Interfacing User Logic with the Microcontroller Subsystem - Libero SoC v11.6 Design Flow

TU0310 Tutorial

Microsemi

Table of Contents

Interfacing User Logic with the Microcontroller Subsystem . . . 3
Introduction . . . 3
Design Requirements . . . 4
Design Files . . . 4
Design Description . . . 4
Design Steps . . . 5
Step 1: Creating a New Libero SoC Project . . . 6
Step 2: Modifying User BFM Script for Simulation . . . 21
Generating Testbench . . . 22
Step 3: Simulating Design Using BFM Models . . . 24
Step 4: Generating Programming File . . . 30
Step 5: Programming the SmartFusion2 Board Using Flash Pro . . . 32
Jumper Settings for SmartFusion2 Security Evaluation Kit Board . . . 32
Jumper Settings for SmartFusion2 Starter Kit Board . . . 32
Programming the Device . . . 33
Step 6: Building the Software Application through SoftConsole . . . 36
Step 7: Configuring the Serial Terminal Emulation Program . . . 43
Step 8: Debugging the Application Project using SoftConsole . . . 45
Step 9: Building Executable Image in Release mode . . . 50
Conclusion . . . 50

A List of Changes . . . 51

B Product Support . . . 52
Customer Service . . . 52
Customer Technical Support Center . . . 52
Technical Support . . . 52
Website . . . 52
Contacting the Customer Technical Support Center . . . 52
Email . . . 52
My Cases . . . 53
Outside the U.S. . . . 53
ITAR Technical Support . . . 53

Interfacing User Logic with the Microcontroller Subsystem

Introduction

This tutorial describes 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 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 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 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 *UG0331: SmartFusion2 Microcontroller Subsystem User Guide* for more information on FIC blocks in the Fabric Interface Controller chapter of.

This tutorial covers the following:

1. Creating a project for a SmartFusion2 SoC FPGA using the Microsemi Libero SoC toolset.
2. Using SmartFusion2 System Builder to Configure MSS and generate System Builder Component.
3. Configuring fabric interface controllers (FIC_0) 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.

Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
• SmartFusion2 Security Evaluation Kit Board or • SmartFusion2 Starter Kit Board (SF2-STARTER-KIT) • FlashPro4 programmer	Rev D or later
USB Cables	–
Desktop Computer or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero SoC	v11.6
SoftConsole	v3.4 SP1
Host PC Drivers	*USB Drivers*

Design Files

You can download the associated project files for this tutorial from the Microsemi website:
http://soc.microsemi.com/download/rsc/?f=m2s_tu0310_liberov11p6_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 MSS block, one CCC block, on-chip 25/50 MHz RC oscillator and two different slaves in the FPGA fabric. The MSS FIC_0 is configured for the AHBL master interface is connected to the slaves CoreAHBLSRAM and CoreGPIO using the CoreAHBLite, CoreAHBTOAPB, and CoreAPB3 bus interfaces. Figure 1 on page 5 shows the block diagram of the design. The ARM® Cortex®-M3 processor or any other MSS master can access these slaves through the FIC blocks. In this design, you can:

- Verify the bus read and write to the fabric peripherals from the MSS side using BFM models.
- Perform read and write to the CoreAHBLSRAM memory, configure the CoreGPIO block, and set GPIO outputs using a BFM script.
- Validate the bus read and write to the CoreAHBLSRAM, set the GPIO to blink the LEDs on the SmartFusion2 Security Evaluation Kit board and SmartFusion2 Starter Kit board.

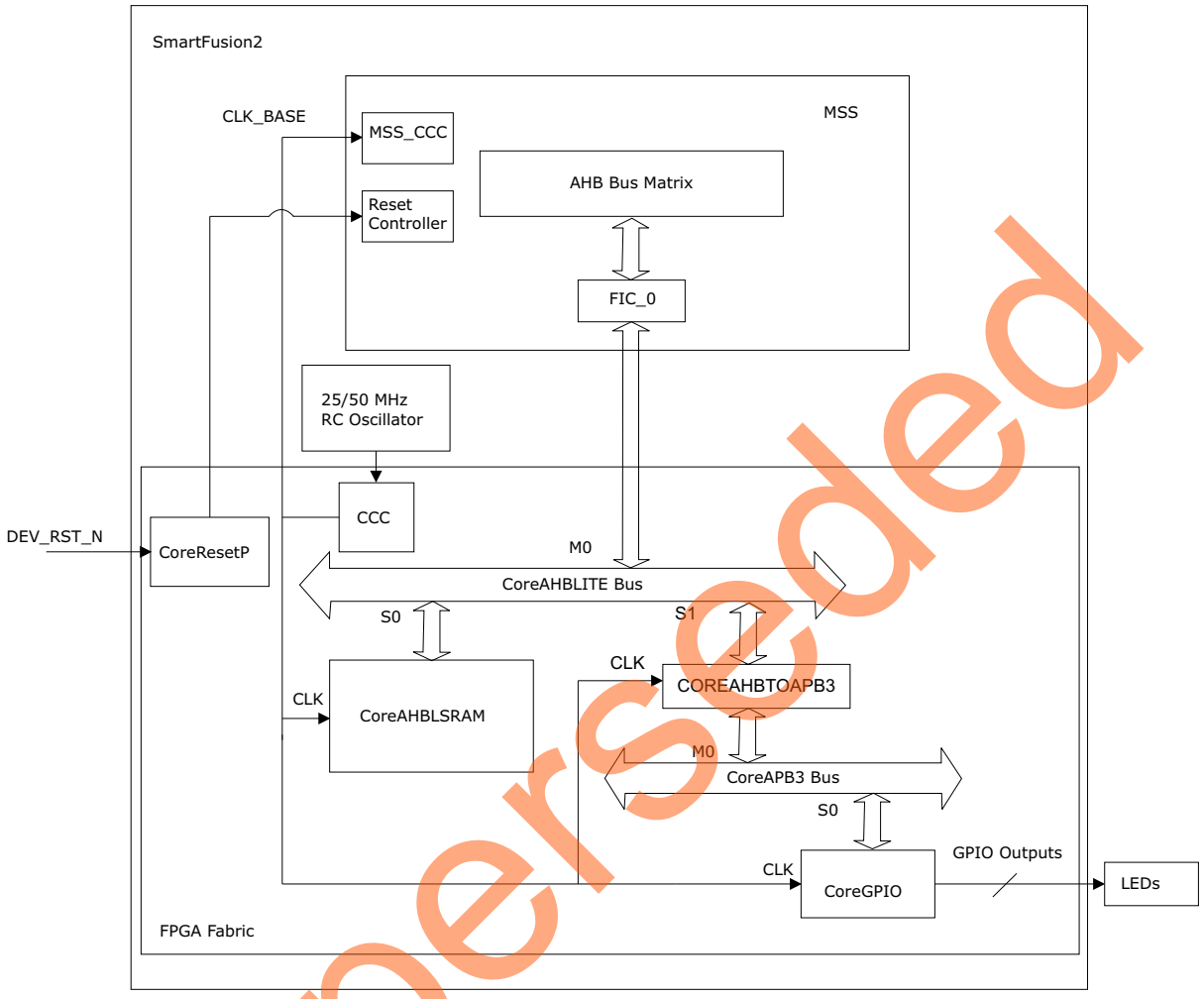

***Figure 1* • Block Diagram of the Design**

Design Steps

The major steps to run this tutorial are as follows:

1. Creating a new Libero SoC project for the SmartFusion2 device.
2. Using SmartFusion2 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 FPGA device.
6. Open the software project in SoftConsole and write the application program.
7. Run the design on the SmartFusion2 Security Evaluation Kit board or SmartFusion2 Starter Kit board.

Step 1: Creating a New Libero SoC Project

The following steps describe how to create a new Libero SoC project:

1. Open Libero SoC design software (**Start > Programs > Microsemi Libero SoC 11.6 > Libero SoC 11.6**) 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 v11.6 or latest.
2. Select **New Project** from the **Project** menu.
3. Enter the following **New Project** information as shown in and click **Next**.
    - **Project Name**: SmartFusion2_FIC_Tutorial
    - **Project Location**: Select an appropriate location (for example, D:/Microsemi_prj)
    - **Preferred HDL Type**: Verilog

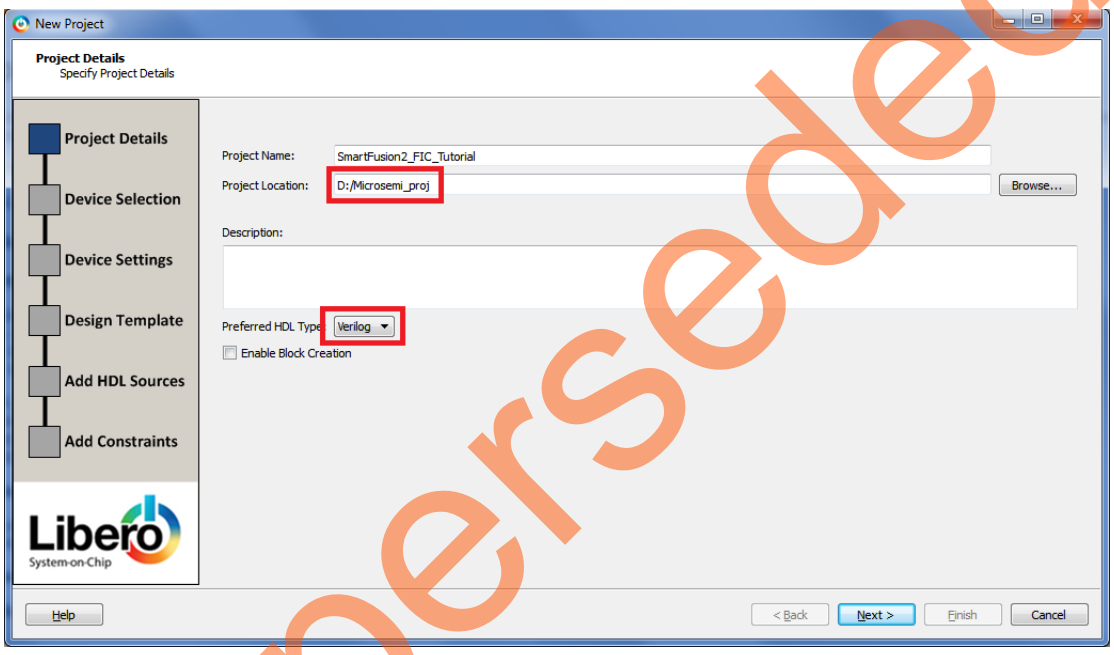

***Figure 2* • Libero SoC New Project Dialog Box**

4. Select the following values from the drop-down lists for **Device Selection**, as shown in Figure 3.
    - **Family**: SmartFusion2
    - **Die**: M2S090TS
    - **Package**: 484 FBGA
    - **Speed**: -1
    - **Core Voltage**: 1.2
    - **Range**: COM

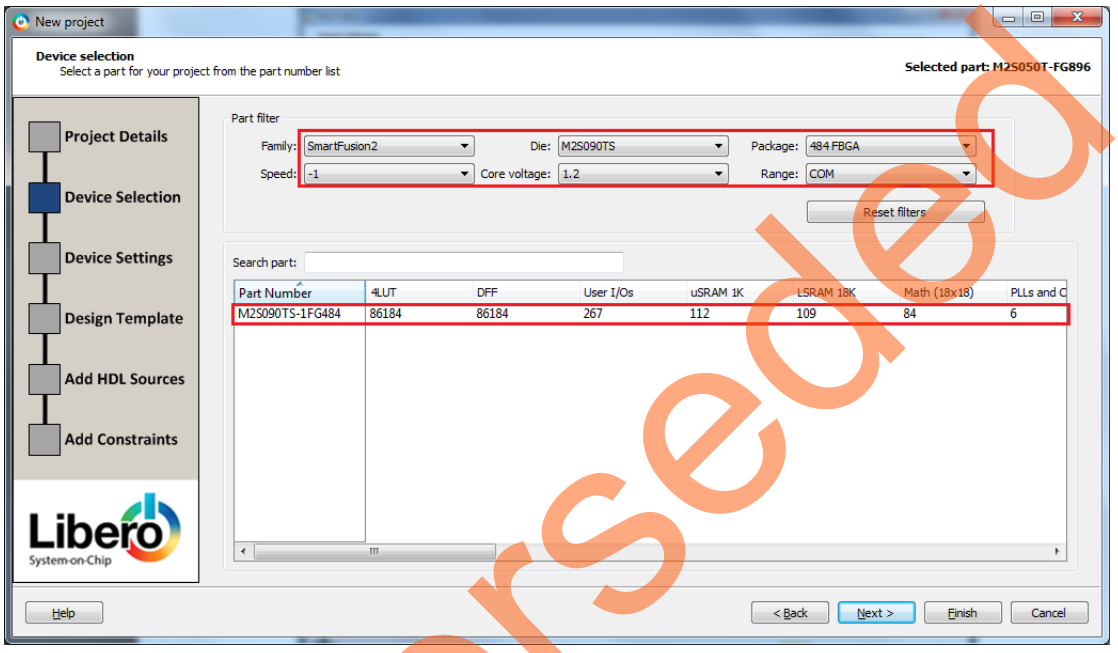

***Figure 3* • New Project - Device Selection**

Note: For SmartFusion2 Starter Kit (SF2-STARTER-KIT with M2S010-FGG484 device), Die: M2S010T, Package: 484 FBGA.

5. Click **Next**.

6. Select the information for **Device Settings** as shown in Figure 4 and click **Next**.

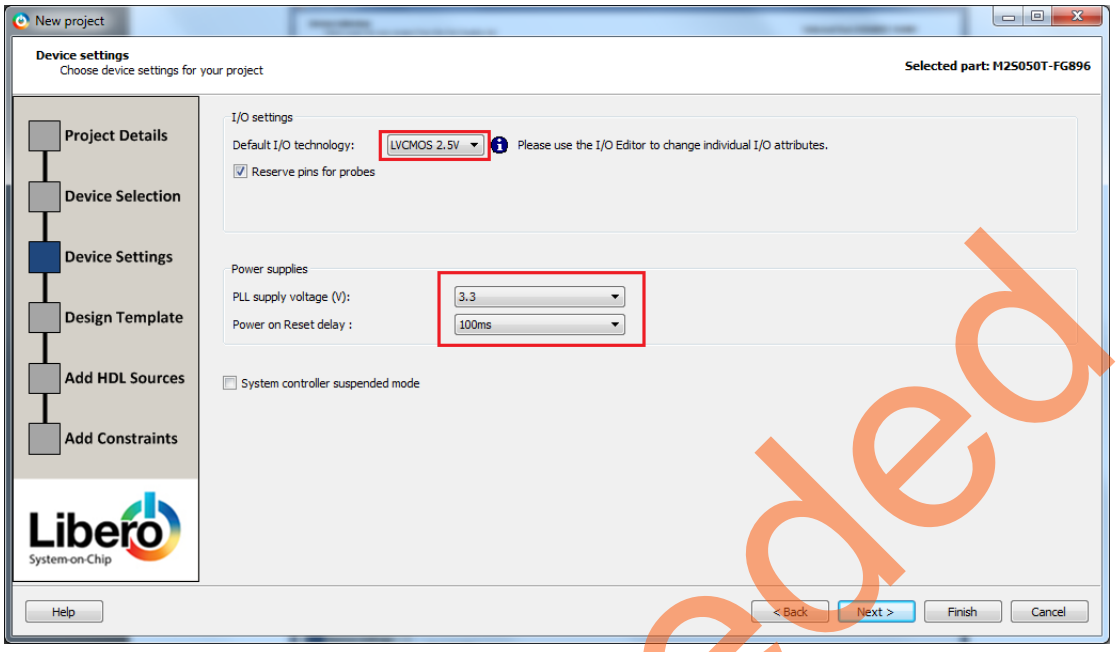

***Figure 4* • New Project - Device Settings**

7. Select **Create a System Builder based design** under **Design Templates and Creators** as shown in Figure 5.

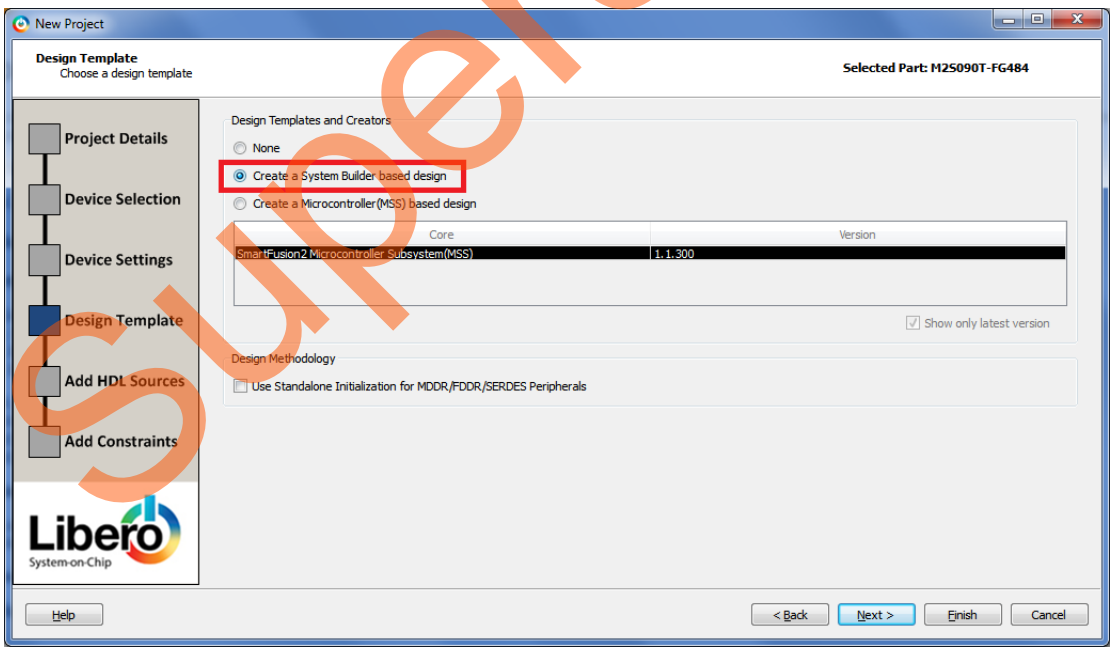

***Figure 5* • New Project - Design Template**

8. Click **Finish**.

9. After selecting the **Use System Builder**, enter the name of the system in the **Enter a name for your system** dialog box, as shown in Figure 6.

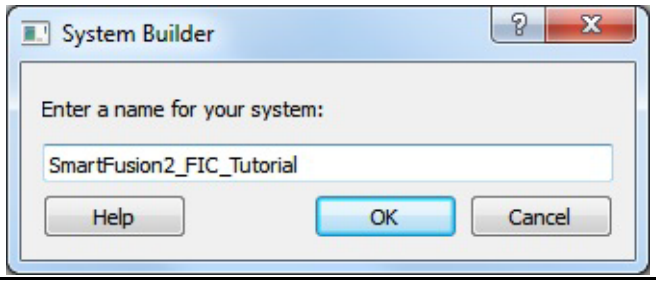

***Figure 6* • Libero SoC Project Window**

10. Enter **SmartFusion2_FIC_Tutorial** as the name of the system and click **OK**. The **System Builder** window is displayed, as shown in Figure 7.

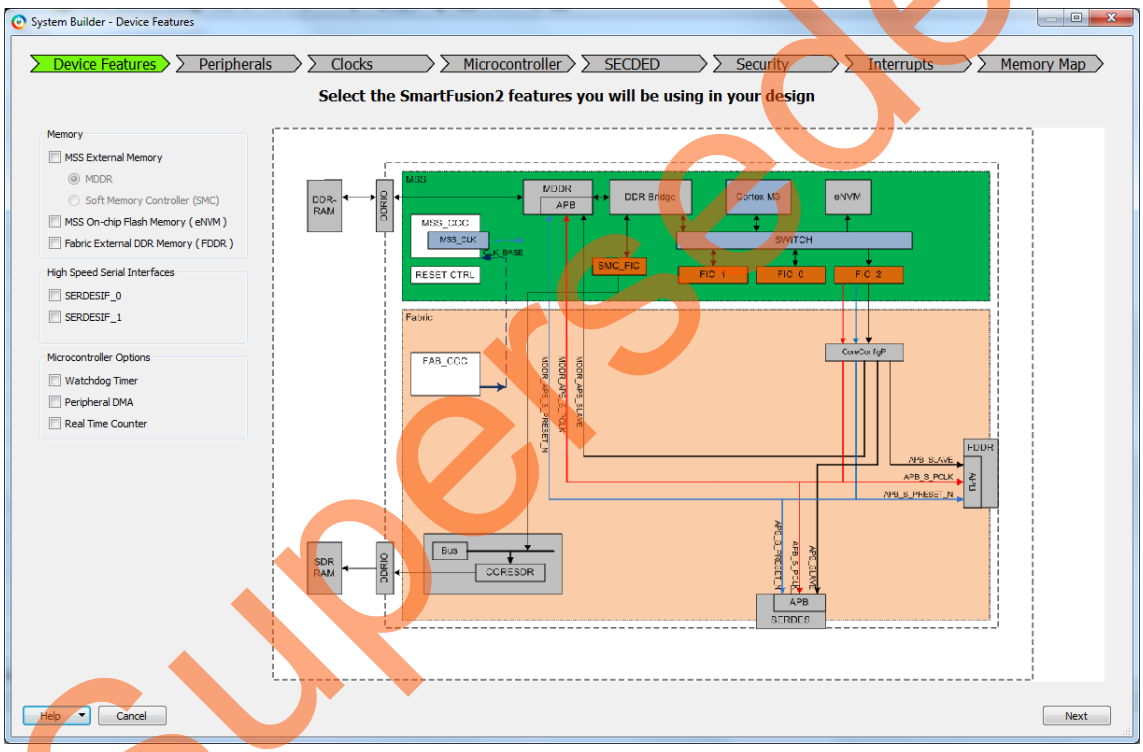

***Figure 7* • SmartFusion2 System Builder Device Features**

11. Select **Next**. **System Builder- Peripherals** page is displayed, as shown in Figure 8 and Figure 9 on page 11.This tutorial uses the MSS MMUART peripherals.

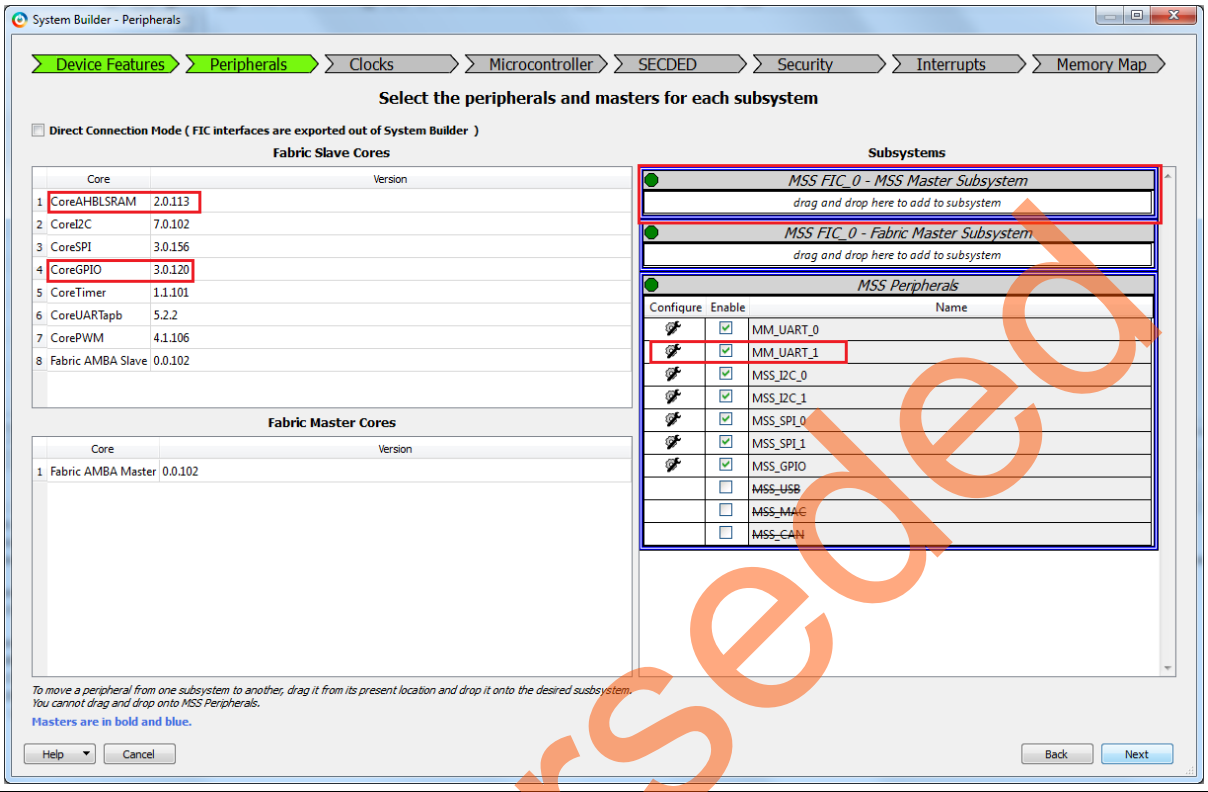

Figure 8 • **SmartFusion2 System Builder Peripherals (M2S090TS Device)**

12. Select **MM_UART_1** for SmartFusion2 Security Evaluation Kit (**M2S090TS** device) or **MM_UART_0** for SmartFusion2 Starter Kit (**M2S010** device) and uncheck all the other peripherals, as shown in Figure 9 on page 11 and Figure 10 on page 12.

13. Drag the **CoreAHBLSRAM** and **CoreGPIO** IPs to **MSS FIC_0 - MSS Master Subsystem** for M2S090TS device**,** as shown in Figure 9. This tutorial uses the CoreAHBLSRAM and CoreGPIO IPs.

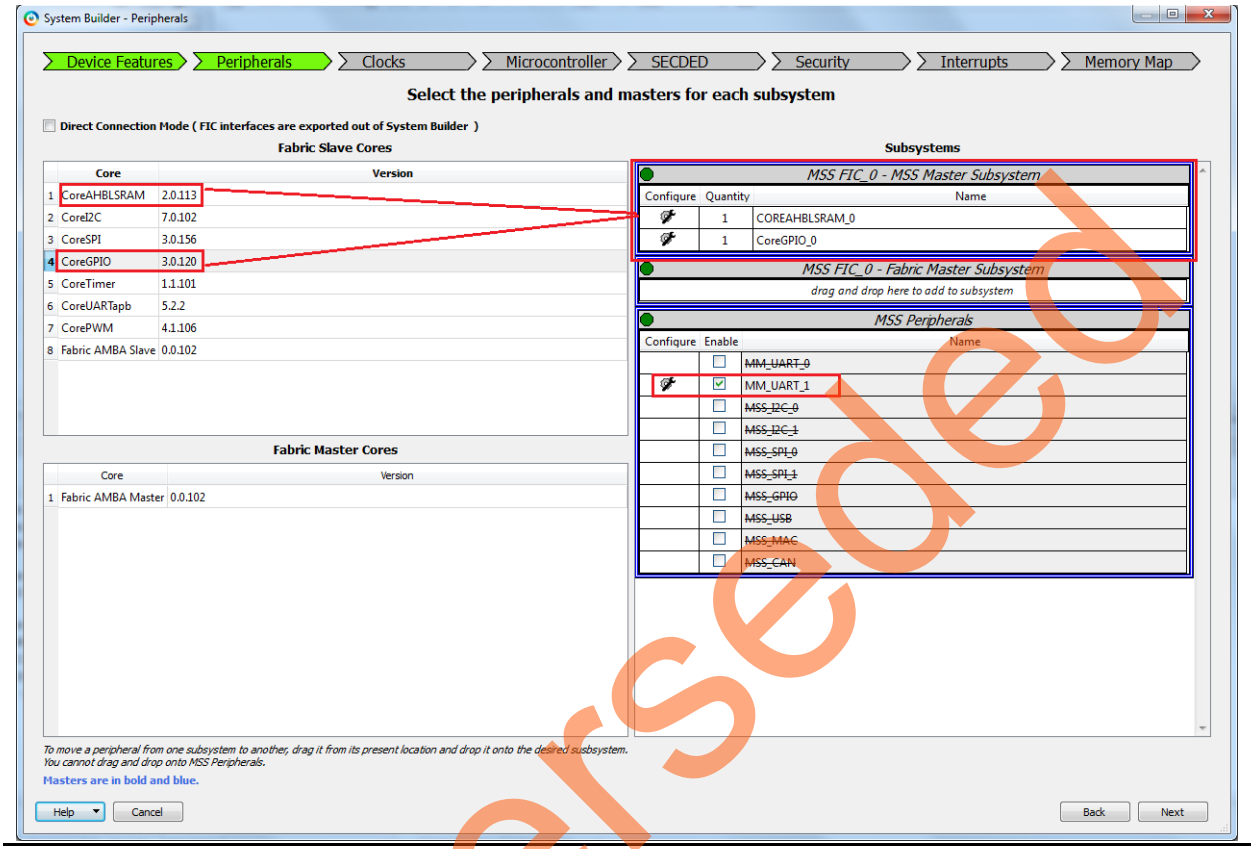

***Figure 9* • SmartFusion2 System Builder MSS Peripherals (M2S090TS Device)**

14. Drag the **CoreAHBLSRAM** and **CoreGPIO IPs** to **MSS FIC_0 - MSS Master Subsystem** for M2S010T device, as shown in Figure 10.

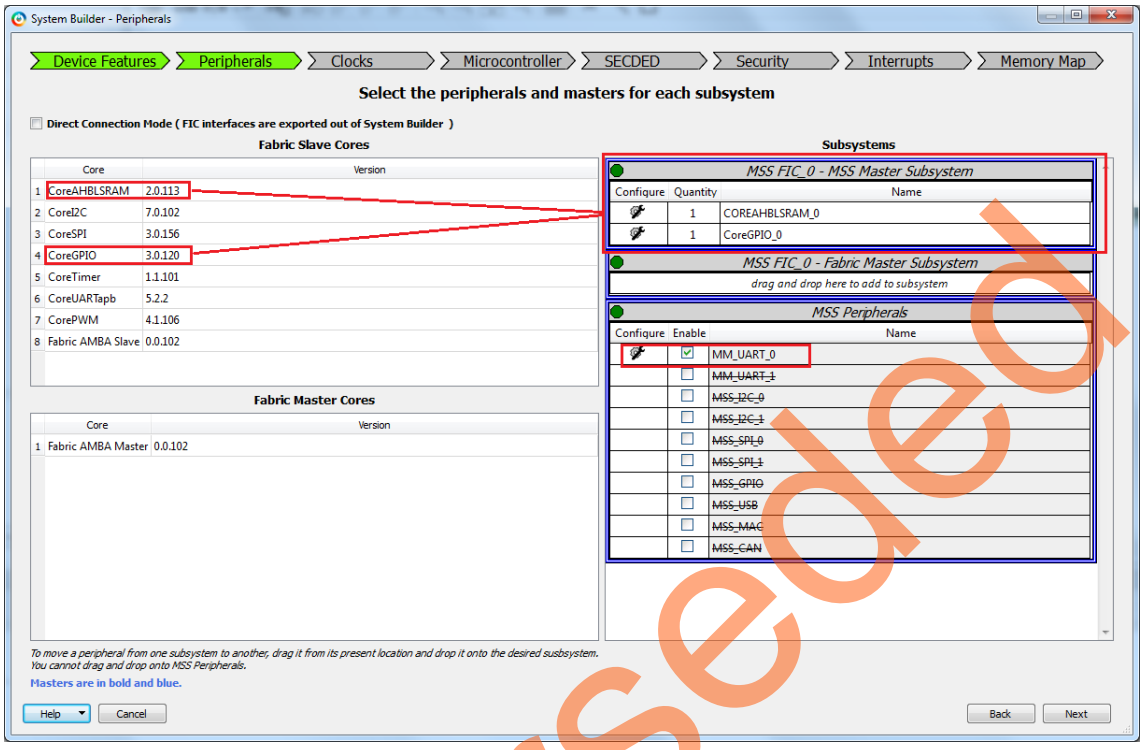

***Figure 10* • SmartFusion2 System Builder MSS Peripherals (M2S010T Device)**

15. Configure **COREAHBLSRAM_0** by clicking the Configure icon, as shown in Figure 11.

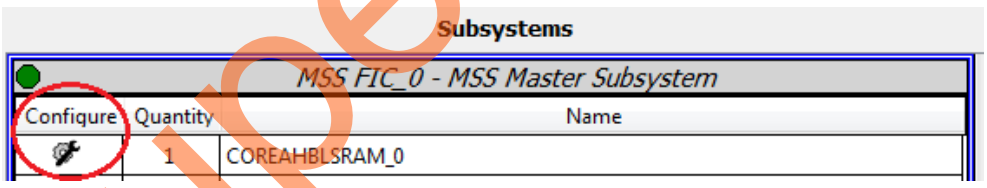

***Figure 11* • CoreAHBLSRAM Configuration**

Use the settings as shown in Figure 12.

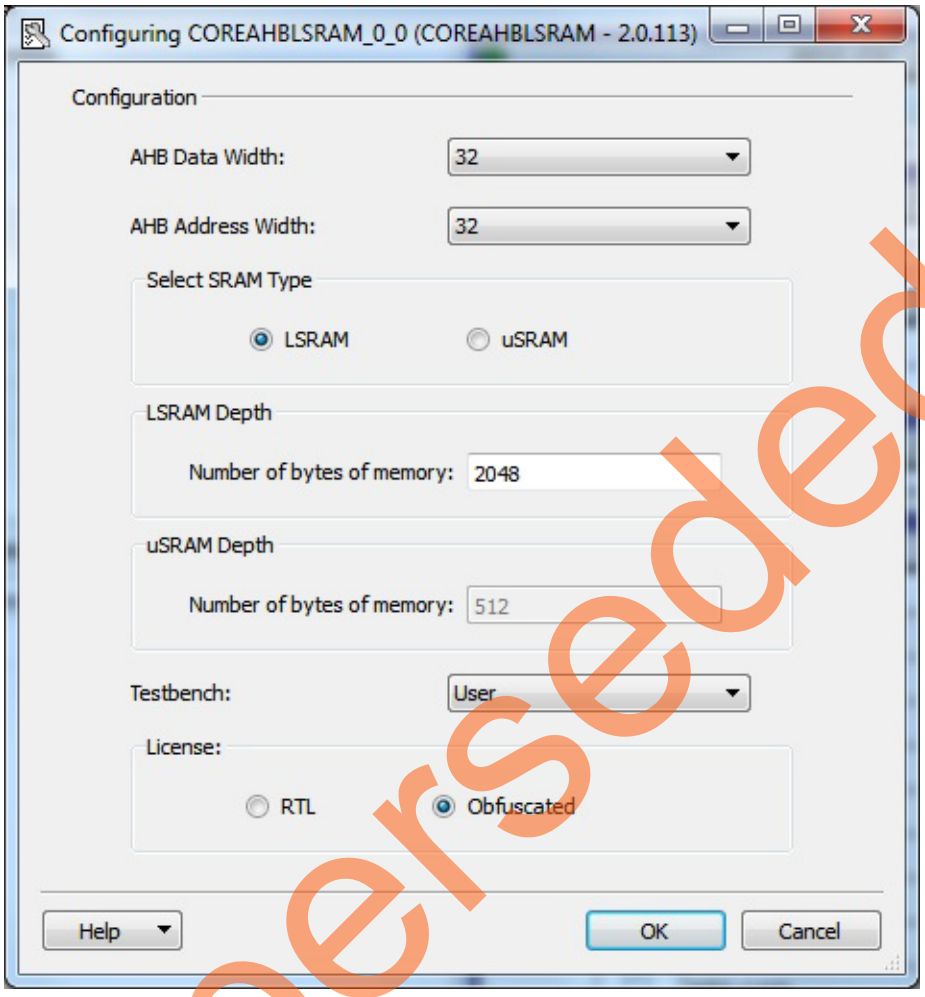

***Figure 12* • CoreAHBLSRAM Configuration**

16. Click **OK** after completion of COREAHBLSRAM configuration.
17. Click **CoreGPIO** Configure icon and use the following settings for SmartFusion2 Security Evaluation Kit board as shown in Figure 13 on page 14, and keep the rest at default states
    - **Number of I/Os**: 8 - For SmartFusion2 Security Evaluation Kit board
    - **Output enable**: Internal
    - **Fixed Config:** Select the check box
    - **I/O Type**: Output

Note: For SmartFusion2 Starter Kit, select Number of I/Os: 2.

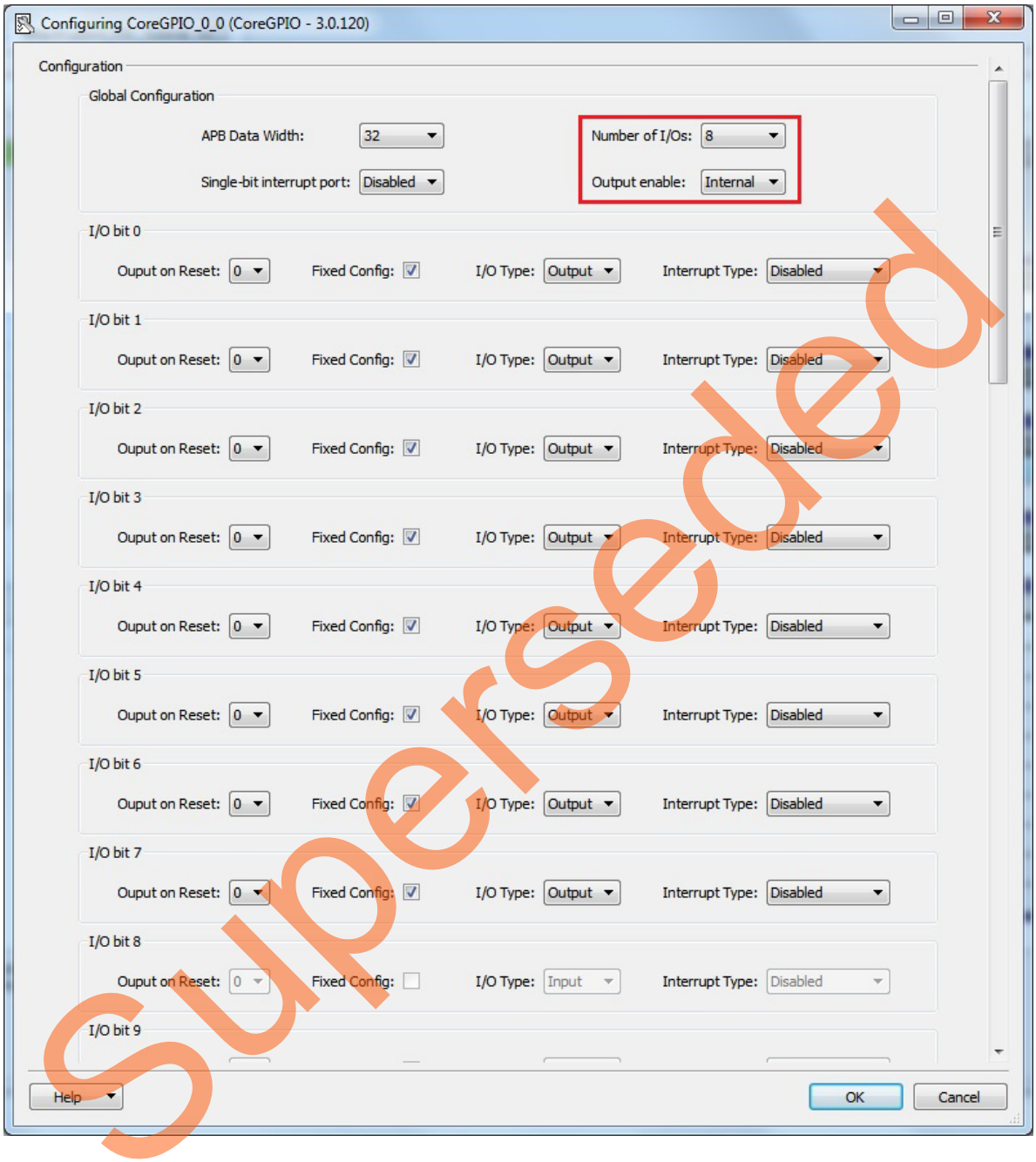

***Figure 13* • CoreGPIO Configuration**

Note: For SmartFusion2 Starter Kit board, the CoreGPIO Configuration window is displayed, as shown in Figure 14.

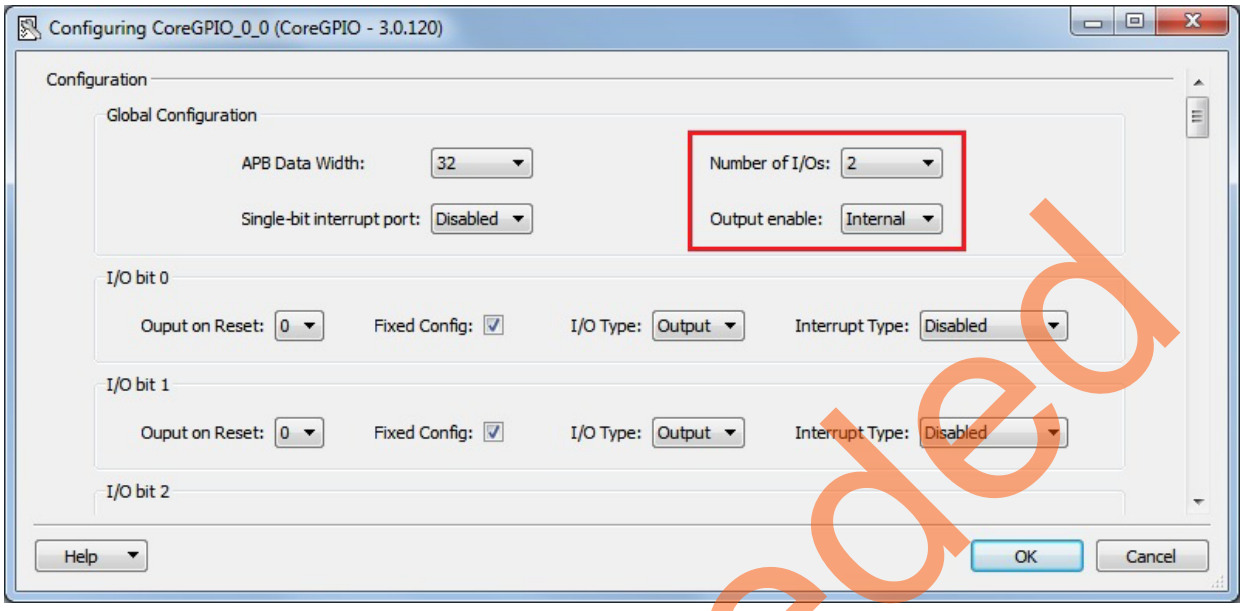

***Figure 14* • CoreGPIO Configuration**

18. Click **OK** after completion of CoreGPIO configuration.
19. Double-click the **MM_UART_1** configure icon for M2S090TS device and **MM_UART_0** configure icon for M2S010T device.
20. Select **IO** from the **Connect To** drop-down list and retain the default settings, as shown in Figure 15.

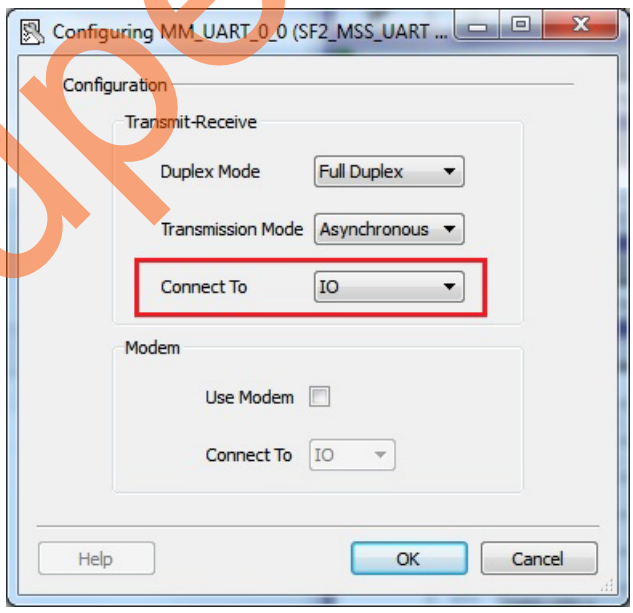

***Figure 15* • MM_UART Configuration**

21. Click **OK**.

22. Select **Next**. **System Builder- Clock Settings** page is displayed, as shown in Figure 16. 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

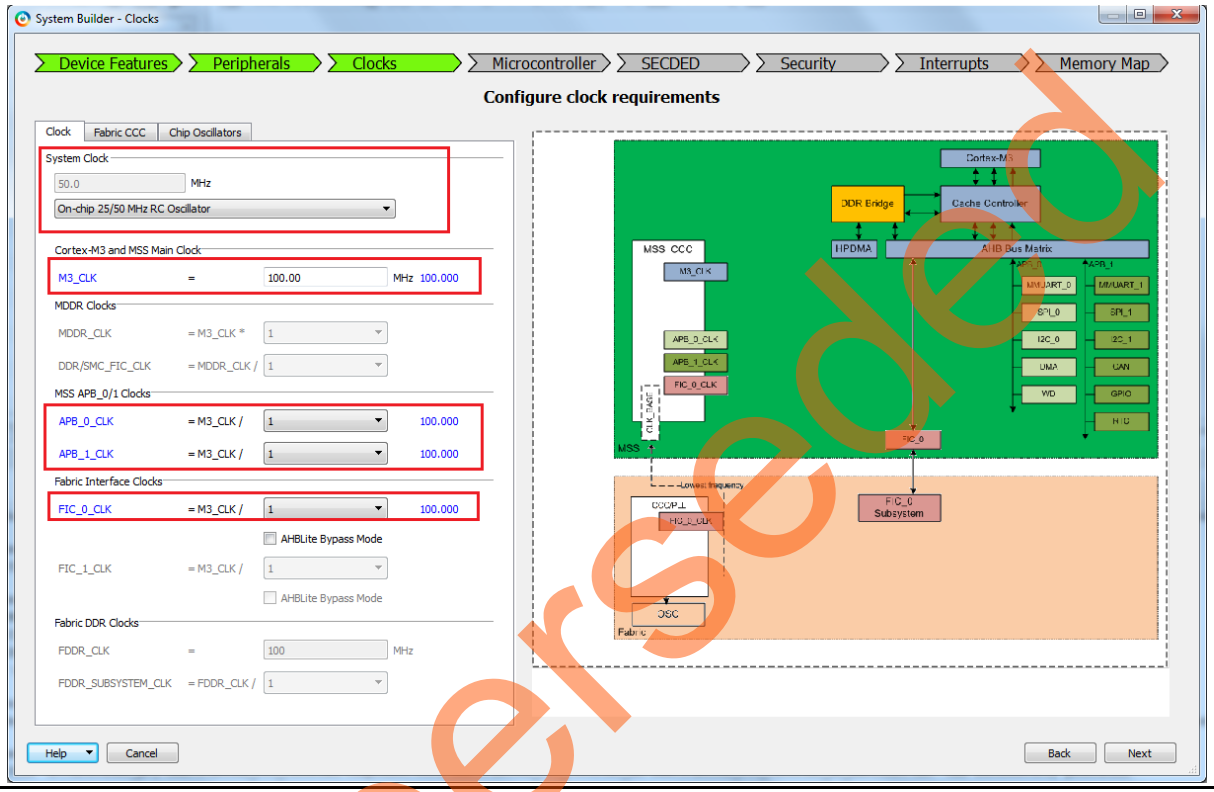

***Figure 16* • SmartFusion2 System Builder Clocks**

23. Click **Next**, the System Builder - **Microcontroller Options** page is displayed.
    - Leave all the Default Selections.
24. Click **Next**, the System Builder - **SECDED Options** page is displayed.
    - Leave all the Default Selections.
25. Click **Next**, the System Builder - **Interrupts Options** page is displayed.
    - Leave all the Default Selections.
26. Click **Next**, the System Builder - **Memory Map Options** page is displayed.
    - Leave all the Default Selections.

Figure 17 shows the address map for AHBL peripherals.

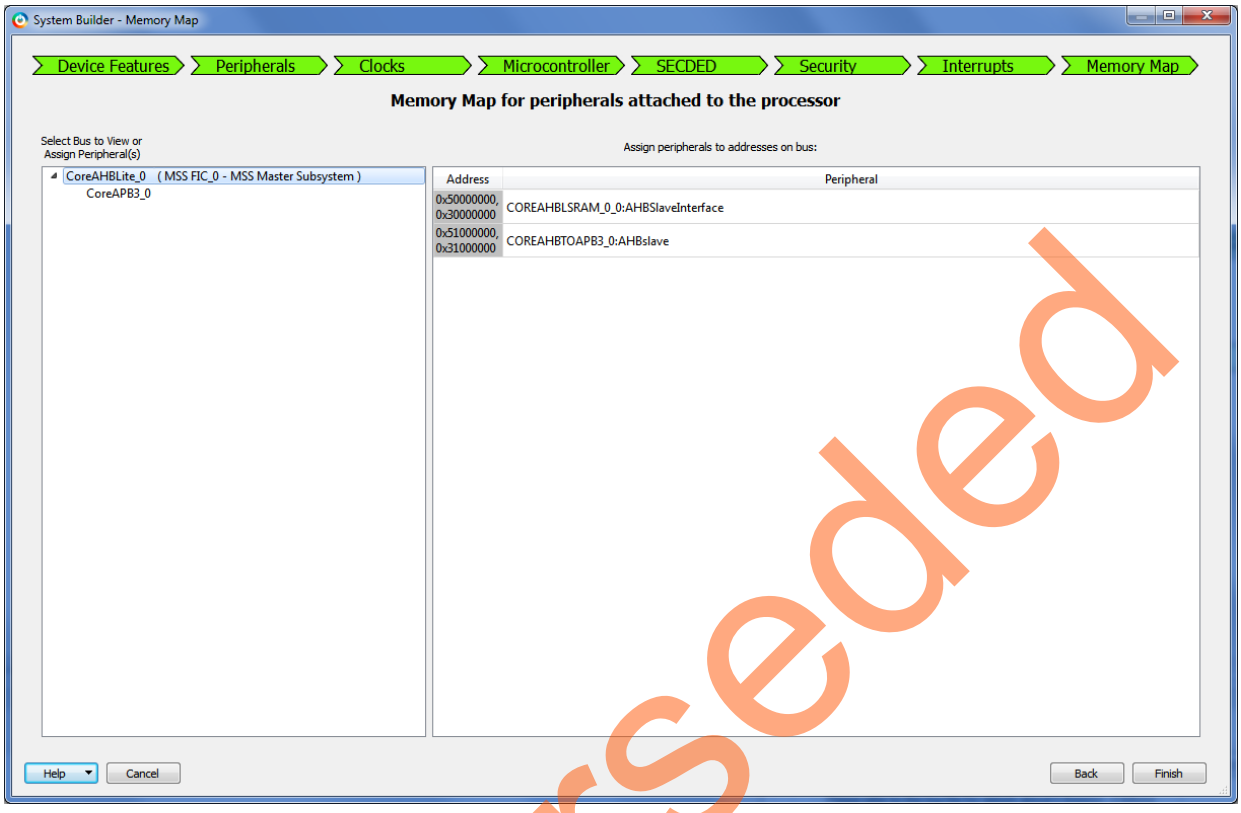

Figure 17 • **SmartFusion2 System Builder CoreAHBLite Address Map (M2S090T Device)**

Figure 18 shows the address map for APB3 peripheral

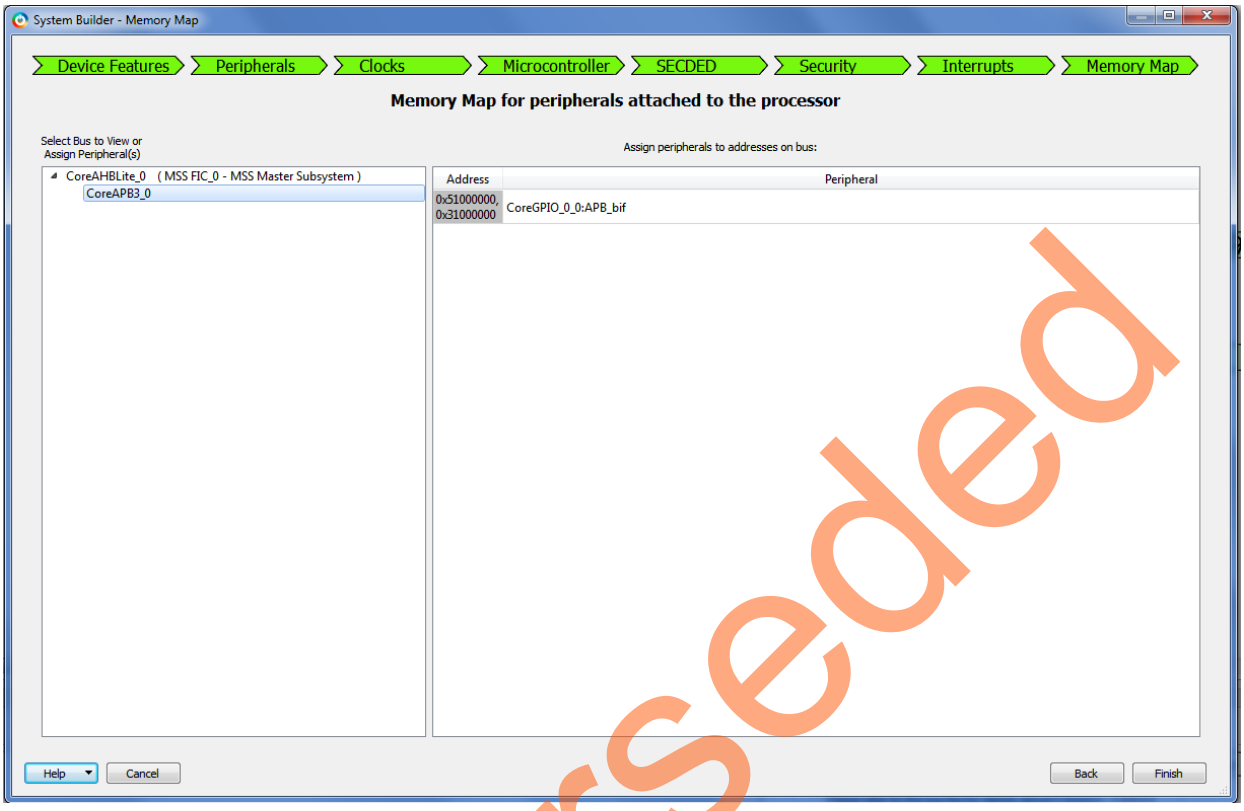

Figure 18 • **SmartFusion2 System Builder CoreAPB Address Map**

27. Click **Finish**.

    The System Builder generates the system based on the selected options.

    The System Builder block is created and added to Libero SoC project, as shown in Figure 19.

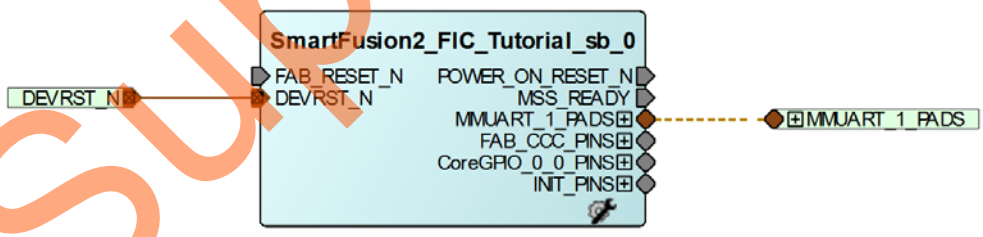

Figure 19 • **SmartFusion2 System Builder Component**

To initialize a user design in the SmartFusion2 devices, Microsemi provides a CoreResetP soft Reset Controller IP. The CoreResetP IP handles a sequence of reset signals in the SmartFusion2 devices. The CoreResetP does automatically be instantiated and connected by the System Builder. Open the System Builder component in the Smart Design canvas to view how these blocks are connected.

28. Connect 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**.
    - 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 connecting the pins, the System Builder block is displayed, as shown in Figure 20.

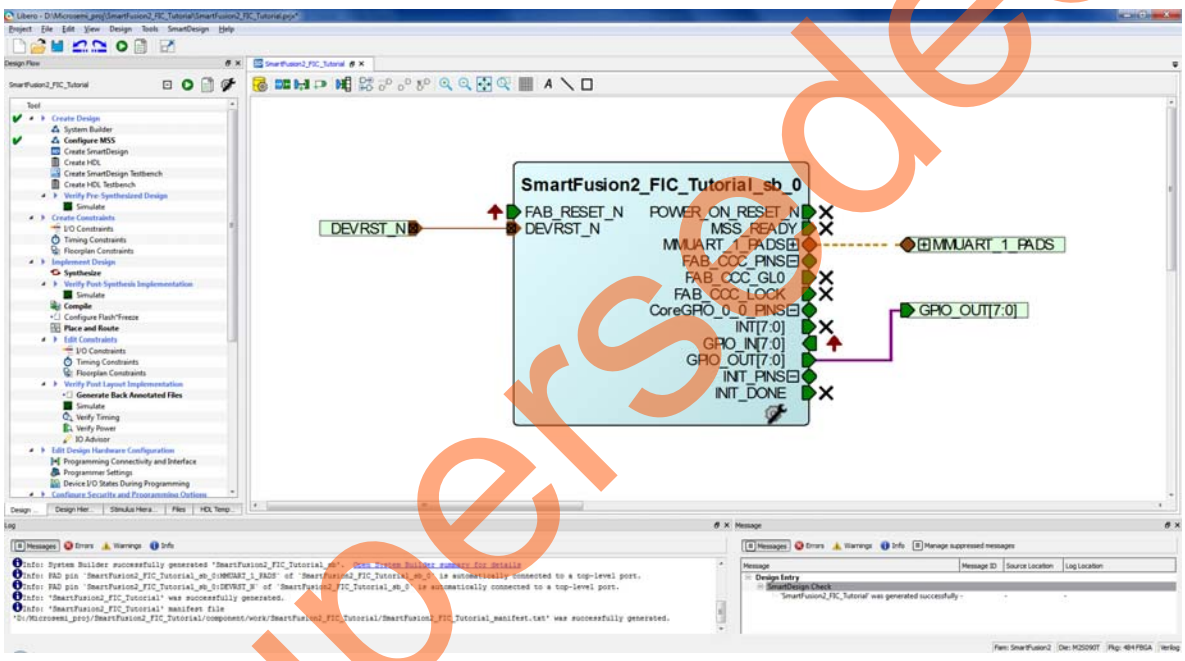

***Figure 20* • SmartFusion2 System Builder Block**

Note: The System Builder block for SmartFusion2 Starter Kit is shown in Figure 21.

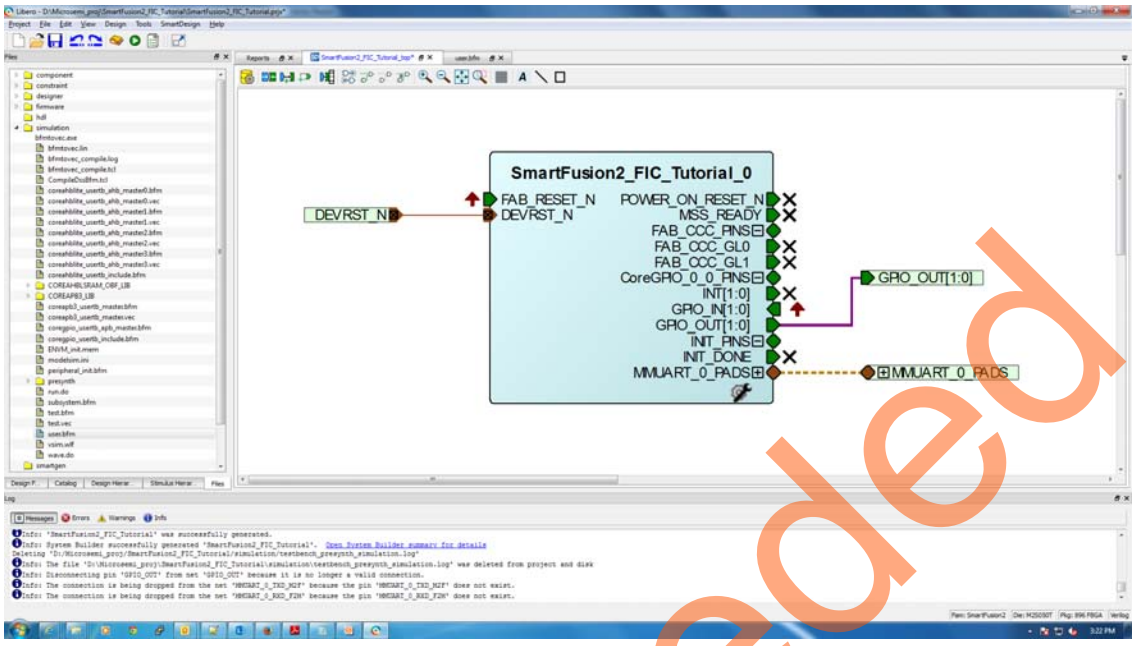

***Figure 21* • SmartFusion2 System Builder Block**

29. Click **Generate Component** icon on the SmartDesign toolbar or right-click on the canvas and select **Generate Component**.

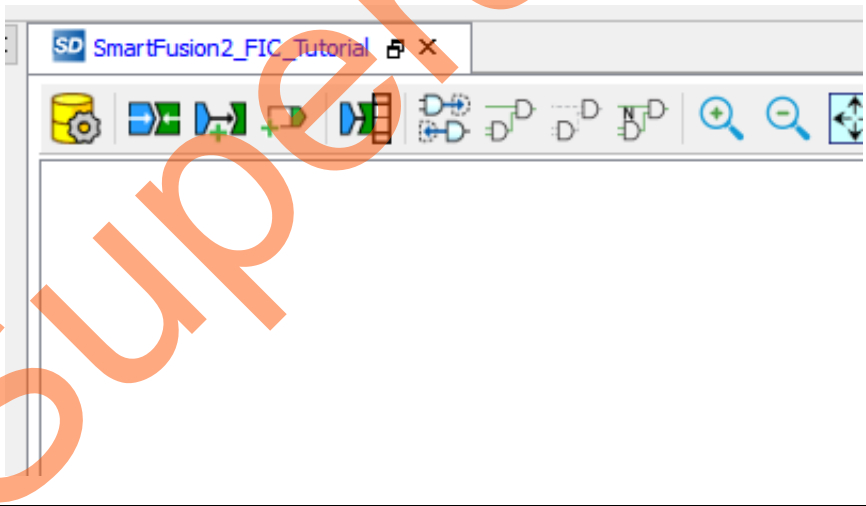

***Figure 22* • Generate Component**

After successful generation of the system, the message **SmartFusion2_FIC_Tutorial was successfully generated** is displayed in the Libero SoC log window if the design is generated without any errors. The log window is displayed, as shown in Figure 23.

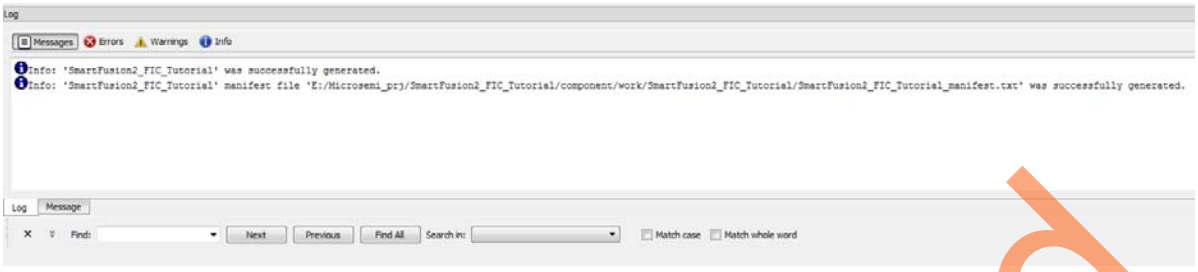

***Figure 23* • Log Window**

Step 2: Modifying User BFM Script for Simulation

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 "Design Files" section on page 4.

1. For SmartFusion2 Security Evaluation 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 (*\SF2_FIC_Tutorial\Source\For_SF2_Eval_Kit_Board\user.bfm*) as shown in Figure 24 or select **Files > Import > Others** to import the `user.bfm` file.
2. Click **Yes to all** to replace the existing `user.bfm` file.

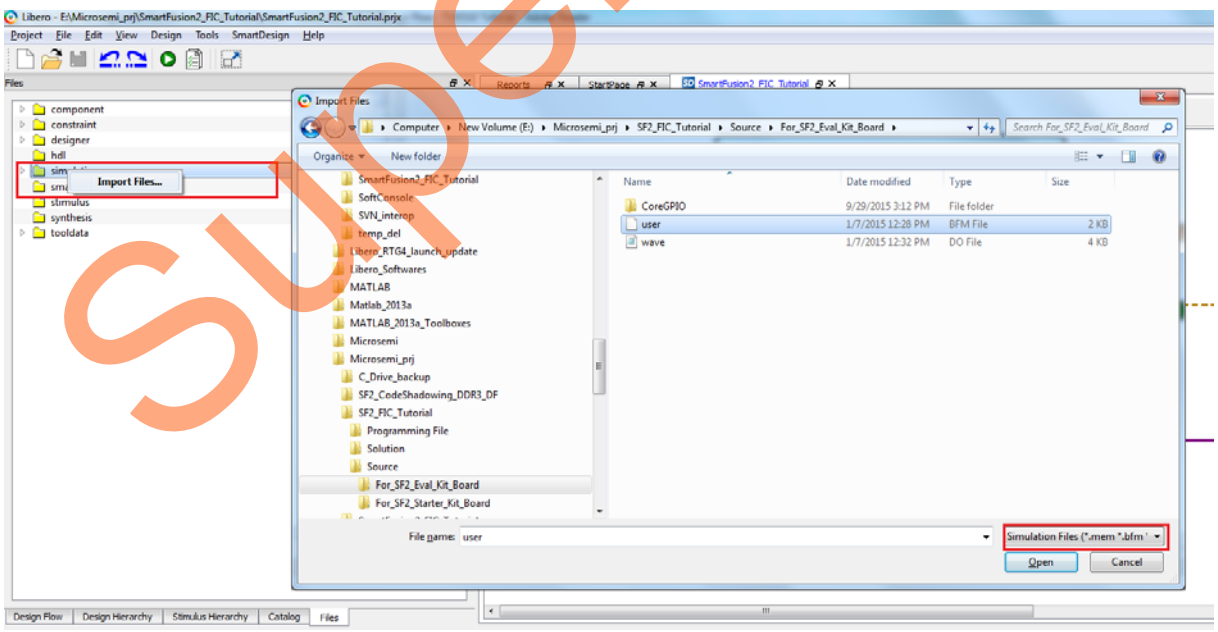

***Figure 24* • Import bfm file**

Note: For SmartFusion2 Starter Kit, import the `user.bfm` to simulation files from design files. (*\SF2_FIC_Tutorial\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 25.

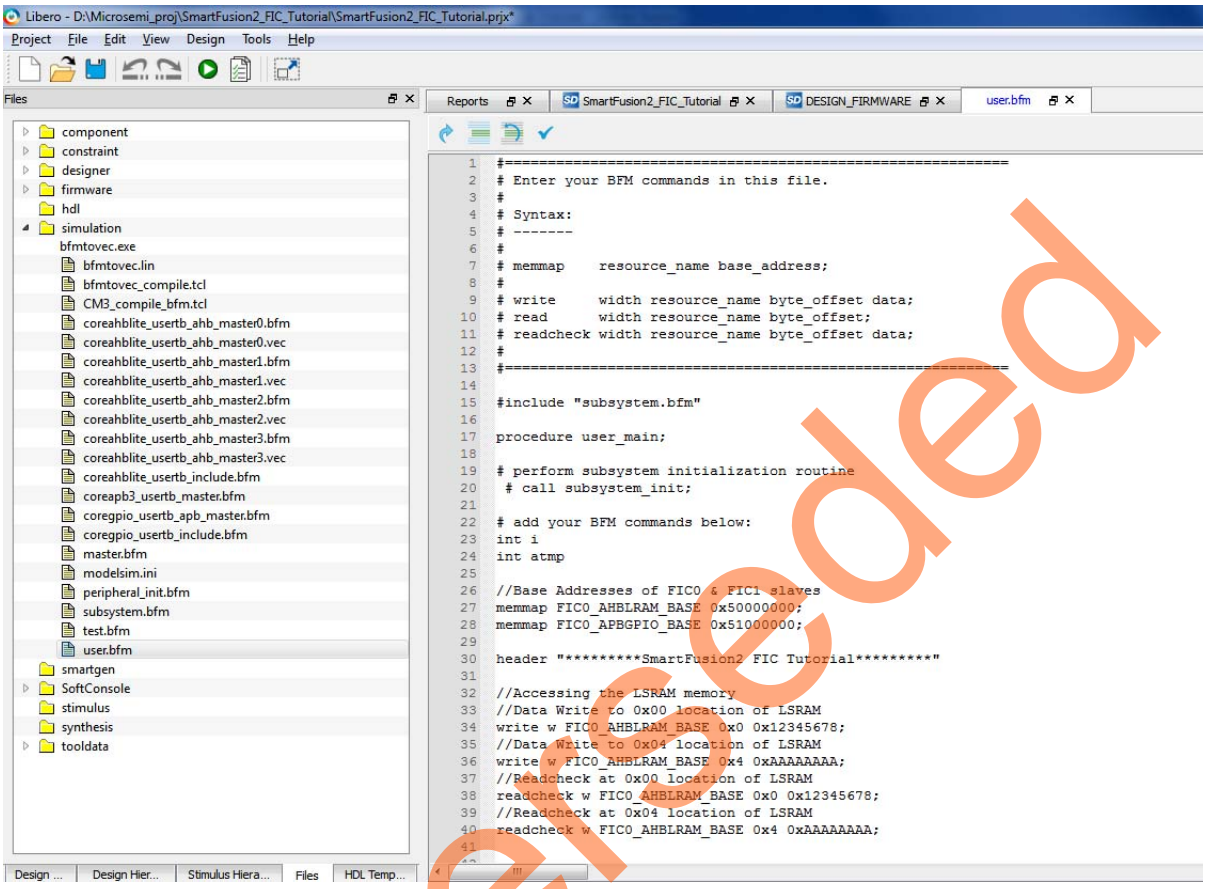

Figure 25 • **user.bfm file**

Generating Testbench

1. From the **File** menu, select **New** > **HDL Testbench** as shown in Figure 26.

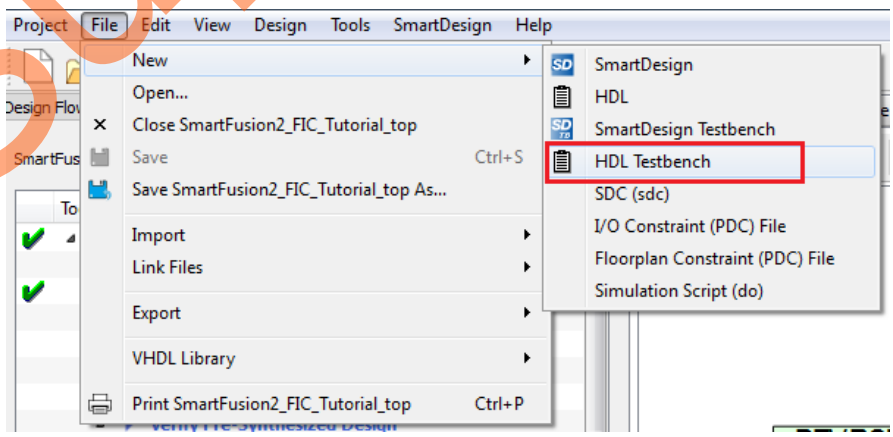

Figure 26 • **HDL Testbench**

The **Create New HDL Testbench File** dialog box is displayed.

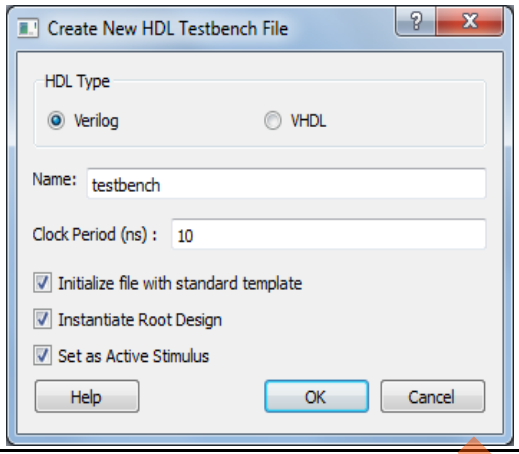

Figure 27 • **Create New HDL Testbench File**

2. Select **HDL Type** as **Verilog** or **VHDL**.
3. Enter **Name** as testbench in the text box and retain the default settings.
4. Enter **Clock Period (ns)** as 10.
5. Click **OK**.

Step 3: Simulating Design Using BFM Models

This section describes how to use the 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: *SF2_FIC_Tutorial\Source\For_SF2_Eval_Kit_Board*. Figure 28 shows the `wave.do` file under simulation folder in the Files window.

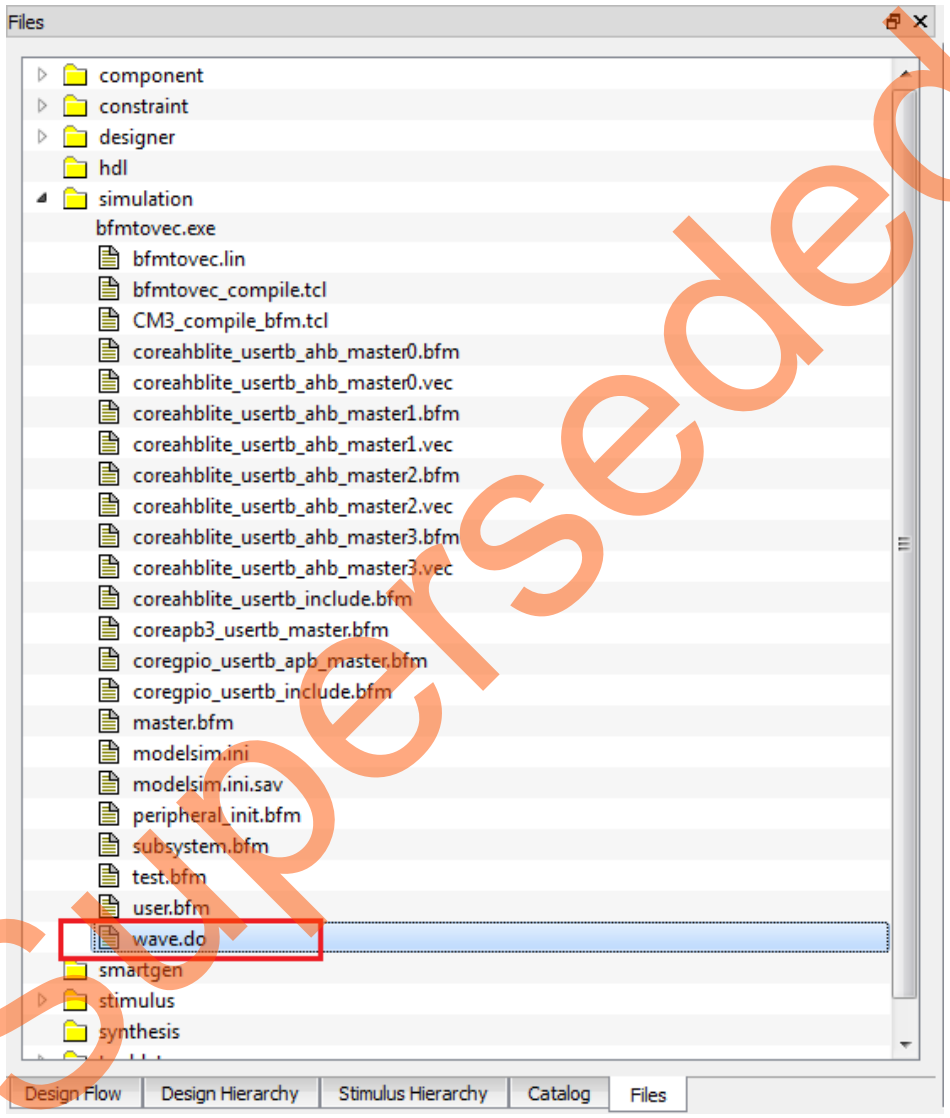

***Figure 28* • wave-do File**

Note: For SmartFusion2 Starter Kit board, browse to the `wave.do` file located in the downloaded design files folder: SF2_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 29.

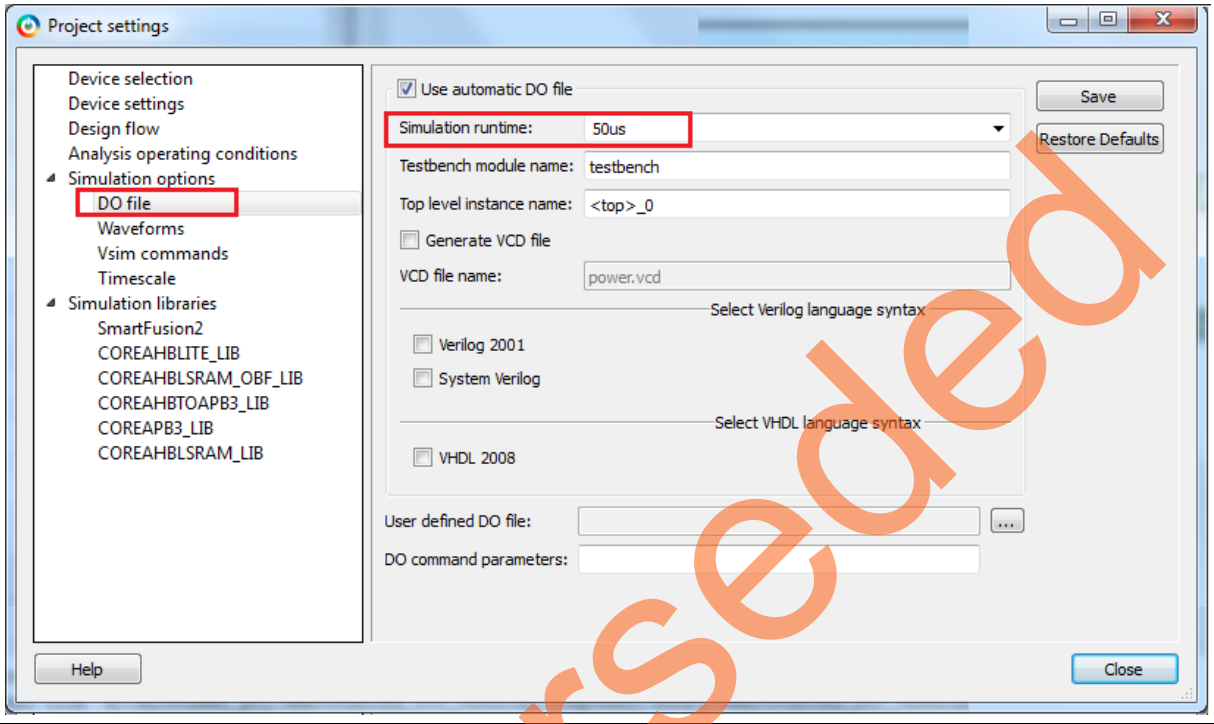

***Figure 29* • Project Settings – Do File**

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

5. Select **Waveforms** under **Simulation Options** as shown in Figure 30:
   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.

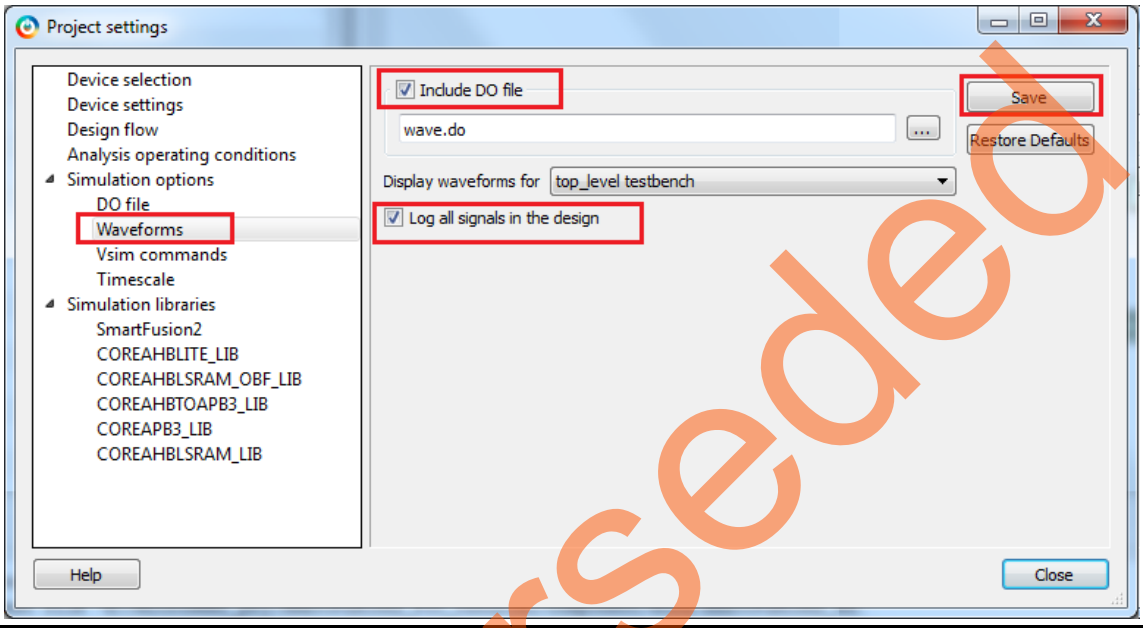

***Figure 30* • Project Settings – Waveforms**

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

7. Expand the **Verify Pre-Synthesized Design**, as shown in Figure 31. Double-click **Simulate** to invoke ModelSim. After invoking ModelSim, the design is loaded. Alternatively, right-click the **Simulate** and select **Open Interactively**.

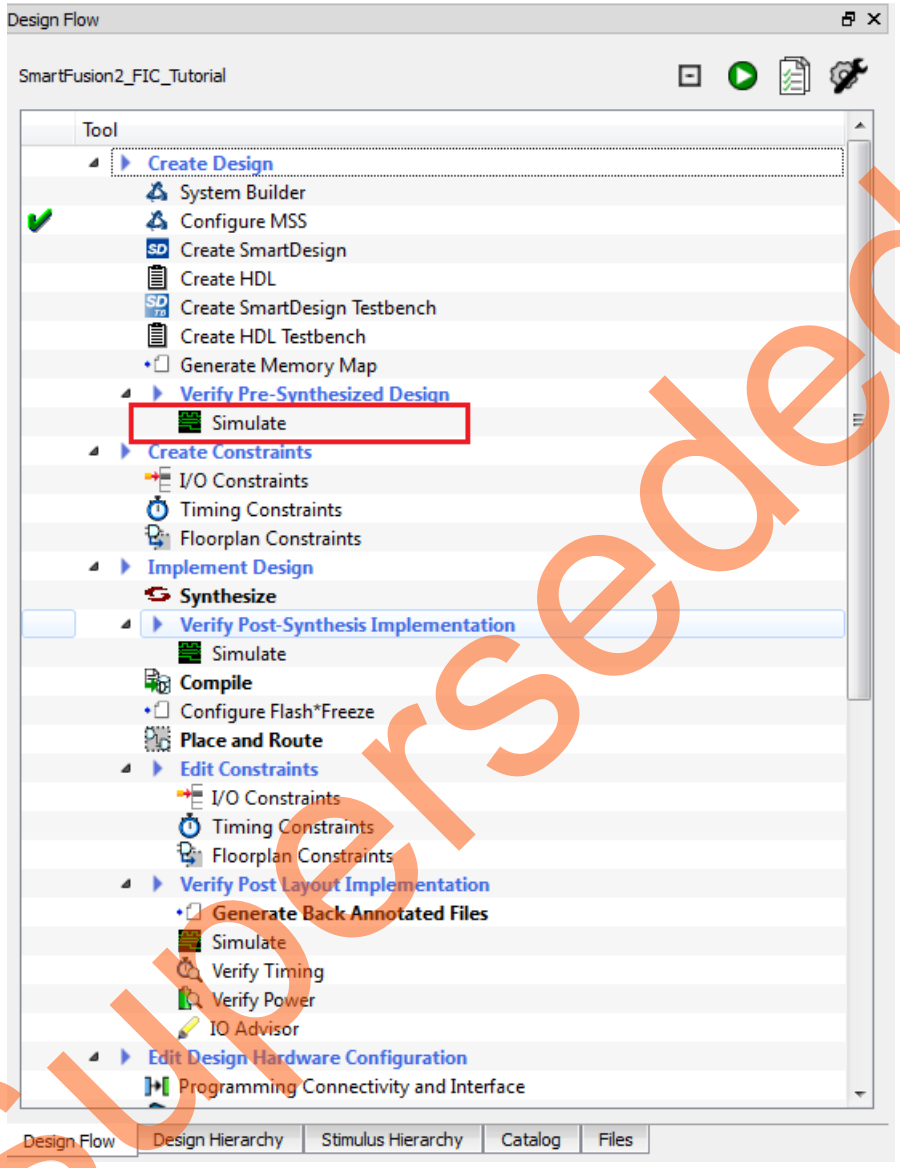

Figure 31 • **Design Flow – Verify Pre-Synthesized Design**

8. Maximize the **ModelSim Transcript** window to see the BFM commands execution. Ensure that there are no errors. Figure 32 shows the ModelSim Transcript window.

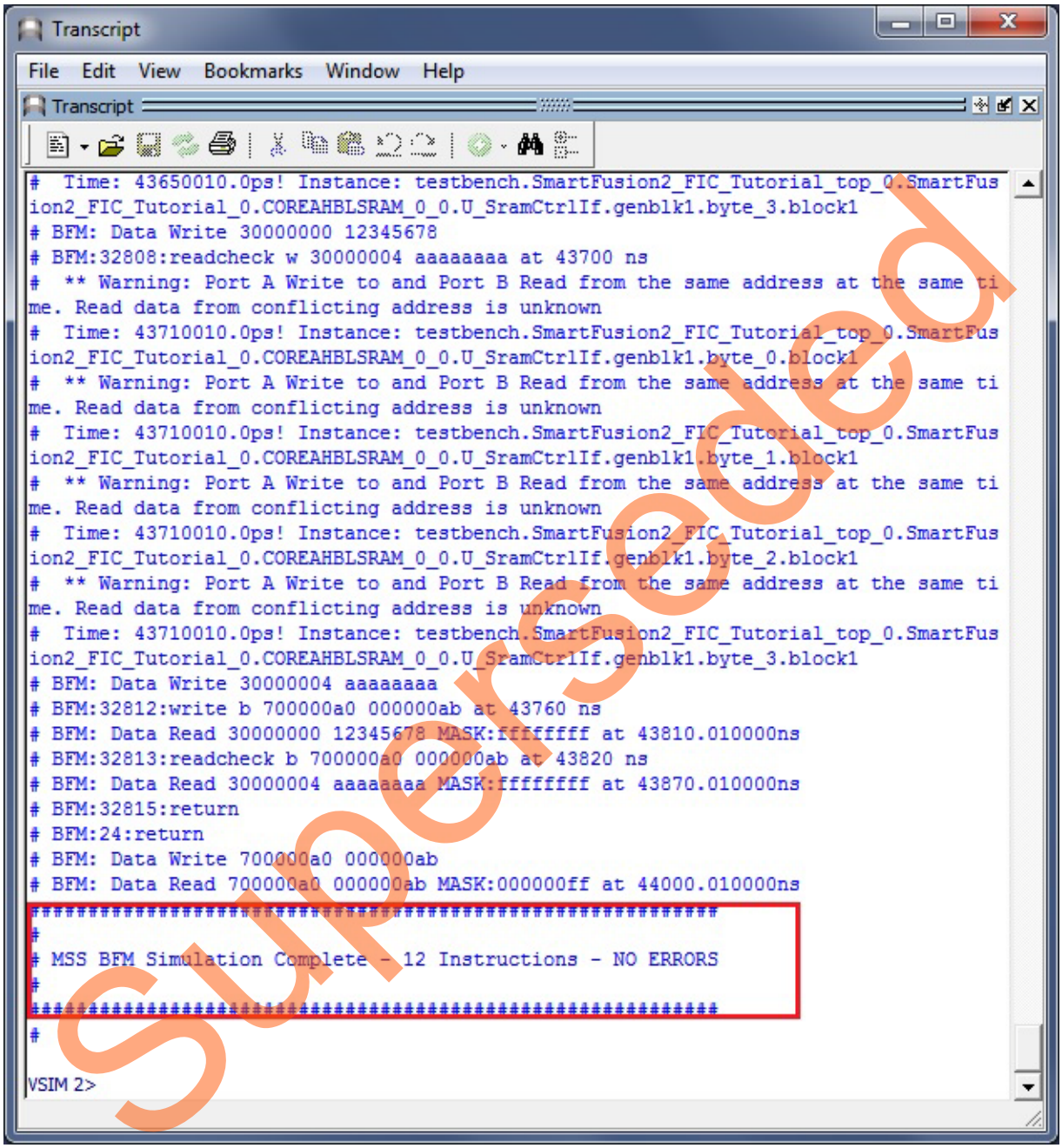

***Figure 32* • ModelSim Transcript Window – BFM Commands**

9. After successful BFM simulation, observe the ModelSim waveform window for the read and write bus transactions to the fabric peripherals, as shown in Figure 33. Notice the result of GPIO configuration BFM commands in GPIO states.

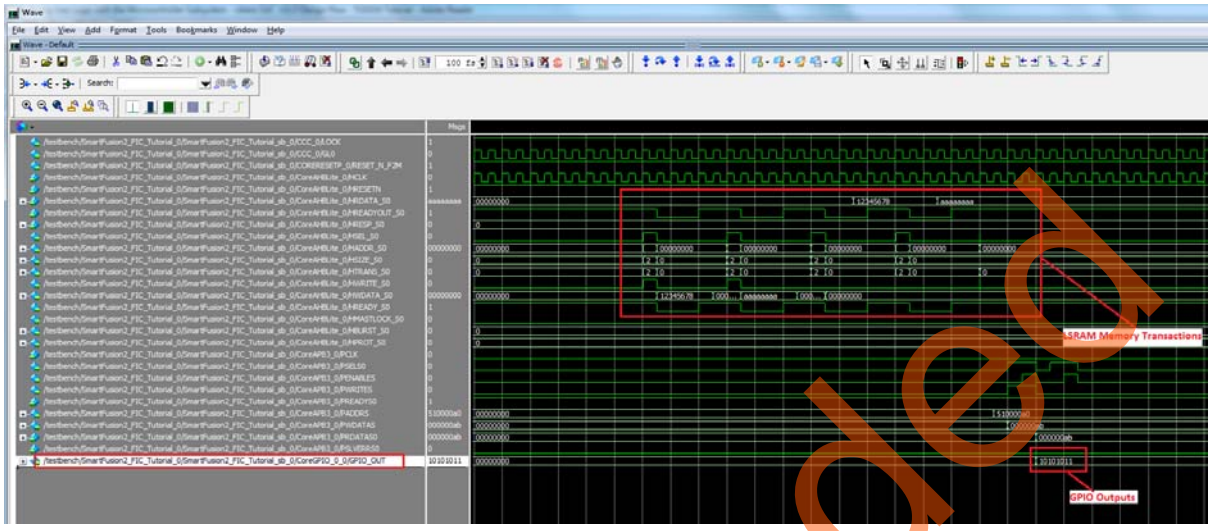

***Figure 33* • Design Simulation Waveforms**

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

Step 4: Generating Programming File

The following steps describe how to generate a programme file:

1. Double-click **Edit Constraints > I/O Constraints** in the **Design Flow** window as shown in Figure 34. The **I/O Editor** window is displayed after completing Synthesize and Compile.

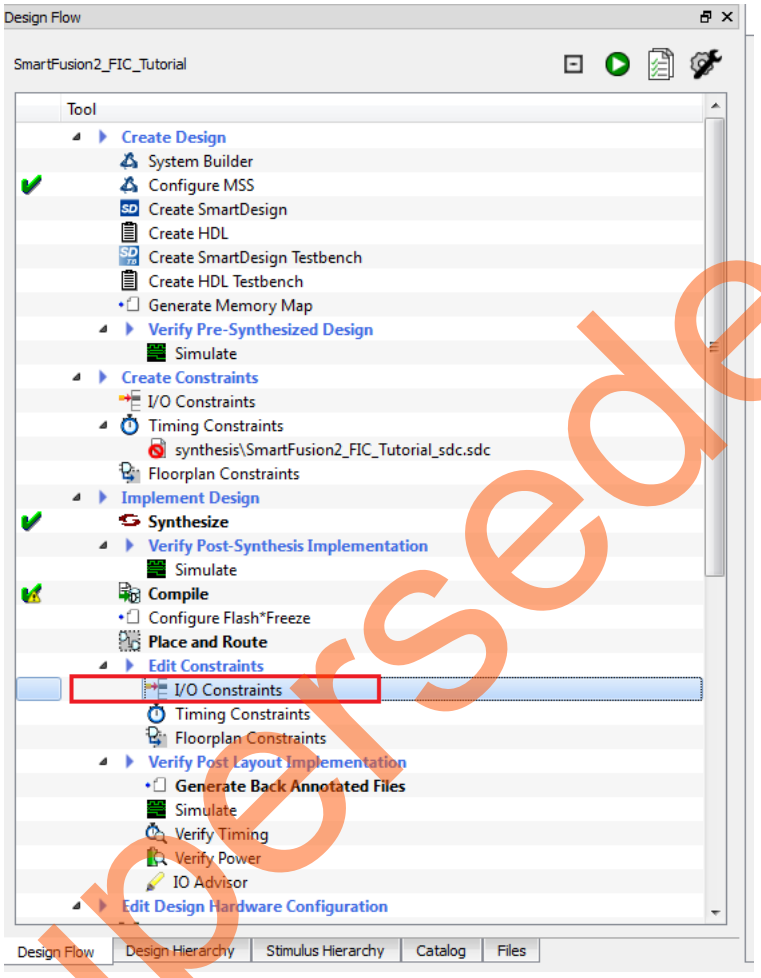

Figure 34 • **I/O Constraints**

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]	E1
GPIO_OUT[1]	F4
GPIO_OUT[2]	F3
GPIO_OUT[3]	G7
GPIO_OUT[4]	H7
GPIO_OUT[5]	J6
GPIO_OUT[6]	H6
GPIO_OUT[7]	H5

After assigning the pins, the **I/O Editor** is displayed, as shown in Figure 35.

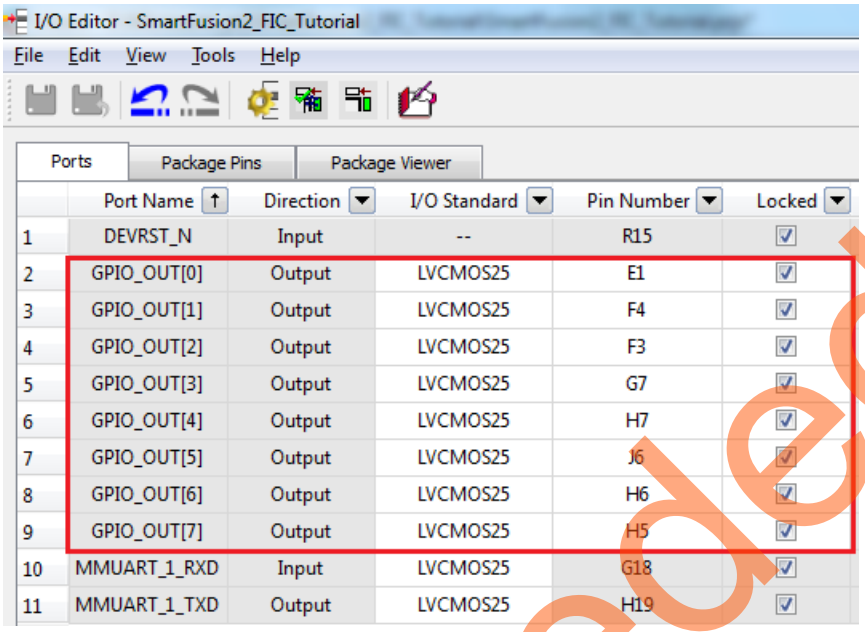

Figure 35 • **I/O Editor**

Note: Table 3 shows the Pin assignments for the SmartFusion2 Starter Kit board.

Table 3 • **Port to Pin Mapping**

Pin Name	Pin Number
GPIO_OUT[0]	AB18
GPIO_OUT[1]	P1

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

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

3. Click **Commit and Check** after Updating the I/O editor.
4. Close the **I/O Editor** window.
5. Click **Generate Bitstream** as shown in Figure 36 to complete place-and-route, verify timing, and generating the programming file.

Figure 36 • **Generate Bitstream**

Step 5: Programming the SmartFusion2 Board Using Flash Pro

Before programming the 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 *Starter Kit Guide* and *UG0594: SmartFusion2 Security Evaluation Kit User Guide* for additional information.

Jumper Settings for SmartFusion2 Security Evaluation Kit Board

Connect the jumpers on the SmartFusion2 Security Evaluation Kit, as shown in Table 4. Switch OFF the power supply switch while connecting the jumper.

Table 4 • **Jumper Settings for Security Evaluation Kit Board**

Jumper	Pin (from)	Pin (to)
J3, J8	1 (default)	2

Jumper Settings for SmartFusion2 Starter Kit Board

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

Table 5 • **Jumper Settings for Starter 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 37 to program the SmartFusion2 SoC FPGA device.

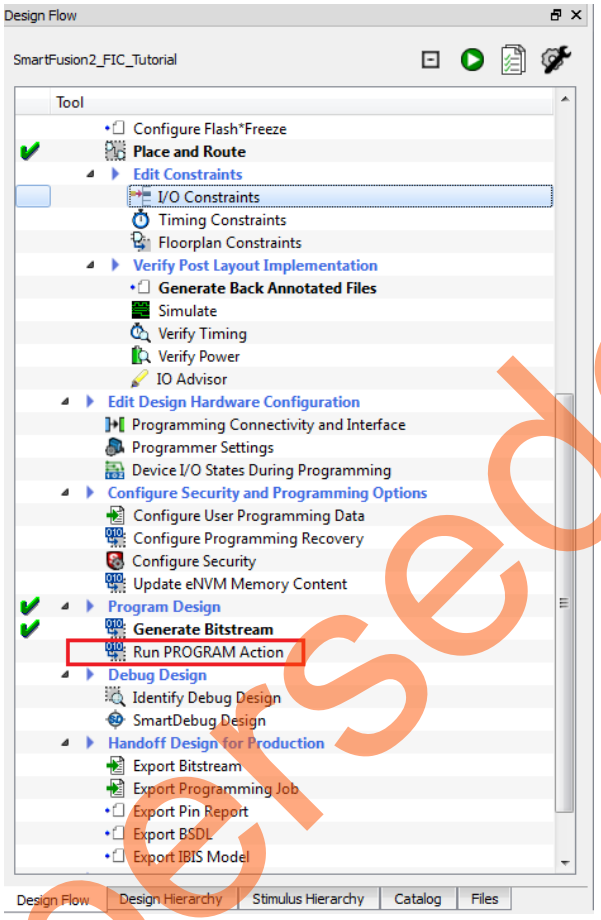

***Figure 37* • Run PROGRAM Action**

Figure 38 shows the board setup for running the application design on the SmartFusion2 Security Evaluation Kit board.

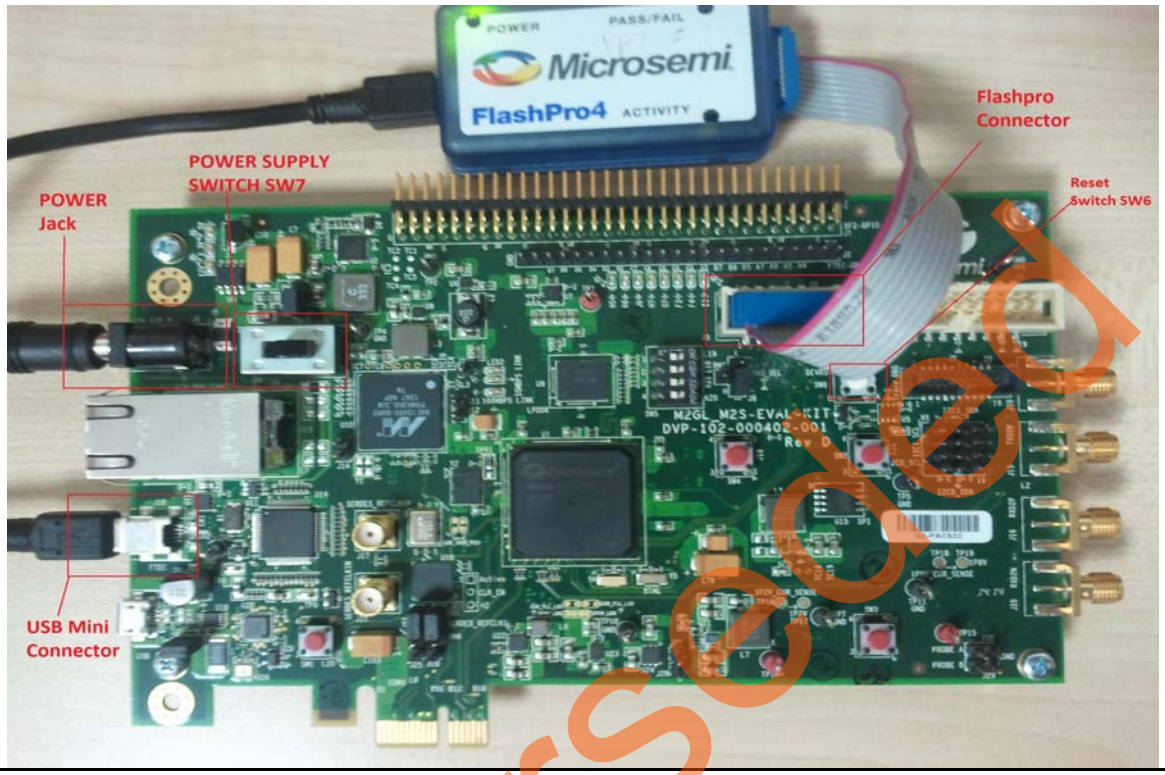

***Figure 38* • SmartFusion2 Security Evaluation Kit Setup**

Figure 39 shows the board setup for running the application design on the SmartFusion2 Starter Kit board.

***Figure 39* • SmartFusion2 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

The following steps describe how to build the software application via SoftConsole:

1. Click **Handoff Design for Firmware Development** > **Export Firmware** in the **Design Flow** window.
2. Right-click and select **Export Firmware…** as shown in Figure 40. The Export Firmware dialog box is displayed.

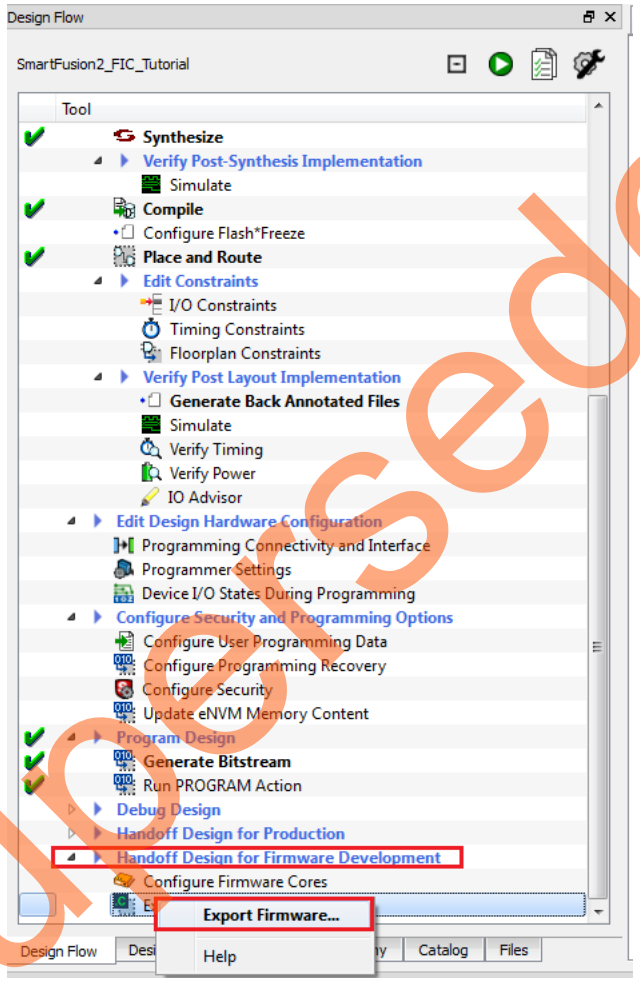

Figure 40 • **Handoff Design for Firmware Development**

3. Enter the following information in the **Export Firmware** dialog box.

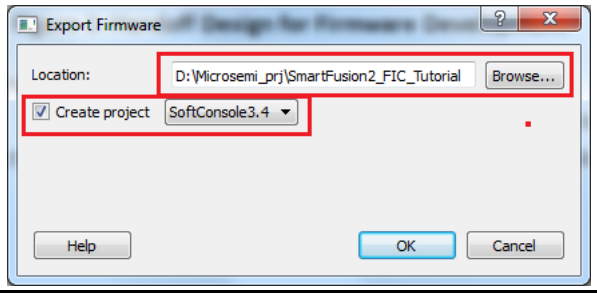

***Figure 41* • Export Firmware**

   – Browse to the **Location** such as <*C:\ or D:\Microsemi_prj\SmartFusion2_FIC_Tutorial*>.
   – Select the **Create project** check box and select **SoftConsole3.4** from the drop-down list.
4. Click **OK**. The **Information** dialog box is displayed as in Figure 42.

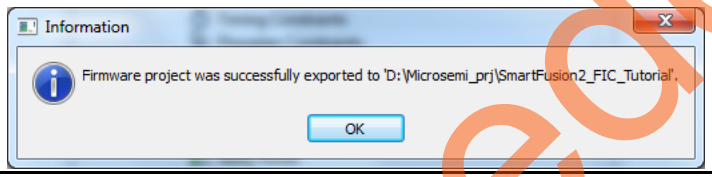

***Figure 42* • Export Firmware-Information**

5. Click **OK**.
6. Click **Start** > **Programs** > **Microsemi SoftConsole v3.4** > **Microsemi SoftConsole IDE v3.4** or double-click the shortcut icon on your desktop. The **SoftConsole Workspace Launcher** is displayed as shown in Figure 43.

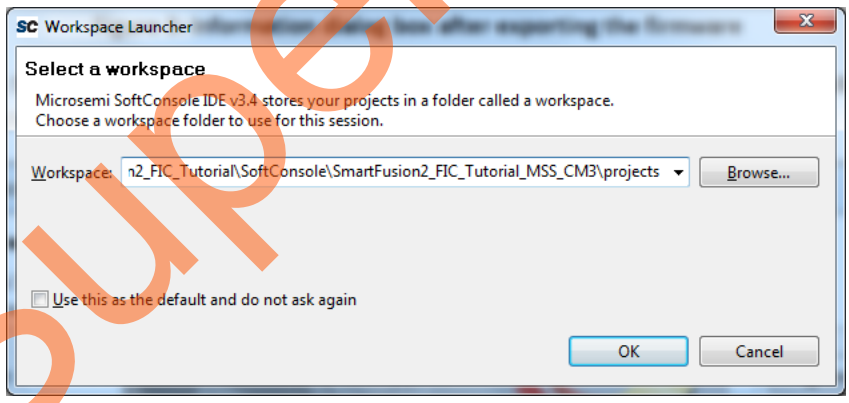

***Figure 43* • SoftConsole Workspace Launcher**

7. Navigate to the **SoftConsole folder** and select **projects** folder as shown in Figure 44.

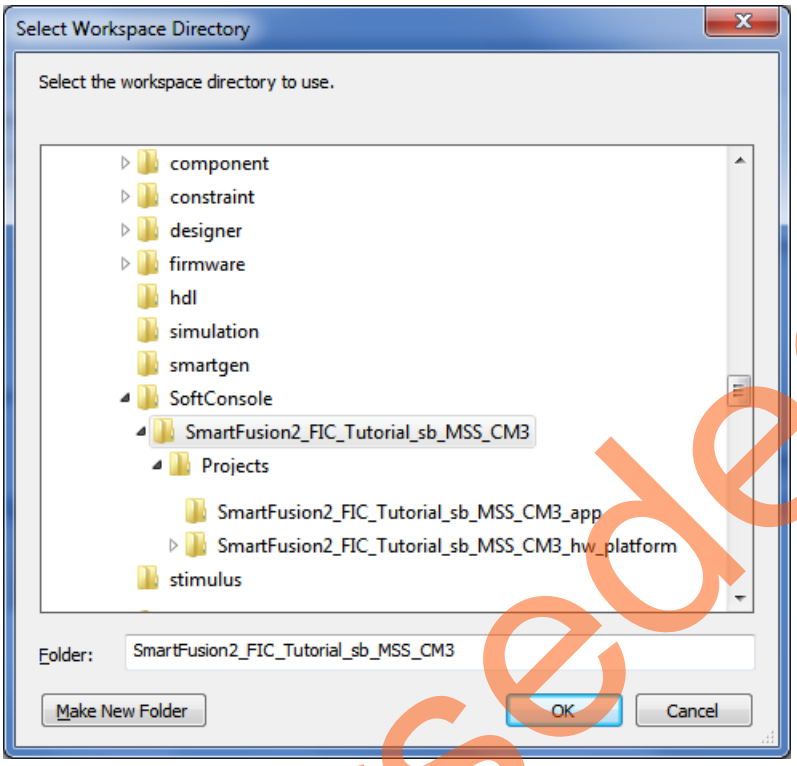

***Figure 44* • Select Workspace Directory**

8. Click **OK**.

The **SoftConsole IDE** window is displayed as shown in Figure 45.

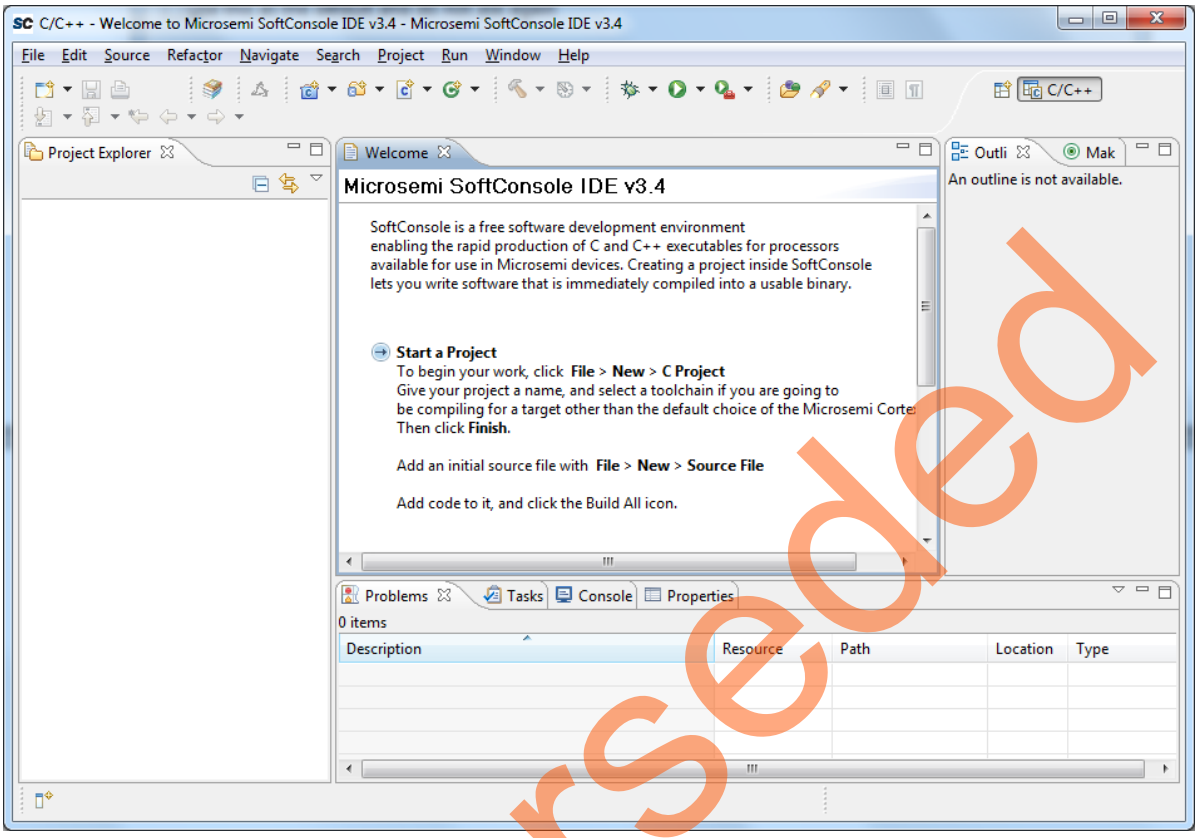

Figure 45 **• SoftConsole IDE**

9. Import the existing project into workspace as shown in Figure 46.

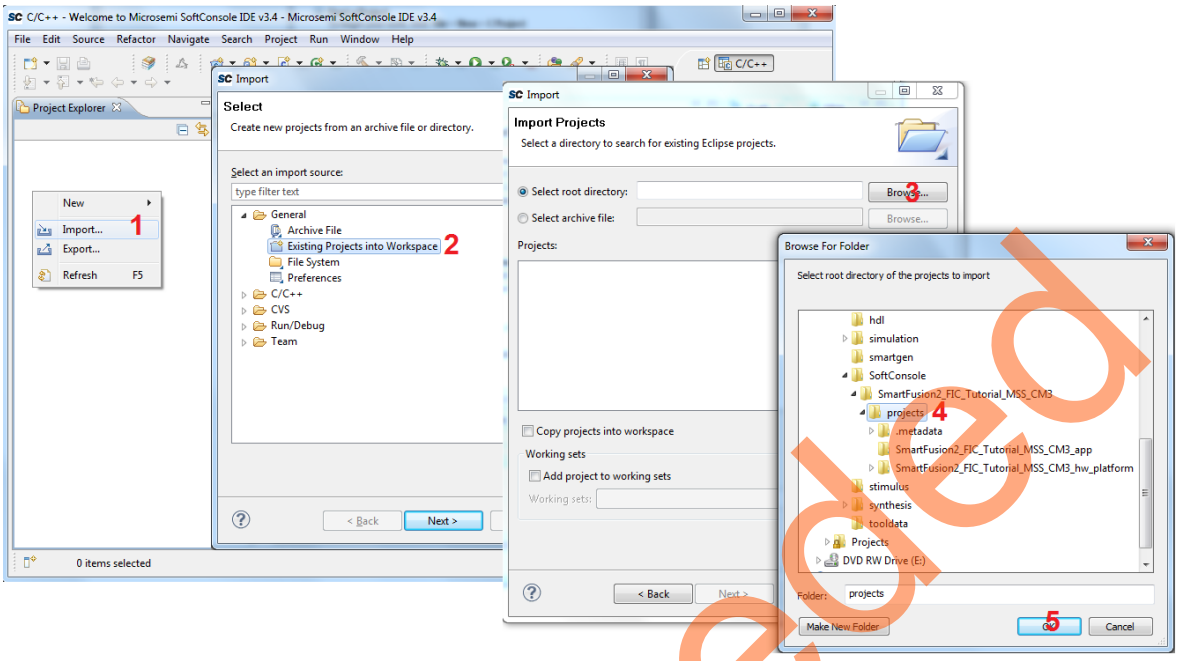

***Figure 46* • Importing the Existing Project into Workspace**

a. Right-click the **Project Explorer** tab and select **Import…**. The **Import** dialog box is displayed.

b. Select **Existing Project into Workspace** under **General** folder and click **Next**. The **Import Projects** dialog box is displayed.

c. Click **Browse** to navigate to the existing project. The **Browse for Folder** dialog box is displayed.

d. Navigate to the **SoftConsole** folder and select **projects** folder as shown in Figure 46.

e. Click **OK** and then click **Finish**.

The SoftConsole Perspective displays similar to Figure 47.

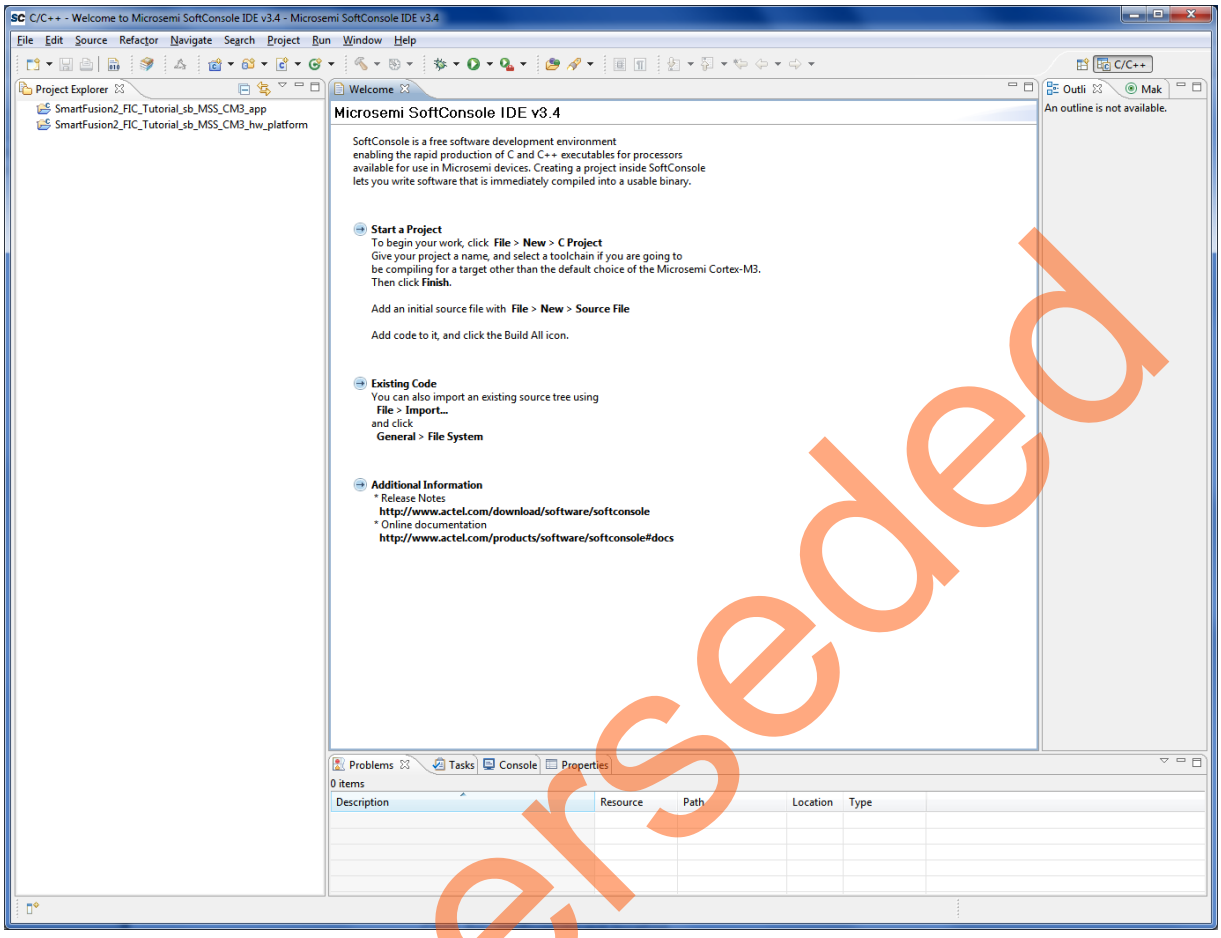

***Figure 47* • SoftConsole Perspective**

10. Go to the source folder in the downloaded design files folder, copy the code from the `Source_eval.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.

11. Copy the CoreGPIO folder from *\SF2_FIC_Tutorial\Source\For_SF2_Eval_Kit_Board* and paste it in *\SoftConsole\SmartFusion2_FIC_Tutorial_MSS_CM3\projects\SmartFusion2_FIC_Tutorial_MSS_CM3_hw_platform\drivers* location.

12. Select **Project > Clean** to perform a clean build. Accept the default settings in the **clean** dialog box and click **OK**, as shown in Figure 48.

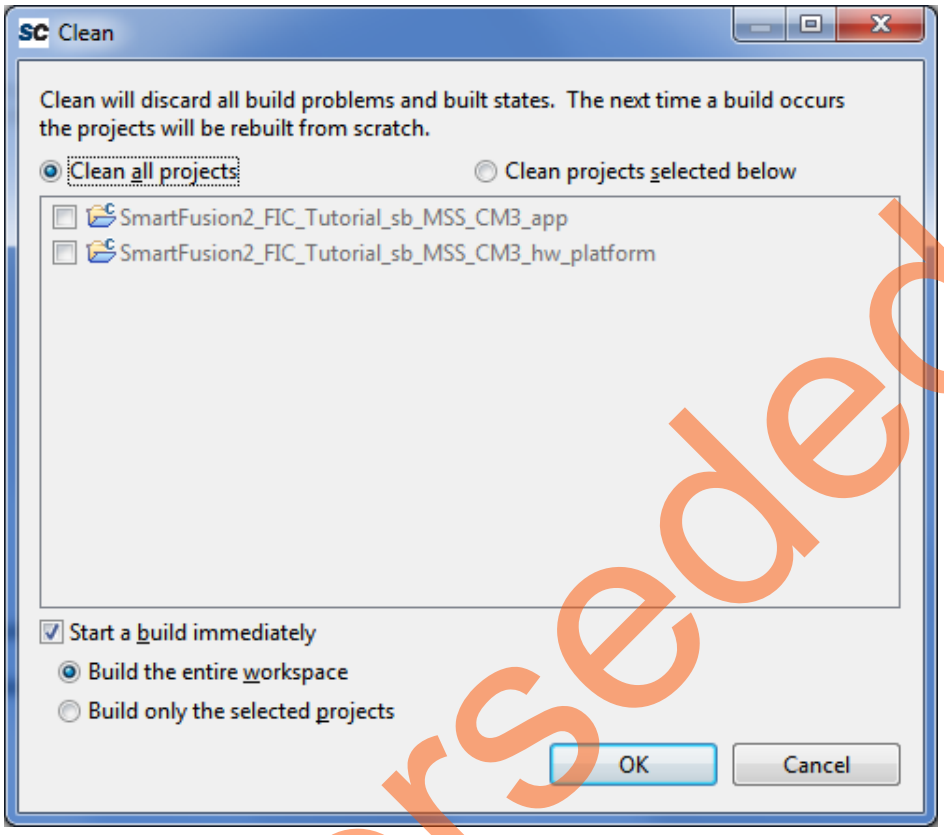

***Figure 48* • Clean and Build window**

13. Ensure that there are no errors.

Step 7: Configuring the Serial Terminal Emulation Program

Prior to running the application program, configure the terminal emulator program on your PC. Perform the following steps to use the SmartFusion2 Security Evaluation 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. Ensure that the USB to UART bridge drivers are automatically detected, as shown in Figure 49 and Figure 50 on page 44.

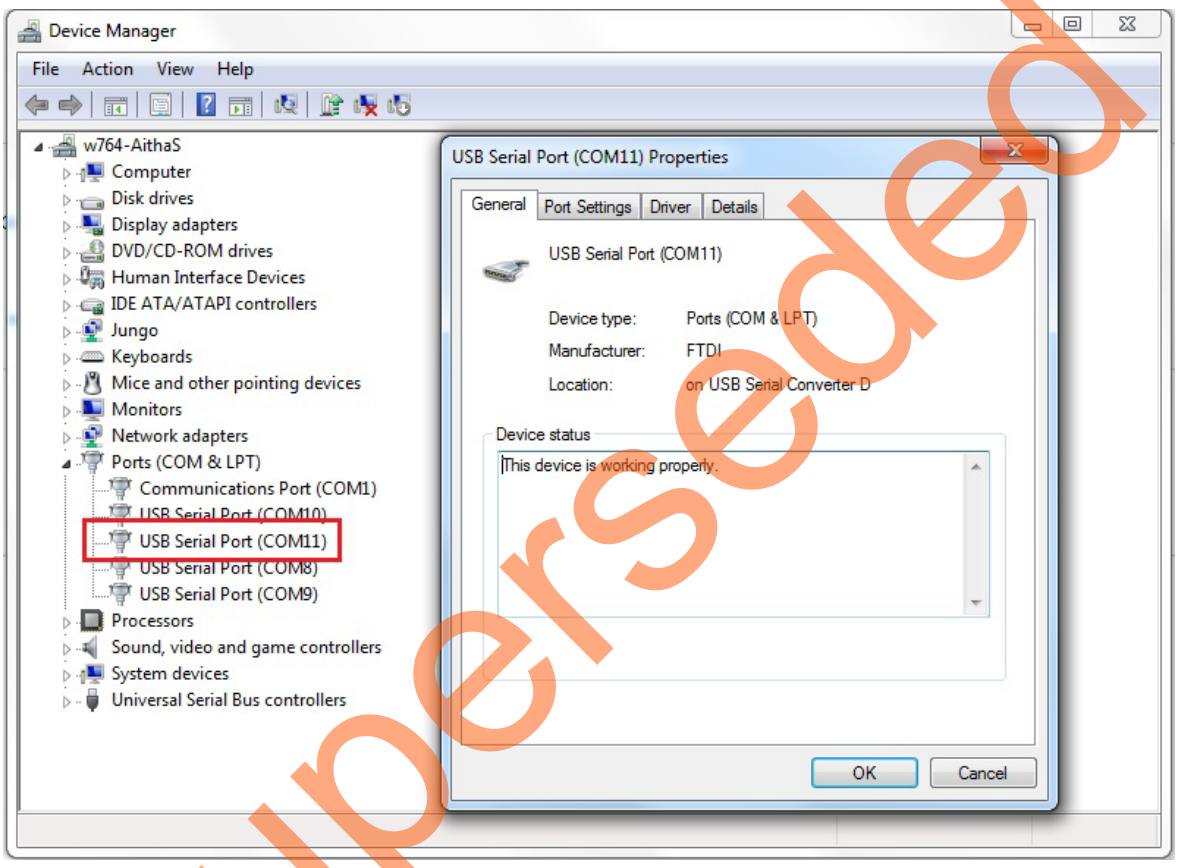

Figure 49 • **SmartFusion2 Security Evaluation Kit USB Serial Port Drivers**

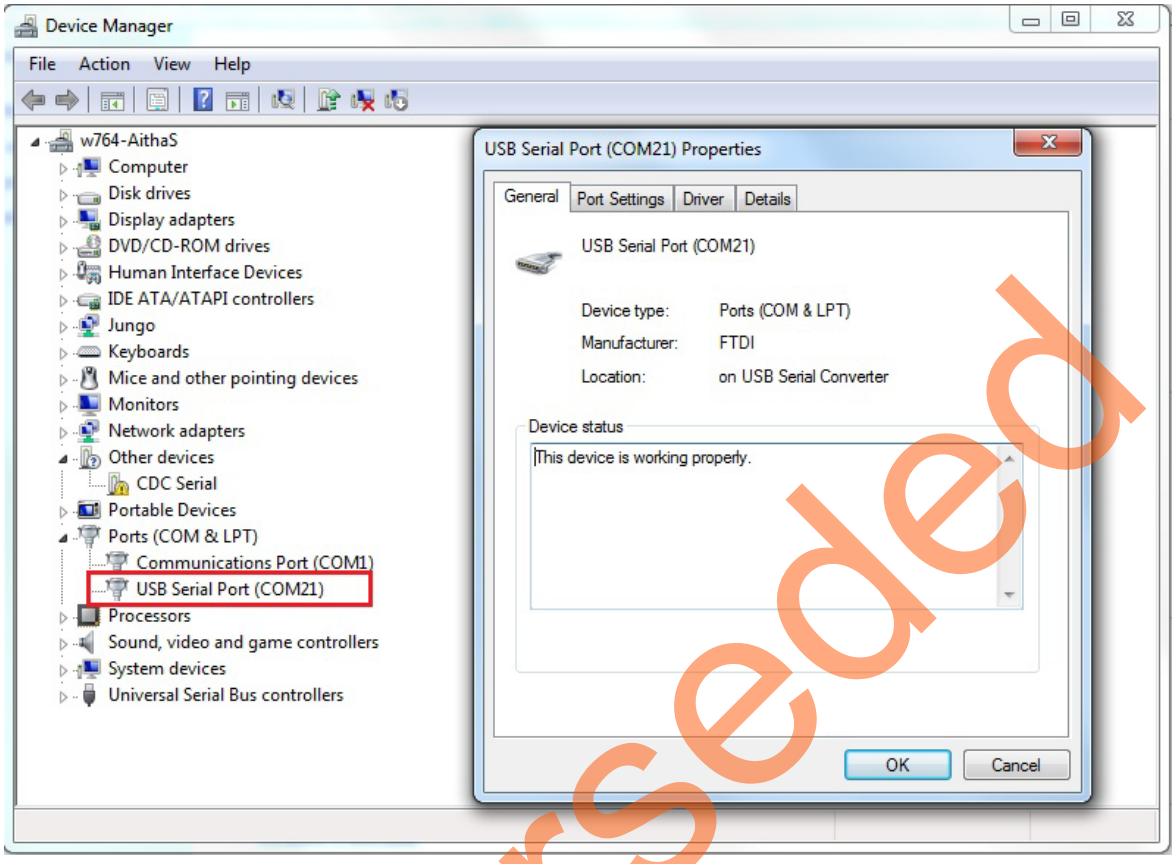

***Figure 50* • SmartFusion2 Starter Kit USB Serial Port Drivers**

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 terminal emulator program with the baud rate set to 57600, 8 data bits, 1 stop bit, no parity, and no flow control. 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_sb_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 51.

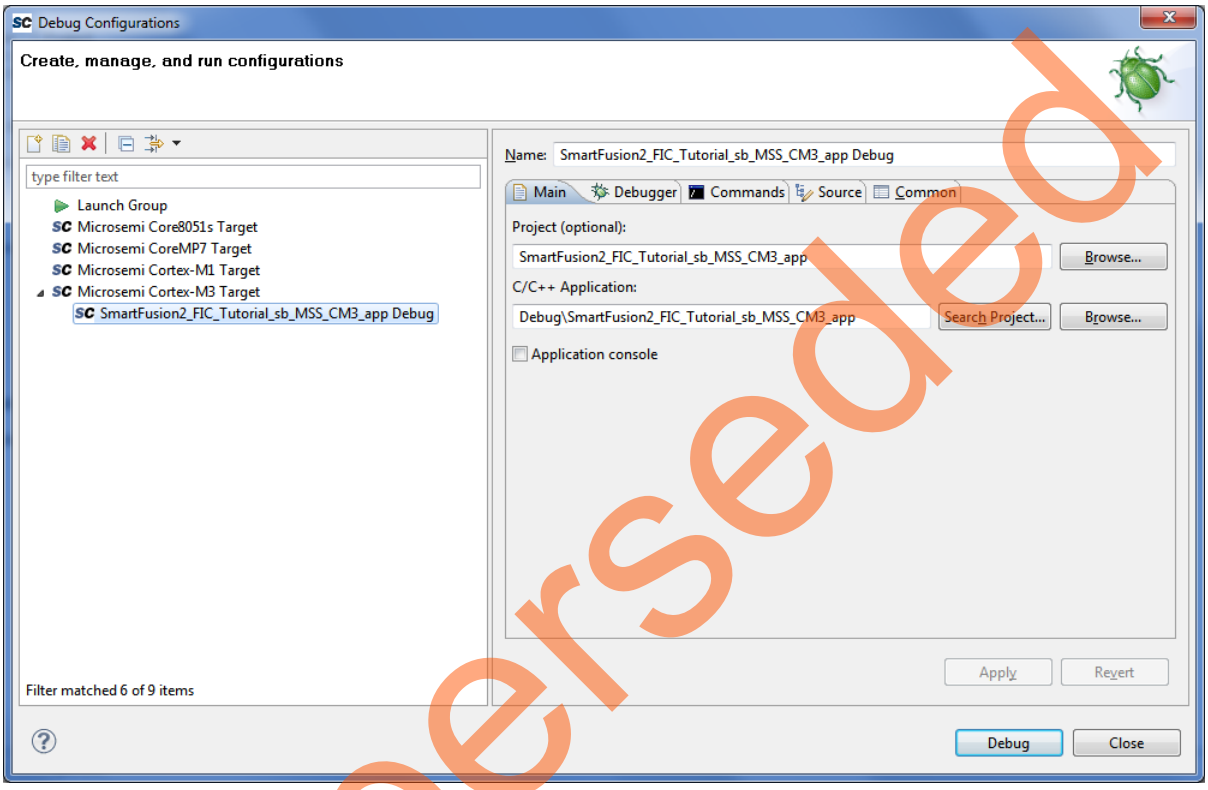

***Figure 51* • Debug Window**

4. Confirm that the following appear on the Main tab in the Debug window:
   - **Name**: SmartFusion2_FIC_Tutorial_sb_MSS_CM3_app Debug
   - **Project**: SmartFusion2_FIC_Tutorial_sb_MSS_CM3_app
   - **C/C++ Application**: DebugSmartFusion2_FIC_Tutorial_sb_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 52.

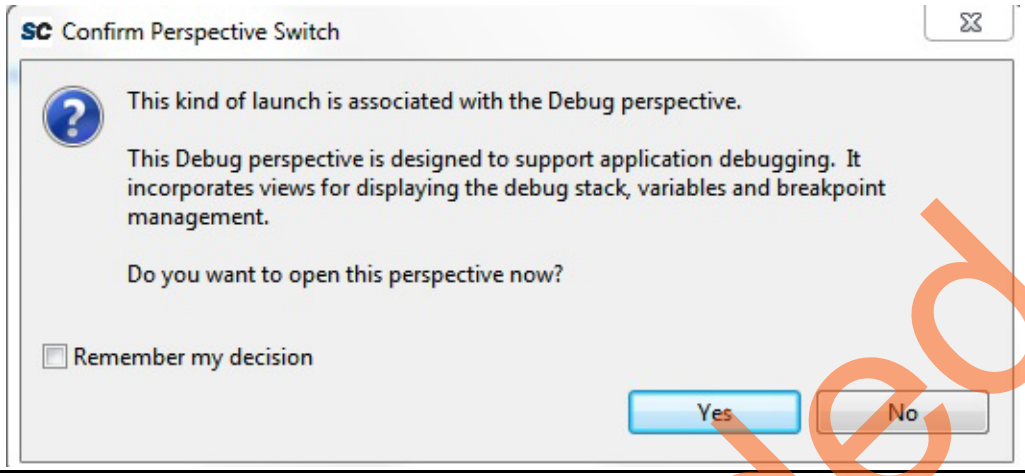

Figure 52 • **Confirm Perspective Switch**

7. Debug Perspective is similar as shown Figure 53.

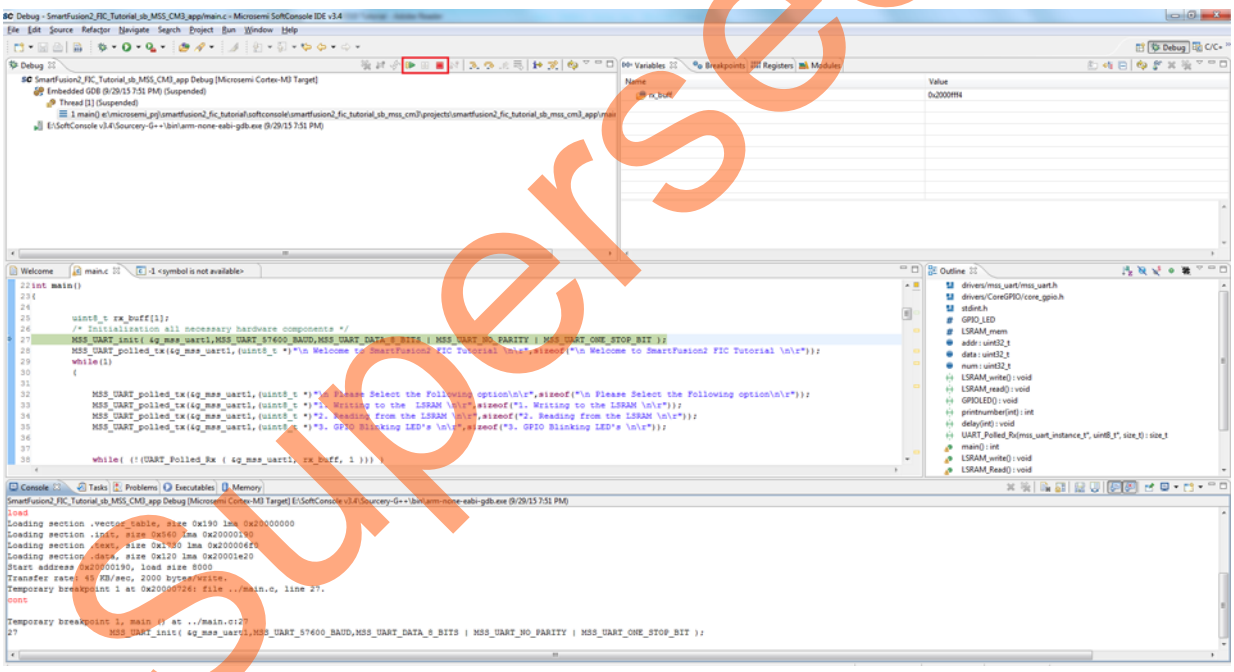

Figure 53 • **Debug Perspective**

8. Run the application by clicking **Run > Resume** or click the **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 54.

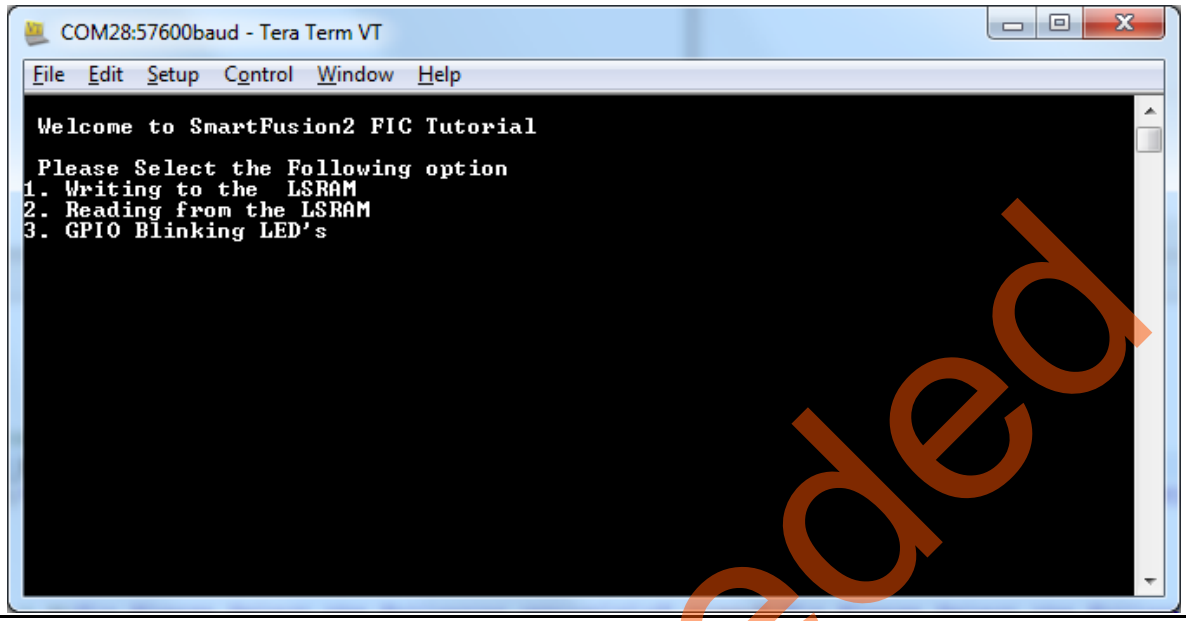

Figure 54 • **Tera Term Window**

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 55.

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

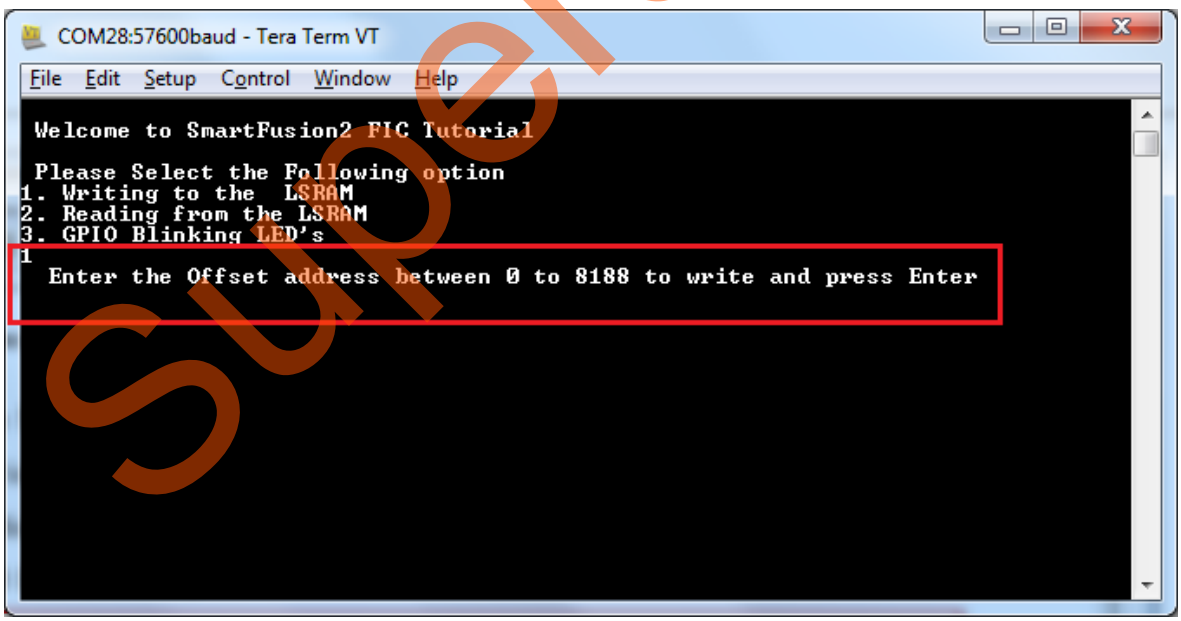

Figure 55 • **Writing to LSRAM**

10. After Entering the offset address, it prompts for **Enter data to write** as shown in Figure 56.

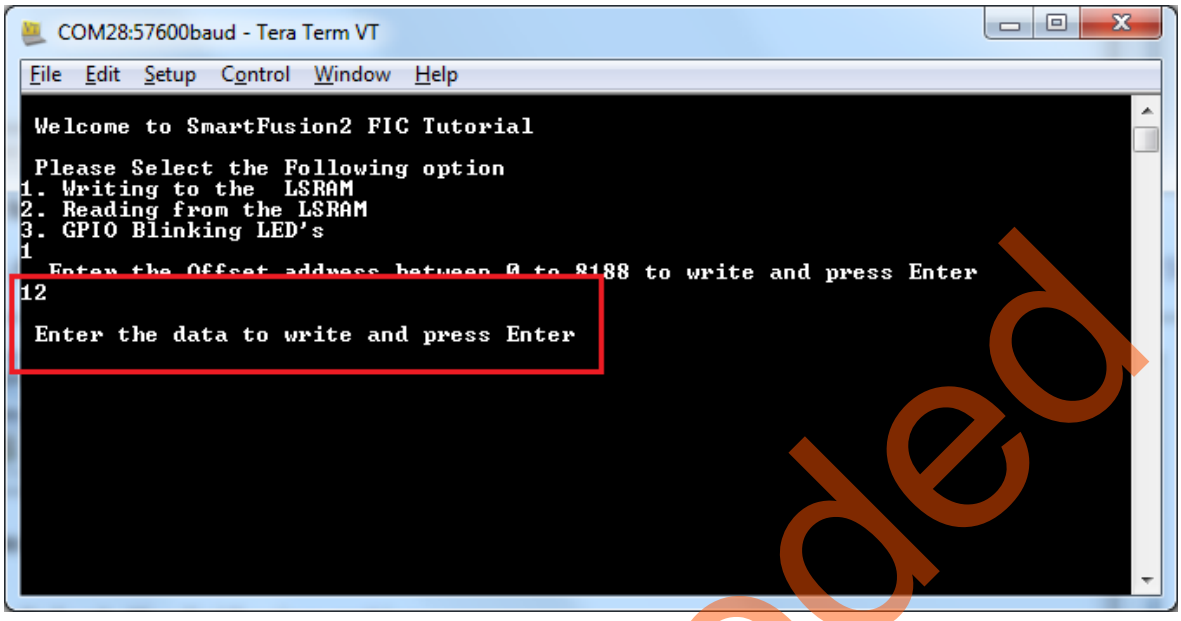

***Figure 56* • Writing to LSRAM**

11. Select **Reading from the LSRAM**, it prompts for **Enter the offset address to read and press Enter** as shown in Figure 57.

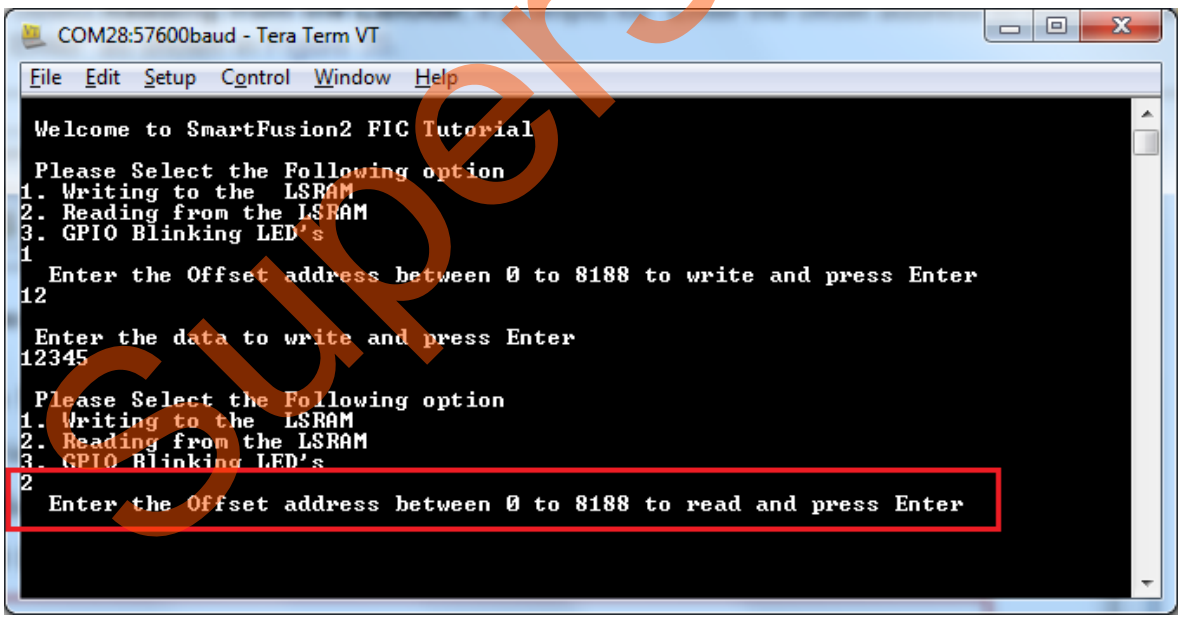

***Figure 57* • Reading from LSRAM**

12. Select GPIO LED Blinking, it prompts for **Enter the number between 0 to 255 and press Enter** as shown in Figure 58 on page 49.

Note: For SmartFusion2 Starter Kit, Tera Term window is displayed, as shown in Figure 59.

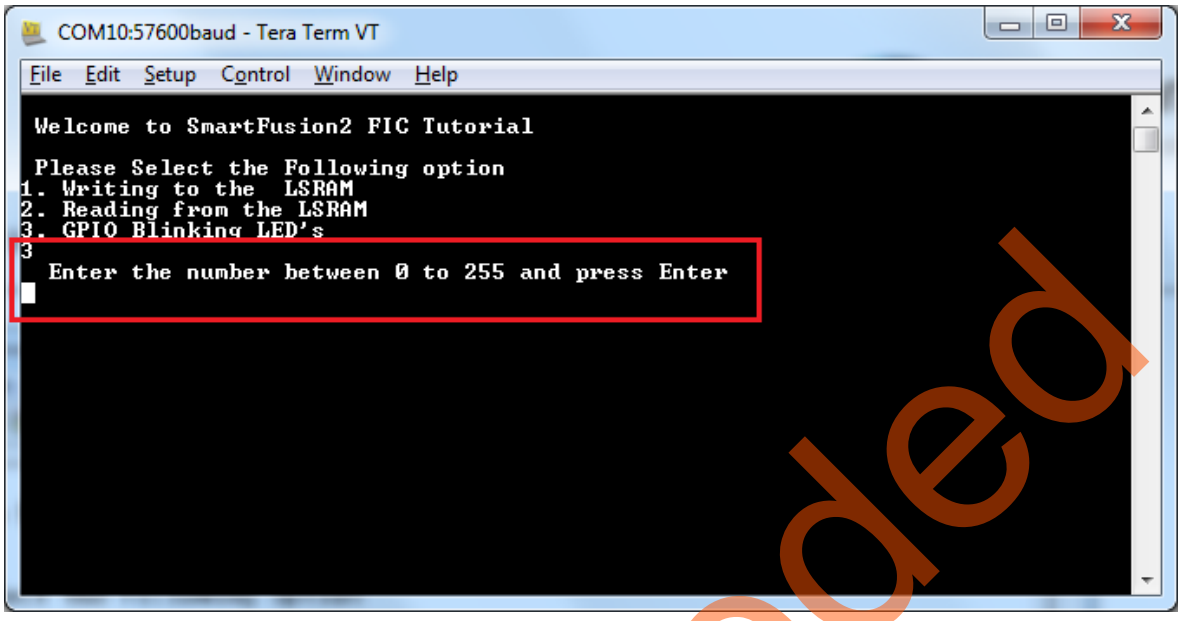

***Figure 58* • Selecting GPIO LED Blinking**

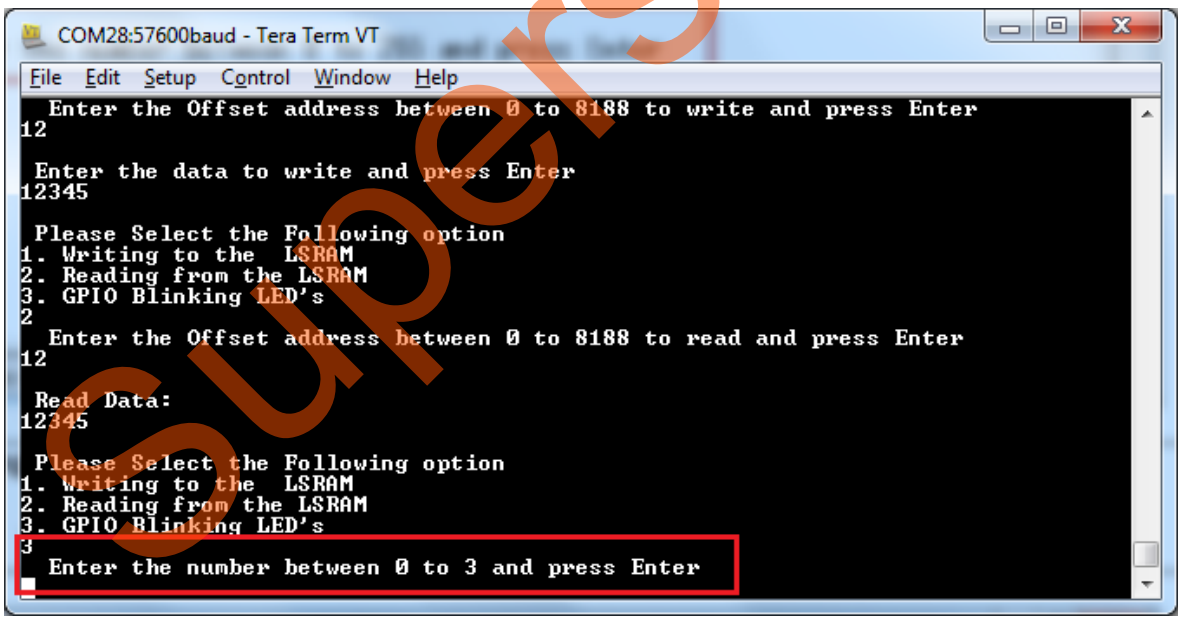

***Figure 59* • Selecting GPIO LED Blinking**

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 the SmartFusion2 SoC FPGA 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 project using Libero SoC design software, configuring the SmartFusion2 MSS, interfacing fabric peripherals to the SmartFusion2 MSS using fabric interface controllers (FIC_0), simulation of the design using BFM commands and running the application design on board.

A – List of Changes

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

Revision	Changes	Page
Revision 12 (December 2015)	Updated Figure 1 and Figure 38 (SAR 73892).	5 and 34
Revision 11 (October 2015)	Updated the document for Libero v11.6 software release changes (SAR 72067).	NA
Revision 10 (February 2015)	Updated the document for Libero v11.5 software release (SAR 64506).	NA
Revision 9 (August 2014)	Updated the document for Libero v11.4 software release (SAR 59820).	NA
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

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

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

Website

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

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. Visit About Us for sales office listings and corporate contacts.

ITAR Technical Support

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

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet Solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at **www.microsemi.com**.

50200310-12/12.15