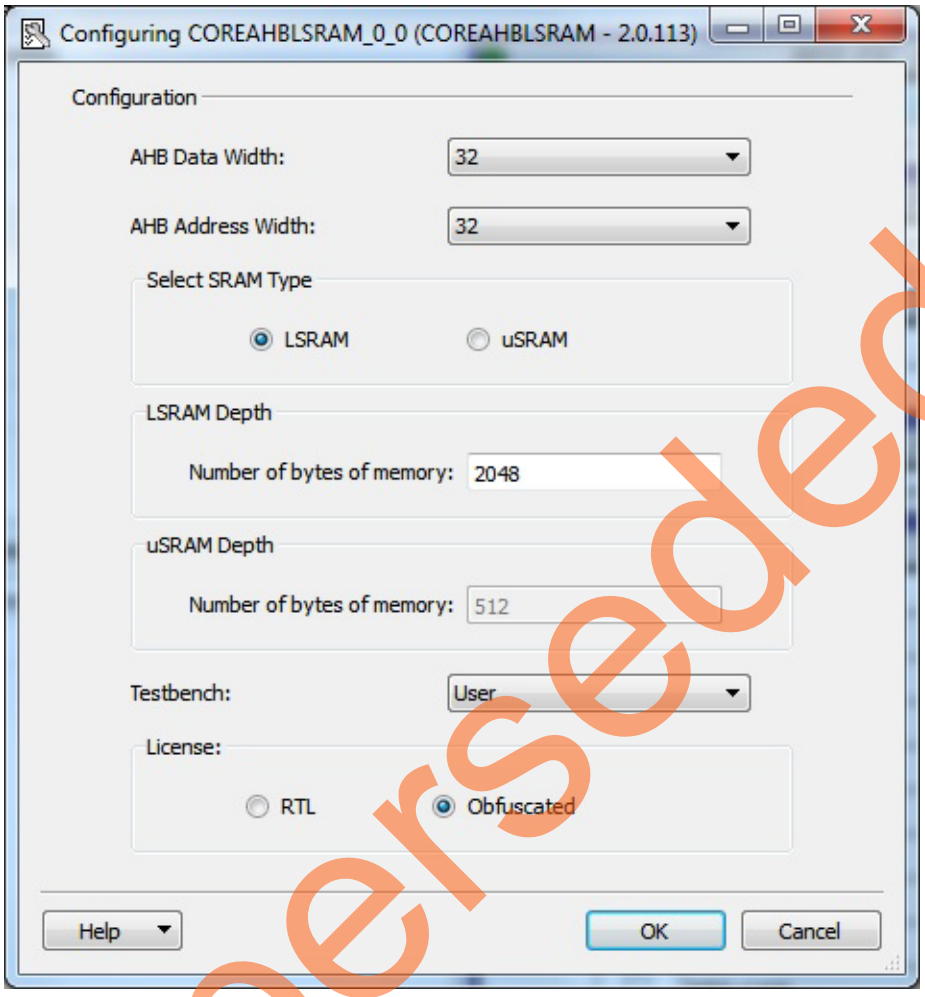

***Figure 8* • CoreAHBLSRAM Configuration**

9. Click **OK** after completion of COREAHBLSRAM configuration.
10. Click **CoreGPIO** Configure icon as shown in Figure 7, and use the following settings for SmartFusion2 Development Kit Board as shown in Figure 9, and keep the rest at default states
    - Number of I/Os: 8 - For SmartFusion2 Development Kit
    - Output enable: Internal
    - Check Fixed Config check box
    - I/O Type: Output

Note: Number of I/Os: 2 - For SmartFusion2 Starter Kit

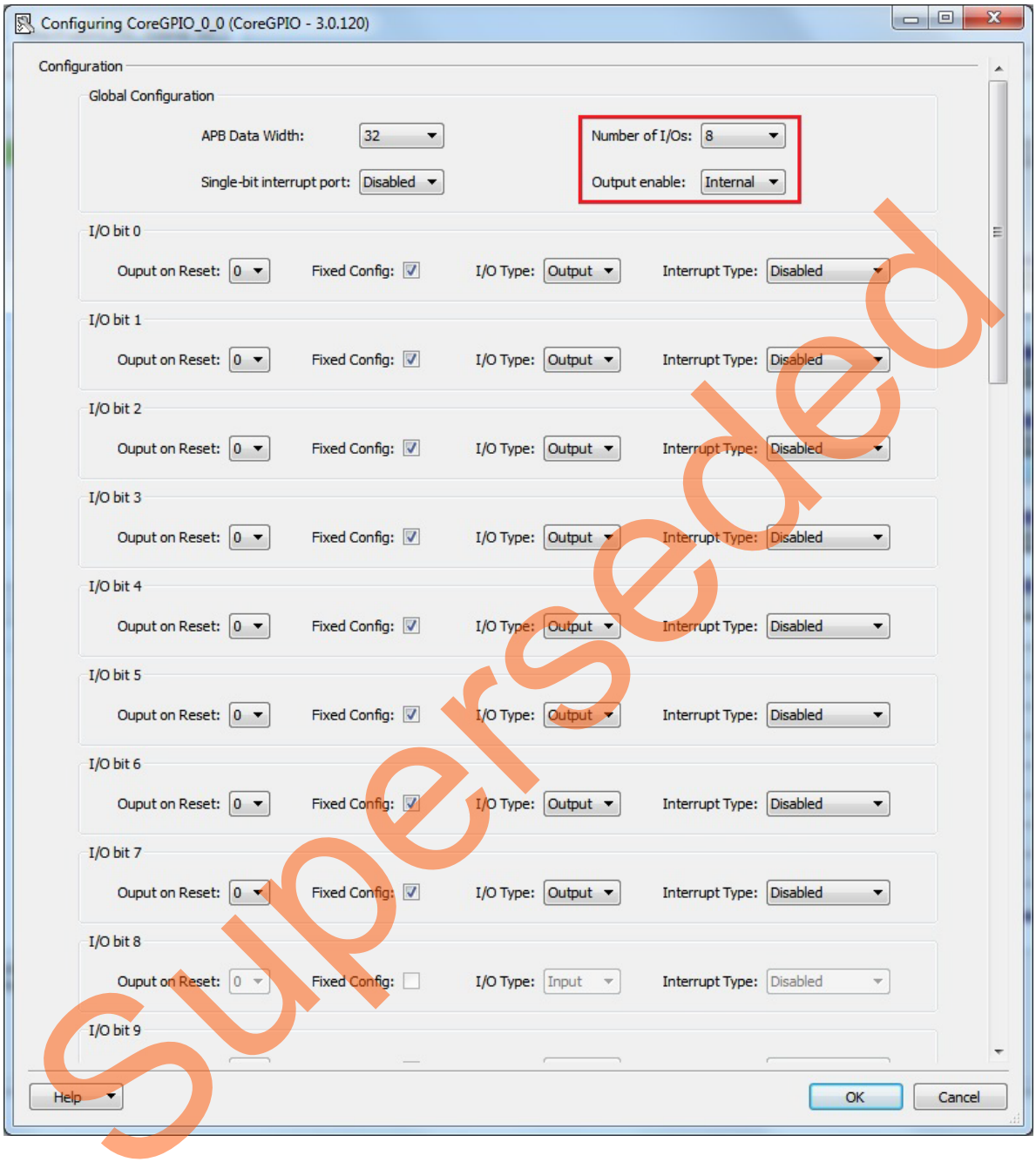

***Figure 9* • CoreGPIO Configuration**

Note: For SmartFusion2 Starter Kit Board CoreGPIO Configuration window is shown in Figure 10.

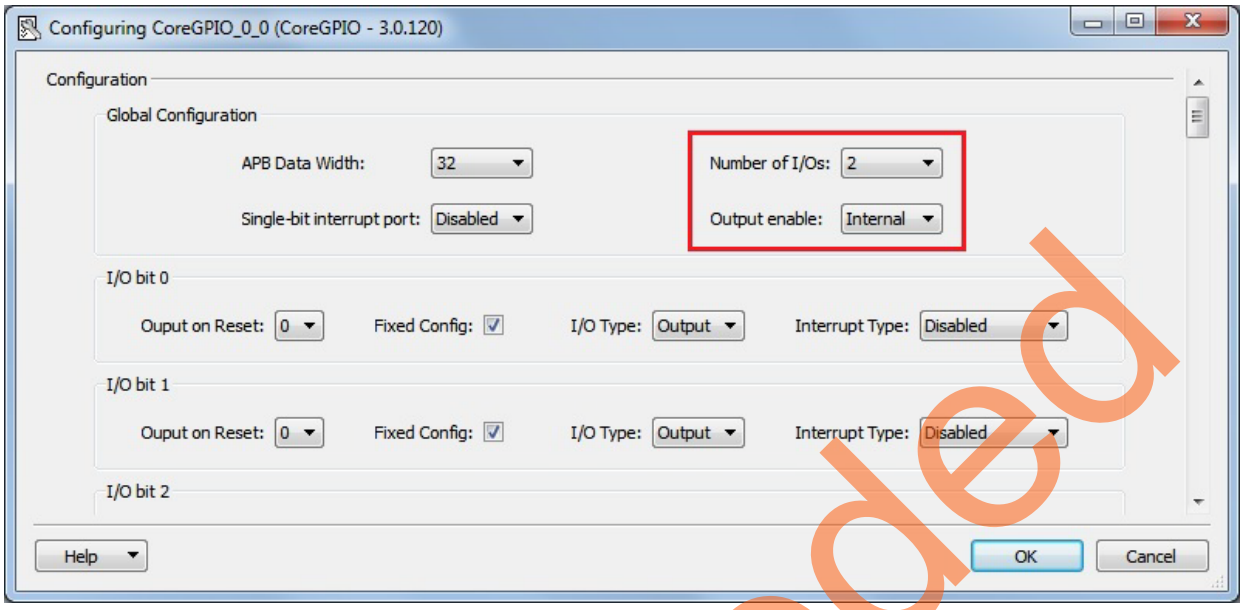

***Figure 10* • CoreGPIO Configuration**

11. Click **OK** after completion of CoreGPIO configuration.
12. Click **MMUART** Configure icon as shown in Figure 7, and use the following settings as shown in Figure 11 and keep the rest at default states:
    - Connect to: Fabric for Development Kit / IO for Starter Kit

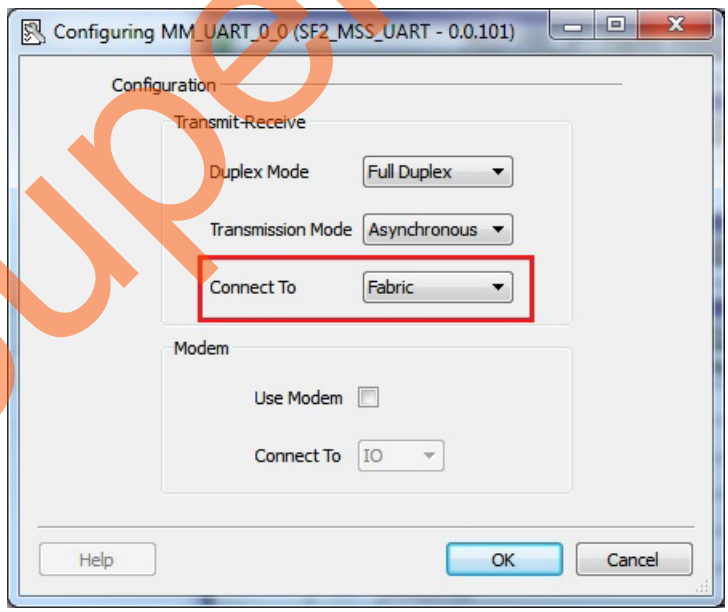

***Figure 11* • MMUART Configuration**

Note: MMUART Configuration window for SmartFusion2 Starter Kit Board is shown in Figure 12.

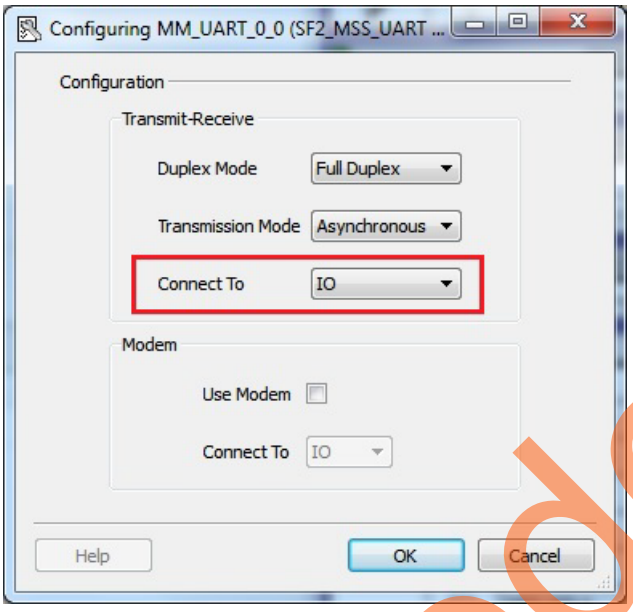

***Figure 12* • MMUART Configuration**

13. Select **Next**. **System Builder- Clock Settings** page is displayed, as shown in Figure 13. Select the Following options:
    - System Clock: Set it to On-chip 25/50 MHz RC Oscillator from the drop down list.
    - M3_CLK: 100 MHz
    - MSS APB_0/1 Clocks: 100 MHz
    - Fabric Interface Clocks: 100 MHz

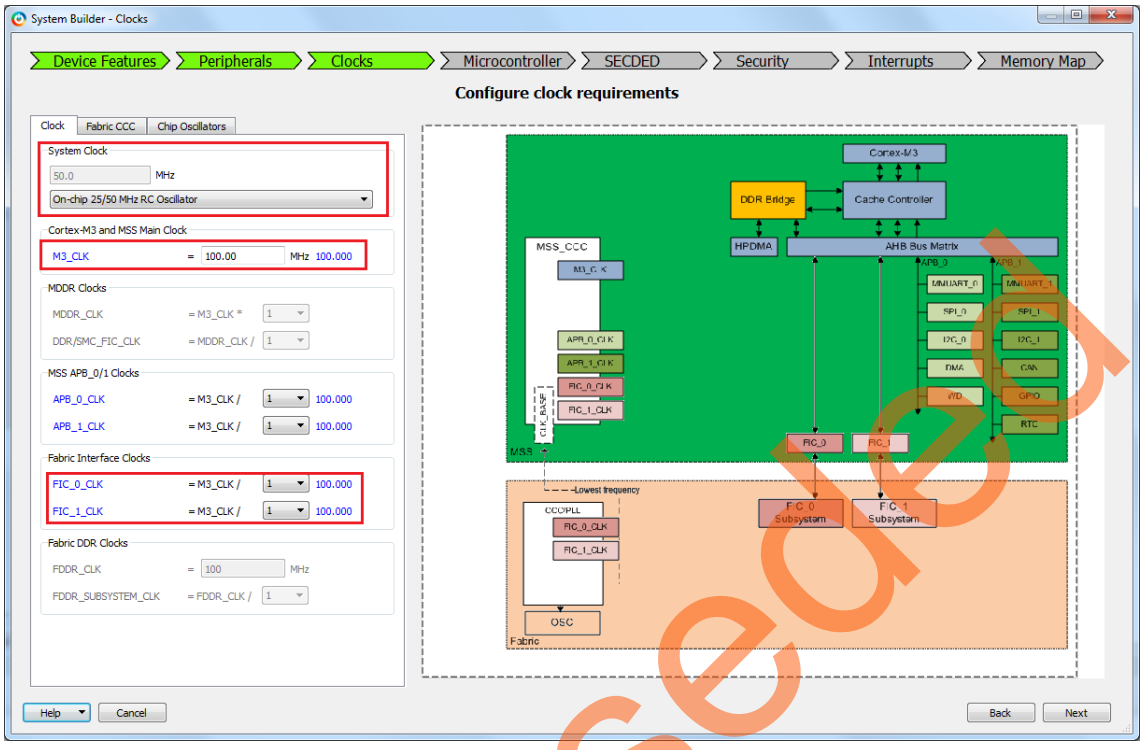

Figure 13 • **SmartFusion2 System Builder Clocks**

14. Click **Next**, the System Builder - **Microcontroller Options** page is displayed.
    - Leave all the Default Selections.
15. Click **Next**, the System Builder - **SECDED Options** page is displayed.
    - Leave all the Default Selections.
16. Click **Next**, the System Builder - **Interrupts Options** page is displayed.
    - Leave all the Default Selections.
17. Click **Next**, the System Builder - **Memory Map Options** page is displayed.
    - Leave all the Default Selections.

- Figure 14 and Figure 15 shows the address maps for AHBL and APB3 peripherals.

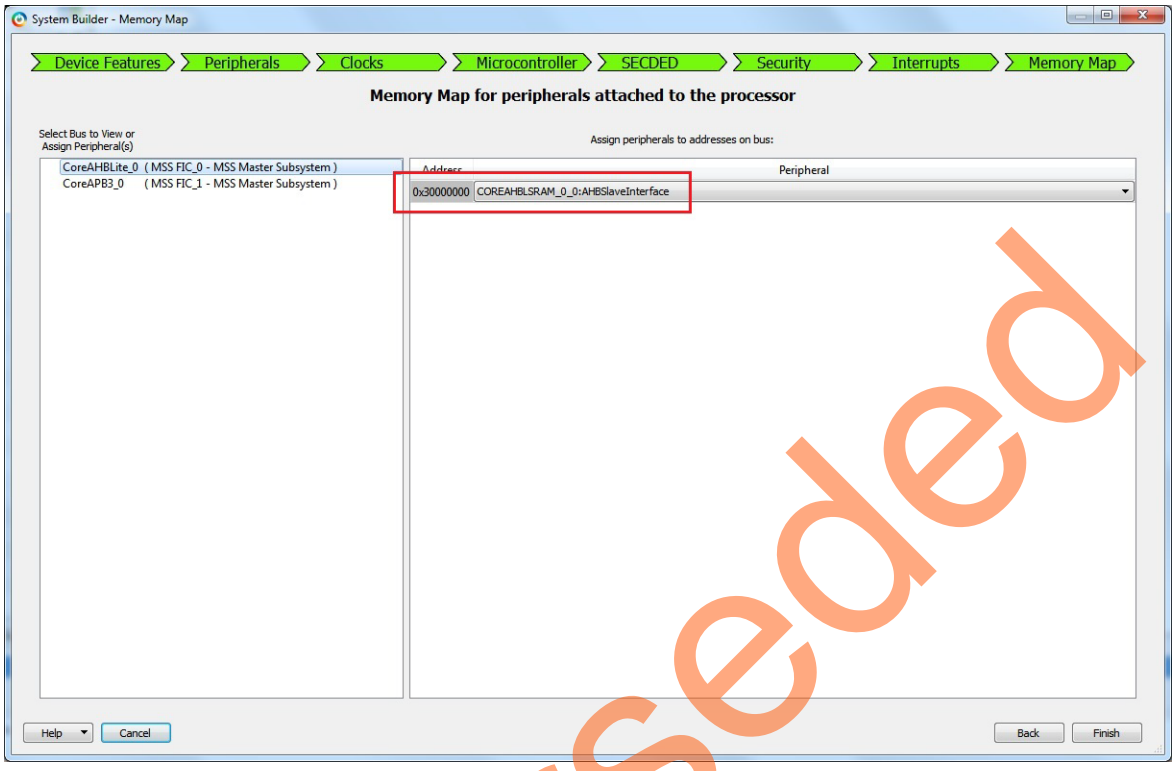

Figure 14 • **SmartFusion2 System Builder CoreAHBLite Address Map**

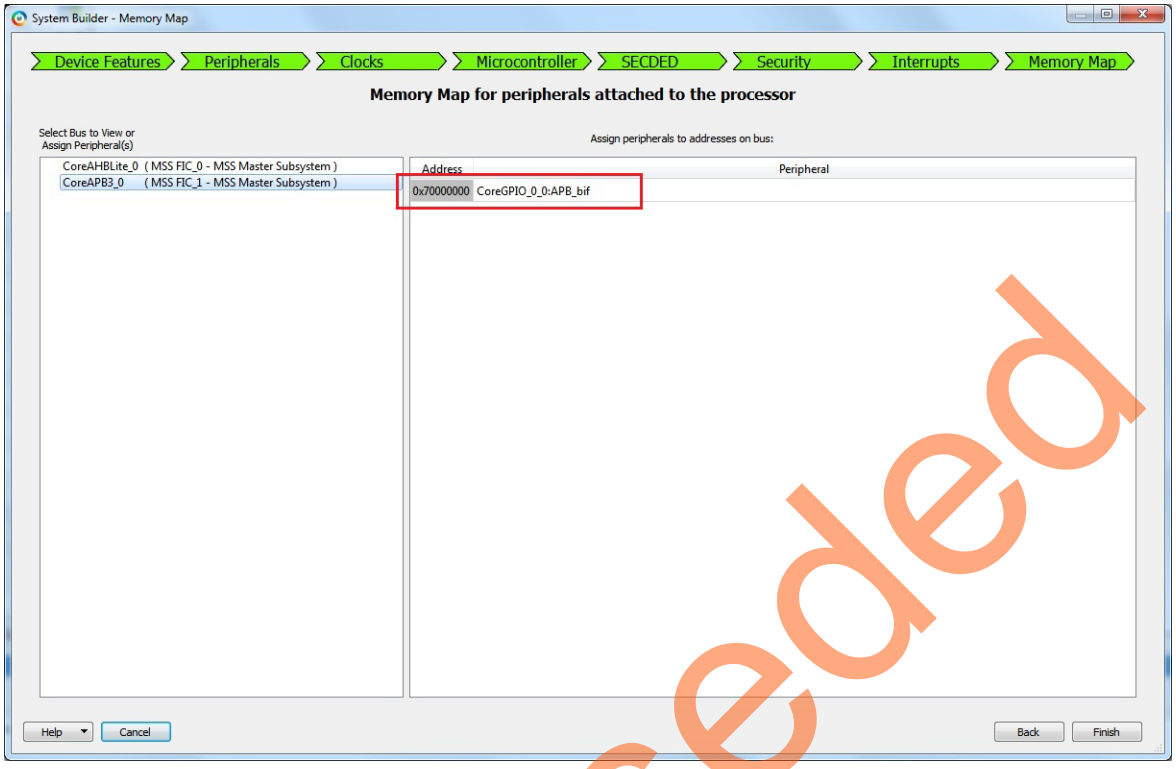

Figure 15 • **SmartFusion2 System Builder CoreAPB Address Map**

18. Click **Finish**.

The System Builder will generate the system based on the selected options.

The System Builder block is created and added to Libero SoC project, as shown in Figure 16.

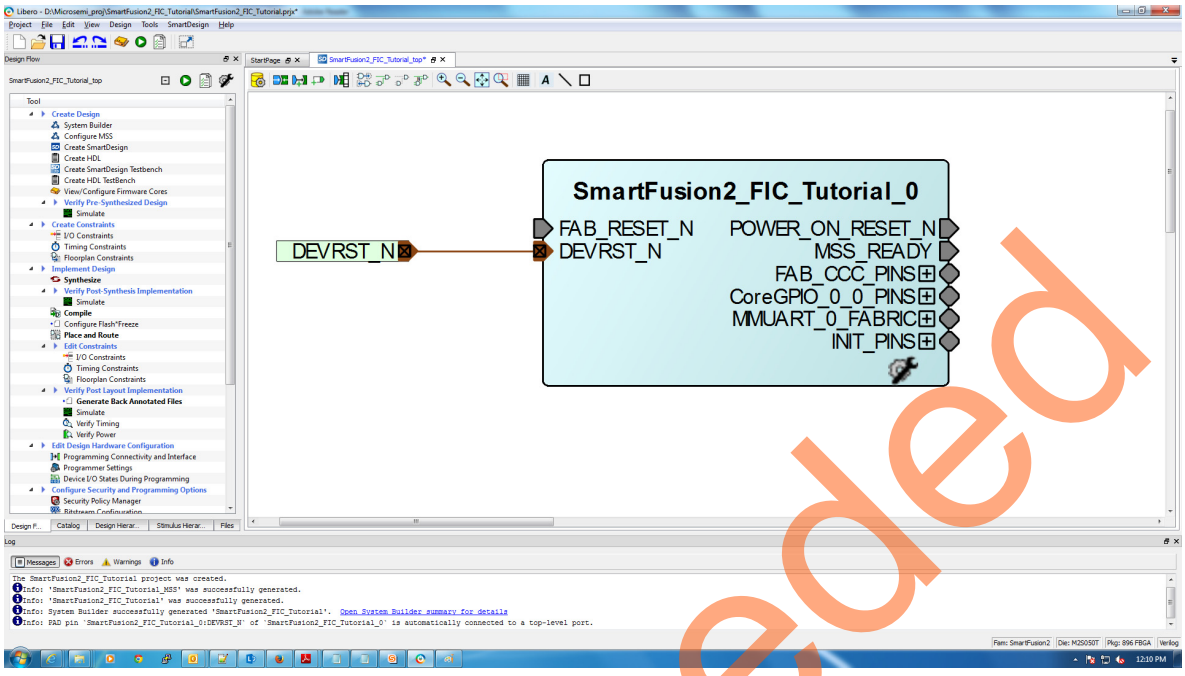

***Figure 16* • SmartFusion2 System Builder Component**

The CoreResetP will automatically be instantiated and connected by the System Builder. How these blocks are connected can be seen by opening the System Builder component in the Smart Design canvas.

19. Make the connections for the pins as follows:
    - Right-click **FAB_RESET_N** and select **Tie High**.
    - To select POWER_ON_RESET_N and MSS_READY pins, hold CTRL key, select pins, right-click and select **Mark Unused**.
    - Expand **INIT_PINS**, right-click **INIT_DONE** and select **Mark Unused**.
    - Promote the **MMUART_0_FABRIC** to top by right-clicking and selecting **Promote to Top Level**.
    - Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL0** and **FAB_CCC_GL1** and select **Mark Unused**.
    - Expand **CoreGPIO_0_0_PINS**,
      – Mark the **INT[7:0] PINS** as unused by right-clicking and selecting **Mark Unused**.
      – Tie the **GPIO_IN[7:0]** to high by right-clicking and selecting **Tie High**.
      – Promote the **GPIO_OUT[7:0]** to top by right clicking and selecting **Promote to Top Level**.

After making all necessary connection the system builder block as shown in Figure 17.

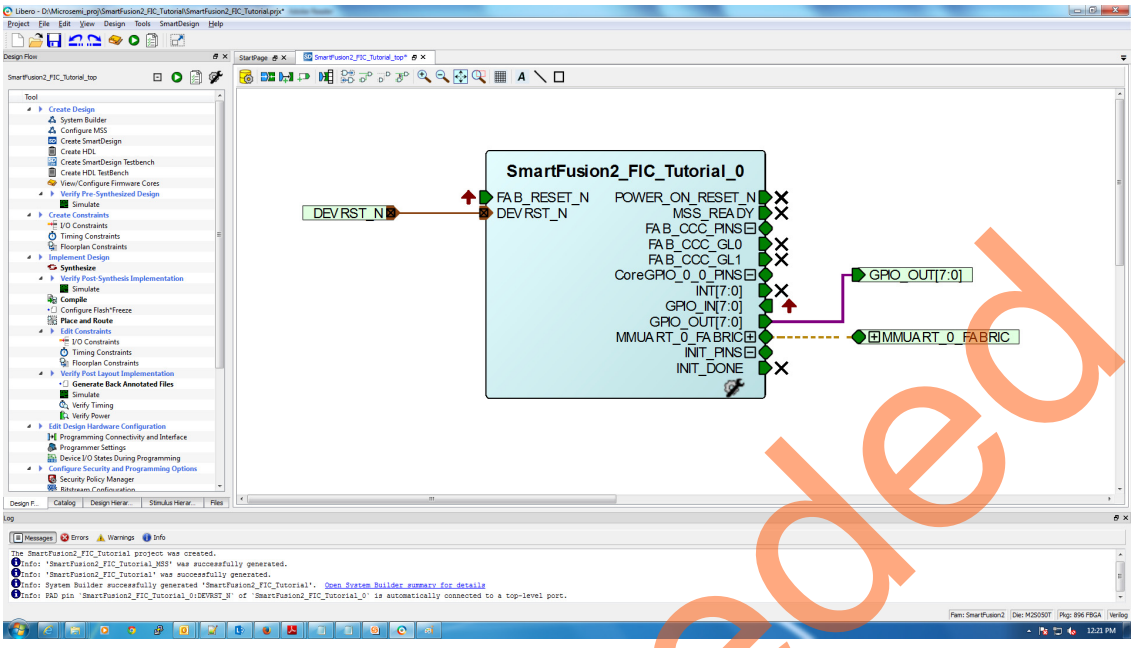

Figure 17 • **SmartFusion2 System Builder Block**

Note: The System Builder Block for SmartFusion2 Starter Kit is shown in Figure 18.

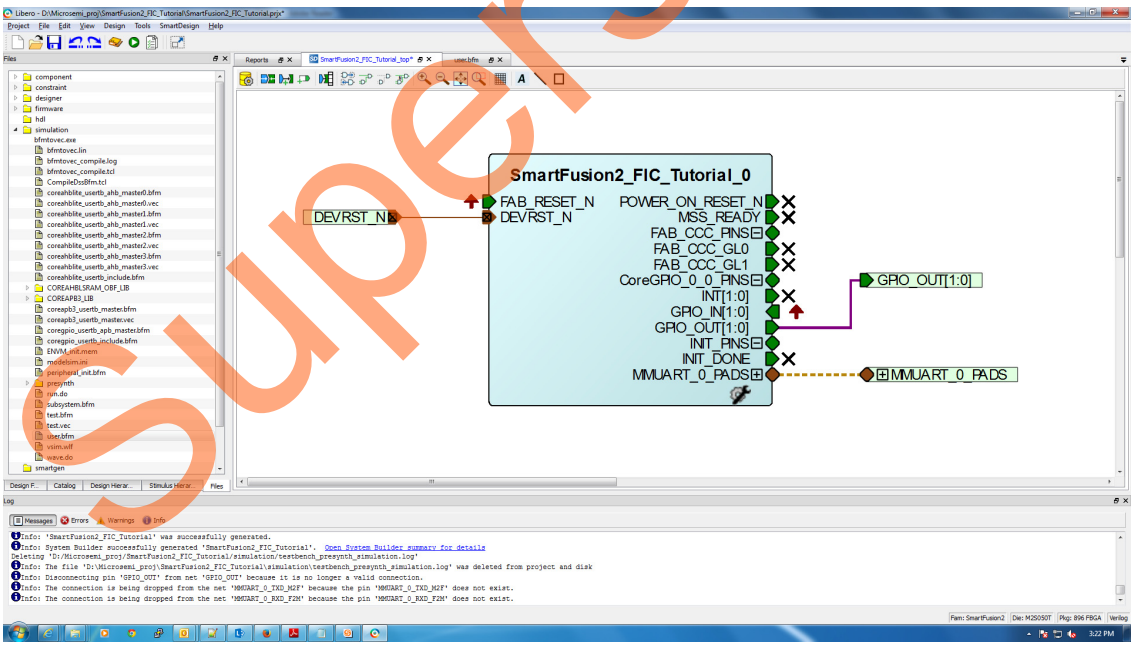

Figure 18 • **SmartFusion2 System Builder Block**

20. Open **Firmware Configuration** window by clicking **Design> Configure Firmware** and select SmartFusion2_CMSIS_0, CoreGPIO_Driver_0, SmartFusion2_MSS_MMUART_Driver_0 and uncheck other drivers as shown in Figure 19.

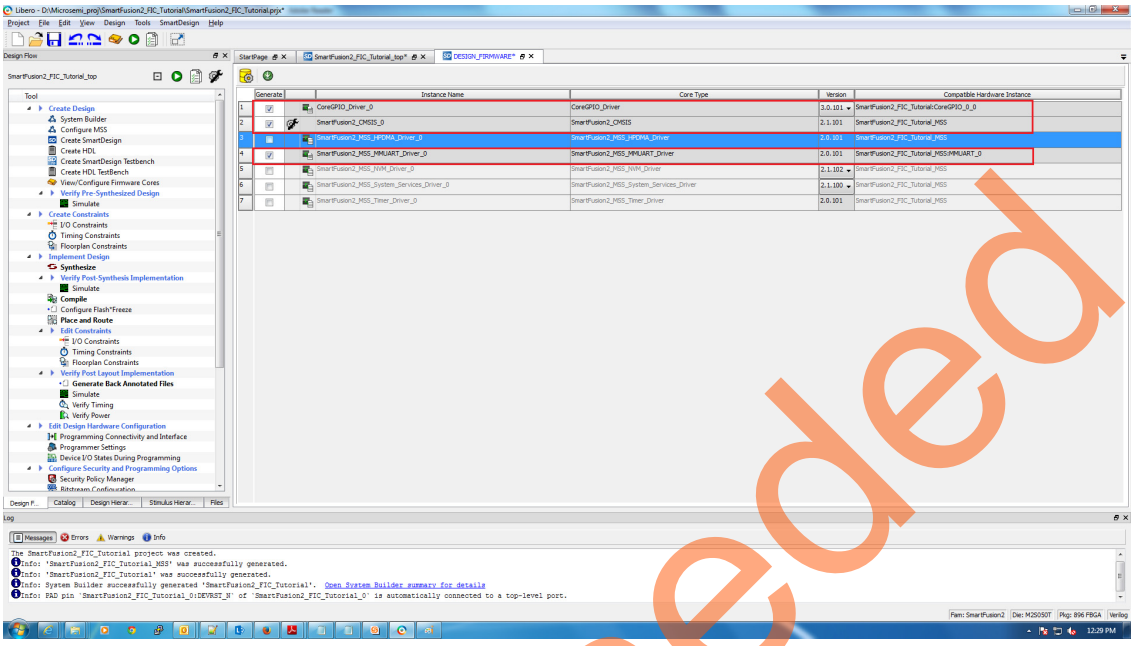

Figure 19 • **Configure Firmware**

21. Open **SmartFusion2_FIC_Tutorial_top** tab and click **Generate Component** icon on the SmartDesign toolbar.

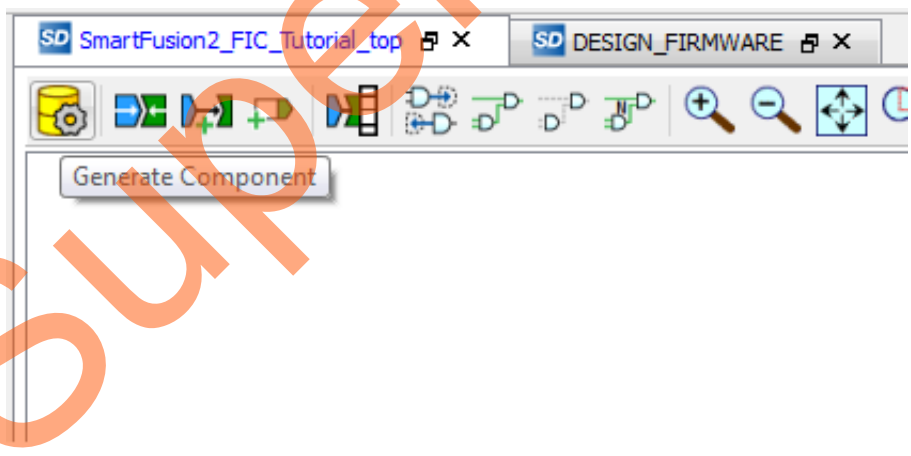

Figure 20 • **Generate Component**

22. You can also right-click on the canvas and select **Generate Component**.

After successful generation of the system, The message **'SmartFusion2_FIC_Tutorial_top' was successfully generated** is displayed in the Libero SoC log window if the design was generated without any errors. Libero also generates the SoftConsole software project. The log window is displayed as shown in Figure 21.

Figure 21 • **Log Window**

Step 2: Modifying User BFM Script for Simulation

You can verify the design by using the BFM master or slave model and a BFM script to drive the AHBL/APB input of the DUT. This setup allows the BFM to write or read to the AHBL/APB register set and to verify that the DUT is behaving as expected.

This step explains adding BFM commands to the user.bfm file to perform design simulation. For more information on BFM commands refer to the *CoreAMBA BFM User Guide*. The user.bfm file is created by Libero SoC Design software and is available in the simulation folder of the project files.

Note: Download the project files. Refer to the "Project Files" section on page 6.

1. For SmartFusion2 Development Kit board, right-click the simulation under project files and select import files to import the `user.bfm` file which is located in downloaded design files (*\SmartFusion2_FIC_TutorialSource\For_SF2_Dev_Kit_Board\user.bfm*) as shown in Figure 22 or select **Files > Import > Others** to import the `user.bfm` file.
2. Click **Yes to all** to replace the existing `user.bfm` file.

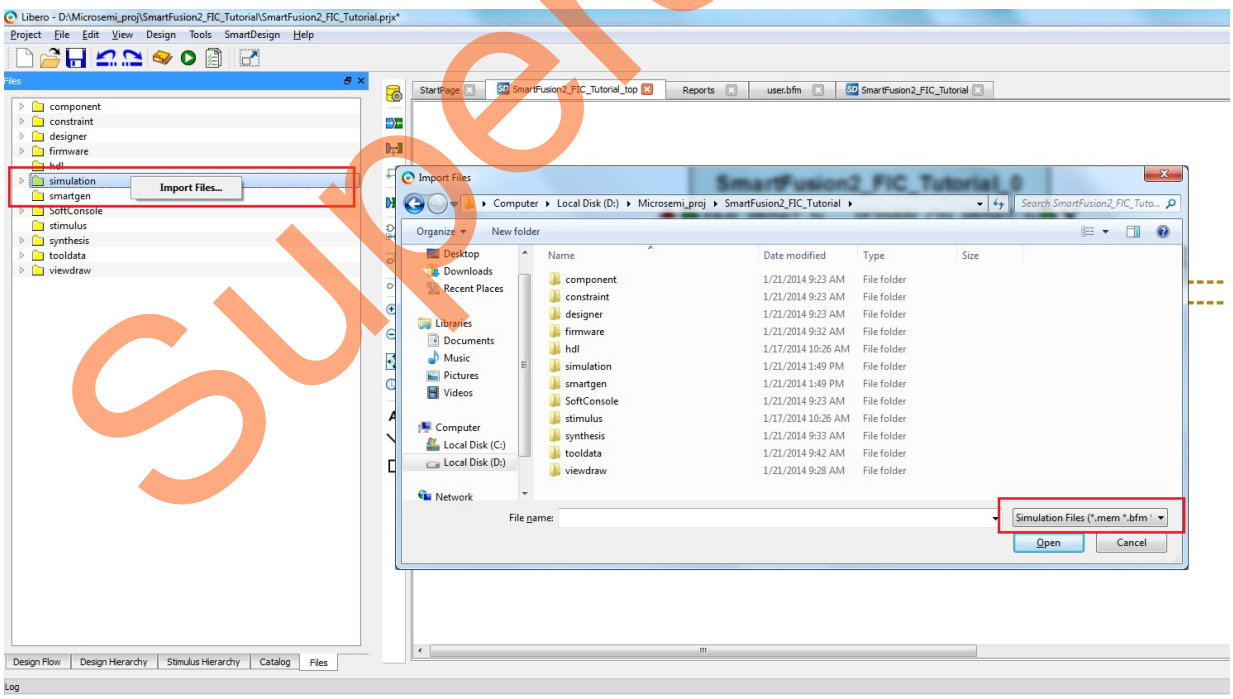

Figure 22 • **Import bfm file**

Note: For SmartFusion2 Starter Kit, import the user.bfm to simulation files from design files. (*\SmartFusion2_FIC_Tutorial_DF\Source\For_SF2_Starter_Kit_Board\user.bfm*)

3. After importing, double-click the user.bfm file under simulation folder. This opens the user.bfm file as a new tab in the project window, as shown in Figure 23.

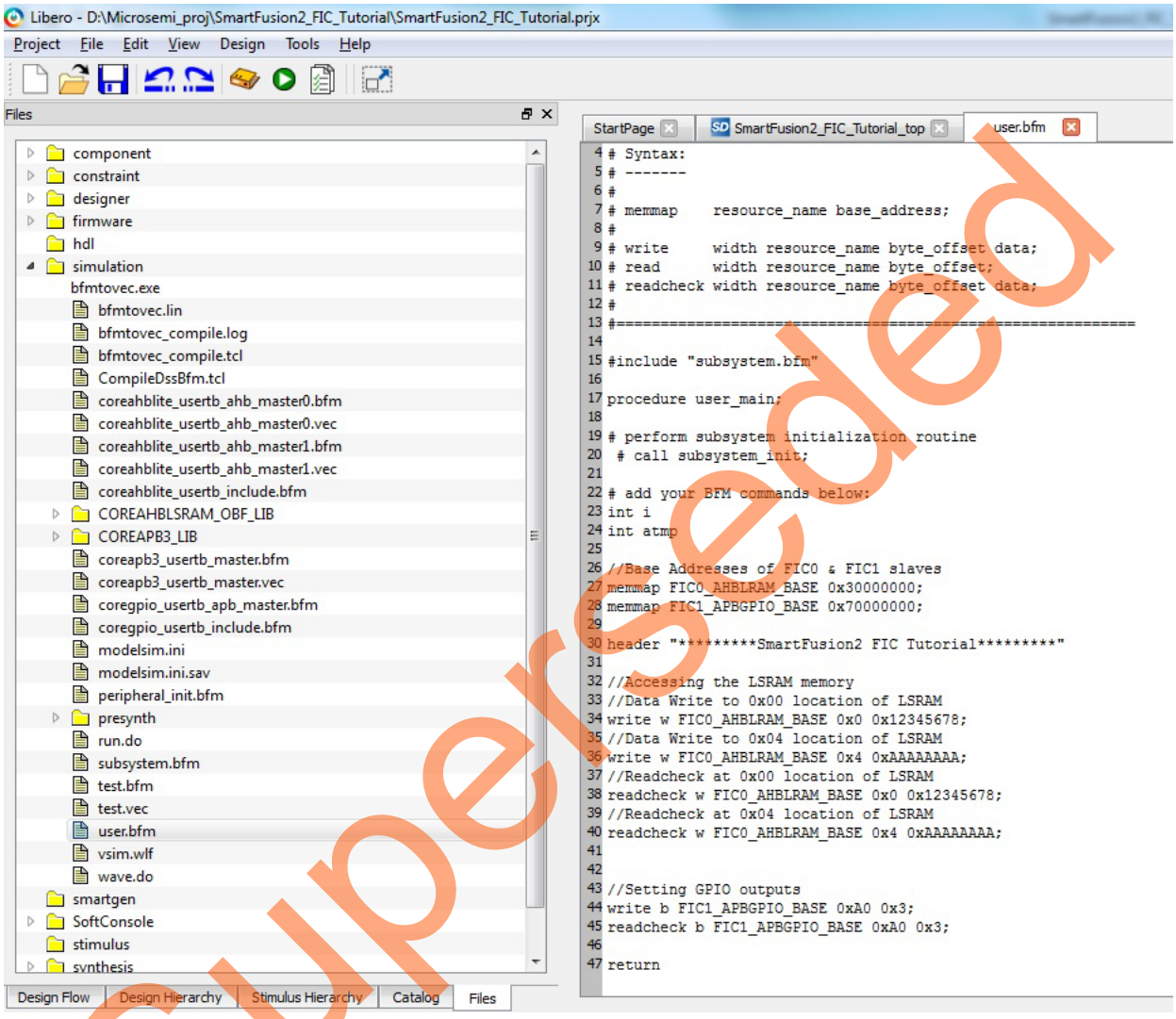

Figure 23 • user.bfm file

Step 3: Simulating Design Using BFM Models

This section describes how to use the SmartDesign testbench and BFM script file to simulate the design.

1. Add the `wave.do` file to the SmartFusion2_FIC_Tutorial design simulation folder by clicking **File > Import > Others**.
2. Browse to the wave.do file location in the design files folder: SmartFusion2_FIC_Tutorial\Source\for_SF2_Dev_Kit_Board. Figure 24 shows the wave.do file under simulation folder in the Files window.

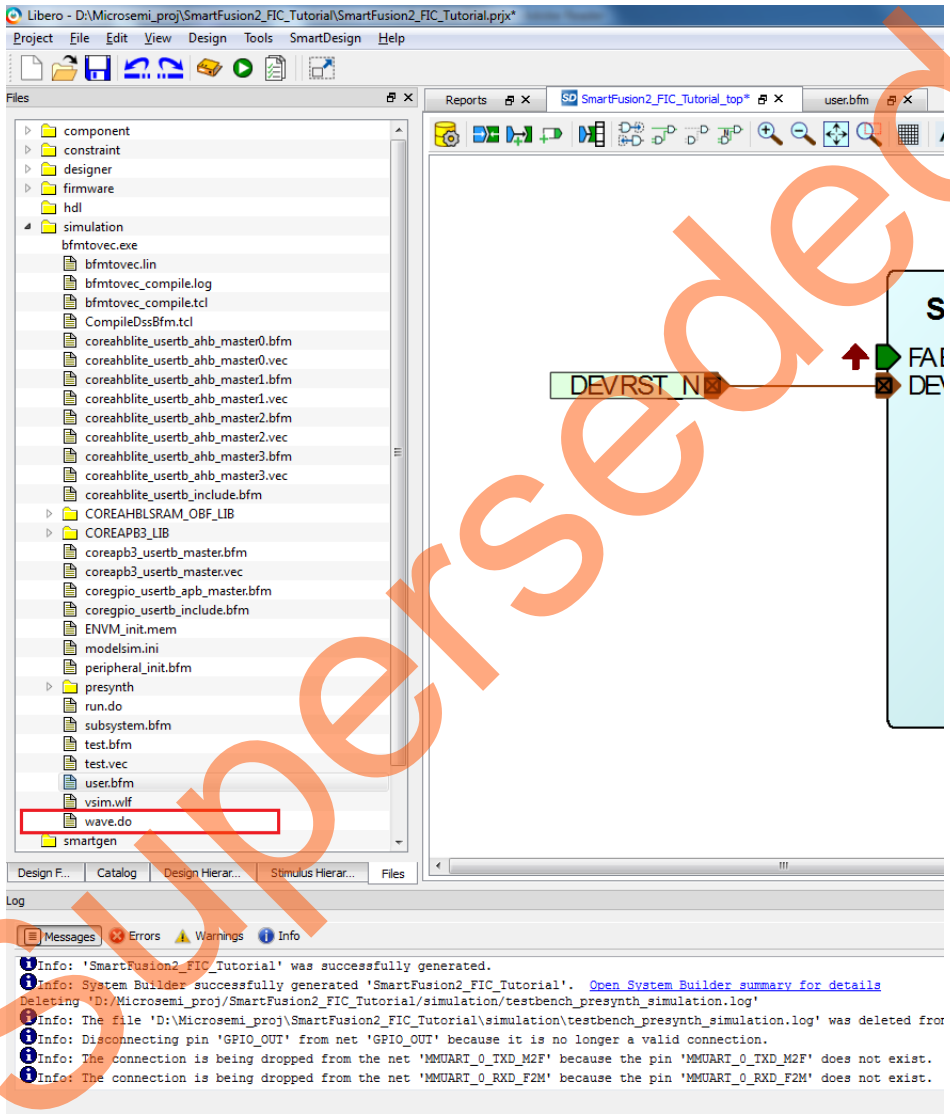

Figure 24 • **wave-do File**

Note: For SmartFusion2 Starter Kit board, browse to the `wave.do` file located in the downloaded design files folder: *SmartFusion2_FIC_Tutorial\Source\for_SF2_Starter_Kit_Board*.

3. Set up the simulation environment as follows:

   Select **Project > Project Settings**. On the Project Settings window, under **Simulation Options**, select **DO File** to change the simulation run time. Enter **50us** in the **Simulation Runtime** field, as shown in Figure 25.

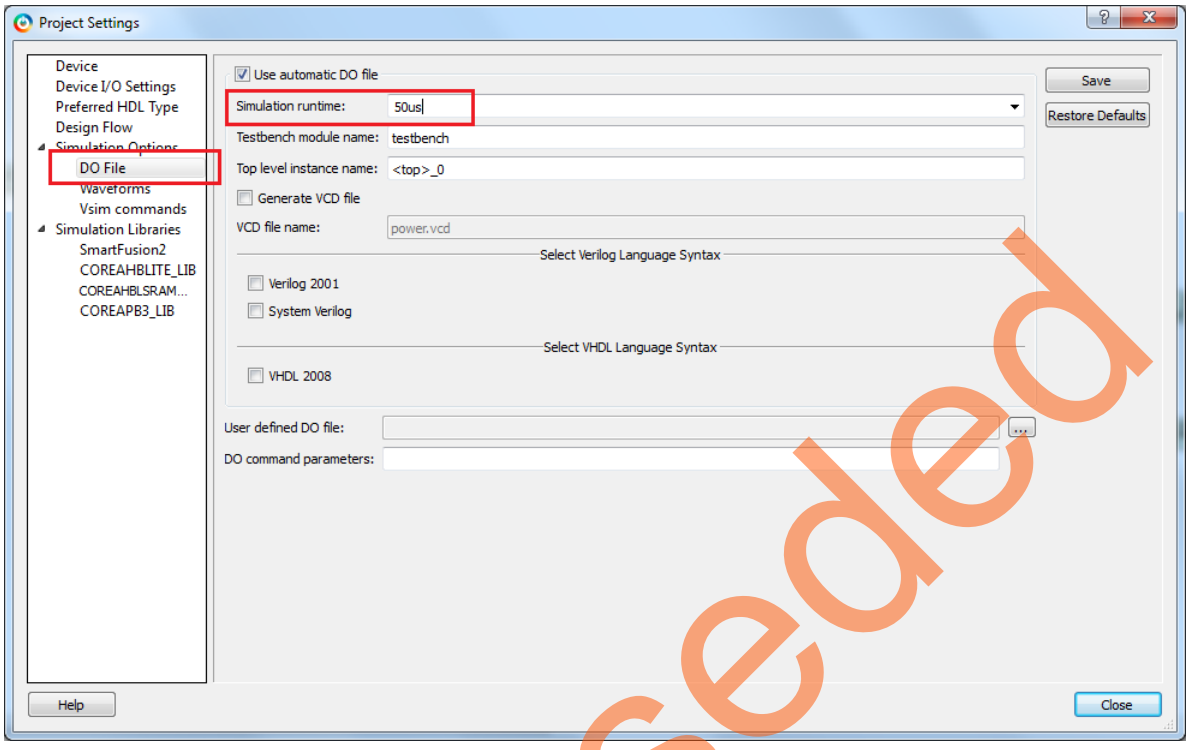

***Figure 25* • Project Settings – Do File**

Save the **Do File** configuration, this can be done by clicking the **Save**.

4. Select **Waveforms** under **Simulation Options** as shown in Figure 26 on page 28:
   a. Select **Include Do** file.
   b. Select **Log all signals in the design** check box.
   c. Click **Close** to close the Project settings dialog box.
   d. Select **Save** when prompted to save the changes.

Note: You can also add ports or signals of interest in the ModelSim software.

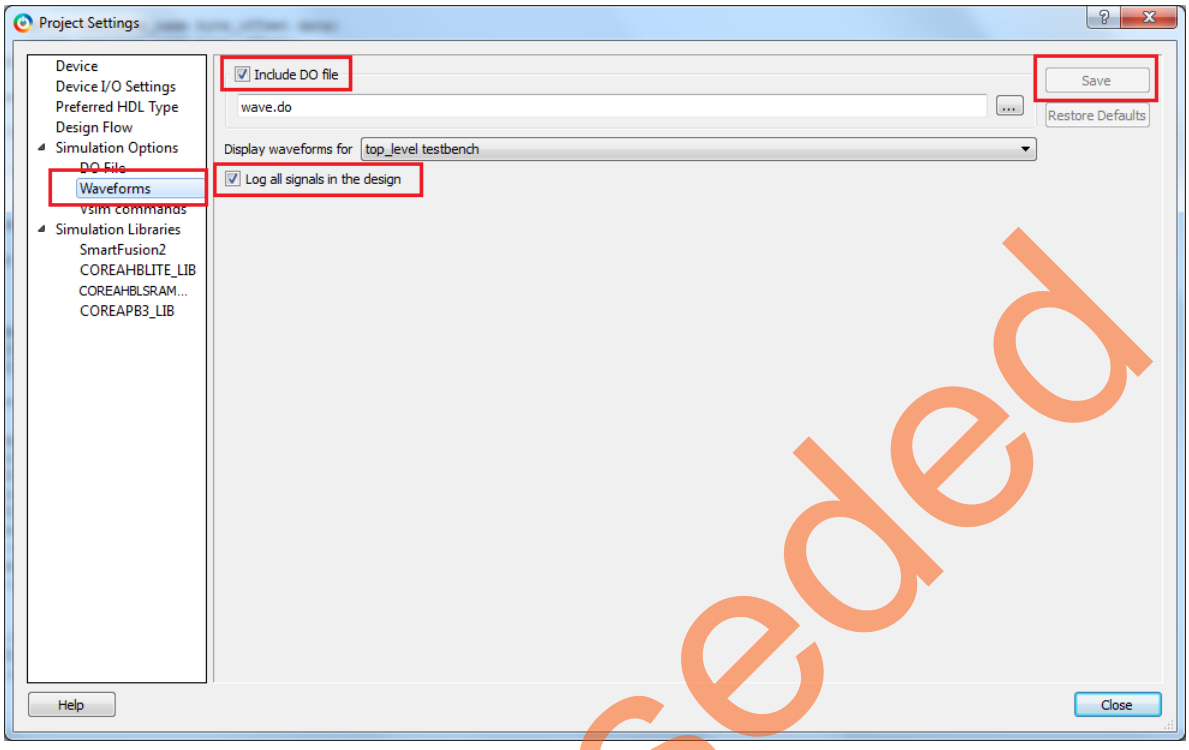

***Figure 26* • Project Settings – Waveforms**

5. Select the **Design Flow** tab in the project window.

6. Expand the **Verify Pre-Synthesized Design**, as shown in Figure 27. Double-click **Simulate** to invoke ModelSim. ModelSim will be invoked and load the design. Alternatively, you can right-click the **Simulate** and select **Open Interactively**.

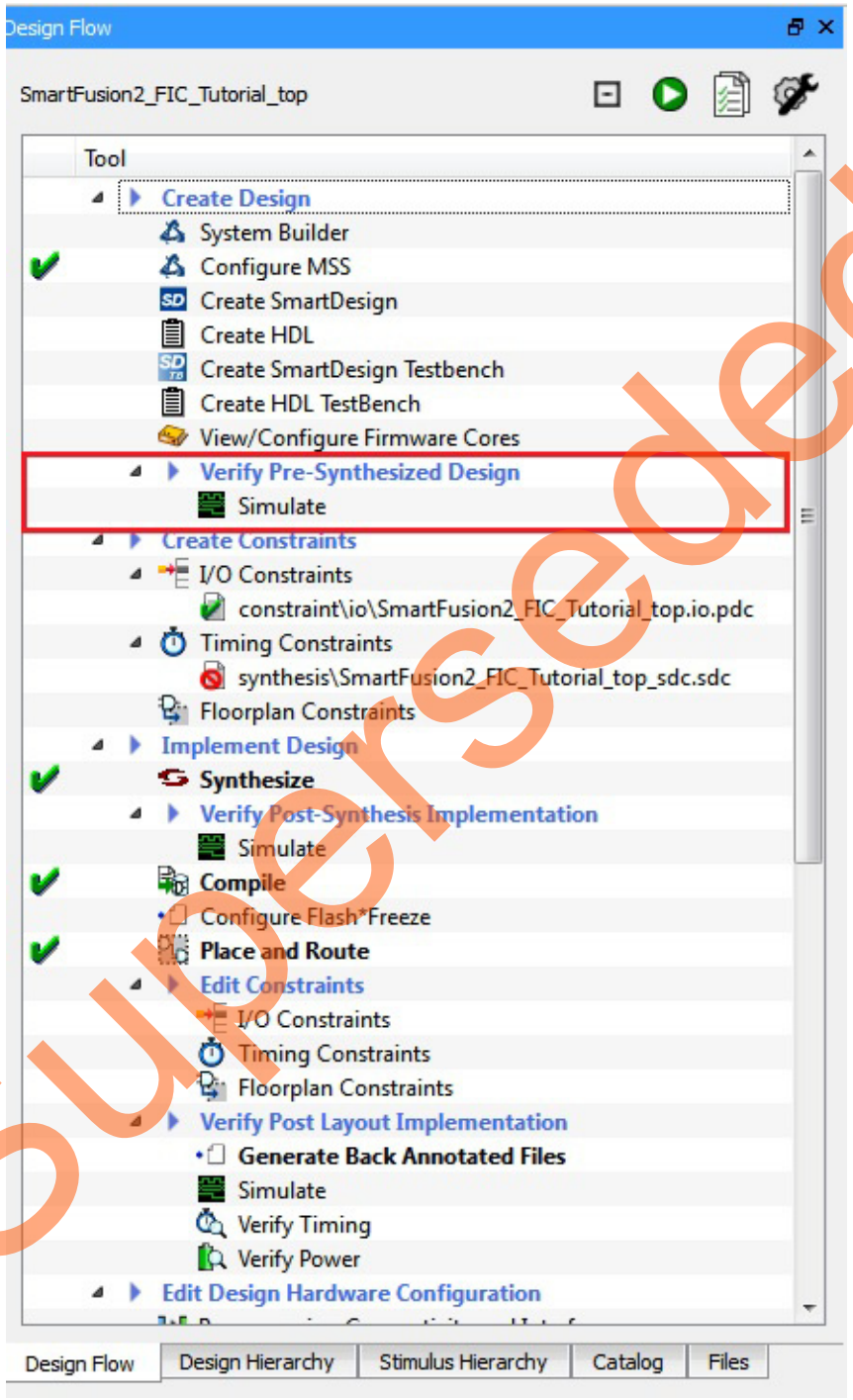

***Figure 27* • Design Flow – Verify Pre-Synthesized Design**

7. Maximize the **ModelSim Transcript** window to see the BFM commands execution. Make sure that there are no errors. Figure 28 shows the ModelSim Transcript window.

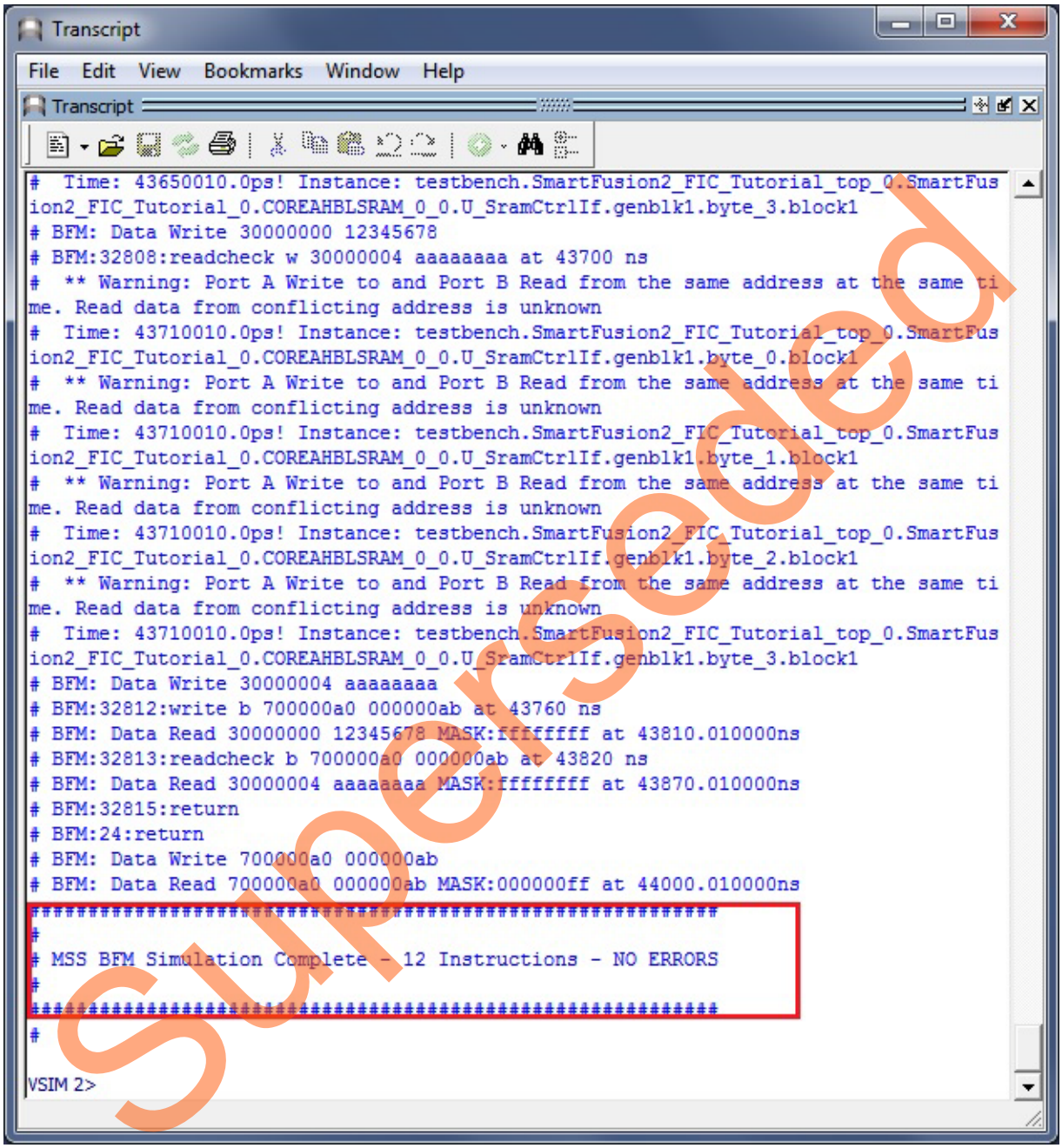

***Figure 28* • ModelSim Transcript Window – BFM Commands**

8. After successful BFM simulation, observe the ModelSim waveform window for the read and write bus transactions to the fabric peripherals, as shown in Figure 29. Notice the result of GPIO configuration BFM commands in GPIO states.

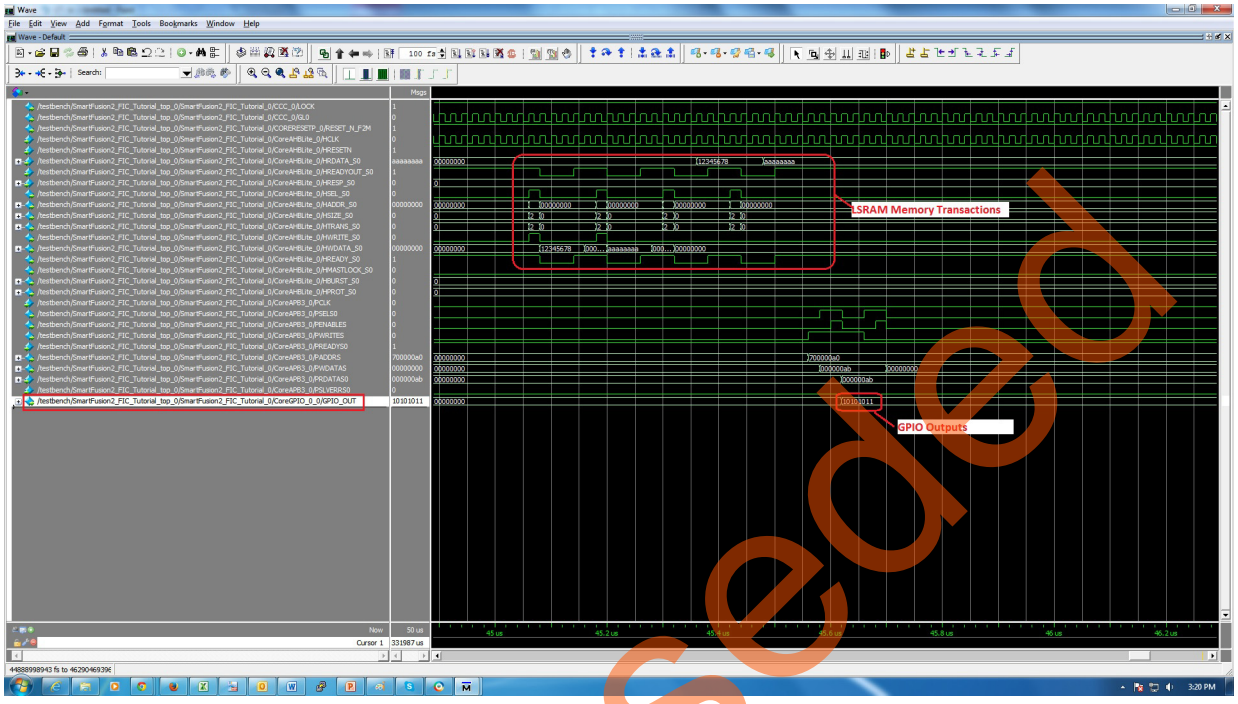

Figure 29 • **Design Simulation Waveforms**

Quit the ModelSim simulator by selecting **File > Quit**.

Step 4: Generating Programming File

1. Double-click **I/O Constraints** in the **Design Flow** window as shown in Figure 30. The **I/O Editor** window is displayed after completing Synthesize and Compile.

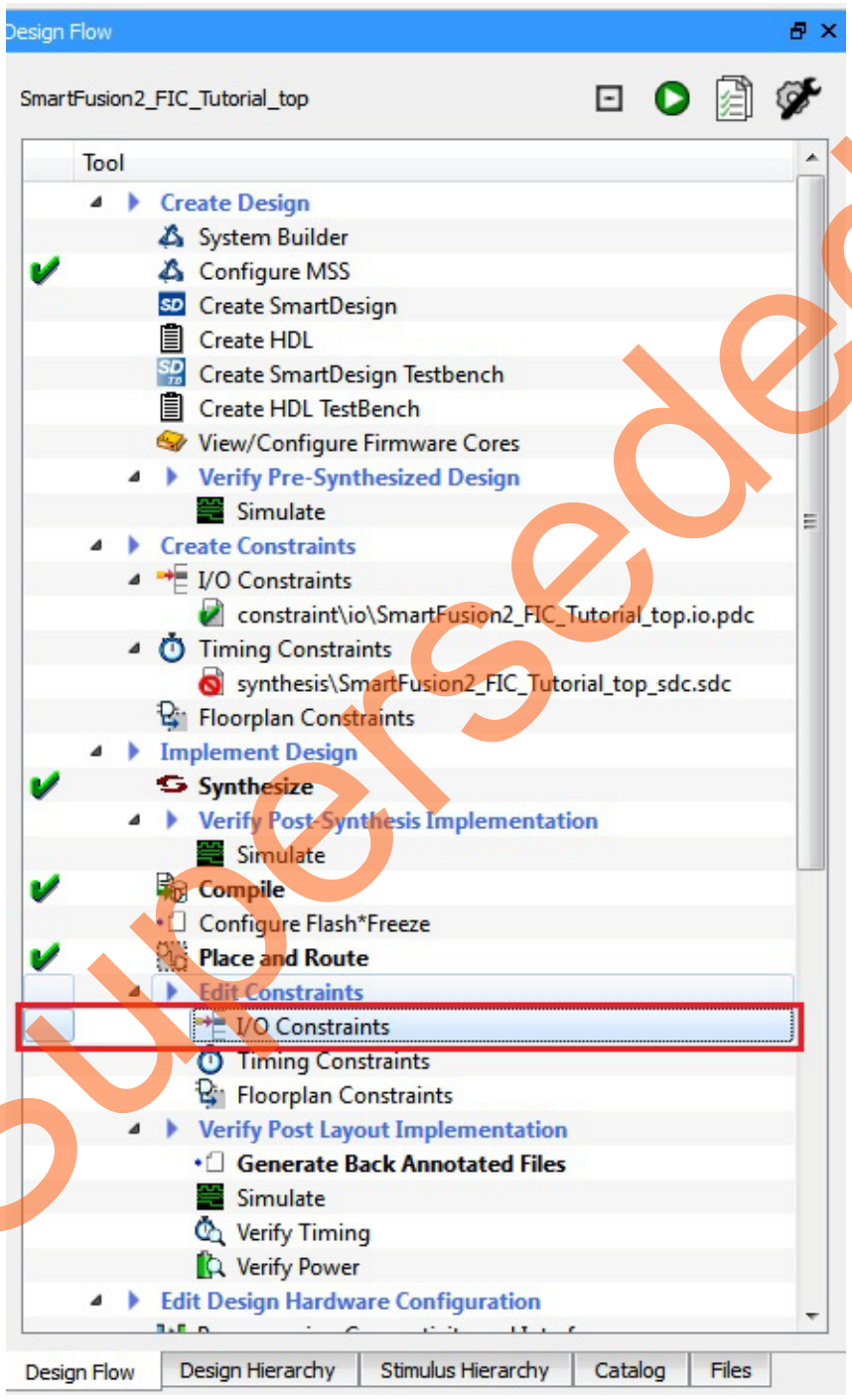

***Figure 30* • I/O constraints**

2. The **I/O Editor** is displayed. Make the pin assignments as shown in Table 2.

Table 2 • **Port to Pin Mapping**

Pin Name	Pin Number
GPIO_OUT[0]	A18
GPIO_OUT[1]	B18
GPIO_OUT[2]	D18
GPIO_OUT[3]	E18
GPIO_OUT[4]	A20
GPIO_OUT[5]	D20
GPIO_OUT[6]	E20
GPIO_OUT[7]	B20
MMUART_0_RXD_F2M	R29
MMUART_0_TXD_F2M	R24

After the pins have been assigned, the I/O Editor is displayed as shown in Figure 31.

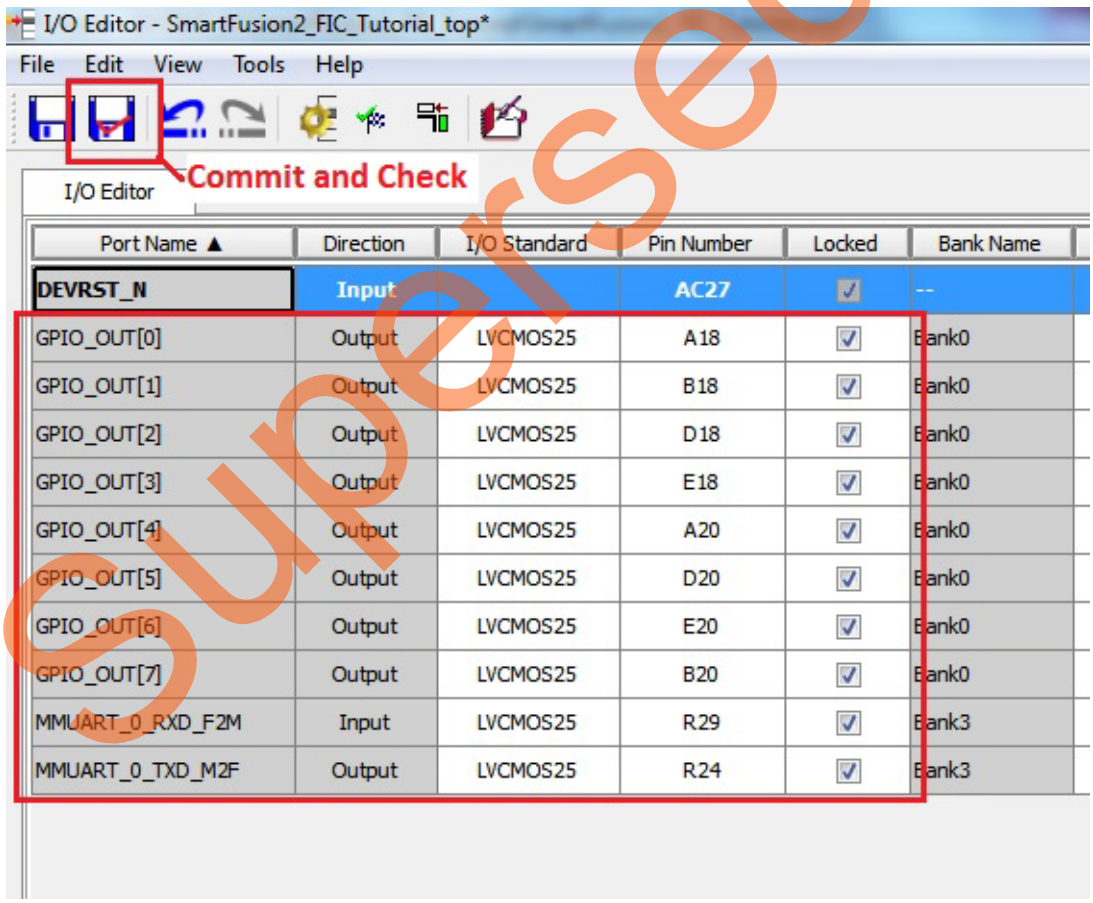

Figure 31 • **I/O Editor**

Note: Pin assignments for the SmartFusion2 Starter Kit Board is shown in Table 3.

Table 3 • **Port to Pin Mapping**

Pin Name	Pin Number
GPIO_OUT[0]	G30
GPIO_OUT[1]	AC1

These pin assignments are for connecting below on the SmartFusion2 Development Kit:

- GPIO_OUT[0] to GPIO_OUT[7] for LEDs
- MMUART to USB

3. After Updating the I/O editor, Click **Commit and Check**.
4. Close I/O editor.
5. Click **Generating Programming Data** as shown in Figure 32 to complete place and route, verify timing, and generating the programming file.

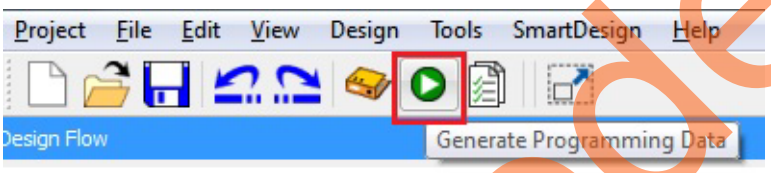

Figure 32 • **Generate Programming Data**

Step 5: Programming the SmartFusion2 Board Using Flash Pro

Before proceed with programming device, ensure that FlashPro4 programmer is properly connected to the Flash Pro Header of board. Use the following details to ensure the correct jumper settings. Refer to the *SmartFusion2 Starter Kit User Guide* and the *SmartFusion2 Development Kit User Guide* for additional information.

Jumper Settings for SmartFusion2 Development Kit Board

Connect the jumpers on the SmartFusion2 SoC FPGA Development Kit, as shown in Table 4. While making the jumper connections the power supply switch SW7 on the board should be in OFF position.

Table 4 • **Jumper settings for Development Kit Board**

Jumper	Pin (from)	Pin (to)
J70, J93, J94, J117, J123, J142, J157, J160, J167, J225, J226, J227	1 (default)	2
J2	1(default)	3
J23	2 (default)	3
For UART Communication		
J129, J133	2	3

Jumper Settings for SmartFusion2 Starter Kit Board

Connect the jumpers on the SmartFusion2 SoC FPGA Starter Kit, as shown in Table 5.

Table 5 • **Jumper settings for Development Kit Board**

Designation	Name	Settings	Description
JP1	VCC3	1-2 Closed	The +3.3 V voltage from the output of the U2 LDO regulator is applied to the SOM and to the SOM-BSB-EXT.
		3-4 open	The +3.3 V voltages from the output of the U2 LDO regulator is not applied to the D1 double diode ORing scheme.
JP2	JTAG Mode Selection	1-2 Open	The SmartFusion2 JTAG controller is in the FPGA programming mode.
		3-4 Closed	The settings of jumpers 3-4 do not affect M2S-SOM.
JP3	VCC5	1-3 open 2-4 closed	The +3.3 V LDO regulator is powered from the +5 V USB power through the P1 mini USB connector.

Programming the Device

Double-click the Run Program Action under Program Design in the Design Flow window as shown in Figure 33 to program the SmartFusion2 SoC device.

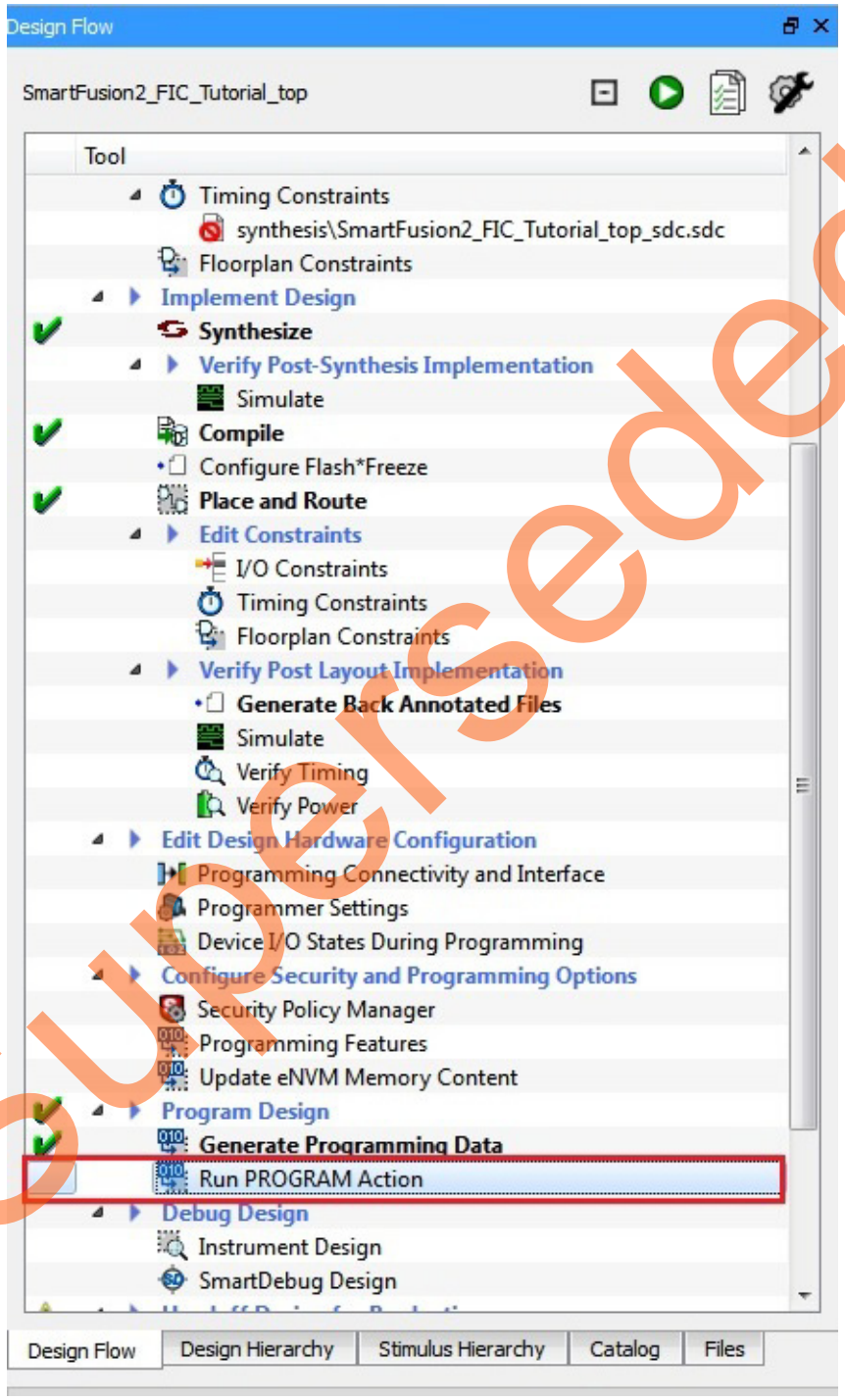

***Figure 33* • Programming the Device**

Figure 34 shows the board setup for running the application design on the SmartFusion2 Development Kit board.

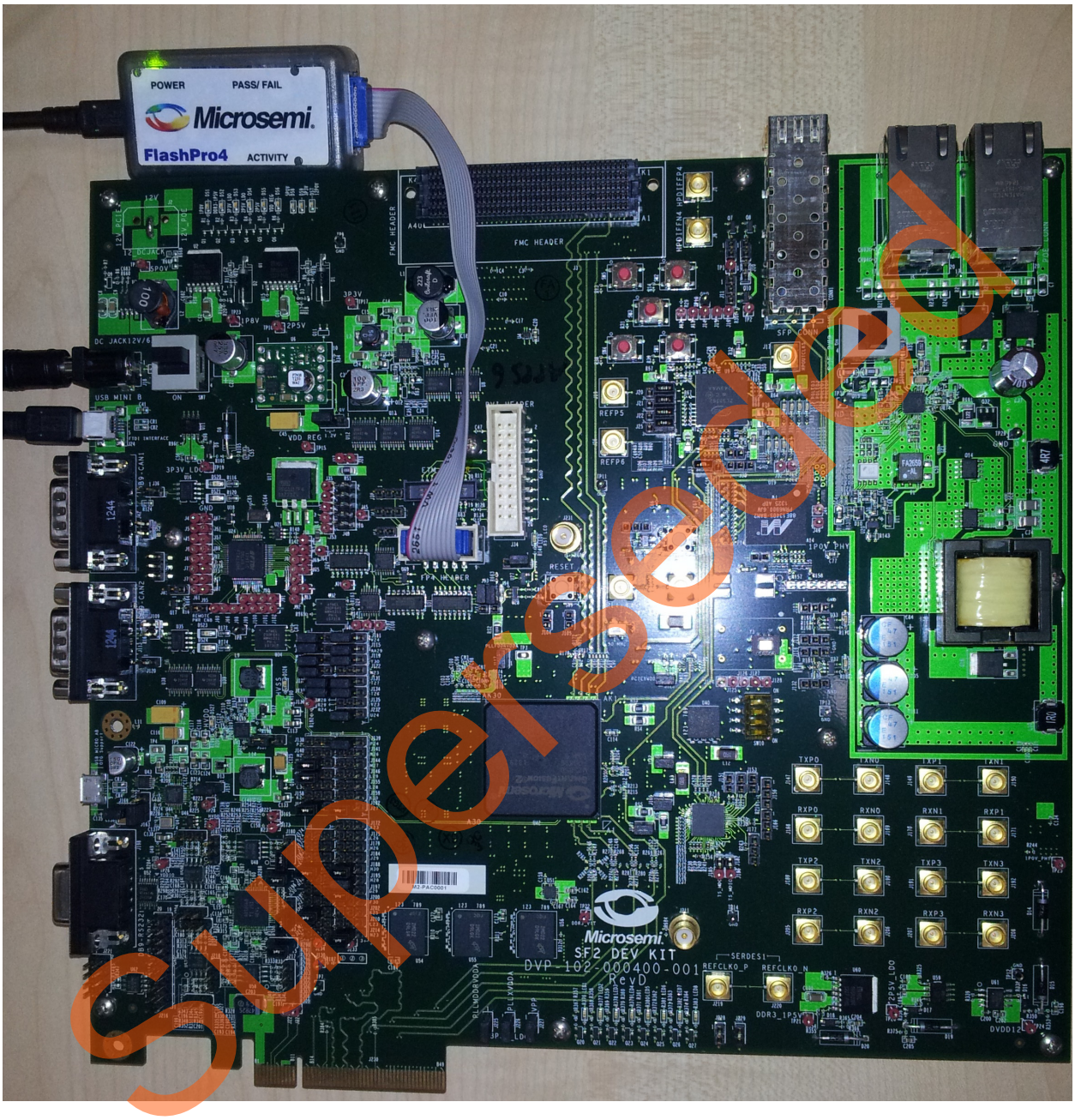

***Figure 34* • SmartFusion2 SoC FPGA Development Kit Setup**

Figure 35 shows the board setup for running the application design on the SmartFusion2 Starter Kit board.

Figure 35 • **SmartFusion2 SoC FPGA Starter Kit Setup**

Note: Do not interrupt the programming sequence; it may damage the device or the programmer. If you face any problems, contact Microsemi Tech Support at *soc_tech@microsemi.com*.

Step 6: Building the Software Application through SoftConsole

1. From the Libero SoC open the SoftConsole Project by double-click on **Write Application Code** under **Develop Firmware** in **Design Flow** window as shown in Figure 36.

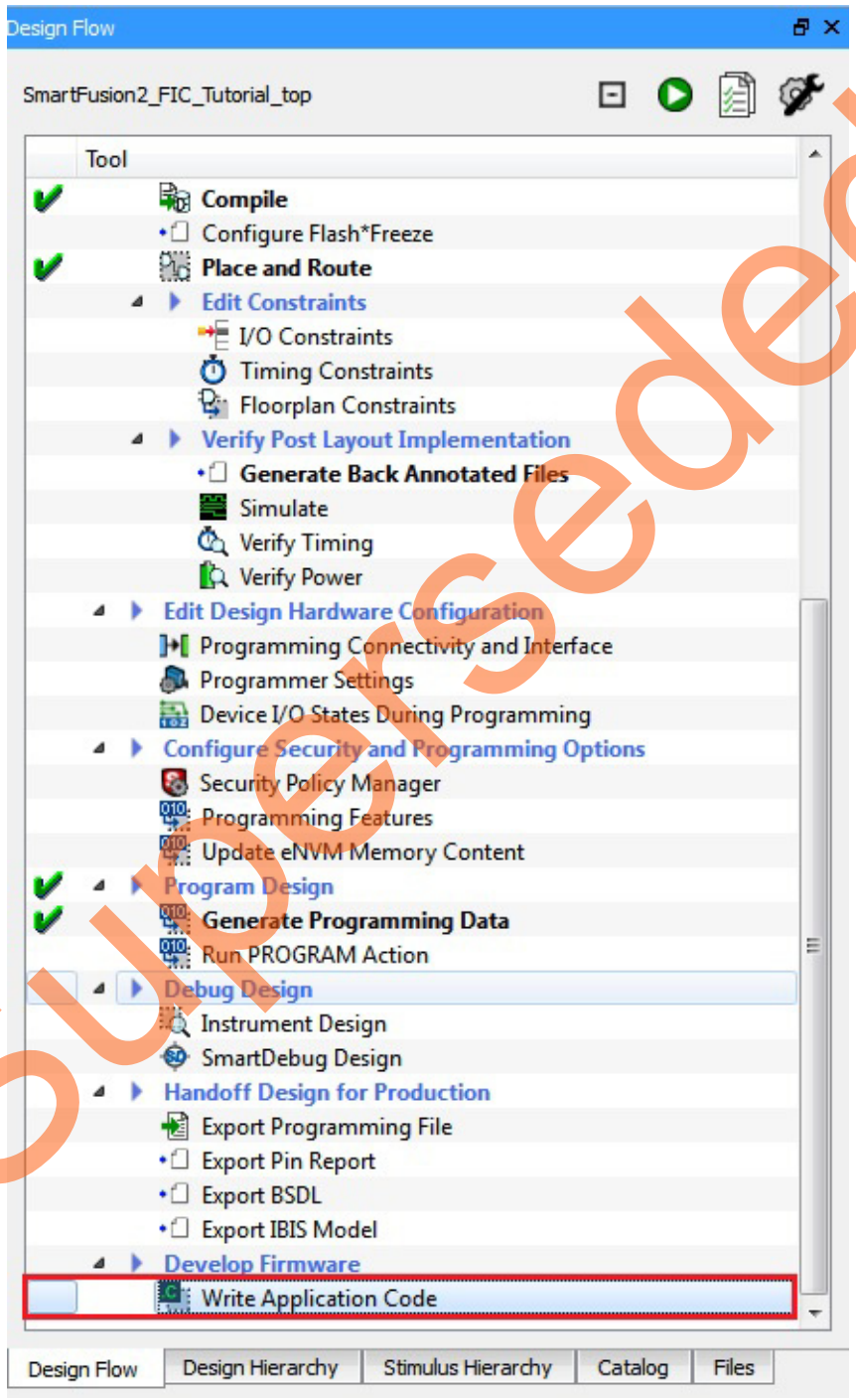

***Figure 36* • Write Application Code**

2. The SoftConsole perspective displays similar to Figure 37.

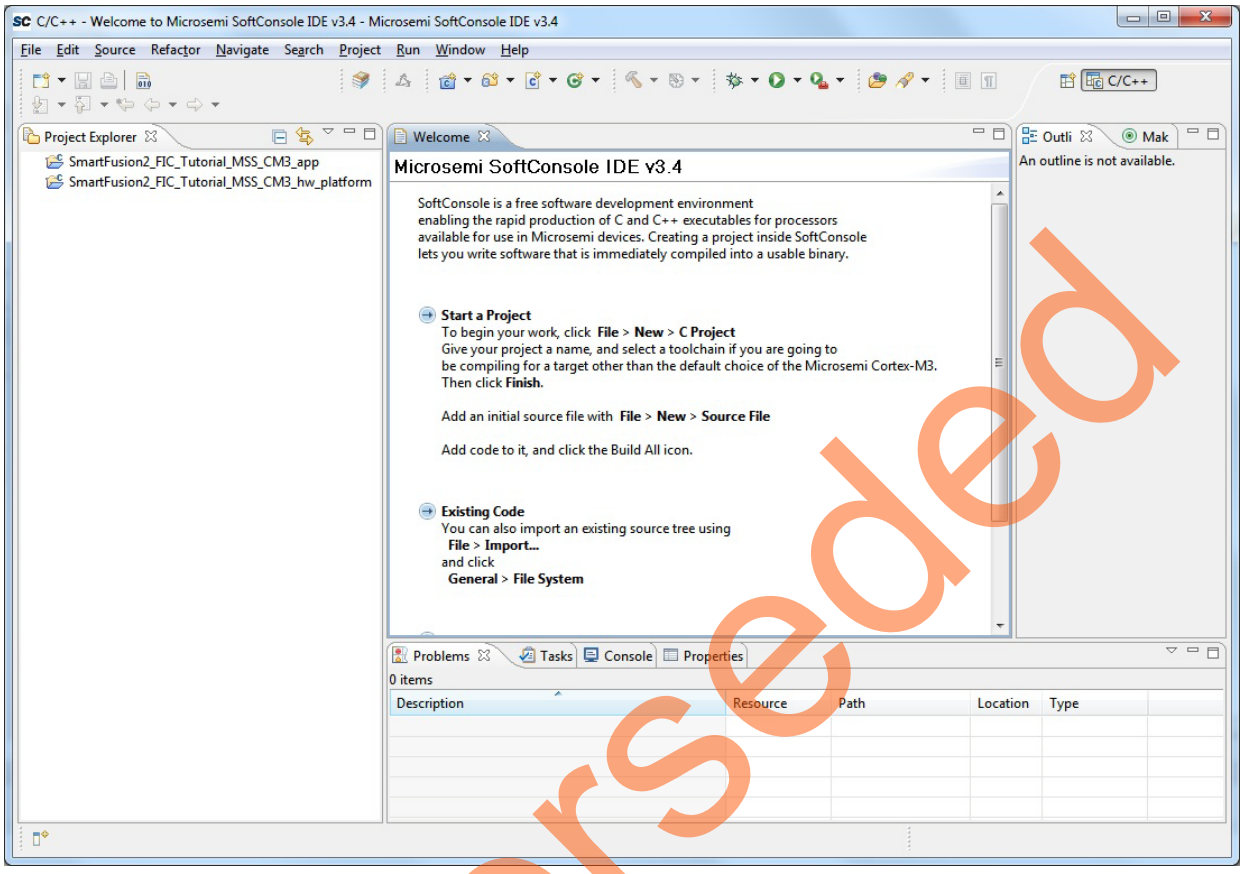

***Figure 37* • SoftConsole Perspective**

3. Go to the source folder in the downloaded design files folder, copy the code from the `source_dev.c` file. In SoftConsole editor under **SmartFusion2_FIC_Tutorial_MSS_CM3_app** project, place the copied code in `the main.c` file and delete the existing code.

Note: For the SmartFusion2 Starter Kit Board, the Code provided in **source_starter.c** in source files.

4. Select **Project > Clean** to perform a clean build. Accept the default settings in the **clean** dialog box and click **OK** as shown in Figure 38.

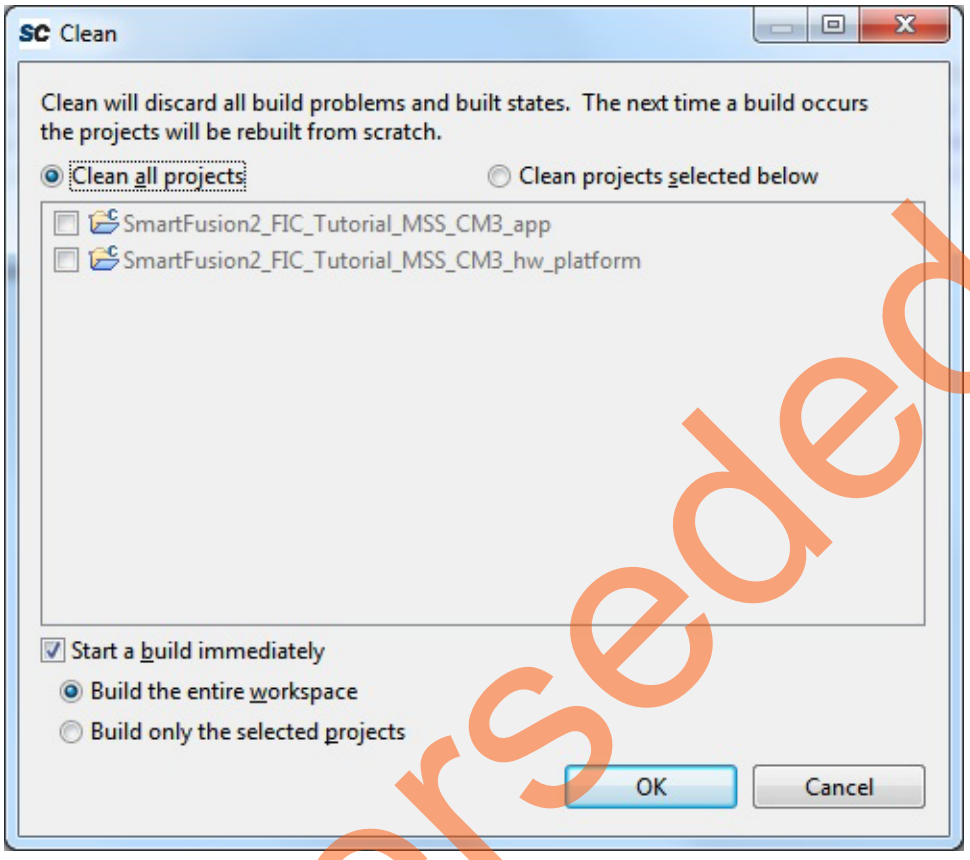

***Figure 38* • Clean and Build window**

5. Ensure that there are no errors and warnings. Use next steps to configure the Serial Terminal.

Step 7: Configuring the Serial Terminal Emulation Program

Prior to running the application program, you need to configure the terminal emulator program (HyperTerminal) on your PC. Perform the following steps to use the SmartFusion2 Development Kit Board or SmartFusion2 Starter Kit Board:

1. Connect one end of the USB mini-B cable to the respective USB connector provided on the SmartFusion2 Board.
2. Connect the other end of the USB cable to the host PC. Make sure that the USB to UART bridge drivers are automatically detected, as shown in Figure 39 and Figure 40.

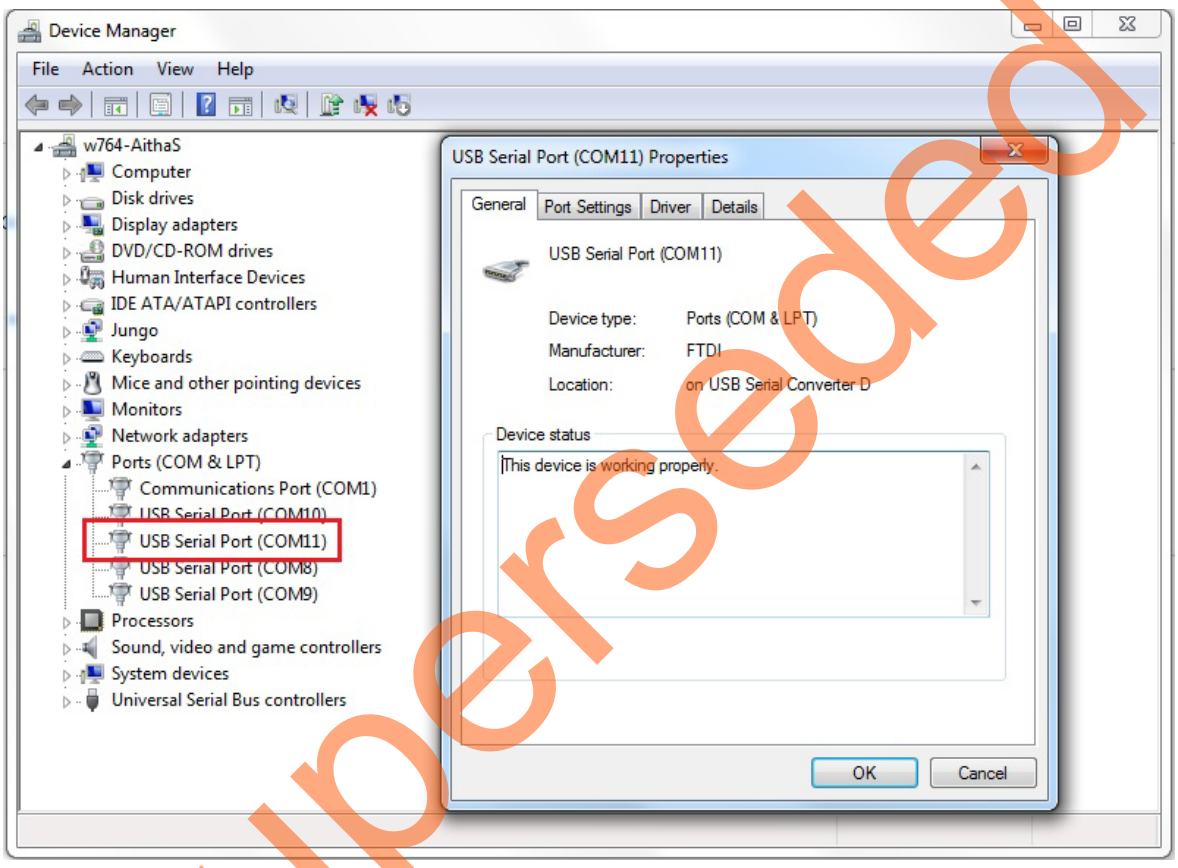

Figure 39 • **SmartFusion2 Development Kit USB Serial Port Drivers**

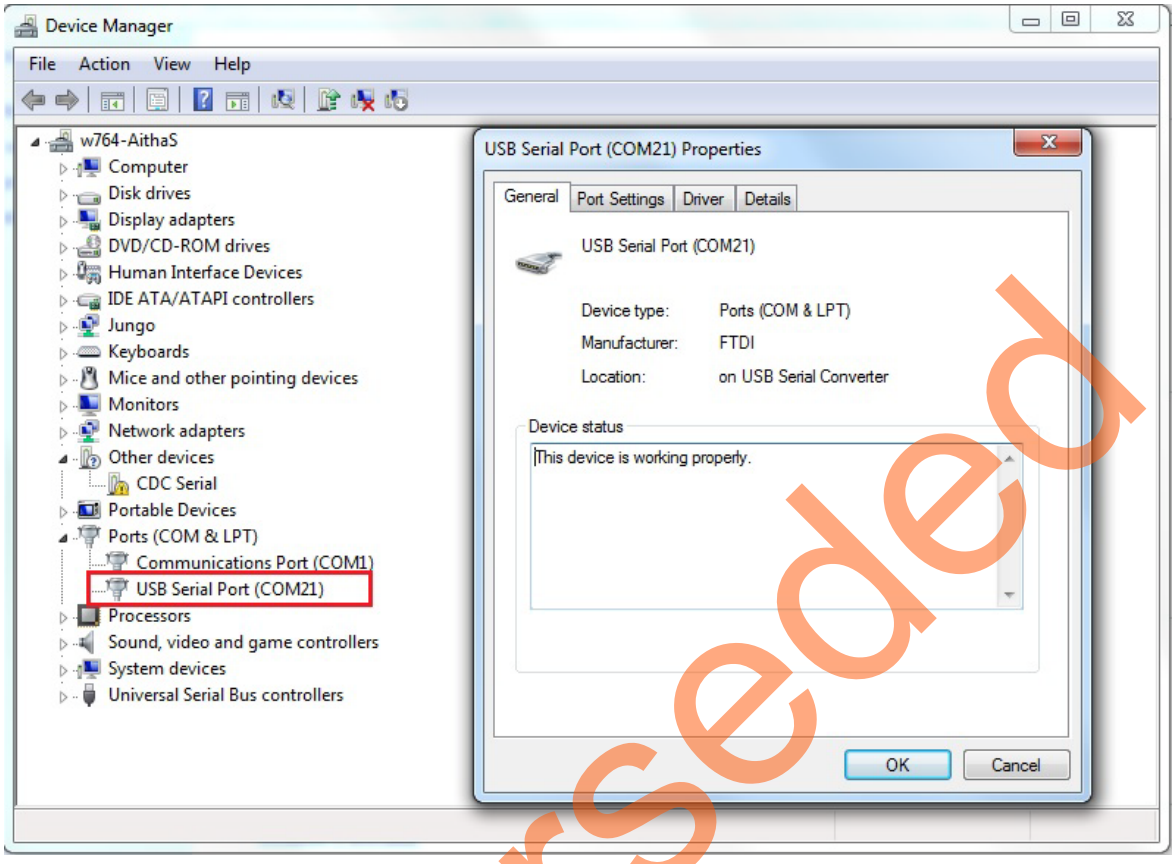

***Figure 40* • SmartFusion2 Starter Kit USB Serial Port Drivers**

3. If USB to UART bridge drivers are not installed, download and install the drivers from *www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip*.
4. Start a HyperTerminal with the baud rate set to 57600, 8 data bits, 1 stop bit, no parity, and no flow control.
5. Refer to the *Configuring Serial Terminal Emulation Programs Tutorial* for configuring HyperTerminal, Tera Term, and PuTTY.

Step 8: Debugging the Application Project using SoftConsole

Use the following steps to debug the application project using SoftConsole:

1. Select **SmartFusion2_FIC_Tutorial_MSS_CM3_app** in Project Explorer.
2. Select the **Debug Configurations** from the **Run** menu of the SoftConsole. The Debug dialog is displayed.
3. Double-click on **Microsemi Cortex-M3 Target** to display an image similar to Figure 41.

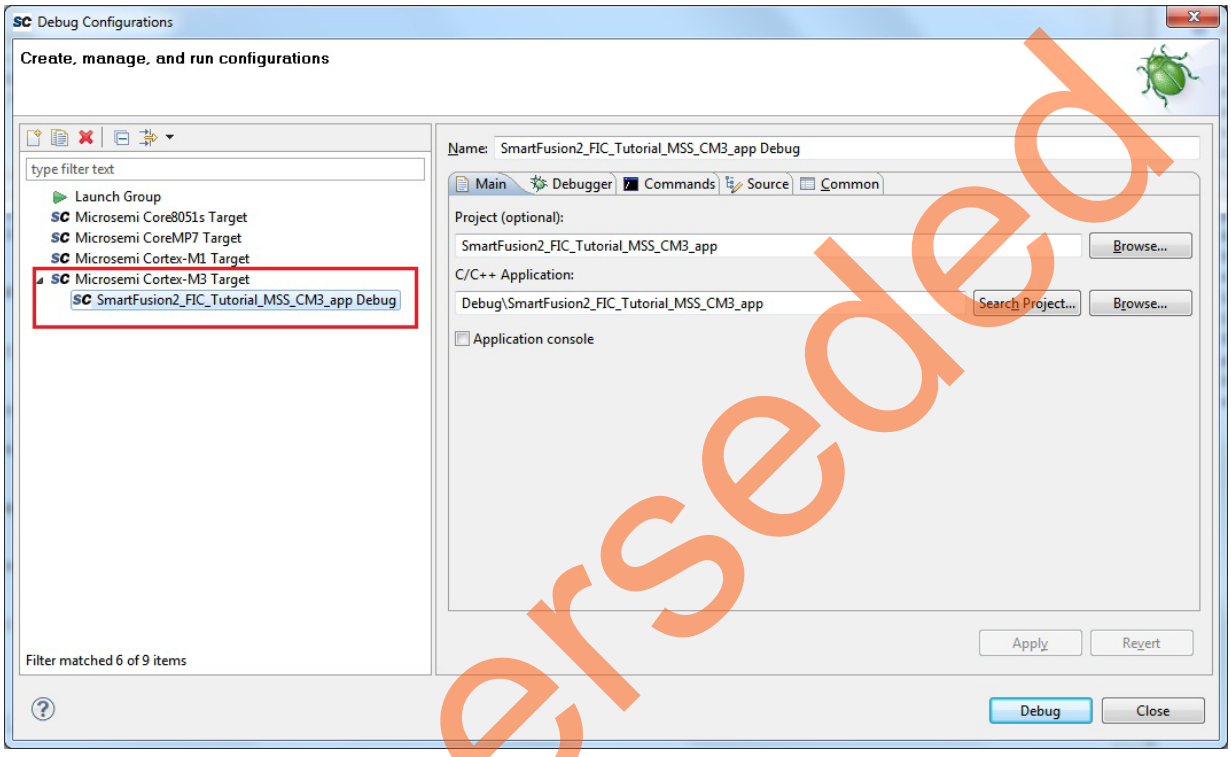

***Figure 41* • Debug Window**

4. Confirm that the following appear on the Main tab in the Debug window:
   - Name: SmartFusion2_FIC_Tutorial_MSS_CM3_app Debug
   - Project:SmartFusion2_FIC_Tutorial_MSS_CM3_app
   - C/C++ Application: DebugSmartFusion2_FIC_Tutorial_CM3_app
5. Click **Apply** and **Debug**.

6. Click **Yes**, when prompted for **Confirm Perspective Switch**. This displays the debug view mode as shown in Figure 42.

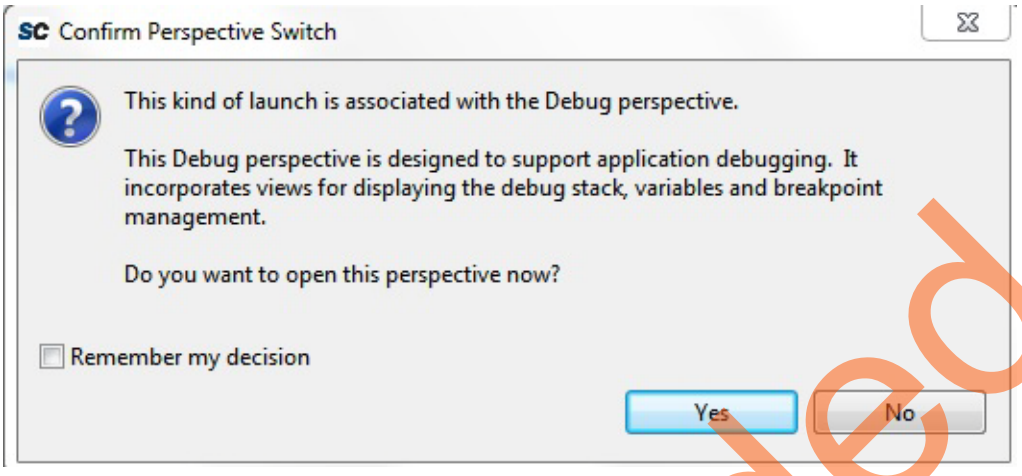

Figure 42 • **Confirm Perspective Switch**

7. Debug Perspective is similar as shown Figure 43.

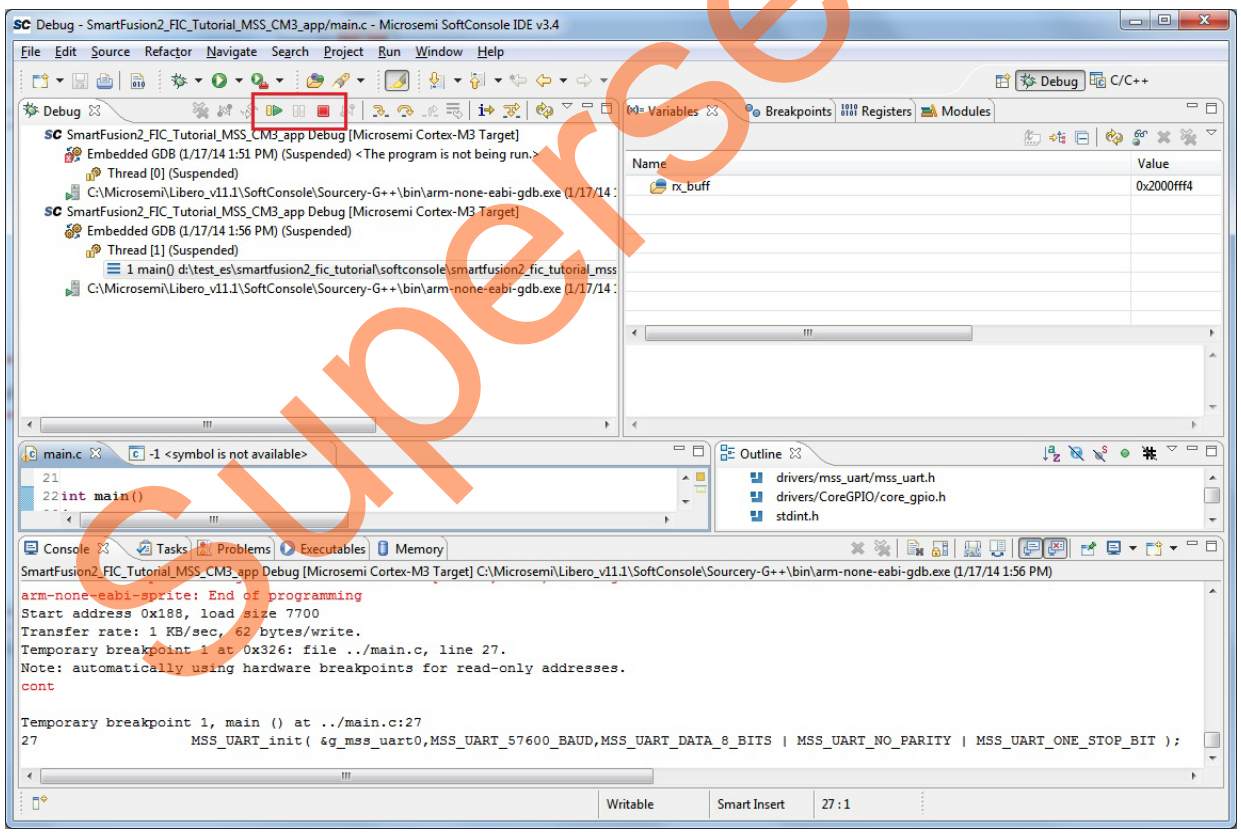

Figure 43 • **Debug Perspective**

8. Run the application by clicking **Run > Resume** or click Run icon on the SoftConsole toolbar.

The Application options along with the greeting message are displayed in the terminal program window as shown in Figure 44.

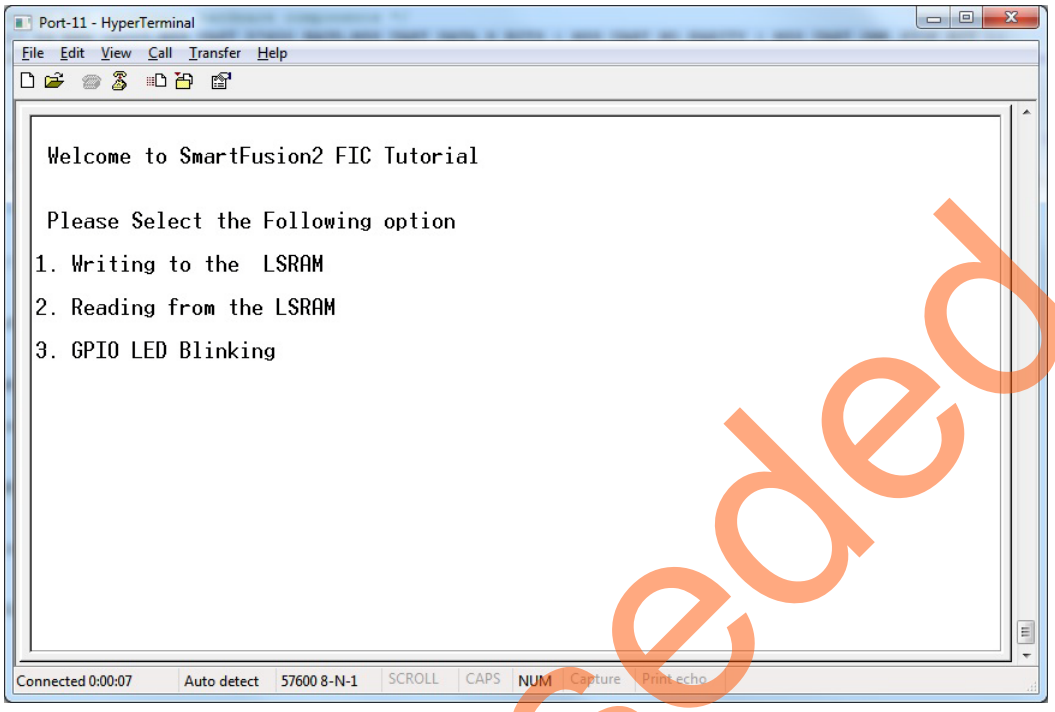

Figure 44 • **HyperTerminal Window**

9. Select **Writing to LSRAM**, It prompts for “Enter the offset address between 0 to 8188 to write and press Enter” as shown in Figure 45.

Note: In PuTTY, press CTRL+J instead of Enter.

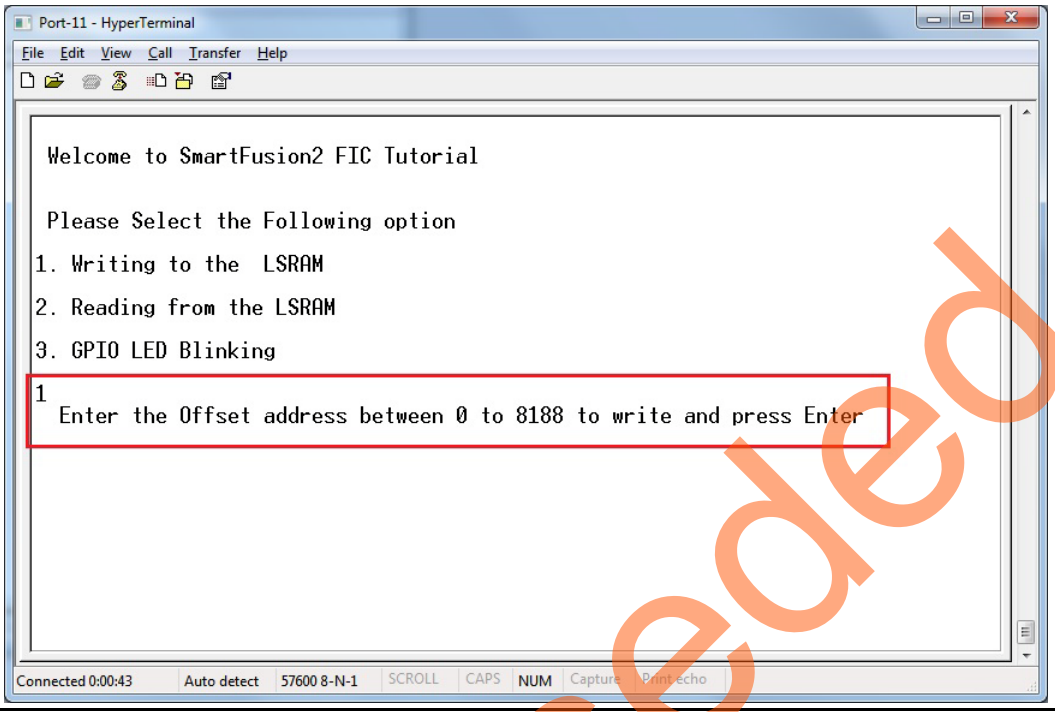

Figure 45 • **Writing to LSRAM**

10. After Entering the offset address, it prompts for “Enter data to write” as shown in Figure 46.

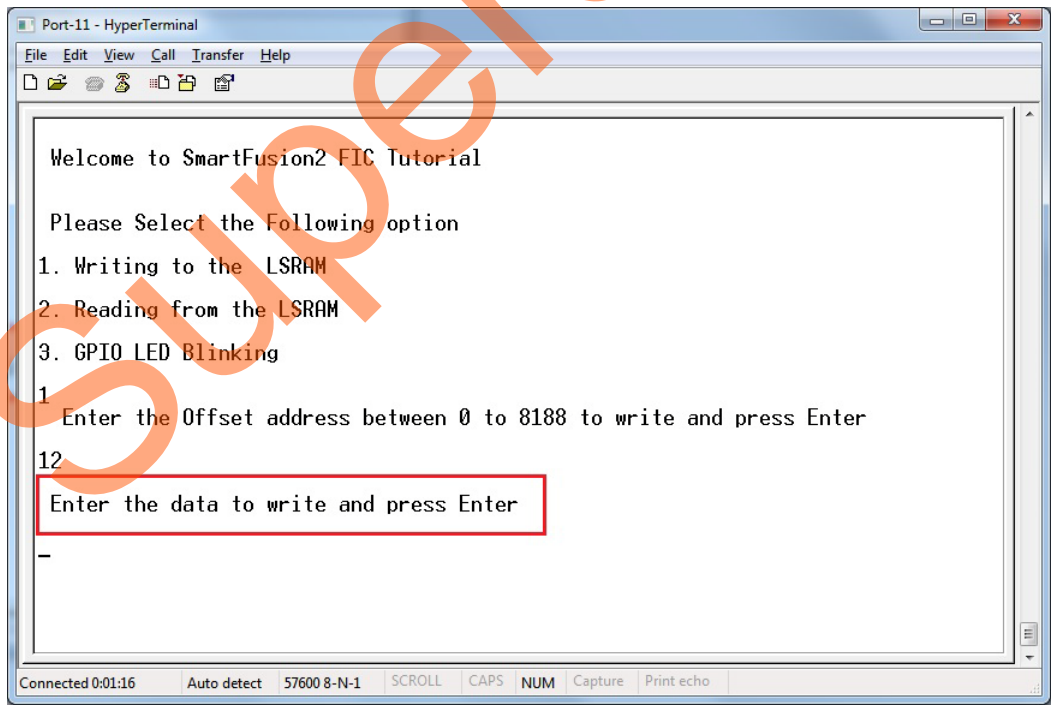

Figure 46 • **Writing to LSRAM**

11. Select **Reading from the LSRAM**, it prompts for “Enter the offset address to read and press Enter” as shown in Figure 47.

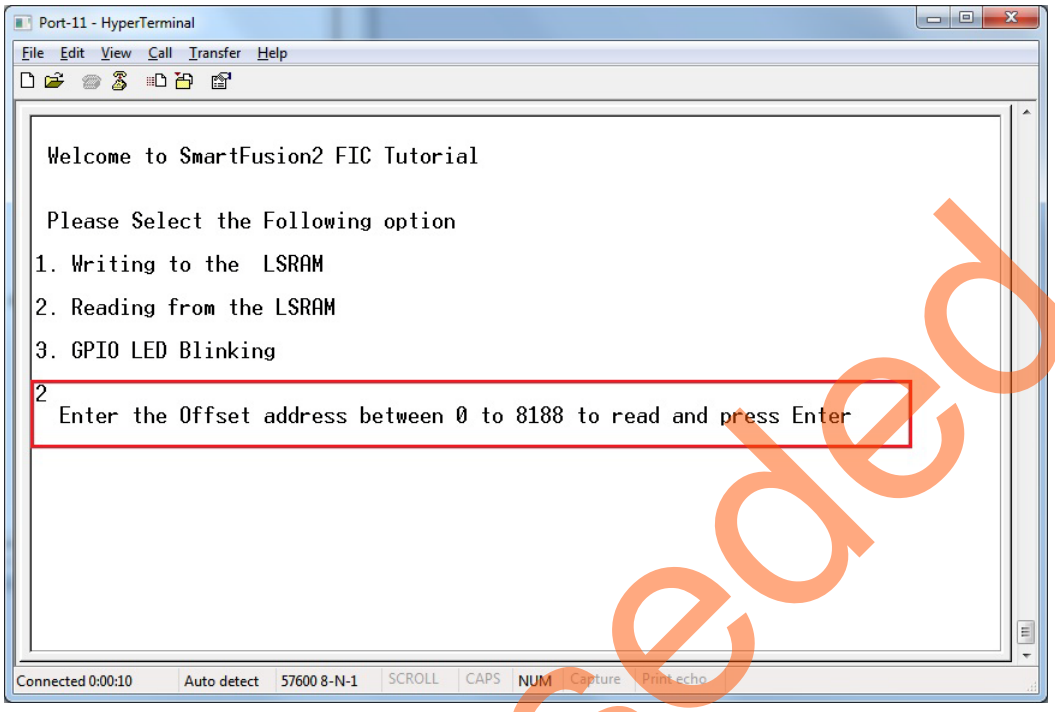

***Figure 47* • Reading from LSRAM**

12. Select GPIO LED Blinking, it Prompts for “Enter the number between 0 to 255 and press Enter” as shown in Figure 48.

Note: For SmartFusion2 Starter Kit HyperTerminal window as shown in Figure 49.

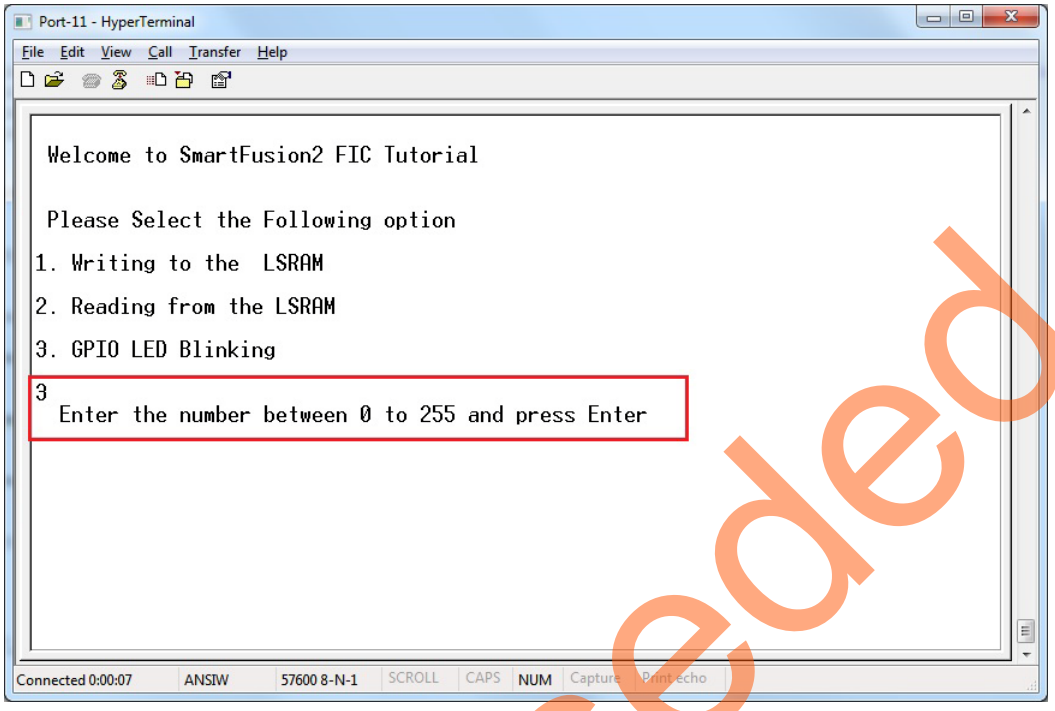

Figure 48 • **Selecting GPIO LED Blinking**

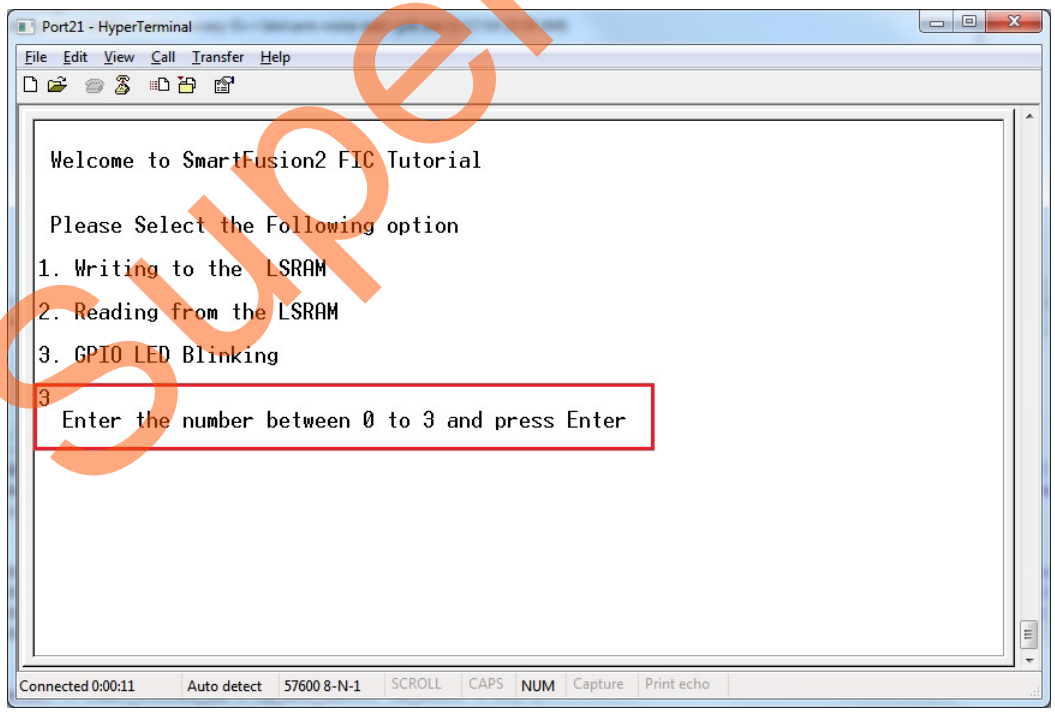

Figure 49 • **Selecting GPIO LED Blinking**

Step 9: Building Executable Image in Release mode

You can build an application executable image in —release modell and load it into eNVM for executing code in eNVM of SmartFusion SoC device. You can load the application executable image into eNVM with the help of eNVM data storage client from System Builder eNVM Configurator. In release mode, you cannot use SoftConsole debugger to load the executable image into eNVM.

Conclusion

This tutorial outlined the design flow for creating a SmartFusion2 SoC FPGA project using Libero SoC design software, configuring the SmartFusion2 SoC FPGA MSS, interfacing fabric peripherals to the SmartFusion2 SoC FPGA MSS using fabric interface controllers (FIC_0 and FIC_1), simulation of the design using BFM commands and running the application design on board.

A – List of Changes

The following table lists critical changes that were made in each revision.

Revision	Changes	Page
Revision 8 (May 2014)	Updated the document for Libero v11.3 software release (SAR 56454).	NA
Revision 7 (February 2014)	Updated the document (SAR 54212).	NA
Revision 6 (November 2013)	Updated the document for Libero version 11.2 (SAR 52904).	NA
Revision 5 (April 2013)	Updated the document for 11.0 production SW release (SAR 47302).	NA
Revision 4 (February 2013)	Updated the document for Libero 11.0 Beta SP1 software release (SAR 44868).	NA
Revision 3 (November 2012)	Updated the document for Libero 11.0 Beta SPA software release (SAR 42904).	NA
Revision 2 (October 2012)	Updated the document for Libero 11.0 Beta launch (SAR 41696).	NA
Revision 1 (May 2012)	Updated the document for LCP2 software release (SAR 38954).	NA

Note: The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.

B – 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.

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

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.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

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.

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.

My Cases

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

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. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

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_itar@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 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 and 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; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.