SmartFusion2 PCIe Data Plane Demo using MSS HPDMA and SMC_FIC - Libero SoC v11.7

DG0535 Demo Guide

Microsemi
Power Matters.™

Contents

Figures

Tables

1 Preface

1.1 Purpose

This demo is for SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

1.2 Intended Audience

This demo guide is intended for:

- FPGA designers
- Embedded designers
- System-level designers

1.3 References

See the following web page for a complete and up-to-date listing of SmartFusion2 device documentation: *http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2*

The following references are used in this document:

- *SmartFusion2 and IGLOO2 High Speed Serial Interface Configuration*
- *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*
- *UG0447: SmartFusion2 and IGLOO2 High Speed Serial Interfaces User Guide*
- *UG0456: SmartFusion2 SoC FPGA PCIe Control Plane Demo User Guide*

2 SmartFusion2 Data Plane Demo using MSS HPDMA and SMC_FIC

2.1 Introduction

This demo describes the usage of the embedded features of the SmartFusion2 devices such as peripheral component interconnect express (PCIe) controller, microcontroller subsystem (MSS) high-performance direct memory access (HPDMA) controller and soft memory controller - fabric interface controller (SMC_FIC). The demo uses all of these embedded features and limited FPGA resources. The objective of this demo is to show ease-of-use, optimized resource utilization and low power. In this demo, the PCIe advanced extensible interface (AXI) is accessed through the SMC_FIC AXI interface. This demo shows the performance of the PCIe and HPDMA through SMC_FIC of the SmartFusion2 device.

An application, **PCIe_Demo** that runs in the host PC is provided for setting up and initiating the DMA transactions from the SmartFusion2 PCIe endpoint to the host PC device. Drivers for connecting the host PC to the SmartFusion2 PCIe endpoint are provided as part of the demo deliverables.

Microsemi provides three different PCIe data plane demos for SmartFusion2 devices:

- *DG0501: SmartFusion2 PCIe MSS HPDMA Demo Guide*: This demo shows the low throughput data transfers between PCIe and double data rate (DDR).
- **PCIe data plane demo using MSS HPDMA and SMC_FIC** (current demo)**:** This demo shows the medium throughput data transfers between PCIe and embedded static random access memory (eSRAM).
- *SmartFusion2 and IGLOO2 PCIe Data Plane Demo using 2 Channel Fabric DMA Demo Guide*: This demo shows the high throughput data transfers between PCIe and large SRAM (LSRAM).

The high-speed serial interface (SERDESIF) available in the SmartFusion2 devices provides a fully hardened PCIe endpoint implementation and is compliant to the PCIe Base Specification Revision 2.0 1.1, and 1.0. For more information, see the *UG0447: SmartFusion2 SoC FPGA High Speed Serial Interfaces User Guide*.

For a tutorial design on how to develop and use the PCIe endpoint including the tools flow and simulation, see the *UG0456: SmartFusion2 SoC FPGA PCIe Control Plane Demo Users Guide*.

2.2 Design Requirements

The following table lists the hardware and software design requirements.

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Advanced Development Kit: • FlashPro5 programmer • 12 V adapter • PCI Edge Card Ribbon Cable	Rev B or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero® System-on-Chip (SoC) software	v11.7
SoftConsole	v3.4 SP1*
FlashPro programming software	v11.7
PCIe_Demo application	13.1 or later
Note: **For this demo guide, SoftConsole v3.4 SP1 is used. For using SoftConsole v4.0, see the TU0546: SoftConsole v4.0 and Libero SoC v11.7 Tutorial.*	

2.3 Demo Design

2.3.1 Introduction

The demo design files are available for download from the following path in the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2s_dg0535_liberov11p7_df*

The demo design files include:

- Drivers_64bit OS
- Libero project
- Programming files
- Readme.txt file

Figure 1 shows the top-level structure of the design files. For more details, see the `readme.txt` file.

Figure 1 • **Top-Level Structure of Demo Design Files**

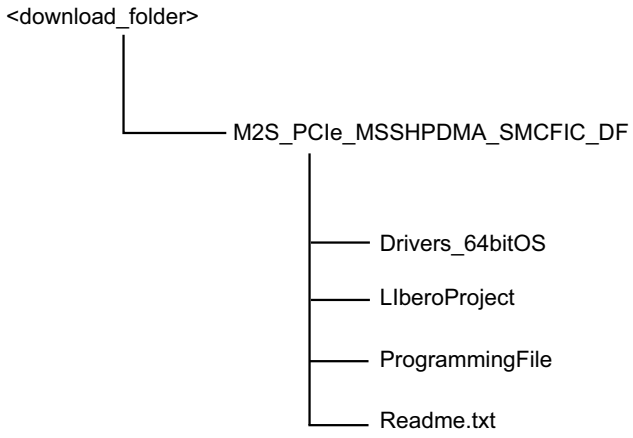


Figure 2 • **PCIe Data Plane Demo Block Diagram**

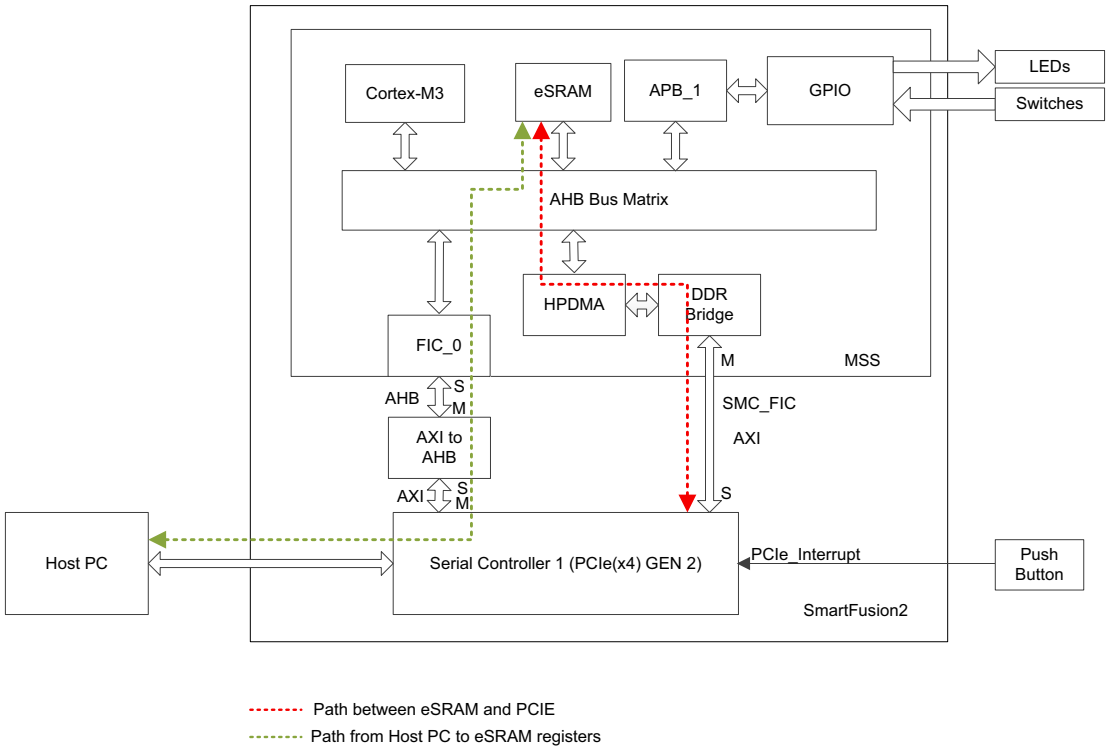


Figure 2 shows the demo-design. The PCIe core in the SmartFusion2 devices supports both AXI and AMBA® high-performance bus (AHB) master and slave interfaces. This demo design uses the AXI master and slave interfaces to achieve maximum bandwidth.

The PCIe_Demo application on the host PC initiates the DMA transfers and the embedded PCIe core in the SmartFusion2 device initiates the AXI transactions through the AXI master interface to the AXI to AHB logic in the FPGA fabric. The AXI to AHB logic initiates AHB transactions to embedded SRAM (eSRAM) through FIC_0 (the green line in Figure 2 shows this path). The firmware running on the ARM® Cortex®-M3 processor reads the registers in eSRAM and initializes the HPDMA controller depending on the type of DMA transfer. In this demo, FIC_0 interface is used only for configuring registers in eSRAM for initiating HPDMA.

The PCIe AXI slave interface is connected to SMC_FIC AXI interface. The HPDMA accesses PCIe through the SMC_FIC and performs the DMA transactions to eSRAM (the red line in Figure 2 shows this path). In this demo, all data transfers occur through the AXI based SMC_FIC interface.

Note:

- The SMC_FIC AXI interface supports only WRAP type of Read burst transactions. The fabric logic converts these WRAP type transactions into INCR type transactions as the PCIe AXI interface supports only INCR type of burst transfers.
- The transaction addresses must be 32 byte aligned.

In this demo design, the following configurations are done:

- The SERDES_IF_0 block in the SmartFusion2 device is configured for PCIe 2.0, x4 lanes and Gen2 rate.
- BAR0 and BAR1 are configured in 32-bit memory mapped memory mode. The AXI master window 0 is enabled and configured to map the BAR0 memory address space to MSS general purpose input output (GPIO) address space to control the MSS GPIOs. The AXI master window 1 is enabled and configured to map the BAR1 memory address space to eSRAM address space to perform read and write operations from the PCIe interface. The AXI slave window 0 is enabled and configured to map the SmartFusion2 local address space to the host PC address space.
- MSS GPIO block is enabled and configured as:
  - 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 AXI interface clock and the Cortex-M3 processor clock are configured to run at 75 MHz.

2.3.2 Demo Design Features

- DMA data transfers between the host PC memory and the eSRAM block.
- Throughput for every DMA data transfer.
- Enables continuous DMA transfers for observing throughput variations.
- Displays the PCIe link enable or disable, negotiated link width, and the link speed on the PCIe_Demo application.
- Displays the position of DIP Switches on the SmartFusion2 Advanced Development Kit board on the PCIe_Demo application.
- Displays the PCIe configuration space on the PCIe_Demo application.
- Controls LEDs on the board according to the command from the PCIe_Demo application.
- Enables read and write operations to scratchpad register in the FPGA fabric.
- Interrupts the host PC, when the Push button is pressed. The PCIe_Demo application displays the count value of the number of interrupts sent from the board.

2.3.3 Demo Design Description

This demo supports the following data transfers:

- Host PC Memory to eSRAM
- eSRAM to Host PC Memory

2.3.3.1 Host PC Memory to eSRAM

A data transfer from PC memory to the eSRAM block happens in the following sequence:

1. HPDMA is setup over the PCIe link based on the settings in the GUI.
2. HPDMA initiates AHB read transactions to the DDR bridge.
3. The DDR bridge converts these AHB read transactions into AXI read transactions (32-byte burst) to the PCIe AXI interface.
4. The PCIe core sends a memory read (MRd) transaction layer packets (TLP) to the host PC.
5. The host PC returns with a completion with data (CplD) TLP to the PCIe link.
6. This return data completes the AXI read initiated by DDR bridge. The DDR bridge stores this data into read buffer.
7. The DDR bridge returns this buffered data to HPDMA. The return data completes the AHB read initiated by HPDMA controller.
8. HPDMA writes the return data to eSRAM.
9. HPDMA repeats this process until the transfer size set in the host PC GUI is completed.

2.3.3.2 eSRAM to Host PC Memory

A data transfer from the eSRAM to PC memory happens in the following sequence:

1. HPDMA is setup over the PCIe link based on the settings in the GUI.
2. HPDMA reads the data from eSRAM by initiating an AHB read transaction to eSRAM.
3. The data is written to the PCIe core as an AHB write transaction through the DDR bridge. The DDR bridge buffers up to 32 bytes of these write transactions.
4. The DDR bridge initiates an AXI write transaction (32 byte burst) to the PCIe AXI interface.
5. The PCIe core sends a memory write (MWr) TLP to the host PC.
6. HPDMA repeats this process until the transfer size set in the host PC GUI is completed.

2.4 Throughput Calculation

This demo uses MSS timer to measure the throughput of DMA transfers. The throughput measured includes all of the overhead of the AXI, PCIe, and DMA controller transactions. The procedure for measuring throughput is:

1. Setup the DMA controller for the data transfer.
2. Start the MSS timer and the DMA controller.
3. Initiate data transfer for the requested number of bytes.
4. Wait until DMA transfer is completed.
5. Record the number of clock cycles consumed for steps 2-4.

To arrive at a realistic system performance, the throughput calculation takes into account all the overheads during a transfer. The throughput formula is as shown below:

Throughput = Transfer Size (Bytes) / (Number of clock cycles taken for a transfer * Clock Period)

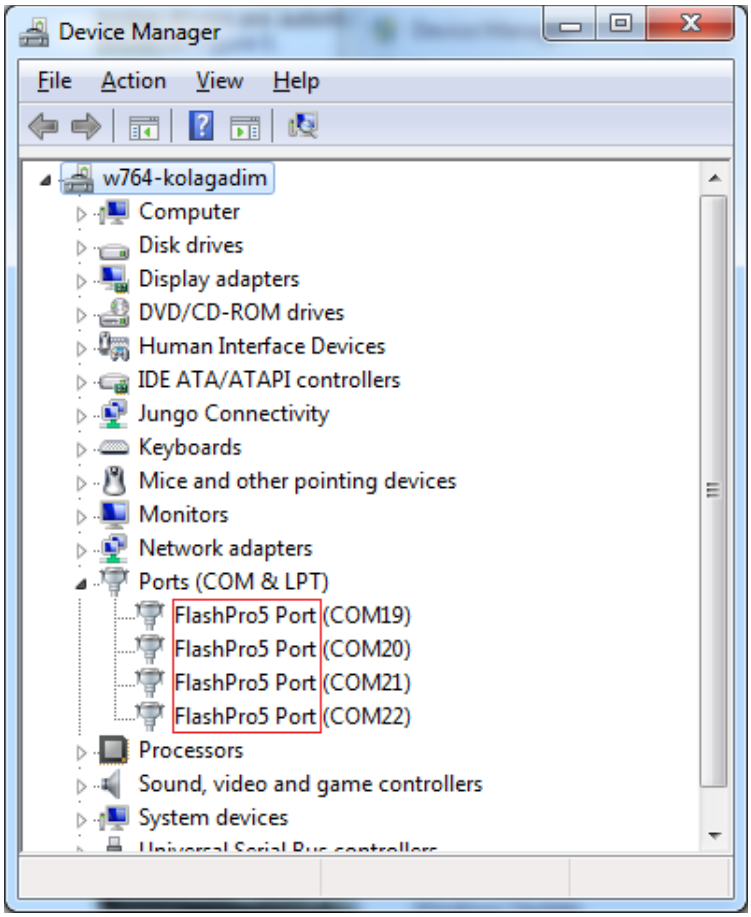
EQ 1

2.5 Setting Up the Demo Design

2.5.1 Jumper Settings for SmartFusion2 Advanced Development Kit

1. Connect the host PC to the **J33** Connector using the USB A to mini-B cable. The USB to UART bridge drivers are automatically detected. Verify, if the detection is made in the device manager as shown in Figure 3.

Figure 3 • **Device Manager**



2. Connect the jumpers on the SmartFusion2 Advanced Development Kit, as shown in Table 2. **CAUTION**: While making the jumper connections, the power supply switch **SW7** must be switched Off.

Table 2 • **SmartFusion2 Advanced Development Kit Jumper Settings**

Jumper	Pin (From)	Pin (To)	Comments
J116, J353, J354, J54	1	2	Default
J123	2	3	Default
J124, J121, J32	1	2	JTAG programming via FTDI

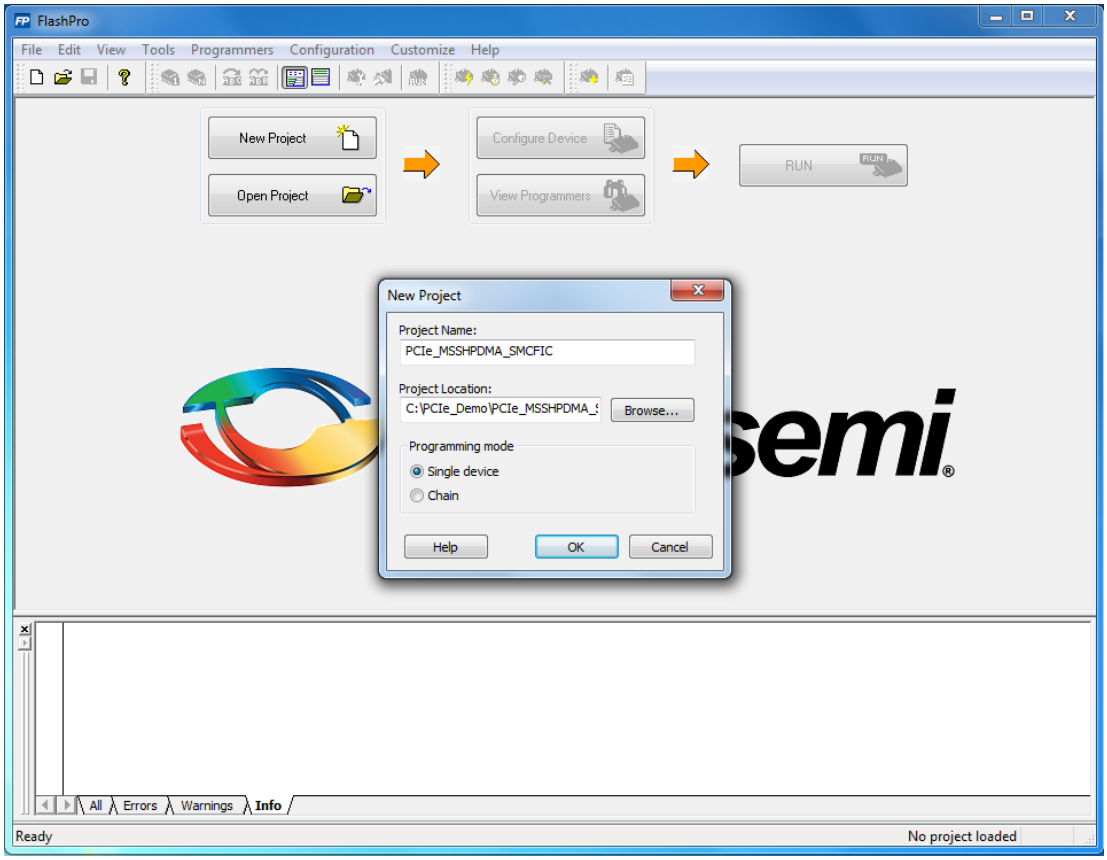
3. Connect the power supply to the **J42** connector.
4. Switch on the power supply switch, **SW7**.

2.5.2 Programming the Device

Download the demo design from:
http://soc.microsemi.com/download/rsc/?f=m2s_dg0535_liberov11p7_df

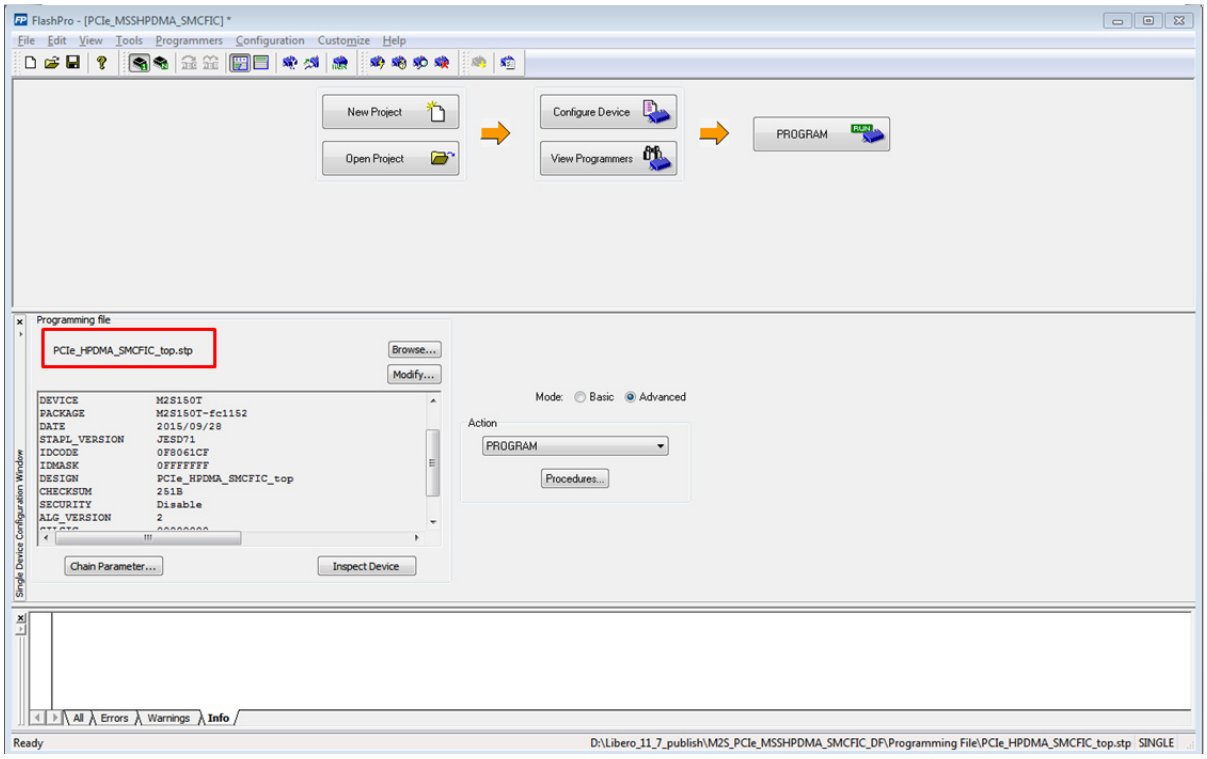
1. Launch the FlashPro software.
2. Click **New Project**. Figure 4 shows the **FlashPro - New Project** dialog.
3. In the **New Project** dialog, enter the project name.

Figure 4 • **FlashPro- New Project Dialog**



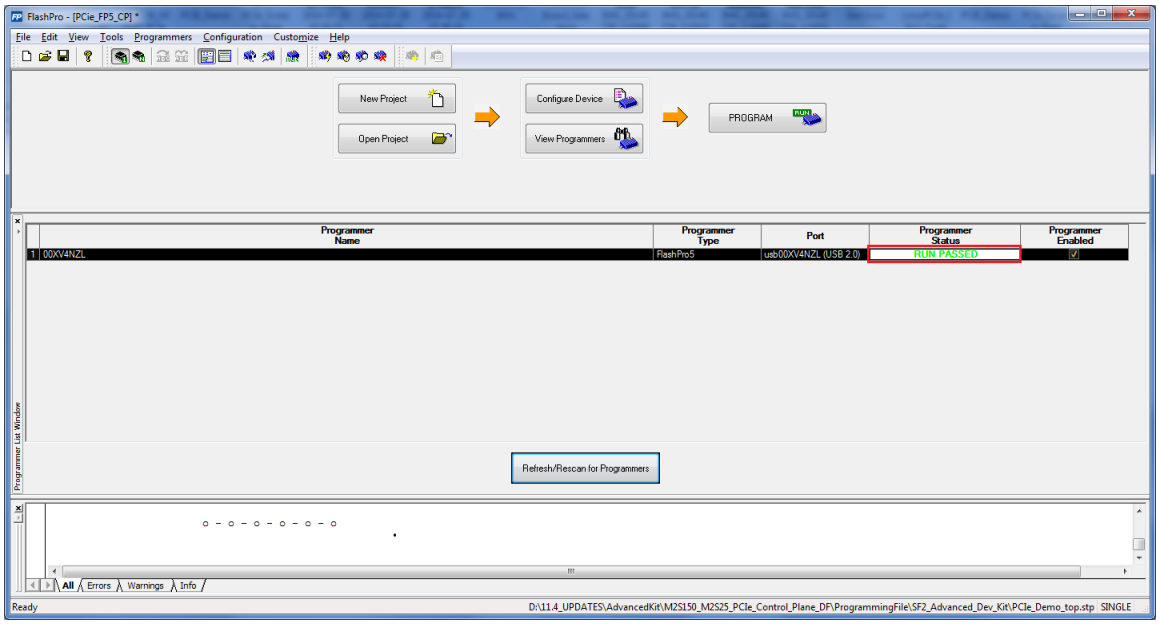
4. Click **Browse** and navigate to the location where it is required to save the project.
5. Select **Single device** as the **Programming mode**.
6. Click **OK** to save the project.
7. Click **Configure Device**.
8. Click **Browse** and navigate to the location where the `PCIe_HPDMA_SMCFIC_top.stp` file is located and select the file. The default location is:
   <download_folder>\M2S_PCIe_MSSHPDMA_SMCFIC_DF\ProgrammingFile
9. Click **Open**. The required programming file is selected and is ready to be programmed in the device.

Figure 5 • **FlashPro Project Configured**



10. Click **PROGRAM** to start programming the device. Wait until you get a message indicating that the program is passed.

Figure 6 • **FlashPro Programming Passed**



2.5.3 Connecting the Kit to the Host PC PCIe Slot

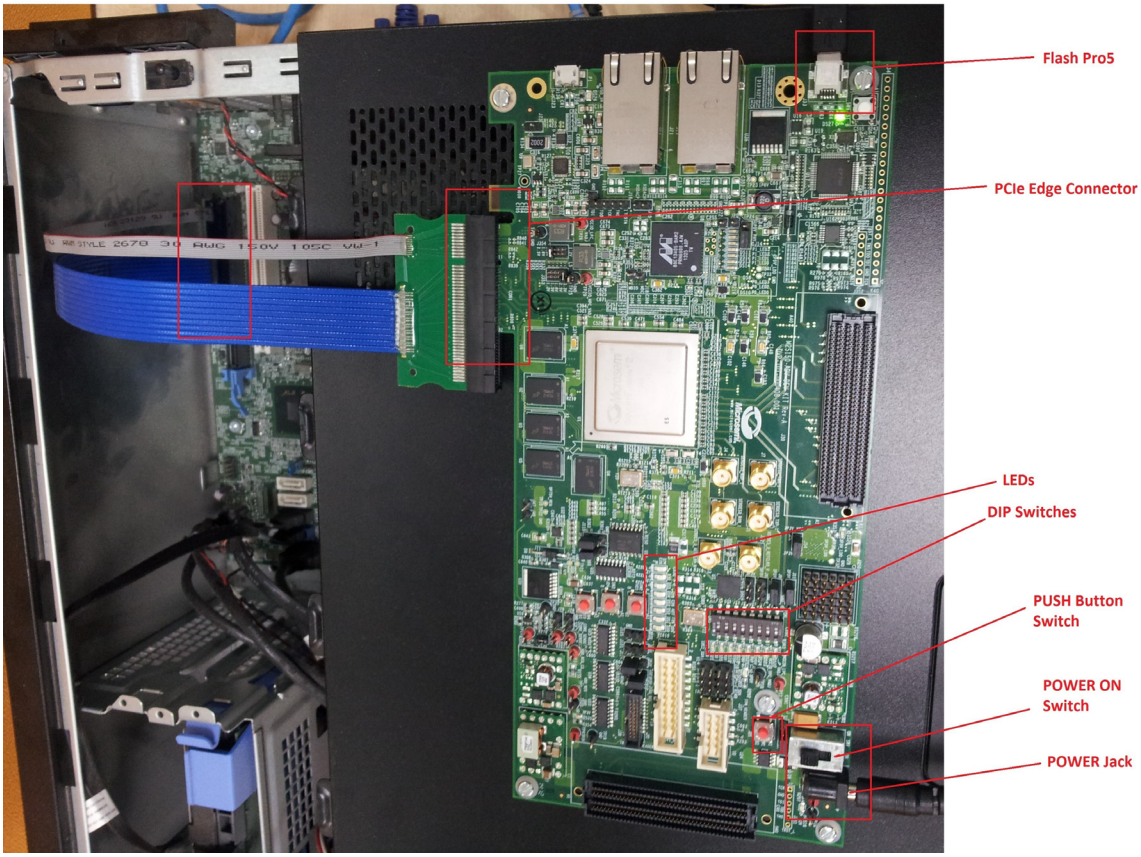
Use the following steps to connect the kit to the host PC PCIe slot:

1. After successful programming, **power off** the SmartFusion2 Advanced Development Kit and **shut down** the host PC.
2. Connect the CON1 - PCIe Edge connector of the SmartFusion2 Advanced Development Kit to the host PC's PCIe slot through the PCI Edge Card Ribbon Cable.

Note: Ensure that the host PC is switched off when plugging the PCIe connector cable to the PCIe slot.

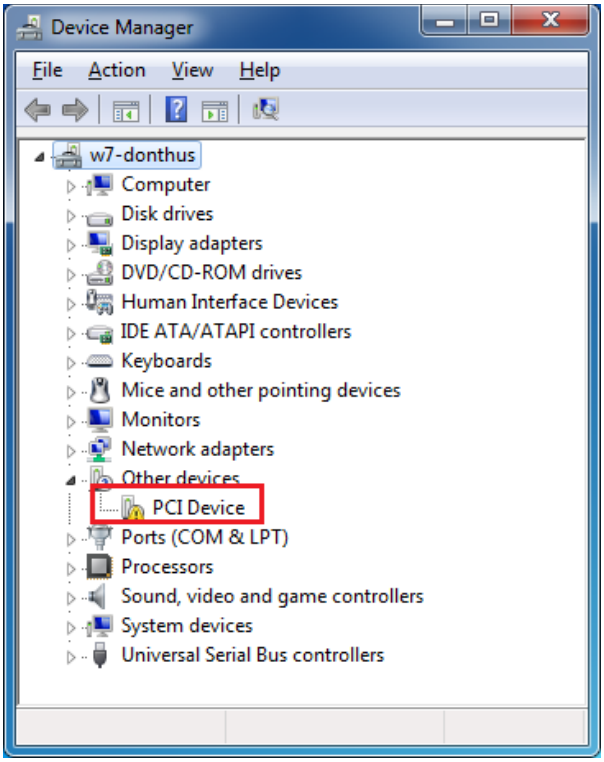
Figure 7 shows the board setup for the host PC in which SmartFusion2 Advanced Development Kit is connected to the host PC PCIe slot.

Figure 7 • **SmartFusion2 Advanced Development Kit Setup**



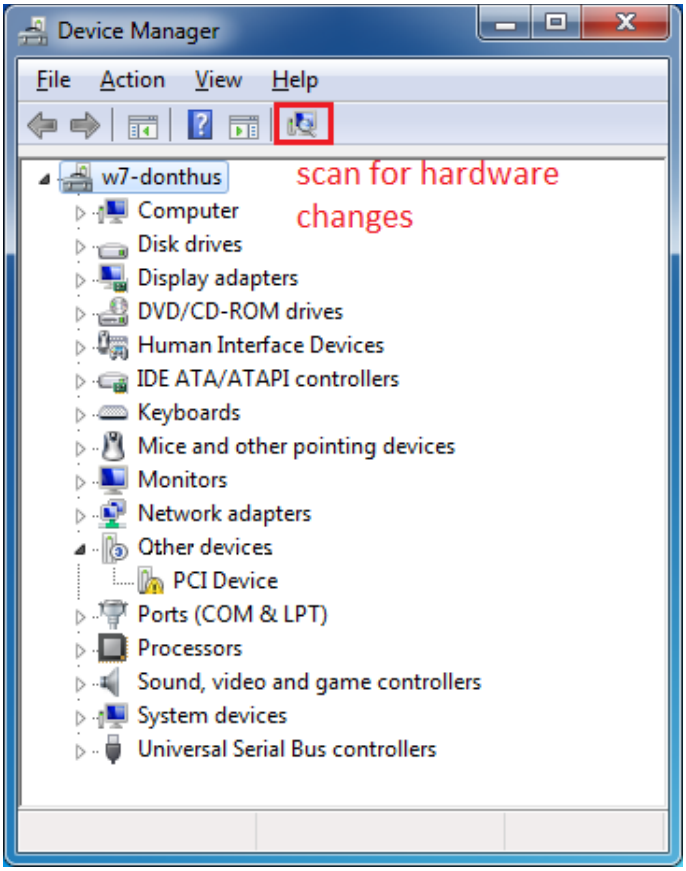
3. Switch on the power supply switch, **SW7**.
4. Power on the host PC and check the **Device Manager** of the host PC for **PCIe Device**. Figure 8 shows the **Device Manager** window.

Figure 8 • **Device Manager - PCIe Device Detection**



5. If the device is not detected, power cycle the SmartFusion2 Advanced Development Kit and click **scan for hardware changes** option in the **Device Manager** window. Figure 9 shows the **scan for hardware changes** option in the **Device Manager.**

Figure 9 • Scan for Hardware Changes Option in the Device Manager Window



Note: If the device is still not detected, check whether or not the BIOS version in the host PC is the latest, and the PCIe is enabled in the host PC BIOS.

2.5.4 Drivers Installation

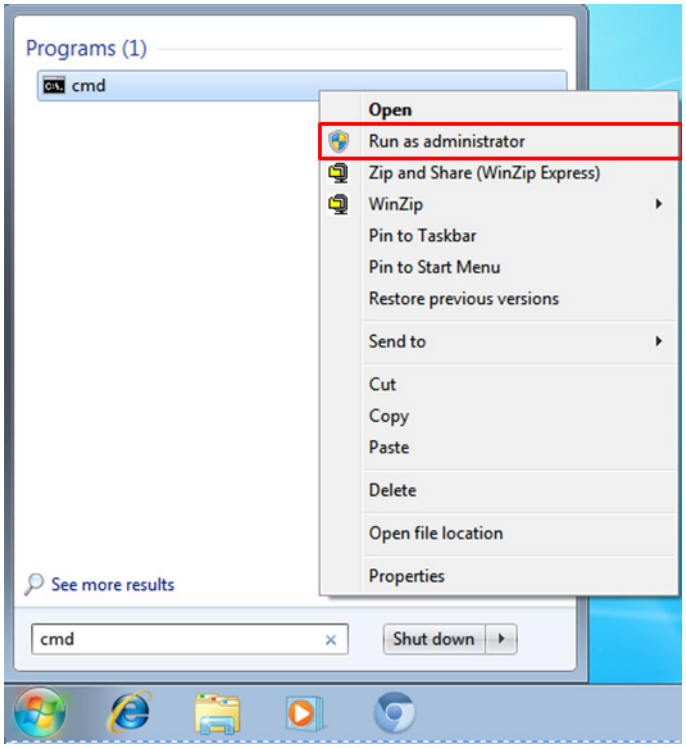
The PCIe Demo uses a driver framework provided by Jungo WinDriverPro. To install the PCIe drivers on the host PC for SmartFusion2 Advanced Development Kit, run the following steps:

1. Extract the PCIe_Demo.rar to C:\ drive. The PCIe_Demo.rar is located in the provided design files: *M2S_PCIe_MSSHPDMA_SMCFIC_DF\Drivers_64bitOS\ PCIe_Demo.rar*
2. Run the batch file *C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat* to install the PCIe drivers for the SmartFusion2 device.

Note: Installing these drivers require host PC Administration rights.

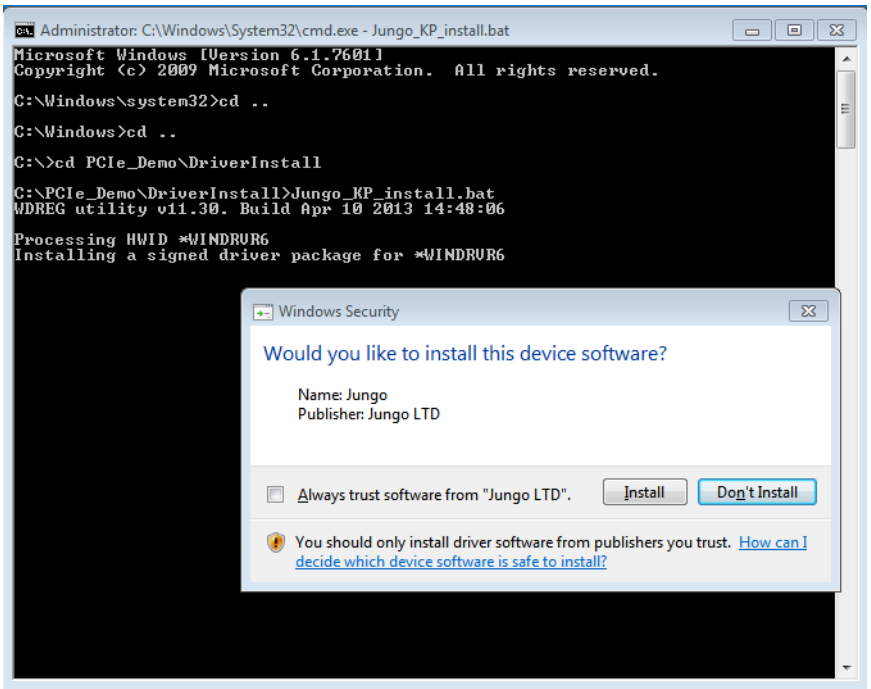
3. To run the batch file **C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat**, open command prompt and select **Run as administrator**, as shown in Figure 10.

Figure 10 • **Command Prompt**



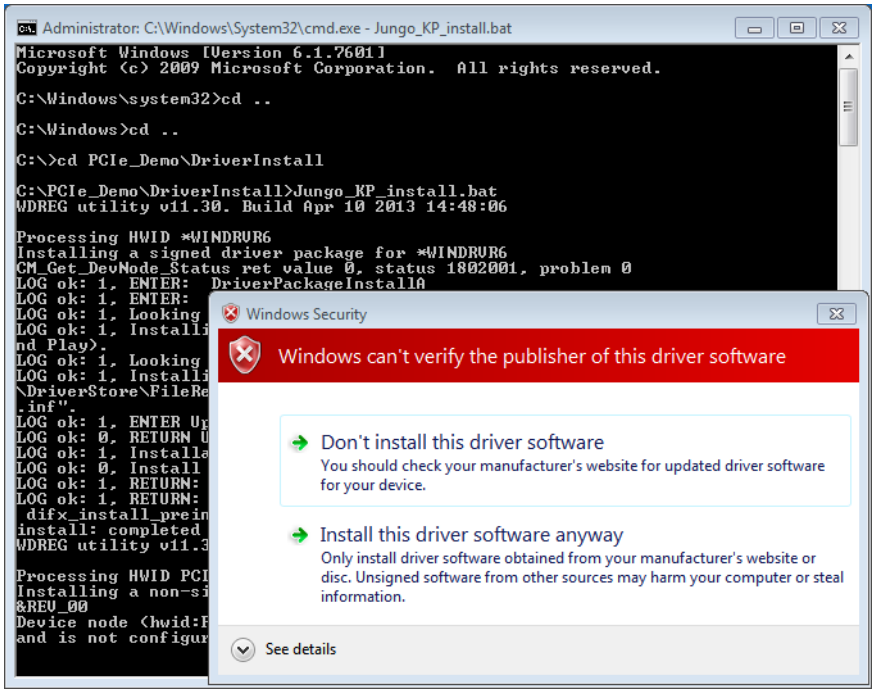
4. Navigate to **C:\ Drive and execute Jungo_KP_install.bat** in command prompt and press **Enter**. A Windows Security dialog box is displayed, as shown in Figure 11. Click **Install**.

Figure 11 • **Jungo Driver Installation**



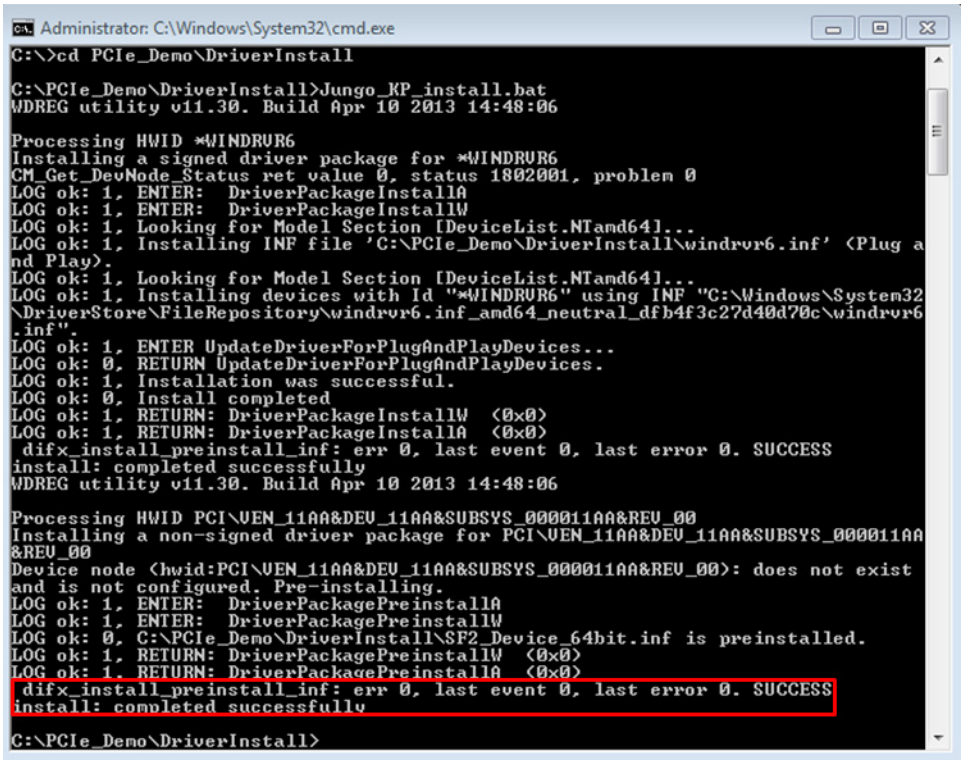
5. If the Windows Security dialog box appears asking whether to install or not, click **Install this driver software anyway**, as shown in Figure 12.

Figure 12 • **Windows Security**



A message **SUCCESS install: completed successfully** is displayed, as shown in Figure 13.

Figure 13 • **Completed Successfully Message**



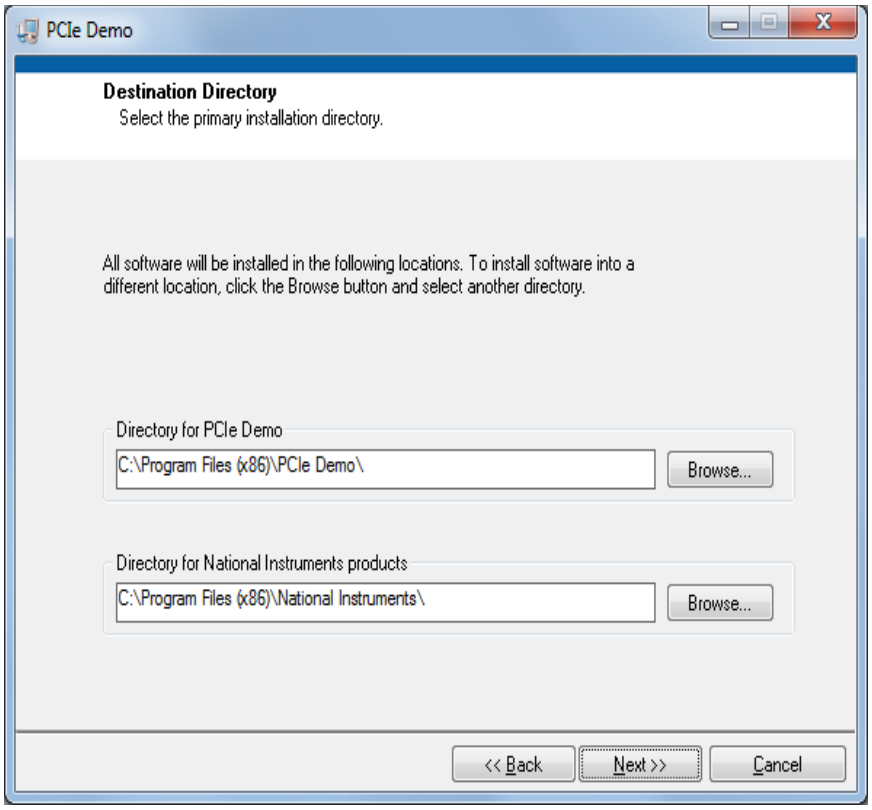
2.5.5 PCIe_Demo Application

The PCIe_Demo application is a simple graphic user interface that runs on the host PC to communicate with the SmartFusion2 PCIe endpoint device. It provides PCIe link status, driver information and demo controls. The PCIe_Demo application invokes the PCIe driver installed on the host PC and provides commands to the driver according to the selection made.

To install the GUI, use the following steps:

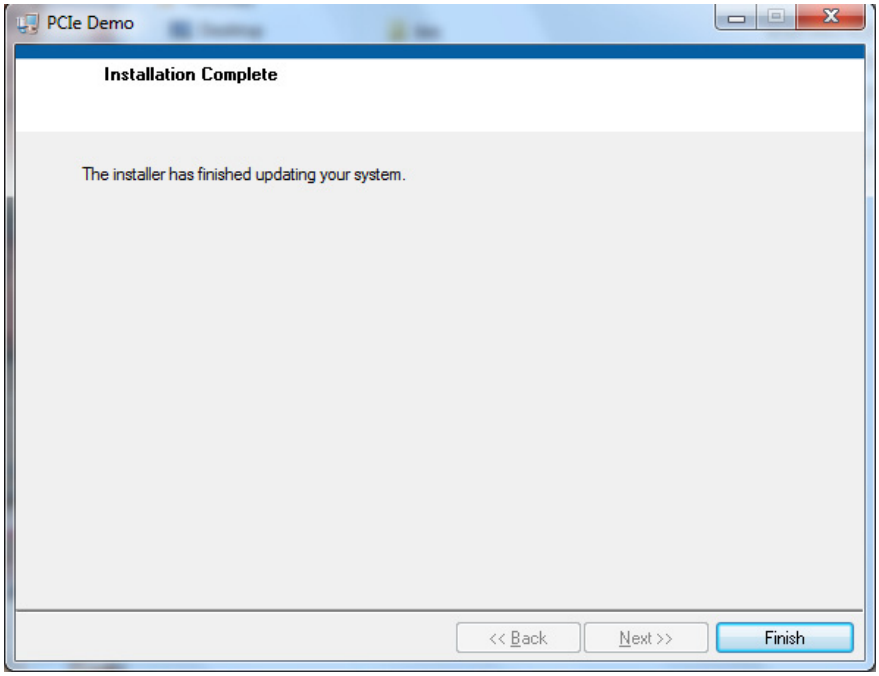
1. Download the PCIe demo GUI installer from *http://soc.microsemi.com/download/rsc/?f=PCIe_Demo_GUI_Installer*
2. Extract the **PCIe_Demo_GUI_Installer.rar**.
3. Double-click the **setup.exe** in the provided GUI installation *(PCIe_Demo_GUI_Installer/setup.exe)*. Apply default options as shown in Figure 14.

Figure 14 • **GUI Installation**



4. Click **Next** and **Finish** to complete the installation. Figure 15 shows the Successful GUI Installation window.

Figure 15 • **Successful Installation of GUI**

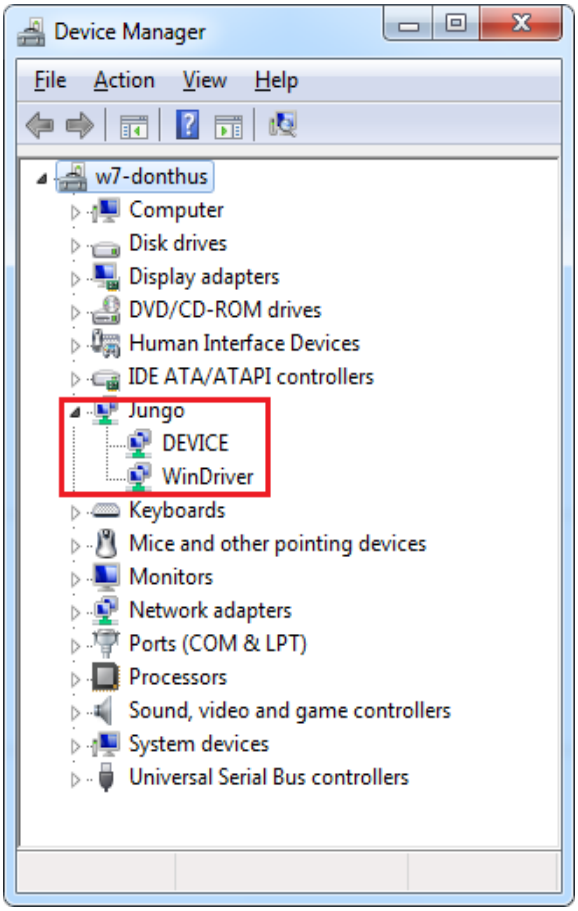


5. Shutdown the host PC.
6. Power cycle the SmartFusion2 Advanced Development Kit board.
7. Restart the host PC.

2.6 Running the Design

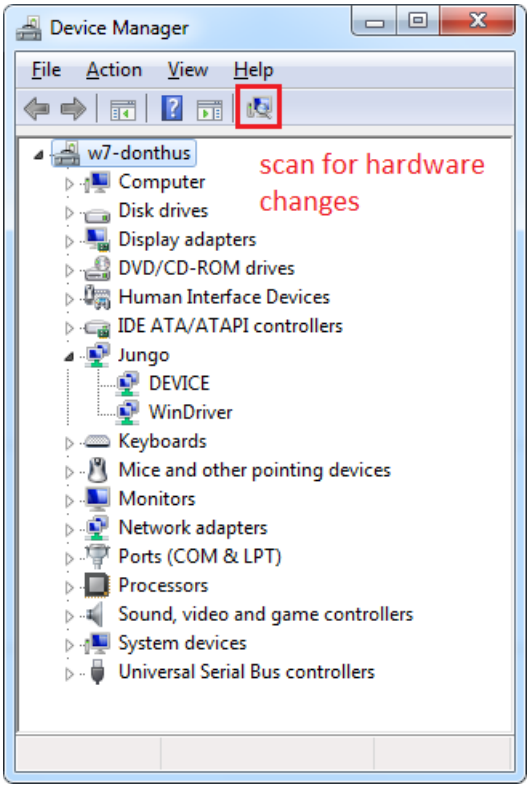
1. Check the host PC **Device Manager** for the drivers. Figure 16 shows the **Device Manager** window highlighting the Jungo drivers installed.

Figure 16 • **Device Manager - PCIe Device Detection**



2. If the device is not detected, power cycle the SmartFusion2 Advanced Development Kit and click **scan for hardware changes** in the **Device Manager** window.

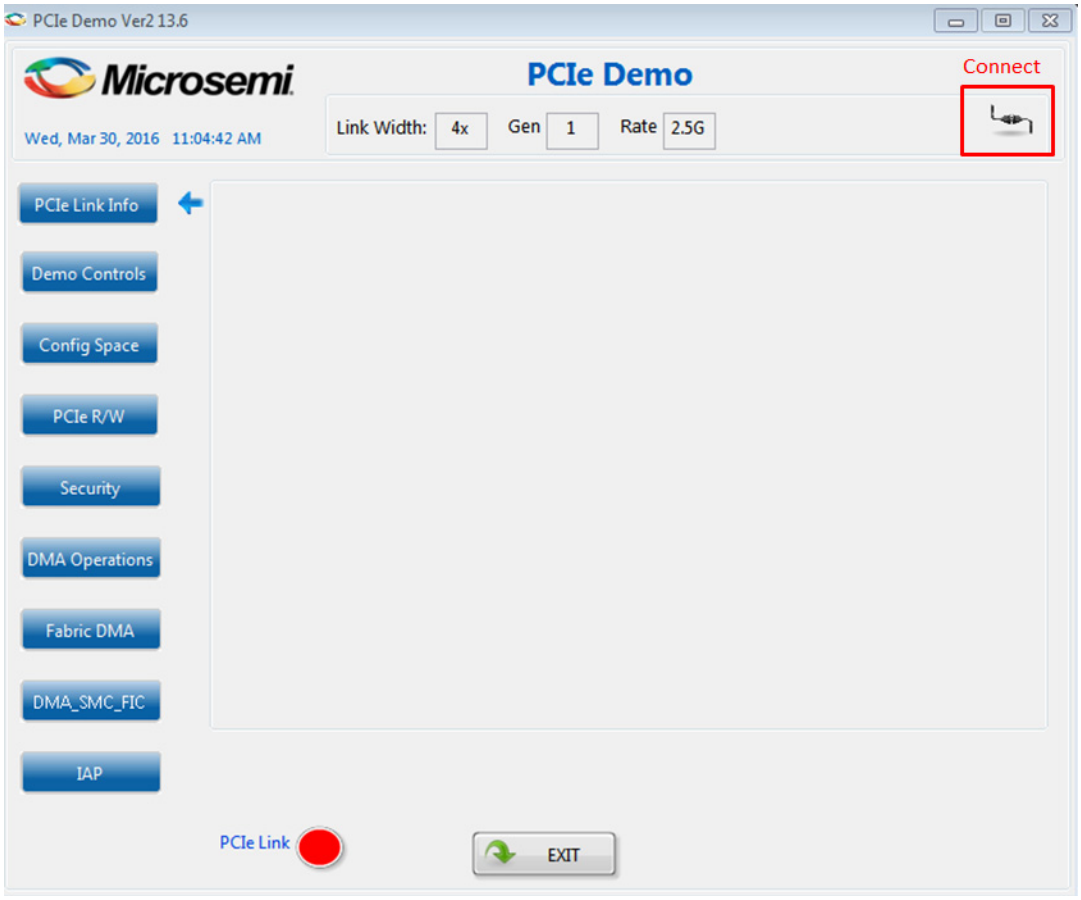
Figure 17 • **Scan for Hardware Changes Option in the Device Manager Window**



Note: If a warning appears on the DEVICE or WinDriver in the **Device Manager** window, uninstall the drivers and start from Step 1 of driver installation.

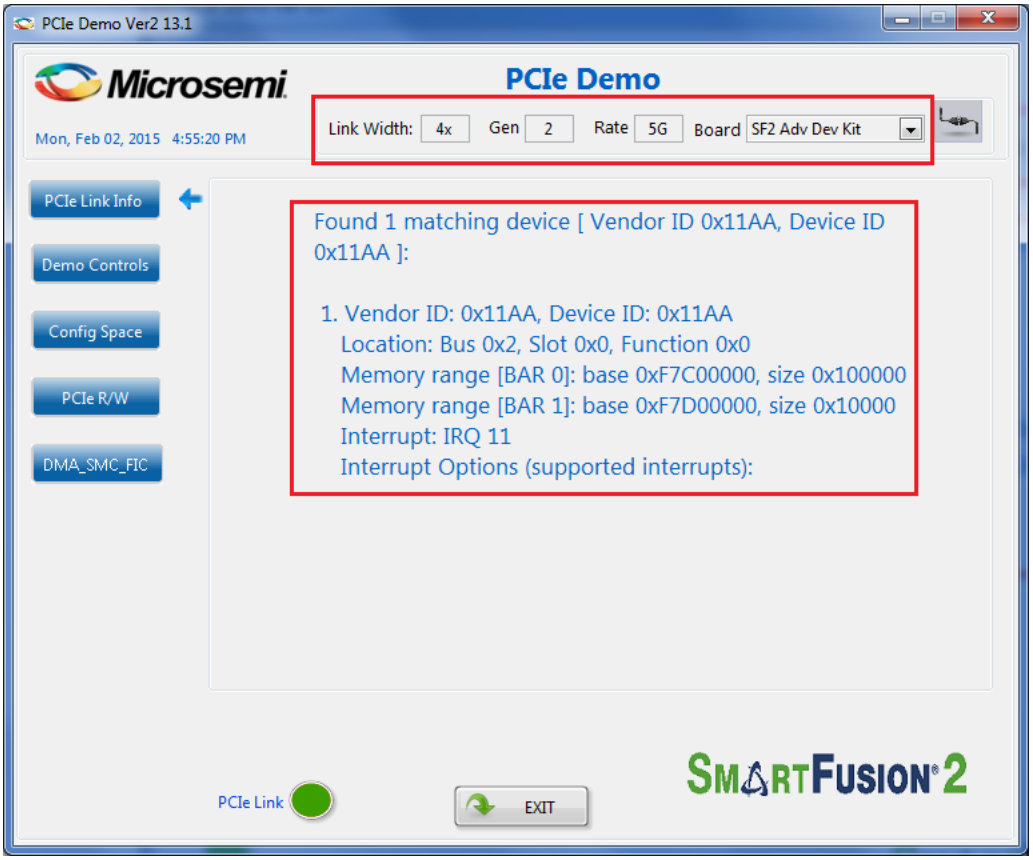
3. Invoke the PCIe_Demo application from **ALL Programs > PCIe Demo > PCIe Demo GUI**. Figure 18 shows the PCIe_Demo launch window.

Figure 18 • **PCIe_Demo Application**



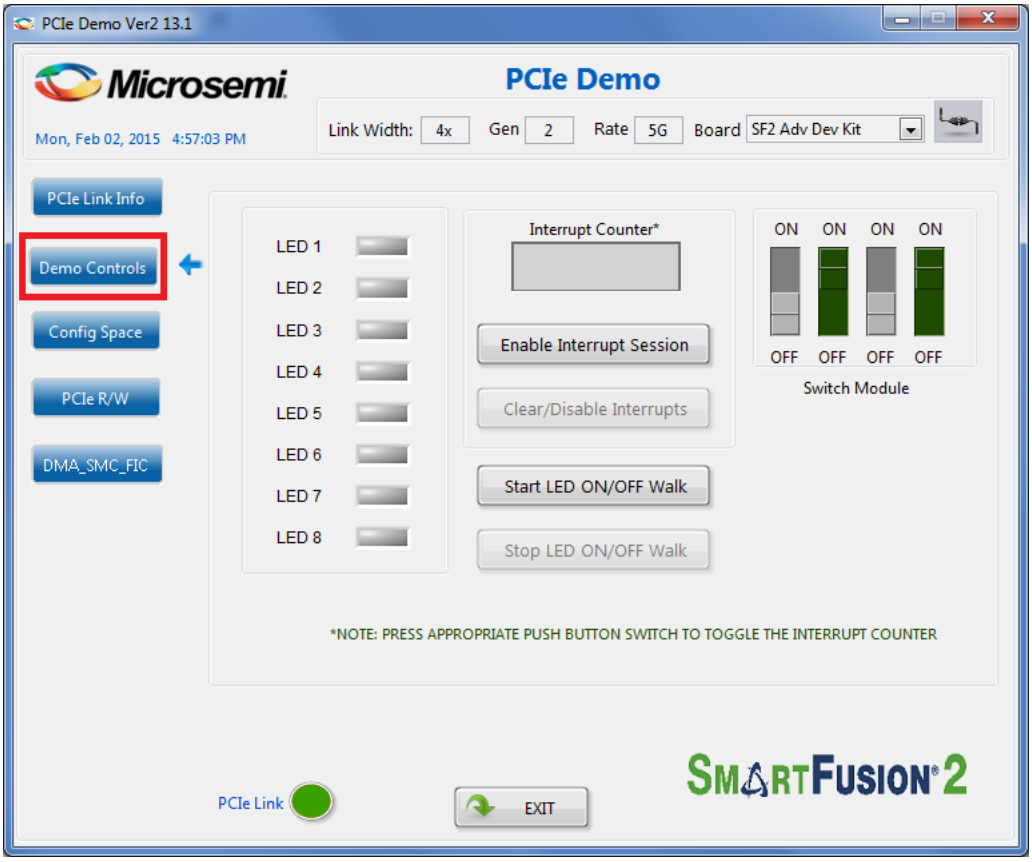
4. Click **Connect** at top right corner of the PCIe_Demo application. The application detects and displays the connected kit, demo design and PCIe link. Figure 19 shows the sample messages after the connection is established.

Figure 19 • **PCIe Device Information**



5. Click **Demo Controls** to display the LEDs options and DIP switch positions. Figure 20 shows the LED options and DIP switch positions in **Demo Controls**.

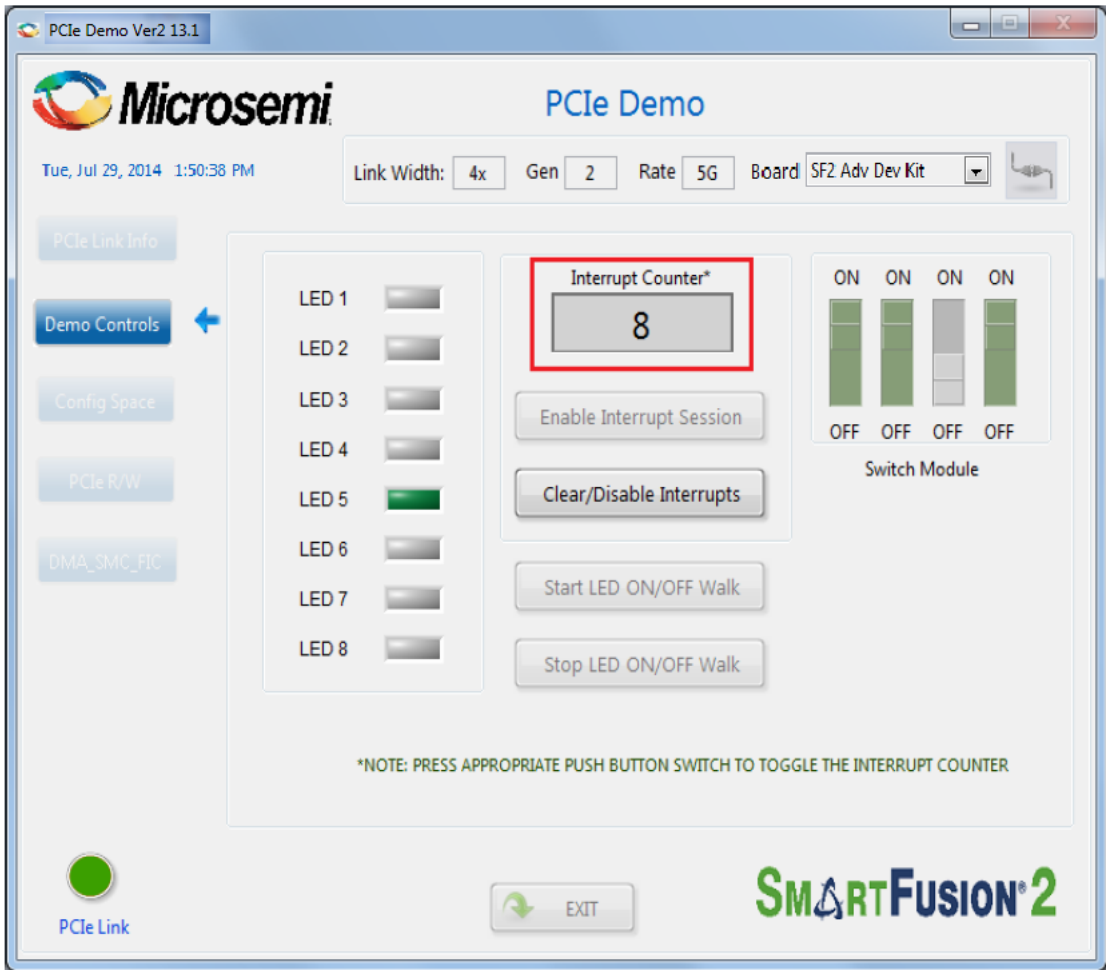
Figure 20 • **LED Options and DIP Switch Positions in Demo Controls**



6. Click **LEDs** to switch **ON** or **OFF** the LEDs on the board.
7. Click **Start LED ON/OFF Walk** to blink the LEDs on the board.
8. Click **Stop LED ON/OFF Walk** to stop the LEDs blinking.
9. Change the DIP switch positions on the board and observe the same reflected in the switches of the **Switch Module** of the PCIe_Demo application.
10. Click **Enable Interrupt Session** to enable the PCIe interrupt.

11. Press the push button, SW1 on the SmartFusion2 Advanced Development Kit board. Observe the interrupt count on the **Interrupt Counter** field in the PCIe_Demo application. Figure 21 shows the **Interrupt Counter** field in PCIe_Demo application.

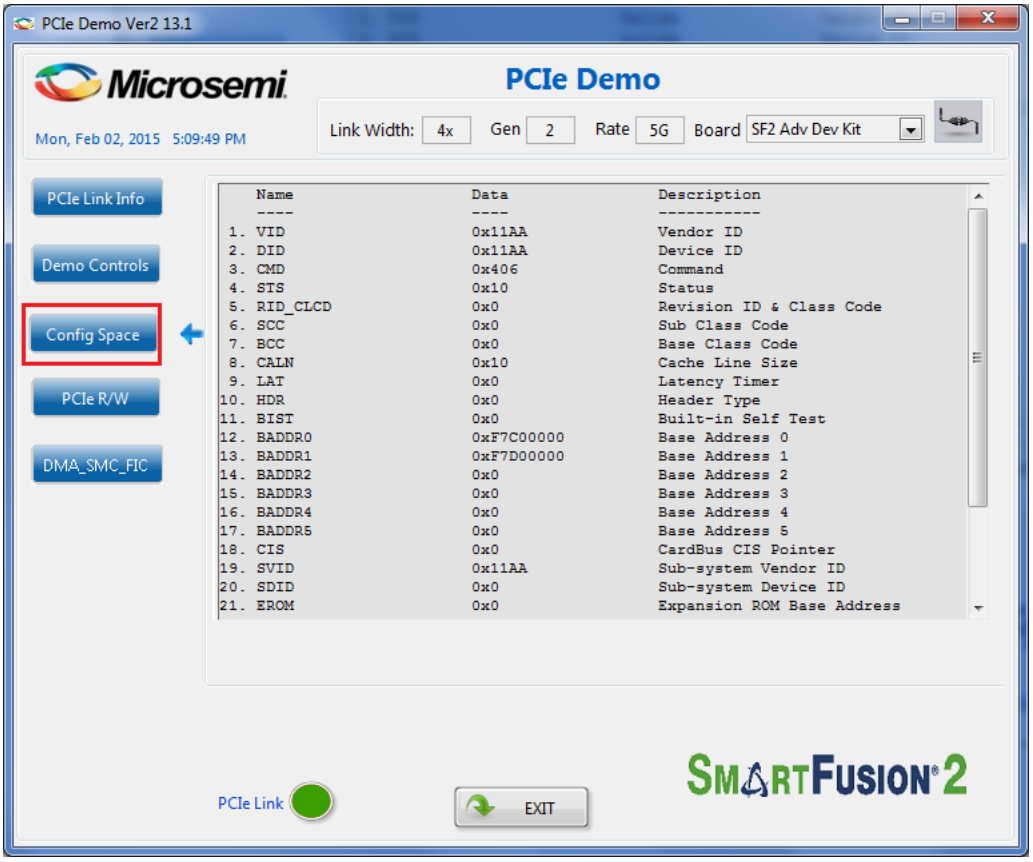
Figure 21 • **Interrupt Counter Field in PCIe_Demo Application**



12. Click **Clear/Disable Interrupts** to clear or disable the PCIe interrupts.

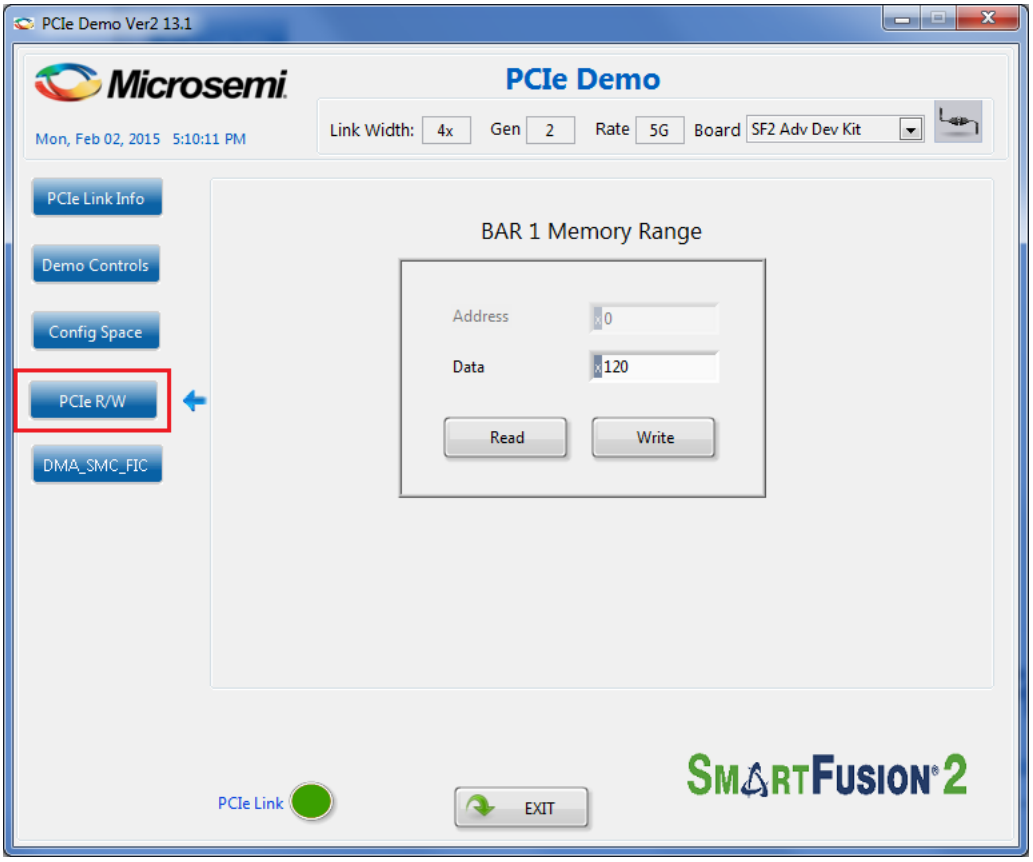
13. Click **Config Space** to see the details about the PCIe configuration space. Figure 22 shows the PCIe configuration space details.

Figure 22 • **PCIe Configuration Space Details**



14. Click **PCIe R/W** to execute read and write to a 32-bit scratchpad register through BAR1 space. Figure 23 shows the PCIe R/W panel.

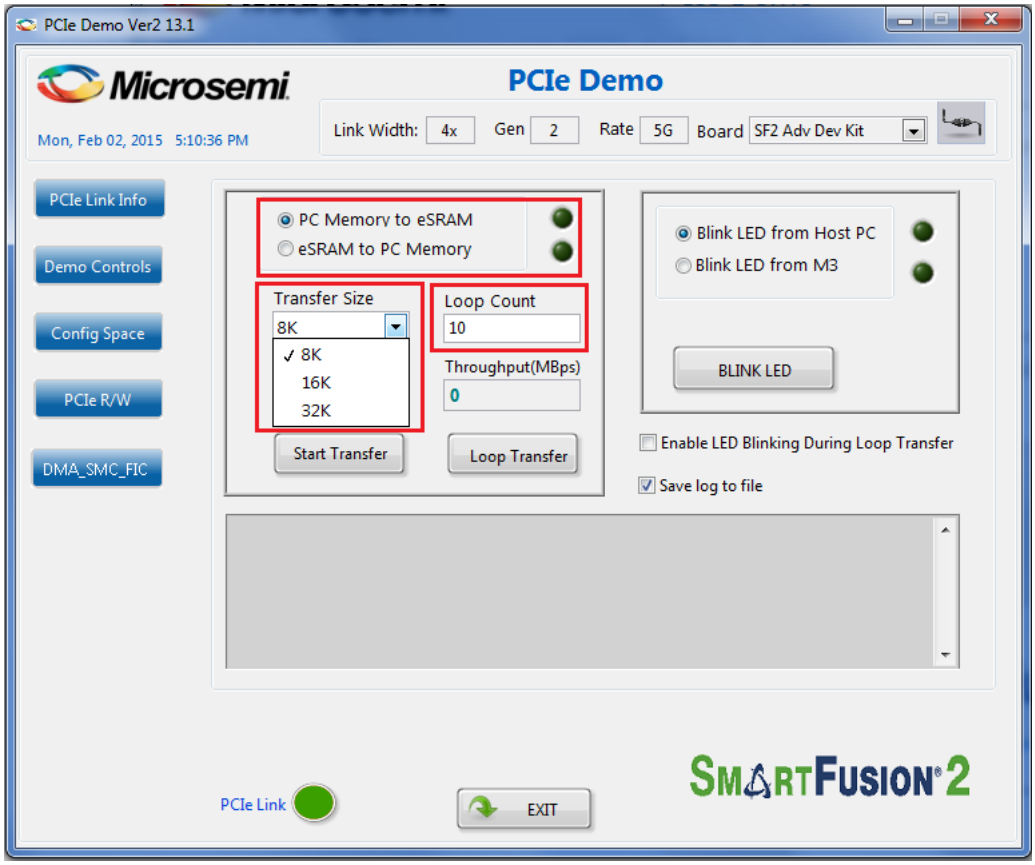
Figure 23 • **Read and Write to Scratchpad Register**



15. Click **DMA_SMC_FIC** to run the DMA operations. Two types of DMA transactions are possible:
    - PC Memory to eSRAM
    - eSRAM to PC Memory

For each operation, **Transfer Size** can be selected from 8 KB to 32 KB as shown in Figure 24. It also has a **Loop Count** field to run the DMA operation in loop. The Burst Size (TLP size) is fixed to 32 bytes to match the DDR bridge buffer size. The actual size of the PCIe packet is the size of a single AXI burst transfer which is 32 bytes.

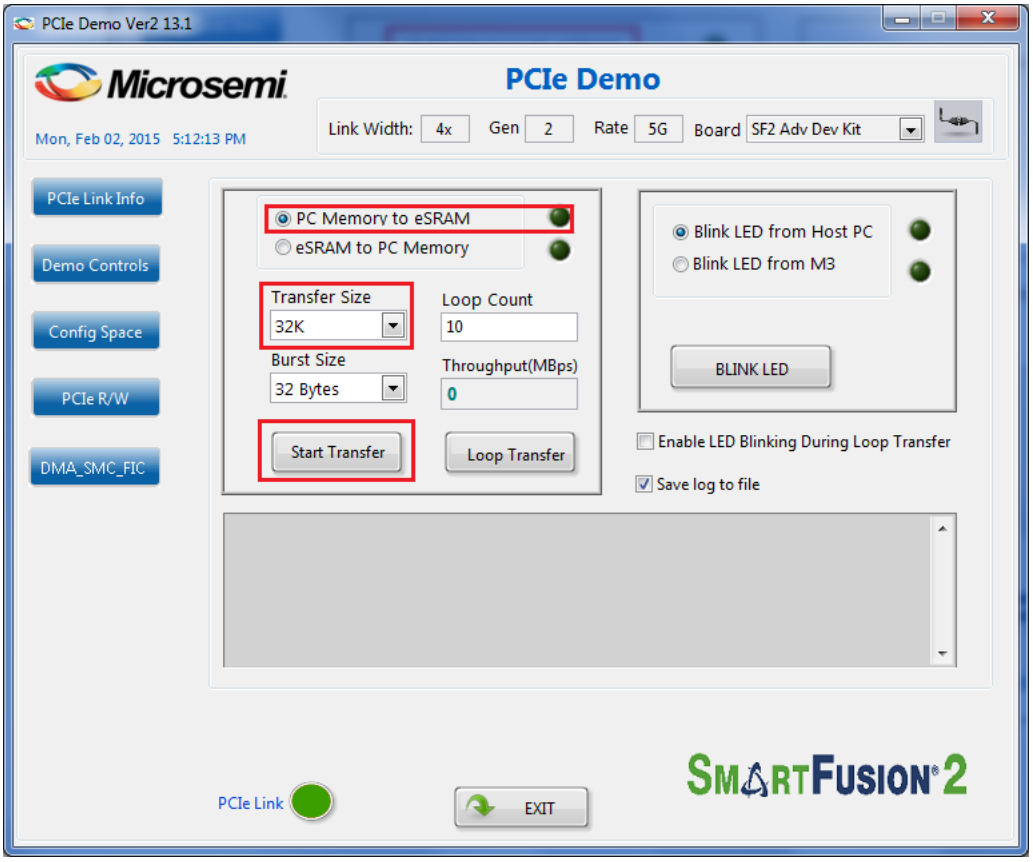
Figure 24 • **Fabric DMA Controls**



16. Select the type of DMA transfer as PC Memory to eSRAM and select 32 K **Transfer Size**.

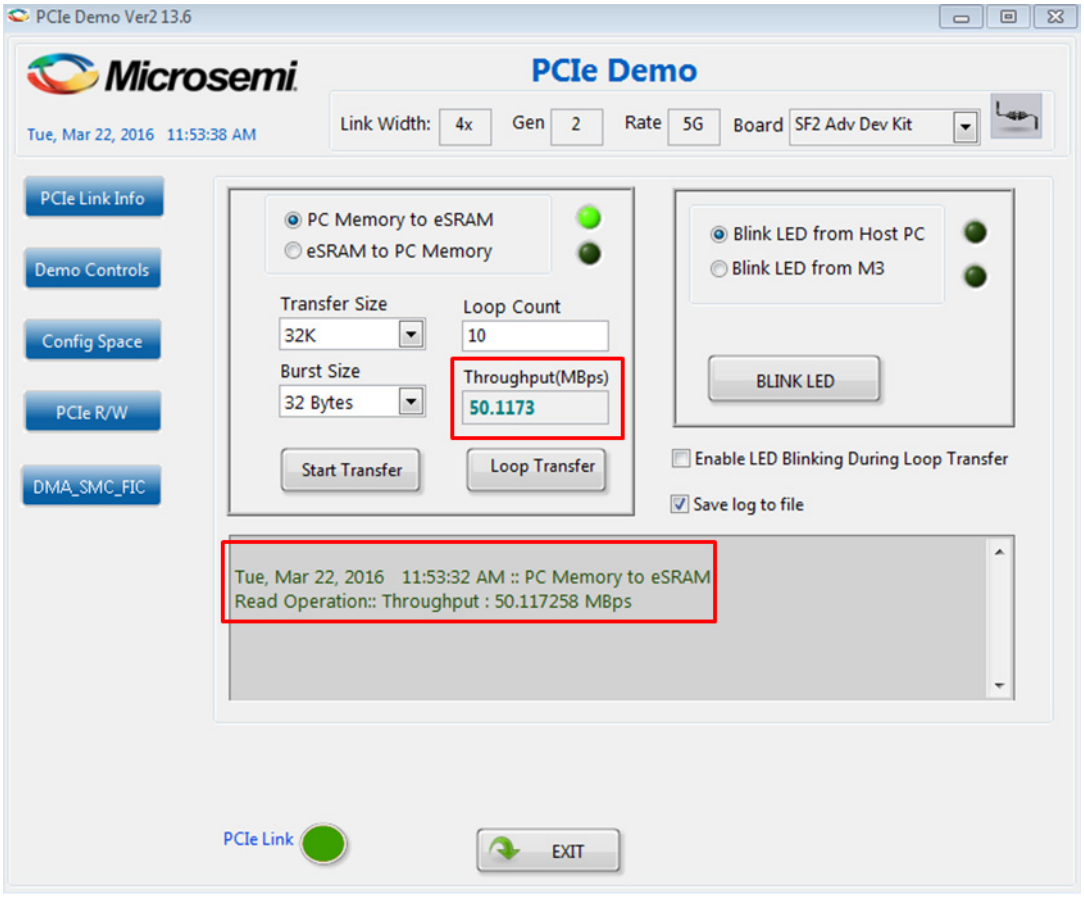
17. Click **Start Transfer**. Figure 25 shows the DMA Transactions between the host PC memory and eSRAM.

Figure 25 • **DMA Transactions between Host PC Memory and eSRAM**



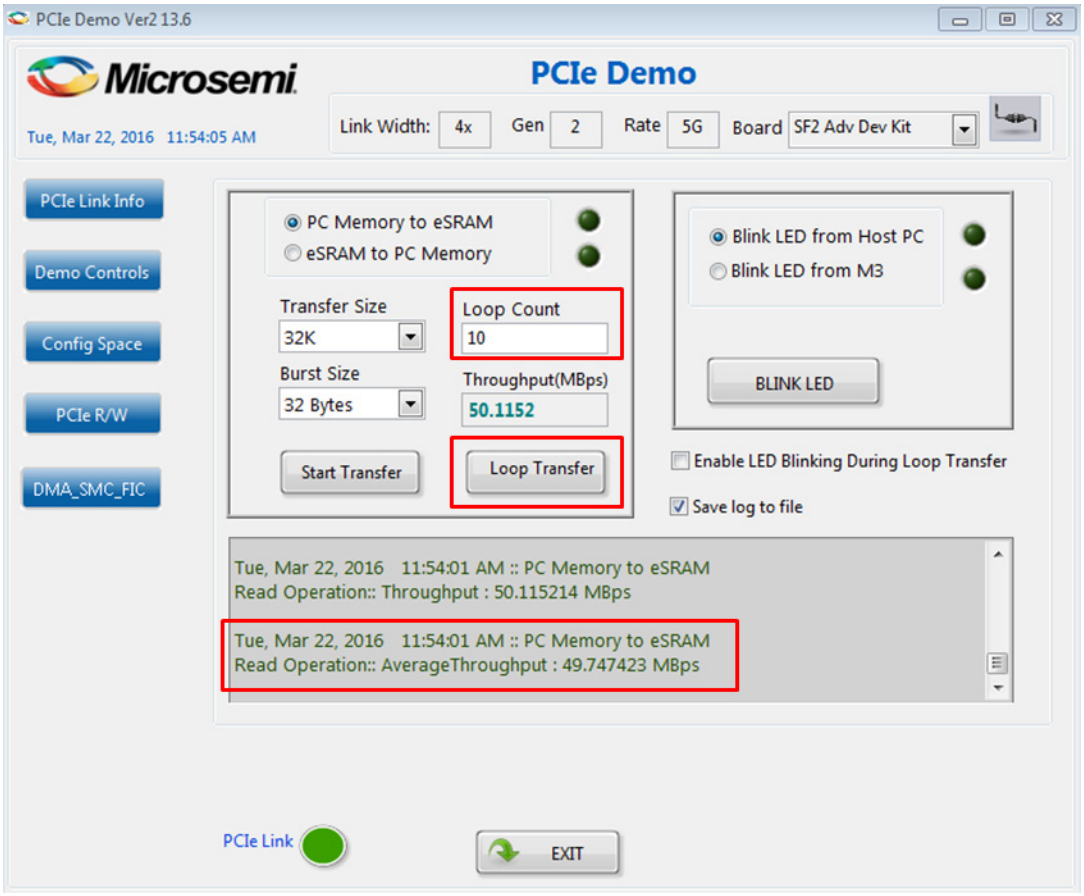
18. After completion of data transfer, the throughput is displayed. Figure 26 shows the throughput in the DMA transactions from the host PC to eSRAM.

Figure 26 • **Throughput in DMA Transactions from Host PC to eSRAM**



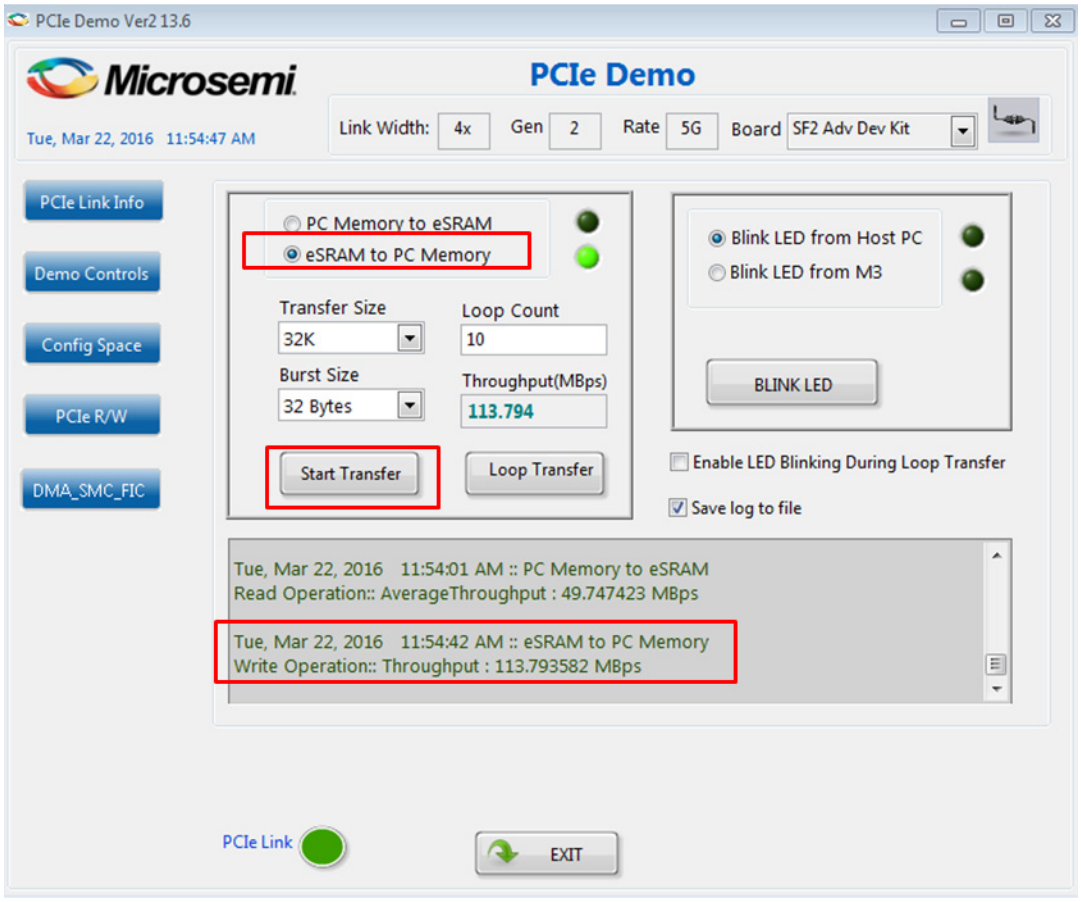
19. Enter **10** in the **Loop Count** field and click **Loop Transfer** to perform 10 sequential DMA transactions. After completion of data transfer, the PCIe_Demo application displays the throughput. Figure 27 shows the throughput in DMA transactions from host PC to eSRAM. The average throughput is also logged. The log file is stored in the host PC at *C:\PCIe_Demo\DriverInstall*.

Figure 27 • **Throughput in the DMA Transactions from the Host PC to eSRAM**



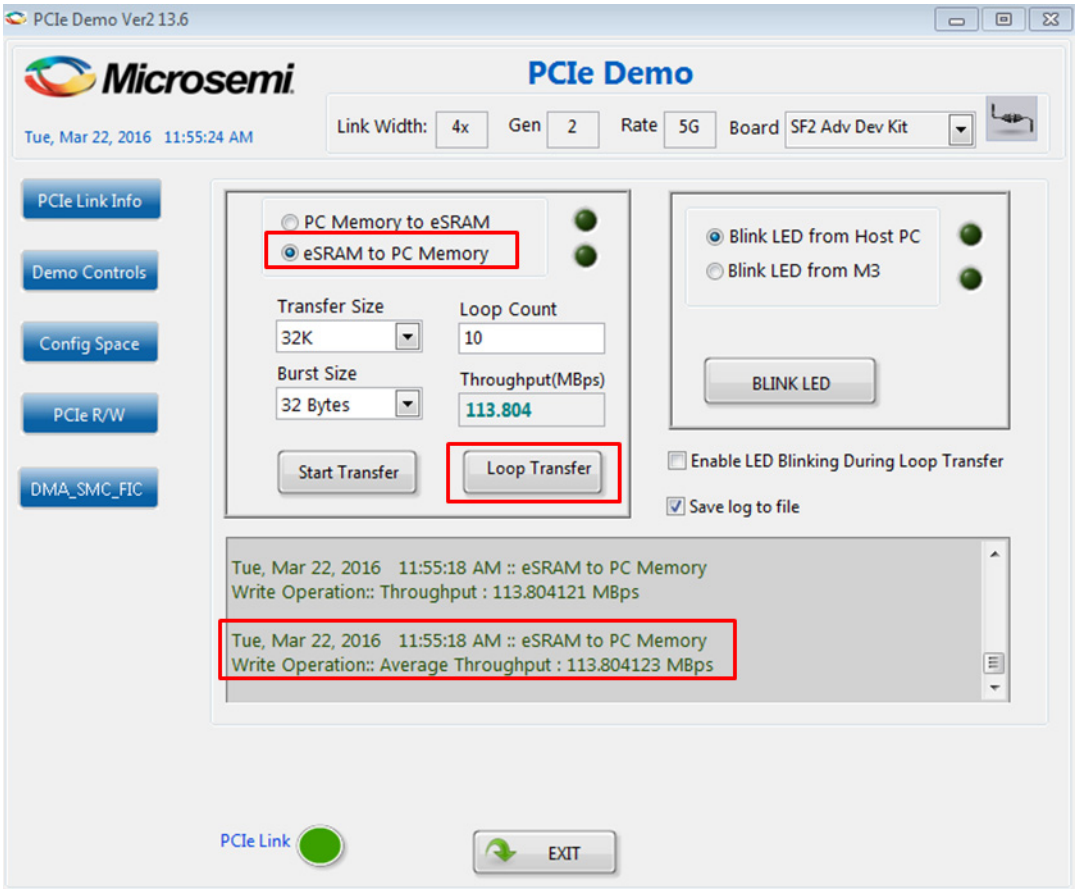
20. Select the type of DMA transfer as eSRAM to PC Memory and select 32 K Transfer Size. Click **Start Transfer** to perform a single DMA transaction. After completion of data transfer, the throughput is displayed. Figure 28 shows the throughput in the DMA transactions from eSRAM to the host PC.

Figure 28 • **Throughput in the DMA Transactions from eSRAM to the Host PC**



21. Enter **10** in the **Loop Count** field and click **Loop Transfer** to perform 10 repeated DMA transactions. After completion of data transfer, the throughput is displayed. Figure 29 shows the throughput in the DMA transactions from eSRAM to the host PC.

Figure 29 • **Throughput in the DMA Transactions from eSRAM to the Host PC**



The LEDs on the board can be blinked in parallel to the DMA operations by using the LED controls on the right side of GUI. The **Enable LED Blinking During Loop Transfer** check box need to be selected to do the LED blinking from host PC and DMA transfers.

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

2.7 Summary

This demo shows how to implement a PCIe Data Plane Design using MSS HPDMA and SMC_FIC. Data transfer occurs between PCIe and SmartFusion2 eSRAM. Throughput for data transfers is dependent on the host PC system configuration, type of PCIe slots used. Table 3 shows the throughput values observed on the HP 230 PCIe slot 2.

Table 3 • **Throughput Summary**

DMA Transfer Type	DMA Transfer Size	Throughput (Mbps)			
		Gen 1		Gen 2	
		Single Transfer	Loop Transfer	Single Transfer	Loop Transfer
Host PC Memory to eSRAM	8 KB	36	36	50	49
	16 KB	36	36	50	49
	32 KB	36	36	50	49
eSRAM to host PC Memory	8 KB	92	92	113	113
	16 KB	92	92	113	113
	32 KB	92	92	113	113

3 Appendix: Register Details

Table 4 shows the registers used to interface with the PCIe MSS HPDMA SMC_FIC design. These registers are in BAR1 address space.

Table 4 • **Register Details**

Register Name	BAR Space	Register Address	Description
PC_BASE_ADDR	BAR 1	0x8028	Host PC memory base address provided by the driver.
DMA_DIR	BAR 1	0x8008	DMA direction **Direction** / **Register value** eSRAM to PCIe / 0x11AA1111 PCIe to eSRAM / 0x11AA2222
DMA_SIZE	BAR 1	0x8010	Size of DMA transfer **Size** / **Register value** 8KB / 0x2000 16KB / 0x4000 32KB / 0x8000
DMA_CLK_CYCLES	BAR 1	0x8018	Number of clock cycles taken to complete the DMA transfer.
DMA_STATUS	BAR 1	0x8020	1: DMA transfer completed 0: DMA transfer is not completed
BLINK_M3	BAR 1	0x8030	Blinks the LEDs from Cortex-M3 if the register value is 0x11AA0F0F.
RW_REG	BAR 1	0x0	Scratchpad register for PCIe R/W
LED_CTRL[7:0]	BAR 0	0x13088	LEDs control register
SWITCH_STATUS[11:8]	BAR 0	0x13080	DIP switch status

4 Revision History

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

Revision	Changes
Revision 5 (April 2016)	Updated the document for Libero v11.7 software release (SAR 78030).
Revision 4 (October 2015)	Updated the document for Libero v11.6 software release (SAR 71687).
Revision 3 (March 2015)	Updated the document for Libero v11.5 software release (SAR 65319).
Revision 2 (August 2014)	Updated the document for Libero v11.4 software release (SAR 59780).
Revision 1 (March 2014)	Initial release.

5 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.

5.1 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

5.2 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.

5.3 Technical Support

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

5.4 Website

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

5.5 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.

5.5.1 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.

5.5.2 My Cases

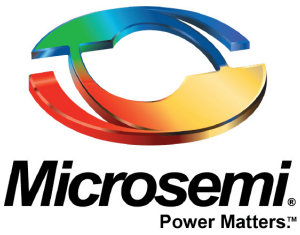
Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

5.5.3 Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Visit About Us for sales office listings and corporate contacts.

5.6 ITAR Technical Support

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



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

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif, and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

50200535-5/04.16