SmartFusion2 SoC FPGA PCIe Control Plane Demo - Libero SoC v11.5

UG0456 User Guide

Superseded

Microsemi

Table of Contents

Superseded

SmartFusion2 SoC FPGA - PCIe Control Plane Demo

Superseded

Introduction

SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) devices integrate a fourth generation flash-based FPGA fabric and an ARM® Cortex®-M3 processor, along with high performance communication interfaces on a single chip. The SmartFusion2 high speed serial interface (SERDESIF) provides a fully hardened PCIe endpoint (EP) implementation and is compliant with PCIe Base Specification Revision 2.0 and 1.1. For more details, refer to the *SmartFusion2 SoC FPGA High Speed Serial Interfaces User's Guide*.

The demo explains the SmartFusion2 embedded PCI Express feature and how this can be used as a low bandwidth control plane interface using the SmartFusion2 Evaluation Kit. The demo provides a simple design to access the SmartFusion2 PCIe EP from a Host PC. A GUI is provided for read and write access to the SmartFusion2 PCIe configuration space and memory space of BAR0 and BAR1. The demo also provides Host PC device drivers for the SmartFusion2 PCIe EP. This demo can run on both windows and Red Hat Linux operating system.

Figure 1 shows the top-level block diagram for the PCIe control plane demo. The demo design uses a SmartFusion2 PCIe interface with a maximum link width of x4 to interface with a Host PC PCIe Gen2 slot. The SmartFusion2 microcontroller subsystem (MSS) GPIOs control the LEDs and switches on the SmartFusion2 Evaluation Kit through the PCIe interface. The Host PC can also read memory and writes to the SmartFusion2 eSRAM through the GUI. The Host PC can also be interrupted by using the push button on the SmartFusion2 Evaluation Kit.

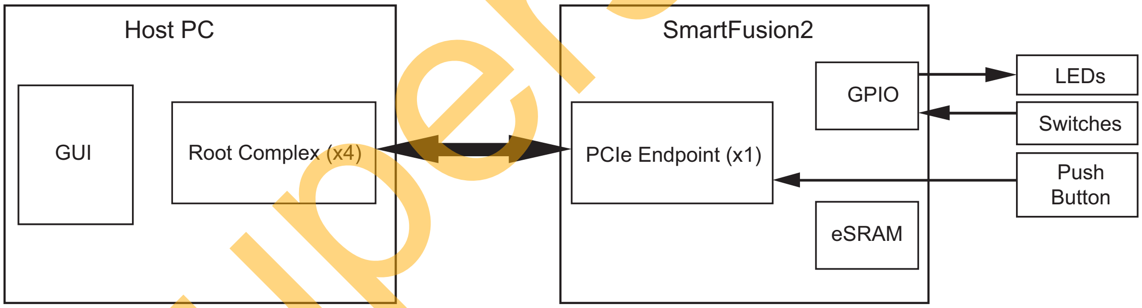


***Figure 1* • PCIe Control Plane Demo Top-Level Block Diagram**

The demo design performs the following tasks:

- Displays the PCIe link enable/disable, negotiated link width, and the link speed.
- Controls the status of LEDs on the SmartFusion2 Evaluation Kit according to the command from the GUI.
- Displays the position of DIP Switches on SmartFusion2 Evaluation Kit.
- Enables read and write to eSRAM.
- Interrupts the Host PC, when the push button is pressed. The GUI displays the count value of the number of interrupts sent from the SmartFusion2 Evaluation Kit.
- Displays the SmartFusion2 PCIe Configuration Space.

Demo Requirements

Hardware and Software Requirements

Table 1 shows the hardware and software required to run the demo.

Table 1 • **Required Hardware and Software to Run the Demo**

Hardware	**Version**
SmartFusion2 Security Evaluation Kit	Rev C or later
12 V adapter (provided along with the kit)	-
FlashPro4 programmer (provided along with the kit)	-
Host PC with an available PCIe 2.0 Gen1 or Gen2 compliant slot	Operating system: Windows XP SP2: 64-bit Windows 7: 64-bit or Red Hat Linux Kernel Version: 2.6.18-308
Software	
Libero® System-on-Chip (SoC)	v11.5
SoftConsole	v3.4SP1
Host PC Drivers (provided along with the design files)	-
GUI executable (provided along with the design files)	-

Design Files

The design files for this demo can be downloaded from the Microsemi website: http://soc.microsemi.com/download/rsc/?f=m2s_ug0456_pcie_control_plane_demo_liberov11p5_df

Design files include:

- Libero project
- Linux_64bit
- ProgrammingFile
- Windows_64bit
- Source files
- Readme file

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

Superseded

Superseded

Demo Design Description

This demo design implements the SmartFusion2 embedded PCI Express interface as a low bandwidth control plane interface. This design provides Host PC drivers and a Host PC interface over PCIe to control the SmartFusion2 device. Figure 2 shows a detailed block diagram of the design implementation. The PCIe EP device receives commands from the Host PC through the GUI and does corresponding memory writes to the SmartFusion2 MSS address space. The MSS address space provides a GPIO block and eSRAM memory block which is accessed through a Fabric Interface Controller (FIC_0).

The SERDES_IF2_0 is configured for a PCIe 2.0, x1 link width with GEN2 speed. The PCIe interface to the fabric uses an AMBA High-speed Bus (AHB). The AHB master interface of SERDESIF is enabled and connected to the AHB slave interface of FIC_0 to access the MSS peripherals. The SmartFusion2 PCIe BAR0 and BAR1 are configured in 32-bit memory mapped memory mode.

The AXI master windows of the SERDESIF PCIe provide address translation for accessing one address space from another address space as the PCIe address is different from SmartFusion2 AHB bus matrix address space. The AXI master window 0 is enabled and configured to translate the BAR0 memory address space to the MSS GPIO address space to control the MSS GPIOs. The AXI master window 1 is enabled and configured to translate the BAR1 memory address space to the eSRAM address space to perform read and writes from PCIe.

MSS GPIO block is enabled and configured as below:

- GPIO_0 to GPIO_7 as outputs and connected to LEDs
- GPIO_8 to GPIO_11 as inputs and connected to DIP switches

The PCIe interrupt line is connected to the SW4 push button on the SmartFusion2 Evaluation Kit. The FPGA clocks are configured to run the FPGA fabric and MSS at 100 MHz.

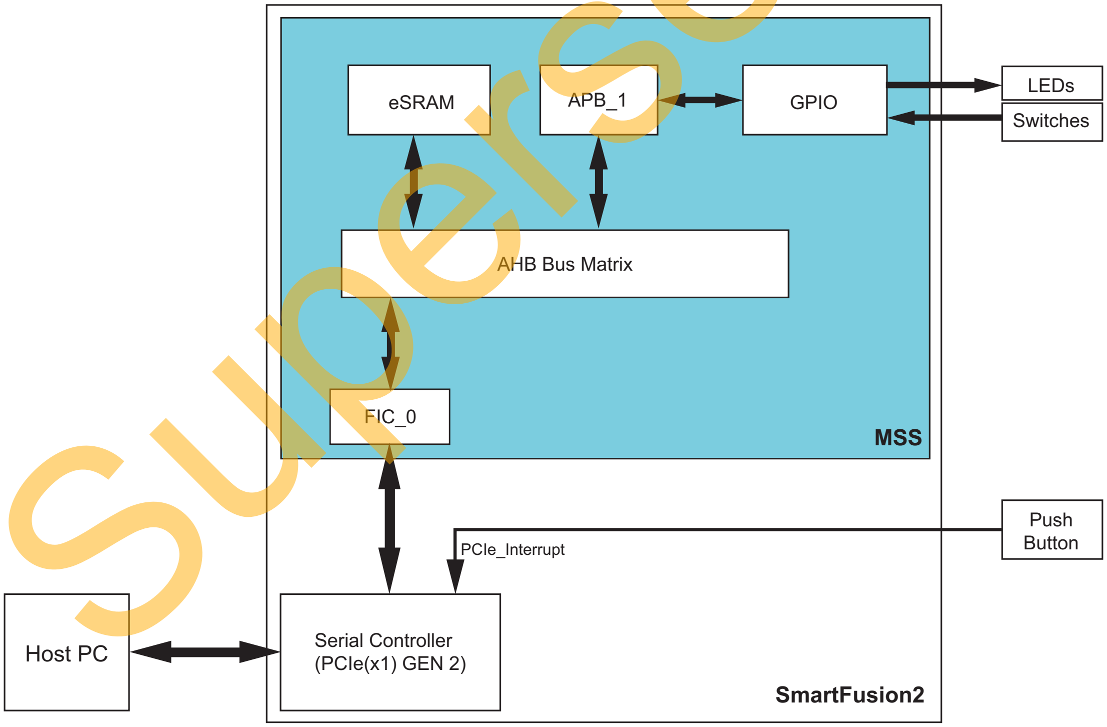


***Figure 2* • PCIe Control Plane Demo Block Diagram**

Superseded

Building the Demo

This demo design provides a complete design flow starting from a new project to a working design on the SmartFusion2 Evaluation Kit. This process includes usage of the tools in the Libero SoC design suite to program the SmartFusion2 device.

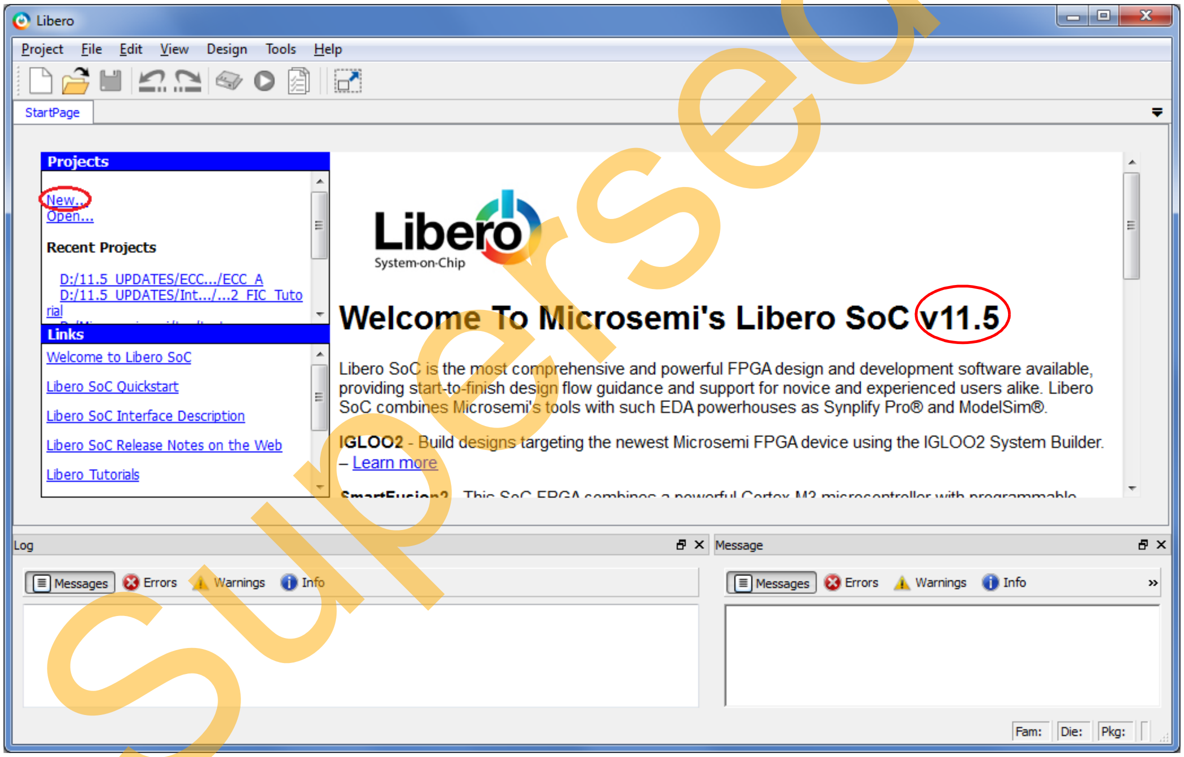
Building the demo involves the following steps:

- Step 1: Creating a Libero SoC Project
- Step 2: Creating an eNVM Client
- Step 3: Developing the Simulation Stimulus
- Step 4: Simulating the Design
- Step 5: Generating the Program File

Step 1: Creating a Libero SoC Project

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

1. Click **Start > Programs > Microsemi Libero SoC v11.5 > Libero SoC v11.5**, or click the shortcut on your desktop. The Libero SoC v11.5 Project Manager is displayed as shown in Figure 3.


Libero
Project File Edit View Design Tools Help
StartPage
Projects
New...
Open...
Recent Projects
D:/11.5_UPDATES/ECC.../ECC_A
D:/11.5_UPDATES/Int.../...2_FIC_Tutorial
Links
Welcome to Libero SoC
Libero SoC Quickstart
Libero SoC Interface Description
Libero SoC Release Notes on the Web
Libero Tutorials
Libero
System-on-Chip
Log
Messages Errors Warnings Info
Message
Messages Errors Warnings Info
Fam: Die: Pkg:

Welcome To Microsemi's Libero SoC v11.5

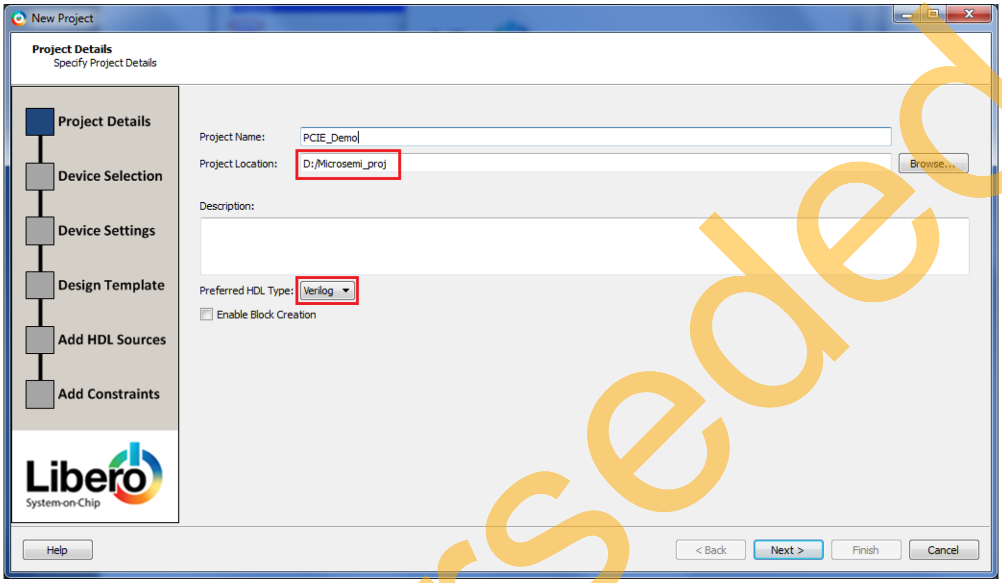
Libero SoC is the most comprehensive and powerful FPGA design and development software available, providing start-to-finish design flow guidance and support for novice and experienced users alike. Libero SoC combines Microsemi's tools with such EDA powerhouses as Synplify Pro® and ModelSim®.

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

SmartFusion2 - This SoC FPGA combines a powerful Cortex M3 microcontroller with programmable

***Figure 3* • Libero SoC v 11.5 Project Manager**

2. Create a new project using one of the following options:
   - Select **New** on the Start Page tab as highlighted in Figure 3 on page 6.
   - Click **Project > New Project** from the Libero SoC menu.
3. Enter the following information in the **New Project-Project Details** tab as shown in Figure 4.
   - Project Name: PCIE_Demo
   - Project Location: Select an appropriate location (for example, D:/Microsemi_prj)
   - Preferred HDL type: Verilog or VHDL



Project Name: PCIE_Demo

Project Location: D:/Microsemi_proj

Description:

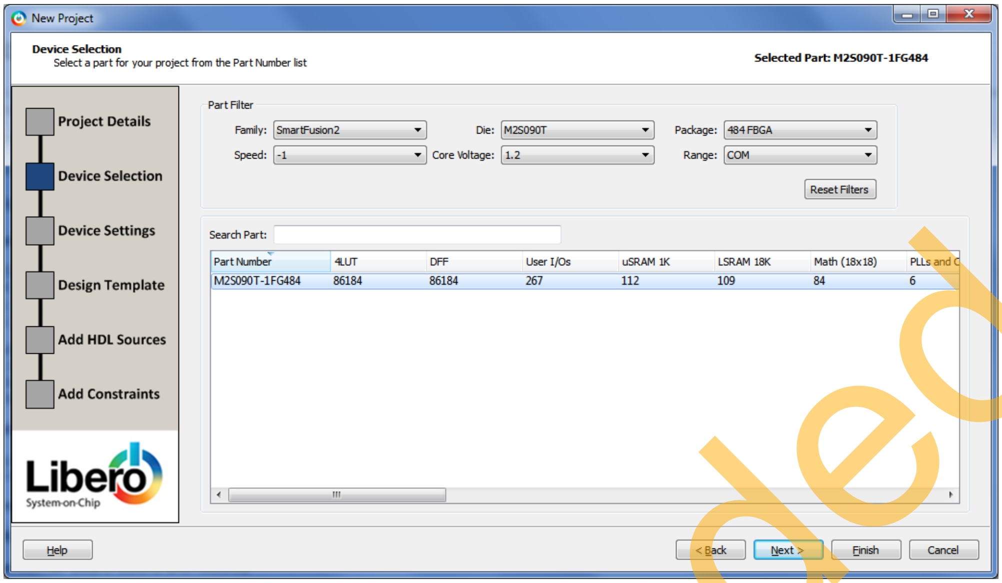
Preferred HDL Type: Verilog

☐ Enable Block Creation

Figure 4 • **Project Details Tab**

4. Select the information for **Device Selection** as shown in Figure 5 on page 8 and click **Next**.
   - Family: SmartFusion2
   - Die: M2S090T
   - Package: 484 FBGA
   - Speed: -1
   - Core Voltage: 1.2
   - Operating conditions: COM

Superseded



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

Selected Part: M2S090T-1FG484

Part Filter

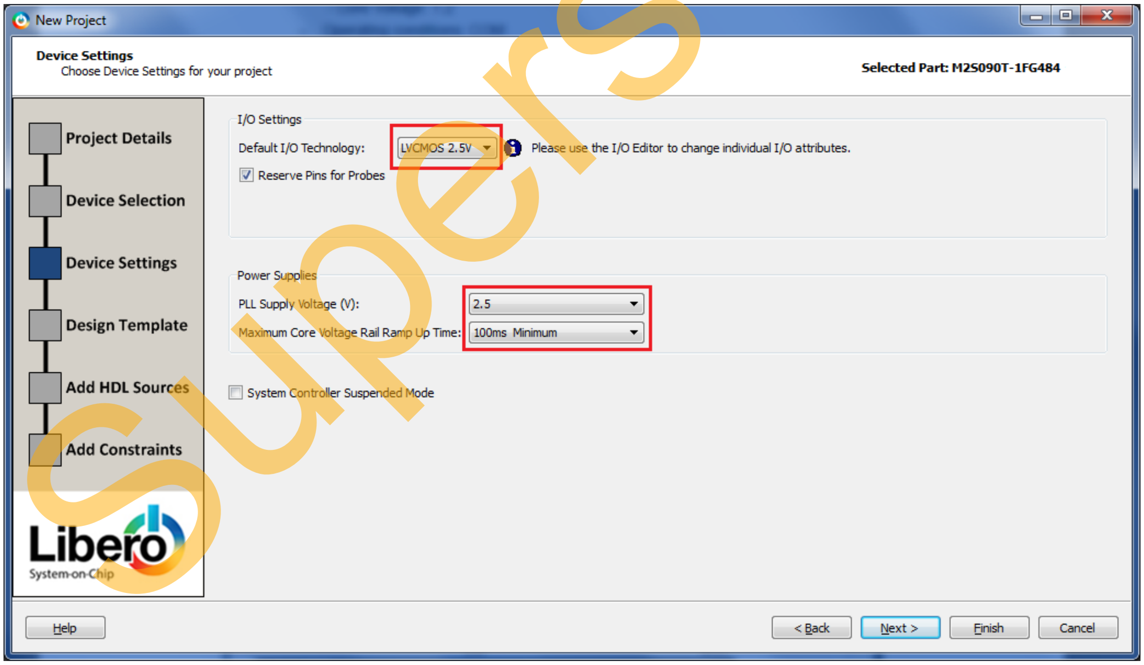
Family: SmartFusion2
Die: M2S090T
Package: 484 FBGA
Speed: -1
Core Voltage: 1.2
Range: COM

Search Part:

Part Number	4LUT	DFF	User I/Os	uSRAM 1K	LSRAM 18K	Math (18x18)	PLLs and C
M2S090T-1FG484	86184	86184	267	112	109	84	6

***Figure 5* • Device Selection Tab**

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



Device Settings
Choose Device Settings for your project

Selected Part: M2S090T-1FG484

I/O Settings

Default I/O Technology: LVCMOS 2.5V

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

☑ Reserve Pins for Probes

Power Supplies

PLL Supply Voltage (V): 2.5

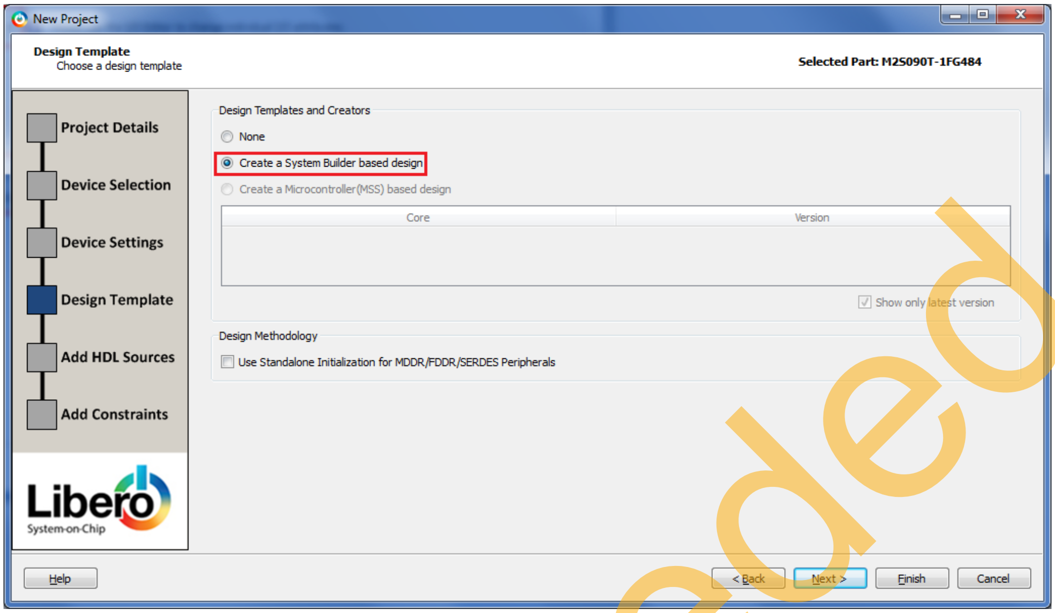
Maximum Core Voltage Rail Ramp Up Time: 100ms Minimum

□ System Controller Suspended Mode

***Figure 6* • Device Settings Tab**

Superseded

6. **Design Template** tab is displayed as shown in Figure 7. Select **Create a System Builder based design** under **Design Templates and Creators** and click **Next**.



Design Template
Choose a design template

Selected Part: M2S090T-1FG484

Design Templates and Creators

○ None
● Create a System Builder based design
○ Create a Microcontroller(MSS) based design

Core	Version

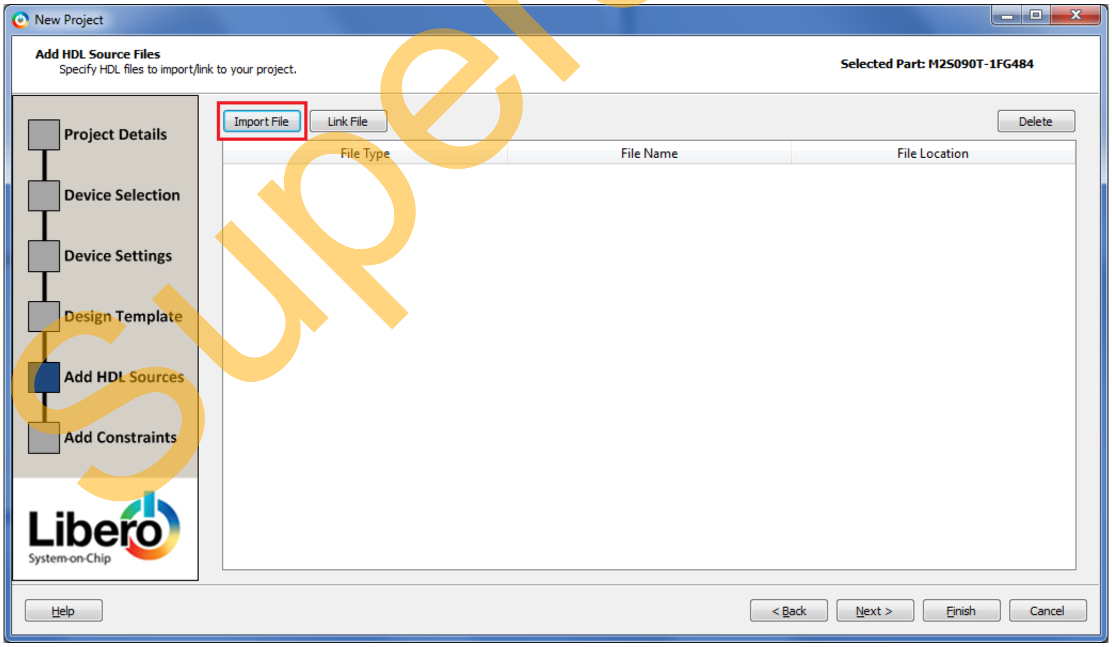
☑ Show only latest version

Design Methodology

☐ Use Standalone Initialization for MDDR/FDDR/SERDES Peripherals

Figure 7 **• Design Template Tab**

7. **Add HDL Sources** tab is displayed as shown in Figure 8. **Verilog/VHDL Source Files** can be added here.



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

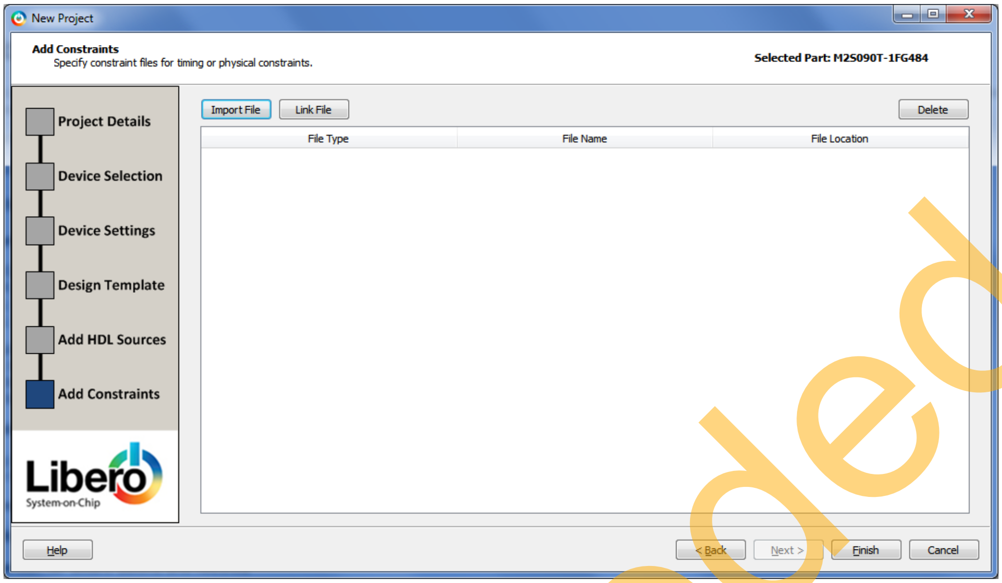
Selected Part: M2S090T-1FG484

File Type	File Name	File Location

Figure 8 **• Add HDL Source Files Tab**

Superseded

8. **Add Constraints** tab is displayed as shown in Figure 9. **Constraints** file can be added using Import option.



New Project

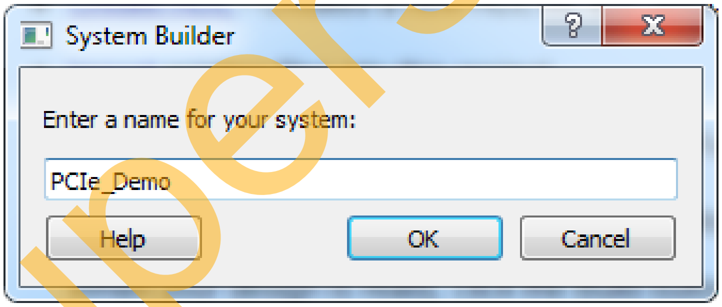
Add Constraints
Specify constraint files for timing or physical constraints.

Selected Part: M2S090T-1FG484

File Type	File Name	File Location

Figure 9 **• Add Constraints Tab**

9. Click **Finish**. This displays the **System Builder** dialog box.
10. Enter a name for your system in the column provided, as shown in Figure 10.



System Builder

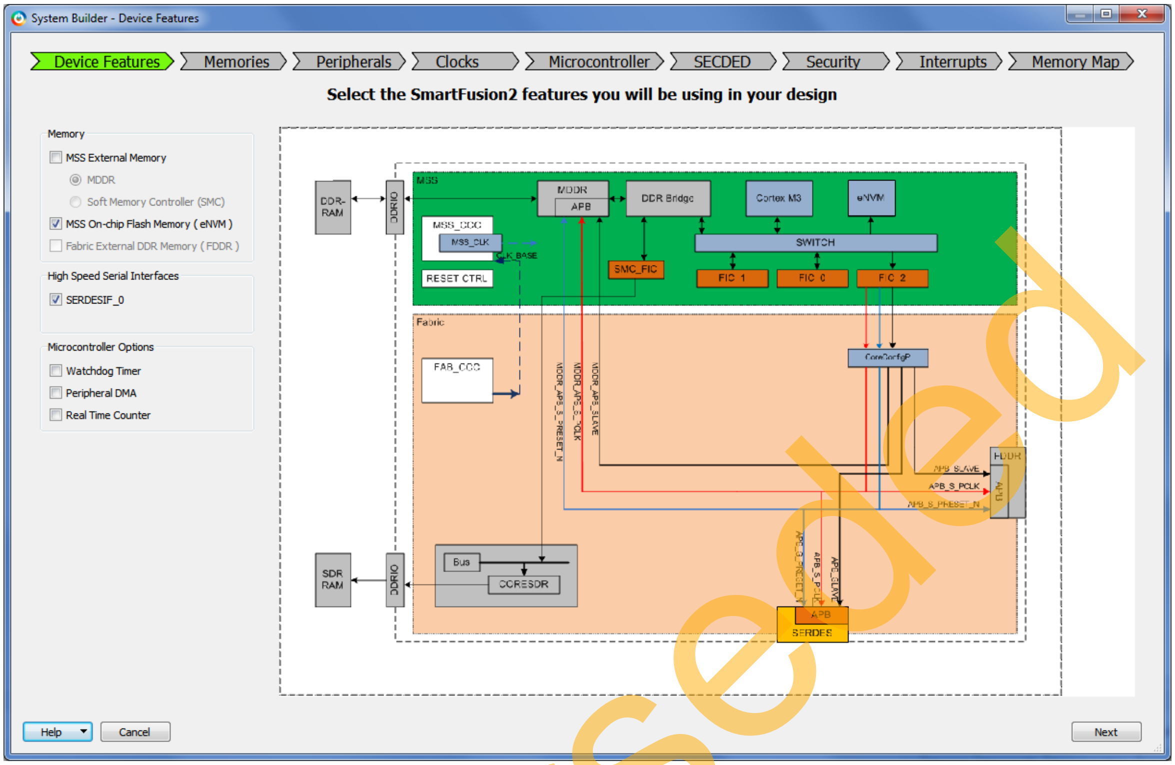
Enter a name for your system:

PCIe_Demo

Figure 10 **• System Builder Dialog Box**

11. Enter **PCIe_Demo** as the name of the system and click **OK**. The System Builder dialog box is displayed with the Device Features page open by default.
12. Enter the following information in the **System Builder – Device Features page** as shown in Figure 11 on page 11:
    – Memory: Clear all except MSS On-chip Flash Memory (eNVM)
    – High-speed serial interfaces: Check SERDESIF_0
    – Microcontroller Options: Clear All

Superseded



Select the SmartFusion2 features you will be using in your design

Memory

- ☐ MSS External Memory
  - ● MDDR
  - ○ Soft Memory Controller (SMC)
- ☑ MSS On-chip Flash Memory (eNVM)
- ☐ Fabric External DDR Memory (FDDR)

High Speed Serial Interfaces

- ☑ SERDESIF_0

Microcontroller Options

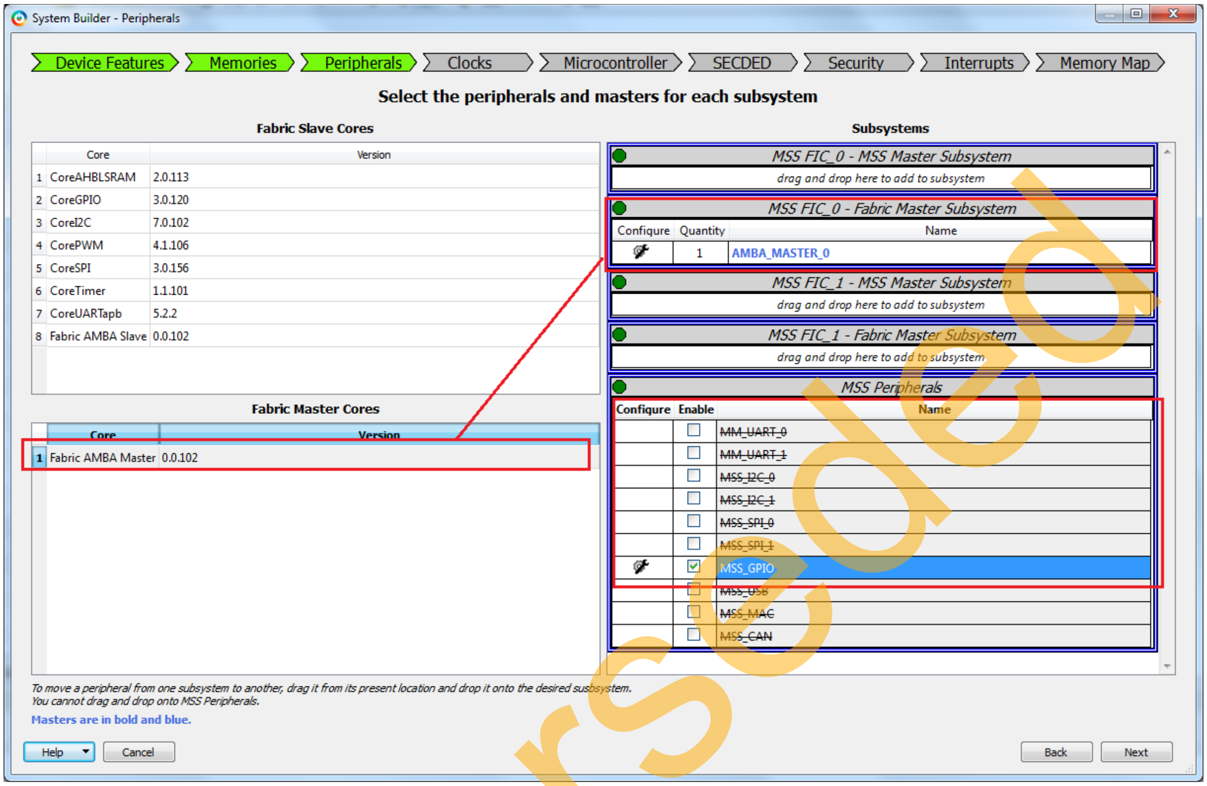
- ☐ Watchdog Timer
- ☐ Peripheral DMA
- ☐ Real Time Counter

Figure 11 **• SmartFusion2 System Builder Configurator**

13. Click **Next**. The **System Builder – Memories** page is displayed.

Superseded

14. Click **Next**. The **System Builder – Peripherals** page is displayed. Drag the **Fabric AMBA Master** to **MSS_FIC_0 – Fabric Master Subsystem** as shown in Figure 12. It enables the MSS FIC_0 slave interface.



System Builder - Peripherals

Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - MSS Master Subsystem
drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_0

MSS FIC_1 - MSS Master Subsystem
drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem
drag and drop here to add to subsystem

MSS Peripherals

Configure	Enable	Name
	☐	~~MM_UART_0~~
	☐	~~MM_UART_1~~
	☐	~~MSS_I2C_0~~
	☐	~~MSS_I2C_1~~
	☐	~~MSS_SPI_0~~
	☐	~~MSS_SPI_1~~
	☑	MSS_GPIO
	☐	~~MSS_USB~~
	☐	~~MSS_MAC~~
	☐	~~MSS_CAN~~

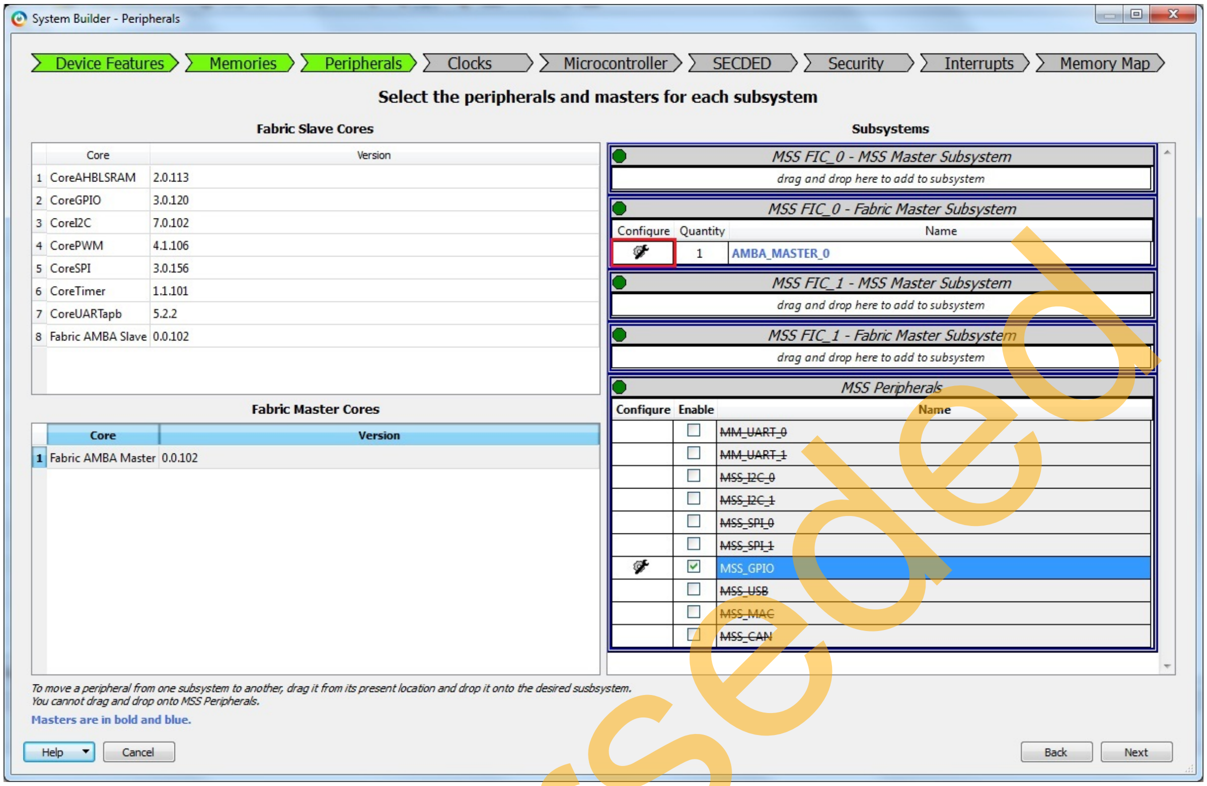
To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired subsystem.
You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

***Figure 12* • System Builder – Peripherals Page**

15. Disable the MSS Peripherals except MSS_GPIO. The **System Builder – Peripherals** page is displayed as shown in Figure 13 on page 13. Configure **MSS_FIC_0 – Fabric Master Subsystem** for AHB-Lite by clicking on the AMBA_MASTER_0 configurator button highlighted in Figure 13 on page 13. This displays a drop-down list as shown in Figure 14 on page 13.

Superseded



Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_0

MSS FIC_1 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem

drag and drop here to add to subsystem

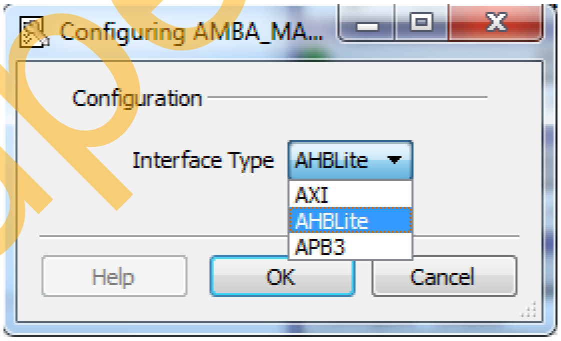
MSS Peripherals

Configure	Enable	Name
	□	~~MM_UART_0~~
	□	~~MM_UART_1~~
	□	~~MSS_I2C_0~~
	□	~~MSS_I2C_1~~
	□	~~MSS_SPI_0~~
	□	~~MSS_SPI_1~~
	☑	MSS_GPIO
	□	~~MSS_USB~~
	□	~~MSS_MAC~~
	□	~~MSS_CAN~~

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem.
You cannot drag and drop onto MSS Peripherals.
Masters are in bold and blue.

Figure 13 **• System Builder – Peripherals Page**

16. Select **AHBLite** from the drop-down list as shown in Figure 14.



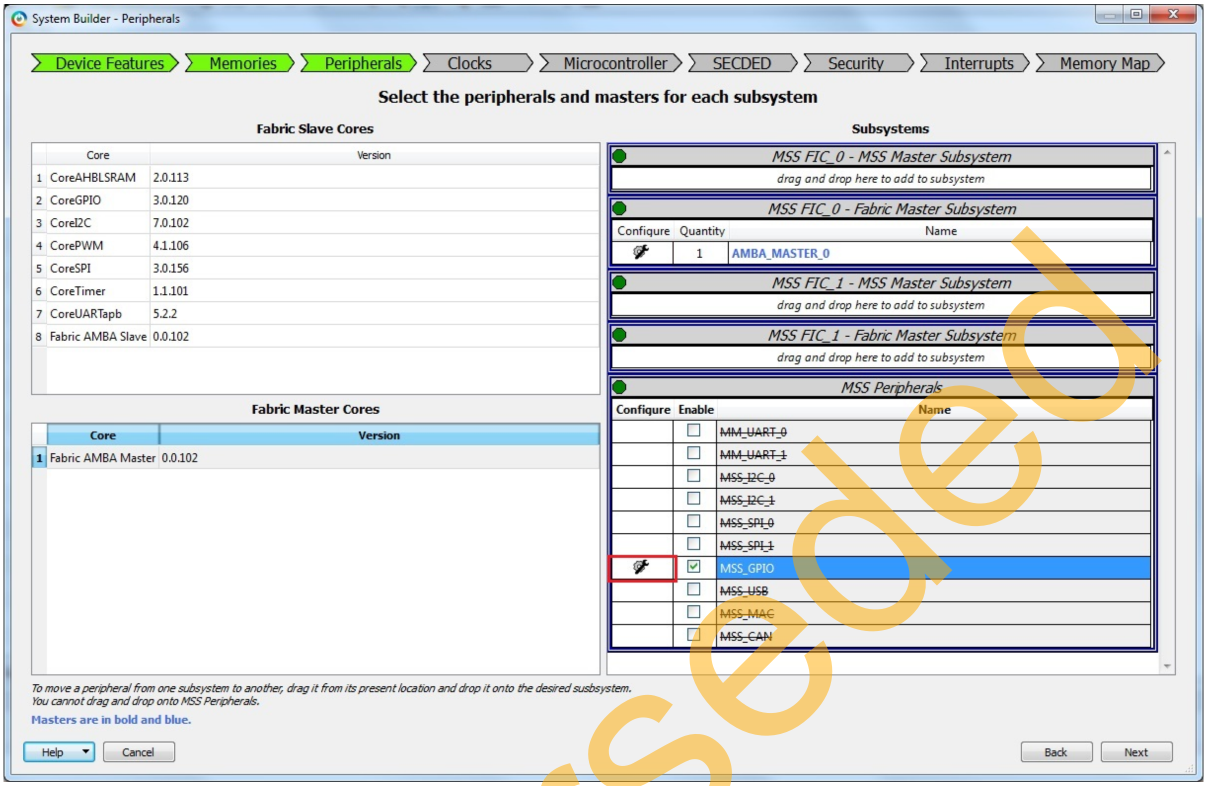
Configuration

Interface Type: AHBLite

Figure 14 **• Configuring AMBA Master**

Superseded

17. Configure MSS_GPIO by clicking **MSS_GPIO Configure** as shown in Figure 15.



System Builder - Peripherals

Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_0

MSS FIC_1 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem

drag and drop here to add to subsystem

MSS Peripherals

Configure	Enable	Name
	□	~~MM_UART_0~~
	□	~~MM_UART_1~~
	□	~~MSS_I2C_0~~
	□	~~MSS_I2C_1~~
	□	~~MSS_SPI_0~~
	□	~~MSS_SPI_1~~
	☑	MSS_GPIO
	□	~~MSS_USB~~
	□	~~MSS_MAC~~
	□	~~MSS_CAN~~

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem.
You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

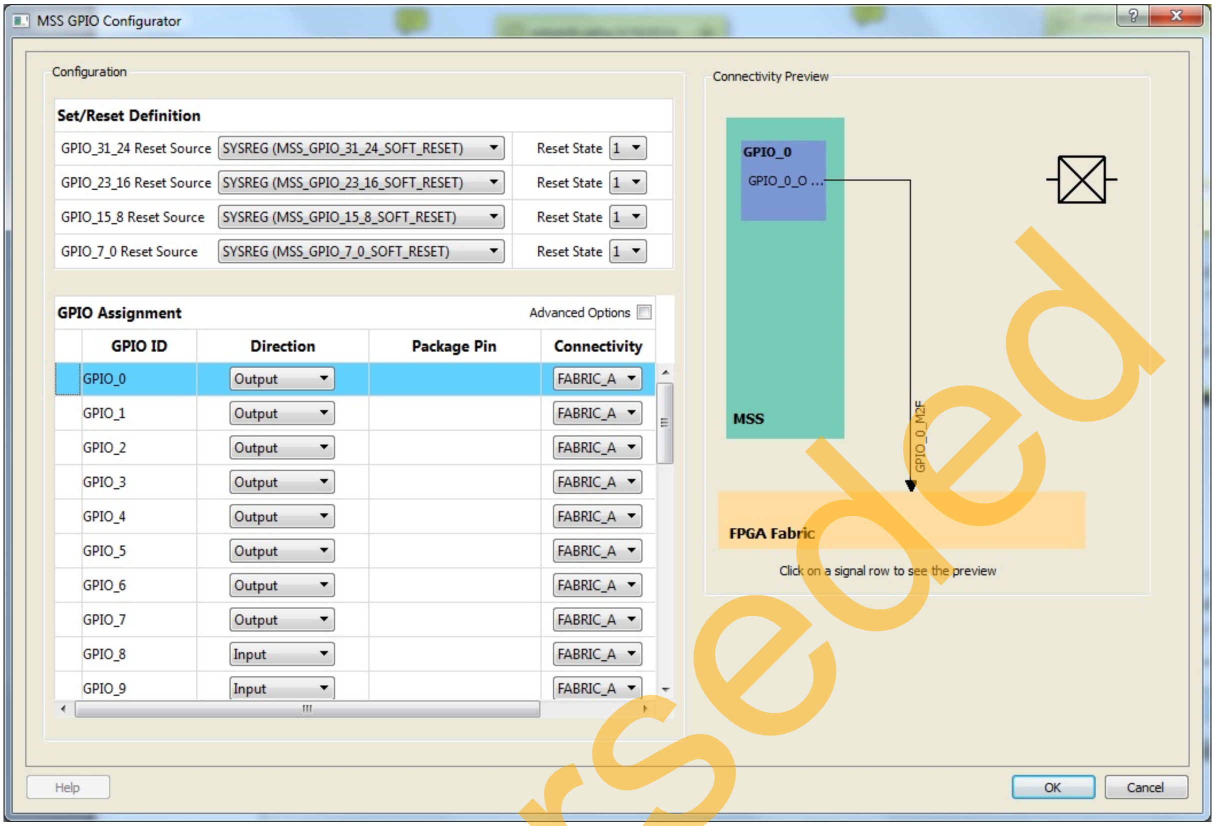
***Figure 15* • System Builder – Peripherals Page**

18. Double-click **MSS_GPIO** configuration button as shown in Figure 15 and configure:
    – GPIO_0 to GPIO_7 as outputs and their connectivity to FABRIC_A to connect with LEDs
    – GPIO_8 to GPIO_11 as inputs and their connectivity to FABRIC_A, to connect with DIP switches

    This design requires configuring GPIO_0 to GPIO_7 to drive LED_1 to LED_8 on the SmartFusion2 Evaluation Kit, and GPIO_8 to GPIO_11 to connect DIP1 to DIP4. These signals are routed through the fabric to the I/O pins.

Superseded

Figure 16 shows the MSS GPIO Configurator.



MSS GPIO Configurator

Configuration

Set/Reset Definition

GPIO_31_24 Reset Source	SYSREG (MSS_GPIO_31_24_SOFT_RESET)	Reset State	1
GPIO_23_16 Reset Source	SYSREG (MSS_GPIO_23_16_SOFT_RESET)	Reset State	1
GPIO_15_8 Reset Source	SYSREG (MSS_GPIO_15_8_SOFT_RESET)	Reset State	1
GPIO_7_0 Reset Source	SYSREG (MSS_GPIO_7_0_SOFT_RESET)	Reset State	1

GPIO Assignment Advanced Options □

GPIO ID	Direction	Package Pin	Connectivity
GPIO_0	Output		FABRIC_A
GPIO_1	Output		FABRIC_A
GPIO_2	Output		FABRIC_A
GPIO_3	Output		FABRIC_A
GPIO_4	Output		FABRIC_A
GPIO_5	Output		FABRIC_A
GPIO_6	Output		FABRIC_A
GPIO_7	Output		FABRIC_A
GPIO_8	Input		FABRIC_A
GPIO_9	Input		FABRIC_A

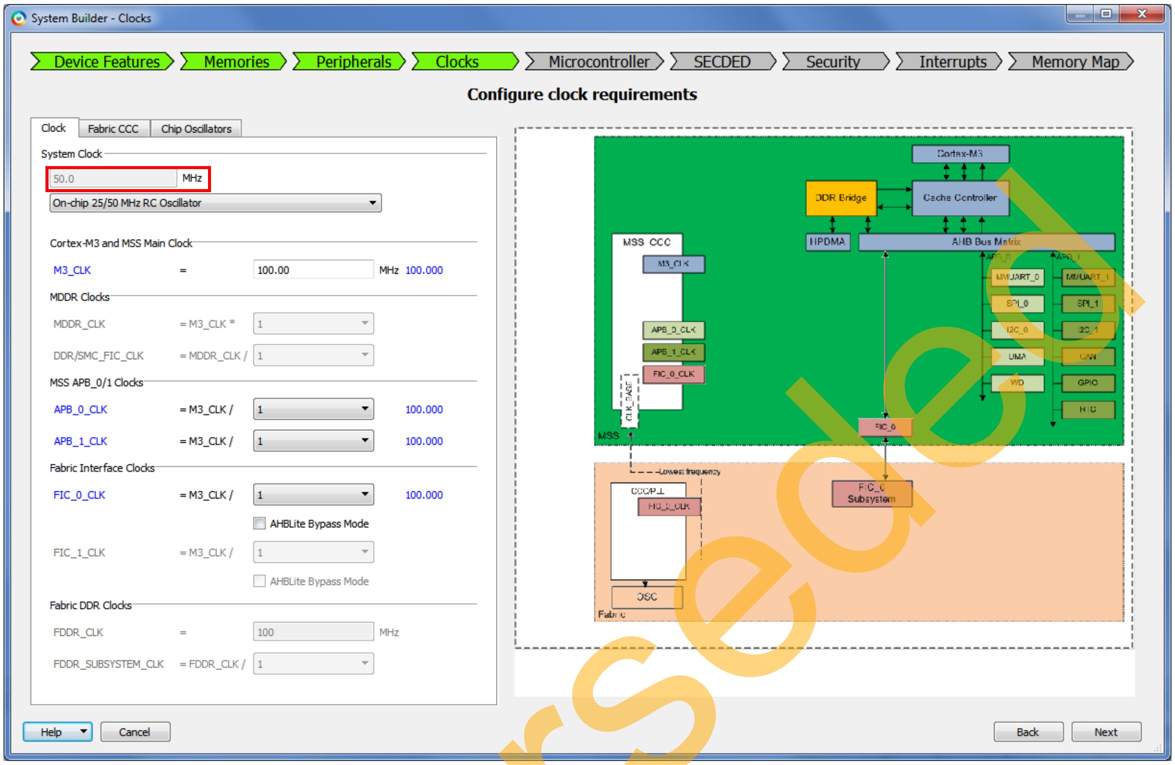
Connectivity Preview

Figure 16 **• GPIO Configuration**

19. Click **OK** on MSS GPIO Configurator.

Superseded

20. Click **Next**. The **System Builder – Clock** page is displayed, as shown in Figure 17. Change the configuration of **System Clock** from 100 MHz to 50 MHz. The dedicated input pad is connected to on board 50 MHz oscillator. The M3_CLK is configured to 100 MHz by default.



System Builder - Clocks

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure clock requirements

Clock | Fabric CCC | Chip Oscillators

System Clock

50.0 MHz

On-chip 25/50 MHz RC Oscillator

Cortex-M3 and MSS Main Clock

M3_CLK = 100.00 MHz 100.000

MDDR Clocks

MDDR_CLK = M3_CLK * 1

DDR/SMC_FIC_CLK = MDDR_CLK / 1

MSS APB_0/1 Clocks

APB_0_CLK = M3_CLK / 1 100.000

APB_1_CLK = M3_CLK / 1 100.000

Fabric Interface Clocks

FIC_0_CLK = M3_CLK / 1 100.000

☐ AHBLite Bypass Mode

FIC_1_CLK = M3_CLK / 1

☐ AHBLite Bypass Mode

Fabric DDR Clocks

FDDR_CLK = 100 MHz

FDDR_SUBSYSTEM_CLK = FDDR_CLK / 1

Figure 17 **• System Builder – Clock Page**

21. Click **Next**. The **System Builder - Microcontroller** page is displayed. Leave all the default selections.

22. Click **Next**. The **System Builder - SECDED** page is displayed. Do not change the default selections.

23. Click **Next**. The **System Builder** - **Security** page is displayed. Do not change the default selections.

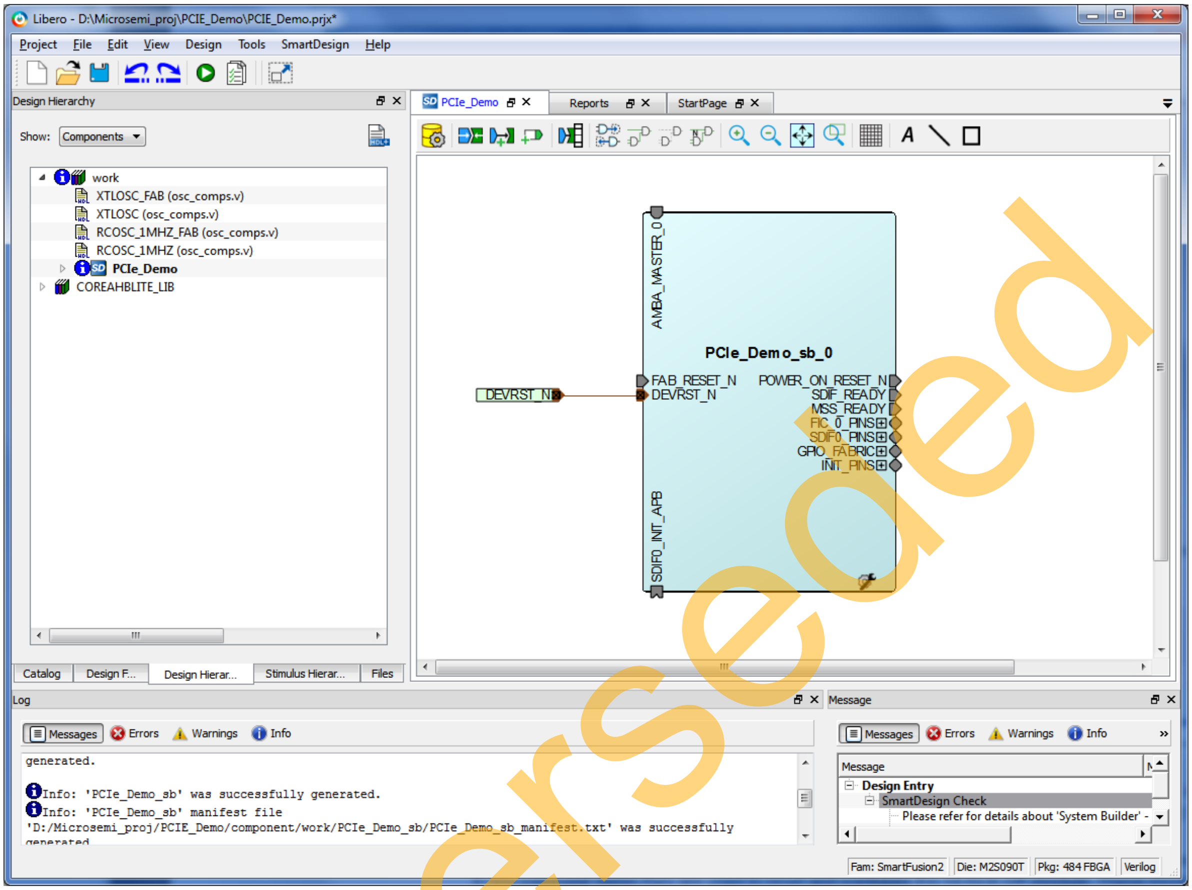
24. Click **Next**. The **System Builder - Interrupts** page is displayed. Do not change the default selections.

25. Click **Next**. The **System Builder - Memory Map** page is displayed. Do not change the default selections.

26. Click **Finish**. The **System Builder** generates the system based on the selected options.

Superseded

The System Builder block is created and added to Libero SoC project automatically, as shown in Figure 18.



```
generated.

Info: 'PCIe_Demo_sb' was successfully generated.
Info: 'PCIe_Demo_sb' manifest file
'D:/Microsemi_proj/PCIE_Demo/component/work/PCIe_Demo_sb/PCIe_Demo_sb_manifest.txt' was successfully
generated.
```

```
Message
Design Entry
  SmartDesign Check
    Please refer for details about 'System Builder' -
```

Figure 18 **• SmartFusion2 System Builder Generated System**

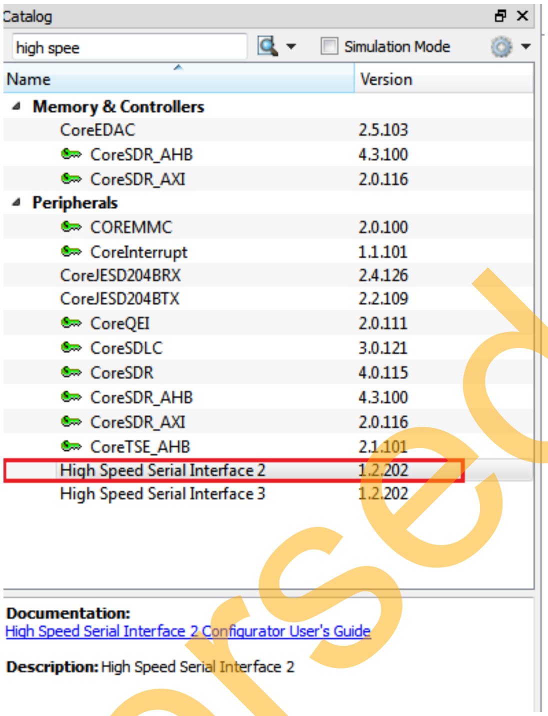
The two soft cores (CoreResetP and CoreConfigP) are automatically instantiated and connected by the System Builder. The block connections can be seen by opening the System Builder component in the SmartDesign canvas.

Note: CoreResetP and CoreConfigP are responsible for the reset and configuration of ASIC peripherals. In this particular demo they are used to reset and configure the SERDESIF module. These modules are included in the System Builder generated component when an ASIC peripheral is selected.

Superseded

Instantiating SERDESIF Component in PCIe_Demo SmartDesign

The Libero SoC Catalog provides IP cores that can be easily dragged-and-dropped into the SmartDesign Canvas workspace. Many of these IPs are free to use while several require a license agreement. The SERDESIF module that supports the PCIe embedded interface is included in the catalog. To instantiate the SERDESIF component in the **PCIe_Demo** SmartDesign, expand the **Peripherals** category in the Libero SoC Catalog.



Name	Version
Memory & Controllers	
CoreEDAC	2.5.103
CoreSDR_AHB	4.3.100
CoreSDR_AXI	2.0.116
Peripherals	
COREMMC	2.0.100
CoreInterrupt	1.1.101
CoreJESD204BRX	2.4.126
CoreJESD204BTX	2.2.109
CoreQEI	2.0.111
CoreSDLC	3.0.121
CoreSDR	4.0.115
CoreSDR_AHB	4.3.100
CoreSDR_AXI	2.0.116
CoreTSE_AHB	2.1.101
High Speed Serial Interface 2	1.2.202
High Speed Serial Interface 3	1.2.202

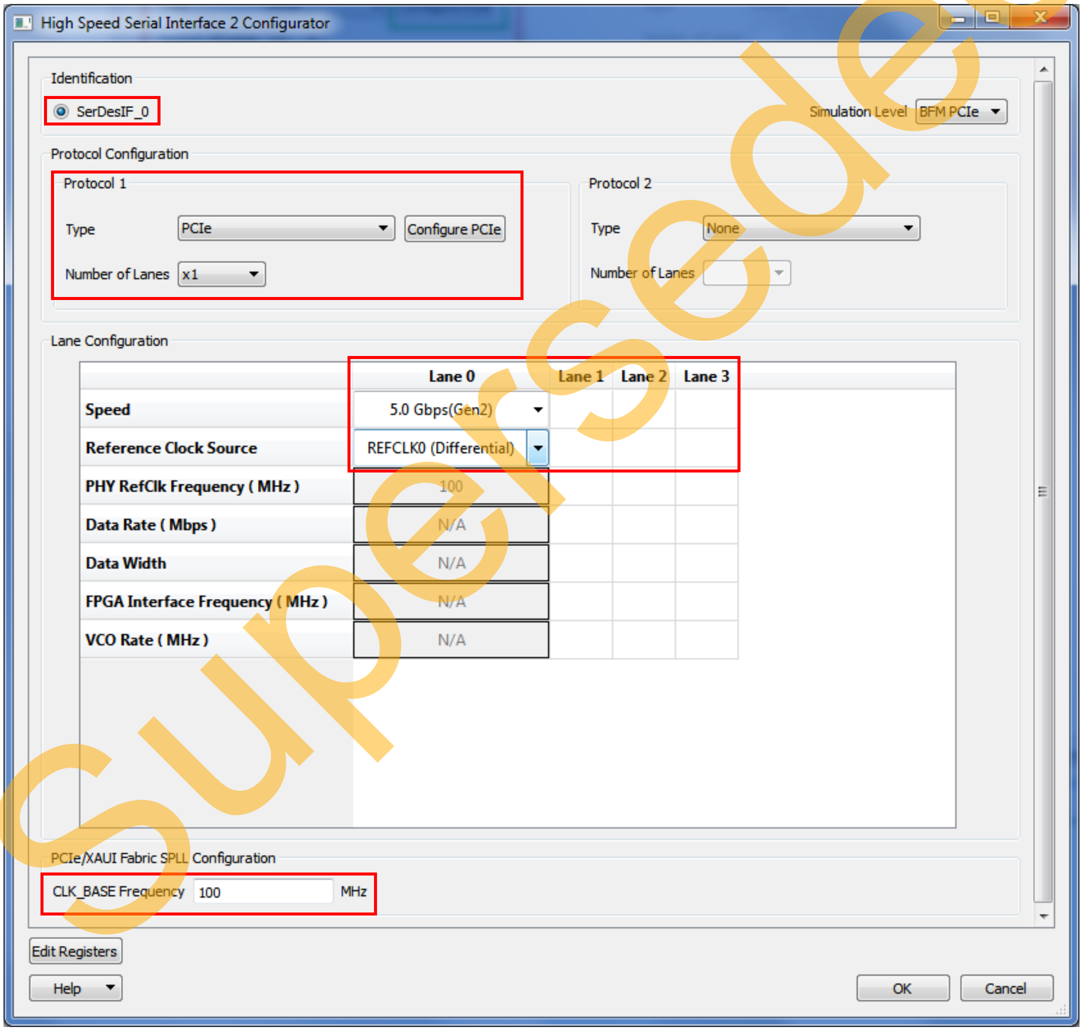
Documentation:
High Speed Serial Interface 2 Configurator User's Guide

Description: High Speed Serial Interface 2

***Figure 19* • IP Catalog**

Superseded

1. Drag the **High Speed Serial Interface2** onto the **PCIe_Demo SmartDesign** canvas. If the component appears shadowed in the Vault, right-click the name and select **Download**.
2. Double-click the **SERDES_IF2_0** component in the SmartDesign canvas to open the **SERDES** configurator. Configure the SERDES with the following settings as shown in Figure 20:
   – Select SERDESIF_0
   – Simulation Level: BFM PCIe
   – Protocol1: Number of Lanes: x1
   – Protocol1: Type: PCIe
   – CLK_BASE Frequency (MHz): 100
   – Lane Configuration: Speed: 5.0 Gbps(Gen2)
   – Lane Configuration:
   – Reference Clock Source: REFCLK0 (Differential)



High Speed Serial Interface 2 Configurator

Identification

● SerDesIF_0

Simulation Level: BFM PCIe

Protocol Configuration

Protocol 1

Type: PCIe

Number of Lanes: x1

Protocol 2

Type: None

Number of Lanes:

Lane Configuration

	Lane 0	Lane 1	Lane 2	Lane 3
Speed	5.0 Gbps(Gen2)			
Reference Clock Source	REFCLK0 (Differential)			
PHY RefClk Frequency (MHz)	100			
Data Rate (Mbps)	N/A			
Data Width	N/A			
FPGA Interface Frequency (MHz)	N/A			
VCO Rate (MHz)	N/A			

PCIe/XAUI Fabric SPLL Configuration

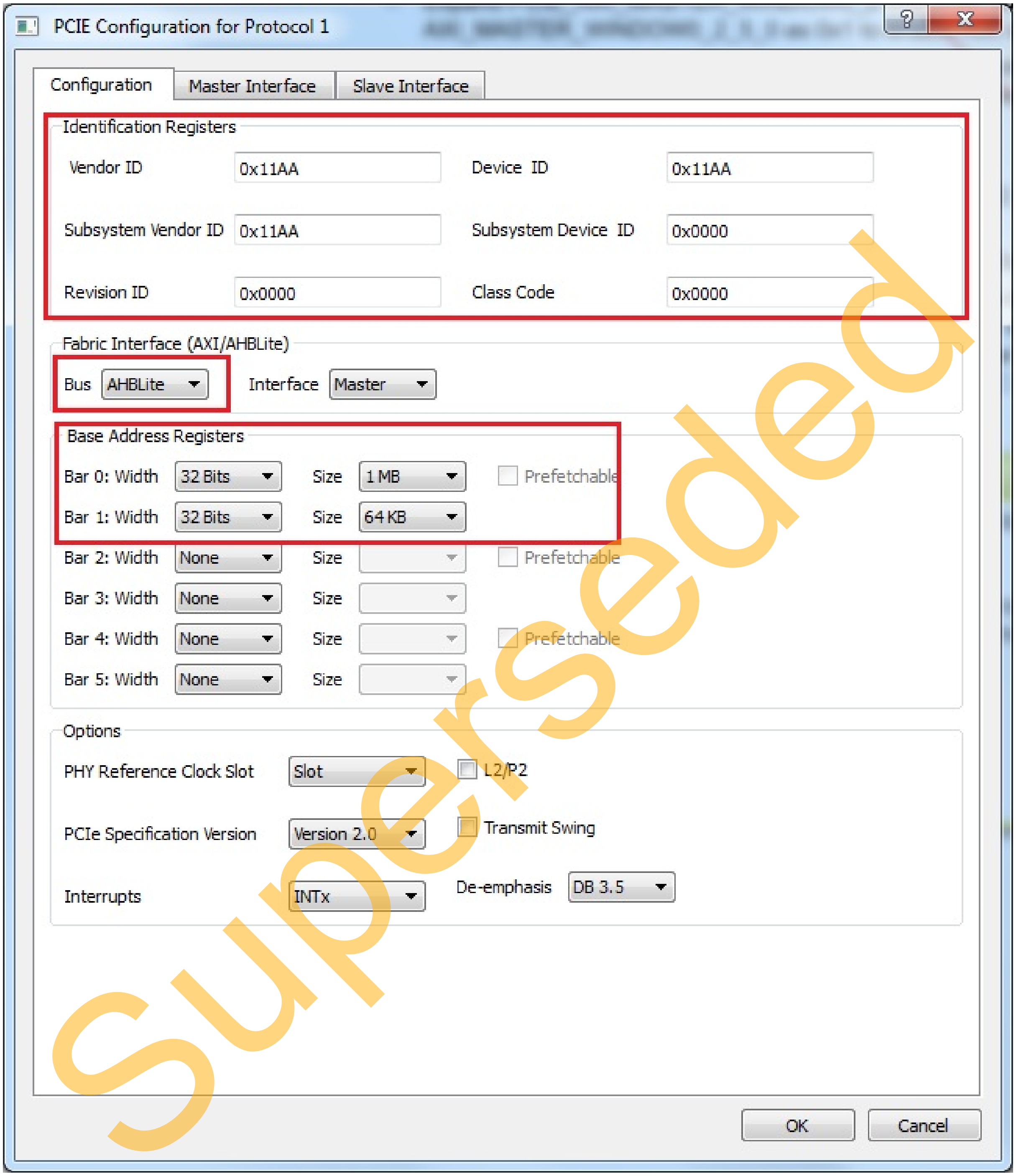
CLK_BASE Frequency: 100 MHz

Figure 20 • **SERDES Configurator**

Superseded

3. Click **Configure PCIe** in Protocol1 as shown in Figure 20 on page 19. Following settings are made in the Configuration tab as shown in Figure 21 on page 21.
   – Fabric Interface (AXI/AHBLite)
     - Bus: select as AHBLite from the drop-down list
   – Base Address Registers
     - BAR 0 Width: 32-bit, Size: 1 MB (to access MSS Peripheral address space)
     - BAR 1 Width: 32-bit, Size: 64 KB (to access eSRAM memory)
   – Identification Registers
     - Device ID: 0x11AA (MicroSemi ID)
     - Subsystem Vendor ID: 0x11AA (MicroSemi ID)

Superseded



PCIE Configuration for Protocol 1

Identification Registers

Vendor ID	0x11AA	Device ID	0x11AA
Subsystem Vendor ID	0x11AA	Subsystem Device ID	0x0000
Revision ID	0x0000	Class Code	0x0000

Fabric Interface (AXI/AHBLite)

Bus AHBLite   Interface Master

Base Address Registers

Bar 0: Width	32 Bits	Size	1 MB	□ Prefetchable
Bar 1: Width	32 Bits	Size	64 KB	
Bar 2: Width	None	Size		□ Prefetchable
Bar 3: Width	None	Size		
Bar 4: Width	None	Size		□ Prefetchable
Bar 5: Width	None	Size		

Options

PHY Reference Clock Slot	Slot	□ L2/P2	
PCIe Specification Version	Version 2.0	□ Transmit Swing	
Interrupts	INTx	De-emphasis	DB 3.5

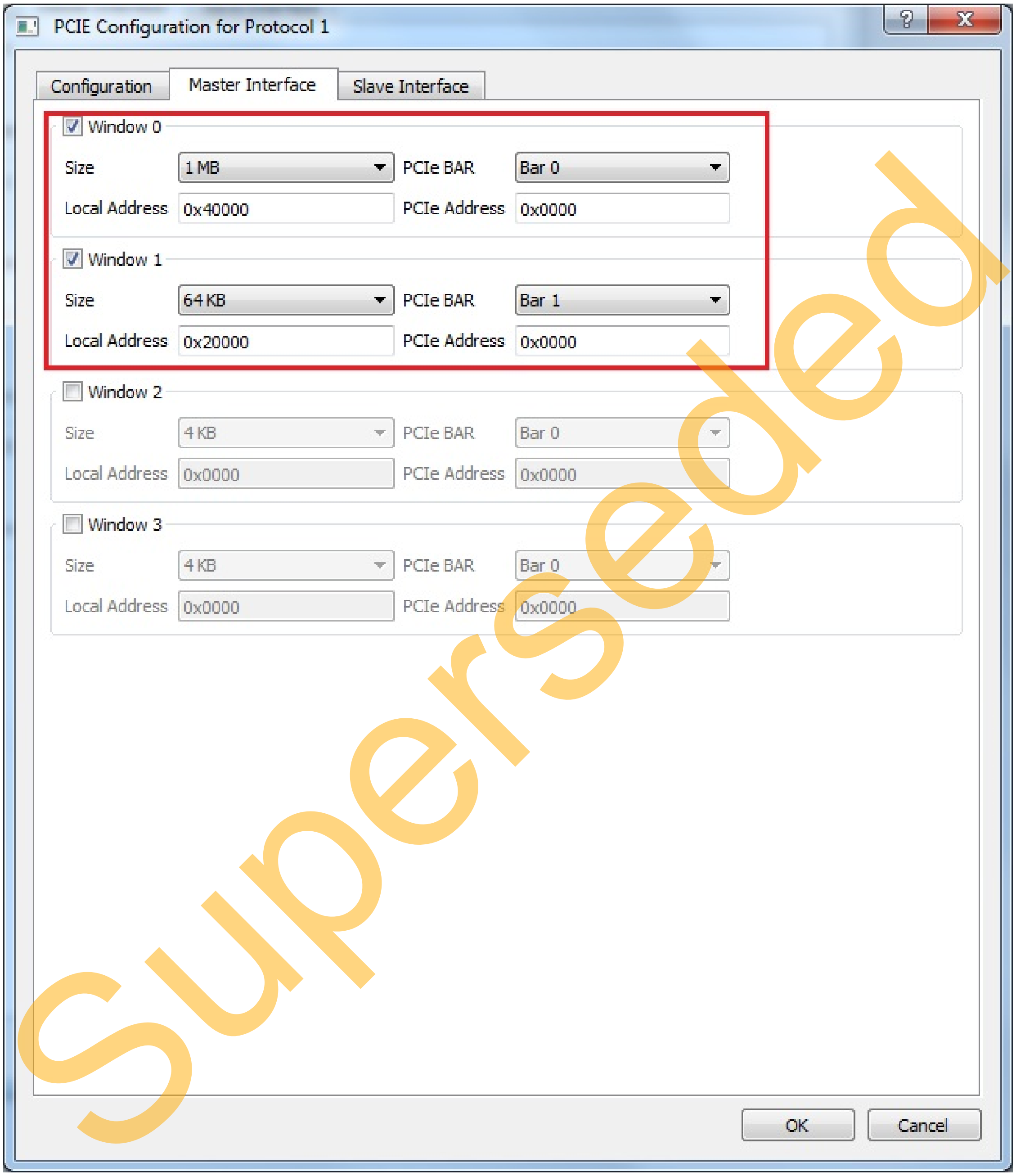
Superseded

Figure 21 **• PCIe Configuration for Protocol 1**

4. Click the **Master Interface** tab to configure the PCIe master windows. The PCIe AXI master windows are used to translate the PCIe address domain to the local device address domain. In this demo the PCIe AXI master windows are used to translate the address of BAR0 and BAR1 to CoreGPIO address and COREAHBLSRAM address. Make settings as shown in Figure 22.
   – Select Window 0 and configure following settings:
     - Size: Select as 1MB from the drop-down list
     - PCIe BAR: Select as Bar0 from the drop-down list
     - Local Address: Enter values as 0x40000 to translate the BAR0 address space to CoreGPIO address (0x4000_0000)
   – Select Window 1 and configure following settings
     - Size: Select as 64KB from the drop-down list
     - PCIe BAR: Select as Bar1 from the drop-down list
     - Local Address: Enter values as 0x20000 to translate the BAR1 address space to COREAHBLSRAM address (0x2000_0000)

Superseded

For more information on PCIe address translation, refer to the "Address Translation on the AXI Master Interface" section of the *SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide*.



☑ Window 0

Size: 1 MB — PCIe BAR: Bar 0
Local Address: 0x40000 — PCIe Address: 0x0000

☑ Window 1

Size: 64 KB — PCIe BAR: Bar 1
Local Address: 0x20000 — PCIe Address: 0x0000

☐ Window 2

Size: 4 KB — PCIe BAR: Bar 0
Local Address: 0x0000 — PCIe Address: 0x0000

☐ Window 3

Size: 4 KB — PCIe BAR: Bar 0
Local Address: 0x0000 — PCIe Address: 0x0000

Superseded

***Figure 22* • PCIe Configuration Memory**

5. Click **OK** to close PCIe Configuration window.
6. Click **OK** to save and close the High Speed Serial Interface Configurator.

Instantiating Debounce Logic in PCIe_Demo SmartDesign

1. The demo provides a push button on the SmartFusion2 Evaluation Kit to send an interrupt to the Host PC. This push button generates switch bounce that causes multiple interrupts to PCIe. Debounce logic is required to avoid the switch bounce.
2. To add the debounce logic to the PCIe demo design, click **File > Import > HDL Source files**.
3. Browse to the Debounce.v or Debounce.vhd file location in the design files folder: *M2S90_PCIE_Control_DEMO_DF/Source Files*. Figure 23 shows the DEBOUNCE component in the Design Hierarchy window.

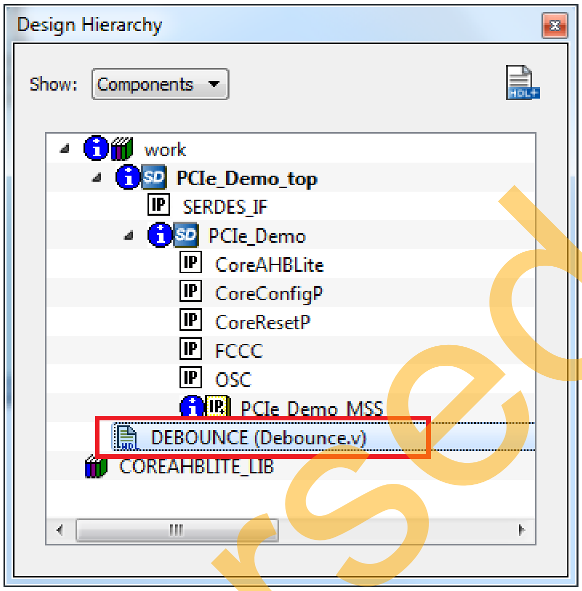
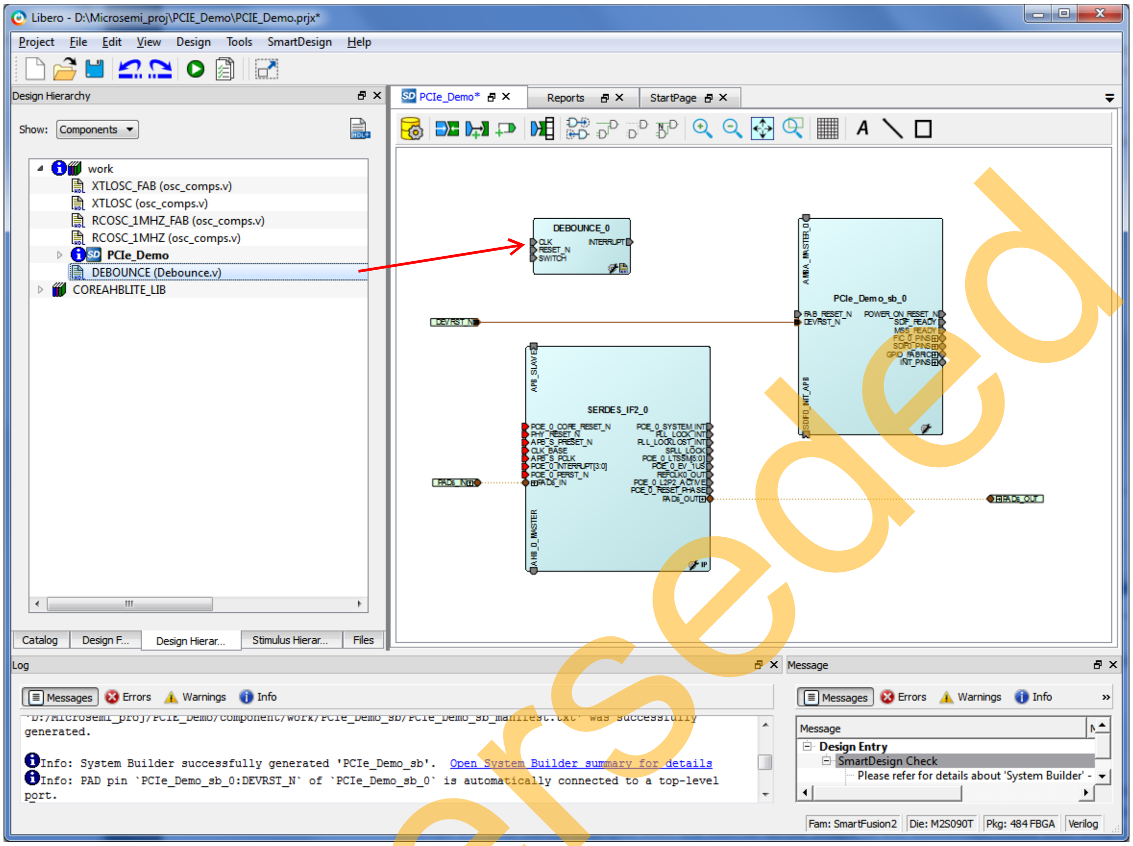


Figure 23 • **DEBOUNCE Component in Design Hierarchy Window**

Superseded

4. Click the **PCIe_Demo** tab and drag the **DEBOUNCE** component from the **Design Hierarchy** into the **PCIe_Demo SmartDesign** canvas as shown in Figure 24. A SmartDesign symbol for the Verilog HDL file is automatically generated.



```
`D:/Microsemi_proj/PCIE_Demo/component/work/PCIe_Demo_sb/PCIe_Demo_sb_manifest.txt` was successfully
generated.

Info: System Builder successfully generated 'PCIe_Demo_sb'.  Open System Builder summary for details
Info: PAD pin `PCIe_Demo_sb_0:DEVRST_N` of `PCIe_Demo_sb_0` is automatically connected to a top-level
port.
```

Figure 24 • **DEBOUNCE Component in Design Hierarchy**

The PCIe_Demo is displayed as shown in Figure 26 on page 27. Connect the pins of all the blocks as described in the "Connecting Components in PCIe_Demo SmartDesign" section.

Connecting Components in PCIe_Demo SmartDesign

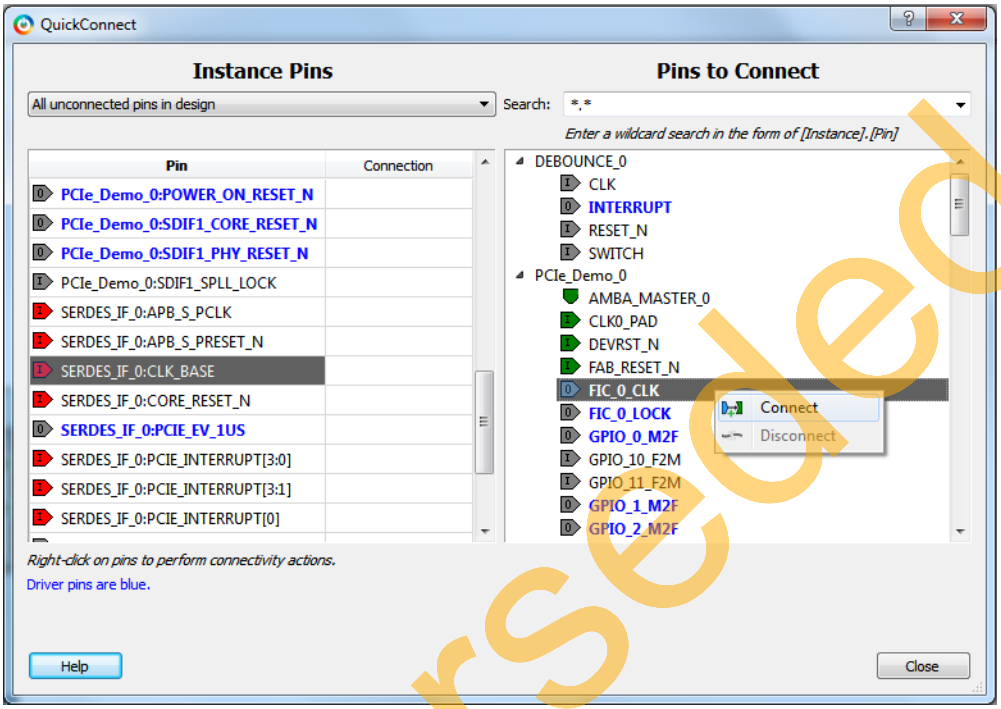
There are three methods for connecting components in PCIe_Demo SmartDesign.

The first method is by using the **Connection Mode** option. To use this method, change the SmartDesign to connection mode by clicking **Connection Mode** on the SmartDesign window, as shown in Figure 26 on page 27. The cursor changes from the normal arrow shape to the connection mode icon shape. To make a connection in this mode, click on the first pin and drag-drop to the second pin that you want to connect.

The second method is by selecting the pins to be connected together and selecting **Connect** from the context menu. To select multiple pins to be connected together, press down the **CTRL** key while selecting the pins. Right-click the input source signal and select **Connect** to connect all the signals together. Similarly, select the input source signal, right-click it, and select **Disconnect** to disconnect the signals already connected.

Superseded

The third method is by using the **Quick Connect** option. To use this method, change the SmartDesign to quick connect mode by clicking **Quick Connect** mode on the SmartDesign window, as shown in Figure 25. **Quick Connect** window opens. Find the Instance Pin that needs to be connected and click to select it. In Pins to Connect, find the pin that needs to be connected, right-click and choose **Connect** as shown in Figure 25.



All unconnected pins in design

Search: *.*

Enter a wildcard search in the form of [Instance].[Pin]

Pin	Connection
PCIe_Demo_0:POWER_ON_RESET_N	
PCIe_Demo_0:SDIF1_CORE_RESET_N	
PCIe_Demo_0:SDIF1_PHY_RESET_N	
PCIe_Demo_0:SDIF1_SPLL_LOCK	
SERDES_IF_0:APB_S_PCLK	
SERDES_IF_0:APB_S_PRESET_N	
SERDES_IF_0:CLK_BASE	
SERDES_IF_0:CORE_RESET_N	
SERDES_IF_0:PCIE_EV_1US	
SERDES_IF_0:PCIE_INTERRUPT[3:0]	
SERDES_IF_0:PCIE_INTERRUPT[3:1]	
SERDES_IF_0:PCIE_INTERRUPT[0]	

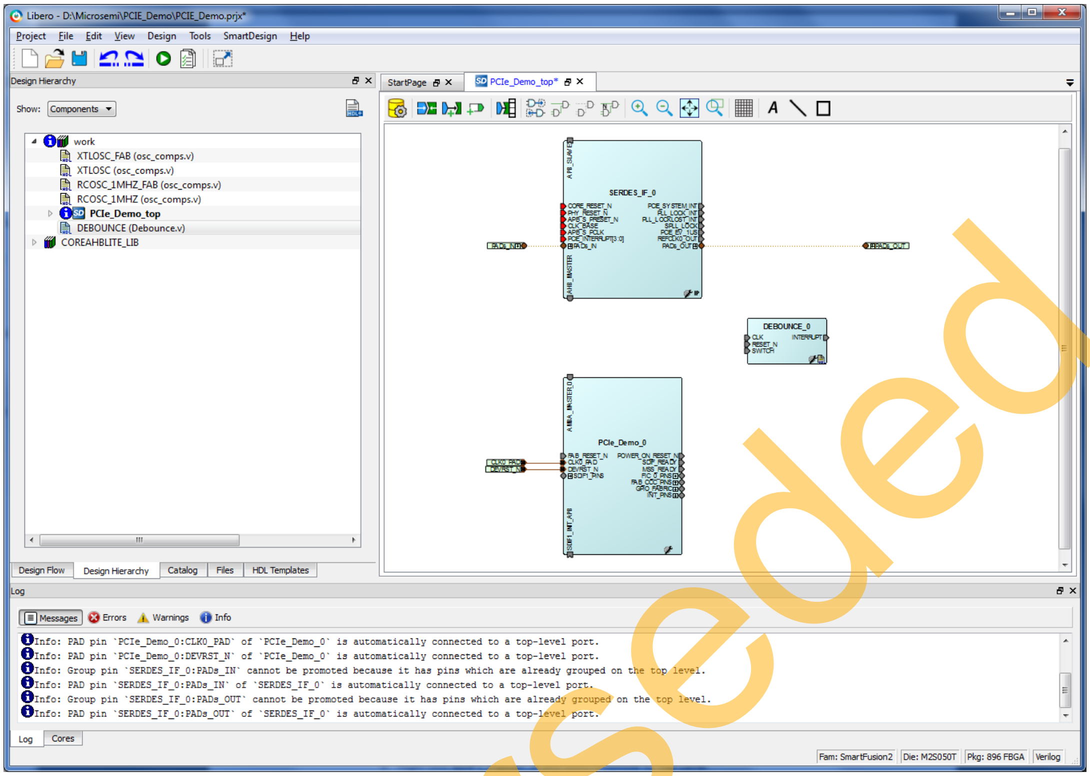
- DEBOUNCE_0
  - CLK
  - **INTERRUPT**
  - RESET_N
  - SWITCH
- PCIe_Demo_0
  - AMBA_MASTER_0
  - CLK0_PAD
  - DEVRST_N
  - FAB_RESET_N
  - FIC_0_CLK
  - **FIC_0_LOCK**
  - **GPIO_0_M2F**
  - GPIO_10_F2M
  - GPIO_11_F2M
  - **GPIO_1_M2F**
  - **GPIO_2_M2F**

Right-click on pins to perform connectivity actions.

Driver pins are blue.

***Figure 25* • Quick Connect Window**

Superseded



```
Info: PAD pin `PCIe_Demo_0:CLK0_PAD` of `PCIe_Demo_0` is automatically connected to a top-level port.
Info: PAD pin `PCIe_Demo_0:DEVRST_N` of `PCIe_Demo_0` is automatically connected to a top-level port.
Info: Group pin `SERDES_IF_0:PADs_IN` cannot be promoted because it has pins which are already grouped on the top level.
Info: PAD pin `SERDES_IF_0:PADs_IN` of `SERDES_IF_0` is automatically connected to a top-level port.
Info: Group pin `SERDES_IF_0:PADs_OUT` cannot be promoted because it has pins which are already grouped on the top level.
Info: PAD pin `SERDES_IF_0:PADs_OUT` of `SERDES_IF_0` is automatically connected to a top-level port.
```

Figure 26 **• PCIe Demo Top in SmartDesign**

Use one of the three options and make the following connections:

1. Expand FIC_0_PINS of PCIe_Demo_sb_0 and make connections as shown in Table 2.
2. Right-click **FIC_0_LOCK** and select **Mark Unused**

***Table 2* • FIC_0_PINS**

From PCIe_Demo_sb_0	To
FIC_0_CLK	CLK_BASE of SERDES_IF2_0
	CLK of DEBOUNCE_0

3. Expand SDIF0_PINS of PCIe_Demo_sb_0 and make connections as shown in Table 3.

***Table 3* • SDIF0_PINS**

From PCIe_Demo_sb_0	To SERDES_IF2_0
SDIF0_PHY_RESET_N	PHY_RESET_N
SDIF0_0_CORE_RESET_N	PCIE_0_CORE_RESET_N
SDIF0_SPLL_LOCK	SPLL_LOCK

4. Right-click **SDIF0_1_CORE_RESET_N** and select **Mark Unused**.
5. Right-click **PCIE_0_PERST_N** and select **Promote to Top Level**.

Superseded

6. Expand INIT_PINS of PCIe_Demo_sb_0 and make connections as shown in Table 4.

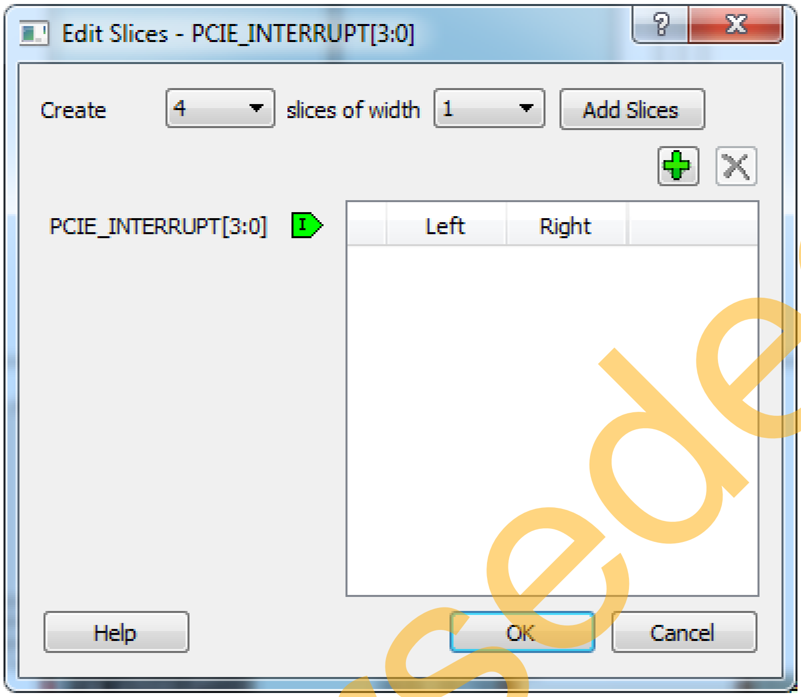
Table 4 • **INIT_PINS**

From PCIe_Demo_sb_0	To SERDES_IF2_0
INIT_APB_S_PCLK	APB_S_PCLK
INIT_APB_S_PRESET_N	APB_S_PRESET_N

7. Right-click **INIT_DONE** and select **Mark Unused**.
8. Connect **MSS_READY** of **PCIe_Demo_sb_0** and **RESET_N** of **DEBOUNCE_0**.
9. Right-click **FAB_RESET_N** of **PCIe_Demo_sb_0** and select **Tie High**.
10. Right-click **GPIO_FABRIC** of **PCIe_Demo_sb_0** and select **Promote to Top Level**.
11. Right-click **POWER_ON_RESET_N** of **PCIe_Demo_sb_0** and select **Mark Unused**.
12. Right-click **SDIF_READY** of **PCIe_Demo_sb_0** and select **Mark Unused**.
13. Connect **AMBA_MASTER_0** of **PCIe_Demo_sb_0** and **AHB_MASTER of SERDES_IF2_0**.
14. Expand **FAB_CCC_PINS**, right-click **FAB_CCC_GL3** and select **Mark Unused**.
15. Connect **SDIF1_INIT_APB** of **PCIe_Demo_sb_0** and **APB_SLAVE of SERDES_IF2_0**.
16. Right-click the **SWITCH** of **DEBOUNCE_0** and select **Promote to Top Level**.
17. Select the following ports of **SERDES_IF2_0** by pressing down the **CTRL** key, right-click, and select **Mark Unused**.
    - PCIE_SYSTEM_INT
    - PLL_LOCK_INT
    - PLL_LOCKLOST_INT
    - PCIE_EV_1US
    - REFCLK0_OUT
    - PCIE_0_LTSSM[5:0]
    - PCIE_0_L2P2_ACTIVE
    - PCIE_0_RESET_PHASE

Superseded

The PCIe supports four interrupts. This design uses only one interrupt out of four by connecting the unused interrupts to logic ‘0’. To connect unused interrupt pins to logic ‘0’ split the interrupt pins to two groups. To do that right-click the **PCIE_INTERRUPT[3:0]** of **SERDES_IF2_0** and select **Edit Slices**. The Edit Slices window is displayed as shown in Figure 27.

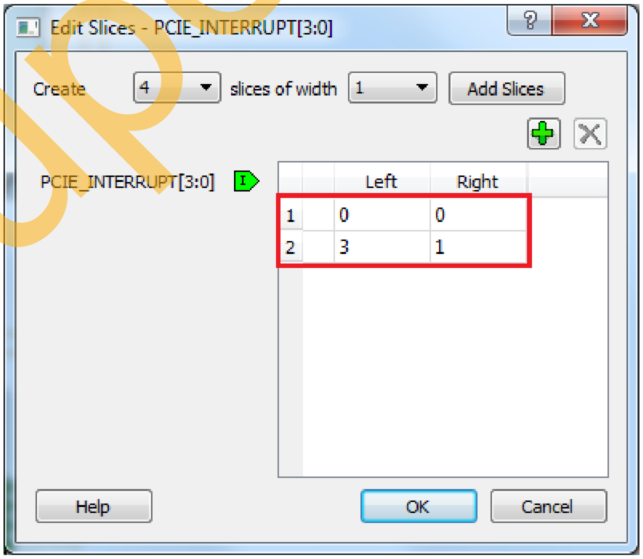

Edit Slices - PCIE_INTERRUPT[3:0]

Create 4 slices of width 1

Left	Right

Figure 27 • **Edit Slices**

18. Click the + sign and create a slice with the Left index 0 and the Right index 0. Click + again to create a second slice with Left index 3 and Right index 1 as shown in Figure 28.


Edit Slices - PCIE_INTERRUPT[3:0]

Create 4 slices of width 1

	Left	Right
1	0	0
2	3	1

Figure 28 • **Edit Slices**

19. Expand **PCIE_INTERRUPT[3:0]**, right-click the **PCIE_INTERRUPT[3:1]**, and select **Tie low**.

Superseded

20. Connect **INTERRUPT** of **DEBOUNCE_0** to the **PCIE_INTERRUPT[0]** of **SERDES_IF2_0**.
21. Click **Auto arrange instances** to arrange the instances and click **File > Save**. The PCIe_Demo is displayed as shown in Figure 29.

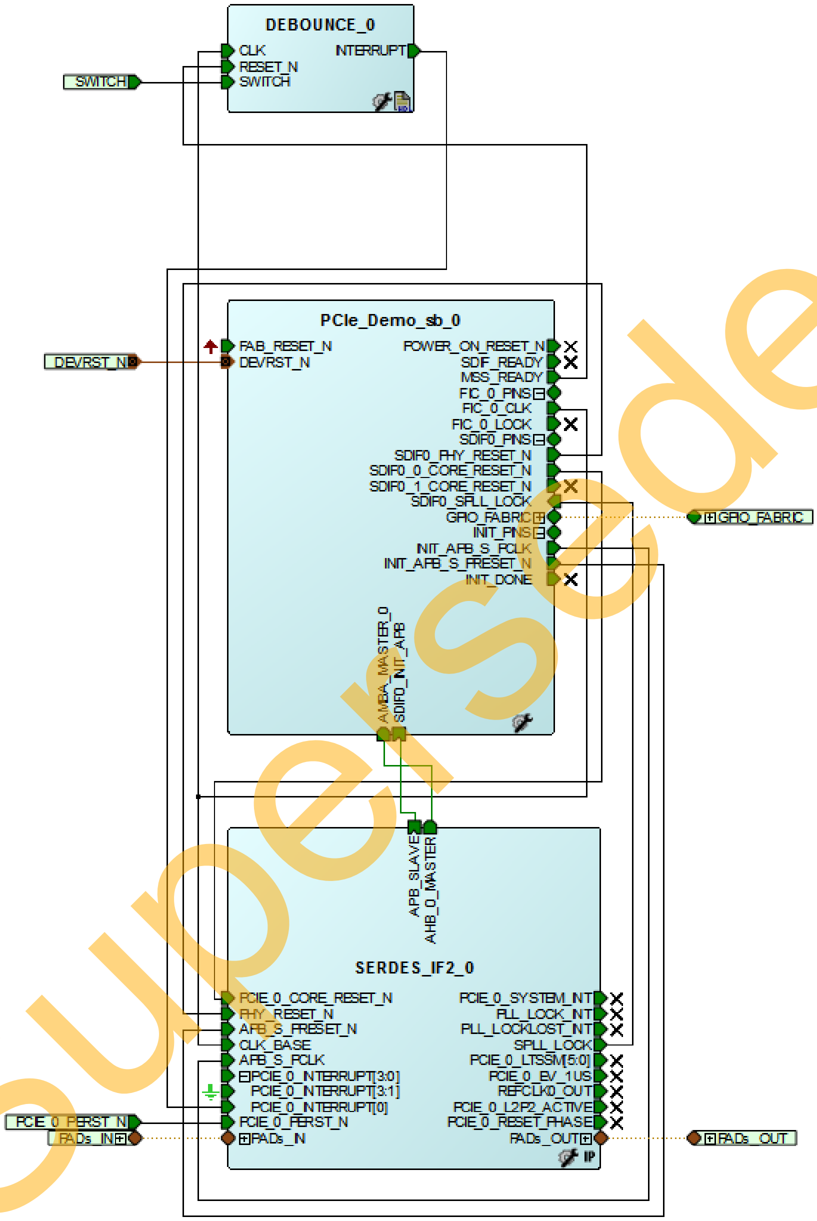


Figure 29 • **PCIe Demo Top Design**

22. Click the **PCIe_Demo** tab and click **Generate Component** icon as shown in Figure 30.

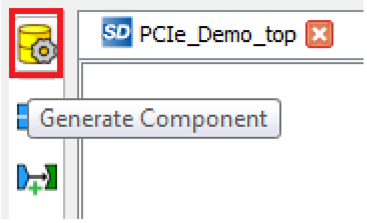
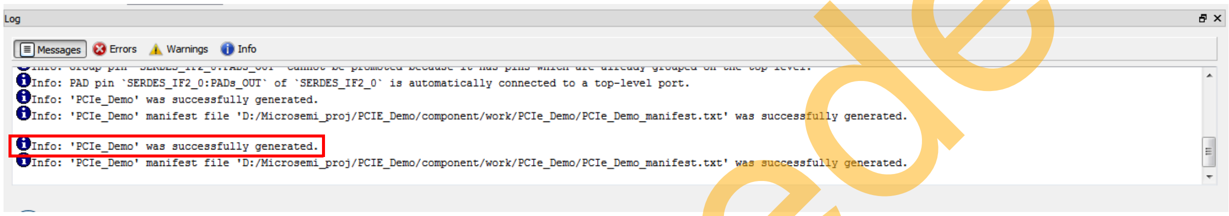


***Figure 30* • Generate Component**

The message “PCIe_Demo” was successfully generated and is displayed in the Libero SoC log window if the design is generated without any error. The log window is displayed on a successful component generation as shown in Figure 31



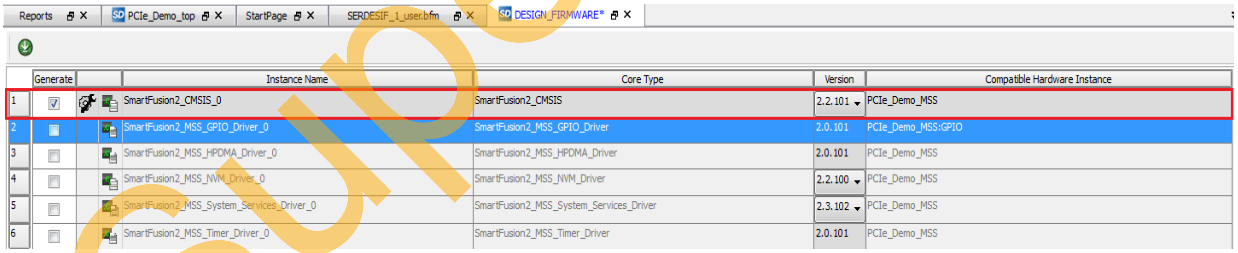
```
Info: PAD pin `SERDES_IF2_0:PADs_OUT` of `SERDES_IF2_0` is automatically connected to a top-level port.
Info: 'PCIe_Demo' was successfully generated.
Info: 'PCIe_Demo' manifest file 'D:/Microsemi_proj/PCIE_Demo/component/work/PCIe_Demo/PCIe_Demo_manifest.txt' was successfully generated.

Info: 'PCIe_Demo' was successfully generated.
Info: 'PCIe_Demo' manifest file 'D:/Microsemi_proj/PCIE_Demo/component/work/PCIe_Demo/PCIe_Demo_manifest.txt' was successfully generated.
```

***Figure 31* • Log Window**

Configuring and Generating Firmware

The following steps describe how to configure and generate firmware.

1. Double click **Configure Firmware Cores** under **Handoff Design for Firmware Development** in Design Flow and clear all drivers except CMSIS as shown in Figure 32.

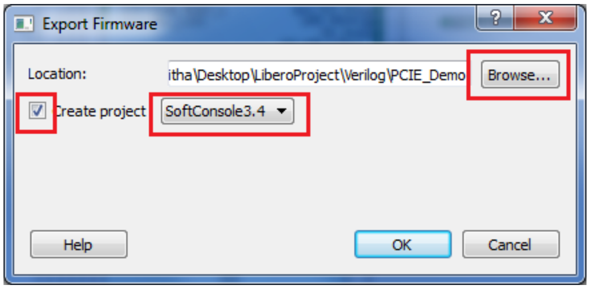


	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	☑	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.2.101	PCIe_Demo_MSS
2	☐	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	2.0.101	PCIe_Demo_MSS:GPIO
3	☐	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	2.0.101	PCIe_Demo_MSS
4	☐	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.2.100	PCIe_Demo_MSS
5	☐	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	2.3.102	PCIe_Demo_MSS
6	☐	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	2.0.101	PCIe_Demo_MSS

***Figure 32* • Configuring Firmware**

Superseded

Click **Export Firmware**. The **Export Firmware** dialog box is displayed as shown in Figure 33.

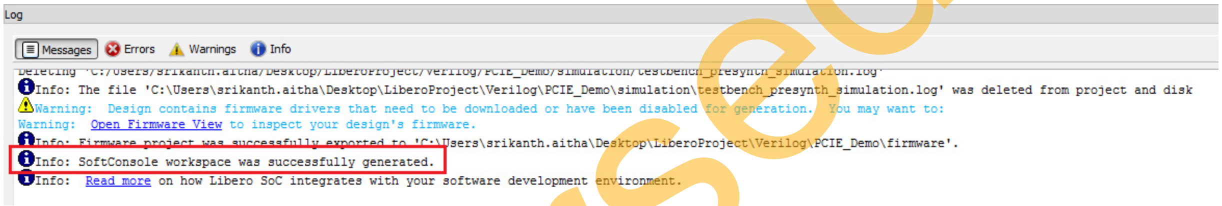


Location: itha\Desktop\LiberoProject\Verilog\PCIE_Demo

☑ Create project SoftConsole3.4

Figure 33 • **Export Firmware Dialog Box**

2. Browse the **Location** to export the firmware project.
3. Select the **Create project** check box.
4. Select **SoftConsole3.4** from the drop down list.
5. Click **OK**. The successful firmware generation window is displayed.
6. Click **OK**. The log window is displayed as shown in Figure 34.



```
Deleting 'C:/Users/srikanth.aitha/Desktop/LiberoProject/Verilog/PCIE_Demo/simulation/testbench_presynth_simulation.log'
Info: The file 'C:\Users\srikanth.aitha\Desktop\LiberoProject\Verilog\PCIE_Demo\simulation\testbench_presynth_simulation.log' was deleted from project and disk
Warning:  Design contains firmware drivers that need to be downloaded or have been disabled for generation.  You may want to:
Warning:  Open Firmware View to inspect your design's firmware.
Info: Firmware project was successfully exported to 'C:\Users\srikanth.aitha\Desktop\LiberoProject\Verilog\PCIE_Demo\firmware'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.
```

Figure 34 • **Log Window**

Superseded

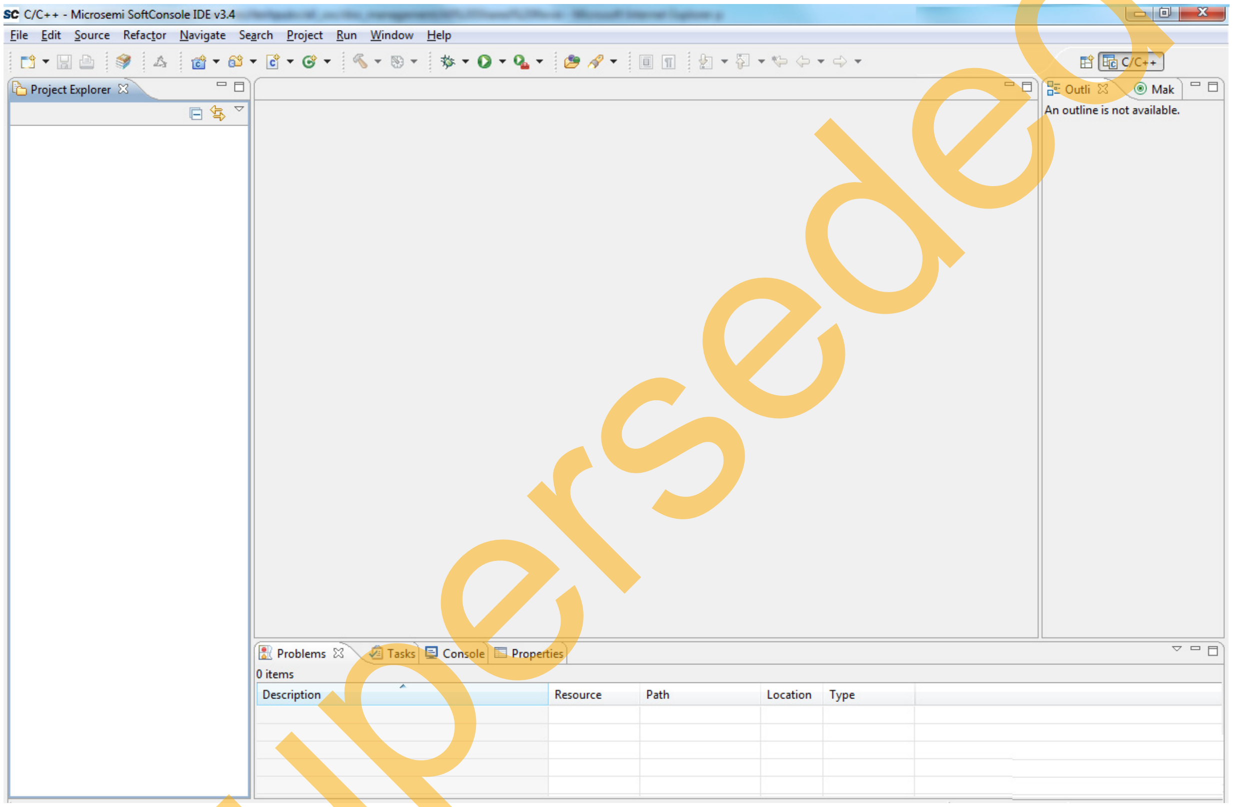
Step 2: Creating an eNVM Client

The HDL and logical design portion of the demo is now complete. The following sections describe the creation of the Cortex-M3 firmware used to initialize the MSS and SERDESIF.

The eNVM client has to be uploaded with the firmware application to initialize the SERDESIF through **CoreConfigP**. The Cortex-M3 processor executes the code in the eNVM after the SmartFusion2 device has been reset. In this design the eNVM client is created with the firmware application code to initialize the SERDESIF.

The following steps describe how to create an eNVM Client:

1. To build the firmware eNVM client, invoke the standalone SoftConsole IDE. The **SoftConsoleIDE Project Workspace** window is displayed as shown in Figure 35.



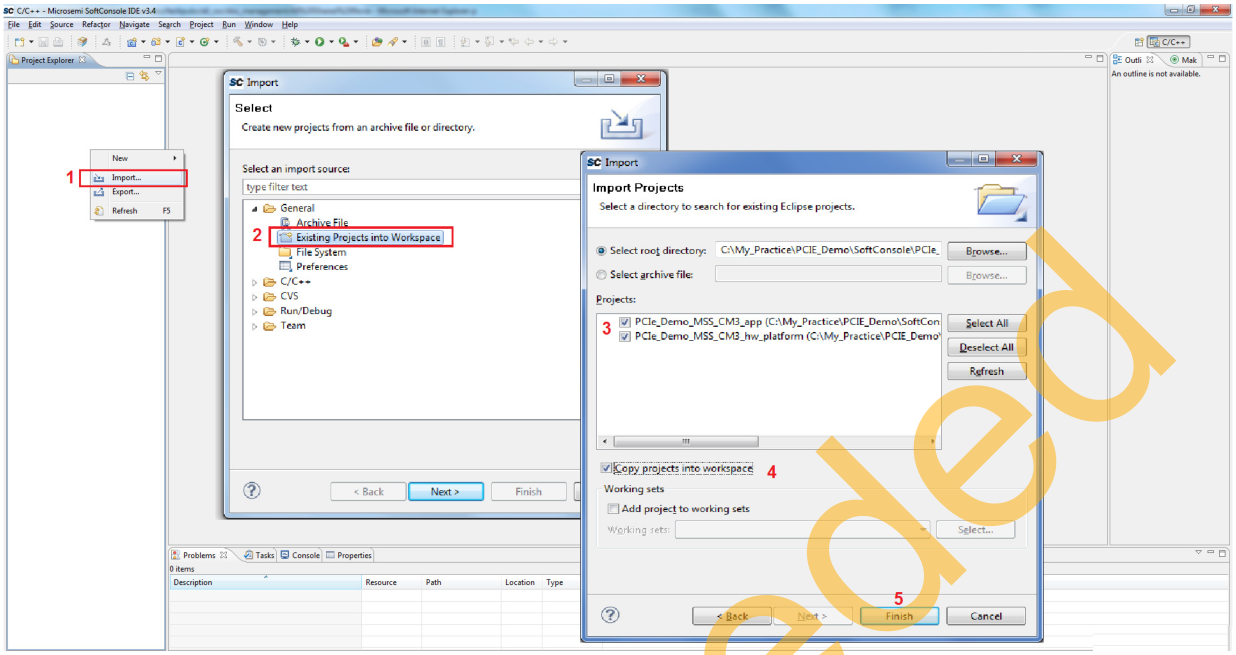
An outline is not available.

0 items

Figure 35 • **SoftConsole IDE Project Workspace**

Superseded

2. Import the existing project into workspace as shown in Figure 36.



● Select root directory: C:\My_Practice\PCIE_Demo\SoftConsole\PCIe_

○ Select archive file:

☑ PCIe_Demo_MSS_CM3_app (C:\My_Practice\PCIE_Demo\SoftCon

☑ PCIe_Demo_MSS_CM3_hw_platform (C:\My_Practice\PCIE_Demo\

☑ Copy projects into workspace

□ Add project to working sets

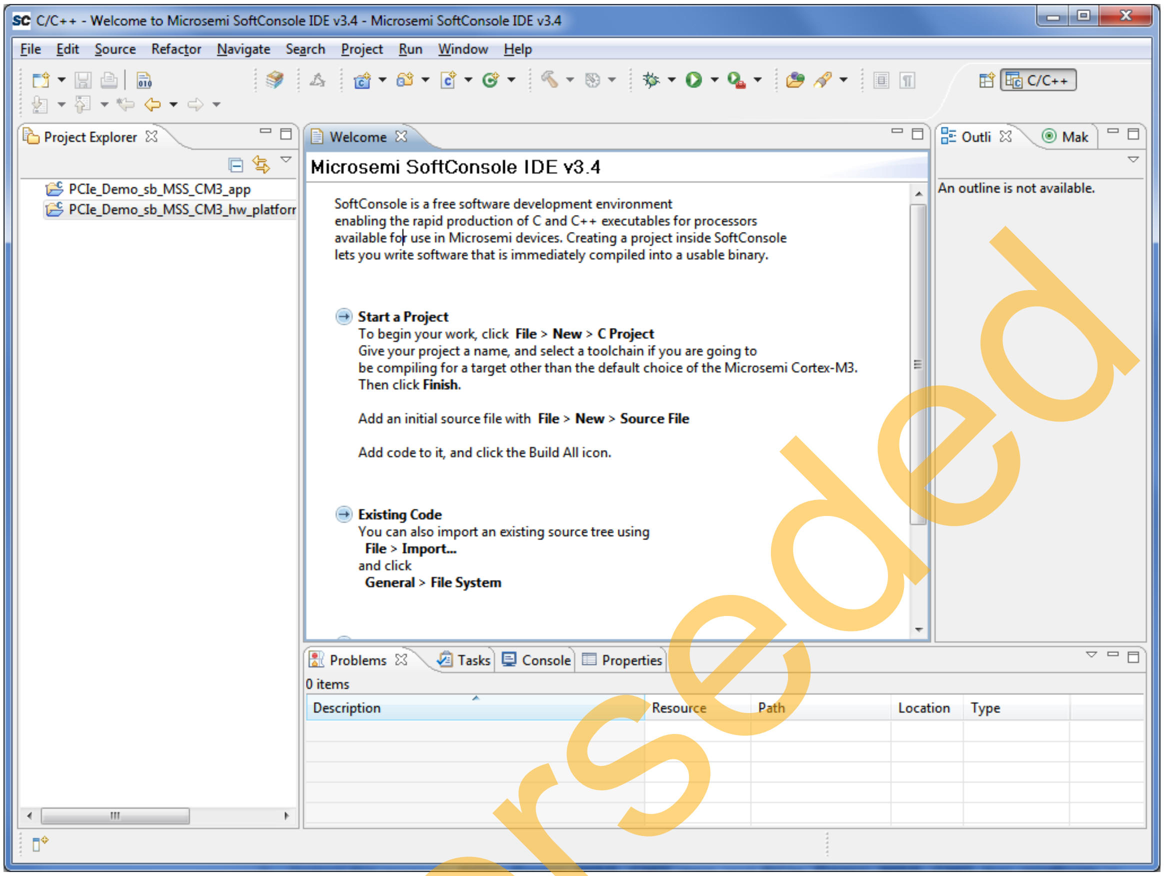
Working sets:

Figure 36 **• Importing Existing Project into Workspace**

3. Right-click **Project Explorer** tab on the left pane and select **Import…**. The **Import** dialog box is displayed.
4. Select **Existing Project into Workspace** under **General** folder and click **Next**. The **Import Projects** dialog box is displayed.
5. Click **Browse** to navigate to the SoftConsole project folder.
6. Select **PCIe_Demo_sb_MSS_CM3_app** and **PCIe_Demo_sb_MSS_CM3_hw_platform** check boxes under **Projects**.
7. Select **Copy projects into workspace** check box.

Superseded

8. Click **Finish**. The **SoftConsole Workspace** window is displayed as shown in Figure 37.


C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
C/C++
Project Explorer
PCIe_Demo_sb_MSS_CM3_app
PCIe_Demo_sb_MSS_CM3_hw_platforr
Welcome
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi Cortex-M3. Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

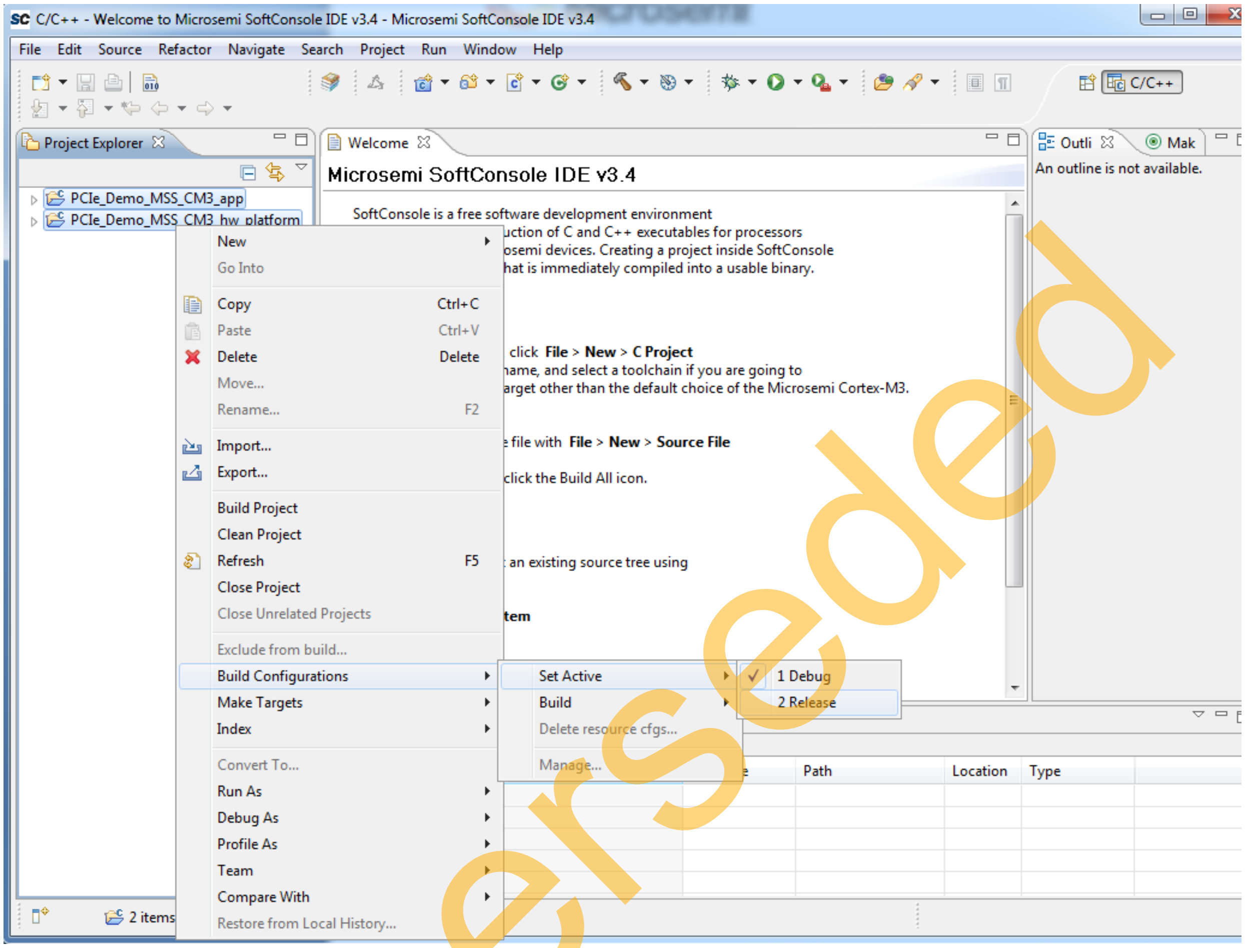
Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

***Figure 37* • SoftConsole Workspace**

Superseded

9. Select the projects **PCIe_Demo_sb_MSS_CM3_app** and **PCIe_Demo_sb_MSS_CM3_hw_platform** in the Project Explorer by using **CTRL** key.

10. Right-click and select **Build Configurations > Set Active > Release** as shown in Figure 38.



SoftConsole is a free software development environment
uction of C and C++ executables for processors
osemi devices. Creating a project inside SoftConsole
hat is immediately compiled into a usable binary.

click **File** > **New** > **C Project**
name, and select a toolchain if you are going to
arget other than the default choice of the Microsemi Cortex-M3.

file with **File** > **New** > **Source File**

click the Build All icon.

an existing source tree using

tem

An outline is not available.

Figure 38 **• Release Mode Option**

Superseded

11. Select **PCIe_Demo_sb_MSS_CM3_app**. Right-click and select **Properties** as shown in Figure 39.

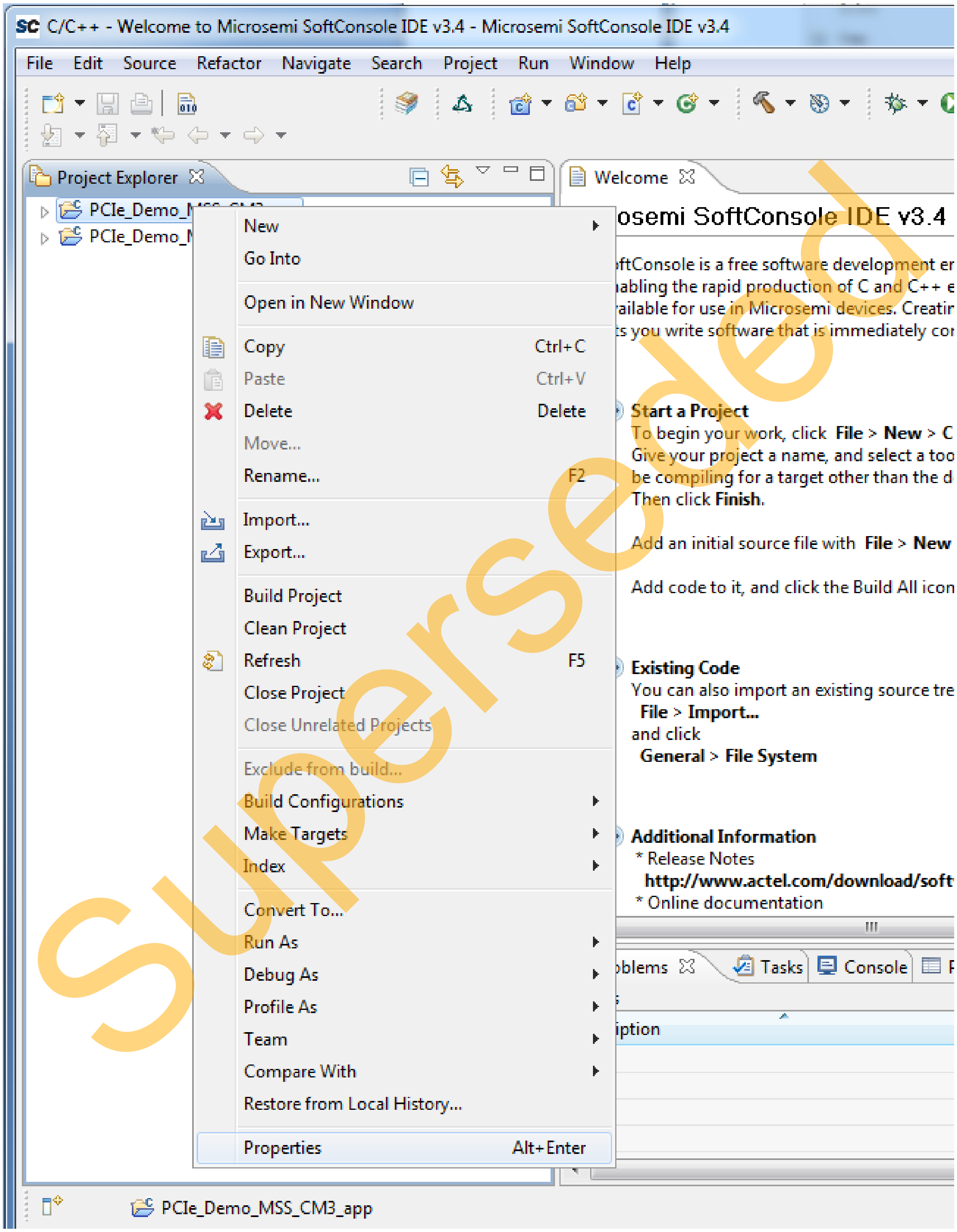
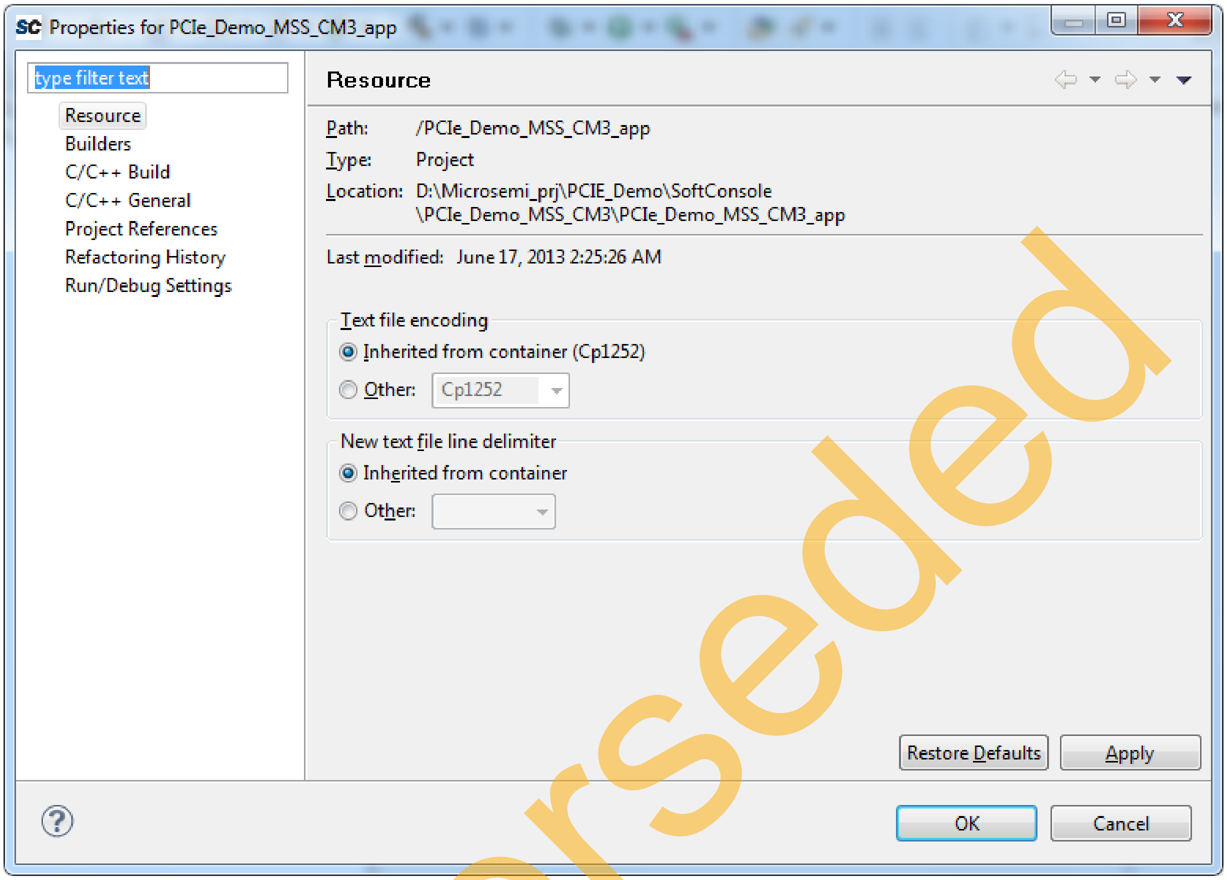

SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
PCIe_Demo_MSS_CM3
PCIe_Demo_M
New
Go Into
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Exclude from build...
Build Configurations
Make Targets
Index
Convert To...
Run As
Debug As
Profile As
Team
Compare With
Restore from Local History...
Properties Alt+Enter
Welcome
osemi SoftConsole IDE v3.4
ftConsole is a free software development er
abling the rapid production of C and C++ e
ailable for use in Microsemi devices. Creatir
s you write software that is immediately cor
Start a Project
To begin your work, click File > New > C
Give your project a name, and select a too
be compiling for a target other than the d
Then click Finish.
Add an initial source file with File > New
Add code to it, and click the Build All icon
Existing Code
You can also import an existing source tre
File > Import...
and click
General > File System
Additional Information
* Release Notes
http://www.actel.com/download/soft
* Online documentation
oblems Tasks Console
iption
PCIe_Demo_MSS_CM3_app
Superseded

Figure 39 • **Properties Option**

The **Properties for PCIe_Demo_sb_MSS_CM3_app** window is displayed as shown in Figure 40.



Path: /PCIe_Demo_MSS_CM3_app

Type: Project

Location: D:\Microsemi_prj\PCIE_Demo\SoftConsole \PCIe_Demo_MSS_CM3\PCIe_Demo_MSS_CM3_app

Last modified: June 17, 2013 2:25:26 AM

Text file encoding

● Inherited from container (Cp1252)

○ Other: Cp1252

New text file line delimiter

● Inherited from container

○ Other:

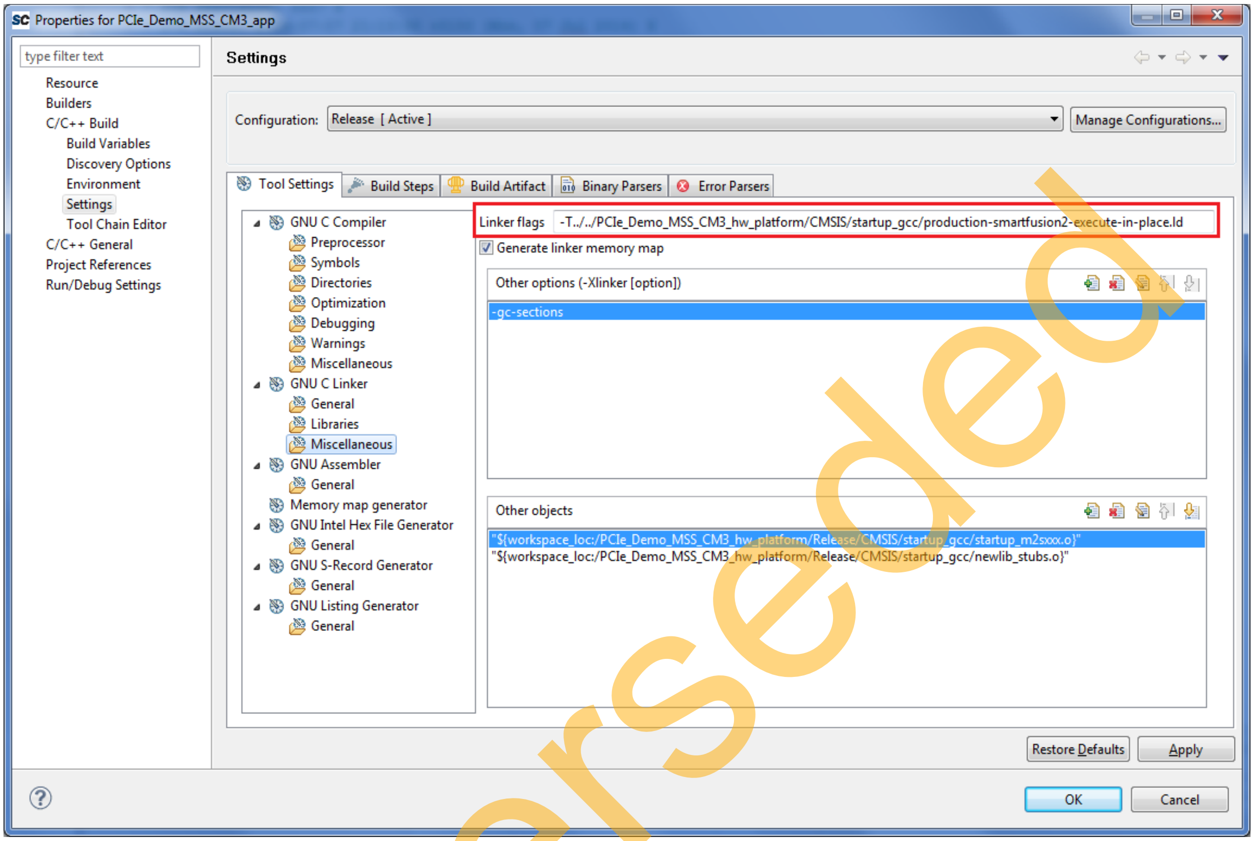
Figure 40 **• Properties Window**

12. In the **Properties for PCIe_Demo_sb_MSS_CM3_app** window, expand the **C/C++ Build** option and select **Settings**.

Superseded

Superseded

13. Select **Miscellaneous** and provide the release mode linker script file to the linker by changing the ‘Linker flags’ field to “*-T../../PCIe_Demo_sb_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-smartfusion2-execute-in-place.ld*” as shown in Figure 41.



Configuration: Release [Active]

Linker flags: -T../../PCIe_Demo_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-smartfusion2-execute-in-place.ld

☑ Generate linker memory map

Other options (-Xlinker [option]):

```
-gc-sections
```

Other objects:

```
"${workspace_loc:/PCIe_Demo_MSS_CM3_hw_platform/Release/CMSIS/startup_gcc/startup_m2sxxx.o}"
"${workspace_loc:/PCIe_Demo_MSS_CM3_hw_platform/Release/CMSIS/startup_gcc/newlib_stubs.o}"
```

Figure 41 **• LD File Option**

14. Click **OK** to close the **Properties for PCIe_Demo_sb_MSS_CM3_app** window.

15. To clean and build the project, select **Project > Clean** as shown in Figure 42.

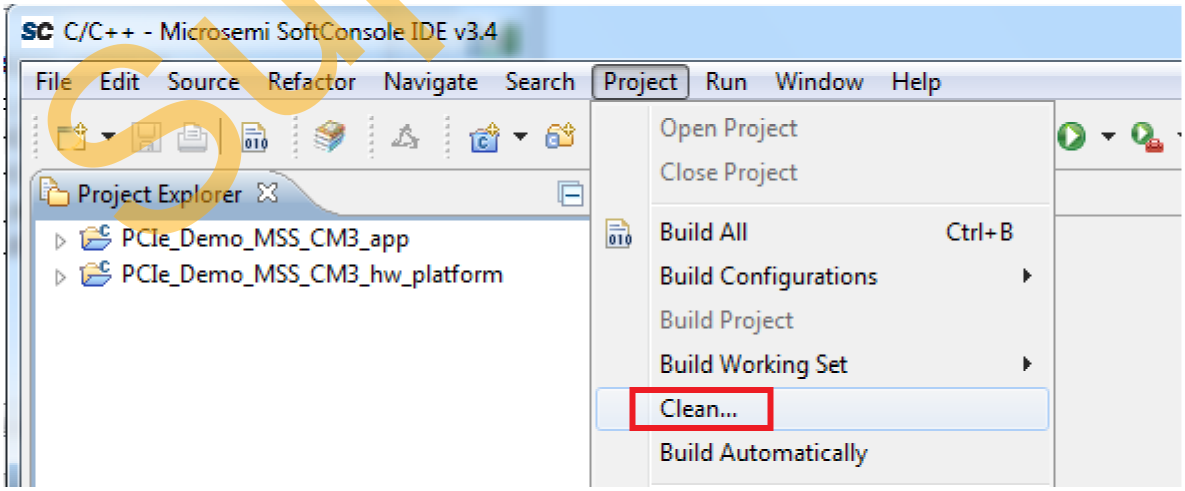
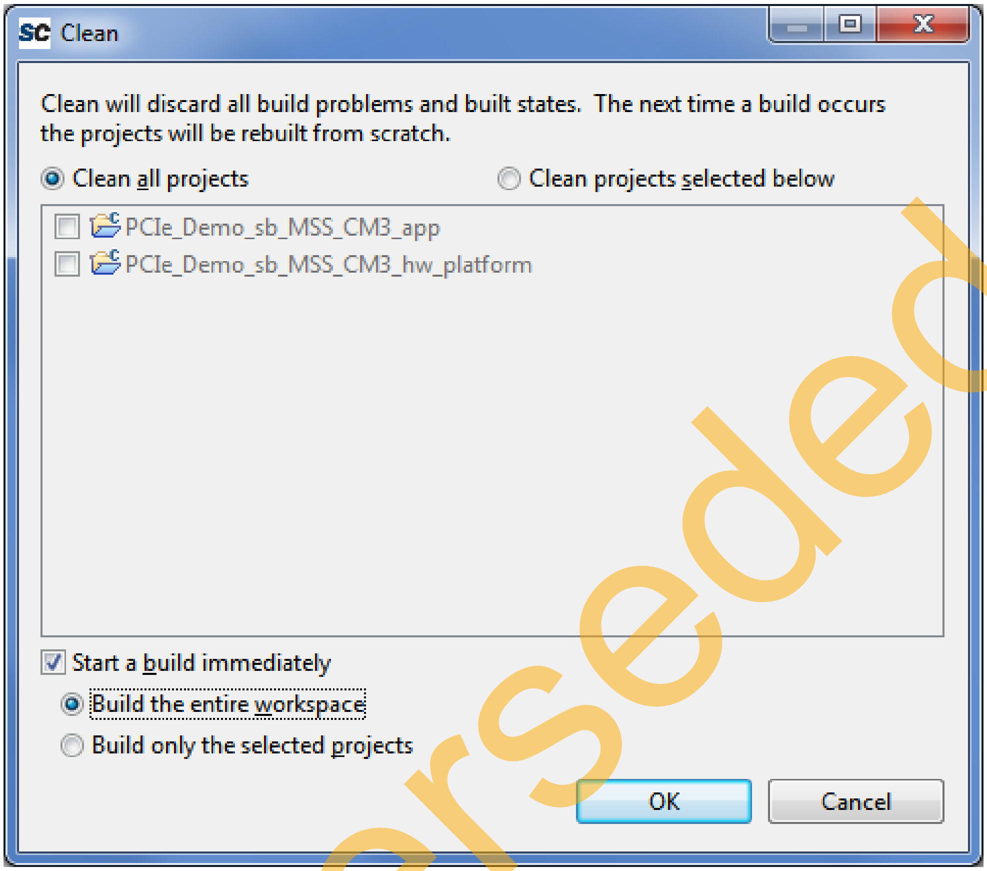


Figure 42 **• Building SoftConsole Project**

16. The **Clean** window is displayed. Click **OK** to build the SoftConsole projects as shown in Figure 43.


Clean

Clean will discard all build problems and built states. The next time a build occurs the projects will be rebuilt from scratch.

● Clean all projects ○ Clean projects selected below

□ PCIe_Demo_sb_MSS_CM3_app
□ PCIe_Demo_sb_MSS_CM3_hw_platform

☑ Start a build immediately

● Build the entire workspace

○ Build only the selected projects

***Figure 43* • Clean and Build SoftConsole Projects**

Superseded

17. The SoftConsole creates a hex file in the **Release** folder under the **PCIe_Demo_sb_MSS_CM3_app** project as shown in Figure 44.

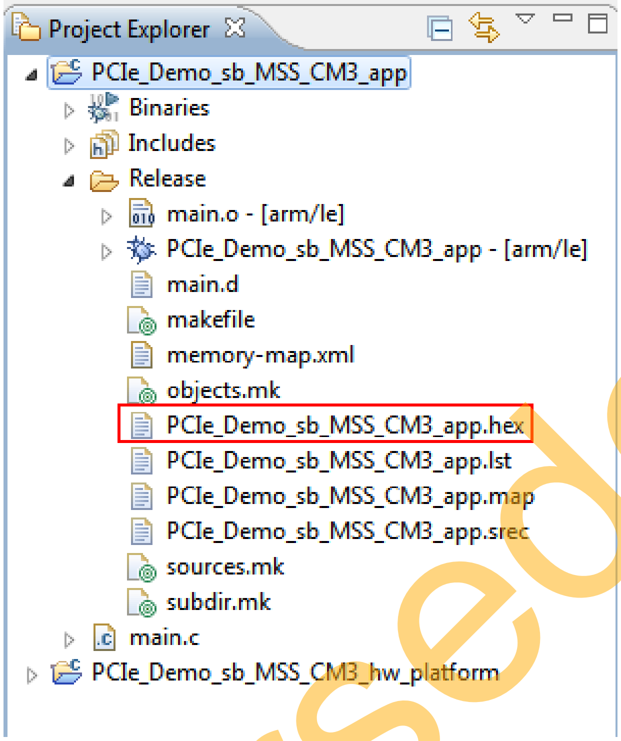
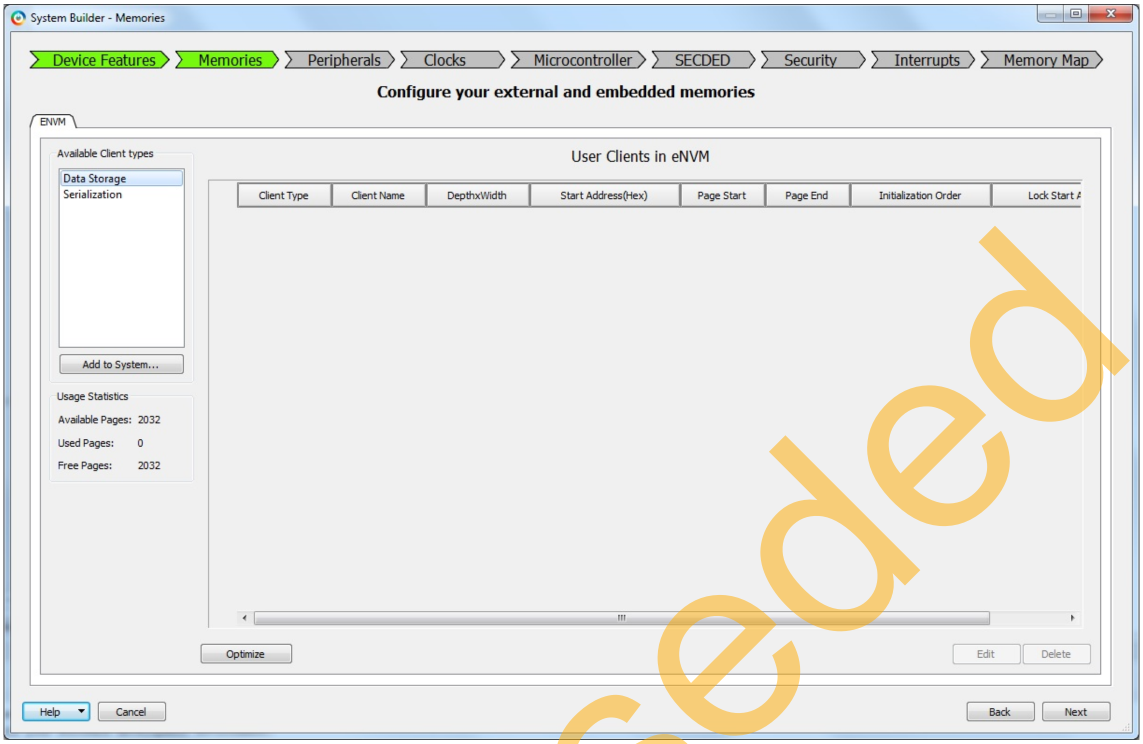

Project Explorer
PCIe_Demo_sb_MSS_CM3_app
Binaries
Includes
Release
main.o - [arm/le]
PCIe_Demo_sb_MSS_CM3_app - [arm/le]
main.d
makefile
memory-map.xml
objects.mk
PCIe_Demo_sb_MSS_CM3_app.hex
PCIe_Demo_sb_MSS_CM3_app.lst
PCIe_Demo_sb_MSS_CM3_app.map
PCIe_Demo_sb_MSS_CM3_app.srec
sources.mk
subdir.mk
main.c
PCIe_Demo_sb_MSS_CM3_hw_platform

Figure 44 **• Generated Hex File**

18. Close the **SoftConsole** project window.
19. Open the Libero project and **PCIe_Demo** tab.
Double-click **PCIe_Demo_sb_0** and go to **System Builder - Memories** tab to add the eNVM data storage client.

Superseded

The eNVM configurator window is displayed as shown in Figure 45.



Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start A

Usage Statistics

Available Pages: 2032

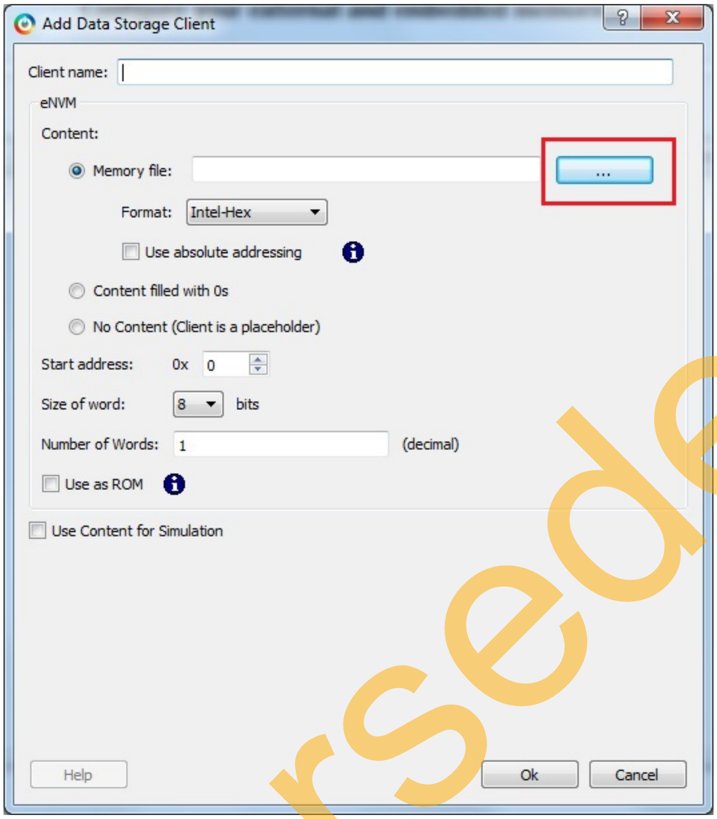
Used Pages: 0

Free Pages: 2032

Figure 45 • System Builder - Memory eNVM

Superseded

20. Select **Data Storage** under the **Available Client types** tab and click **Add to System**. The **Add Data Storage Client** window is displayed as shown in Figure 46.



Add Data Storage Client

Client name:

eNVM

Content:

● Memory file:

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

Number of Words: 1 (decimal)

□ Use as ROM

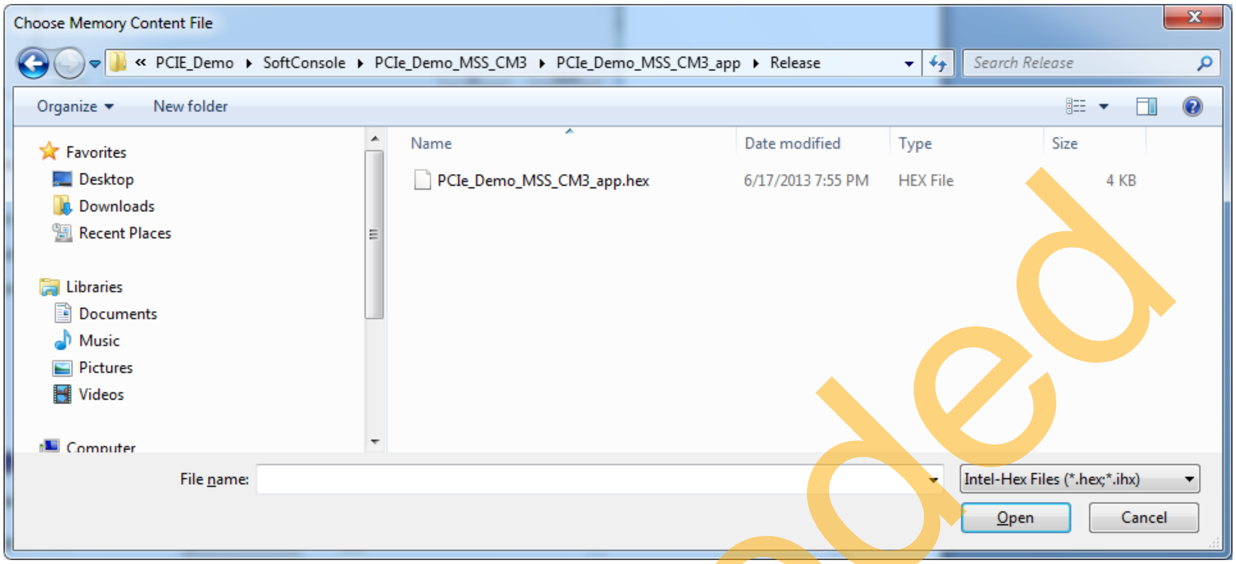
□ Use Content for Simulation

Figure 46 **• Add Data Storage Client**

21. Enter **Client Name** as eNVM in the **Add Data Storage Client** window.

Superseded

22. Browse for the.hex file generated (as shown in Figure 44 on page 41). The generated executable image can be found in the **Release** folder under the SoftConsole project workspace as shown in Figure 47.



Choose Memory Content File

« PCIE_Demo ▸ SoftConsole ▸ PCIe_Demo_MSS_CM3 ▸ PCIe_Demo_MSS_CM3_app ▸ Release

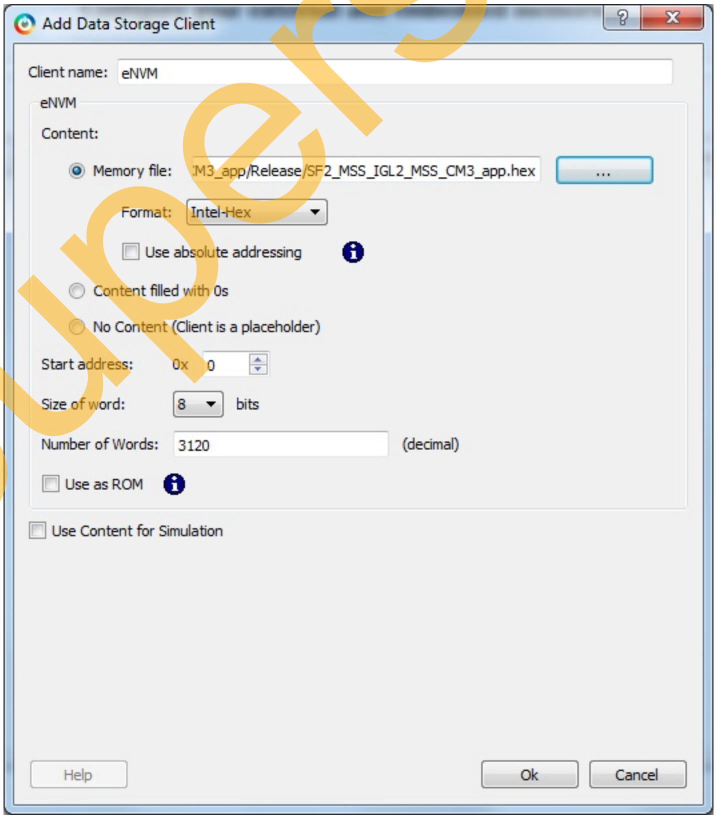
Name	Date modified	Type	Size
PCIe_Demo_MSS_CM3_app.hex	6/17/2013 7:55 PM	HEX File	4 KB

File name:

Intel-Hex Files (*.hex;*.ihx)

Figure 47 • **Browsing for .hex File**

23. Click **OK** in the **Add Data Storage Client** window as shown in Figure 48.



Add Data Storage Client

Client name: eNVM

eNVM

Content:

● Memory file: :M3_app/Release/SF2_MSS_IGL2_MSS_CM3_app.hex

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

Number of Words: 3120 (decimal)

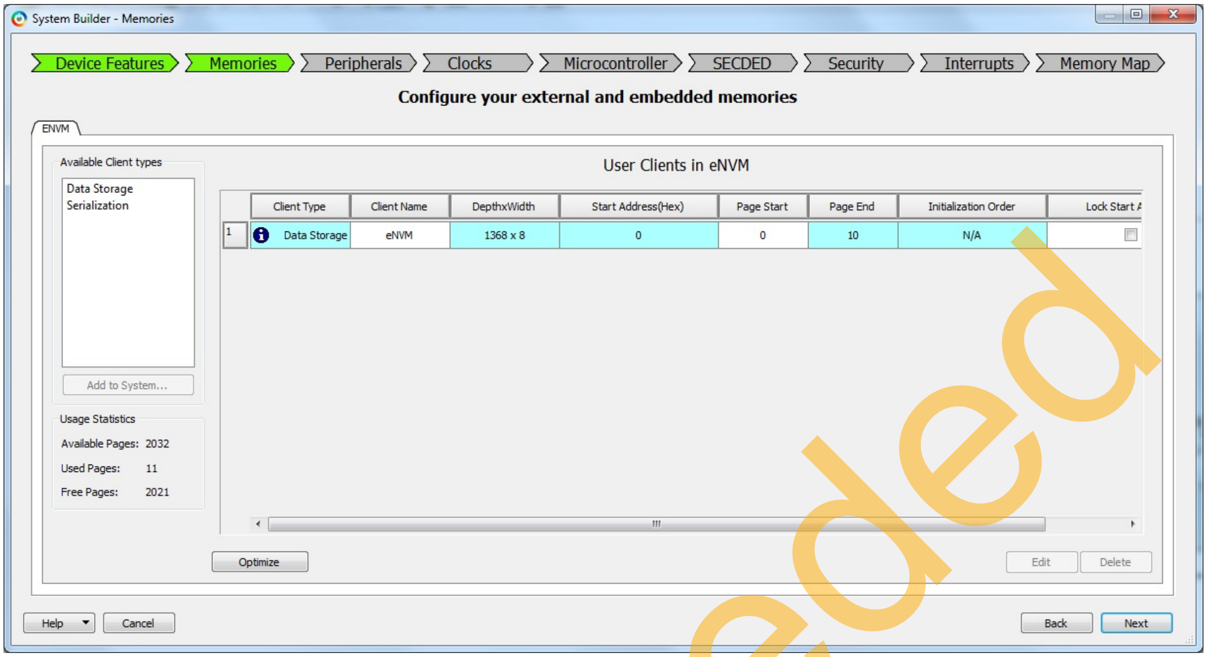
□ Use as ROM

□ Use Content for Simulation

Figure 48 • **Add Data Storage Client**

Superseded

24. Click **Next** and keep the rest of the System Builder tabs as default.



Configure your external and embedded memories

Usage Statistics

Available Pages:	2032
Used Pages:	11
Free Pages:	2021

User Clients in eNVM

	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start A
1	Data Storage	eNVM	1368 x 8	0	0	10	N/A	□

Figure 49 **• Modify Core - ENVM**

25. Save **PCIe_Demo** and regenerate the **PCIe_Demo** component by clicking **Generate Component** in SmartDesign.

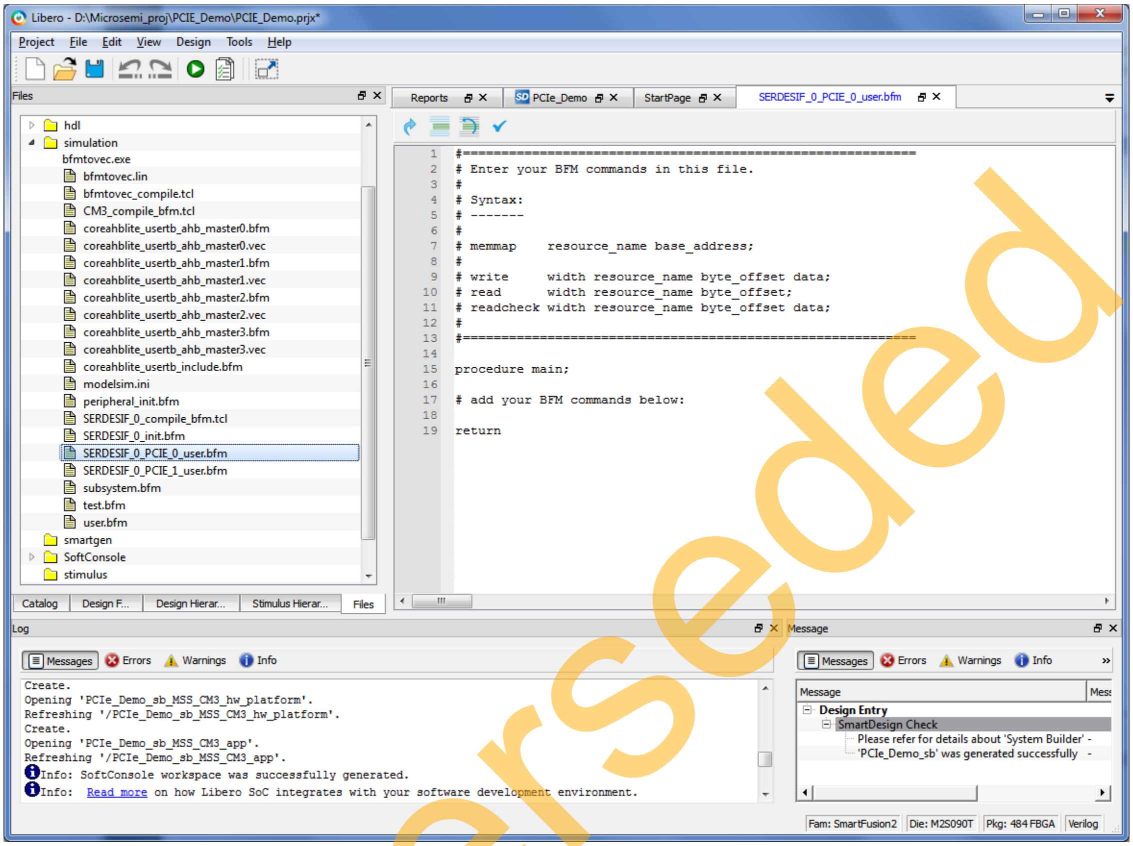
Step 3: Developing the Simulation Stimulus

During the design process, SERDESIF is configured for the BFM simulation model. The BFM simulation model replaces the entire PCIe interface with a simple BFM that can send write transactions and read transactions over the AHB-Lite interface. These transactions are driven by a file and allow easy simulation of the FPGA design connected to a PCIe interface. This simulation methodology has the benefit of focusing on the FPGA design since the SmartFusion2 PCIe interface is a fully hardened and verified interface.

This section describes how to modify the BFM script (user.bfm) file that is generated by SmartDesign. The BFM script file simulates PCIe writing/reading to/from the MSS through the FIC_0.

Superseded

1. Open the `SERDESIF_0_PCIE_0_user.bfm` file. To open the `SERDESIF_0_PCIE_0_user.bfm`, go to the **Files** tab > **Simulation** folder, and double-click the `SERDESIF_0_PCIE_0_user.bfm`. The `SERDESIF_0_PCIE_0_user.bfm` file is displayed, as shown in Figure 50.


Libero - D:\Microsemi_proj\PCIE_Demo\PCIE_Demo.prjx*
Project File Edit View Design Tools Help
Files
hdl
simulation
bfmtovec.exe
bfmtovec.lin
bfmtovec_compile.tcl
CM3_compile_bfm.tcl
coreahblite_usertb_ahb_master0.bfm
coreahblite_usertb_ahb_master0.vec
coreahblite_usertb_ahb_master1.bfm
coreahblite_usertb_ahb_master1.vec
coreahblite_usertb_ahb_master2.bfm
coreahblite_usertb_ahb_master2.vec
coreahblite_usertb_ahb_master3.bfm
coreahblite_usertb_ahb_master3.vec
coreahblite_usertb_include.bfm
modelsim.ini
peripheral_init.bfm
SERDESIF_0_compile_bfm.tcl
SERDESIF_0_init.bfm
SERDESIF_0_PCIE_0_user.bfm
SERDESIF_0_PCIE_1_user.bfm
subsystem.bfm
test.bfm
user.bfm
smartgen
SoftConsole
stimulus
Catalog
Design F...
Design Hierar...
Stimulus Hierar...
Files
Reports
PCIe_Demo
StartPage
SERDESIF_0_PCIE_0_user.bfm
Log
Messages
Errors
Warnings
Info
Message
Fam: SmartFusion2
Die: M2S090T
Pkg: 484 FBGA
Verilog

```
#===========================================================
# Enter your BFM commands in this file.
#
# Syntax:
# -------
#
# memmap    resource_name base_address;
#
# write     width resource_name byte_offset data;
# read      width resource_name byte_offset;
# readcheck width resource_name byte_offset data;
#
#===========================================================

procedure main;

# add your BFM commands below:

return
```

```
Create.
Opening 'PCIe_Demo_sb_MSS_CM3_hw_platform'.
Refreshing '/PCIe_Demo_sb_MSS_CM3_hw_platform'.
Create.
Opening 'PCIe_Demo_sb_MSS_CM3_app'.
Refreshing '/PCIe_Demo_sb_MSS_CM3_app'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.
```

Message
Design Entry
SmartDesign Check
Please refer for details about 'System Builder' -
'PCIe_Demo_sb' was generated successfully -

Figure 50 **• SmartDesign Generated SERDESIF_1_user.bfm File**

2. Modify the `SERDESIF_0_PCIE_0_user.bfm` to add the following bfm commands of writing and reading:

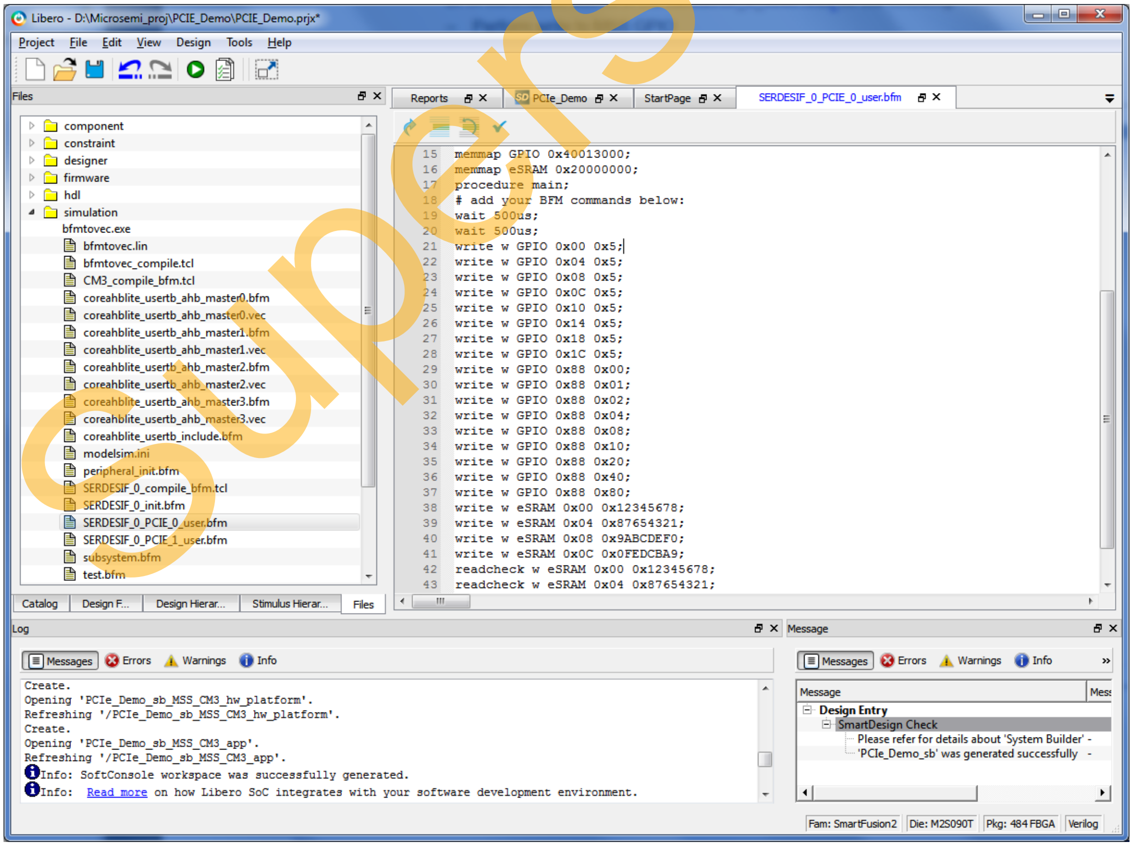
```
memmap GPIO 0x40013000;
memmap eSRAM 0x20000000;
procedure main;

# add your BFM commands below:
wait 500us;
wait 500us;
write w GPIO 0x00  0x5;
write w GPIO 0x04  0x5;
write w GPIO 0x08  0x5;
write w GPIO 0x0C  0x5;
write w GPIO 0x10  0x5;
write w GPIO 0x14  0x5;
write w GPIO 0x18  0x5;
write w GPIO 0x1C  0x5;

write w GPIO 0x88 0x00;
write w GPIO 0x88 0x01;
write w GPIO 0x88 0x02;
```

Superseded

Superseded

```
write w GPIO 0x88 0x04;
write w GPIO 0x88 0x08;
write w GPIO 0x88 0x10;
write w GPIO 0x88 0x20;
write w GPIO 0x88 0x40;
write w GPIO 0x88 0x80;

write w eSRAM 0x00  0x12345678;
write w eSRAM 0x04  0x87654321;
write w eSRAM 0x08  0x9ABCDEF0;
write w eSRAM 0x0C  0x0FEDCBA9;

readcheck w eSRAM 0x00  0x12345678;
readcheck w eSRAM 0x04  0x87654321;
readcheck w eSRAM 0x08  0x9ABCDEF0;
readcheck w eSRAM 0x0C  0x0FEDCBA9;

return
```

3. The modified BFM file appears similar to the file shown in Figure 51.
   BFM commands are added in the `SERDESIF_0_PCIE_0_user.bfm`. Perform the following:
   – Write to MSS GPIO
   – Write to eSRAM
   – Read-check from eSRAM


Libero - D:\Microsemi_proj\PCIE_Demo\PCIE_Demo.prjx*
Project
File
Edit
View
Design
Tools
Help
Files
component
constraint
designer
firmware
hdl
simulation
bfmtovec.exe
bfmtovec.lin
bfmtovec_compile.tcl
CM3_compile_bfm.tcl
coreahblite_usertb_ahb_master0.bfm
coreahblite_usertb_ahb_master0.vec
coreahblite_usertb_ahb_master1.bfm
coreahblite_usertb_ahb_master1.vec
coreahblite_usertb_ahb_master2.bfm
coreahblite_usertb_ahb_master2.vec
coreahblite_usertb_ahb_master3.bfm
coreahblite_usertb_ahb_master3.vec
coreahblite_usertb_include.bfm
modelsim.ini
peripheral_init.bfm
SERDESIF_0_compile_bfm.tcl
SERDESIF_0_init.bfm
SERDESIF_0_PCIE_0_user.bfm
SERDESIF_0_PCIE_1_user.bfm
subsystem.bfm
test.bfm
Catalog
Design F...
Design Hierar...
Stimulus Hierar...
Files
Reports
SD PCIe_Demo
StartPage
SERDESIF_0_PCIE_0_user.bfm
Log
Message
Messages
Errors
Warnings
Info
Messages
Errors
Warnings
Info
Message
Mess
Fam: SmartFusion2
Die: M2S090T
Pkg: 484 FBGA
Verilog

```
memmap GPIO 0x40013000;
memmap eSRAM 0x20000000;
procedure main;
# add your BFM commands below:
wait 500us;
wait 500us;
write w GPIO 0x00 0x5;
write w GPIO 0x04 0x5;
write w GPIO 0x08 0x5;
write w GPIO 0x0C 0x5;
write w GPIO 0x10 0x5;
write w GPIO 0x14 0x5;
write w GPIO 0x18 0x5;
write w GPIO 0x1C 0x5;
write w GPIO 0x88 0x00;
write w GPIO 0x88 0x01;
write w GPIO 0x88 0x02;
write w GPIO 0x88 0x04;
write w GPIO 0x88 0x08;
write w GPIO 0x88 0x10;
write w GPIO 0x88 0x20;
write w GPIO 0x88 0x40;
write w GPIO 0x88 0x80;
write w eSRAM 0x00 0x12345678;
write w eSRAM 0x04 0x87654321;
write w eSRAM 0x08 0x9ABCDEF0;
write w eSRAM 0x0C 0x0FEDCBA9;
readcheck w eSRAM 0x00 0x12345678;
readcheck w eSRAM 0x04 0x87654321;
```

```
Create.
Opening 'PCIe_Demo_sb_MSS_CM3_hw_platform'.
Refreshing '/PCIe_Demo_sb_MSS_CM3_hw_platform'.
Create.
Opening 'PCIe_Demo_sb_MSS_CM3_app'.
Refreshing '/PCIe_Demo_sb_MSS_CM3_app'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.
```

Design Entry
SmartDesign Check
Please refer for details about 'System Builder' -
'PCIe_Demo_sb' was generated successfully -

Figure 51 • **Modified SERDES User BFM**

Step 4: Simulating the Design

The design supports the BFM_PCIe simulation level to communicate with the High Speed Serial Interface block through the master AXI bus interface. Although no serial communication actually goes through the High Speed Serial Interface block, this scenario allows validating the fabric interface connections. The SERDESIF_1_user.bfm file under the <Libero project>/simulation folder contains the BFM commands to verify the read/write access to MSS GPIOs and eSRAM.

The following steps describe how to use the SmartDesign testbench and BFM script file to simulate the design.

1. To generate the HDL testbench file follow the below instructions,

   a. From the **File** menu, choose **New** > **HDL Testbench** as shown in Figure 52.

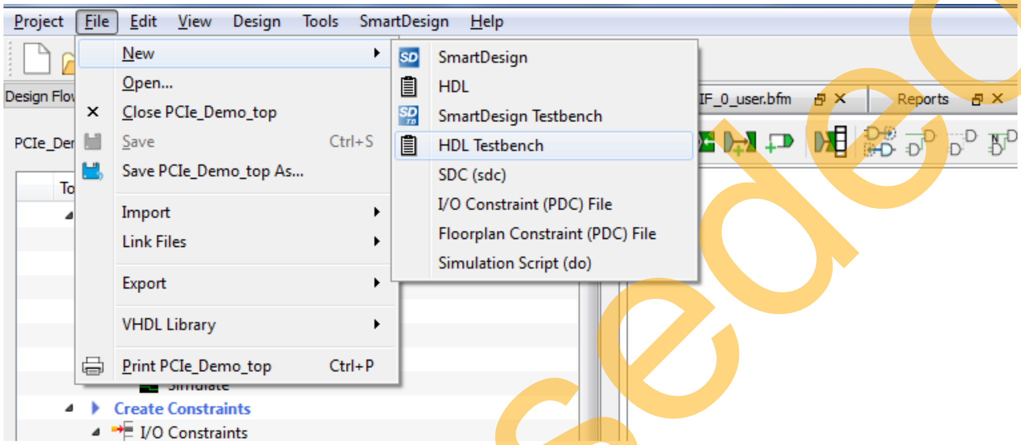
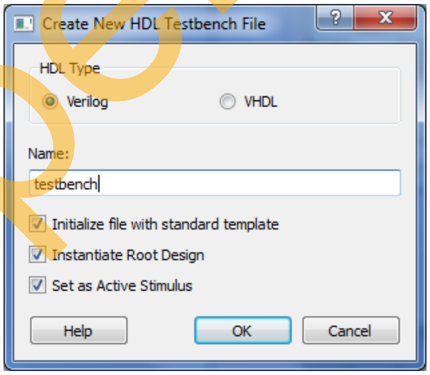


Figure 52 **• HDL Testbench**

Create New HDL Testbench File dialog box is displayed as shown in Figure 53.


Create New HDL Testbench File

HDL Type

● Verilog ○ VHDL

Name: testbench

☑ Initialize file with standard template

☑ Instantiate Root Design

☑ Set as Active Stimulus

Figure 53 **• Create New HDL Testbench File**

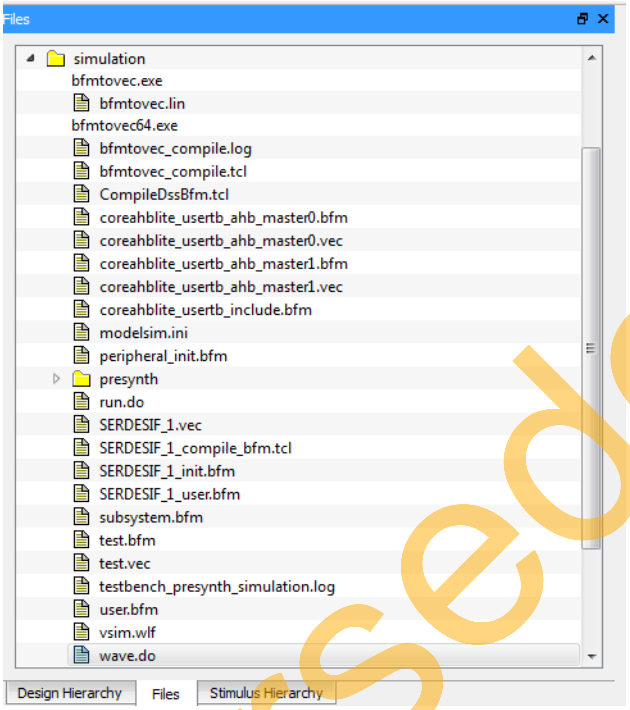
   b. Select **Verilog** or **VHDL** under **HDL Type**.

   c. Enter testbench as a name of the new hdl testbench file and click **OK**.

2. Add the wave do file to the PCIe demo design simulation folder by clicking **File > Import > Others**.

Superseded

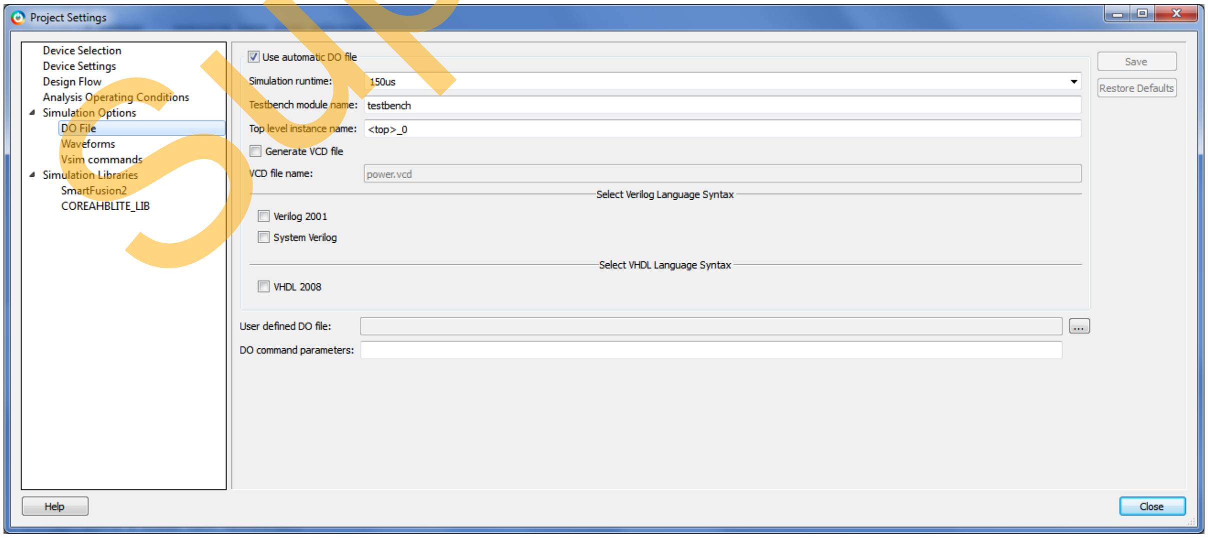
3. Browse to the wave.do file location in the design files folder: *M2S90_PCIE_Control_Demo_DF/Source Files*. Figure 54 shows the wave.do file under simulation folder in the **Files** window.



simulation
bfmtovec.exe
bfmtovec.lin
bfmtovec64.exe
bfmtovec_compile.log
bfmtovec_compile.tcl
CompileDssBfm.tcl
coreahblite_usertb_ahb_master0.bfm
coreahblite_usertb_ahb_master0.vec
coreahblite_usertb_ahb_master1.bfm
coreahblite_usertb_ahb_master1.vec
coreahblite_usertb_include.bfm
modelsim.ini
peripheral_init.bfm
presynth
run.do
SERDESIF_1.vec
SERDESIF_1_compile_bfm.tcl
SERDESIF_1_init.bfm
SERDESIF_1_user.bfm
subsystem.bfm
test.bfm
test.vec
testbench_presynth_simulation.log
user.bfm
vsim.wlf
wave.do

***Figure 54* • Wave.do File under Simulation Folder**

4. Open the Libero SoC project settings (**Project > Project Settings**).
5. Select **Do File** under **Simulation Options** in the Project Settings window. Change the **Simulation runtime** to **150us**, as shown in Figure 55.
6. Click **Save**.

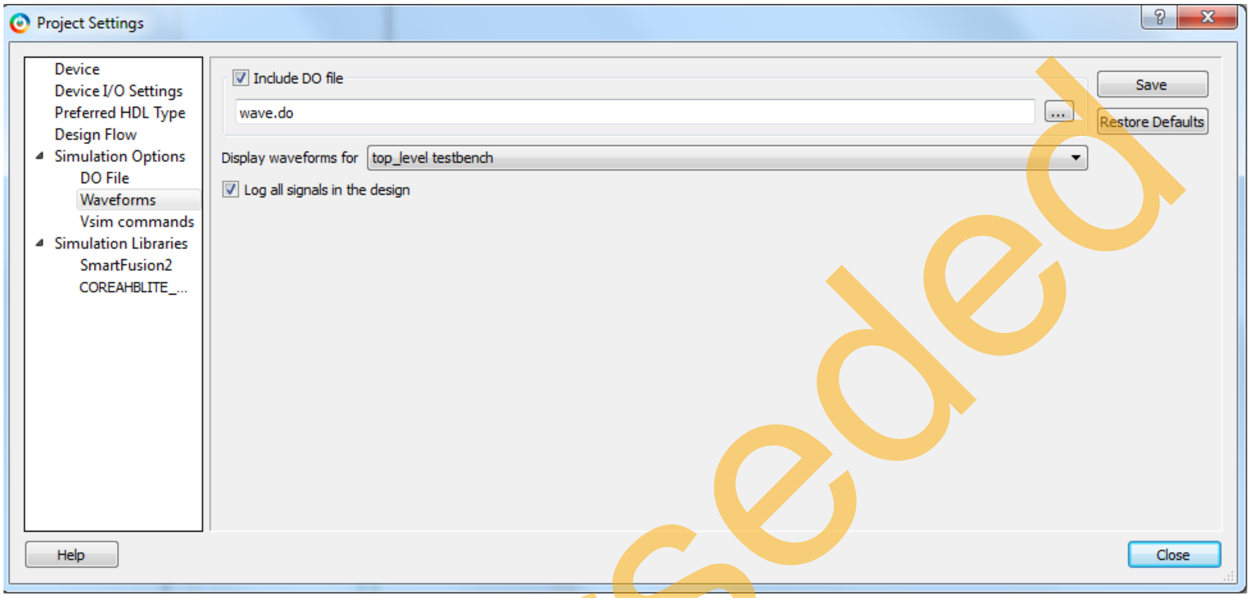


☑ Use automatic DO file
Simulation runtime: 150us
Testbench module name: testbench
Top level instance name: <top>_0
□ Generate VCD file
VCD file name: power.vcd
□ Verilog 2001
□ System Verilog
□ VHDL 2008
User defined DO file:
DO command parameters:

***Figure 55* • Project Setting – Do File Simulation Runtime Setting**

Superseded

7. Select **Waveforms** under **Simulation Options** as shown in Figure 56:
   - Select **Include Do file**.
   - Select **Log all signals in the design**.
   - Click **Close** to close the Project settings dialog box.
   - Select **Save** when prompted to save the changes.



☑ Include DO file

wave.do

Display waveforms for: top_level testbench

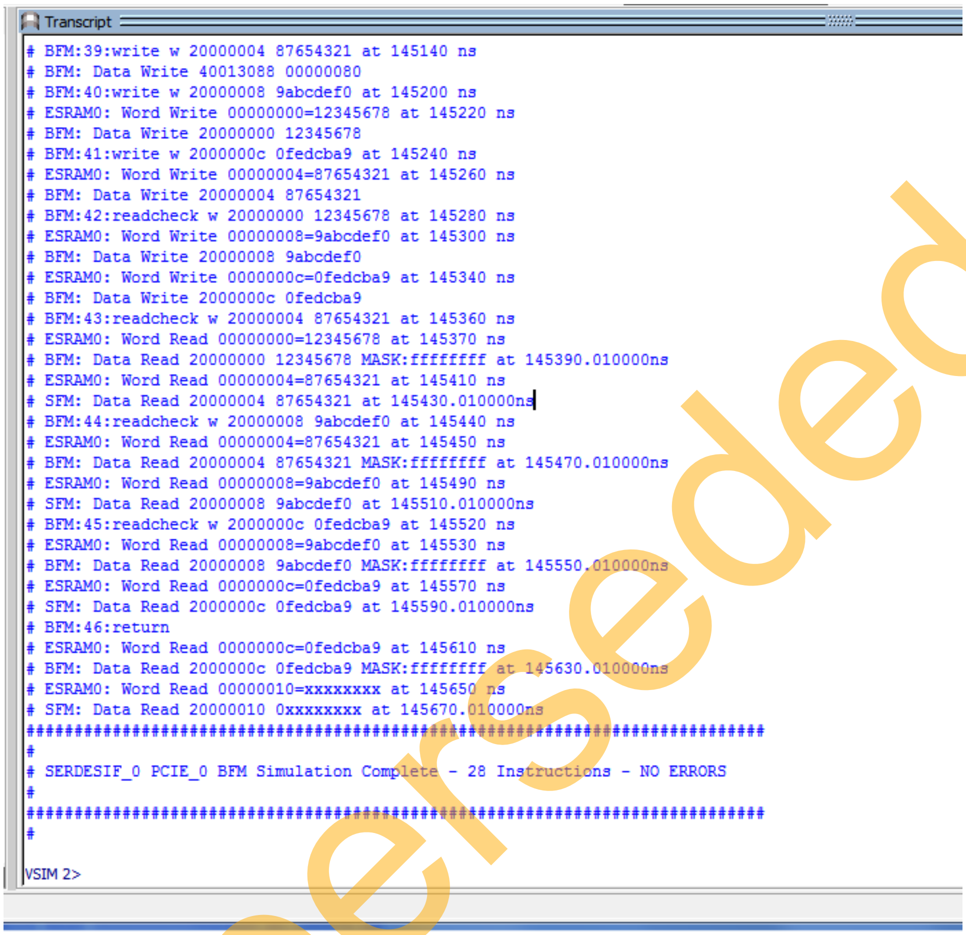
☑ Log all signals in the design

Figure 56 **• Project Setting – Waveform**

To run the simulation, double-click **Simulate** under **Verify Pre-Synthesized Design** in the **Design Flow** window.

Superseded

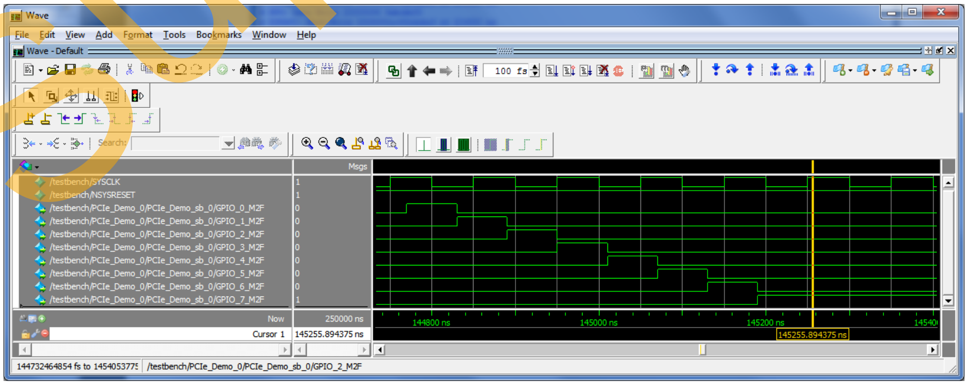
ModelSim runs the design for approximately 150us. The ModelSim transcript window displays the BFM commands and the BFM simulation completed with no errors, as shown in Figure 57.



```
# BFM:39:write w 20000004 87654321 at 145140 ns
# BFM: Data Write 40013088 00000080
# BFM:40:write w 20000008 9abcdef0 at 145200 ns
# ESRAM0: Word Write 00000000=12345678 at 145220 ns
# BFM: Data Write 20000000 12345678
# BFM:41:write w 2000000c 0fedcba9 at 145240 ns
# ESRAM0: Word Write 00000004=87654321 at 145260 ns
# BFM: Data Write 20000004 87654321
# BFM:42:readcheck w 20000000 12345678 at 145280 ns
# ESRAM0: Word Write 00000008=9abcdef0 at 145300 ns
# BFM: Data Write 20000008 9abcdef0
# ESRAM0: Word Write 0000000c=0fedcba9 at 145340 ns
# BFM: Data Write 2000000c 0fedcba9
# BFM:43:readcheck w 20000004 87654321 at 145360 ns
# ESRAM0: Word Read 00000000=12345678 at 145370 ns
# BFM: Data Read 20000000 12345678 MASK:ffffffff at 145390.010000ns
# ESRAM0: Word Read 00000004=87654321 at 145410 ns
# SFM: Data Read 20000004 87654321 at 145430.010000ns
# BFM:44:readcheck w 20000008 9abcdef0 at 145440 ns
# ESRAM0: Word Read 00000004=87654321 at 145450 ns
# BFM: Data Read 20000004 87654321 MASK:ffffffff at 145470.010000ns
# ESRAM0: Word Read 00000008=9abcdef0 at 145490 ns
# SFM: Data Read 20000008 9abcdef0 at 145510.010000ns
# BFM:45:readcheck w 2000000c 0fedcba9 at 145520 ns
# ESRAM0: Word Read 00000008=9abcdef0 at 145530 ns
# BFM: Data Read 20000008 9abcdef0 MASK:ffffffff at 145550.010000ns
# ESRAM0: Word Read 0000000c=0fedcba9 at 145570 ns
# SFM: Data Read 2000000c 0fedcba9 at 145590.010000ns
# BFM:46:return
# ESRAM0: Word Read 0000000c=0fedcba9 at 145610 ns
# BFM: Data Read 2000000c 0fedcba9 MASK:ffffffff at 145630.010000ns
# ESRAM0: Word Read 00000010=xxxxxxxx at 145650 ns
# SFM: Data Read 20000010 0xxxxxxxx at 145670.010000ns
##############################################################################
#
# SERDESIF_0 PCIE_0 BFM Simulation Complete - 28 Instructions - NO ERRORS
#
##############################################################################
#

VSIM 2>
```

Figure 57 **• SERDES BFM Simulation**

Figure 58 shows the waveform window with MSS GPIO output signals.



```
/testbench/SYSCLK                                   1
/testbench/NSYSRESET                                1
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_0_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_1_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_2_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_3_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_4_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_5_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_6_M2F    0
/testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_7_M2F    1
Now        250000 ns
Cursor 1   145255.894375 ns
144732464854 fs to 145405377!   /testbench/PCIe_Demo_0/PCIe_Demo_sb_0/GPIO_2_M2F
```

Figure 58 **• Simulation Result with MSS GPIO Signals**

Superseded

Step 5: Generating the Program File

The following steps describe how to generate the program file.

1. Double-click **I/O Constraints** in the **Design Flow** window as shown in Figure 59. The **I/O Editor** window is displayed after completing Synthesize and Compile.

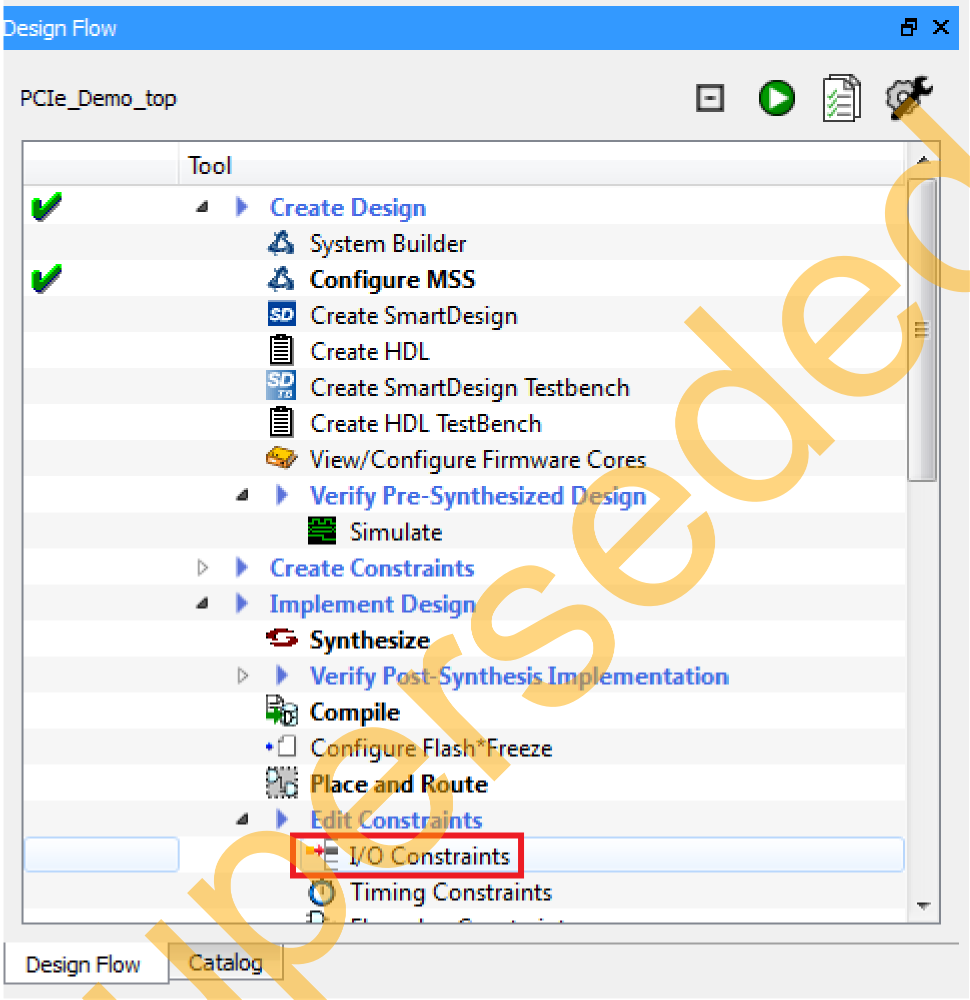


Figure 59 • **I/O Constraints**

Superseded

2. The **I/O Editor** is displayed. Make the pin assignments shown in Table 5. After the pins have been assigned, the I/O Editor is displayed as shown in Figure 60 on page 54.

Table 5 • **Port to Pin Mapping**

Port Name	Pin Number
GPIO_0_M2F	E1
GPIO_1_M2F	F4
GPIO_2_M2F	F3
GPIO_3_M2F	G7
GPIO_4_M2F	H7
GPIO_5_M2F	J6

Table 5 • **Port to Pin Mapping (continued)**

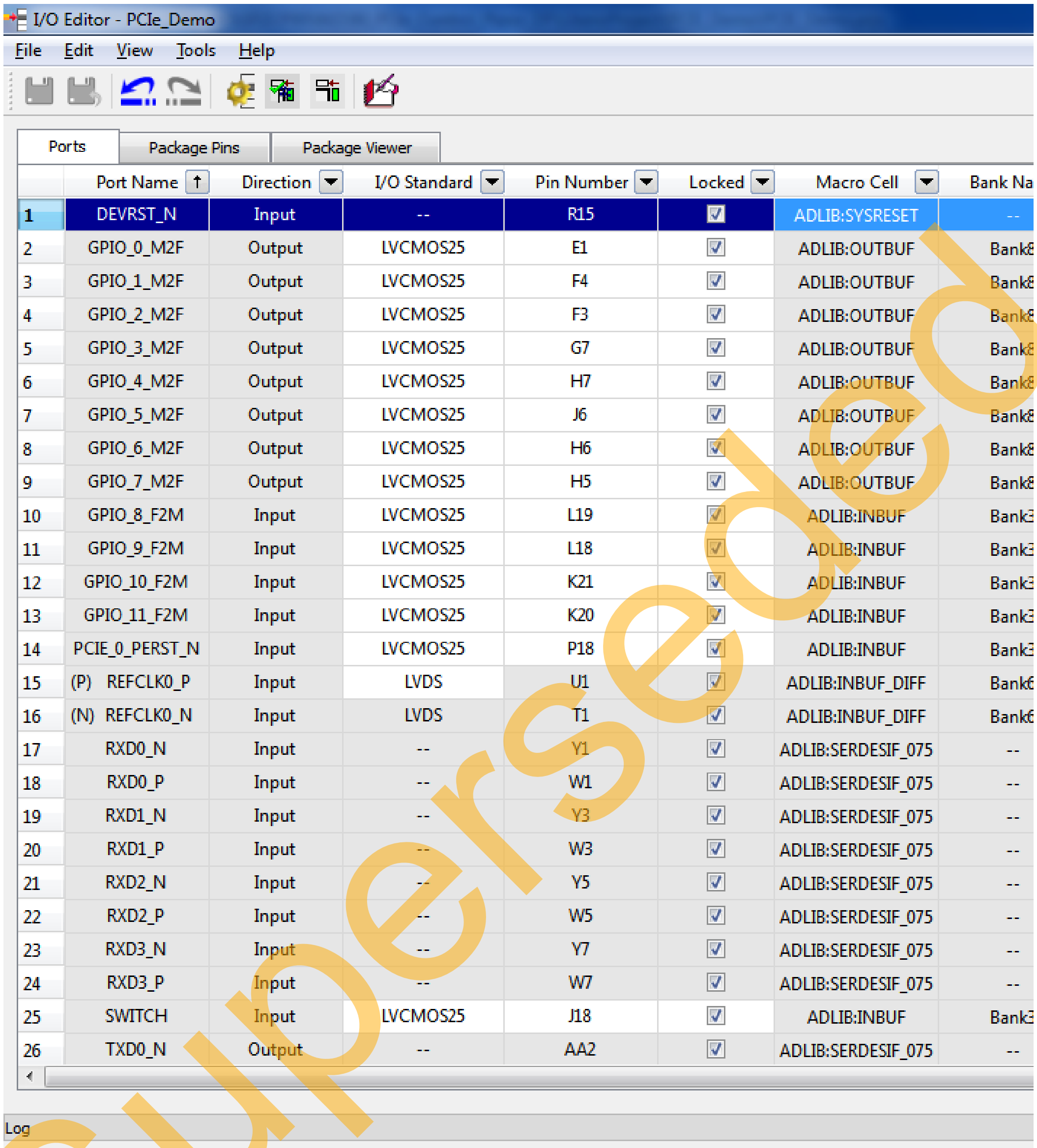
Port Name	Pin Number
GPIO_6_M2F	H6
GPIO_7_M2F	H5
GPIO_8_M2F	L19
GPIO_9_M2F	L18
GPIO_10_M2F	K21
GPIO_11_M2F	K20
SWITCH	J18
PCIE0_PERST_N	P18

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

– CLK0_PAD to 50 MHz Clock Oscillator
– GPIO_0 to GPIO_8 for LEDs
– GPIO_8 to GPIO_11 for DIP switches
– SWITCH for SW4
– PCIE_0_PERST_N to PERST of PCIe Edge connector

Superseded

Superseded



	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Na
1	DEVRST_N	Input	--	R15	☑	ADLIB:SYSRESET	--
2	GPIO_0_M2F	Output	LVCMOS25	E1	☑	ADLIB:OUTBUF	Bank8
3	GPIO_1_M2F	Output	LVCMOS25	F4	☑	ADLIB:OUTBUF	Bank8
4	GPIO_2_M2F	Output	LVCMOS25	F3	☑	ADLIB:OUTBUF	Bank8
5	GPIO_3_M2F	Output	LVCMOS25	G7	☑	ADLIB:OUTBUF	Bank8
6	GPIO_4_M2F	Output	LVCMOS25	H7	☑	ADLIB:OUTBUF	Bank8
7	GPIO_5_M2F	Output	LVCMOS25	J6	☑	ADLIB:OUTBUF	Bank8
8	GPIO_6_M2F	Output	LVCMOS25	H6	☑	ADLIB:OUTBUF	Bank8
9	GPIO_7_M2F	Output	LVCMOS25	H5	☑	ADLIB:OUTBUF	Bank8
10	GPIO_8_F2M	Input	LVCMOS25	L19	☑	ADLIB:INBUF	Bank3
11	GPIO_9_F2M	Input	LVCMOS25	L18	☑	ADLIB:INBUF	Bank3
12	GPIO_10_F2M	Input	LVCMOS25	K21	☑	ADLIB:INBUF	Bank3
13	GPIO_11_F2M	Input	LVCMOS25	K20	☑	ADLIB:INBUF	Bank3
14	PCIE_0_PERST_N	Input	LVCMOS25	P18	☑	ADLIB:INBUF	Bank3
15	(P) REFCLK0_P	Input	LVDS	U1	☑	ADLIB:INBUF_DIFF	Bank6
16	(N) REFCLK0_N	Input	LVDS	T1	☑	ADLIB:INBUF_DIFF	Bank6
17	RXD0_N	Input	--	Y1	☑	ADLIB:SERDESIF_075	--
18	RXD0_P	Input	--	W1	☑	ADLIB:SERDESIF_075	--
19	RXD1_N	Input	--	Y3	☑	ADLIB:SERDESIF_075	--
20	RXD1_P	Input	--	W3	☑	ADLIB:SERDESIF_075	--
21	RXD2_N	Input	--	Y5	☑	ADLIB:SERDESIF_075	--
22	RXD2_P	Input	--	W5	☑	ADLIB:SERDESIF_075	--
23	RXD3_N	Input	--	Y7	☑	ADLIB:SERDESIF_075	--
24	RXD3_P	Input	--	W7	☑	ADLIB:SERDESIF_075	--
25	SWITCH	Input	LVCMOS25	J18	☑	ADLIB:INBUF	Bank3
26	TXD0_N	Output	--	AA2	☑	ADLIB:SERDESIF_075	--

***Figure 60* • I/O Editor**

3. After updating I/O editor, click **Commit and Check**.
4. Close the I/O editor.
5. Click **Generate Bitstream** as shown in Figure 61 to complete place and route, verify timing, and generate the programming file.

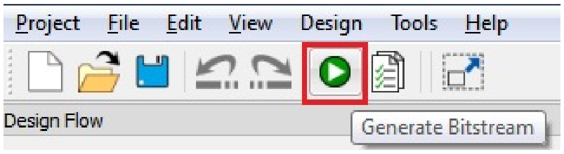


***Figure 61* • Generate Bitstream**

Superseded

Running the Demo

Demo Setup

Following are the steps to setup the demo for SmartFusion2 Evaluation Kit Board:

1. Connect the FlashPro4 programmer to the J5 connector of the SmartFusion2 Evaluation Kit Board.
2. Connect the jumpers on the SmartFusion2 Evaluation Kit Board as shown in Table 4.

CAUTION: The power supply switch SW7 on the board should be in OFF position, while making the jumper connections.

Table 6 • **SmartFusion2 FPGA Evaluation Kit Jumper Settings**

Jumper	Pin (from)	Pin (to)	Comments
J22, J23, J24, J8, J3	1	2	These are the default jumper settings of the SmartFusion2 Evaluation Kit Board. Ensure these jumpers are set accordingly.

3. Connect the power supply to the **J6** connector.
4. Switch the power supply switch **SW7** to **ON** position.
5. To program the SmarFusion2 device double-click **Run PROGRAM Action** in the **Design Flow** tab as shown in Figure 62.

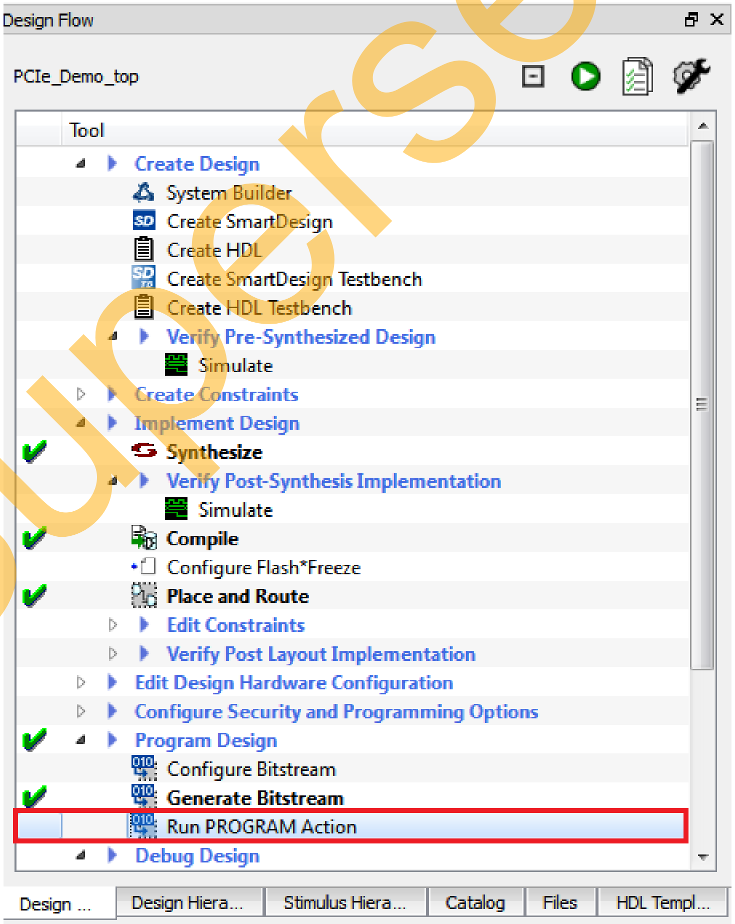


Figure 62 • **Run PROGRAM Action**

6. After Successful programming, power OFF the SmartFusion2 Evaluation Kit and shut down the Host PC.
7. Following are the steps to connect the **CON1-PCIe Edge Connector** either to Host PC or laptop:

   a. Connect the CON1-PCIe Edge Connector to Host PC PCIe Gen 2 slot or Gen 1 slot as applicable. If the Host PC does not support the Gen 2 compliant slot, the design switches to the Gen 1 slot.

   b. Connect the CON1-PCIe Edge Connector to the laptop PCIe slot using the express card adapter. If you are using a laptop, the express card adapters typically support only Gen 1 and the design works on Gen 1 slot.

CAUTION: Host PC or laptop should be powered OFF while inserting the PCIe Edge Connector. If the system is not powered OFF, the PCIe device detection and selection of Gen 1 or Gen 2 does not occur properly. It is recommended that the Host PC or laptop should be powered OFF during the PCIe card insertion.

The board setup is shown in Figure 63.

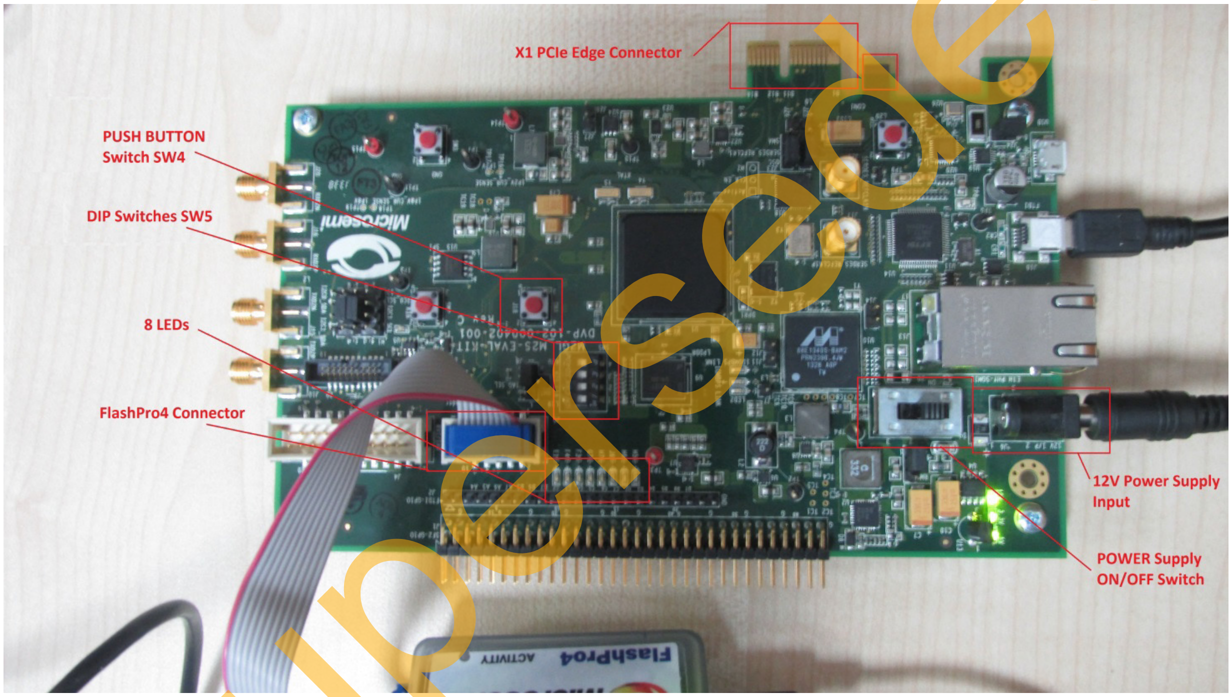


Figure 63 • **SmartFusion2 Evaluation Kit Setup**

8. Switch ON the power supply switch, **SW7**.

Running the Demo Design

This demo can run on both windows and Red Hat Linux operating system.

To run the demo on Windows operating system GUI, Jungo drivers are provided. Refer to "Running the Demo Design on Windows" on page 57.

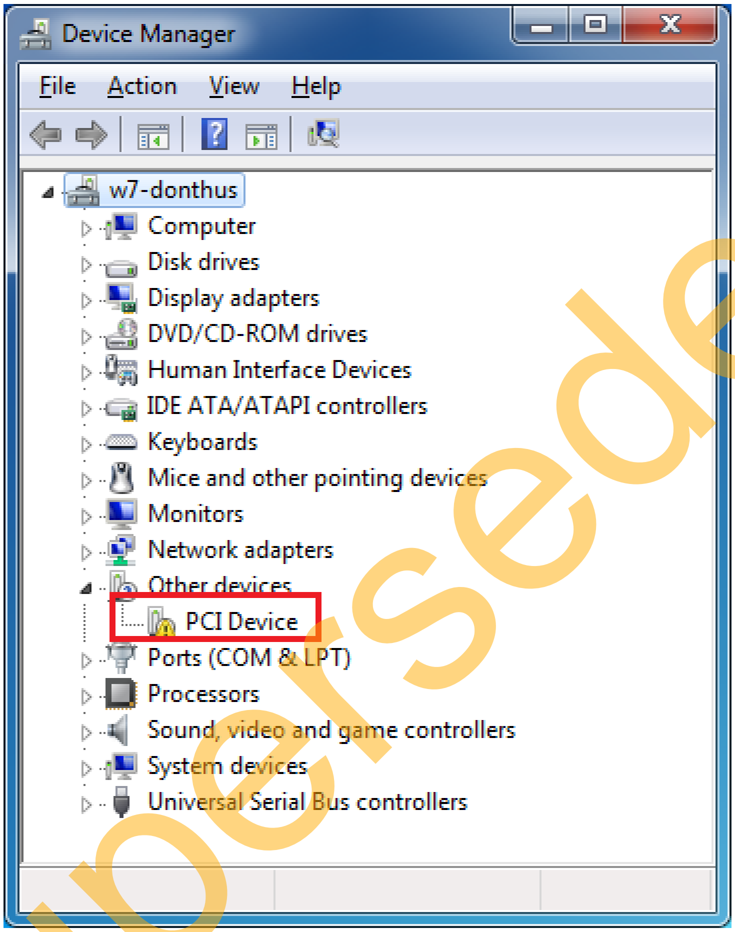
To run the demo on Linux operating system native Red Hat Linux drivers and command line scripts are provided. Refer to "Running the Demo Design on Linux" on page 70.

Superseded

Running the Demo Design on Windows

The following steps describe how to run the Demo Design on Windows:

1. Power on the Host PC and check the Host PC Device Manager for PCIe Device. It is similar to Figure 64. If the device is not detected, power cycle the SmartFusion2 Evaluation Kit and click **Scan for hardware changes** in **Device Manager** window.

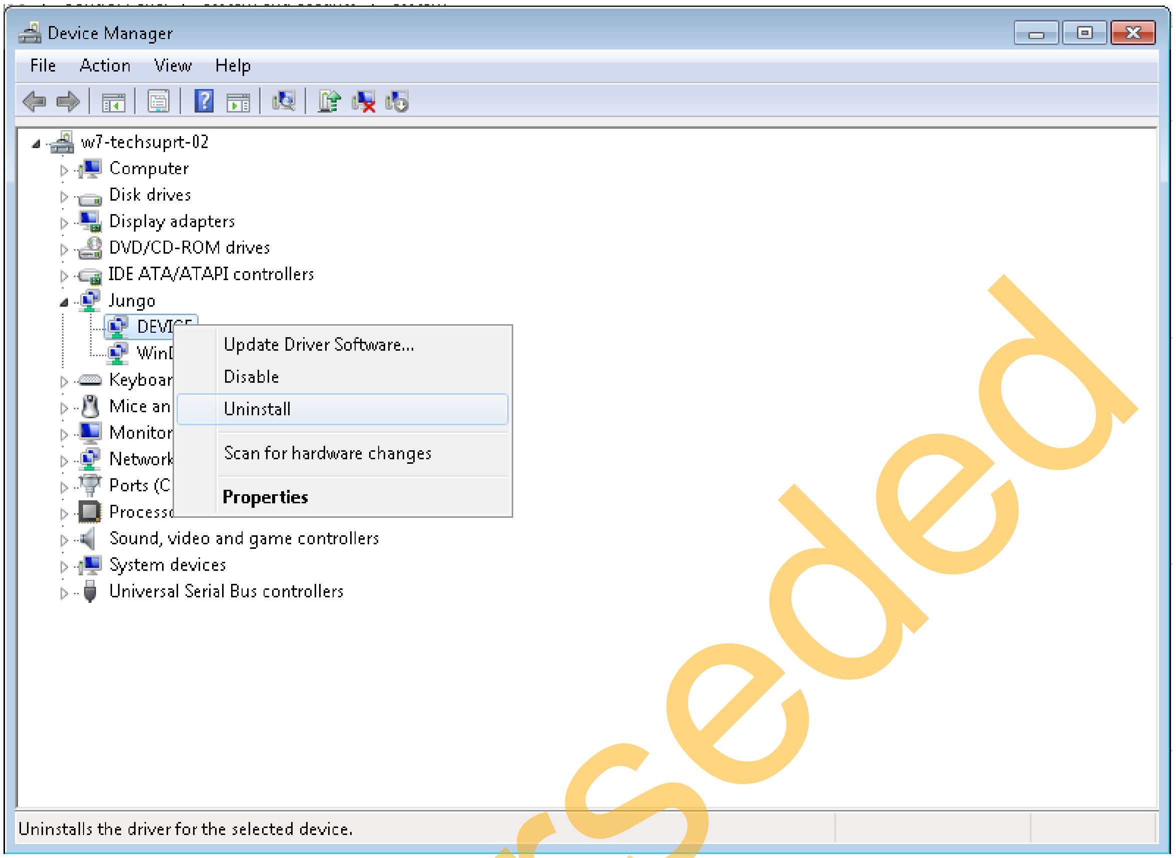


w7-donthus
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
  - PCI Device
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

***Figure 64* • Device Manager - PCIe Device Detection**

2. If the Host PC has any other installed drivers (previous versions of Jungo drivers) for the SmartFusion2PCIe device, uninstall them. To uninstall previous versions of Jungo drivers follow steps 12 and 13.
3. To uninstall previous Jungo drivers go to **Device Manager,** right-click on **DEVICE,** and click **Uninstall** as shown in Figure 65 on page 58.

Superseded

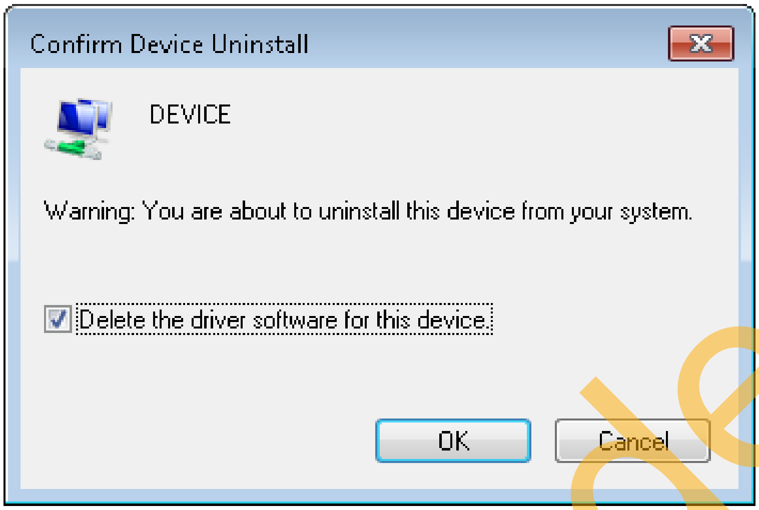


Uninstalls the driver for the selected device.

Figure 65 **• Device Manager Window**

Superseded

4. **Confirm Device Uninstall** window is displayed as shown in Figure 66 Confirm Device Uninstall Select **Delete the driver software for this device**. After uninstalling previous Jungo drivers, ensure that the PCI Device is detected in the **Device Manager** window as shown in Figure 66.



Confirm Device Uninstall

DEVICE

Warning: You are about to uninstall this device from your system.

☑ Delete the driver software for this device.

Figure 66 • **Confirm Device Uninstall Dialogue Box**

Note: If the device is still not detected, check if the BIOS version in Host PC is latest, and if PCI is enabled in the Host PC BIOS.

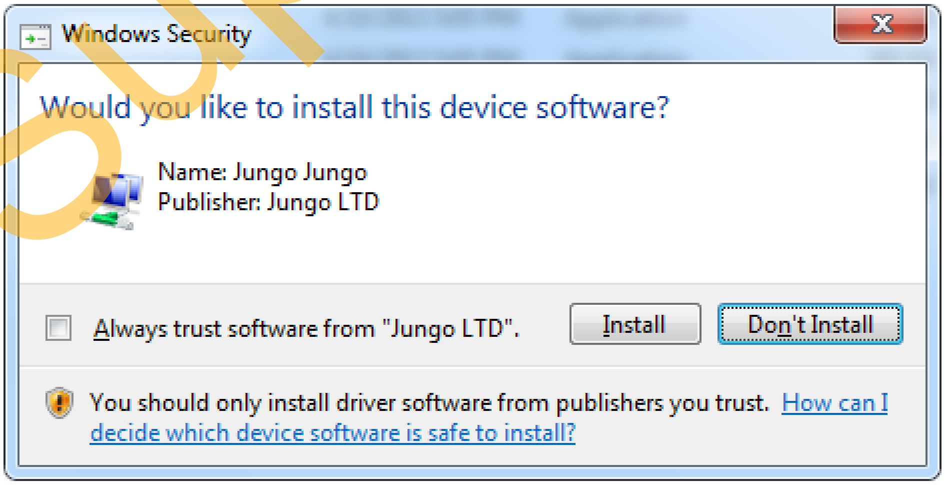
Superseded

Drivers Installation

The PCIe Demo uses a driver framework provided by Jungo WinDriverPro. To install the PCIe drivers on Host PC for SmartFusion2 Evaluation Kit, use the following steps:

1. Extract the **PCIe_Demo.rar** to C:\ drive. The PCIe_Demo.rar is located in the provided design files:
   – *M2S90_PCIE_Control_DEMO_DF\Windows_64bit\Drivers\PCIe_Demo.rar*
2. Run the batch file **C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat**

   Note:Installing these drivers require Host PC Administration rights.
3. In the WIndows Security dialog box, click **Install** as shown in Figure 67.



Windows Security

Would you like to install this device software?

Name: Jungo Jungo
Publisher: Jungo LTD

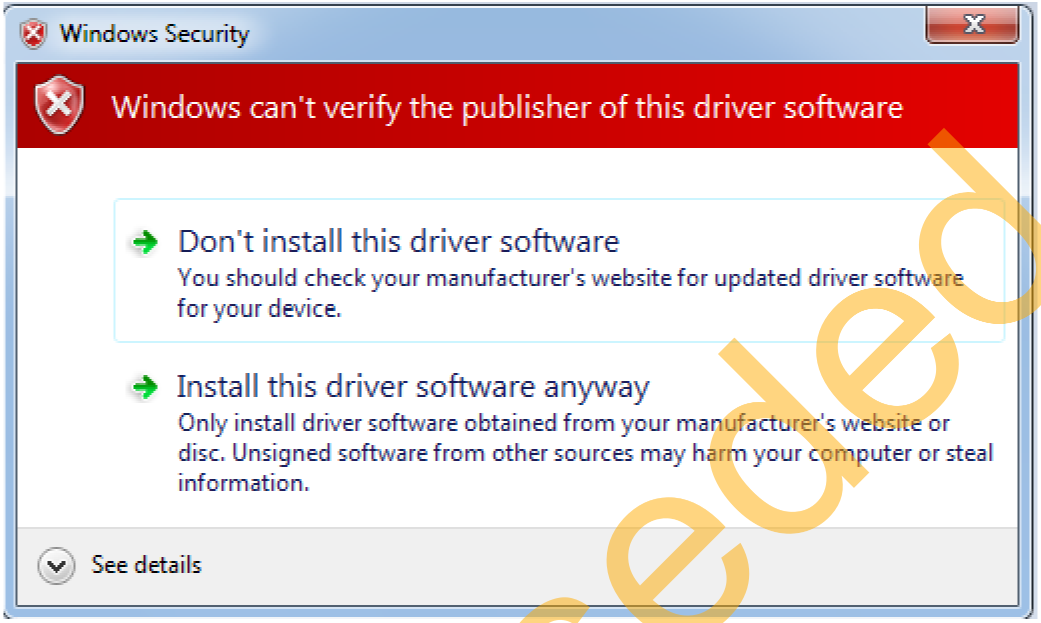
□ Always trust software from "Jungo LTD".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Figure 67 • **Jungo Driver Installation**

Note: If the installation is not in progress, right-click on the command prompt and select Run as administrator. Run the batch file C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat from command prompt.

4. Click **Install this driver software anyway** as shown in Figure 68.



Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Figure 68 **• Windows Security**

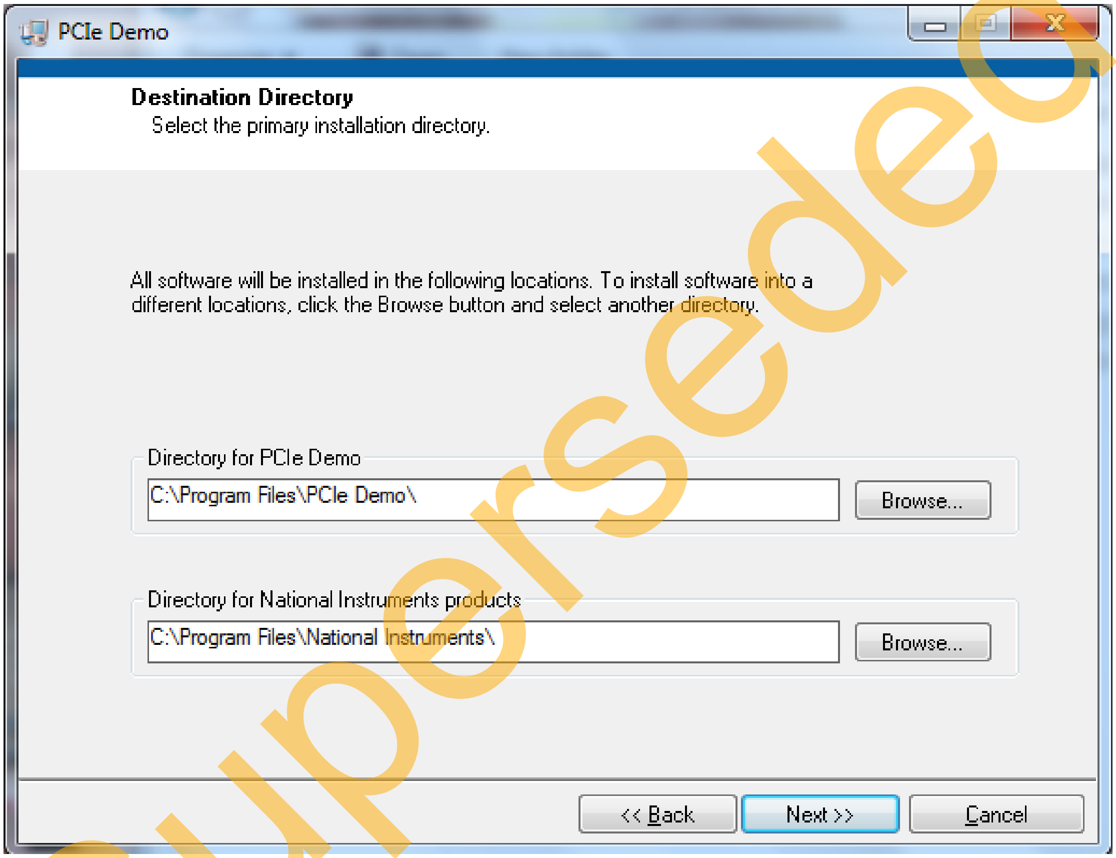
Superseded

PCIe Demo GUI

SmartFusion2 PCIe Demo GUI is a simple graphic user interface that runs on the Host PC to communicate with the SmartFusion2 PCIe EP device. The GUI provides the PCIe link status, driver information, and demo controls. The GUI invokes the PCIe driver installed on the Host PC and provides commands to the driver according to the user selection.

The following steps are used to install the GUI:

1. Download the PCIe_Demo_GUI Installer from the following link.
   http://soc.microsemi.com/download/rsc/?f=PCIe_Demo_GUI_Installer
2. Double-click **setup.exe** in the provided GUI installation (PCIe_Demo_GUI_Installer\setup.exe). Apply default options as shown in Figure 69.



Destination Directory
Select the primary installation directory.

All software will be installed in the following locations. To install software into a different locations, click the Browse button and select another directory.

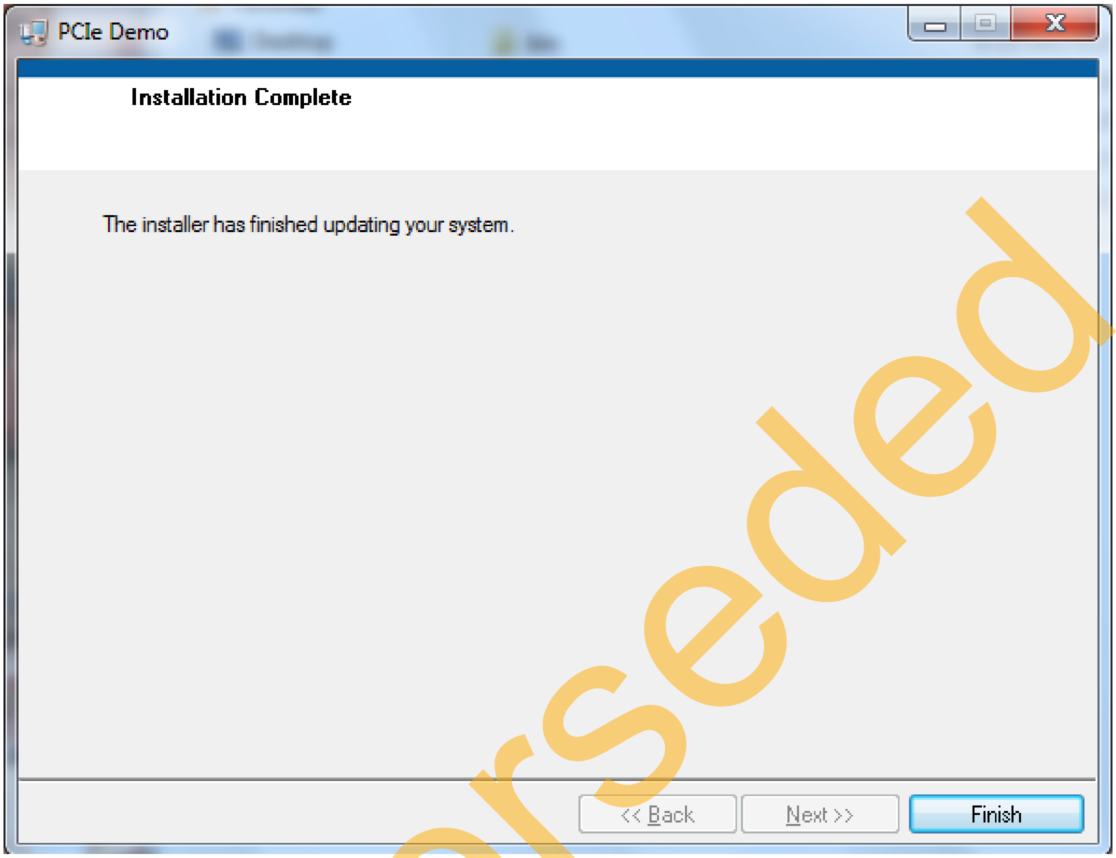
Directory for PCIe Demo: C:\Program Files\PCIe Demo\

Directory for National Instruments products: C:\Program Files\National Instruments\

Figure 69 • **GUI Installation**

Superseded

3. Click **Next** to complete the installation. After successful installation, the following window is displayed.



The installer has finished updating your system.

***Figure 70* • Successful GUI Installation**

4. Restart the host PC.

Superseded

Running the Design

The following steps describe how to run the design.

1. Check the Host PC **Device Manager** for the drivers. If the device is not detected, power cycle the SmartFusion2 Evaluation Kit and click **Scan for hardware changes** in Device Manager. Ensure that the board is switched on.

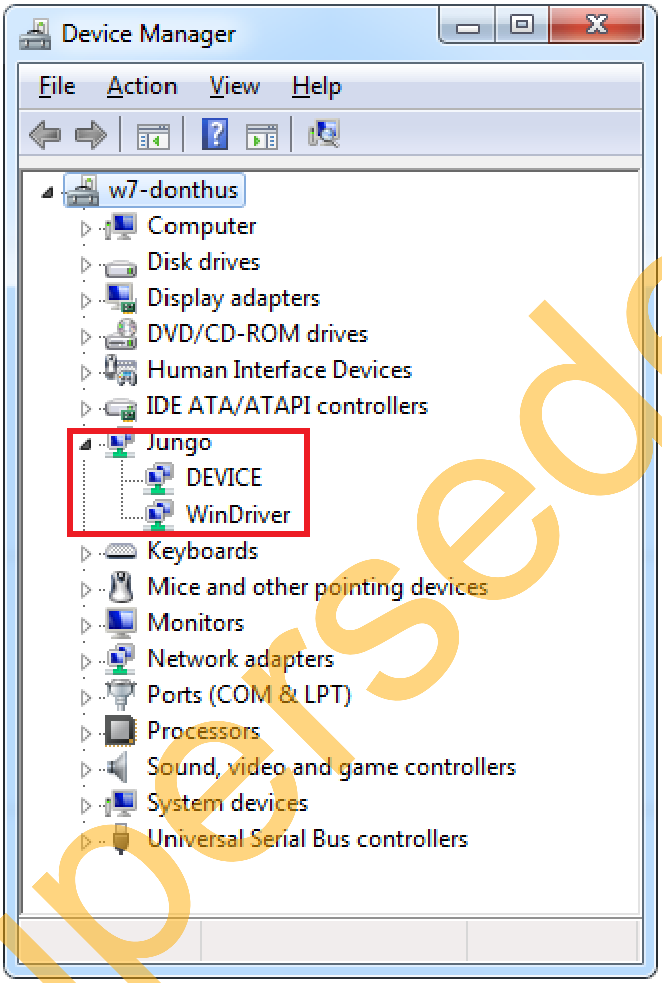
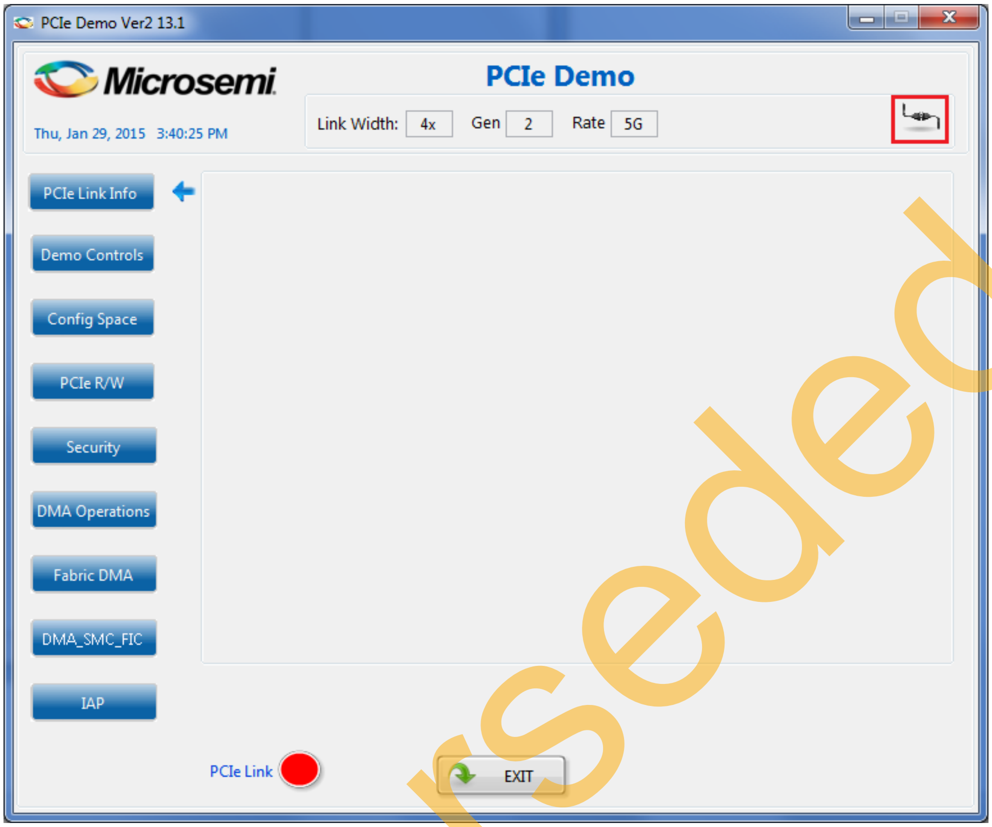


Figure 71 • **Device Manager - PCIe Device Detection**

Note: If a warning symbol is displayed on the **DEVICE** or **WinDriver** icons in the **Device Manager**, uninstall them and start from step1 of "Drivers Installation" on page 59.

Superseded

2. Invoke the GUI from **ALL Programs > PCIeDemo > PCIe Demo**. The GUI is displayed as shown in Figure 72.



PCIe Demo Ver2 13.1

PCIe Demo

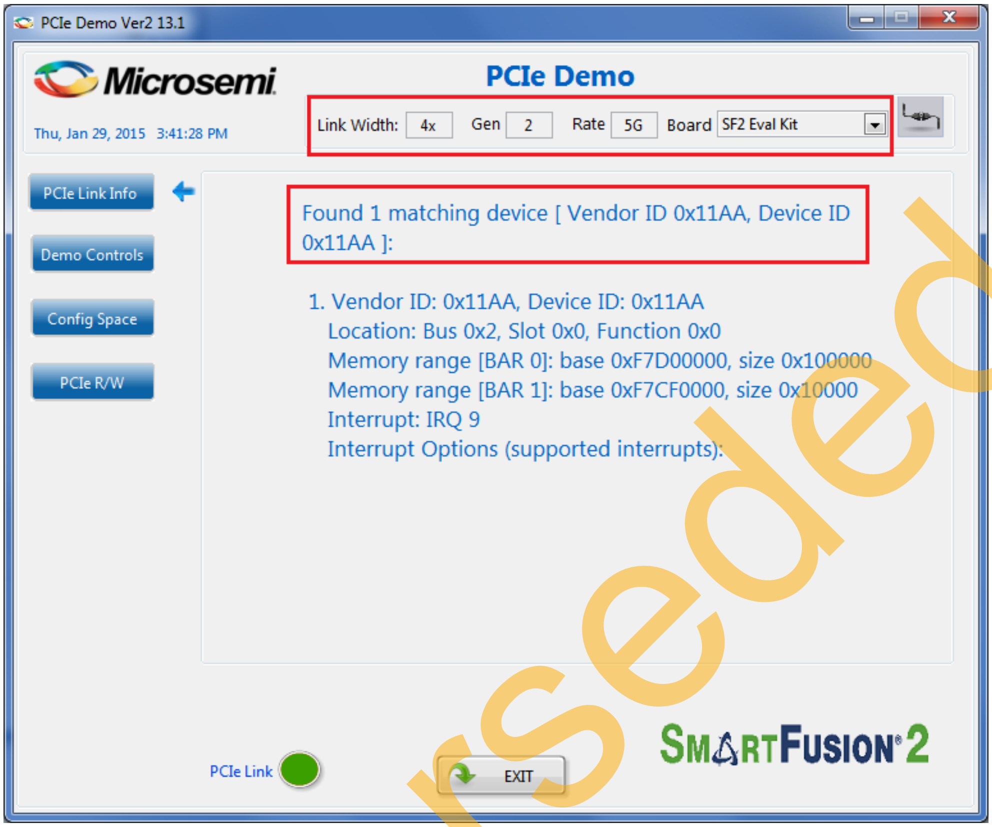
Thu, Jan 29, 2015 3:40:25 PM

Link Width: 4x Gen 2 Rate 5G

Figure 72 **• PCIe Demo GUI**

Superseded

3. Click **Connect** button at the top-right corner of the GUI. The messages are displayed on the GUI as shown in Figure 73.



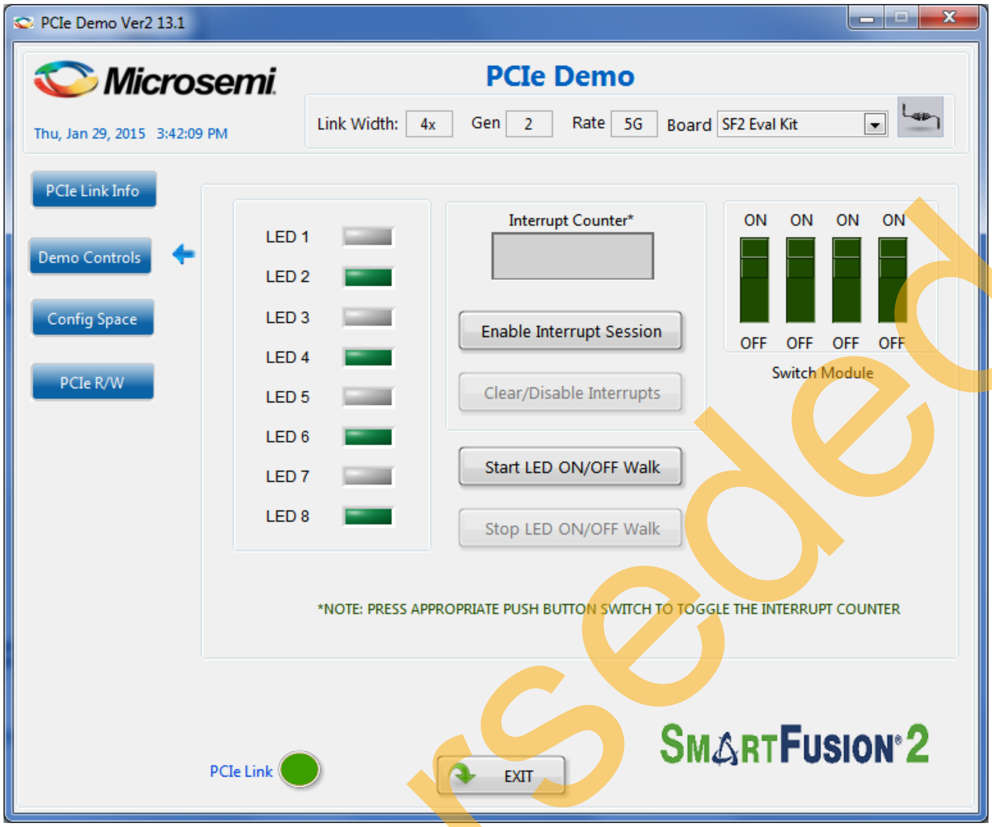
Link Width: 4x Gen 2 Rate 5G Board SF2 Eval Kit

```
Found 1 matching device [ Vendor ID 0x11AA, Device ID
0x11AA ]:

1. Vendor ID: 0x11AA, Device ID: 0x11AA
   Location: Bus 0x2, Slot 0x0, Function 0x0
   Memory range [BAR 0]: base 0xF7D00000, size 0x100000
   Memory range [BAR 1]: base 0xF7CF0000, size 0x10000
   Interrupt: IRQ 9
   Interrupt Options (supported interrupts):
```

Figure 73 **• Version Information**

Superseded

4. Click **Demo Controls** in the GUI displays the LEDs options and DIP switch positions as shown in Figure 74.



PCIe Demo Ver2 13.1

PCIe Demo

Thu, Jan 29, 2015 3:42:09 PM

Link Width: 4x Gen 2 Rate 5G Board SF2 Eval Kit

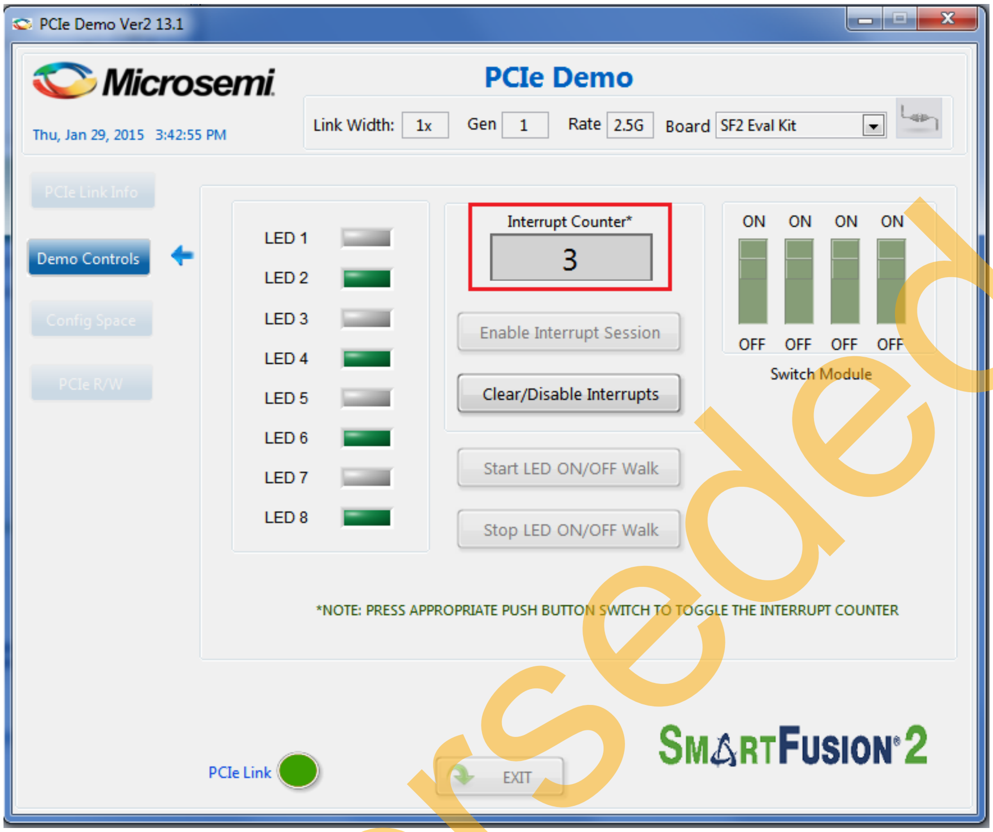
*NOTE: PRESS APPROPRIATE PUSH BUTTON SWITCH TO TOGGLE THE INTERRUPT COUNTER

Figure 74 • **Demo Controls**

5. Click LEDs in GUI to ON/OFF the LEDs on the SmartFusion2 Evaluation Kit.
6. Click **Start LED ON/OFF Walk** to blink the LEDs on the SmartFusion2 Evaluation Kit.
7. Click **Stop LED ON/OFF Walk** to stop the LEDs blinking.
8. Change the DIP switch positions on the SmartFusion2 Evaluation Kit (SW10) and observe the similar position of switches in GUI SWITCH MODULE.
9. Click **Enable Interrupt Session** to enable the PCIe interrupt.

Superseded

10. Press the push button SW3 on the SmartFusion2 Evaluation Kit and observe the interrupt count on the **Interrupt Counter** field in GUI as shown in Figure 75.



Thu, Jan 29, 2015 3:42:55 PM

Link Width: 1x Gen 1 Rate 2.5G Board SF2 Eval Kit

Interrupt Counter* 3

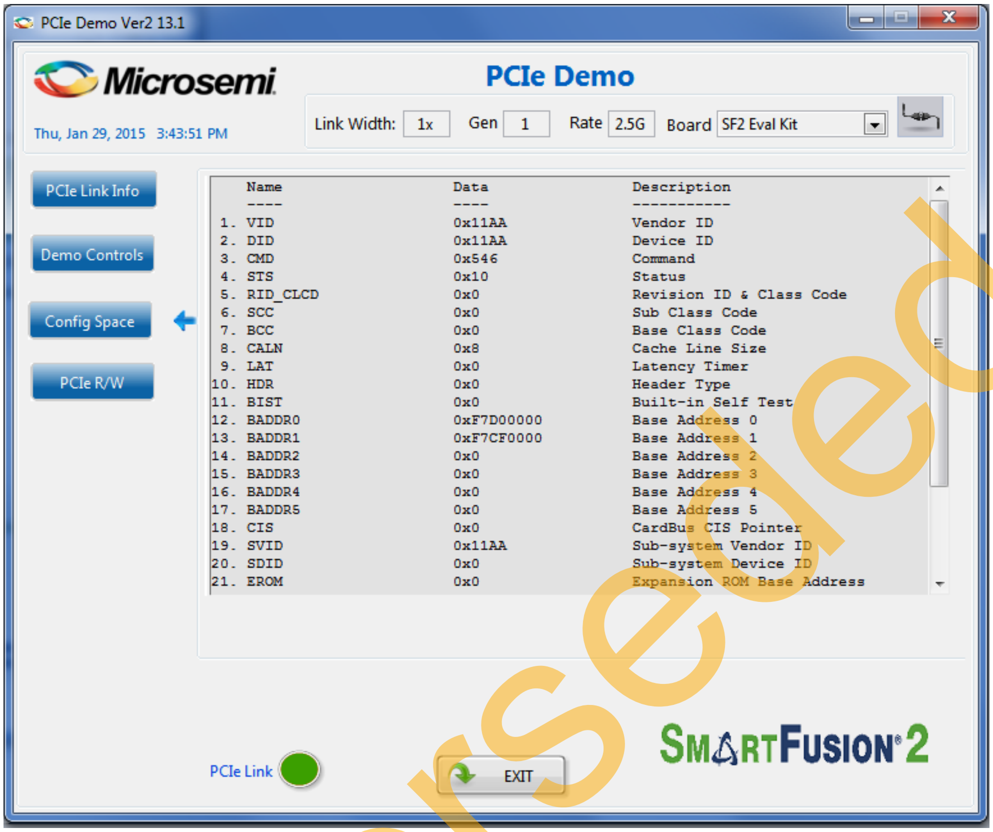
*NOTE: PRESS APPROPRIATE PUSH BUTTON SWITCH TO TOGGLE THE INTERRUPT COUNTER

Figure 75 **• Interrupt Counter**

11. Click **Clear/Disable Interrupts** to clear and disable the PCIe interrupts.

Superseded

12. Click **Config Space** to read details about the PCIe configuration space. Figure 76 shows the PCIe configuration space.



Thu, Jan 29, 2015 3:43:51 PM

Link Width: 1x Gen 1 Rate 2.5G Board SF2 Eval Kit

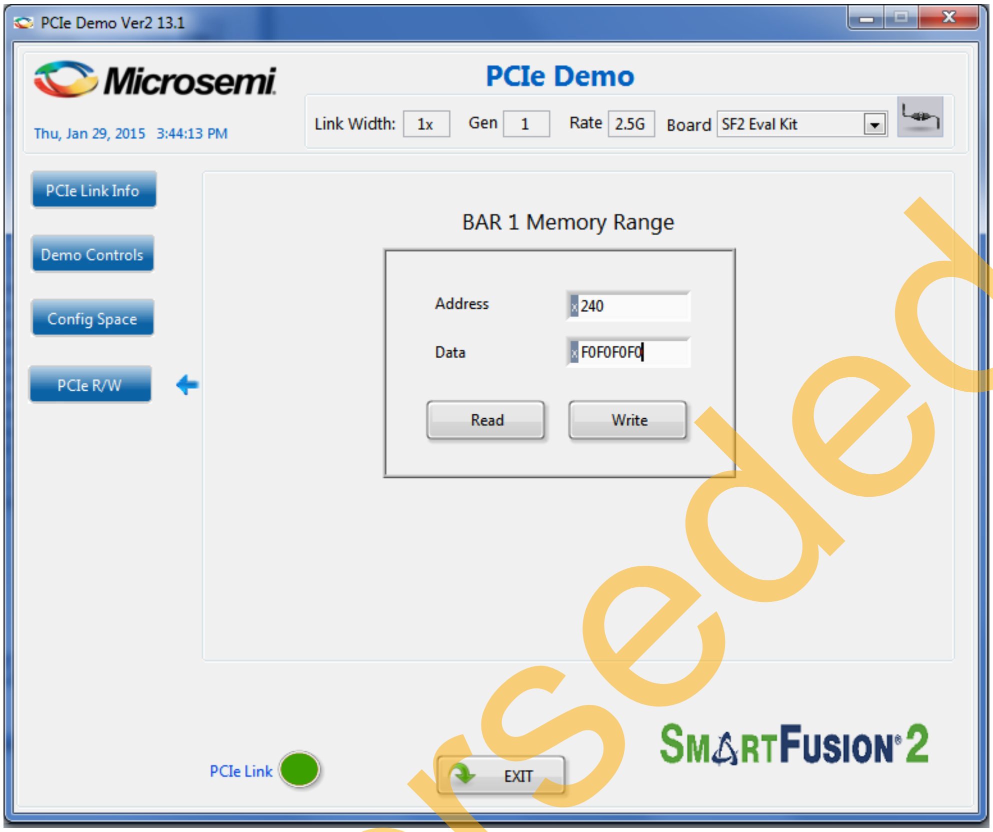
```
     Name              Data              Description
     ----              ----              -----------
 1.  VID               0x11AA            Vendor ID
 2.  DID               0x11AA            Device ID
 3.  CMD               0x546             Command
 4.  STS               0x10              Status
 5.  RID_CLCD          0x0               Revision ID & Class Code
 6.  SCC               0x0               Sub Class Code
 7.  BCC               0x0               Base Class Code
 8.  CALN              0x8               Cache Line Size
 9.  LAT               0x0               Latency Timer
10.  HDR               0x0               Header Type
11.  BIST              0x0               Built-in Self Test
12.  BADDR0            0xF7D00000        Base Address 0
13.  BADDR1            0xF7CF0000        Base Address 1
14.  BADDR2            0x0               Base Address 2
15.  BADDR3            0x0               Base Address 3
16.  BADDR4            0x0               Base Address 4
17.  BADDR5            0x0               Base Address 5
18.  CIS               0x0               CardBus CIS Pointer
19.  SVID              0x11AA            Sub-system Vendor ID
20.  SDID              0x0               Sub-system Device ID
21.  EROM              0x0               Expansion ROM Base Address
```

Figure 76 **• Configuration Space**

13. Click PCIe R/W to perform read and writes to eSRAM memory through BAR1 space. Figure 77 shows the PCIe R/W window.

Superseded

14. Enter the address in the Address field between 0x0000 to 0xFFFC. The Data field accepts a 32-bit hexadecimal value.



Link Width: 1x Gen 1 Rate 2.5G Board SF2 Eval Kit

Address x240

Data xF0F0F0F0

Figure 77 • **Perform Read and Write to eSRAM Using PCIe**

15. Click **Exit** to quit the demo.

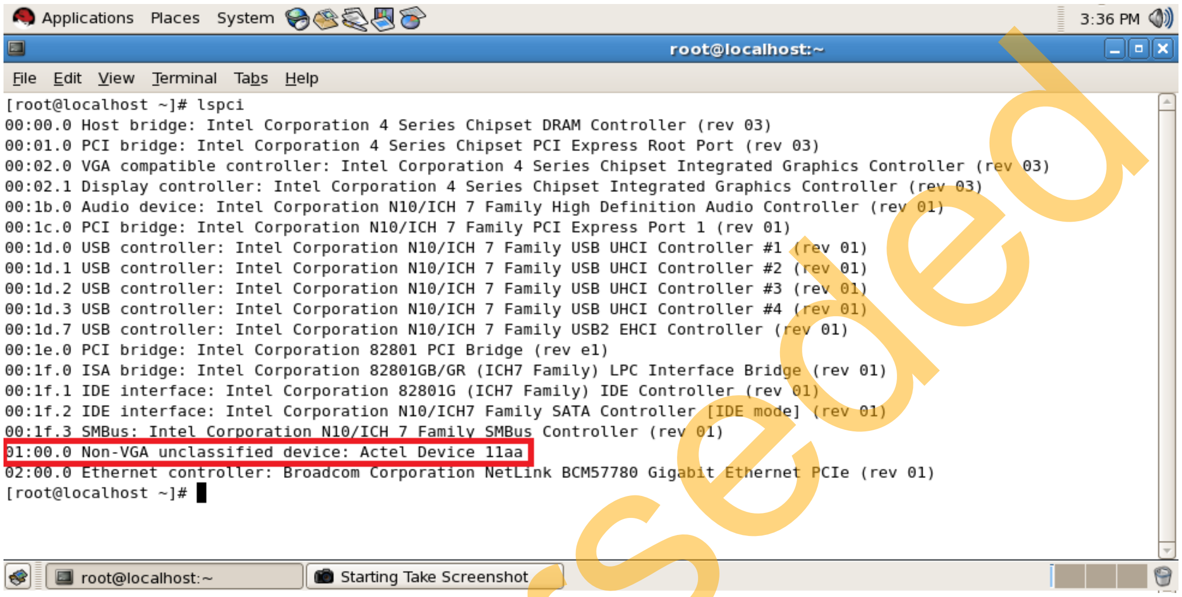
Superseded

Running the Demo Design on Linux

The following steps describe how to run the Demo Design on Linux.

1. Switch **ON** the Red Hat Linux Host PC.
2. Red Hat Linux Kernel detects the SmartFusion2 PCIe end point as Actel Device.
3. On Linux Command Prompt Use `lspci` command to display the PCIe info.

   # lspci



```
[root@localhost ~]# lspci
00:00.0 Host bridge: Intel Corporation 4 Series Chipset DRAM Controller (rev 03)
00:01.0 PCI bridge: Intel Corporation 4 Series Chipset PCI Express Root Port (rev 03)
00:02.0 VGA compatible controller: Intel Corporation 4 Series Chipset Integrated Graphics Controller (rev 03)
00:02.1 Display controller: Intel Corporation 4 Series Chipset Integrated Graphics Controller (rev 03)
00:1b.0 Audio device: Intel Corporation N10/ICH 7 Family High Definition Audio Controller (rev 01)
00:1c.0 PCI bridge: Intel Corporation N10/ICH 7 Family PCI Express Port 1 (rev 01)
00:1d.0 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #1 (rev 01)
00:1d.1 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #2 (rev 01)
00:1d.2 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #3 (rev 01)
00:1d.3 USB controller: Intel Corporation N10/ICH 7 Family USB UHCI Controller #4 (rev 01)
00:1d.7 USB controller: Intel Corporation N10/ICH 7 Family USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev e1)
00:1f.0 ISA bridge: Intel Corporation 82801GB/GR (ICH7 Family) LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801G (ICH7 Family) IDE Controller (rev 01)
00:1f.2 IDE interface: Intel Corporation N10/ICH7 Family SATA Controller [IDE mode] (rev 01)
00:1f.3 SMBus: Intel Corporation N10/ICH 7 Family SMBus Controller (rev 01)
01:00.0 Non-VGA unclassified device: Actel Device 11aa
02:00.0 Ethernet controller: Broadcom Corporation NetLink BCM57780 Gigabit Ethernet PCIe (rev 01)
[root@localhost ~]#
```

Figure 78 • **PCIe Device Detection**

Drivers Installation

Enter the following commands in the Linux command prompt to install the PCIe drivers:

1. Create the **sf2** directory under the **home/** directory using the following command:

   `# mkdir /home/sf2`

2. Copy the M2S90_PCIe_Control_Plane_DF design files folder under **/home/sf2** directory, which contains the Linux PCIe device driver files and Linux PCIe application utility files.
3. Copy the Linux PCIe Device Driver file (PCIe_Driver.rar) from M2S90_PCIe_Control_Plane_DF/design files folder.

   ```
   cp -rf/home/sf2/M2S90_PCIe_Control_Plane_DF/Linux_64bit/Drivers/
   PCIe_Driver.rar/home/sf2# unzip PCIe_Driver.rar
   ```

4. Execute `ls` command to display the contents of /home/sf2 directory.

   `# ls`

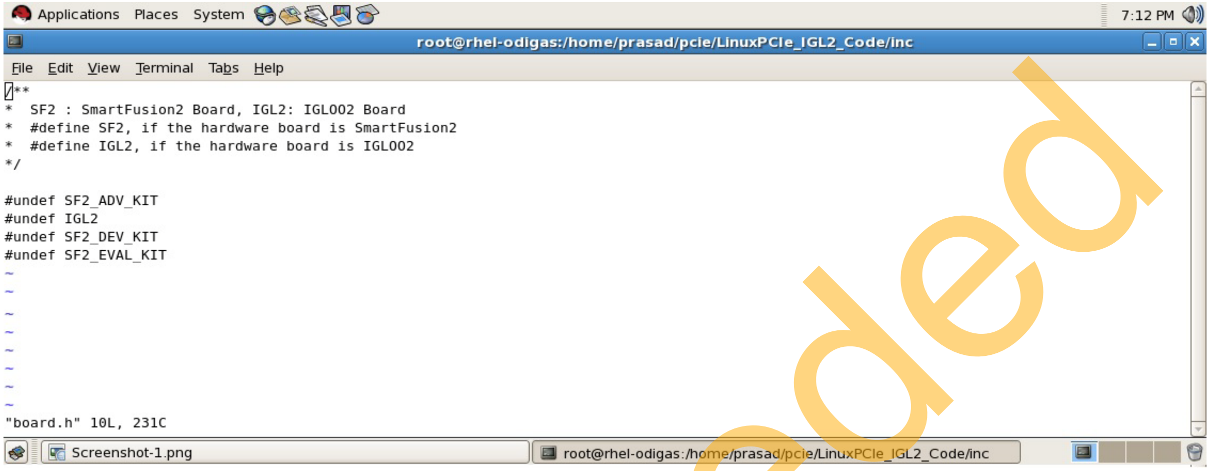
5. Change to inc/ directory by using the following command:

   `#cd /home/sf2/inc`

Superseded

6. Edit the `board.h` file for SmartFusion2 Evaluation Kit as shown in Figure 79.

```
#vi board.h
#undef SF2_ADV_KIT
#undef IGL2
#undef SF2_DEV_KIT
#define SF2_EVAL_KIT
```


Applications
Places
System
7:12 PM
root@rhel-odigas:/home/prasad/pcie/LinuxPCIe_IGL2_Code/inc
File
Edit
View
Terminal
Tabs
Help
Screenshot-1.png
root@rhel-odigas:/home/prasad/pcie/LinuxPCIe_IGL2_Code/inc

```
/**
*  SF2 : SmartFusion2 Board, IGL2: IGLOO2 Board
*  #define SF2, if the hardware board is SmartFusion2
*  #define IGL2, if the hardware board is IGLOO2
*/

#undef SF2_ADV_KIT
#undef IGL2
#undef SF2_DEV_KIT
#undef SF2_EVAL_KIT
~
~
~
~
~
~
~
~
"board.h" 10L, 231C
```

***Figure 79* • Edit board.h File**

7. To save the selected file, execute the `:wq` command
8. Change to PCIe_Driver/ directory using the `cd` command:

```
#cd /home/sf2/PCIe_Driver
```

9. To compile the Linux PCIe device driver code, execute `make` command.

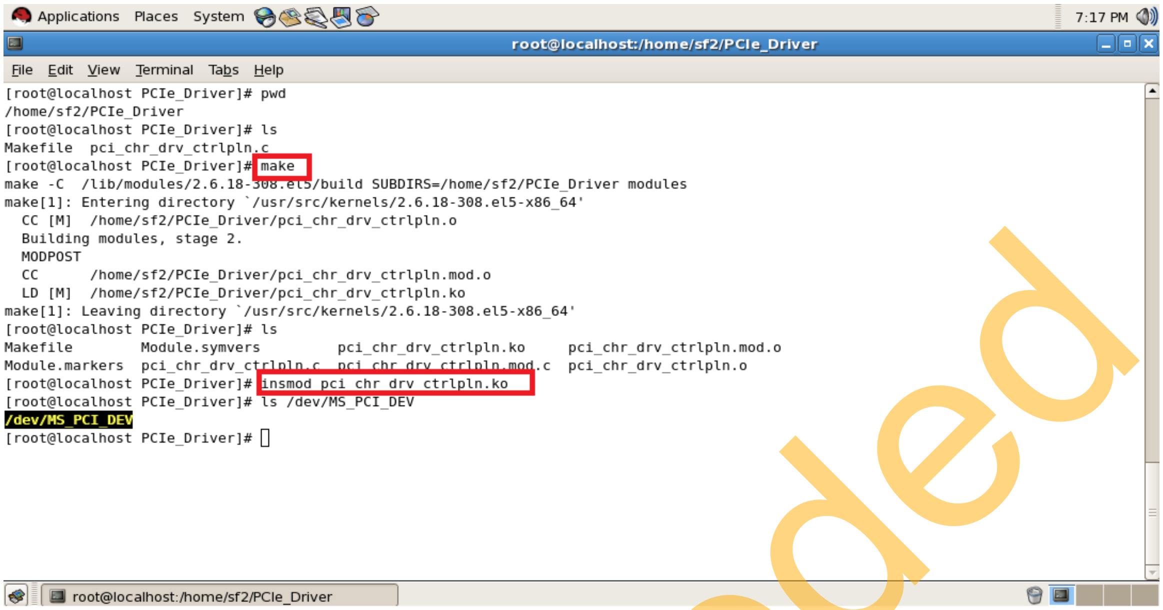
```
#make clean [To clean any *.o, *.ko files]
#make
```

10. The kernel module, `pci_chr_drv_ctrlpln.ko` creates in the same directory.
11. To insert the Linux PCIe device driver as a module, execute `insmod` command.

```
#insmod pci_chr_drv_ctrlpln.ko
```

Note: Root privileges are required to execute this command.

Superseded


Applications
Places
System
7:17 PM
root@localhost:/home/sf2/PCIe_Driver
File
Edit
View
Terminal
Tabs
Help
root@localhost:/home/sf2/PCIe_Driver

```
[root@localhost PCIe_Driver]# pwd
/home/sf2/PCIe_Driver
[root@localhost PCIe_Driver]# ls
Makefile  pci_chr_drv_ctrlpln.c
[root@localhost PCIe_Driver]# make
make -C  /lib/modules/2.6.18-308.el5/build SUBDIRS=/home/sf2/PCIe_Driver modules
make[1]: Entering directory `/usr/src/kernels/2.6.18-308.el5-x86_64'
  CC [M]  /home/sf2/PCIe_Driver/pci_chr_drv_ctrlpln.o
  Building modules, stage 2.
  MODPOST
  CC      /home/sf2/PCIe_Driver/pci_chr_drv_ctrlpln.mod.o
  LD [M]  /home/sf2/PCIe_Driver/pci_chr_drv_ctrlpln.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.18-308.el5-x86_64'
[root@localhost PCIe_Driver]# ls
Makefile        Module.symvers         pci_chr_drv_ctrlpln.ko     pci_chr_drv_ctrlpln.mod.o
Module.markers  pci_chr_drv_ctrlpln.c  pci_chr_drv_ctrlpln.mod.c  pci_chr_drv_ctrlpln.o
[root@localhost PCIe_Driver]# insmod pci_chr_drv_ctrlpln.ko
[root@localhost PCIe_Driver]# ls /dev/MS_PCI_DEV
/dev/MS_PCI_DEV
[root@localhost PCIe_Driver]# 
```

Figure 80 **• PCIe Device Driver Installation**

12. After successful Linux PCIe device driver installation, check `/dev/MS_PCI_DEV` got created by using the following Linux command:

    `#ls /dev/MS_PCI_DEV`

Note: `/dev/MS_PCI_DEV` interface is used to access the SmartFusion2 PCIe end point from Linux user space.

Linux PCIe Application Compilation and PCIe Control Plane Utility Creation

1. Change to the */home/sf2/* directory using the following command:

   `#cd /home/sf2`

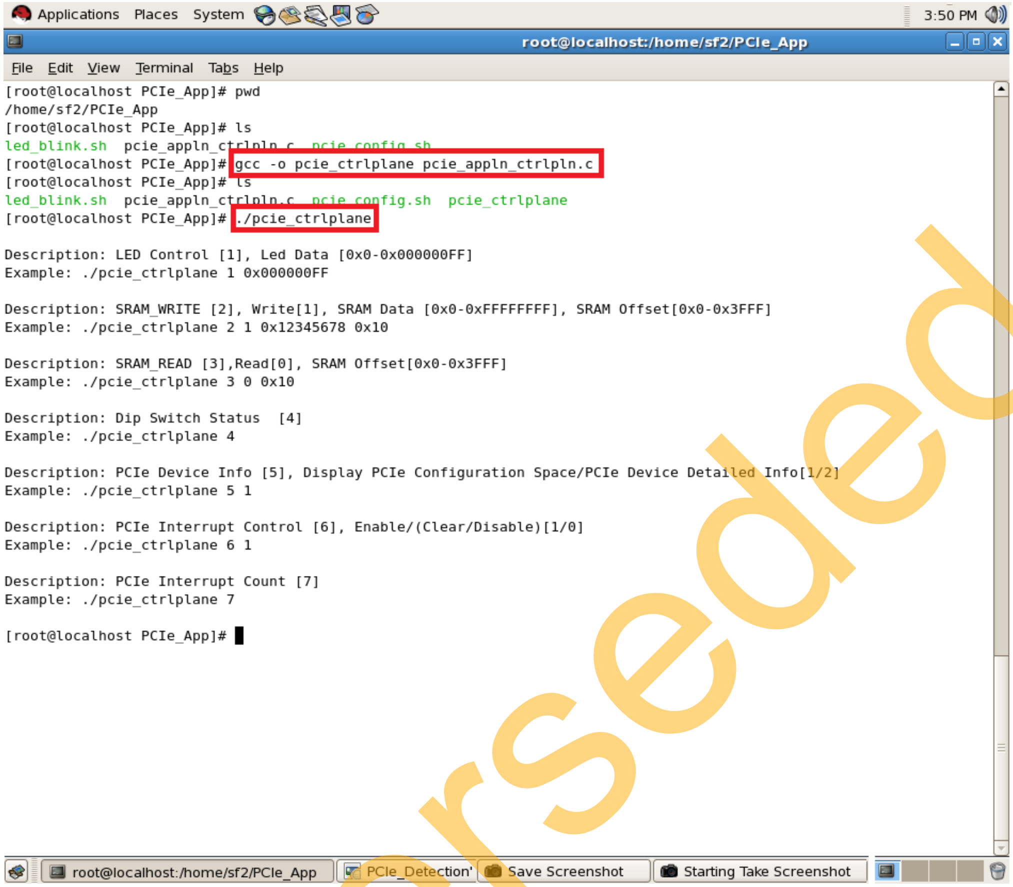
2. Copy the *M2S90_PCIE_Control_DEMO_DF\Linux_64bit\Util\PCIe_App* folder from the Windows host PC and place it into the /home/sf2 directory of RedHat Linux host PC.
3. Change to the **/home/sf2/PCIe_App** directory using the following command:

   `#cd /home/sf2/PCIe_App`

4. Compile the Linux user space application `pcie_appln_ctrlpln.c` by using `gcc` command.

   `#gcc -o pcie_ctrlplane pcie_appln_ctrlpln.c`

Superseded


Applications Places System
3:50 PM
root@localhost:/home/sf2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/sf2/PCIe_App
PCIe_Detection'
Save Screenshot
Starting Take Screenshot

```
[root@localhost PCIe_App]# pwd
/home/sf2/PCIe_App
[root@localhost PCIe_App]# ls
led_blink.sh  pcie_appln_ctrlpln.c  pcie_config.sh
[root@localhost PCIe_App]# gcc -o pcie_ctrlplane pcie_appln_ctrlpln.c
[root@localhost PCIe_App]# ls
led_blink.sh  pcie_appln_ctrlpln.c  pcie_config.sh  pcie_ctrlplane
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# █
```

***Figure 81* • Linux PCIe Application Utility**

5. After successful compilation, Linux PCIe application utility `pcie_ctrlplane` creates in the same directory.
6. On Linux Command Prompt run the `pcie_ctrlplane` utility as:

   `#./pcie_ctrlplane`

Help menu displays as shown in Figure 81.

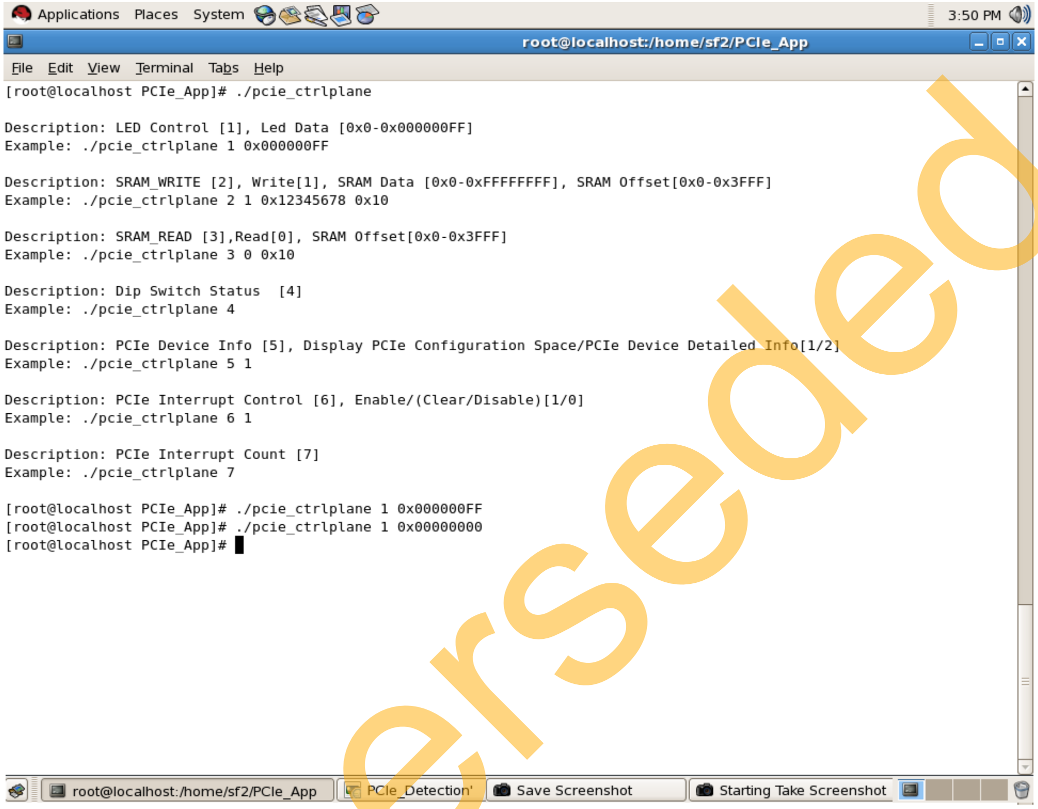
Superseded

Execution of Linux PCIe Control Plane Features

LED Control

LED1 to LED8 is controlled by writing data to SmartFusion2 LED Control Registers.

```
#./pcie_ctrlplane 1 0x000000FF [LED OFF]
#./pcie_ctrlplane 1 0x00000000 [LED ON]
```



```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 1 0x000000FF
[root@localhost PCIe_App]# ./pcie_ctrlplane 1 0x00000000
[root@localhost PCIe_App]# █
```

Figure 82 **• Linux Command - LED Control**

`led_blink.sh`, contains the shell script code to perform LED Walk ON where as `Ctrl C` exits the shell script and LED Walk turns OFF.

Run the led_blink.sh shell script using `sh` command.

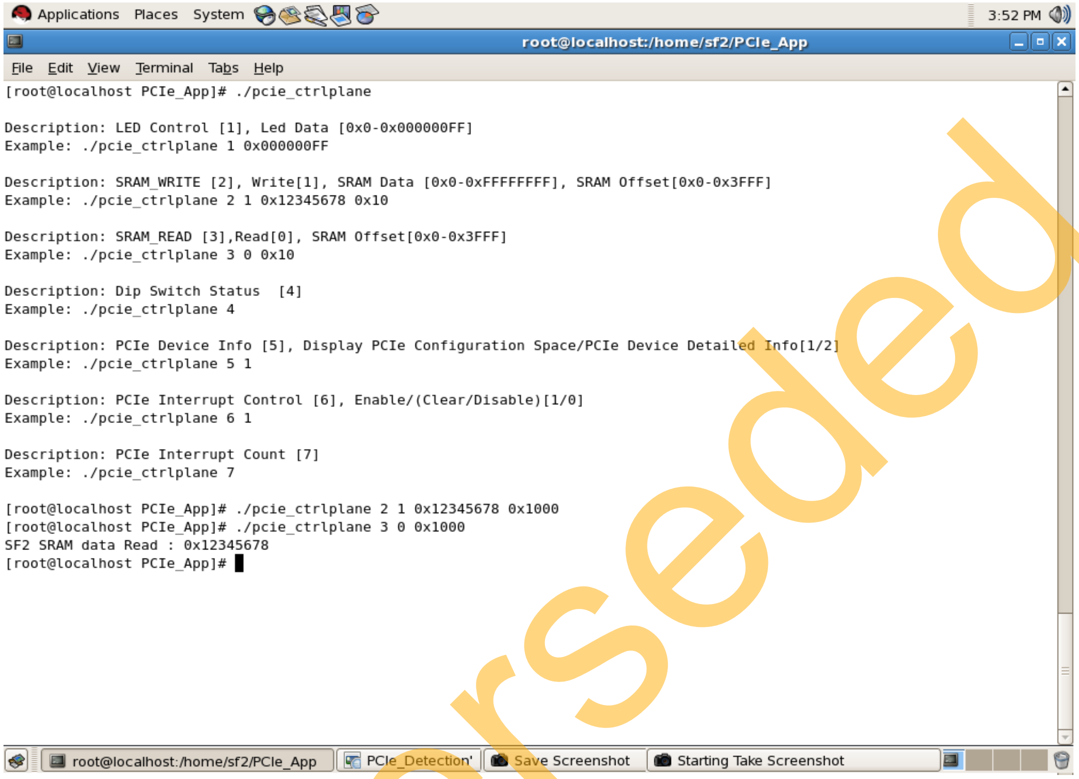
```
#sh led_blink.sh
```

Superseded

SRAM Read/Write

64 KB SRAM is accessible for SmartFusion2 Evaluation Kit.

```
#./pcie_ctrlplane 2 1 0xFF00FF00 0x1000 [SRAM WRITE]
#./pcie_ctrlplane 3 0 0x1000 [SRAM READ]
```


Applications Places System
3:52 PM
root@localhost:/home/sf2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/sf2/PCIe_App
PCIe_Detection'
Save Screenshot
Starting Take Screenshot

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 2 1 0x12345678 0x1000
[root@localhost PCIe_App]# ./pcie_ctrlplane 3 0 0x1000
SF2 SRAM data Read : 0x12345678
[root@localhost PCIe_App]# █
```

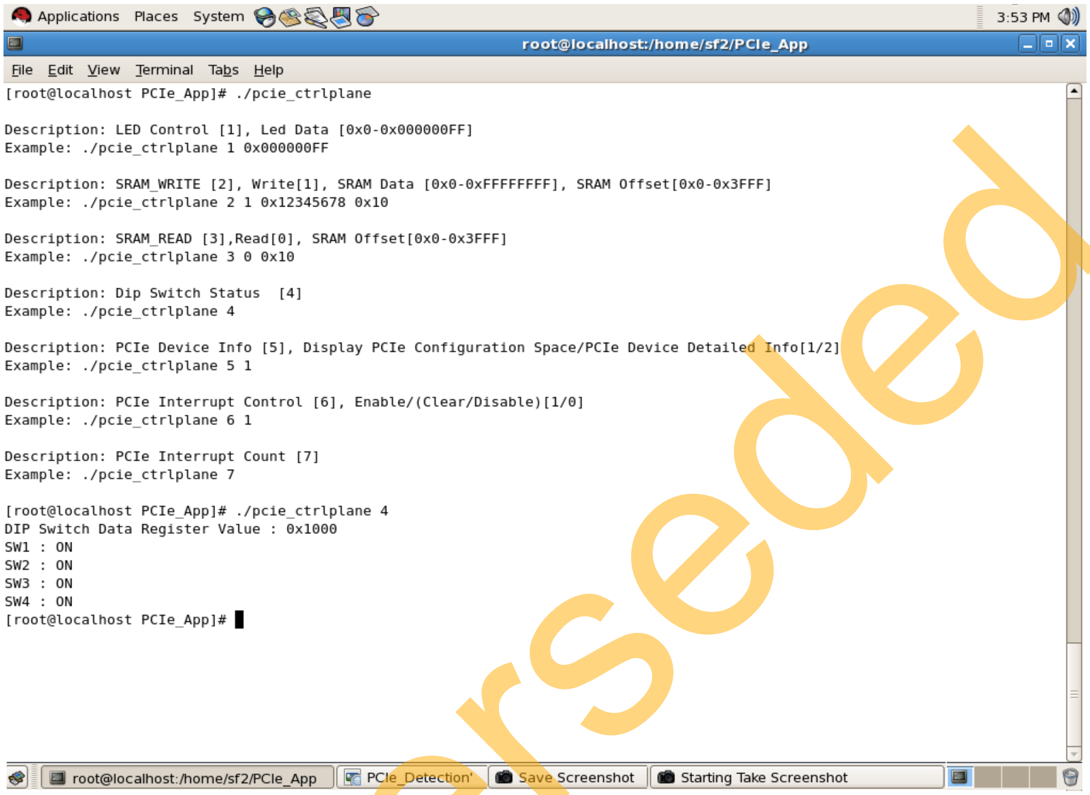
Figure 83 **• Linux Command - SRAM Read/Write**

Superseded

DIP Switch Status

Dip Switch on SmartFusion2 Evaluation Kit consists of 4 electric switches to hold the device configurations. Linux PCIe utility reads the corresponding switches (ON/OFF) state.

```
#./pcie_ctrlplane 4 [DIP Switch Status]
```



```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 4
DIP Switch Data Register Value : 0x1000
SW1 : ON
SW2 : ON
SW3 : ON
SW4 : ON
[root@localhost PCIe_App]# █
```

Figure 84 **• Linux Command - DIP Switch**

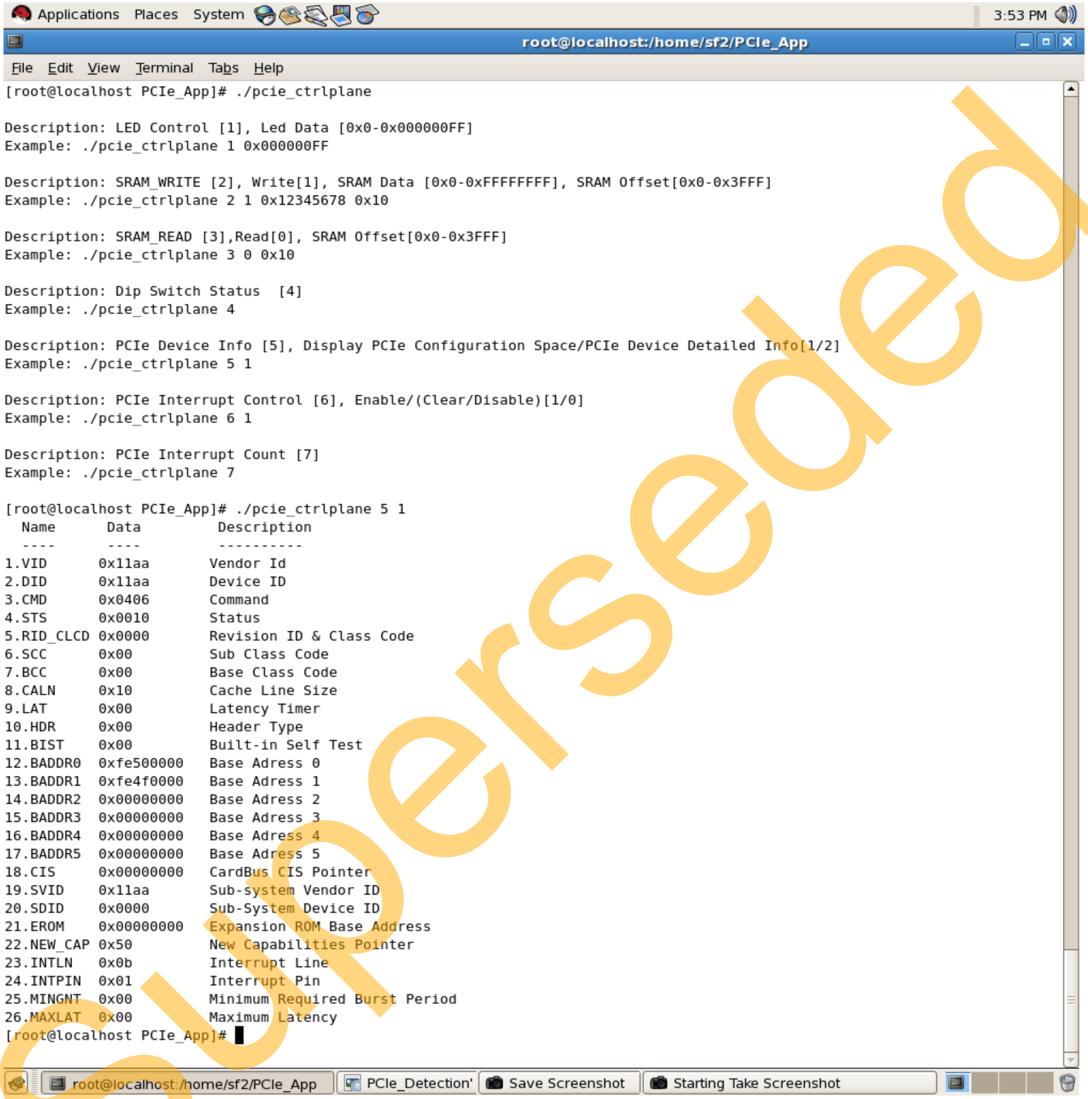
Superseded

PCIe Configuration Space Display

PCIe Configuration Space contains the PCIe device data, such as Vendor ID, Device ID, and Base Address 0.

Note: Root Privileges are required to execute this command.

```
#./pcie_ctrlplane 5 1 [Read PCIe Configuration Space]
```


Applications Places System
3:53 PM
root@localhost:/home/sf2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/sf2/PCIe_App
PCIe_Detection'
Save Screenshot
Starting Take Screenshot

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 5 1
  Name      Data         Description
  ----      ----         ----------
1.VID       0x11aa       Vendor Id
2.DID       0x11aa       Device ID
3.CMD       0x0406       Command
4.STS       0x0010       Status
5.RID_CLCD  0x0000       Revision ID & Class Code
6.SCC       0x00         Sub Class Code
7.BCC       0x00         Base Class Code
8.CALN      0x10         Cache Line Size
9.LAT       0x00         Latency Timer
10.HDR      0x00         Header Type
11.BIST     0x00         Built-in Self Test
12.BADDR0   0xfe500000   Base Adress 0
13.BADDR1   0xfe4f0000   Base Adress 1
14.BADDR2   0x00000000   Base Adress 2
15.BADDR3   0x00000000   Base Adress 3
16.BADDR4   0x00000000   Base Adress 4
17.BADDR5   0x00000000   Base Adress 5
18.CIS      0x00000000   CardBus CIS Pointer
19.SVID     0x11aa       Sub-system Vendor ID
20.SDID     0x0000       Sub-System Device ID
21.EROM     0x00000000   Expansion ROM Base Address
22.NEW_CAP  0x50         New Capabilities Pointer
23.INTLN    0x0b         Interrupt Line
24.INTPIN   0x01         Interrupt Pin
25.MINGNT   0x00         Minimum Required Burst Period
26.MAXLAT   0x00         Maximum Latency
[root@localhost PCIe_App]#
```

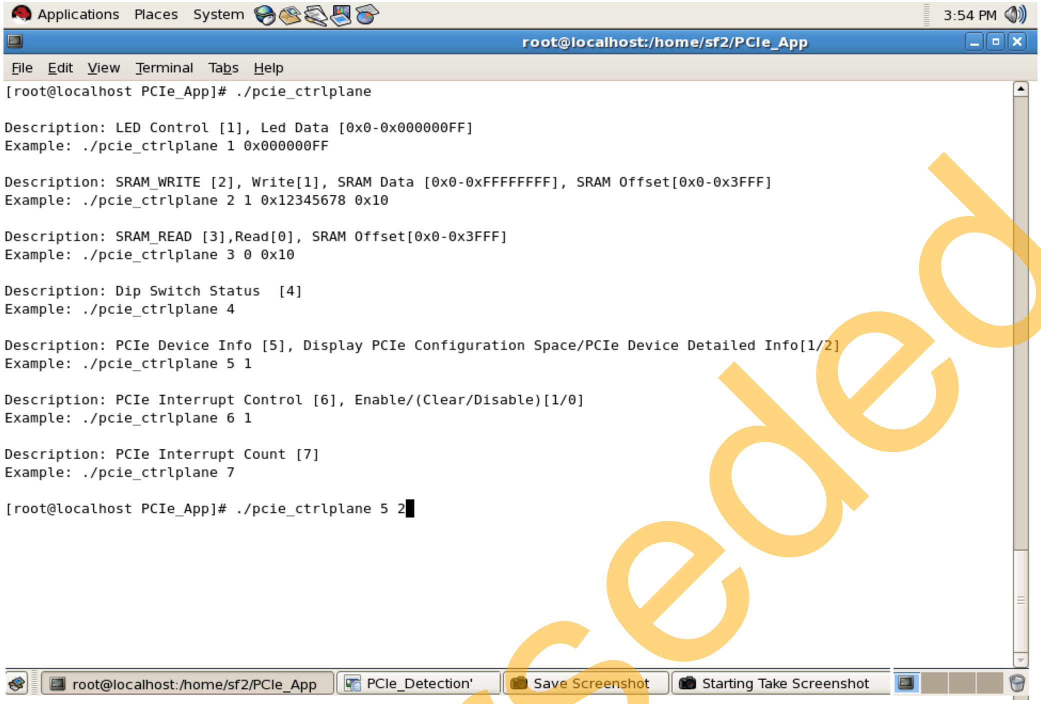
Figure 85 **• Linux Command - PCIe Configuration Space Display**

Superseded

PCIe Link Speed and Width

Note: Root Privileges are required to execute this command.

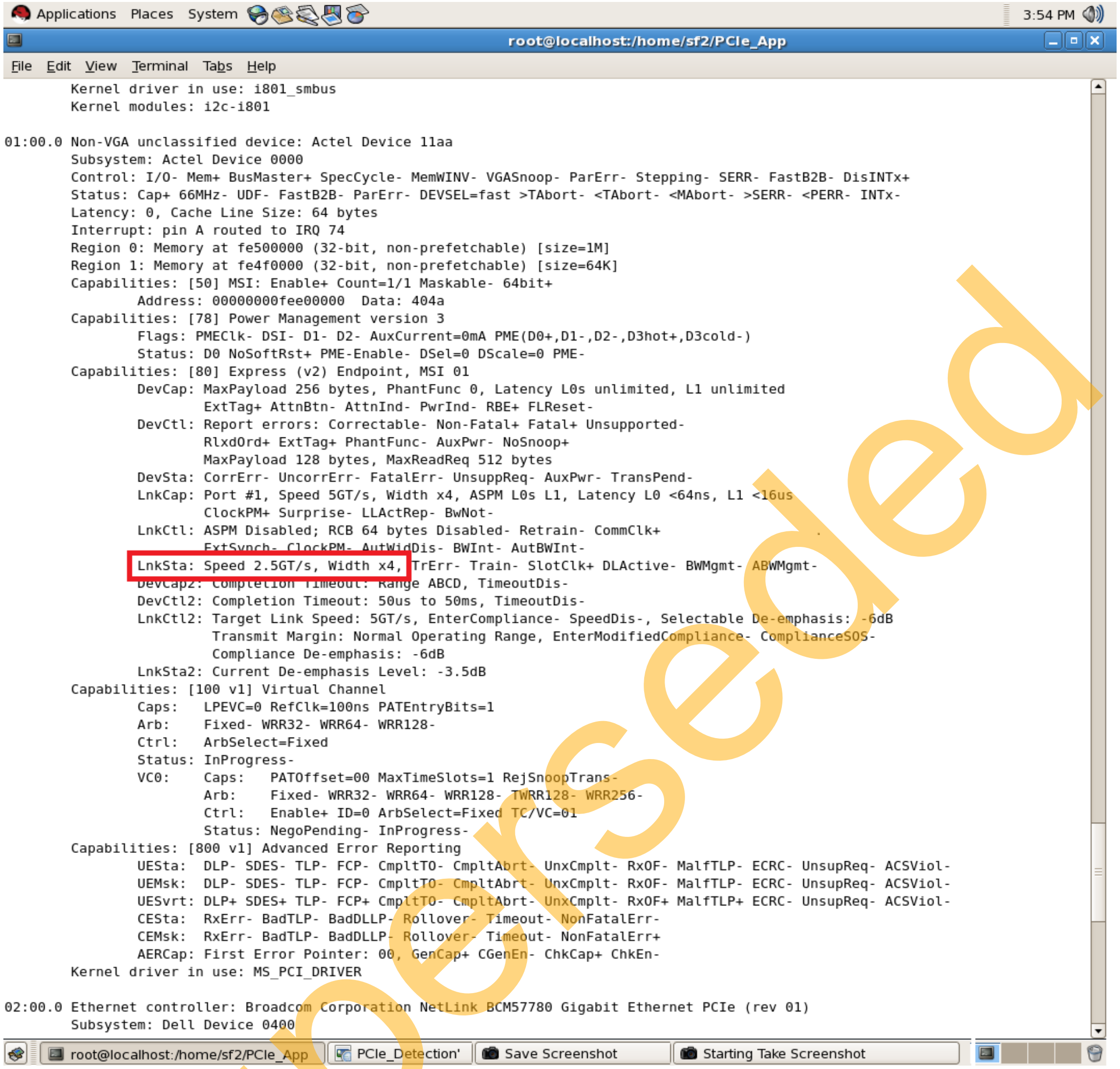
```
#./pcie_ctrlplane 5 2 [Read PCIe Link Speed and Link Width]
```


Applications Places System
3:54 PM
root@localhost:/home/sf2/PCIe_App
File Edit View Terminal Tabs Help
root@localhost:/home/sf2/PCIe_App
PCIe_Detection'
Save Screenshot
Starting Take Screenshot

```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 5 2
```

Figure 86 • **Linux Command - PCIe Link Speed and Width**

Superseded



```
        Kernel driver in use: i801_smbus
        Kernel modules: i2c-i801

01:00.0 Non-VGA unclassified device: Actel Device 11aa
        Subsystem: Actel Device 0000
        Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
        Latency: 0, Cache Line Size: 64 bytes
        Interrupt: pin A routed to IRQ 74
        Region 0: Memory at fe500000 (32-bit, non-prefetchable) [size=1M]
        Region 1: Memory at fe4f0000 (32-bit, non-prefetchable) [size=64K]
        Capabilities: [50] MSI: Enable+ Count=1/1 Maskable- 64bit+
                Address: 00000000fee00000  Data: 404a
        Capabilities: [78] Power Management version 3
                Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0+,D1-,D2-,D3hot+,D3cold-)
                Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
        Capabilities: [80] Express (v2) Endpoint, MSI 01
                DevCap: MaxPayload 256 bytes, PhantFunc 0, Latency L0s unlimited, L1 unlimited
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset-
                DevCtl: Report errors: Correctable- Non-Fatal+ Fatal+ Unsupported-
                        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+
                        MaxPayload 128 bytes, MaxReadReq 512 bytes
                DevSta: CorrErr- UncorrErr- FatalErr- UnsuppReq- AuxPwr- TransPend-
                LnkCap: Port #1, Speed 5GT/s, Width x4, ASPM L0s L1, Latency L0 <64ns, L1 <16us
                        ClockPM+ Surprise- LLActRep- BwNot-
                LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- Retrain- CommClk+
                        ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
                LnkSta: Speed 2.5GT/s, Width x4, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
                DevCap2: Completion Timeout: Range ABCD, TimeoutDis-
                DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-
                LnkCtl2: Target Link Speed: 5GT/s, EnterCompliance- SpeedDis-, Selectable De-emphasis: -6dB
                         Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
                         Compliance De-emphasis: -6dB
                LnkSta2: Current De-emphasis Level: -3.5dB
        Capabilities: [100 v1] Virtual Channel
                Caps:   LPEVC=0 RefClk=100ns PATEntryBits=1
                Arb:    Fixed- WRR32- WRR64- WRR128-
                Ctrl:   ArbSelect=Fixed
                Status: InProgress-
                VC0:    Caps:   PATOffset=00 MaxTimeSlots=1 RejSnoopTrans-
                        Arb:    Fixed- WRR32- WRR64- WRR128- TWRR128- WRR256-
                        Ctrl:   Enable+ ID=0 ArbSelect=Fixed TC/VC=01
                        Status: NegoPending- InProgress-
        Capabilities: [800 v1] Advanced Error Reporting
                UESta:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
                UEMsk:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
                UESvrt: DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+ ECRC- UnsupReq- ACSViol-
                CESta:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr-
                CEMsk:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- NonFatalErr+
                AERCap: First Error Pointer: 00, GenCap+ CGenEn- ChkCap+ ChkEn-
        Kernel driver in use: MS_PCI_DRIVER

02:00.0 Ethernet controller: Broadcom Corporation NetLink BCM57780 Gigabit Ethernet PCIe (rev 01)
        Subsystem: Dell Device 0400
```

Figure 87 **• Linux Command - PCIe Link Speed and Width**

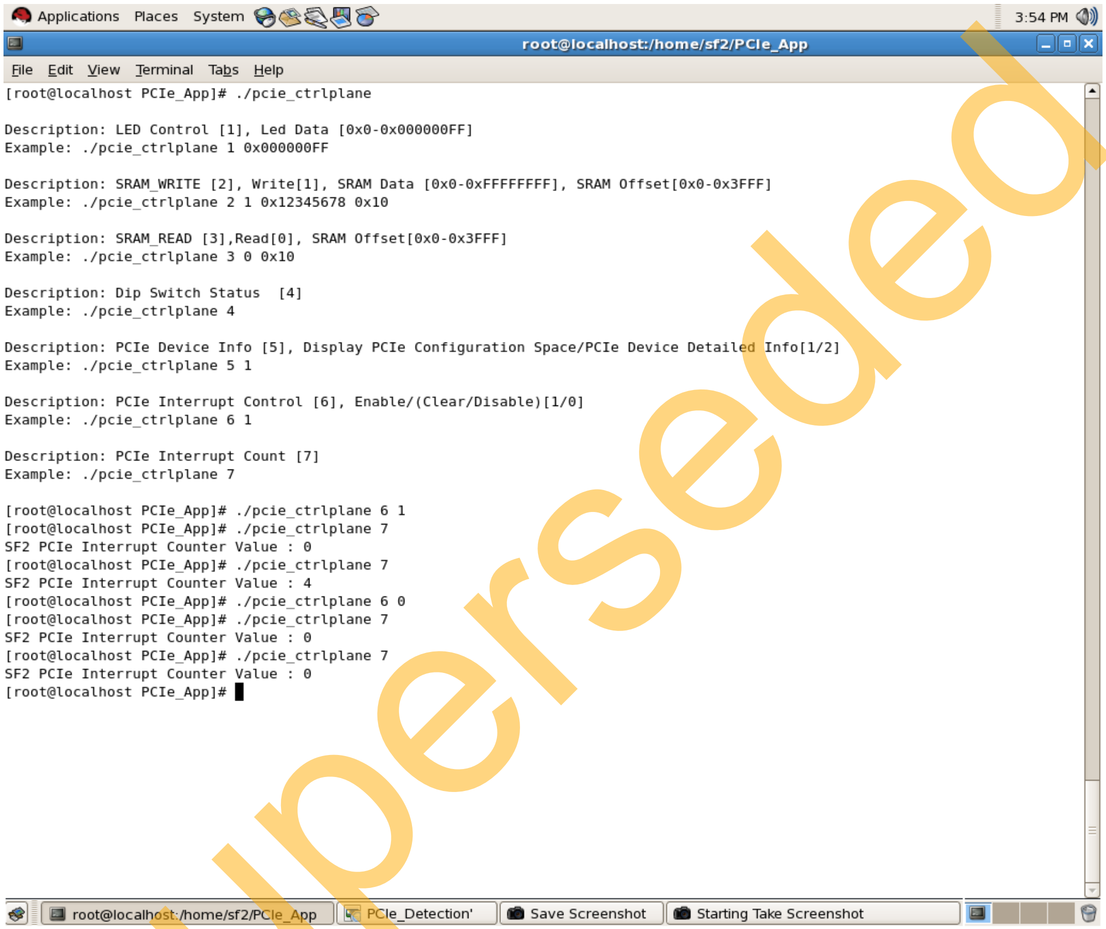
Superseded

PCIe Interrupt Control (Enable/Disable) and Interrupt Counter

SmartFusion2 Evaluation Kit enable/disable the MSI interrupts by writing data to its PCIe configuration space.

Interrupt Counter holds the number of MSI interrupts got triggered by pressing the SW3 Push Button.

```
#. /pcie_ctrlplane 6 0 [Disable Interrupts]
#. /pcie_ctrlplane 6 1 [Enable Interrupts]
#. /pcie_ctrlplane 7 [Interrupt Counter Value]
```



```
[root@localhost PCIe_App]# ./pcie_ctrlplane

Description: LED Control [1], Led Data [0x0-0x000000FF]
Example: ./pcie_ctrlplane 1 0x000000FF

Description: SRAM_WRITE [2], Write[1], SRAM Data [0x0-0xFFFFFFFF], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 2 1 0x12345678 0x10

Description: SRAM_READ [3],Read[0], SRAM Offset[0x0-0x3FFF]
Example: ./pcie_ctrlplane 3 0 0x10

Description: Dip Switch Status  [4]
Example: ./pcie_ctrlplane 4

Description: PCIe Device Info [5], Display PCIe Configuration Space/PCIe Device Detailed Info[1/2]
Example: ./pcie_ctrlplane 5 1

Description: PCIe Interrupt Control [6], Enable/(Clear/Disable)[1/0]
Example: ./pcie_ctrlplane 6 1

Description: PCIe Interrupt Count [7]
Example: ./pcie_ctrlplane 7

[root@localhost PCIe_App]# ./pcie_ctrlplane 6 1
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 0
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 4
[root@localhost PCIe_App]# ./pcie_ctrlplane 6 0
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 0
[root@localhost PCIe_App]# ./pcie_ctrlplane 7
SF2 PCIe Interrupt Counter Value : 0
[root@localhost PCIe_App]# 
```

Figure 88 • **Linux Command - PCIe Interrupt Control**

Conclusion

This demo describes how to access the PCIe endpoint features of SmartFusion2, create a simple design, and verify the design using BFM simulation. This demo demonstrates that the Host PC can easily communicate with the SmartFusion2 Evaluation Kit through the provided GUI and Drivers. This demo also provides a Linux PCIe application for accessing PCIe EP device through Linux PCIe Device Driver.

Superseded

A – List of Changes

The following table lists the critical changes that were made in each revision of the chapter in the user guide.

Date	Changes	Page
Revision 4 (February 2015)	Updated the document for Libero v11.5 software release (SAR 64184).	NA
Revision 3 (August 2014)	Updated the document for Libero v11.4 software release (SAR 59644).	NA
Revision 2 (April 2014)	Updated the document for Libero v11.3 software release (SAR 56081).	NA
Revision 1 (December 2013)	Updated the document for Libero v11.2 software release (SAR 52109) (SAR 52909) and (SAR 50779).	NA
Revision 0 (June 2013)	Initial Release	NA

Superseded

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/designsupport/fpga-soc-support.

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.

Superseded

My Cases

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

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

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

Superseded

Superseded

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; 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**.

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

© 2015 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.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.