Libero® SoC Quick Start Guide

for Software v11.4





Table of Contents

Introduction and Design Overview 3 Tutorial Requirements 3 Design Overview 4 Interrupt Generator Block Description 5
Step 1 - Creating a Libero SoC Project and Configuring the SmartFusion2 Microcontroller Subsystem 6 Libero SoC Interface Description 8
Step 2 - Configuring the SmartFusion2 MSS
Step 3 - Import Timer Blocks
Step 4 - Performing Pre-Synthesis Simulation
Step 5 - Implementing the Design
Software Implementation44Step 1 – Invoking SoftConsole44Step 2 - Configuring Hyper Terminal/Other Terminal Emulator Programs48Step 3- Debugging the Application Project Using SoftConsole52Step 4 - Building an Executable Image in Release mode55
Installing Drivers for the USB-UART 60
Product Support 63 Customer Service 63 Customer Technical Support Center 63 Technical Support 63 Website 63 Contacting the Customer Technical Support Center 63 ITAR Technical Support 64



Introduction and Design Overview

This tutorial introduces you to the Microsemi[®] system-on-chip (SoC) development flow using Libero[®] SoC. It is a starting point for any FPGA design engineer who is new to Microsemi FPGAs, or just wants to learn more about Libero SoC. It demonstrates the basics on how to use Libero SoC and its tools to create a simple design. The sample design incorporates the Libero SoC Catalog IP core macros and hard embedded microcontroller subsystem (MSS) and guides you through synthesis, simulation and programming.

The tutorial was developed using the SmartFusion2[®] Evaluation Kit Board and M2S025T device. For details on the SmartFusion2 Evaluation Kit Board, refer to the SmartFusion2 Evaluation Kit User Guide. This tutorial covers the following tools and features:

- Libero SoC
 - Design flow
 - SmartDesign
 - Catalog
- Mentor Graphics[®] ModelSim[®] ME
- Synopsys Synplify Pro ME
- Microsemi Designer
 - Compile
 - Place and Route
 - I/O Attribute Editor
 - Pin Editor
 - SmartTime
- FlashPro
- SoftConsole

Tutorial Requirements

Software Requirements

This tutorial requires that the following software is installed on your computer:

- Libero SoC v11.4, which can be downloaded from http://www.microsemi.com/soc/download/software/libero/default.aspx
- FlashPro v10.0 or higher, which is installed as part of the Microsemi Libero SoC installation and can be launched from within Libero SoC or standalone.
- SoftConsole v3.4 or higher, which is installed as part of the Microsemi Libero SoC installation and can be launched from within Libero SoC or standalone.
- HyperTerminal or similar software (PuTTy or Tera Term), normally under Start > Programs > Accessories > Communications > HyperTerminal.

Note: HyperTerminal is no longer a standard feature on newer windows installations. For other alternatives to HyperTerminal, see

Alternatives to HyperTerminal in windows 7 | Windows Reference

• USB Drivers for USB to UART connection.

Install the driver from http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files



Hardware Requirements

You will need the SmartFusion2 Evaluation Kit, which consists of the following hardware:

- SmartFusion2 Evaluation Kit Board (M2S-EVAL-KIT DVP-102-000402-001 Rev.C)
- FlashPro4 JTAG Programmer
- 12V/2A Wall-Mounted Power Supply
- USB 2.0 A-male to mini-B Y-Cable for UART/power interface (up to 1A) to PC

Extracting Source Files

Download the Design Files from http://www.microsemi.com/document-portal/doc_download/134397-liberosoc-qs-df. Extract design files in the root directory of your local drive (e.g., C :\) using 7-zip.

The design files for this tutorial include:

- Timer_0 and Timer_1 HDL files HDL source files for the Interrupt Generator block. Verilog and VHDL versions are provided.
- Main.c file C file for running the Application in SoftConsole
- Post-layout_wave.do file wave.do file for running post-layout simulation in ModelSim

Design Overview

SmartFusion2 cSoC FPGA devices contain a 166 MHz ARM® Cortex[™]-M3 processor, Ethernet MAC, DMA engine, real-time counter (RTC), embedded nonvolatile memory (eNVM), embedded SRAM (eSRAM), and FPGA fabric consisting of programmable 4-input LUTs and D Flip Flops, static random access memory (SRAM), Clock Conditioning Circuitry (CCC) with dedicated phase locked loops (PLLs) and MATH blocks. The Cortex-M3 processor includes an interrupt controller called the nested vectored interrupt controller (NVIC).

The SmartFusion2 device has 16 dedicated fabric-to-MSS interrupt lines, MSS_INT_F2M[15:0].

In addition, there are 32 General Purpose Inputs/Outputs (GPIOs) which can be configured as Inputs, Outputs or Bi-directional signals., When configured as Inputs and routed to the Fabric, the 32 GPIOs can be used as additional Fabric-to-MSS interrupt resources.

The design example (Figure 1) uses two GPIOs and one dedicated Fabric Interrupt to interrupt the MSS from the FPGA fabric. The interrupt generator block has two timer blocks, Timer0 and Timer1. The universal asynchronous receiver/transmitter (MMUART) in MSS is used for printing interrupt messages to the terminal emulator.

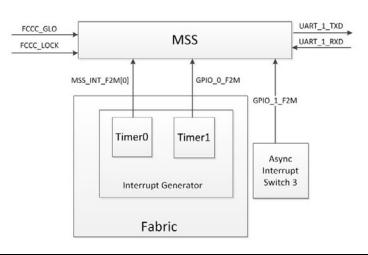


Figure 1 • Tutorial Design Block Diagram



The example design contains the following blocks:

- MSS
- 25/50 MHz RC Oscillator
- Fabric Clock Conditioning Circuit (CCC) with dedicated PLL
- SYSRESET to generate the System Reset signal
- Interrupt generator which consists of two counters: Timer0 and Timer1

The example design contains the following inputs and outputs:

- Inputs: Clock, Reset and External interrupt signal
- Outputs: MMUART to HyperTerminal/Tera Term

Interrupt Generator Block Description

The interrupt generator (Figure 1) has two timer blocks, Timer0 and Timer1. The two timers are implemented in the FPGA fabric and run with the same clock source. The clock conditioning circuit (CCC) generates a 10 MHz clock, which is the clock source for the timers.

Timer 0 and Timer 1 are internal synchronous interrupts. The output from Timer0 is connected to MSS_INT_F2M[0] of the MSS and the Timer1 block is connected to the GPIO_0_F2M. The third interrupt is generated asynchronously by pressing SW3 on the board. SW3 is connected to FPGA I/O pin K18, which is connected to the GPIO_1_F2M of the MSS. Figure 1 shows the block diagram of the interrupt generator.

Whenever any of the interrupts occur (Timer 0, Timer 1, or the SW3 switch interrupt), the Cortex M-3 processor executes the corresponding interrupt service routine, which sends a message to the MMUART indicating the source of the interrupt. The terminal displays the message.

This tutorial provides step-by-step instructions on how to configure the MSS and timer blocks, simulate the timer block, synthesize, place and route, and generate a programming file for the entire design. It also guides you on how to program the design on the SmartFusion2 Evaluation Kit Board and how to run the firmware applications on the Cortex M-3 using SoftConsole.



1 – Step 1 - Creating a Libero SoC Project and Configuring the SmartFusion2 Microcontroller Subsystem

1. Click Start > Programs > Microsemi Libero SoC v11.4> Libero SoC v11.4 Libero SoC opens (Figure 1-1).



Figure 1-1 • Libero SoC



2. From the **Project** menu, choose **New Project**. The New Project dialog box appears (Figure 1-2).

ew Project							
Enable Block Cre	ation						
Name:	Interrupt_	MSS_GPIO					
Location:	C:/Micros	emi_proj			Browse		
Prefered HDL type:	Verilog) 🔘 VHDL					
Description:					*		
🗼 Edit Tool Profile:	5						
Device							
Family:	SmartFusio	n2 🔻					
Die:	M2S025T	•					
Package:	484 FBGA	▼					
Speed:	STD	▼	_		_		
Core Voltage (V):	1.2	▼ Ram	ip Rate: 10	10ms Minimum 🔻			
Operating Conditions	:						
		Range	Best	Typical	Worst	_	
Junction Temp		сом -	0	25	85		
Core Voltage	(V)	СОМ 🗸	1.260	1.200	1.140	*	
System Controlle PLL Supply Voltage (Design Templates and Use Design	V): 2.5 d Creators	e Core		v	ersion		•
SmartFusion2 Micro SmartFusion2 Micro					1.201		
SmartFusion2 Micro	controller Subs	ystem (MSS)			1.100		H
STHAT CTUSIONZ MICTO	Vendor: A	Actel martFusion2MSS			Show or	nly latest ve	rsior
	Name: M	SS					•
Help		SS			ОК	Car	► ncel

Figure 1-2 • Libero SoC New Project Dialog Box



Step 1 - Creating a Libero SoC Project and Configuring the SmartFusion2 Microcontroller Subsystem

- 3. Enter the information as shown in Figure 1-2. If a value in the dialog box is not specified below you can use the default.
- Name: Interrupt_MSS_GPIO
- Location: C:/Microsemiprj
- Preferred HDL type: VHDL (for VHDL projects) or Verilog (for Verilog projects)
- Family: SmartFusion2
- Die: M2S025T
- Package: 484 FBGA
- Speed: STD
- Core Voltage: 1.2
- Check the Use Design Tool checkbox
- Select MSS Core Version 1.1.100 or later

If the SmartFusion2 MSS text is in italics, the core is available for download but not in your IP vault. Double-click the core name to download the latest MSS core

- 4. Click **OK** to close the New Project dialog box.
- Note: Click Yes if the software prompts you to download the MSS. This occurs when the vault does not have the selected MSS core.

Libero SoC Interface Description

The Libero SoC interface enables a push-button design flow via several tabs and an expanded work area.

- Work window Displays the SmartDesign canvas, HDL editor or Report view. Click the Maximize/Restore Work Area button to show/hide the other interface elements.
- **Design Hierarchy tab** Lists components and modules in your design. Use it to manage your design files.
- Files tab Lists your project files by directory; use it to manage your project files directly.
- **Design Flow window** Enables you to execute the push-button design flow or, if you prefer, open the tools interactively and specify custom settings.
- **Catalog** Lists the cores available for use in your design. Click and drag them onto your Canvas and add them to your design.
- HDL Templates Lists HDL templates for common constructs; double-click a template to copy it to the clipboard, and then paste it into your HDL to use it in your design.
- Log window Lists all messages, errors, warnings, and info for the SoC tools. Click each type to filter accordingly.



2 – Step 2 - Configuring the SmartFusion2 MSS

If you completed the steps in the previous chapter, the **SmartDesign** Canvas opens with the SmartFusion2 MSS component (Figure 2-1).

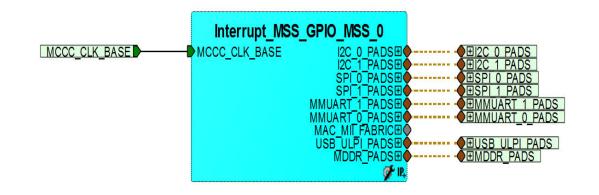


Figure 2-1 • SmartFusion2 MSS on the Canvas



 Double-click Interrupt_MSS_GPIO_MSS_0 to open the SmartFusion2 MSS configurator (Figure 2-2)

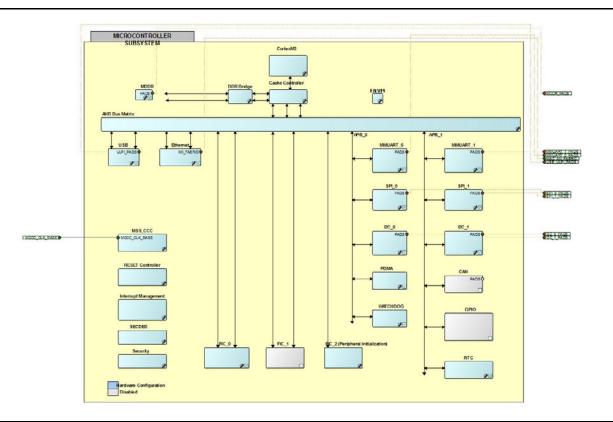


Figure 2-2 • SmartFusion2 MSS Configurator

Configurable MSS peripherals have a small wrench symbol and checkbox in the lower right corner (Figure 2-3). To disable a peripheral, select the peripheral, right-click and choose **Disable**, or click the checkbox. The peripheral turns grey to indicate it has been disabled.

Disabled peripherals can be enabled by repeating the procedure.

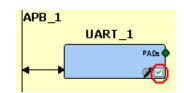


Figure 2-3 • Disabling the Peripheral

- 2. Disable the following peripherals:
 - External Memory Controller (MDDR)
 - MMUART_0
 - SPI_0
 - SPI_1
 - I2C_0
 - I2C_1
 - FIC_1
 - PDMA



- WATCHDOG
- RTC
- CAN
- USB
- Ethernet

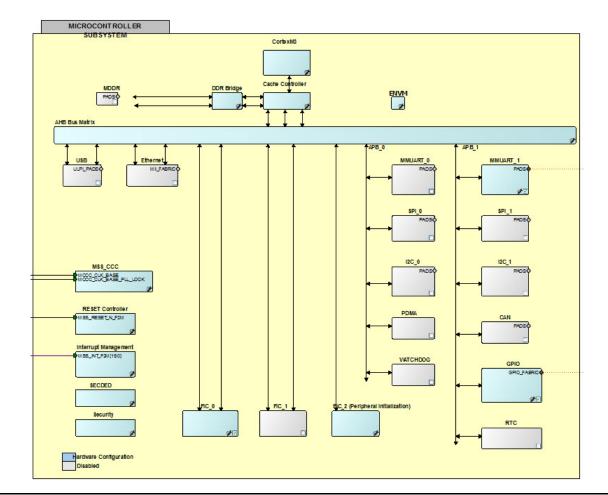


Figure 2-4 • SmartDesign MSS in the SmartDesign Canvas

- 3. Double-click the MSS CCC block to configure the MSS CCC (Figure 2-4).
- 4. Set the MSS_CLK Configurator options as follows (shown in Figure 2-5):
 - CLK_BASE: 10 MHz
 - Monitor FPGA Fabric PLL Lock (CLK_BASE_PLL_LOCK): checkbox should be checked.
 - M3_CLK: 160 MHz

Use the default for all other settings.



Use the default for all other settings.

	F
Clock Source	Cortex-M3
CLK_BASE 10 MHz	
Monitor FPGA Fabric PLL Lock (CLK_BASE_PLL_LOCK)	Cache Controller
Cortex-M3 and MSS Main Clock	Cache Controller
M3_CLK 160 MHz 160.000 MHz	
THE SECONDERING	MSS_CCC HPDMA SWITCH
MDDR Clocks	MO_CLK TAPB_0 TAPB_1
MDDR_CLK = M3_CLK = 2	
DDR_SMC_FIC_CLK = MDDR_CLK / 1 *	
	AP8_0_CUK 120_0 120_1
MSS APB_0/1 Sub-busses Clocks	AP9_T_CUK DMA CAN
APB 0_CLK = M3_CLK / 1 160.000 MHz	FIC.A.CIK
APB 1 CLK = M3_CLK / 1 - 160.000 MHz	
FPGA Fabric Interface Clocks	MSS FIC_1
FIC 0 CLK = M3_CLK / 1 T	Lowest frequency
FIC 1_CLK = M3_CLK / 1 *	
	FPGA fabric FPGA fabric
	FIC_1_CLK

Figure 2-5 • MSS Clock Conditioning Circuitry Configurator

- 5. Click OK to close the MSS Clock Conditioning Circuitry Configurator box.
- 6. Double-click Interrupt Management to open and configure the MSS Interrupts. Check the option Use Fabric to MSS Interrupt (Figure 2-6). Click OK to continue.

Configu	uration
	Use Fabric to MSS Interrupt 🔽
	Use MSS to Fabric Interrupt

Figure 2-6 • MSS Interrupts Management Configuration

7. Double-click the **GPIO** block in MSS to configure it. Configure GPIO_0 and GPIO_1 as **Inputs** (as shown in Figure 2-7) and select Fabric_A from the drop-down list for the Connection. Leave the



rest of the ports at their default settings. Click **OK** to close the GPIO configuration dialog box. The F2M_GPI[1:0] ports are promoted to the top level automatically.

Configuration					Connectivity Preview	
Set/Reset Defi	nition					
PIO_31_24 Reset Sou	Irce SYSREG (MSS_GPI	0_31_24_SOFT_RES •	Reset State 1	•	REF	5
PIO_23_16 Reset Sou	Irce SYSREG (MSS_GPI	0_23_16_SOFT_RES •	Reset State 1	-		->
PIO_15_8 Reset Sour	ce SYSREG (MSS_GPI	0_15_8_SOFT_RESE -	Reset State 1	-		
PIO_7_0 Reset Source	e SYSREG (MSS_GP	IO_7_0_SOFT_RESET	Reset State 1			
GPIO Assignme			vanced Options			
GPIO ID	Direction	Package Pin	Connecti.			
GPIO_0	Input -		FABRIC A -	- Âl		
GPIO_1	Input •		FABRIC_A V	Ξ	MSS	
GPIO_2	Not Used 🔻		IO_A *			
GPIO_3	Not Used 👻		IO_A -			
GPIO_4	Not Used 🔻		IO_A *		FPGA Fab	
GPIO_5	Not Used 🔹		TO_A *			
GPIO_6	Not Used 👻		10_A ~		Click on a signal row to s	ee the pr
GPIO_7	Not Used 🔹		IO_A *			
GPIO_8	Not Used 💌		[10_A *]			
GPIO 9	Not Used 👻		IO A T	*		
•		III	,			
						-

Figure 2-7 • MSS GPIO Configurator

- 8. Double-click the Reset Controller block inside the MSS to open the Reset Controller Configurator.
- Check Enable FPGA Fabric to MSS Reset (MSS_RESET_N_F2M). Uncheck the other items (Figure 2-8).

Configuration			
Enable FPGA Fabric to	o MSS Reset (N	MSS_RESET_N_	F2M) 🔽
Enable FPGA Fabric to	o M3 Reset (M	13_RESET_N)	
Enable MSS to FPGA	Fabric Reset (N	MSS_RESET_N_	_M2F)
Enable MSS to FPGA	Fabric Reset (1	MSS_RESET_N_	_M2F)
Help 🔻		ОК	Cancel

Figure 2-8 • Reset Configurator

 Double-click the eNVM module available in the SmartDesign MSS Configurator to add an eNVM data storage client. The Data Storage Client stores the Cortex-M3 application code. The eNVM Configurator appears (Figure 2-9).

wailable Client type	es				User Client	ts in eNVM			
Data Storage Serialization	Q-	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start Address
Senanzation	Create a new Da	ata Storage Client							
Add to Sy	ystem								
sage Statistics									
Isage Statistics Ivailable Pages: 2									

Figure 2-9 • eNVM Configurator

11. Double-click Data Storage under Available Client Types to add a Data Storage Client. Enter My_app in the Client name field and select the **No Content (Client is a placeholder)** radio button. This creates a placeholder in the eNVM to store the Application code.



12. Enter 32 for Size of word and 2048 for Number of Words. See Figure 2-10).

Memory fil	e:		
Format	Intel-Hex 🔹		
Use	absolute addressing	0	
Content fill	ed with 0s		
No Content	(Client is a placeholder)		
Start address:	0x 0 🚔		
Size of word:	32 🔻 bits		
Number of Words:	2048	(decimal)	
Use as ROM	0		
Use Content for	Simulation		

Figure 2-10 • Add Data Storage Client

13. Click **OK** to close the dialog box. The Data Storage Client is created with the correct depth and width (Figure 2-11).

Available Client types				User Clien	ts in eNVM			
Data Storage Serialization	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start Address
	1 Data Storage	My_app	2048 x 32	0	0	63	N/A	
Usage Statistics Available Pages: 2032 Used Pages: 64								

Figure 2-11 • User Client created in eNVM

- 14. Click OK to exit.
- 15. Save the Interrupt_MSS_GPIO component (File > Save Interrupt_MSS_GPIO).



The MSS component (Interrupt_MSS_GPIO_MSS_0) appears on the SmartDesign Canvas. The Warning symbol indicates that the port list for the SmartFusion2 component has changed.

16. Right-click and choose Update Instance(s) with Latest Component (Figure 2-12).

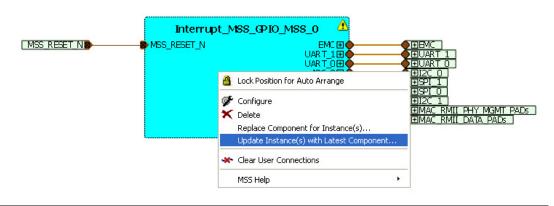


Figure 2-12 • Update the MSS Component

Device I/O Standard

I/O Bank 1 and I/O Bank 2 of the SmartFusion2 device are connected to 3.3 V on the Evaluation Kit Board. Set the I/O standard of the design's I/O to LVCMOS 3.3 as follows:

- From the Project menu, choose Project Settings and then click Device I/O Settings (Project > Project Settings > Device I/O Settings).
- 2. From the Default I/O Technology drop-down list, choose LVCMOS 3.3V (Figure 2-13).

Device Device I/O Setti	Reserve Pins for Pro	bes			Save
Preferred HDL T Default I/O Technology: LVCMOS 3.3V Design Flow UVTTL			Please use the I/O Editor to change individual I/O attributes.		
Simulation Opti		LVCMOS 3.3V LVCMOS 2.5V	oltage Range		
DO File Waveforms	Range	LVCMOS 1.8V LVCMOS 1.5V	Typical	Maximum	
Vsim comma	СОМ	LVCMOS 1.2V 1.14	1.2	1.26	
Simulation Libra SmartFusion2	COM	1.425	1.5	1.575	
SmartFusion2	COM	1.71	1.8	1.89	
	COM	2.375	2.5	2.625	
	COM	3.15	3.3	3.45	

Figure 2-13 • Device I/O Settings

3. Click **Save** and **Close** to close the Project Settings dialog box.



3 – Step 3 - Import Timer Blocks

The two counter blocks required in the design are provided with this tutorial. Both Verilog and VHDL versions of the counters are available. The two counters serve as timers to generate synchronous interrupts to the MSS.

1. Import the Timer Block HDL (File > Import > HDL Source Files) into the Project (Figure 3-1).

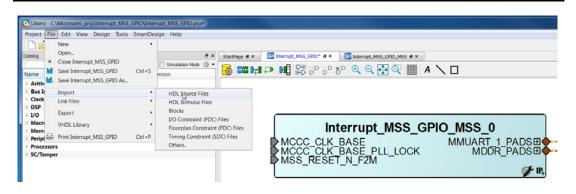


Figure 3-1 • Import Timer Block HDL File via Import HDL Source Files

 Navigate to the folder location of the Verilog or VHDL timer files where you have extracted the source files for this tutorial. Select the Verilog files (Timer_0.v and Timer_1.v) or the VHDL files (Timer_0.vhd and Timer_1.vhd) to match your Preferred HDL Type in your Project Settings. Click Open. Timer_0 and Timer_1 appear in the Design Hierarchy after the Import (Figure 3-2).

Organize 👻 New folder		III • 🗍
 Favorites Desktop Downloads Recent Places Libraries Documents Music Pictures Videos 	I	Documents library Source_files Arrange by: Folder ▼ Name Imer_1.vhd Imer_0.vhd Timer_0.vhd Timer_1.v Timer_1.v
Computer Local Disk (C:) DATA (D:) timing_doc (\\dm6\sqaAX\vishakh\work\2014\Feb14) (Q:) simple_test (\\dm6\sqaAX\vishakh\work\Dec13\ip_encryp) (R:) File name:	÷	HDL Source Files (*.vhd *.v * Open Cancel

Figure 3-2 • Timer_0 and Timer_1 HDL Source Files



- 3. From the Design Hierarchy, drag and drop Timer_0 and Timer_1 HDL blocks into the SmartDesign Canvas.
- 4. From the Catalog, expand the Macro Library Group, and drag and drop a two-input AND gate into the SmartDesign Canvas.
- 5. Drag and drop SYSRESET (under the Macro Library group) into the SmartDesign Canvas.
- 6. From the Catalog, expand the Clock and Management group. Drag and drop the Chip Oscillator (OSC) into the SmartDesign Canvas. Double-click the OSC to open the Configurator..
- 7. Check the On-chip 25/50 MHz RC Oscillator box and select Drives Fabric CCC(s) (Figure 3-3).

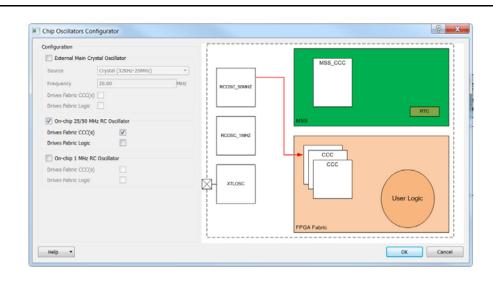


Figure 3-3 • Oscillator (OSC) Configurator

- 8. Click OK to close the OSC Configurator.
- 9. Drag and drop the Clock Conditioning Circuitry (FCCC) into the SmartDesign Canvas.
- 10. Double-click the FAB CCC to open the Configurator.



11. Click the **Basic** tab. For the Reference Clock, select **Oscillators** and then **25/50 MHz Oscillator** (**Oscillators** > **25/50 MHz Oscillator**). Click **OK** to close the FAB CCC Configurator (Figure 3-4).

Reference Clock 50.000 MH2 5/50 MH2 Oscillato ▼	СССС	Sert	Exact Value	Frequency	Actual
Dedicated Input Fabric Input Oscillators	Crystal Oscillator	🔲 GL1	E	100 MHz	MH2
	1 MHz Oscillator 25/50 MHz Oscillator		E	100 MHz	MH2
				100 MHz	MHz
		LOCK			
		J			

Figure 3-4 • FCCC Configurator

- On the SmartDesign Canvas, hold the SHIFT key and click RCOSC_25_50MHZ_CCC_OUT of the Oscillator (OSC) and RCOSC_25_50MHZ_CCC_IN of the Clock Conditioning Circuit (FCCC) to select both ports.
- 13. Right-click and choose **Connect** from the drop-down menu to connect the two selected ports (Figure 3-5).

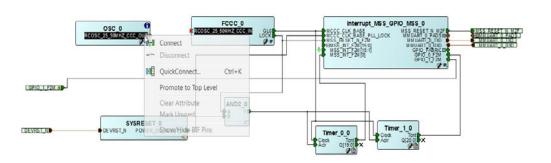


Figure 3-5 • Making Connections in the SmartDesign Canvas

14. Repeat the procedure to make the following connections in the SmartDesign Canvas:

- From GLO output of Clock Conditioning Circuit to Clock input of bothTimer0 and Timer1 and MCCC_CLK_BASE of the Interrupt_MSS_GPI0_MSS_0 Block.
- From LOCK output of the CCC to the A input of the AND gate and MSS_RESET_N_F2M Input of the Interrupt_MSS_GPIO_MSS_0 block.



- From POWER_ON_RESET_N output of SYSRESET_0 block to the B input of the AND gate and MCCC_CLK_BASE_PLL_LOCK input of Interrupt_MSS_GPIO_MSS_0 Block.
- From the Y output of the AND gate to the Aclr input of both Timer_0 and Timer_1.
- From Tcnt of Timer1 to GPIO_0_F2M of the Interrupt_MSS_GPIO_MSS_0 block.
- From Tcnt Output of Timer_0 to MSS_INT_F2M[0] of the Interrupt_MSS_GPIO_MSS_0 block.
- Note: You need to slice the bus MSS_INT_F2M [15:0] first before you make the connection.

To slice the bus signal, right-click the MSS_INT_F2M[15:0] bus and choose **Edit Slice**. The Edit Slice dialog box appears.

Click the green + sign to add a slice. Enter 0 for the Left and Right columns for the first slice and enter 15 for the Left and 1 for the Right column for the second slice (as shown in Figure 3-6). Click **OK**.

MSS_INT_F2M[15:0] Left Right 1 0 0 2 15 1	Edit Slices - MSS_INT_ Create 16 • slices of w			dd Slices
	MSS_INT_F2M[15:0]		Left	Right
2 15 1		1	0	0
		2	15	1
		2	15	1

Figure 3-6 • Edit Slice Dialog Box

- Tie MSS_INT_F2M[15:1] to low. (Right-click > Tie Low).
- 15. Right-click and choose Promote to Top Level:
 - GPIO_1_F2M output of the Interrupt_MSS_GPIO_MSS_0 block
 - MSS_RESET_N_M2F output of the Interrupt_MSS_GPIO_MSS_0 block
- 16. Right-click and choose Mark Unused:
 - Q[19:0] of Timer_0
 - Q[20:0] of Timer_1

After making all the connections, the **Interrupt_MSS_GPIO** appears as shown in Figure 3-7. Pad ports and nets with dedicated pin assignments are displayed in brown.

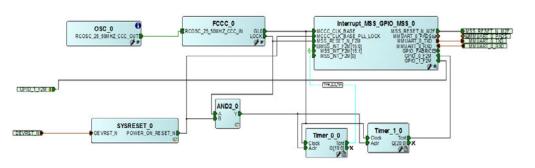


Figure 3-7 • SmartDesign Canvas After Making Connections



- 17. From the Design Flow window, right-click Configure Firmware Cores and choose **Open Interactively**.
- 18. In the DESIGN_FIRMWARE tab, the following required firmware drivers should have been selected for you and the checkbox under Generate checked. If the driver is greyed out and in italics, it is not in your vault. Download it into your vault. See Figure 3-8.
 - SmartFusion2_CMSIS_0 Version 2.2.101
 - SmartFusion2_MSS_GPIO_Driver_0 Version 2.0.101
 - SmartFusion2 MSS HPDMA Driver 0 Version 2.0.101
 - SmartFusion2_MSS_MMUART_Driver_1 Version 2.0.101
 - SmartFusion2_MSS_NVM_Driver_0 Version 2.1.102
 - SmartFusion2_MSS_System_Services_Driver_0 Version 2.0.103
 - SmartFusion2_MSS_Timer_Driver_0 Version 2.0.101

Note: If you select the wrong core version, the Application may not build successfully.

SD	DESIGN	FIRMWA	RE & X So Interrupt_MSS_GPIO_MSS & X	Reports 🗗 × 🔛 Interrupt_MSS_GP	10 🗗 × Sta	rtPage & ×	Timer_0.v ₽×	Timer_1.v & ×
C								
	Senerat		Instance Name	Core Type	Version		Compatible Hardware In	stance
	v	Ø 4	SmartFusion2 CMSIS 0	SmartFusion2_CMSIS	2.2.101 +	Interrupt_MSS	GPIO MSS	
	V	-	SmartFusion2_MSS_GPI0_Driver_0	SmartFusion2_MSS_GPI0_Driver	2.0.101	Interrupt_MSS	_GPIO_MSS:GPIO	0
	1	4	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.0.101	Interrupt_MSS	_GPIO_MSS	
	1	-	SmartFusion2_MSS_MMUART_Driver_1	SmartFusion2_MSS_MMUART_Driver	2.0.101	Interrupt_MSS	_GPIO_MSS:MMUART_1	
	1	-	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.1.102 -	Interrupt_MSS	_GPIO_MSS	
	1	-	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.0.103 -	Interrupt_MSS	GPIO_MSS	
,	1	Z .	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.0.101	Interrupt_MSS	GPIO_MSS	

Figure 3-8 • Configure Firmware

19. From the SmartDesign menu, choose Generate Component.

If you do not see **Generate Component** in the SmartDesign menu, make sure that you are viewing the Interrupt_MSS_GPIO tab (Figure 3-9).

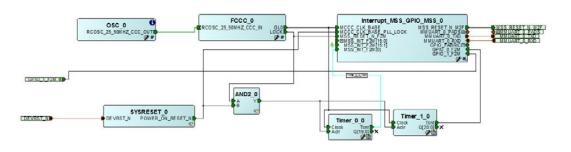


Figure 3-9 • Generate Component



A green check mark under Create Design in the Libero SoC Design Flow window indicates the design was created without any errors (Figure 3-10).

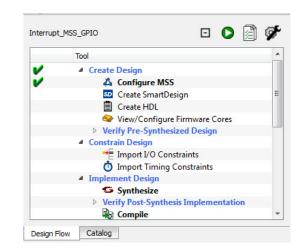


Figure 3-10 • Successful Generation of SmartDesign Component

After successful MSS Component generation, the Log window displays the message Info: 'Interrupt_MSS_GPIO' was successfully generated (Figure 3-11).

MSS_GPIO.v'. _MSS_GPIO_FCCC_0_FCCC.v'. _MSS_GPIO_OSC_0_OSC.v'. .v'. _MSS_GPIO_MSS.v'. _MSS_GPIO_MSS_syn.v'.
_MSS_GPIO_OSC_0_OSC.v'. .v'. _MSS_GPIO_MSS.v'. _MSS_GPIO_MSS_syn.v'.
MSS GPIO_MSS.v'. MSS GPIO_MSS_syn.v'.
MSS_GPIO_MSS_syn.v'.
b.v'.
· •
· .
·
project was opened.
GPIO' was successfully generated.
uments/Work/Liberoll.3 Quick Start UG/Tim working/Interrupt MSS GPIO/sim
ayout simulation.log'
ayout_simulation.log
nts\Work\Libero11.3_Quick_Start_UG\Tim_working\Interrupt_MSS_GPIO\simula

Figure 3-11 • Log Window

20. From the **Design** menu choose **Reports > Interrupt_MSS_GPIO_DataSheet.xml**. Examine the DataSheet.xml file to familiarize yourself with the Generated Files, Firmware and Memory Map



sections (click the hyperlink at the top of the datasheet to move to the section of interest). See Figure 3-12.

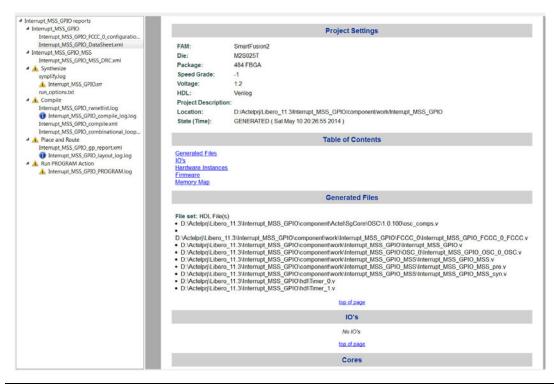


Figure 3-12 • Datasheet File



4 – Step 4 - Performing Pre-Synthesis Simulation

Libero can create an HDL testbench for you or you can create a testbench using SmartDesign. For this tutorial, you use an HDL testbench from Libero.

1. In the Design Hierarchy window, right-click **Interrupt_MSS_GPIO** and choose **Create Testbench** > **HDL**. Enter **Interrupt_MSS_GPIO_tb** for the testbench name (Figure 4-1).

how: Components 💌			
▲ O work			
XTLOSC_FAB (osc_comps.v)			
A XTLOSC (osc_comps.v)			
RCOSC_25_50MHZ_FAB (osc_	_comps.v)		
RCOSC_1MHZ_FAB (osc_com	ips.v)		
RCOSC_1MHZ (osc_comps.v)			
▲ SD Interrupt_MSS_GPIO			
Interrupt_MSS_GPIO_F	Open Component	C.v)	
Interrupt_MSS_GPIO_1	Generate Component		
Interrupt_MSS_GPIO_C)	
Timer_0 (Timer_0.v)	Open HDL File		
🗎 Timer_1 (Timer_1.v)	Check HDL File		
	Create I/O Constraint from Module		
	Create ViewDraw Symbol		
	Create Testbench	SmartDesign	
	Delete from Project	HDL	
	Delete from Disk and Project		1
	Properties		
	Show Module	pdc	

Figure 4-1 • Testbench Creation

2. From the **Project** menu, choose **Project Settings** and choose **DO File** under **Simulation Options**. Under **Simulation Runtime**, enter a value of **1ms** (Figure 4-2).



Device Device I/O Setti	Vse automatic DO file		Save
Preferred HDL T	Simulation runtime:	1ms 👻	Restore Defaults
Design Flow	Testbench module name:	Interrupt_MSS_GPIO_tb	[]
Simulation Opti DO File	Top level instance name:	<top>_0</top>	
Waveforms	Generate VCD file		
Vsim comma	VCD file name:	power.vcd	
Simulation Libra SmartFusion2	Verilog 2001	Select Verilog Language Syntax	
	VHDL 2008	Select VHDL Language Syntax	
	User defined DO file: DO command parameters:		
	bo command parameters.		

Figure 4-2 • Project Settings Dialog Box

- 3. Select Vsim commands in the Project Settings dialog box to set Vsim command options.
- 4. Change the default simulation resolution from 1fs to 1ps in the Resolution field. Ignore the warning message. This will speed up the simulation. See Figure 4-3.

Device Device I/O Setti Preferred HDL T Design Flow Simulation Opti DO File	SDF Timing Delay Minimum Typical Maximum	S Disable Pulse Filtering during SDF based Simulations	Save Restore Defaults
Waveforms Vsim comma Simulation Libra SmartFusion2	Resolution: Additional options:	1ps -novopt	▲

Figure 4-3 • Vsim command Settings

5. Under **Simulation Options**, click **Waveforms** and click the checkbox to enable the option **Log all** signals in the design (Figure 4-4).



Device Device I/O Setti	Include DO file	Save
Preferred HDL T	ments/Work/Ubero11.3_Quick_Start_UG/Tim_working/Interrupt_MSS_GPIO/simulation/post-layout_wave.do	Restore Defaults
Design Flow		
Simulation Opti	Display waveforms for top_level testbench	
DO File	V Log all signals in the design	
Waveforms		
Vsim comma		
Simulation Libra		
SmartFusion2		

Figure 4-4 • Changing Waveform Options

- 6. Click **Save** to save the changes to the project settings. Click **Close** to close the Project Settings dialog box.
- 7. In the Design Flow window, expand Verify Pre-Synthesized Design and right-click **Simulate** and choose **Open Interactively** to launch ModelSim in GUI mode (Figure 4-5).

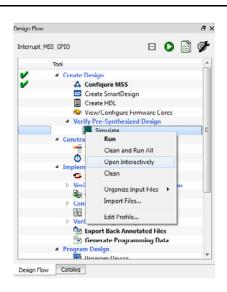


Figure 4-5 • Invoking ModelSim

ModelSim opens and automatically imports a **run.do** macro file that contains the links to the design files and gives simulation commands. The simulator compiles the source files and loads the design. The Wave window appears, as shown in Figure 4-6.



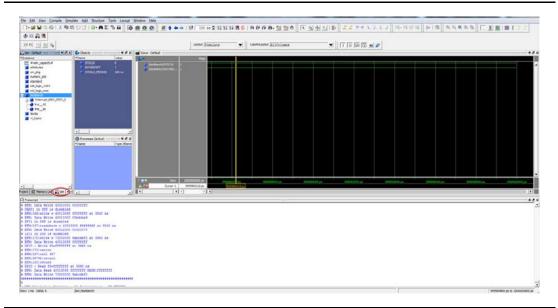


Figure 4-6 • ModelSim Wave Window

You must add additional signals to the Wave window to confirm the design is functioning properly.

 Click the ModelSim sim tab (circled in Figure 4-6). Expand the Design Hierarchy and select Timer_1_0 (Figure 4-7).

Instance	△ Design unit	Design unit type	Visibility	Total coverage
😤 #vsim_capacity#		Capacity	+acc=<	
attributes	attributes	Package	+acc=<	
📕 cm_pkg	cm_pkg	Package	lacc=<	
📕 numeric_std	numeric_std	Package	+acc=<	
📕 standard	standard	Package	+acc=<	
std_logic_1164	std_logic_1	Package	+acc=<	
📕 std_logic_misc	std_logic_mise	Package	+acc=<	•
	testbench(b	. Architecture	+acc-<	• 03
- Interrupt_MSS_GPIO_0	interrupt_m	Architecture	+acc=<	
+ Interrupt_MSS_GPIO_MSS_0	interrupt_m	. Architecture	+acc=<	
	timer_0(def	Architecture	+acc=<	
+ Timer_1_0	timer_1(def	. Architecture	+acc=<	10
	GND	Module	+duc=<	
	VCC	Module	+acc=<	•
	testbench(b	.Process	+acc=<	•
-3 line_56	testbench(b	.Process	+acc=<	•
🗾 textio	textio	Package	+acc=<	• 3
vl_types	vl_types	Package	+acc=<	•

Figure 4-7 • Timer_1_0 Selected in the ModelSim sim Tab

- 9. Click the Objects tab. Ctrl + click to select Aclr, Clock, Q and Tcnt.
- 10. Add the signals to the Wave window: from the **ModelSim** menu, choose **Add > Wave > Selected Signals** (Figure 4-8).

File Edit View	Add Tools	Window	1.00		
Objects	<u>W</u> ave	Selected Signals	-		
🗋 • 🖨 🖶 🐔	List	Signals in <u>R</u> egion Signals in Design		🕯 🔕 • 🚧 🖺 🖪 📙 🛛 🗛 • 🔕	u 🌮
Name	Dataflow +		kind	Mode	
🔷 Arlr	1		Signal	In	
Clock	1		Signal	In	
🔷 Tent	0		Signal	Out	
	0000000	10011100001100	Signal		
NU_0				Internal	
4 NU_1				Internal	
4 NU_2				Internal	
4 NU_3				Internal	
4 NU_4				Internal	-
4 NU_5				Internal	
4 NU_6				Internal	
🦘 NU_/				Internal	
4 NU_8				Internal	
🤣 NU_9				Internal	
4 NU 10				Internal	
4 NU_11				Internal	
12 NU_12				Internal	
13 NU_13				Internal	
NU_14				Internal	
NU_15	0			Internal	
NU_16 NU_17				Internal Internal	
NU 17	0			Internal Internal	
NU_18	0			internal Internal	
NU_19	0			Internal	
NU_1_2	0			Internal	
NU_0_1_2	0			Internal	

Figure 4-8 • Adding Signals to the Wave Window

- 11. Click the ModelSim **sim** tab (circled in Figure 4-6). Expand the Design Hierarchy and select **Timer_0_0**.
- 12. Click the **Objects** tab. Ctrl + click to select **AcIr**, **Clock**, **Q** and **Tcnt**.
- 13. Add the signals to the Wave window: from the **ModelSim** menu, choose **Add > Wave > Selected Signals**.

After the signals are added the Wave window should look like Figure 4-9.

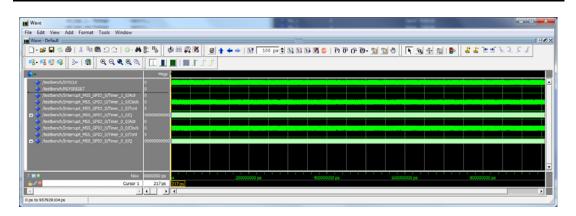


Figure 4-9 • ModelSim Wave Window After Adding Timer_1_0 and Timer_0_0 Signals



14. Observe the operation of the counters to confirm the design is working. Use the zoom buttons to zoom in and out as necessary (Figure 4-8).

Undock the Wave window to make it easier to observe the signals.

Change the radix of the Timer Q output to hex to make it easier to view the values (Figure 4-10).

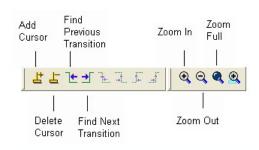


Figure 4-10 • Wave Window Zoom Controls

ModelSim displays waveforms of the Timers, as shown in Figure 4-11.

lave																			- -
Edit View Add Format Tools Window																			
lave - Default																			3 ± #
· 📽 🖬 🛸 🖓 🌡 🐂 📾 오스 이· 🕯	1	🖄 🗸 🞬 🌢	0 1 4		100	ps 🛊 🗎	1 27 2	1 🕅 😂	0 (7)	₽ .	1	0	54	1	D 4	1 L L	₹£:	153	
।•• 🗣 🗣 🛛 ≫ । 🎕 🗍 🔍 🔍 🔍 🧟 🤋		.	1.1																-
•	Msgs																		
(hetsbend)(1913LX (hetsbend)(1913LX) (hetsbend)(1913LX) (hetsbend)(1913LX) (hetsbend)(1913LX)(1915)(1910)(1913LX) (hetsbend)(1913LX)(1915)(1910)(1913LX) (hetsbend)(1913LX)(1915)(1910)(1913LX) (hetsbend)(1913LX)(1915)(1915)(1913LX) (hetsbend)(1913LX)(1915)	0 001178 1																		
	(7621269 ps	8		4000	20 ps			600	o 1 1			1200	00 ps			1600	000 ps		
to 1870956 ps																			F

Figure 4-11 • ModelSim Wave Window

15. You can quit the Simulator after you confirm the Timers are counting. From the ModelSim toolbar **File** menu, choose **Quit** to exit the simulator. Click **Yes** when asked if you want to quit.

Synthesizing the Design using Synplify® Pro

Synplify Pro compiles and synthesizes the design into an EDIF (*.edn) file. Your EDIF Netlist is then automatically translated by Libero SoC into an HDL Netlist. The resulting EDIF (*.edn), Verilog (*.v) or VHDL (*.vhd) files are visible in the **Files** tab under **Synthesis**.

1. In the Design Flow window, right-click **Synthesis** and choose **Open Interactively** to launch Synplify Pro (Figure 4-12).



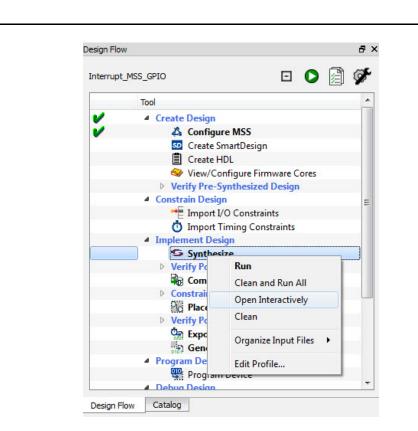


Figure 4-12 • Invoking the Synthesis Tool

You can change the frequency in the Synplify Pro GUI to meet your design requirements. In this design, we are not changing the frequency (Figure 4-13).



* 🐚 🖬 🖏 🗐 🐮 🔇 R R R 9 1 1 1 1 1 1 1 1 1	0 0 4 4 4 6 9 5 5 5 5 5 6 4 4 4 4 4 4 4 4 4 4 5 5 5 5	ୟ ପ୍ରୁ ପ୍ର	થ છા હા		h;								
◎ Run	Synplify Pro®												
-Run	Done: 0 errors, 15 warnings, 30 notes										Searc	h Solvil	
🖏 Open Project	Project Files Design Hierarchy	Project Status	Impleme	ntation	Dire	ctory	Process	View					
没 Close Project	Microsemi SmartFusion2 : M2S025T : F8GA484 : -1	Project Settings											
👍 Add File	 Interrupt_MSS_GPIO_syn] - C:\Users\alex.wong\D Verilog 	Project Name Interrupt MSS GPIO_syn					a second and a second		plementat	ion Name	synth	esis	
Change File	Interrupt_MSS_GPIO_FCCC_0_FCCC.v [work] ->	Top Module Interrupt_MSS_GPIO				SPIO	Re	timing	0				
Add Implementation	Minterrupt_MSS_GPIO_MSS_syn.v (work) ->NO	Resource Shar	-					Fanout Guide 1000				0	
Implementation Options	Interrupt_MSS_GPIO_MSS.v [work] ->WARNIF	Disable I/O Inse	ertion (0				FS	M Compil	er	1		
Add P&R Implementation	 d osc_comps.v [work] ->NOTES: 1 d Interrupt MSS_GPIO_OSC_0_OSC.v [work] ->\ 	0					Run Status	5					
🛕 View Log	Timer_0.v [work] ->NOTES: 2			CPU Time	Rei	al Time	Memory	Date/T	ime				
Frequency (MHz):	 Timer_1.v [work] ->NOTES: 3 Interrupt_MSS_GPIO.v [work] ->WARNINGS: 1 	Compile Input Detailed report	Complete	17	6	0	•	0m	:01s		5/14/2		
• 100 🗘 🔿 Auto Const.	G synthesis	Pre-mapping Detailed report	Complete	2	1	0	0m:00s	0m	:00s	133MB	5/14/2		
Continue On Error FSM Compiler		Map & Optimize Detailed report	Complete	10	8	0	0m:01s	0m	:01s	133MB	5/14/2		
Resource Sharing		Θ			_	2	Area Summi	агу					
Retiming	-	Carry Cells						45	Sequen	tial Cells		42	
		DSP Blocks (M		used)				0	I/O Cell			3	
		Global Clock B Detailed report	uffers					2	LUTs (to	otal_luts)		63	
					_		Index Course	_				_	

Figure 4-13 • Synplify Pro GUI

- 2. Click **Run** to map the design (Figure 4-13). When **Ready** in Synplify Pro changes to **Done:0** errors, the design has been mapped successfully.
- 3. From the **File** menu, choose **Exit** to close Synplify Pro. Click **Yes** if prompted to save changes to the project.

In the Libero SoC Design Flow window, a green check mark adjacent to Synthesis indicates that the design has been synthesized without any errors (Figure 4-14).



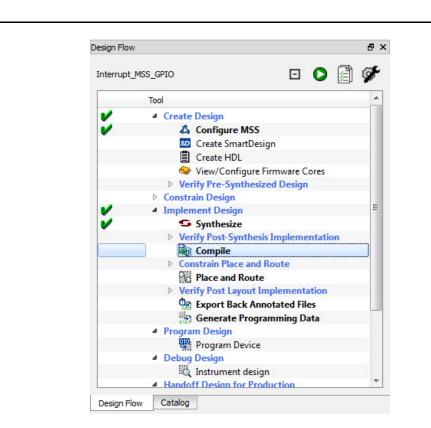


Figure 4-14 • Successful Synthesis



5 – Step 5 - Implementing the Design

The next step is to implement the design. You must assign I/O and timing constraints, run Compile, then Place and Route. Once place and route is complete, you can perform the static timing and power analysis.

In this design, all ports are hard-wired except for GPIO_1_F2M, which must be assigned to an I/O pin.

1. Expand Place and Route. Right-click **I/O Constraints** and choose **Open Interactively** to open the I/O Constraints Editor (Figure 5-1).

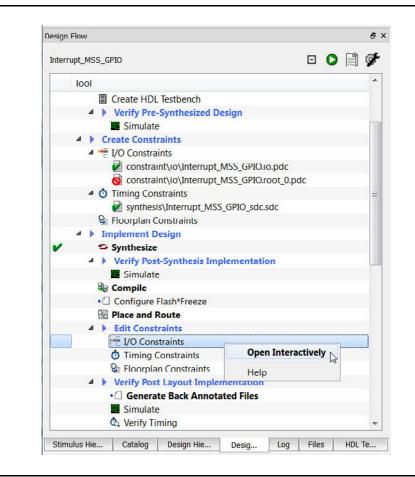


Figure 5-1 • I/O Constraints Editor Invocation

Libero SoC runs Compile. After successful completion, a green check mark appears next to Compile in the Design Flow window. The I/O Editor then appears for you to assign the I/O pins (Figure 5-2).



Ports Pa Port Name DEVRST_N GPIO_1_F2 MMUART_1_1 MMUART_1_1	M Input	Package Viewer I/O Standa LVCMOS		Number 💌 R15			Bank Name 💌	state in Flash*Freeze 💌	Resistor Pull 💌	lable in Flash*Free▼
GPIO_1_F2	M Input		2 10	R15						
MMUART_1_I		LVCMOS	22 810		V	ADLIB:SYSRESET				
	RXD Input		010 020	-	V	ADLIB:INBUF	Bank2	TRISTATE	None	No
MMUART_1_1		LVCMOS:	33 16	*	V	ADLIB:INBUF	Bank1	TRISTATE	None	No
	TXD Outpu	LVCMOS:	J20		v	ADLIB:TRIBUFF	Bank1	TRISTATE	None	No
		m	J22 K6 K7 K15							
Messages 😮	Errors 🔺 Warni	ngs 🕕 Info	K16 K17 K18							

Figure 5-2 • Pin Assignment of Port GPIO_1_F2M

- In the I/O Editor, assign Pin K18 to the Port GPIO_1_F2M and check the Locked checkbox. Choose Commit and Check to save the pin assignment. Correct any errors that are reported in the Log window.
- 3. From the File menu, choose Exit to close the I/O Pin Editor.
- 4. Right-click **Place and Route Layout** in the Design Flow window and choose **Configure Options** to set the Place and Route Options (Figure 5-3).

Configuring	
Configuration	
Timing-driven	
Power-drive	n 🔲
High Effort L	.ayout
Incremental Lay	vout
Help OK	Cancel

Figure 5-3 • Layout Options

- 5. Click the checkbox to enable Timing-Driven place and route and leave all other options unchecked. Click **OK** to close the Configuring Options dialog box.
- 6. Double-click Place and Route.

Libero generates the *io.pdc file (Pin Assignment file) and passes the file to Place and Route. A green check mark appears next to Place and Route to indicate successful completion (Figure 5-4).



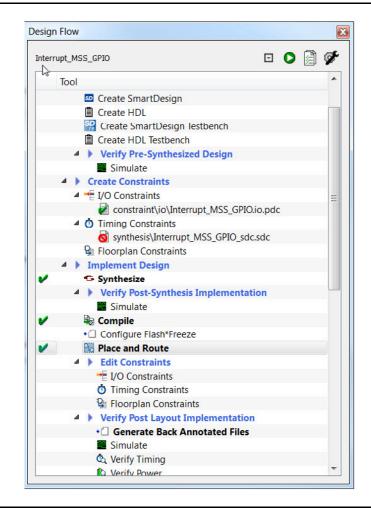


Figure 5-4 • Successful Compile and Place and Route

Timing Analysis

The next step is to perform static timing analysis using SmartTime. SmartTime reads your design and displays post-layout timing information (pre-layout, if invoked before place and route). For more information, see the Help included with the software.

SmartTime includes a Constraints Editor and a Timing Analyzer.

 In the Design Flow window, right-click Verify Timing and choose Open Interactively to open SmartTime. Confirm that the fabric clock (mss_ccc_glb) meets the timing requirement (10 MHz). (Figure 5-5).



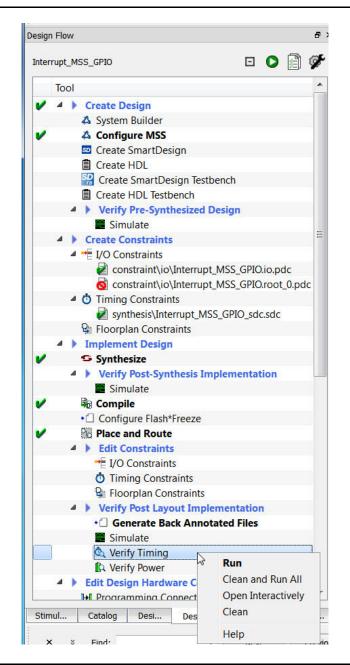


Figure 5-5 • Verify Timing

 The Maximum Delay Analysis View window opens. Maximum Delay Analysis checks for any setup violations. A green check mark next to **Register to Register delay** indicates that there are no setup violations. Select the **Register to Register** path set for the GL0_net_inferred_clock domain (Figure 5-6).



e Edit View Tools Help												
um Delay Analysis View												
												_
Analysis for scenario Primary												
🖓 Summary 🔺	Fr	om =				то -						
✓ ✓ ● FCCC_0/CCC_INST/INST_CCC_L ✓ Register to Register	0	ustomize table							Apply Filter	Store Filter	keset Filter	-
External Setup Clock to Output		Source Pin	Sink Pin	Delay (ns)	Slack (ns)	Arrival (ns)	Required (ns)	Setup (ns)	Minimum Period (ns)	Skew (ns	9	
Register to Asynchronous External Recovery	1	Timer_0_0/Qaux[11]:CLK	Interrupt_MSS_GPI0_MSS_0/	2.487	95.899	5.704	102.603	0.700	3.101		-0.086	ŀ
Asynchronous to Register	2	Timer_0_0/Qaux[12]:CLK	Interrupt_MSS_GPIO_MSS_0/	2.386	97.009	5.594	102.603	0.700	2.991		-0.095	
Register to Register External Setup	3	Timer_0_0/Qaux[10]:CLK	Interrupt_MSS_GPIO_MSS_0/	2.369	97.017	5.586	102.603	0.700	2.983		-0.086	
Clock to Output		Timer_0_0/Qaux[17]:CLK	Interrupt_MSS_GPIO_MSS_0/		97.092	5.511	102.603	0.700	2.908		-0.086	
		Timer_1_0/Qaux[5]:CLK	Timer_1_0/Qaux[17]:0	2.572	97.126	5.789	102.915	0.254	2.874		0.048	
90		Timer_1_0/Qaux[2]:CLK	Timer_1_0/Qaux[17]:D	2.567	97.129	5.786	102.915	0.254	2.871		0.050	
72	7	Timer_0_0/Qaux[6]:CLK	Timer_0_0/Qaux[19]:D	2.577	97.134	5.772	102.906	0.254	2.866		0.035	•
	Na	ime				Туре	Net					1
54		Summary										1
36		data required time										
36		data arrival time										
~		slack										
		Data_arrival_time_calculati										
18		FCCC_0/CCC_INST/INST_C										
		FCCC_0/CCC_INST/INST_C	CC_IP:GL0			Clock source						
						Clock generati	on					
96.4835 96.899 97.3145 97.73 98.1		FOOD AVELO BICT.A.					FCCC 0/C1	0				1
slack distribution(ns)	•										,	

Figure 5-6 • SmartTime Max Delay Timing Analysis

3. Double-click one of the source pins to view detailed timing analysis for the selected path in the path details (as shown in Figure 5-7).

ile Edit View Tools Help 김 오 오 강 글 도 중 영 저					10			
mmary for path m: Timer, 0, 0/Qawi [10]:CLK interrupt, MSS, GPD, MSS, QAK3B, QAK3B, DIST/INST_MSS, 025_JPJ?2H_J ita Required Time (m) Data Arrival Time (m) Slack (m) 5.566 (97.017)	HTERUMPT[0]		Can 	Delay L315	R	th Prof	THE DEBY 63.69%	
ame	Туре	Net	Macro	Op	Delay 1	Total	Fanout Edge	-
Data_arrival_time_calculation								
FCCC_0/CCC_INST/INST_CCC_IP:GL0					0.000 0	000.		
FCCC_0/CCC_INST/INST_CCC_IP:GL0	Clock source			+ 1	0.000 0	000.	r	
	Clock genera					045		
FCCC_0/GL0_INST:An	net	FCCC_0/GL0_net				2.045	r	
FCCC_0/GL0_INST:YEn	cell		ADLIB:GBM				5 f	
FCCC_0/GL0_INST/U0_RGB1_RGB2:An	net	FCCC_0/GL0_INST/U0_YWn_GEast				478	f	
FCCC_0/GL0_INST/U0_RGB1_RGB2:YR	cell		ADLIB:RGB				15 r	
Timer_0_0/Qaux[10]:CLK	net	FCCC_0/GL0_INST/U0_RGB1_RGB2_rgbr_net_1				.217	r	
Timer_0_0/Qaux[10]:Q	cell		ADLIB:SLE				2 f	
Timer_0_0/Tcnt_5:A	net	Timer_0_0/Qaux[10]				.770	f	
Timer_0_0/Tcnt_S:Y	cell		ADLIB:CFG2				1 f	
Timer_0_0/Tcnt_16:D	net	Timer_0_0/Tcnt_5		+ 1	0.303 4	.237	1	_
RCC_ADALART + N= GRA	POCC. 67							

Figure 5-7 • Detailed View of the Register Path in SmartTime



- 4. From the **SmartTime Tools** menu, choose **Minimum Delay Analysis** to open the Minimum Delay Analysis View window. Minimum Delay Analysis checks for hold time violations. A green check mark next to **Register to Register delay** indicates no hold time violations.
- 5. From the File menu, choose Exit to close SmartTime.

Power Analysis

SmartPower enables you to estimate the power consumption in your design. This enables you to make adjustments to reduce power consumption.

1. From the Design Flow window, right-click **Verify Power** and choose **Open Interactively** to start power analysis (Figure 5-8).

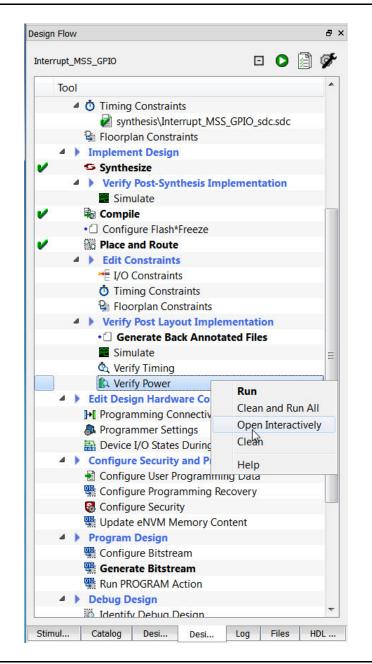


Figure 5-8 • Power Analysis



Figure 5-9 shows the SmartPower Analysis window.

Pomains Panaly PPe > pge Frequency>=0	Total 2.12 mW	Static 14.74 mi how @ Char Power Usa	w 57.3	namic 79 mW		
n 72 r/pe ▼ bge[Frequency>=0 ▼ Frequency>=0	Total 2.12 mW S	Static 14.74 mi how @ Char Power Usa	W 57.3 Grid Grid Grid Grid Grid Grid Grid Grid	namic 79 mW		
rpe ▼] bge[Frequency>=0 ▼] Frequency>=0	8.12 mW S	14.74 ml	W 57.3 Grid Grid Srid	79 mW		
pe v sge[Frequency>=0 v] Frequency>=0	8.12 mW S	14.74 ml	W 57.3 Grid Grid Srid	79 mW		
pe v sge[Frequency>=0 v] Frequency>=0	S Built-in Blocks	how () Char () Power Usa	Grid Srid	5		
bge[Frequency>=0 ▼ Frequency>=0	Built-in Blocks	Power Usa	nge Not 0.009 mW 9.165 mW VO 0.053 mW VCore State 1.97 mW VPP State	be		
Frequency>=0	Buit-in Blocks 48.072 mW		Gate 9.165 m/V VO 0.053 m/V Core Statu 10.27 m/V Banks Stat 197 m/V VPP Status	be		
Frequency>=0	Buit-in Blocks 48.072 mW		Gate 9.165 m/V VO 0.053 m/V Core Statu 10.27 m/V Banks Stat 197 m/V VPP Status	be		
Frequency>=0	Built-in Blocks 48.072 mW		0.053 mVV Core Static 10.27 mVV Banks Static 1.97 mVV VPP Static	be		
Frequency>=0	48.072 mW		1.97 mWV VPP Static			
Frequency>=0						
Frequency>=0						
Frequency>=0					Hic	de Annotation Cov
	VCD Annotation	Manual Annotation	SmartTime Constraint	Vectorless Estimation	Fixed Values	Total
	0(0%)	0(0%)	8(100%)	0(0%)	0(0%)	8
puts	0(0%)	0(0%)	0(0%)	0(0%)	43(100%)	43
nal outputs	0(0%)	0(0%)	0(0%)	0(0%)	286(100%)	286
ets	0(0%)	0(0%)	0(0%)	0(0%)	16(100%)	16
ıts	0(0%)	0(0%)	0(0%)	0(0%)	1(100%)	1
	0(0%)	0(0%)	0(0%)	0(0%)	0(0%)	0
	0(0%)	0(0%)	0(0%)	0(0%)	506(100%)	506
ons : Typical Operating Conditions				Value		
				1.2 V		
VDD				2.5 V		
VDD MDDR_PLL_VDDA				2.5 V		
MDDR_PLL_VDDA				2.5 V		
-	is : Typical Operating Conditions Junction Temperature Process VDD	0(0%) s: Typical Operating Conditions Junction Temperature Process VDD MD0R_PLL_VDDA	0(0%) 0(0%) s: Typical Operating Conditions Junction Temperature Process VDD MDDR_PLL_VDDA	0(0%) 0(0%) 0(0%) is : Typical Operating Conditions Junction Temperature Process VDD MDDR_PLL_VDDA	0(0%) 0(0%) 0(0%) is : Typical Value Junction Temperature 25 C Process Typical VDD 1.2 V MDDR_PLL_VDDA 2.5 V	0(0%) 0(0%) 0(0%) 506(100%) s: Typical

Figure 5-9 • SmartPower Analysis Window

SmartPower includes the following tabs (as shown in Figure 5-9):

- **Summary** Displays the total power consumption, temperature and voltage operating conditions, battery capacity in mA/hr, and the projected battery life.
- **Domains** Displays a list of existing domains with their corresponding clock and data frequencies. Use the **Domains** tab to set different clock frequencies and observe the effect on power consumption.
- Analysis Displays detailed hierarchical reports of the power consumption.
- Frequencies Used to attach switching frequency attributes to the interconnects of the design.
- Probabilities Used to control the probabilities for all pins.

SmartPower enables you to globally visualize power consumption and potential power consumption problems within your design. You can then make adjustments, when possible, to reduce power consumption to meet your design's power requirements.

For more information, see the Help included with the software.

2. Select File > Exit to close SmartPower.



Back-Annotation

Back-Annotation generates a *.sdf file that contains timing information for your design. It is used for postlayout and timing simulation.

1. From the Design Flow window, right-click **Generate Back Annotate Files** and choose **Run** to extract post-layout timing delay from your design for simulation. (Figure 5-10)

iterrupt_MSS_GPIO		9
Tool		-
 Configure Flash*Freeze 		
Place and Route		
Edit Constraints		
F I/O Constraints		
Timing Constraints		
🚱 Floorplan Constraints		
Verify Post Layout Implementation		
• Generate Back Annotated Files	Dun .	
Simulate 📃	Run	Ξ
🖏 Verify Timing	Clean and Run All	
🛱 Verify Power	Clean	
 Edit Design Hardware Configuration Programming Connectivity and Inter 	Configure Options	_
	Help	
Programmer Settings	9	
Programmer Settings Device I/O States During Programming		
Device I/O States During Programming		

Figure 5-10 • Generated Back Annotated Files

Post-Layout Simulation

A green check mark appears next to the Generate Back Annotated Files in the Design Flow window when a back annotated file is generated successfully.

- 2. Click Project > Project Settings > Simulation Options > DO File to open the dialog box.
- 3. Check Use automatic DO File.
- 4. Enter 1ms for Simulation runtime.
- 5. Click Save and then click Close.
- Click Project > Project Settings > Simulation Options > Vsim Commands to open the dialog box.
- 7. In Simulation Resolution, enter **1ps**. This will speed up the simulation significantly. If you see warnings when you make the change, they can be ignored. The default value is 1 fs.
- 8. Click **Project > Project Settings > Simulation Options > Waveforms** and in the open window, uncheck **Log all signals in the design**. This will speed up the simulation run. After the simulation run, you may add to the waveform window the signals you want to examine.

For Verilog projects, a wave.do file is available for you to include. Check **Include Do File**. Browse to the location of the post-layout_wave.do file you have extracted from the source files included in



this tutorial. Select the file and click **Open**. This file will be loaded into the simulator's waveform window and the signals will be displayed during simulation (Figure 5-11).

9. In the Libero Design Flow window, expand Verify Post Layout Implementation, right-click Simulate and choose Open Interactively to open ModelSim.

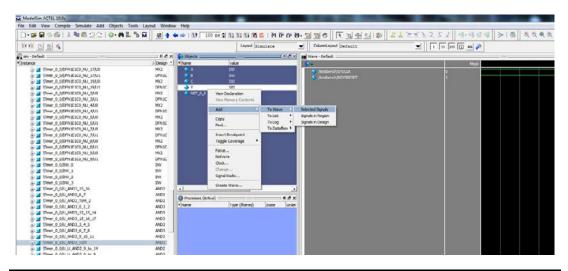


Figure 5-11 • Adding Signals to the Wave Window in ModelSim

10. In ModelSim, let the simulation run for 100 - 300 ms of run time. This may take a a few hours to complete. You may want to stop the simulation (Simulate > Break) after you have verified that the lower bits of the two counters are counting/toggling correctly in the waveform window. Use the zoom buttons to change the scale and view details in the waveforms (Figure 5-12).

Note that the Tnct signal of Counter_0 asserts at about 100 ms and Tnct of Counter1_0 asserts at about 200 ms.

Figure 5-12 shows the waveform window after 300 ms of simulation runtime.

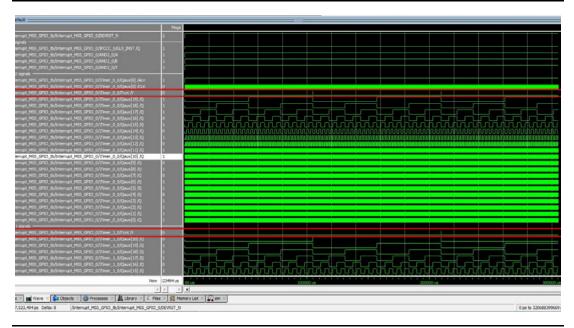


Figure 5-12 • Post-Layout Simulation Window after 300 ms of simulation time



Note: If ModelSim reports signals not found, you must edit the post-layout_wave.do file. Open the file in the Libero Text Editor. Replace the testbench name Interrupt_MSS_GPIO_tb in the post-layout_wave.do file with your testbench name. Replace the Component name Interrupt_MSS_GPIO_0 with your top level component name. Save the file and restart ModelSim.

You can stop the simulation from the ModelSim menu (Simulate > Break).

11. Close ModelSim.

Programming

In this step you will launch FlashPro and program the SmartFusion2 device on the Evaluation Board. Refer to the SmartFusion2 Evaluation Kit User Guide for instructions on how to set up the hardware for programming.

- Connect one of the mini USB cables between the JTAG Programming Header and a USB port on your PC. Install the FlashPro4 drivers if prompted. The drivers are located in the <drive>:\<Microsemi \<Libero SoC v11.4 installation>\Designer\Drivers folder.
- 2. Connect the AC adaptor to the 12 V Power Supply Input on the board and plug the Adaptor to the AC wall outlet.
- 3. Turn the on/off switch on the board to the ON position.
- 4. In the Design Flow window, expand Program Design.
 - To program the device directly, double-click **Run PROGRAM Action**.

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

The following message is visible in the Libero SoC log window when the device is programmed successfully: **Chain Programming PROGRAM PASSED.**

A green check mark next to **Run PROGRAM Action** appears in the Design Flow window to indicate that the programming completed successfully (Figure 5-13).

🗋 🧀 🖆 🕰 🗅 💽 🔂 👘	Reports # X Timer 0 th pre	with simulation log AX	testhench v R X	run do e X	testhench postlavout simulation log A	9 ×
sign Flow	 Interrupt, MSS, GPIO rep. Interrupt, MSS, GPID rep. Inter	<pre>'epen_project' comm 'ide'rod Therry LM30 'ast_programming fi 'ast_programming fi 'enable_device' con 'sst_programming ac 'ammor '\$2075' i 3c 'ammor '\$2075' i 3c''''''''''''''''''''''''''''''''''</pre>	ashPro4 cong/Documents/Nor and succeeded. MCCONG/Documents/Nor and succeeded. GPIO, CHECKIMN : le' command succe an Chain tion' command succeeded. vice 'M200257' : vice 'M20057' : vice 'M	<pre>iork/Liberoll.3_Qu price; PBB_VERBI; edded. for device 'sd202 cceeded. been selected on ; in PASED. Executing action ExPORT 10C ENABLE EXFORT CRCERM; [1 TEMPORADE: ROOM EXPORT 10N: [8] = Finished Thu May ExPORT Daw[128] = Finished Thu May Expont Daw[128] =</pre>	57'. programmer AND an externall PROGRAM RHULVF(32] = 007a6644] = 0 [3] = 2 = 66 7a Array 09-4aa010000cd5s00c30000c0 15 15:18:03 2014 (Elapsed	errupt_ nterrup ly prov

Figure 5-13 • Design Flow Window After Programming

Exporting Firmware

1. From the Libero SoC Design Flow window, right-click **Export Firmware** and choose **Export Firmware**.



interru	IPT_MSS_GPIO 🖸 🖸	ø
T	lool	
	🖏 Verify Timing	
	R Verify Power	
	Edit Design Hardware Configuration	
	Programming Connectivity and Interface	
	Programmer Settings	
	Device I/O States During Programming	
	Configure Security and Programming Options	
	Security Policy Manager	
	🚆 Update eNVM Memory Content	
	Program Design	
	Sitstream Configuration	
~	🎇 Generate Programming Data	
-	Run PROGRAM Action	
	Debug Design	i n
	🖏 Identify Debug Design	
	SmartDebug Design	
- 4	Handoff Design for Production	
	🛃 Export Programming File	
	Export Data for Programming Job	
	 Export Pin Report 	1
	 Export BSDL 	
	Export IBIS Model	
- 4	Handoff Design for Firmware Development	
	Sconfigure Firmware Cores	
~	🖳 Export Firmware	
	Export Firmw	ar

Figure 5-14 • Export Firmware

- 2. Accept the default location for the destination of the firmware exported.
- 3. Check Create Project and choose SoftConsole from the drop-down menu (Figure 5-15).

Location:	_Start_UG\TimC\Interrupt_MSS_GPIO
Create project	SoftConsole3.4 💌

Figure 5-15 • Export Firmware Dialog Box



6 – Software Implementation

Step 1 – Invoking SoftConsole

Note: A software patch SP1 is available for SoftConsole 3.4. Before your proceed, install the SP1 patch first from

http://www.soc.microsemi.com/download/reg/download.aspx?p=f=SoftConsolev34_SP1.

1. Double-click the SoftConsole icon on your Desktop or choose Start > All Programs > Microsemi SoftConsole v.3.4.

SoftConsole opens and prompts for a Workspace Location (Figure 6-1).

folder called a workspace.
GPIO_MSS_CM3 ▼ Browse
OK Cancel

Figure 6-1 • SoftConsole Workspace

- Make sure the Workspace location is set to: <Proj_location>\SoftConsole\Interrupt_MSS_GPIO_MSS_CM3.
- 2. Expand Interrupt_MSS_GPIO_MSS_MSS_CM3_0_app and double-click main.c to open it.
- Delete the existing code in main.c and copy the code below and paste it in the main.c file. Note: The main.c file is included in the Source File you have extracted.



```
* This main.c file invokes three fabric to MSS Interrupts (two synchronous from timers and
 * one asynchronous from the switch of the SmartFusion2 Evaluation Kit).
 ^{\star} Interrupt messages are output from the MSS to the Terminal Emulator via MMUART of
 * the MSS.
/*Including Directories*/
#include<stdio.h>
#include "mss_uart.h"
#include "mss_gpio.h"
#include "cortex_nvic.h"
/*_____
 Messages displayed over the UART.
-----*/
const uint8 t g greeting msg[] = "\r\n\r\n
This example project demonstrates the use of the SmartFusion2\r\n\
Fabric and GPIO interrupts.\r\n";
/*Interrupt Handlers*/
/* GPIO 0 Interrupt Handler */
/\,\star\, This Interrupt handler executes upon the occurrence of
  GPIO 0 interrupt, which is from the Timer 1 in the FPGA
  Fabric. This function prints source of the interrupt to the
  hyperterminal */
void GPIO0_IRQHandler( void )
 {
const uint8_t tim2[] = "\n\r Timer 1 Interrupt occurred - GPIO 0 \n\r";
MSS_UART_polled_tx (&g_mss_uart1, tim2, sizeof(tim2));
MSS_GPIO_clear_irq( MSS_GPIO_0 );
NVIC_ClearPendingIRQ( GPIO0_IRQn );
 }
/* GPIO 1 Interrupt Handler */
/* This Interrupt handler executes upon the occurrence of
  GPIO 1 interrupt, which is from the Switch.
  This function prints source of the interrupt to the
  hyperterminal */
void GPI01_IRQHandler( void )
 {
const uint8_t sw1[] = "\n\r Switch Interrupt occurred - GPIO 1 \n\r";
MSS_UART_polled_tx (&g_mss_uart1, sw1, sizeof(sw1));
MSS_GPIO_clear_irq( MSS_GPIO_1 );
NVIC_ClearPendingIRQ( GPIO1_IRQn );
 }
/* Fabric Interrupt Handler */
/* This Interrupt handler executes upon the occurrence of
  fabric interrupt, which is from the Timer 0.
  This function prints the source of the interrupt to the
  hyperterminal */
void FabricIrq0_IRQHandler( void )
{
      const uint8_t tim1[] = "\n\r Timer 0 Interrupt occurred - FABINT \n\r";
      MSS_UART_polled_tx (&g_mss_uart1, tim1, sizeof(tim1));
      NVIC_ClearPendingIRQ (FabricIrq0_IRQn);
 }
/*_____
```



```
Display greeting message when application is started.
 */
static void display_greeting(void)
{
   MSS_UART_polled_tx(&g_mss_uart1, g_greeting_msg, sizeof(g_greeting_msg));
}
Function Main
             **********
*****
int main()
{
/*UART initialization*/
MSS_UART_init(
&g_mss_uart1,
MSS_UART_57600_BAUD,
MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT );
   /* Display greeting message */
   display_greeting();
/* Enabling of GPIO 0, GPIO 1 and Fabric Interrupts*/
   /* MSS GPIO_0 used for Timer 2 input */
   NVIC_EnableIRQ(GPI00_IRQn);
MSS_GPIO_config( MSS_GPIO_0, MSS_GPIO_INPUT_MODE | MSS_GPIO_IRQ_EDGE_POSITIVE );
MSS_GPIO_enable_irq( MSS_GPIO_0 );
/* Configure MSS GPIO_1 - used for switch input */
NVIC_EnableIRQ(GPIO1_IRQn);
MSS_GPIO_config( MSS_GPIO_1, MSS_GPIO_INPUT_MODE | MSS_GPIO_IRQ_LEVEL_LOW );
MSS_GPIO_enable_irq( MSS_GPIO_1 );
/* Enable Fabric_0 IRQ - used for Timer 1 input */
NVIC_EnableIRQ (FabricIrq0_IRQn);
for(;;)
{
}
return 0;
}
End of function Main
```

4. From the **Project** menu, choose **Clean** to perform a clean build. Accept the default settings in the Clean dialog box and click **OK** (Figure 6-2).



	Clean projects selected below
Interrupt_MSS_GPIO_MS	S_MSS_CM3_0_app
🔲 😂 Interrupt_MSS_GPIO_MS	S_MSS_CM3_0_hw_platform
Start a build immediately	
Start a <u>b</u> uild immediately Build the entire <u>w</u> orkspace	

Figure 6-2 • Settings for a Clean Build

5. Click the **Problems** tab and make sure there are no errors and warnings. Correct errors before you continue (Figure 6-3).

File Edit Source Refactor Navigate Search Proje	t Run Window He	lp		
□ - </th <th>• 83 • 63 • 63 • \$> \$> • \$ •</th> <th>% - ⊗ -</th> <th>. 🖪 [</th> <th>占 Resource 🚿</th>	• 83 • 63 • 63 • \$> \$> • \$ •	% - ⊗ -	. 🖪 [占 Resource 🚿
🗄 Problems 😣 🧟 Tasks) 🗳 Console) 💷 Properties) 💈	F Debug			□ ₽
0 items				
Description	R	esource	Path	Location
				_
				-
	1			•
				P

Figure 6-3 • Softconsole Problems tab



Step 2 - Configuring Hyper Terminal/Other Terminal Emulator Programs

Prior to running the application program, you need to configure a terminal emulator program (HyperTerminal, included with Windows®) on your PC. Perform the following steps to use the SmartFusion2 Evaluation Kit Board:

- Connect a second mini USB cable between the USB connector (J14) on the SmartFusion2 Evaluation Kit Board and a USB port of your computer. If Windows prompts you to connect to Windows Update, select No, not at this time and click Next.
- If the USB Serial Ports are automatically detected (displayed in Device Manager), as shown in Figure 6-4, it means that the USB-UART driver is already installed on your PC. Four consecutively numbered COM ports are listed (COM 4 through COM7, as in Figure 6-4). For your PC, the four COM port numbers may be different.
- Note: If COM ports are not detected in the Device Manager, see "Appendix A Step 3 Installing Drivers for the USB-UART" to install the USB-UART driver before proceeding.

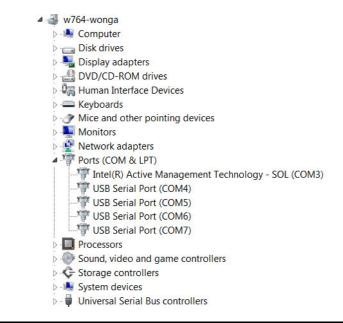


Figure 6-4 • Device Manager Listing USB Serial Port

3. From the Windows Start menu, choose Programs > Accessories > Communications > HyperTerminal. This opens HyperTerminal. If your PC does not have HyperTerminal, use any free serial terminal emulation program, such as PuTTY or Tera Term. Refer to the Configuring Serial Terminal Emulation Programs tutorial for configuring HyperTerminal, Tera Term, and PuTTY.



4. Enter **Hyperterminal** in the **Name** field in the Connection Description dialog box and click **OK** (Figure 6-5).



Figure 6-5 • New Connection

 Choose the appropriate COM port (to which USB-RS232 drivers are pointed) from the Connect using drop-down list and click OK (Figure 6-6). Different systems may show different COM Port#. Choose the highest COM port displayed (COM7 in this case, as displayed in the Device Manager in Figure 6-4)..

Connect To	? 🛛
Ryperter	minal
Enter details for	the phone number that you want to dial:
Country/region:	India (91) 💌
Ar <u>e</u> a code:	91
Phone number:	
Connect using:	Сом7
	OK Cancel

Figure 6-6 • Selecting the COM Port

- 6. Set the following in the COM Properties window and click **OK** (Figure 6-7):
 - Bits per second: 57600
 - Data bits: 8
 - Parity: None
 - Stop Bits: 1



- Flow control: None

Port Settings	
Bits per second:	57600
<u>D</u> ata bits:	8
<u>P</u> arity:	None
<u>S</u> top bits:	1
Elow control:	None

Figure 6-7 • Setting the COM Properties

7. Click **OK** to close the HyperTerminal Properties dialog box.

Alternatives to Hyper Terminal

Hyper Terminal is no longer a standard feature on newer Windows installations. One alternative to the Hyper Terminal is the Tera Term, available for download from:

http://ttssh2.sourceforge.jp/

Another alternative is PuTTY, available for download from: http://en.softonic.com/s/putty-download-windows-7

Configuring Tera Term

To configure Tera Term after installation:

- 1. Click the Tera Term icon to launch Tera Term.
- 2. When the New Connection dialog box appears (Figure 6-8), select Serial.
- 3. From the drop-down list of ports, choose the highest COM # from the group of four consecutive COM ports.
- Note: Only one COM Port is activated on the SmartFusion2 kit. In this case, select COM7:USB Serial PORT (COM7). Your PC may have different COM numbers.



© TCP/IP	Host:	myhost.exar	nple.com	
	Service:	 History Telnet SSH Other 	TCP port#: 22 SSH version: SSH2 Protocol: UNSPEC	
Serial	Port:	COM3: Intel	(R) Active Management Te	
	ОК	COM4: USB	R) Active Management Te Serial Port (COM4) Serial Port (COM5)	chnology - SOL (CON

Figure 6-8 • Configuring the Serial Port in Tera Terminal

- 4. Click **OK** to close the Tera Term: New connection dialog box.
- 5. In the new pop-up Tera Terminal, click **Setup** and choose **Serial Port**. Enter the configuration as follows (Figure 6-9):
 - Port: COM7
 - Different systems may display different COM Port#. Only one COM port is activated on the Board for serial communication use. Always select the Port with the highest COM# (COM7 in this case).
 - Baud Rate: 57600
 - Data: 8 bit
 - Parity: None
 - Stop: 1 bit
 - Flow Control: None

Tera Term: Serial port setu	nb
Port:	СОМ7 • ОК
Baud rate:	57600 -
Data:	8 bit 🔹 Cancel
Parity:	none 🔹
Stop:	1 bit 🔹 Help
Flow control:	none 👻
Transmit dela 0 mse	y c/char 0 msec/line

Figure 6-9 • Tera Term Serial Port Setup

6. Click **OK** to close the Tera Term: Serial port setup dialog box.



7. Click **Setup** and choose **Windows**. The Tera Term: Window setup dialog box appears (Figure 6-10).

Cursor shape Block Vertical line Hide title bar Hide menu bar 16 Colors (PC style) 16 Colors (aixterm style) 256 Colors (xterm style) 256 Colors (xterm style) Enable bold font Color Color	Cance Help
	0 lin
Text Attribute Normal Sackground Reverse	
R: 255	
G: 255 ·	
B: 255 ·	

Figure 6-10 • Tera Term Window Setup Dialog Box

8. Click **Reverse** to change the background to white (black text on a white background) for easy viewing.

Step 3- Debugging the Application Project Using SoftConsole

To debug the application project using SoftConsole:

1. From the **Run** menu in SoftConsole, choose **Debug Configurations**. The Debug Configurations dialog box appears.



2. Double-click **Microsemi Cortex-M3 RAM Target** to display the Options for the RAM target (Figure 6-11).

reate, manage, and run configurations			Ś
 Image: Solution of the second state of the second sta	Name: Interrupt_MSS_GPIO_MSS_MSS_CM3_0_app Debug Main >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>		vse
Filter matched 8 of 11 items		Apply Re	<u>v</u> ert

Figure 6-11 • Debug Configurations Dialog Box

- 3. Confirm that the following appear on the Main tab in the Debug Configuration dialog box:
 - Name: Interrupt_MSS_GPIO_MSS _MSS_CM3_app Debug
 - **Project**: Interrupt_MSS_GPIO_MSS_MSS_CM3_app
 - C/C++ Application: Debug\ Interrupt_MSS_GPIO_MSS _MSS_CM3_app
- 4. Click Apply and Debug.
- 5. Click **Yes** when prompted for **Confirm Perspective Switch** (Figure 6-12). This displays the Debug view mode.

SC Con	rm Perspective Switch
\bigcirc	This kind of launch is associated with the Debug perspective.
	This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.
*	Do you want to open this perspective now?
🗖 <u>R</u> en	nember my decision
	Yes <u>N</u> o

Figure 6-12 • Confirm Perspective Switch



C Debug - Interrupt_MSS_GPIO_MSS_MSS_CM3_0_app/main.c - File Edit Source Refactor Navigate Search Project R	-					
📬 • 🗟 👜 🖹 🏇 • 🕥 • 💁 • 🤌 • 🛛	🍠 🧏 🕶 🖓 🕶	\$~ \$	腔 <mark>梦 Debug</mark> 暍 C/C+-			
🏇 Debug 🛿	- D (M=	Variables 🛛 💊 Breakpo	oints 🚻 Registers 🛋 Modules 🛛 🗧 🗖			
🙀 🖉 🗇 💷 💷 🖓 🐘 🔤 🔤	▶ 😵 🖄 👘		🖾 🏘 🖻 🔯 📽 💥 🏹			
🚀 Embedded GDB (11/8/11 6:01 PM) (Suspended)	^ Na	ime	Value			
Thread [1] (Suspended) 1 main() f(misraceminri)interrunt mer anio)	oftconcolo					
I main() f:\microsemiprj\interrupt_mss_gpio\softconsole\i F:\Microsemi\Libero_v9.1\SoftConsole\Sourcery-G++\bin\arm-nc						
<	• •		-			
🖻 main.c 🖾 💽 -1 < symbol is not available>			🗝 🗖 🔚 Outline 🛛 👘 🖻 🗖			
76			^ ↓ <mark>1</mark> 2 😿 🖋 ● 🗰 🤇			
<pre>77 /* Enabling of GPIO 0, GPIO 1 and Fa 78 NVIC EnableIRQ(GPIO0 IROn);</pre>	abric Interrupt	ts*/	📲 stdio.h 🧳			
79 MSS GPIO config(MSS GPIO 0, MSS GPI	IO INPUT MODE	MSS GPIO IRO EDGE	mss_uart.h			
<pre>80 MSS_GPIO_enable_irq(MSS_GPIO_0);</pre>		· ~_ ·	- Inss_gpio.h Inss_gpio.h Inss_gpio.h Inss_gpio.h			
81			GPIOU_IKQHandler(void) : GPIO1 IRQHandler(void) :			
82 NVIC_EnableIRQ(GPI01_IRQn); MSS_CDIO_config(_MSS_CDIO_1_MSS_CDI	TO TNIDUT MODE	I MEE COTO TRO PROP	Fabric IRQHandler(void):			
<		4	∢			
🖳 Console 🛿 🖉 Tasks 🚼 Problems 🚺 Executables 🚺	Memory	🗙 🔆 📄 🚮	🔜 🖳 🕞 🚝 🛃 🛃 ▼ 📑 🗖 🗖			
nterrupt_MSS_GPIO_MSS_MSS_CM3_0_app Debug [Microsemi Co	ortex-M3 RAM Target]] F:\Microsemi\Libero_v9.1\S	oftConsole\Sourcery-G++\bin\arm-none-eabi			
<pre>78 NVIC_EnableIRQ(GPI00_IRQn);</pre>			· · · · · · · · · · · · · · · · · · ·			
<			4			
□ [◆] Writable	Smart Insert	78:1				

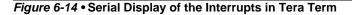
Your Debug Perspective should resemble Figure 6-13:

Figure 6-13 • Debug Perspective

6. Click **Run** and choose **Resume** to run the application.

The interrupt sequence messages are displayed in the terminal program window. Press the SW3 switch on the SmartFusion2 Evaluation Kit and observe the message in the serial terminal that indicates the switch interrupt occurred (Figure 6-14).

file Edit Setup Control Window Help Timer 0 Interrupt occurred - Fabric INT	
Timer 1 Interrupt occurred - GPIO 0	
Timer 0 Interrupt occurred - Fabric INI	
Timer 0 Interrupt occurred - Fabric INT	1
Timer 1 Interrupt occurred - GP10 0	
Timer Ø Interrupt occurred - Fabric INI	
Timer 0 Interrupt occurred - Fabric INI	
Timer 1 Interrupt occurred - GPIO Ø	
Timer 0 Interrupt occurred - Fabric INT	
Timer 0 Interrupt occurred - Fabric INT	
Timer 1 Interrupt occurred - CPIO 0	
Timer 0 Interrupt occurred - Fabric INT	
Timer Ø Interrupt occurred - Fabric INI	
Timer 1 Interrupt occurred - GPIO 0	
Timer Ø Interrupt occurred - Fabric INT	
Timer 0 Interrupt occurred - Fabric INT	
Timer 1 Interrupt occurred - GPIO 0	
Switch Interrupt occurred - GPIO 1	
Switch Interrupt occurred - GPIO 1	
Switch Interrupt occurred - GPIO 1	
Timer 0 Interrupt occurred - Fabric INI	



7. Terminate the Debugger (Run > Terminate) on SoftConsole.

Step 4 - Building an Executable Image in Release mode

You can build an application executable image with the Release Configuration and load it into eNVM. The Application code can then be executed from the eNVM after the SmartFusion2 device is programmed.

To load the application image into the eNVM of the SmartFusion2 device and use it for execution, you need to:

- 1. Configure the eNVM for a data storage client.
- 2. Program the eNVM data storage client.

In release mode, you cannot use the SoftConsole debugger to load the executable image into eNVM. Make sure you exit the Debugger.

Steps

 Right-click on both project names in the Project Explorer view and choose Build Configurations > Set Active > Release, as shown in Figure 6-15.



can source menacion manyate si	earch	Project Run Window	Help						
) 🕶 🗟 🛍 🔒 💁 🕶 🍃	11	0 + Ø + \$\$ \$ + \$	*					E 💽	Resource 🌣
Project Explorer 83			a main.c 😫 🚺						-
Interrupt_MSS_GPIO_MSS_CM3_app			82int main 83{	()					^
င်္တေါ်nterrupt_MSS_GPIO_MSS_CM3_hw_pi	6	New Go Into Copy Paste Delete Move Rename	Ctrl+C Ctrl+V Delete	T initialization*/ ART_init(&g_mss_uart1, MSS_UART_57600_BAU MSS_UART_DATA_8_BI splay greeting messag	TS	MSS_UAR	T_NO_PARIT	Y MSS_UART	ONE_STO
Dutline Si	Import Export Build Proje Clean Proje Close Proje Close Unree			ay_greeting();					
		Build Project Clean Project Refresh Close Project Close Unrelated Projects		abling of GPIO 0, GPI S GPIO_0 used for Tim EnableIRQ(GPIO0_IRQn) PIO_config(MSS_GPIO_ PIO_enable_irq(MSS_G		IRQ_EDGE			
stdio.h		1							
- autom		Exclude from build		Set Active		1 Debug		-	÷ 0
mss_uart.h				Del Active	-	√ 2 Release			
mss_uart.h mss_gpio.h		Build Configurations Make Targets		Build	- 1-				
 mss_uart.h mss_gpio.h cortex_nvic.h 		Make Targets Index		Build Delete resource cfgs	•	· Be unione	Kesource	Path	Loc
mss_uart.h mss_gpio.h		Make Targets	•		•	12-10-000	Kesource	Path	Loc
 mss_uarth mss_gpio.h cortex_nvic.h g_greeting_msg : const uint8_t[] GPIO0_IRQHandler(void) : void 		Make Targets Index Convert To		Delete resource cfgs	•	1.8	Kesource	Path	Loc

Figure 6-15 • Configuring Release Mode

- 2. Select **Project > Properties** from the SoftConsole menu. Expand C/C++ Build and select 'Settings'.
- Provide the release mode linker script file to the linker by changing the 'Linker flags' field to "-T../../Interrupt_MSS_GPIO_MSS_CM3_0_hw_platform/CMSIS/startup_gcc/production-executein-place.ld", as shown in Figure 6-16. This step directs SoftConsole to use the production linker script, instead of the default linker script (that allows debugging), to build the application image for the Release Application.



rpe filter text	Settings		🗢 🕶 🗢 👻
Resource Builders C/C++ Build Build Variables	Configuration: Release [Active]		Manage Configurations
Discovery Options Environment Settings Tool Chain Editor C/C++ General Project References Refactoring History Run/Debug Settings	B GNU C Compiler Preprocessor Symbols Directories Optimization Debugging Warnings Miscellaneous General Dibraries Miscellaneous	uld Artifact Binary Parsers Error Parsers Unker flags -T_J_Interrupt_MSS_GPIO_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-execute-in-place.ld Generate linker memory map Other options (-Xlinker [option]) -ocsections	ଇ ଇ ଇ ଲ ଲ ଲ
	 GNU Assembler General Memory map generator GNU Intel Hex File Generator GNU S-Record Generator GNU S-Record Generator GNU S-General GNU Listing Generator General 	Other objects "Stworkspace.loc/Interrupt MSS GPI0 MSS CM3 hw.platform/Release/CMSIS/startup.goc/rearling.m2ixxxx.01" "\$(workspace.loc/Interrupt.MSS_GPI0_MSS_CM3_hw.platform/Release/CMSIS/startup.goc/newlib_stubs.o)"	ଶ ଈ ଲ ହା ହା
		Res	tore Defaults Apply

Figure 6-16 • Linker Script Settings

- 4. Click Apply and then click OK.
- 5. Build the project (**Project > Build All**) and observe that the '.hex' file is generated in the 'Release' folder created in the project folder, as shown in Figure 6-17 and Figure 6-18.

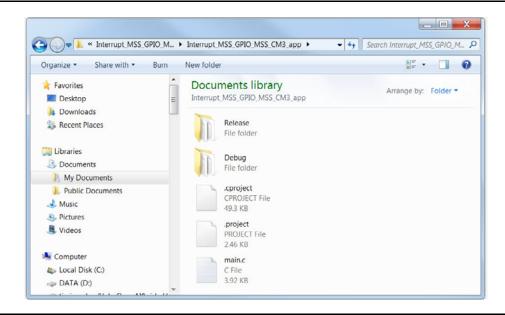


Figure 6-17 • Release Folder



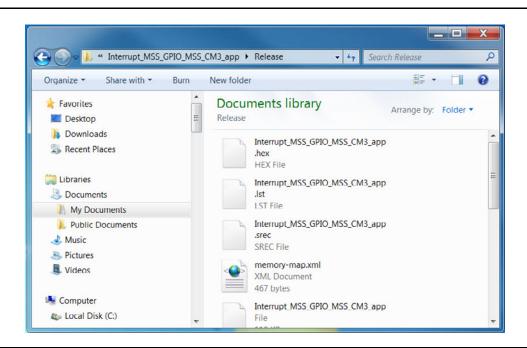


Figure 6-18 • Hex File in Release Folder of SoftConsole Project Location

6. Close SoftConsole.

Loading the Executable Image into eNVM

This section provides the steps required to load the generated executable image into eNVM of the SmartFusion2 cSoC device using the SmartDesign MSS Configurator.

1. From the Libero SoC Design Flow window, right-click **Update eNVM Memory Content** and choose **Open Interactively** (Figure 6-19).



Interru	pt_M	SS_GPIO 🖸 🖸 🖉] 🜮
	Тоо		*
V		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	III
	4	Edit Design Hardware Configuration	
		Programming Connectivity and Interface	
		Programmer Settings	
		Device I/O States During Programming	
	4	Configure Security and Programming Options	
		Security Policy Manager	
		Update eNVM Memory Content	

Figure 6-19 • Update eNVM Memory Content

2. The Update eNVM Memory Content dialog box appears (Figure 6-20).

		er clients in eNVM					
Program	Client Type	Client Name	Depth x Width or umber of Page	Start Address (Hex)	Update Content		
V	Data Storage	My_app	2048 x 32	0	C:\Users\alex.wong\Documents\Work\Libero11.3_Quick_Start_UG\Tim_working\Interrupt_MSS_GPI0\SoftConsole\Interrup		
serialization	clients, the maxim	um devices to pro	ogram:				
serialization	clients, the maxim	um devices to pro	ogram:				

Figure 6-20 • Update eNVM Memory Content Dialog Box

- 3. Click the Program checkbox.
- Click the Browse button (second from right) to navigate to the location of the *.hex file (<SoftConsole Project Folder>/../Interrupt_MSS_GPIO_MSS_CM3_app/Release folder) created by SoftConsole when you build the Application codes. Click the *.hex file and click **Open** (Figure 6-21).



Kelease Kelease	- + Search Release
Organize 🔻 New folder	🏭 🔹 🔲 🔞
	Documents I Arrange by: Folder Release
S Recent Places	Name Interrupt_MSS_GPIO_MSS_CM3_app.hex
Libraries	
Documents	
Jusic State	
S Pictures	
S Videos	
🕼 Computer	

Figure 6-21 • Loading the *hex file into eNVM Memory Content

- 5. Click **OK** to exit the dialog box.
- 6. In the Design Flow window, right-click Bitstream Configuration and choose Configure Options.
- 7. Under Selected Features, deselect Fabric. Make sure eNVM is selected.

Program		
O Updated feature	es only	
Selected feature	es	
Security		
Fabric		
📝 eNVM		
Help	ОК	Cancel

Figure 6-22 • Bitstream Configuration for eNVM

- 8. Click **OK** to exit.
- Make sure the SoftConsole Debugger is terminated (Run > Terminate). Right-click Run Program Action and choose Update and Run.

A green check mark appears next to **Run PROGRAM Action**. The application is programmed into the eNVM of the SmartFusion2 device. It is ready to run off the device's eNVM without launching SoftConsole.



- 10. Disconnect the FlashPro Programmer from the SF2-Eval-Board. Make sure that SoftConsole closes.
- 11. Open the Terminal Emulator. Verify that the Application Code is running off the eNVM flash memory of the device.

Congratulations! You have successfully completed this tutorial.





A – Installing Drivers for the USB-UART

Note: You must have Admin privileges on your PC to install the USB-RS232 drivers.

To install the USB to UART driver:

- 1. Connect a second mini USB cable between the USB connector (J14) on the SmartFusion2 Evaluation Kit Board and a USB port of your computer.
- 2. Download the USB to RS232 bridge USB to UART driver from http://www.microsemi.com/document-portal/doc_download/131593-usb-uart-driver-files
- 3. Click Downloads.
- 4. Extract the downloaded zip file into a folder on your PC.
- 5. Open the Device Manager on your PC.
- 6. Right-click Intel® Active Management Technology SOL (COM3) under Ports (COM & LPT)
- 7. Choose Update Driver Software (Figure A-1).

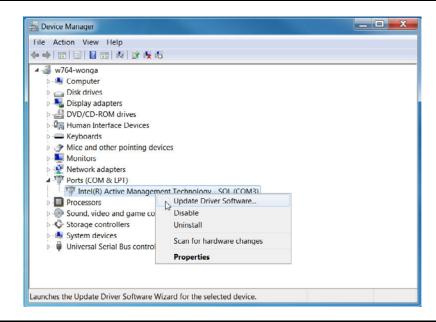


Figure A-1 • Device Manager - Update Driver Software

8. Select Browse my computer for driver software (Figure A-2).



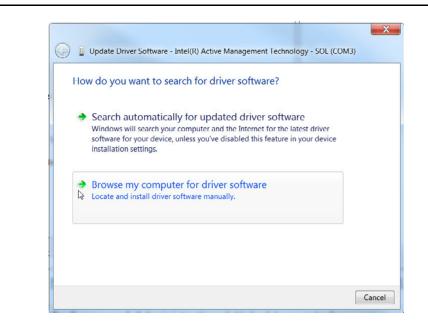


Figure A-2 • Update Driver Software from driver software located locally on PC

9. Browse to the folder location of your driver on your PC (Figure A-3).

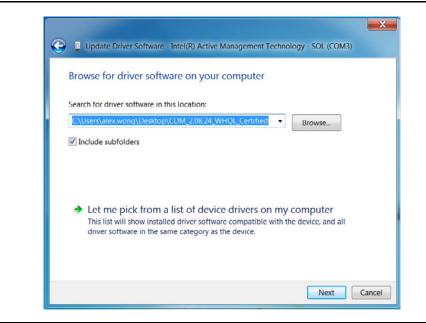


Figure A-3 • Local Drive Software Location

- 10. Follow the prompts to complete the driver installation.
- 11. When the installation is complete, click **OK**. The Ports (COM & LPT) section of the Device Manager (Figure A-4) lists four USB Serial Ports (COMn) numbered consecutively where n can be any number. The USB Serial Port numbers on your PC may be different from Figure A-4.



🛛 🚽 w764-wonga
⊳- 🖳 Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
- Intel(R) Active Management Technology - SOL (COM3)
- USB Serial Port (COM4)
- USB Serial Port (COM5)
- Transformer (COM6)
USB Serial Port (COM7)
Processors
Sound, video and game controllers
Storage controllers
System devices
🔈 🖷 Universal Serial Bus controllers

Figure A-4 • Device Manager USB Serial Ports

•



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, choose **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**.

© 2014 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.