IGLOO2 FPGA In-Application Programming Using PCIe Interface

DG0681 Demo Guide

Microsemi
Power Matters.™

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
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center 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.

50200681-1/5.16

1 Revision History

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

Revision	Changes
Revision 1 (May 2016)	Initial release.

Contents

Figures

Tables

2 Preface

2.1 Purpose

This demo is for IGLOO®2 field programmable gate array (FPGA) devices. It provides instructions on how to use the reference design.

2.2 Intended Audience

This demo guide is intended for:

- FPGA designers
- System-level designers

2.3 References

See the following web page for a complete and up-to-date listing of IGLOO2 device documentation:
http://www.microsemi.com/products/fpga-soc/fpga/igloo2-fpga#documentation

The following documents are referred in this demo guide:

- *UG0451: IGLOO2 FPGA and SmartFusion2 SoC FPGA Programming User Guide*
- *UG0450: SmartFusion2 SoC FPGA and IGLOO2 FPGA System Controller User Guide*

3 IGLOO2 FPGA In-Application Programming Using PCIe

3.1 Introduction

In-application programming (IAP) is an IGLOO2 device programming feature used for reprogramming the device for design iterations and field upgrades. By using the IAP feature, the application can re-program the flash components of the IGLOO2 devices. The IGLOO2 devices support IAP using a number of different interfaces. In this demo design, PCIe is used to transfer the new programming data. This document describes how to program the IGLOO2 devices through the PCIe interface and use the IGLOO2 Core system services IP.

Note: For more information about different programming modes supported by IGLOO2 FPGAs, see the *UG0451: IGLOO2 FPGA and SmartFusion2 SoC FPGA Programming User Guide*. For more information about system controller programming services, see the *UG0450: SmartFusion2 SoC FPGA and IGLOO2 FPGA System Controller User Guide*.

3.2 Design Requirements

Table 1 • **Design Requirements**

Design Requirements	Description
Hardware Requirements	
IGLOO2 Evaluation Kit: • 12 V adapter (included with the kit) • USB A to Mini-B cable (included with the kit)	IAP is only supported on Rev C or higher versions of IGLOO2. This reference design is applicable for Rev 1 or higher silicon versions of the IGLOO2 010 device. **The existing Evaluation kits may have the Rev 0 device and the silicon needs to be replaced to Rev 1 or higher**.
Host PC or Laptop	Windows 7 64-bit Operating System
Software Requirements	
Libero® System-on-Chip (SoC)	v11.7
FlashPro programming software	v11.7
Host PC Drivers (included with the design files)	–
PCIe Demo application	PCIe Demo GUI Installer

3.3 Demo Design

3.3.1 Introduction

The demo design files are available for download from the following path in the Microsemi website: *http://soc.microsemi.com/download/rsc/?f=m2gl_dg0681_liberov11p7_df*

The design files include:

- Libero SoC software project
- PCIe driver
- STAPL programming file
- Sample programming files
- Readme file

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

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

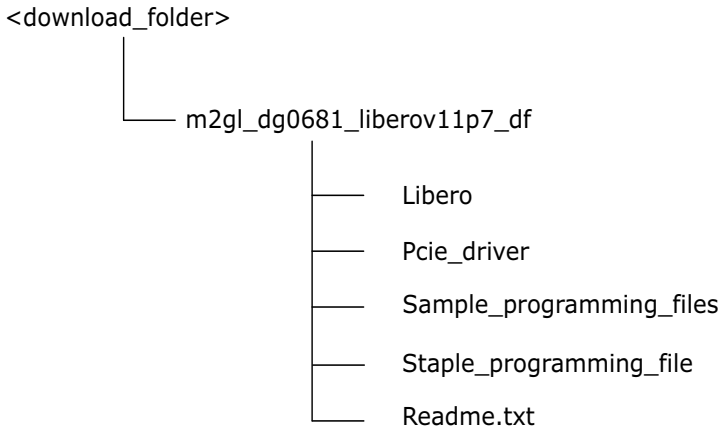
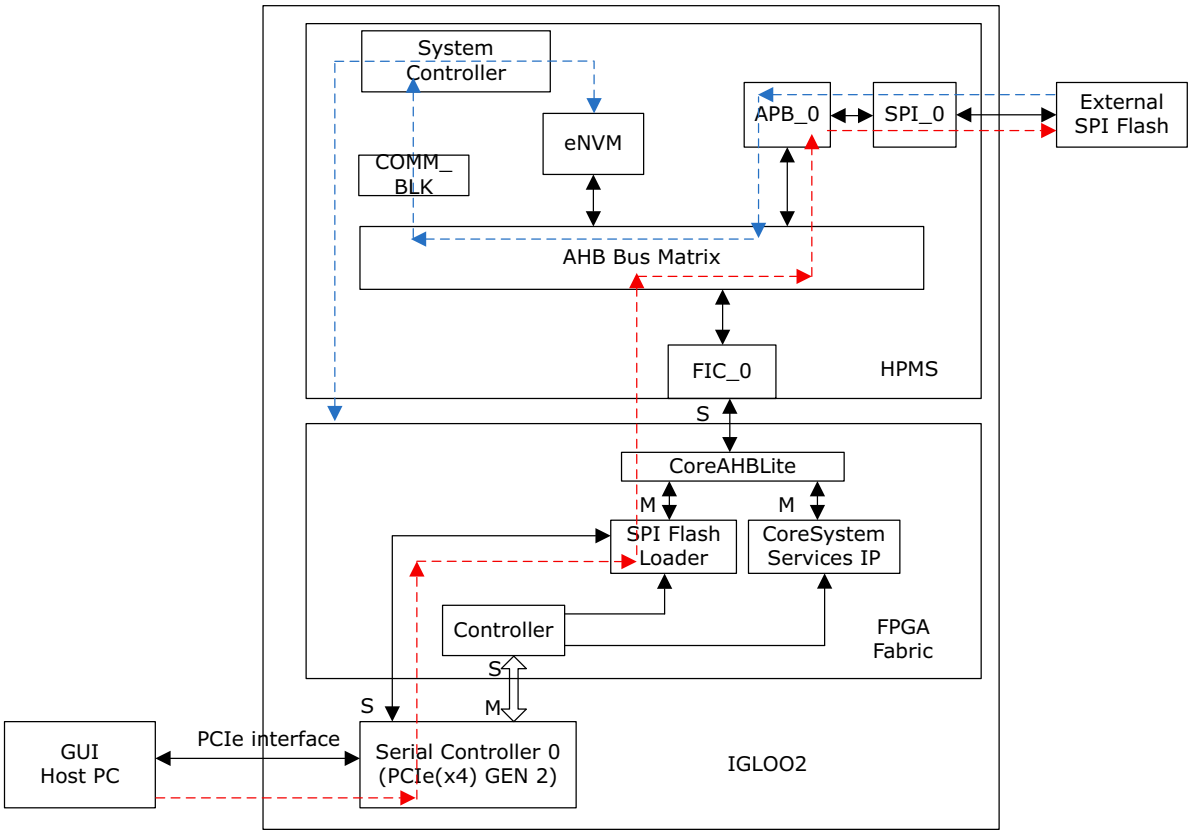


Figure 2 shows the top-level block diagram of the demo design. The arrows in red show the data flow between the host PC and the on-board external serial peripheral interface (SPI) flash memory using the PCIe interface. The SPI flash loader logic copies the programming data from the host PC to the SPI flash by using the large static random access memory (LSRAM) memory as a temporary buffer.

The arrows in blue show the system controller reading data from the external SPI flash memory to program the IGLOO2 device. For more information about IAP process, see the "Description" section on page 12.

Figure 2 • **Top-Level Block Diagram**



Step1 Transferring data bitstream from host PC to the external flash through PCIe interface

Step2 System controller reads data bitstream from the external flash to program the IGLOO2 device

3.3.2 Features

The demo design performs the following types of programming based on the input provided by the programming file:

- **eNVM programming**: The IAP programming service programs only the eNVM. In this case, the input programming file has only the eNVM content.
- **FPGA fabric programming**: The IAP programming service programs only the FPGA fabric. In this case, the input programming file has only the FPGA fabric content.
- **eNVM and FPGA fabric programming**: The IAP programming service programs both the FPGA fabric and the eNVM. In this case, the input programming file has both the FPGA fabric and the eNVM content.

3.3.3 Description

Implementing the IAP feature in IGLOO2 devices is a two-step process:

Step1: Loading SPI Flash with Programming Bitstream

The DMA logic of the SPI flash loader block implemented in the FPGA fabric receives bitstream data in blocks of 2 KB from the host PC through the PCIe interface and stores the bitstream data to the LSRAM 2 KB temporary buffer. The SPI flash loader logic reads the bitstream data that is stored in the LSRAM temporary buffer and writes to the external SPI flash connected to the HPMS SPI_0 controller. The SPI flash loader continues to receive the next block of 2 KB bitstream data through the PCIe interface until the entire programming file gets transferred from the host PC to the external SPI flash.

Step 2: Initiating the IAP Service

The bitstream data is verified by requesting the AUTHENTICATE IAP service from the system controller. The system controller reads the bitstream data from the external SPI flash using the SPI interface to check the data integrity of the bitstream data. During authentication, the remainder of the device functions normally. On successful authentication, the device can be programmed by requesting the PROGRAM IAP system controller system service. The system controller fetches the bitstream data from the SPI flash and programs the flash components of the IGLOO2 device. Programming can be done for the FPGA fabric, the eNVM, or both the FPGA fabric and the eNVM.

The system controller executes the IAP programming service in the following modes:

- **Authenticate:** The system controller IAP service validates the integrity of the programming bitstream.

For security and reliability reasons, Microsemi recommends that the bitstream must be authenticated before the program is executed, using Authenticate Operation mode. The IGLOO2 device application must commit the bitstream for programming, only after successful authentication and validation of the integrity of the bitstream.

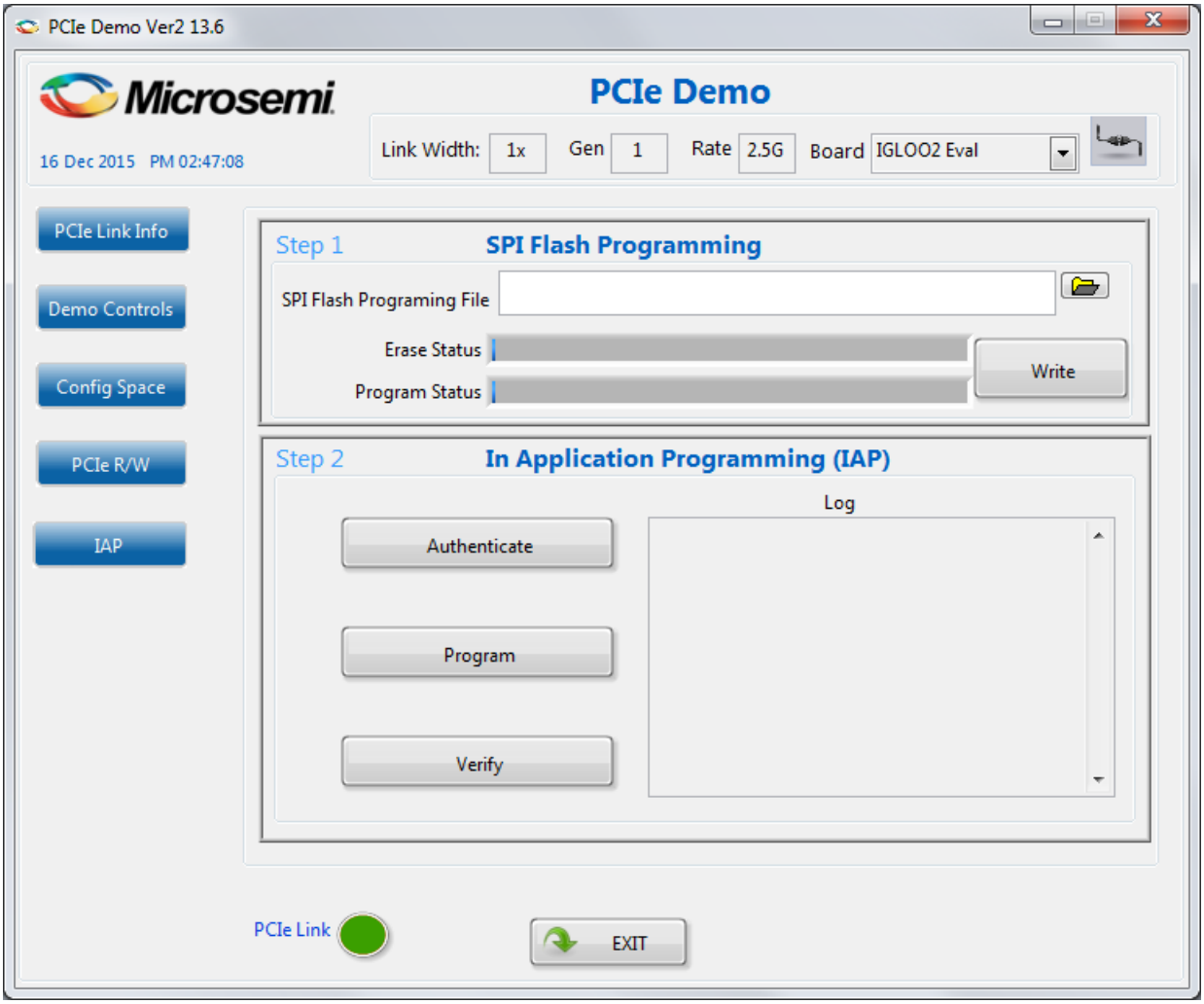
- **Program:** Depending on the programming bitstream, system controller IAP service programs the following:
  - eNVM
  - FPGA fabric
  - Both eNVM and FPGA fabric
- **Verify:** The system controller IAP service verifies the contents of the IGLOO2 device against the programming bitstream data stored in the SPI flash.

Note: The FPGA fabric is not operational during the Program or Verify operations as the device enters into Flash*Freeze (F*F) mode. During the Program or Verify operations, the PCIe communication link is in reset state. On completion of the Verify operation, the PCIe communication link is up. On completion of the Program operation, the PCIe communication link is restored to the active state when the device is programmed with the PCIe-enabled programming file. For more information on hardware implementation, see the "Appendix: Hardware Implementation" section on page 40.

3.3.4 Running GUI Application on Host PC

The GUI application is an executable program running on the host PC, which transfers the programming file (*.spi) from the host PC to the IGLOO2 Evaluation Kit on-board SPI flash through the PCIe interface. The GUI also allows the user to perform the IAP operations (Authenticate, Program, and Verify) by clicking the corresponding options, as shown in Figure 3.

Figure 3 • **IAP GUI Application**



3.3.4.1 Programming Files

Sample programming files with the file extension `.spi` are provided to program the following:

- eNVM
- FPGA fabric
- Both eNVM and FPGA fabric

The folder *<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files* contains the following sample programming files:

iap_envm_only.spi: Programs only the eNVM. The eNVM client has incremental data.

iap_fabric_only.spi: Programs only the FPGA fabric. The FPGA fabric has an LED blinking logic.

iap_fabric_and_envm.spi: Programs both the FPGA fabric and the eNVM. The eNVM client has incremental data and the FPGA fabric has an LED blinking logic. The folder *<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files\fabric_and_envm* contains the Libero design to generate the sample programming file.

pcie_iap_top.spi: This is the `.spi` file format version of the pcie_iap_top.stp file provided in *<download_folder>\m2gl_dg0681_liberov11p7_df\staple_programming_file*.

For more information on generating the `.spi` programming files, see the "Appendix: Generating .spi Programming File using Libero" section on page 43.

3.3.4.2 IAP Execution Flow

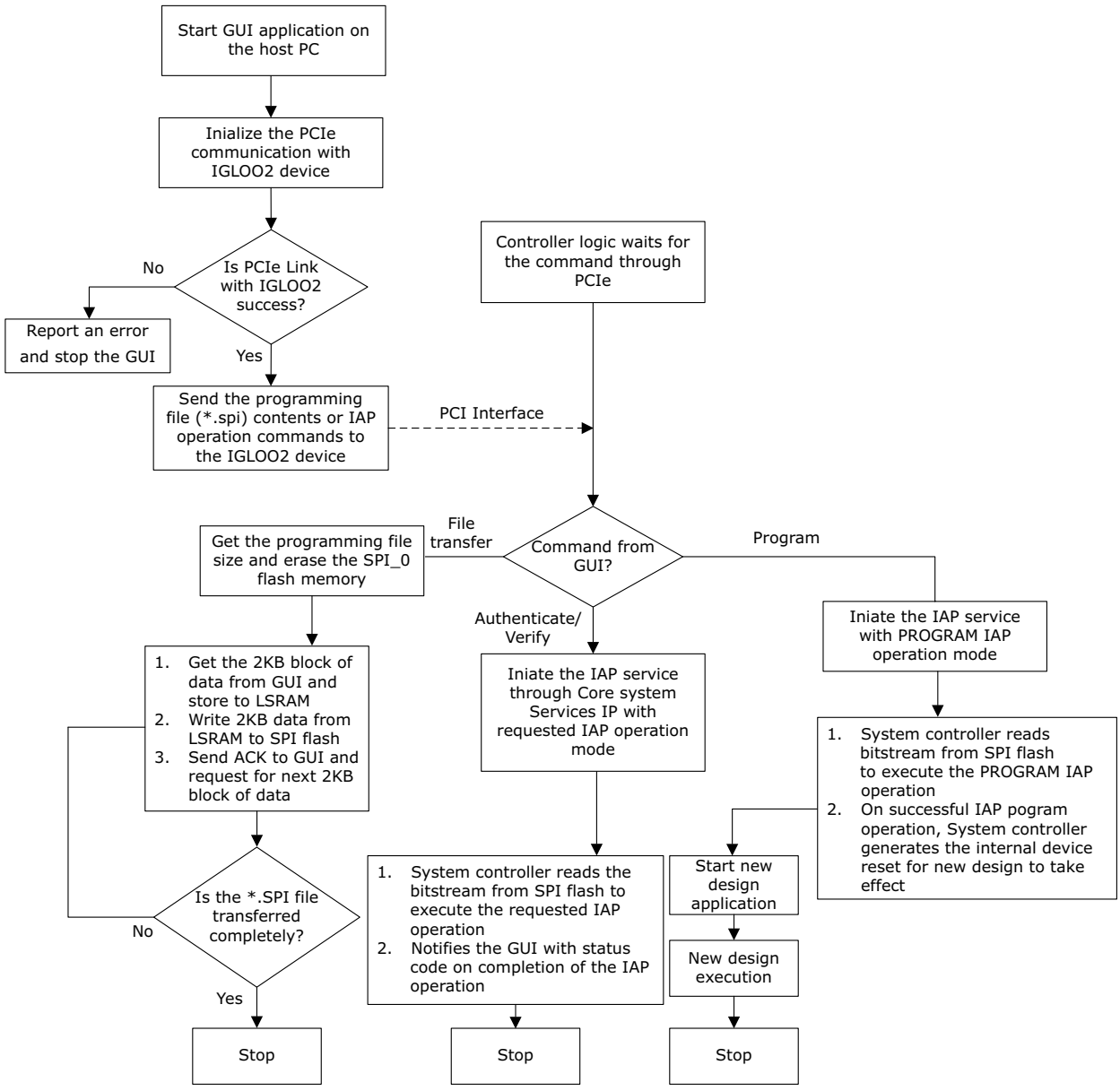
Figure 4 on page 15 shows the IAP flow. Browse to the `.spi` programming file and click **WRITE** in the GUI. The GUI application running on the host PC starts communicating with the IGLOO2 device using the PCIe interface. On connecting with the IGLOO2 device, the GUI sends the programming file size to the SPI flash loader logic and requests to erase the external SPI flash contents available on the IGLOO2 Evaluation Kit for storing the bitstream data. The SPI flash loader logic receives the bitstream data from the GUI through the PCIe interface and stores the data into the SPI flash. The GUI application copies 2 KB bitstream data from the programming file to the host PC buffer. The SPI flash loader logic reads the bitstream data from the host PC buffer through the PCIe interface and copies the data to the LSRAM 2 KB temporary buffer. The SPI flash loader logic writes the SPI flash memory with the data bitstream stored in the LSRAM. The SPI flash loader logic sends the acknowledgment to the GUI for every 2 KB block of bitstream data and requests the GUI for the next block of 2 KB data. The GUI sends the bitstream data in 2 KB blocks until the entire programming file is transferred from the host PC to the external SPI flash.

The IAP services can be executed using the GUI. If the Authenticate or Verify option is selected from the GUI, the controller logic in the fabric initiates the IAP service through the Core System Services IP with the requested IAP Operation mode and notifies the GUI with a status code indicating the completion of the authentication or verification service. For Program mode, the controller logic in the fabric does not notify the GUI with any status code as the flash components of the device are programmed with new bitstream data. On successful IAP program operation, an internal device reset is generated for the new design to take effect.

Note: You can modify/edit the programming file (*.spi) contents and run the authenticate operation to get the authentication fail message with the corresponding error code. For information about error codes, see the "Appendix: Error Codes" section on page 42.

Figure 4 shows the IAP execution flow.

Figure 4 • **IAP Execution Flow**



3.4 Setting Up the Design

The following steps describe how to setup the demo design:

1. Connect the FlashPro4 programmer to the **J5** connector of the IGLOO2 FPGA Evaluation Kit board.
2. Connect the jumpers on the IGLOO2 FPGA Evaluation Kit board, as shown in Table 2.

CAUTION: While making the jumper connections, switch **OFF** the power supply switch (**SW7**).

Table 2 • **IGLOO2 Evaluation Kit Jumper Settings**

Jumper	Pin (From)	Pin (To)	Comments
J22	1	2	Default
J23	1	2	Default
J24	1	2	Default
J8	1	2	Default
J3	1	2	Default

3. Connect the power supply to the **J6** connector.

3.4.1 Board Setup

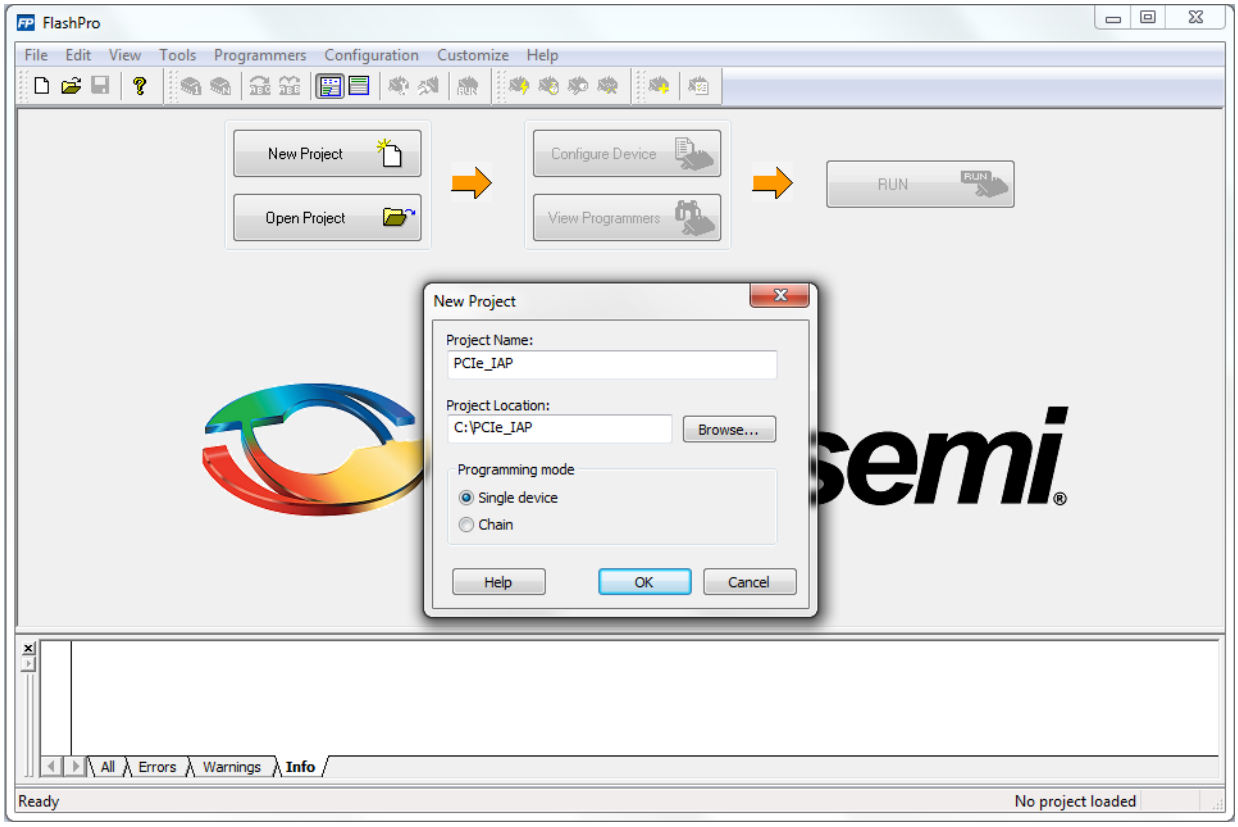
Snapshots of the IGLOO2 Evaluation Kit board with the complete setup are given in the "Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop" section on page 45

3.4.2 Programming the Demo Design

The following steps describe how to program the demo design:

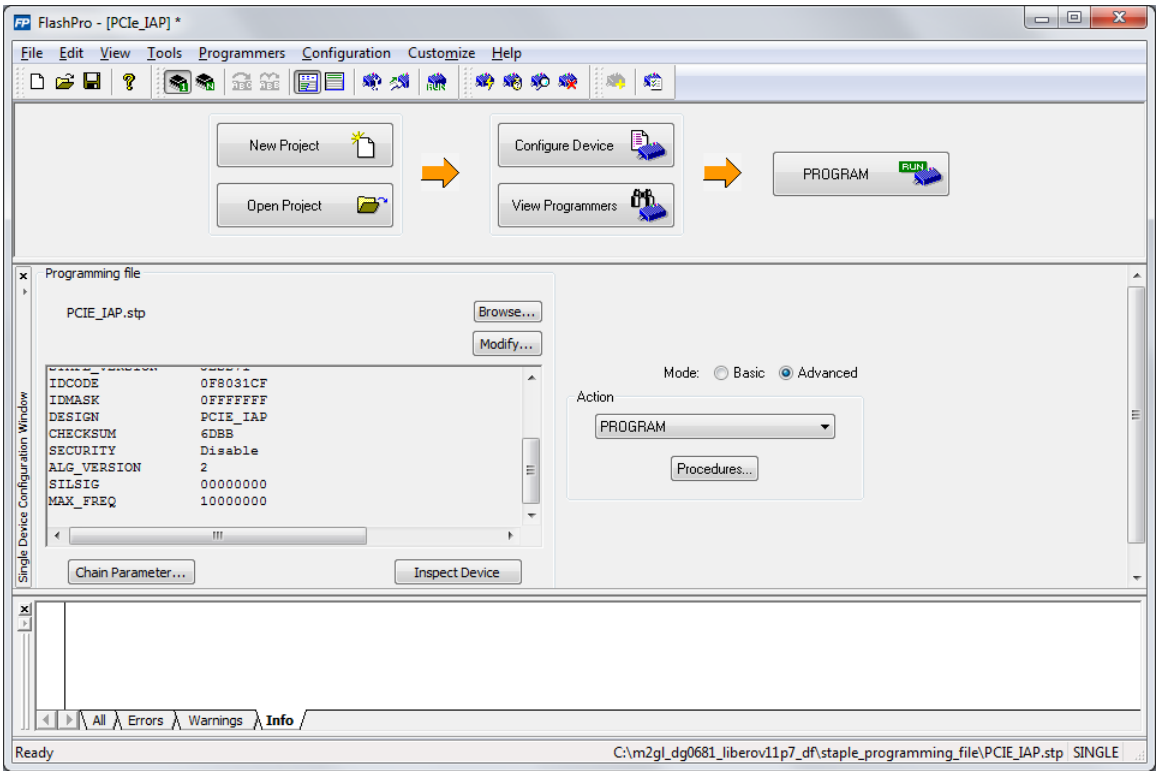
1. Download the demo design from: *http://soc.microsemi.com/download/rsc/?f=m2s_dg0681_liberov11p7_df*.
2. Switch **ON** the power supply switch (**SW7**).
3. Launch the **FlashPro** software.
4. Click **New Project**.
5. In the **New Project** window, type the **Project Name**.

Figure 5 • **FlashPro New Project**



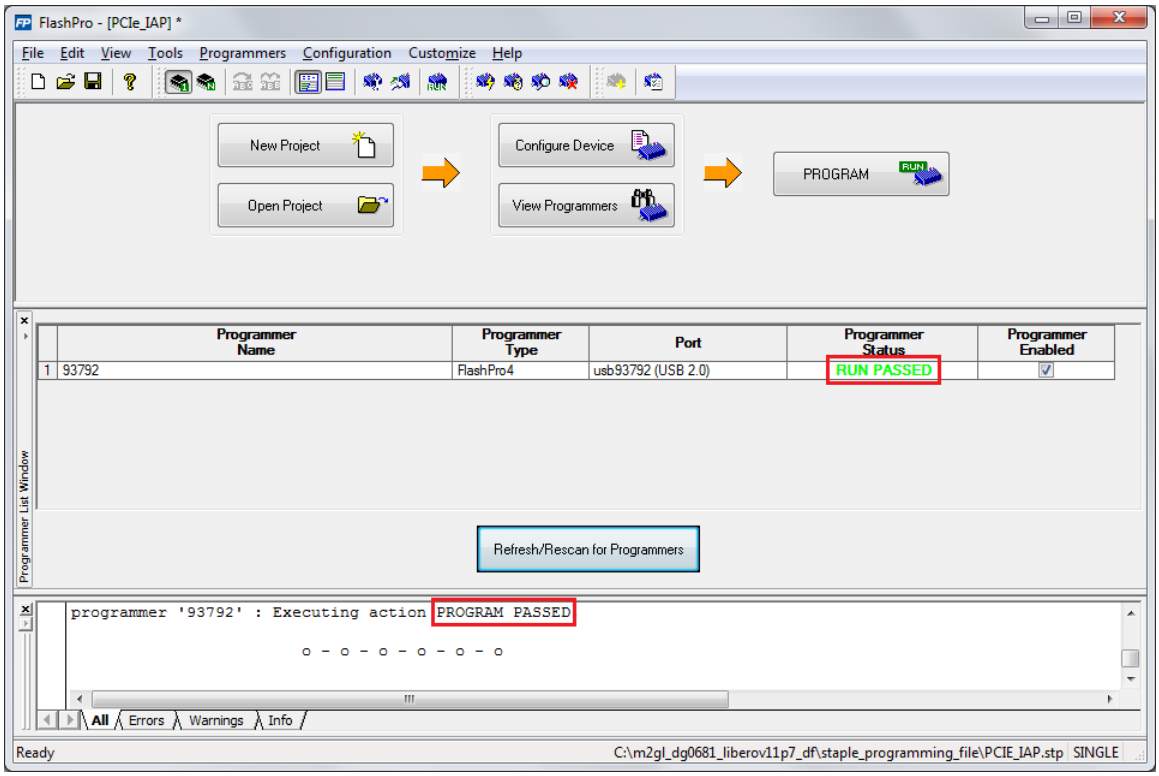
6. Click **Browse** and navigate to the location where you want to save the project.
7. Select **Single device** as the Programming mode.
8. Click **OK** to save the project.
9. Click **Configure Device**.
10. Click **Browse** and navigate to the location where the `PCIe_IAP.stp` file is located and select the file. The default location is: *IGLOO2: <download_folder>\m2gl_dg0681_liberov11p7_df\staple_programming_file*
11. Click **Open**. The required programming file is selected and is ready to be programmed in the device.

Figure 6 • **FlashPro Project Configured**



12. Click **PROGRAM** to start programming the device. Wait until you get a message indicating that the **PROGRAM PASSED**, as shown in Figure 7.

Figure 7 • **FlashPro Programming Passed**



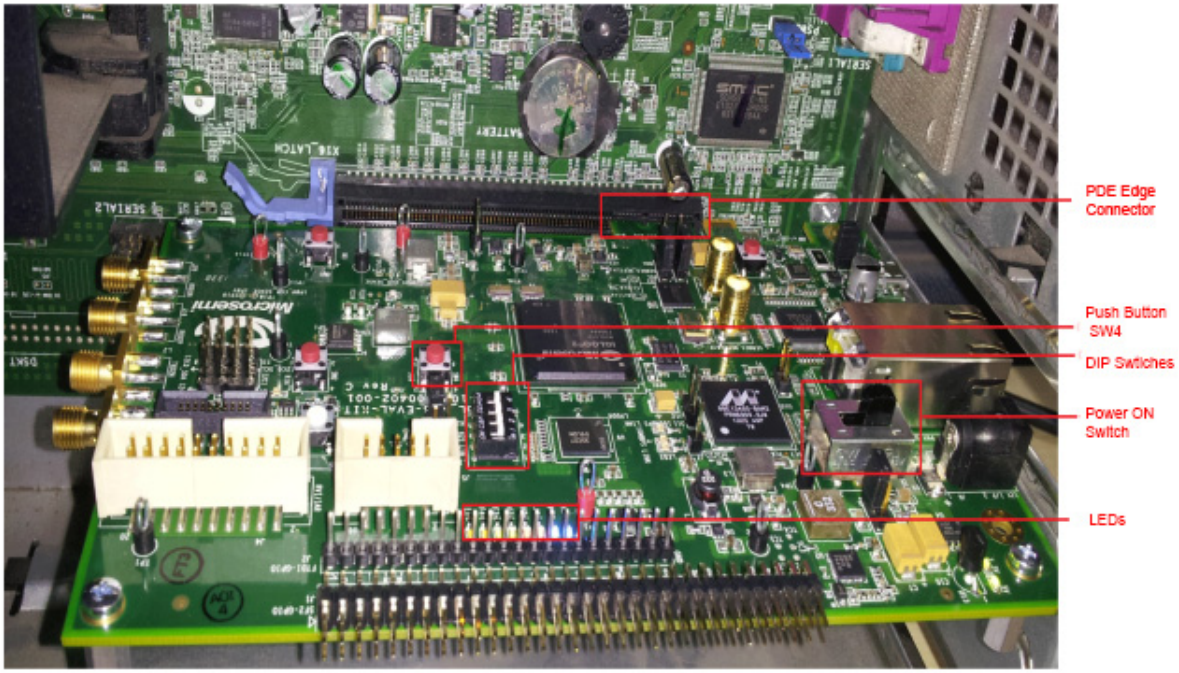
3.4.3 Connecting the Kit to the Host PC PCIe Slot

1. After successful programming, switch **OFF** the IGLOO2 Evaluation Kit and **shut down** the host PC.
2. The following steps describes how to connect the **CON1–PCIe Edge connector** either to the host PC or the laptop:
   - Connect the **CON1–PCIe Edge connector** to the host PC PCIe Gen2 slot or Gen1 slot as applicable.
   - Connect the **CON1–PCIe Edge connector** to the laptop PCIe slot using the Express card adapter.

Note: The host PC or the laptop must be switched OFF while inserting the PCIe Edge connector. If the system is not switched OFF, the PCIe device detection and selection of Gen1 or Gen2 do not occur properly.

Figure 8 shows the board setup for the host PC in which the IGLOO2 Evaluation Kit board is connected to the host PC PCIe slot. To connect the IGLOO2 Evaluation Kit board to the laptop using Express card adapter, see the "Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop" section on page 45.

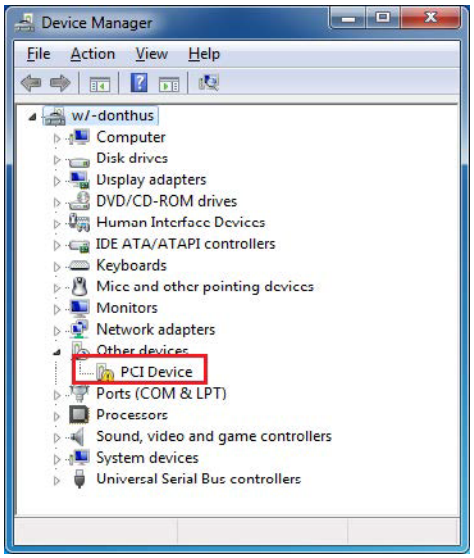
Figure 8 • **IGLOO2 Evaluation Kit Setup**



3. Switch **ON** the power supply switch (**SW7)**.

4. Switch **ON** the host PC and open **Device Manager** for the PCIe device, as shown in Figure 9. If the PCIe device is not detected, power cycle the IGLOO2 Evaluation Kit board and click **scan for hardware changes** in the **Device Manager**.

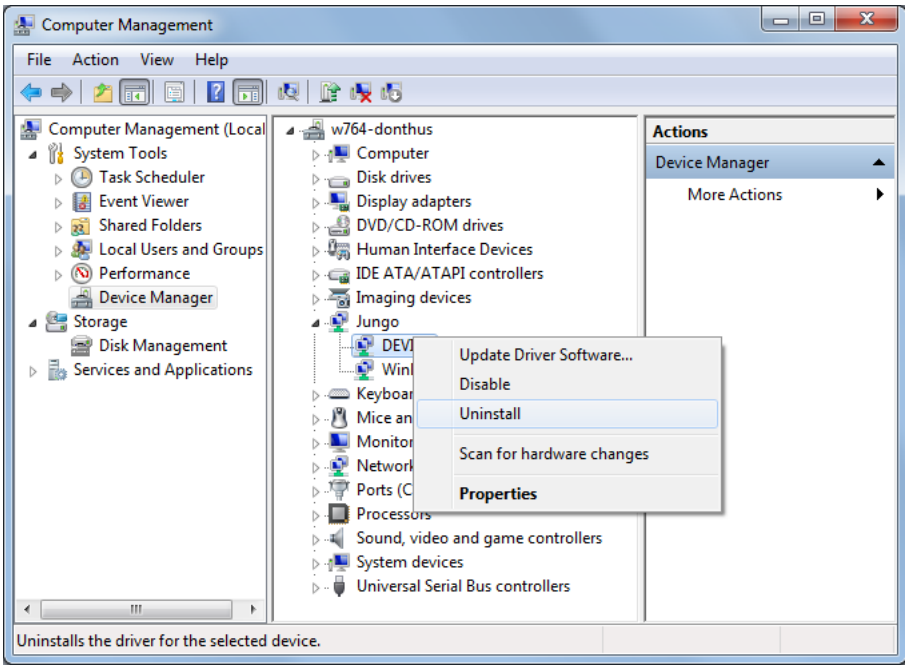
Figure 9 • **Device Manager - PCIe Device Detection**



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

5. If the host PC has any other installed drivers (previous versions of Jungo drivers) for the PCIe device, uninstall them. To uninstall previous versions of Jungo drivers, follow these steps:
   a. Go to **Device Manager** and right-click **DEVICE**, see Figure 10.

Figure 10 • **Uninstall Jungo Driver**



   b. On the **Confirm Device Uninstall** dialog box, select **Delete the driver software for this device** and click **OK**. After uninstalling the previous Jungo drivers, ensure that the PCI device is detected in the Device Manager window, see Figure 11.

Figure 11 • **Confirm Device Uninstall Dialog**

3.4.4 Drivers Installation

The PCIe demo uses a driver framework provided by Jungo WinDriverPro.

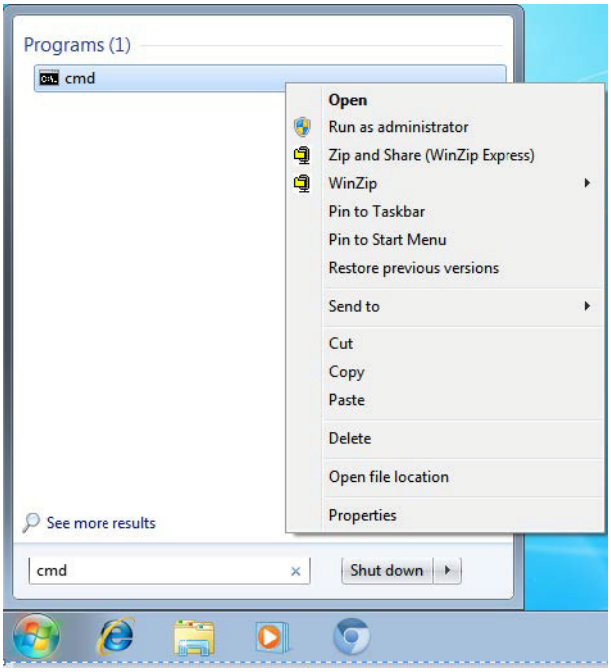
To install the PCIe drivers on the host PC for IGLOO2 Evaluation Kit, use the following steps:

1. Extract the PCIe_Demo.rar to C:\ drive. The PCIe_Demo.rar is located in the provided design files: *m2gl_dg0681_liberov11p7_df\pcie_driver\Driver_64bitOS\PCIe_Demo\DriverInstall*

Note: Installing these drivers require host PC administration rights.

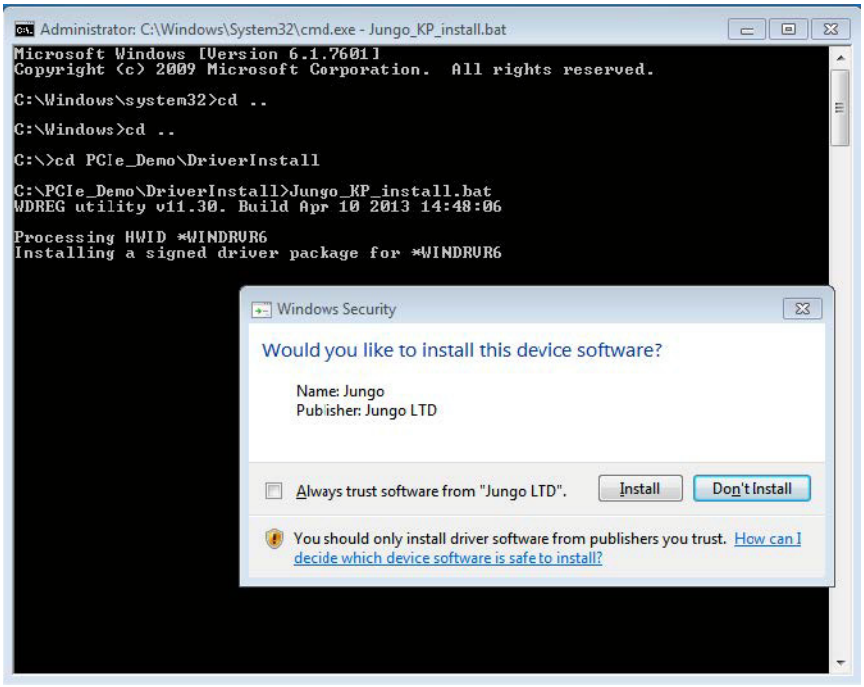
2. Run the batch file *C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat*
3. To run the batch file *C:\PCIe_Demo\DriverInstall\Jungo_KP_install.bat*, open command prompt and select **Run as administrator**, as shown in Figure 12.

Figure 12 • **Command Prompt**



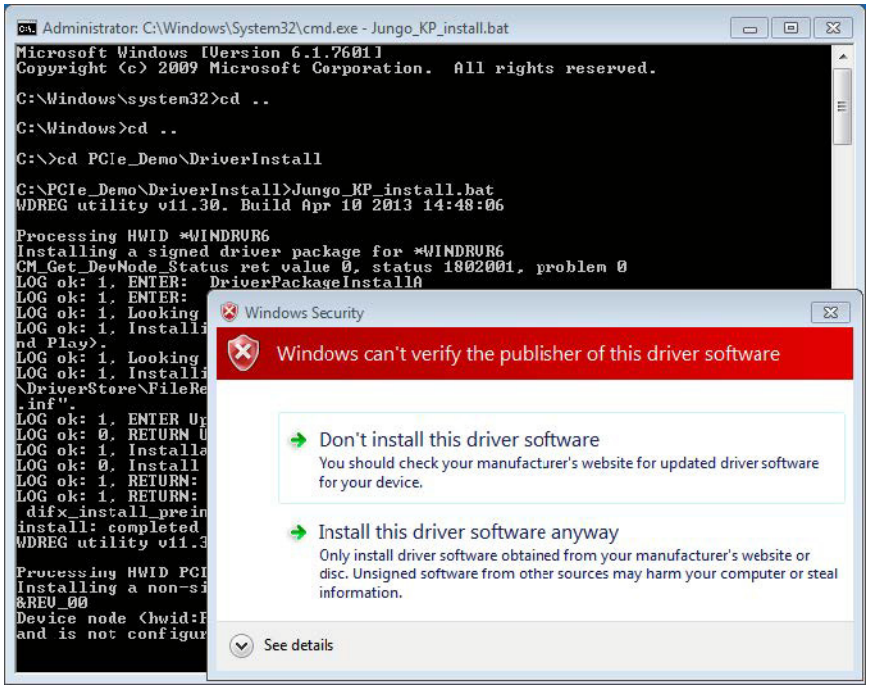
4. Navigate to *C:\ Drive and execute Jungo_KP_install.bat* in command prompt and press **Enter**. A Windows Security dialog box is displayed, as shown in Figure 13.
5. Click **Install**.

Figure 13 • **Jungo Driver Installation**



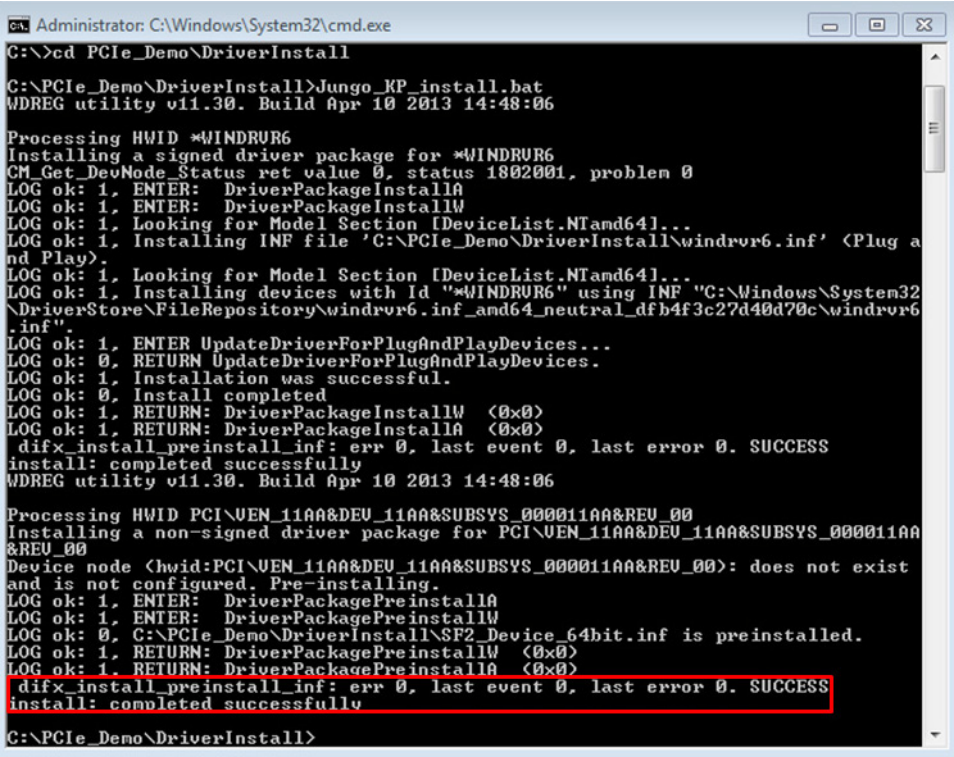
6. Click **Install this driver software anyway**, if the windows appear as shown in Figure 14.

Figure 14 • **Windows Security**



A message **SUCCESS install: completed successfully** is displayed, as shown in Figure 15.

Figure 15 • **Completed Successfully Message**



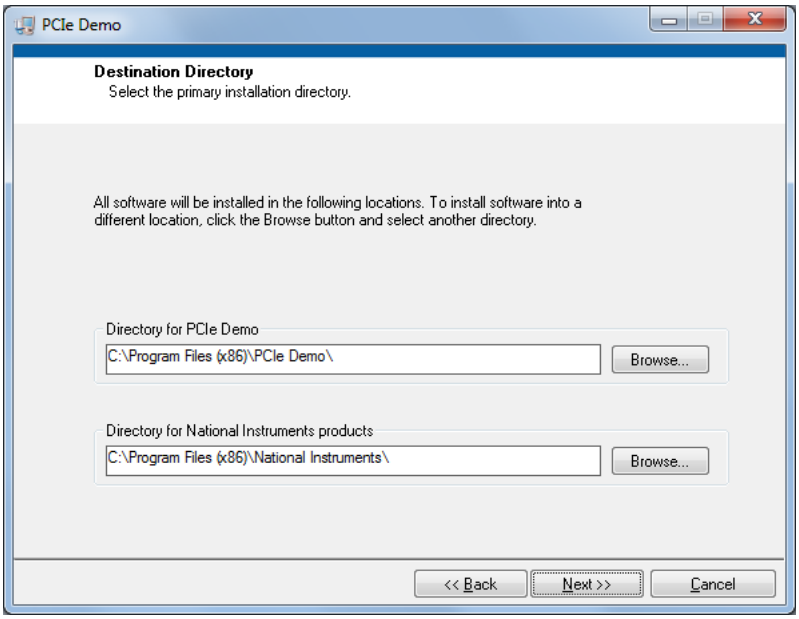
3.4.5 PCIe_Demo Application

The PCIe_Demo application is a graphic user interface that runs on the host PC to communicate with the IGLOO2 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.

The following steps describe how to install the PCIe_Demo application:

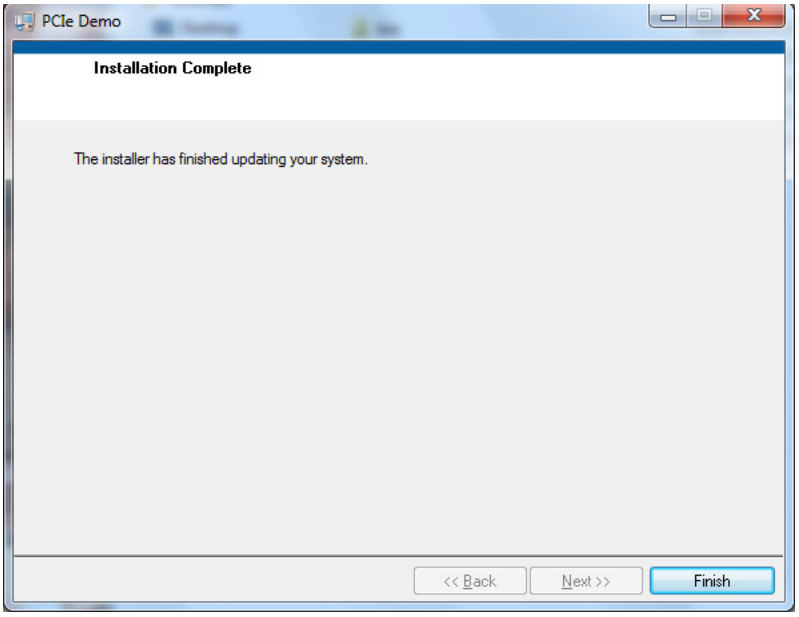
1. Download and extract the **PCIe_Demo_GUI_Installer.rar** from the provided *PCIe Demo GUI Installer*.
2. Double-click **setup.exe** in the provided GUI installation (*PCIe_Demo_GUI_Installer\setup.exe*). Apply default options, as shown in Figure 16.

Figure 16 • **GUI Installation**



3. Click **Next** to complete the installation. Figure 17 is displayed after successful installation.

Figure 17 • **Successful Installation of GUI**



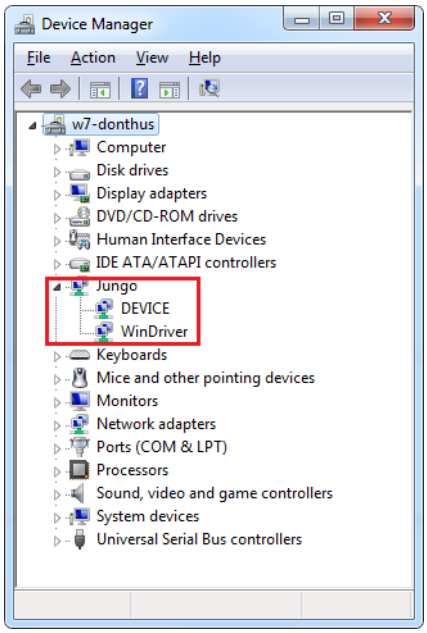
4. Restart the host PC.

3.5 Running the Design

The following steps describe how to run the demo design:

1. Check the host PC **Device Manager** for the drivers. If the device is not detected, power cycle the IGLOO2 Evaluation Kit and click **scan for hardware changes** in the **Device Manager**.

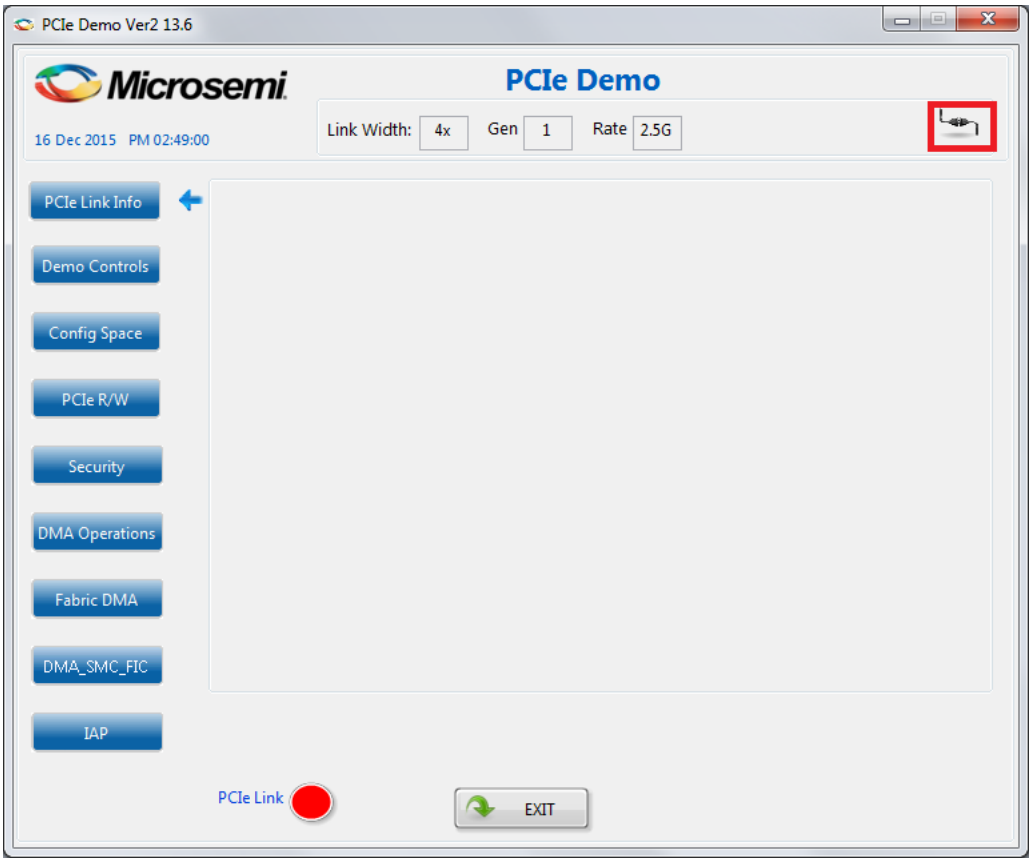
Figure 18 • **Device Manager - PCIe Device Detection**



Note: If a warning symbol appears on the **DEVICE** or **WinDriver** icons in the **Device Manager**, uninstall them and start from step1 of "Drivers Installation" section on page 22.

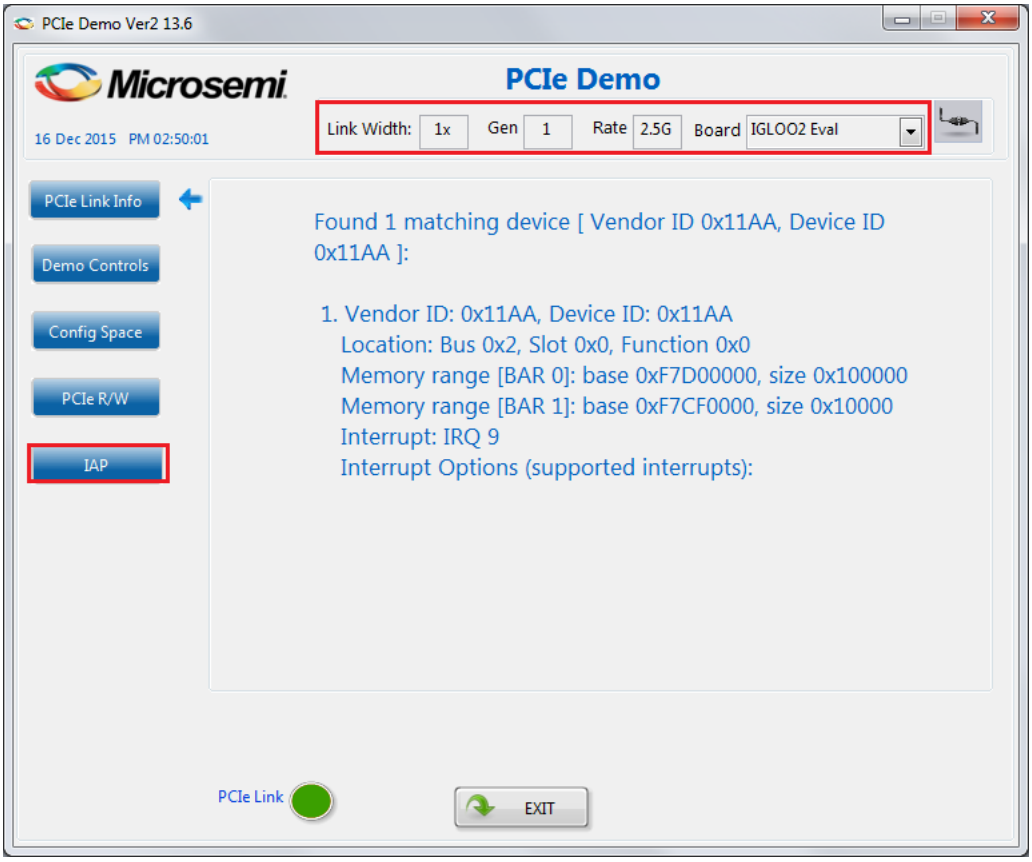
2. Start the PCIe_Demo application from **ALL Programs** > **PCIe_Demo** > **PCIe_Demo.exe**. The PCIe_Demo application is displayed, as shown in Figure 19.

Figure 19 • **PCIe_Demo Application**



3. Click **Connect**. The application detects and displays the connected kit, demo design, and PCIe link. A message is displayed, as shown in Figure 20.

Figure 20 • **PCIe Device Information**

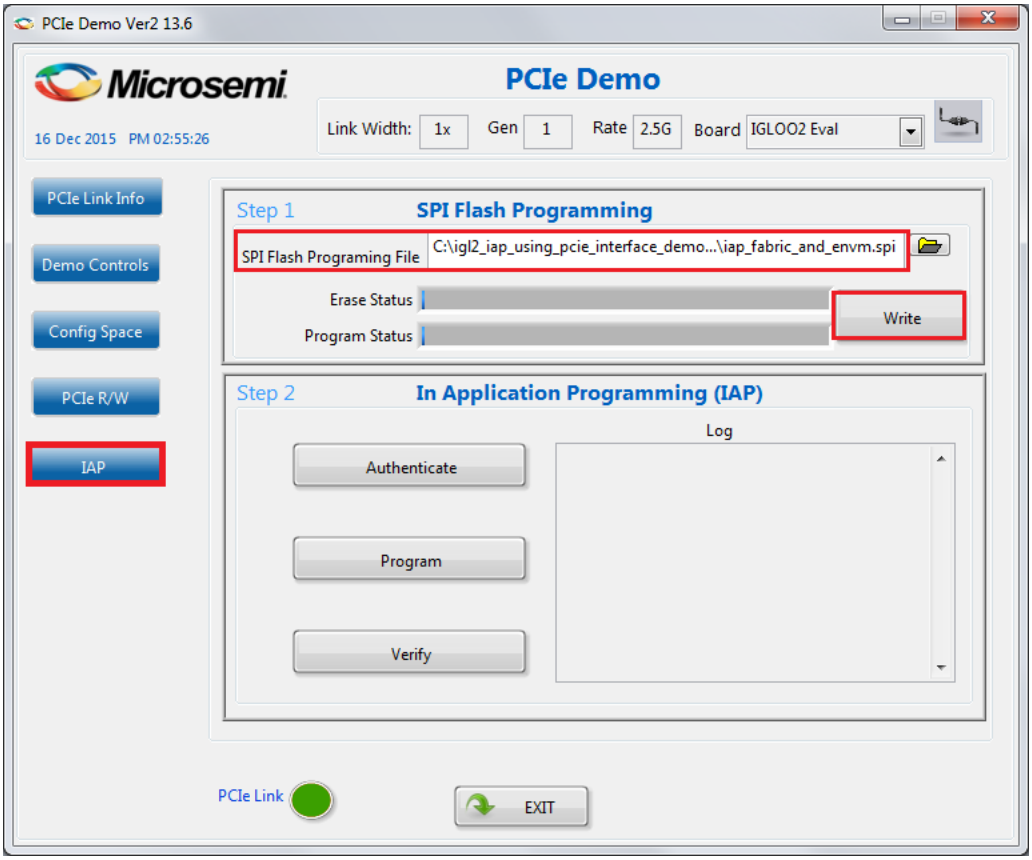


3.5.1 IAP Step1: Loading SPI Flash with Programming Bitstream

The following steps describe how to load the SPI flash with programming bitstream:

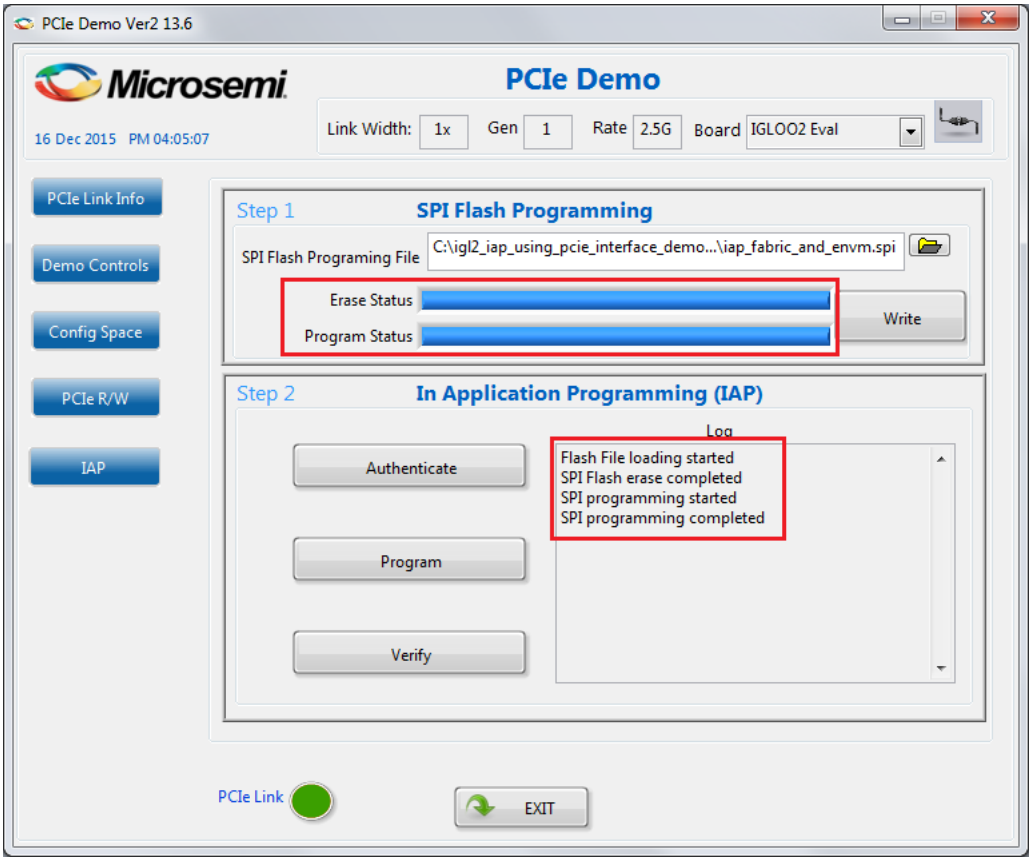
1. Click **IAP**, as shown in Figure 21.
2. Click **Browse** to select the `*.spi` programming file to write the bitstream data to the SPI flash memory, as shown in Figure 21. The sample `*.spi` programming file can be selected from *<download_folder>\m2gl_dg0681_liberov11p7_df\ sample_programming_files.*

Figure 21 • **Selecting Programming File**



3. Click **Write** to move the programming file to the SPI flash memory. The SPI flash memory is erased according to the programming file size and the programming file bit stream is written to the SPI flash memory. The status of the SPI flash erase and the SPI flash program operations can be observed on the status bars and the messages can be viewed on the log window, as shown in Figure 22.

Figure 22 • **SPI Flash Programming**



3.5.2 IAP Step2: Initiating the IAP Services

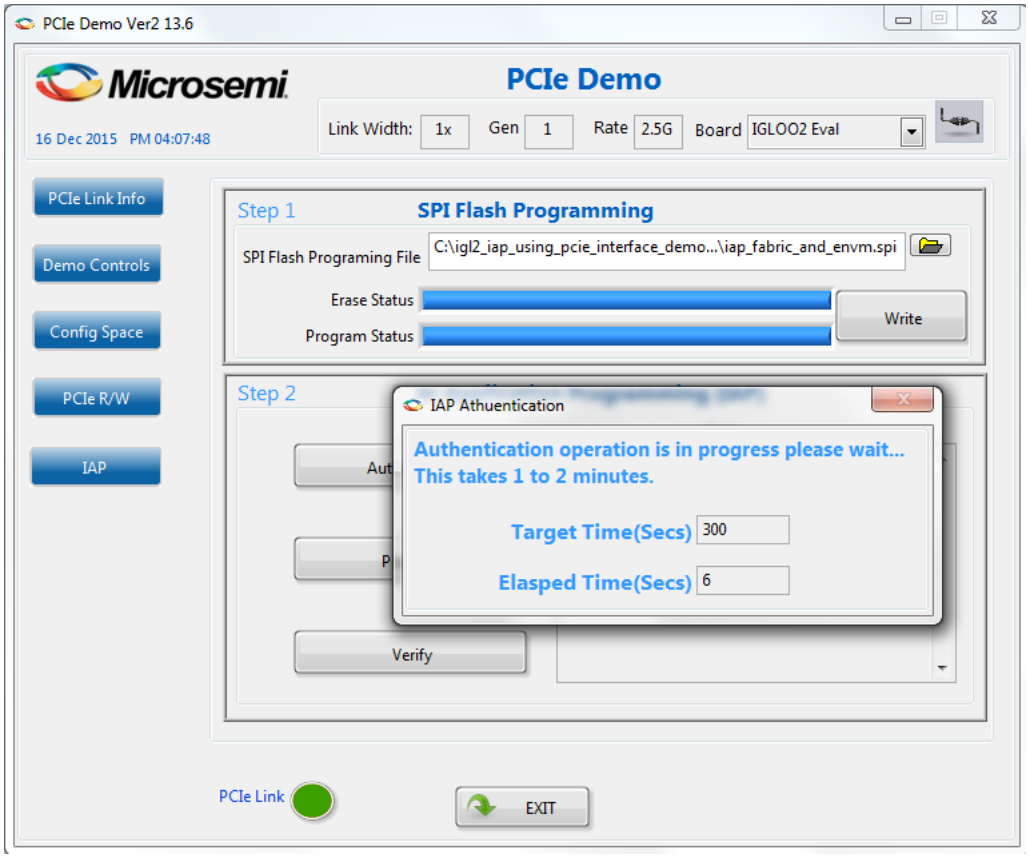
The IAP services can be executed by clicking **Authenticate**, **Verify**, or **Program**.

3.5.2.1 Authenticate and Program Operation Mode

The following steps describe how to authenticate and program the operation mode:

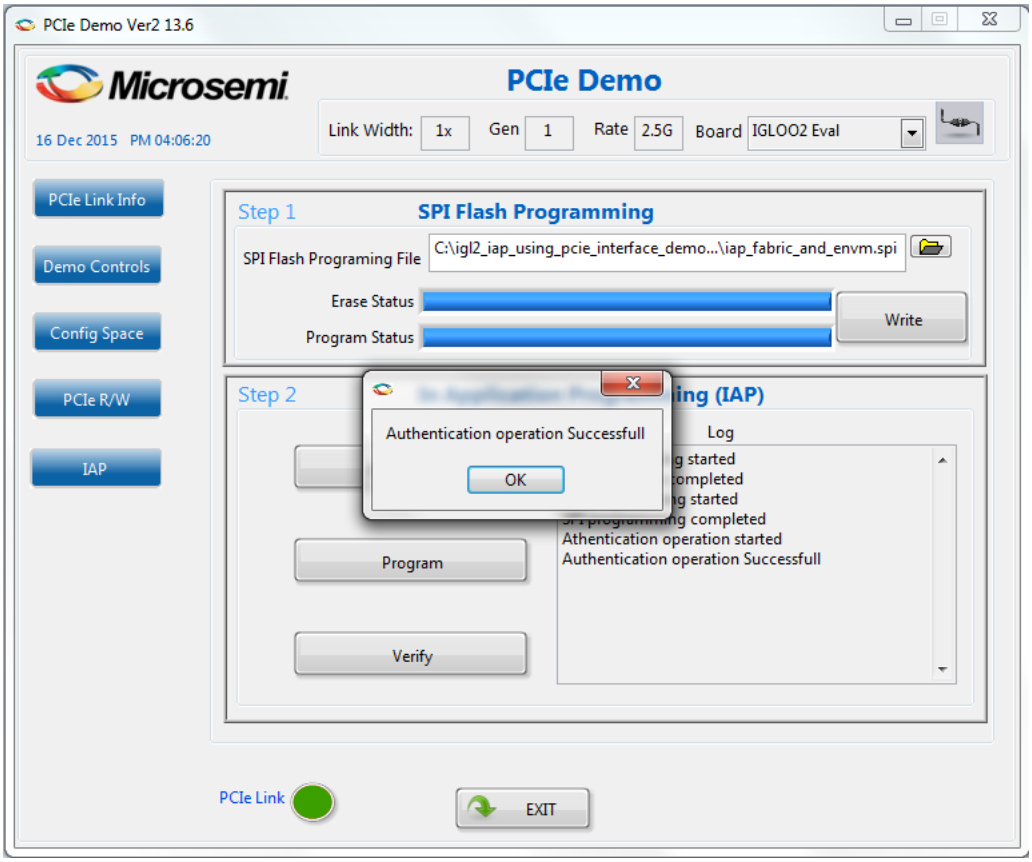
1. Click **Authenticate** to check the data integrity of the bitstream data stored in SPI flash. Figure 23 shows the **IAP operation is in progress** message.

Figure 23 • **IAP Authentication**



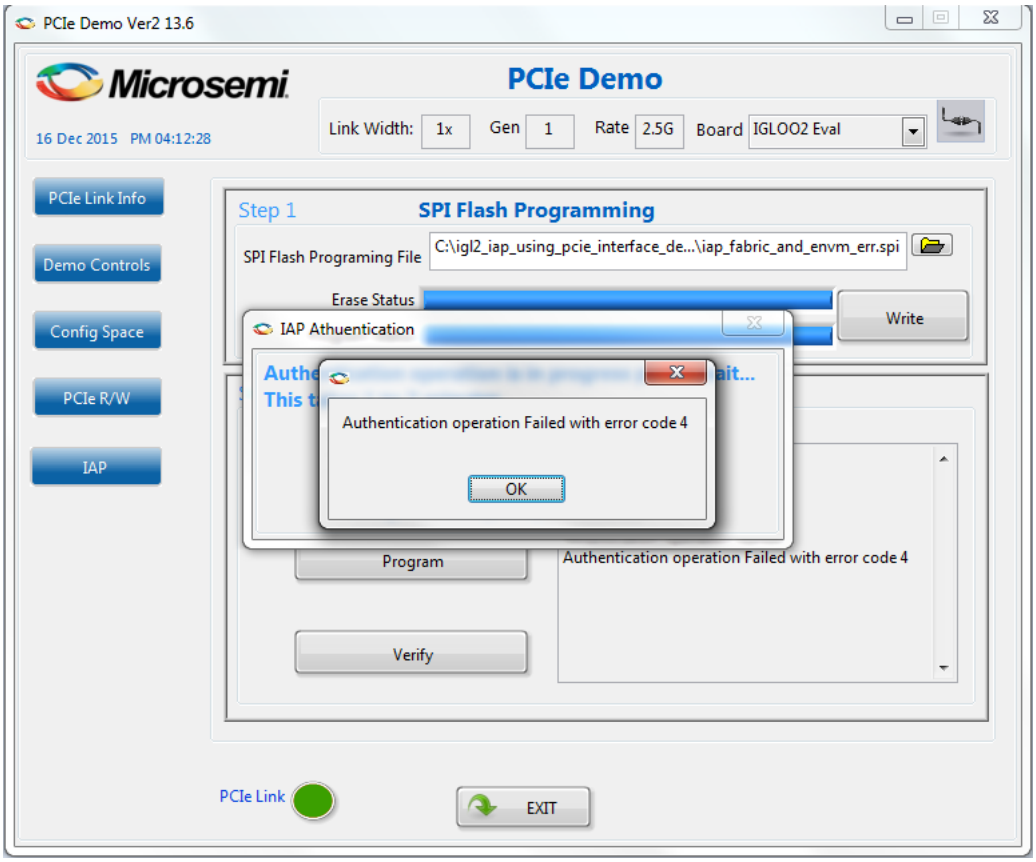
On completion of the IAP authentication, an **Authentication operation successful** message is displayed, as shown in Figure 24.

Figure 24 • **IAP Authentication Success Message in PCIe Demo GUI**



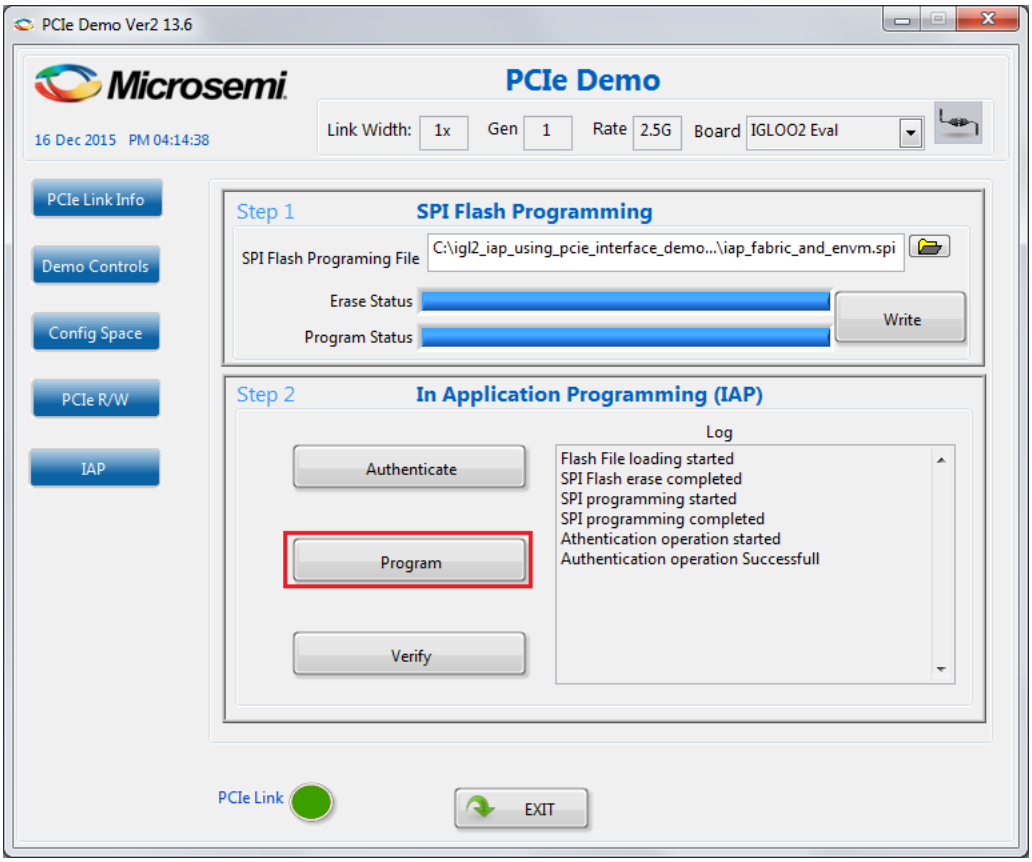
The demo GUI also provides an option to create an authentication failure scenario, which can be achieved by modifying the programming file (`*.spi`) contents and running the authenticate operation. Figure 25 shows a sample authentication failed error scenario message. For information about error codes, see the "Appendix: Error Codes" section on page 42.

Figure 25 • **Authentication Failed**



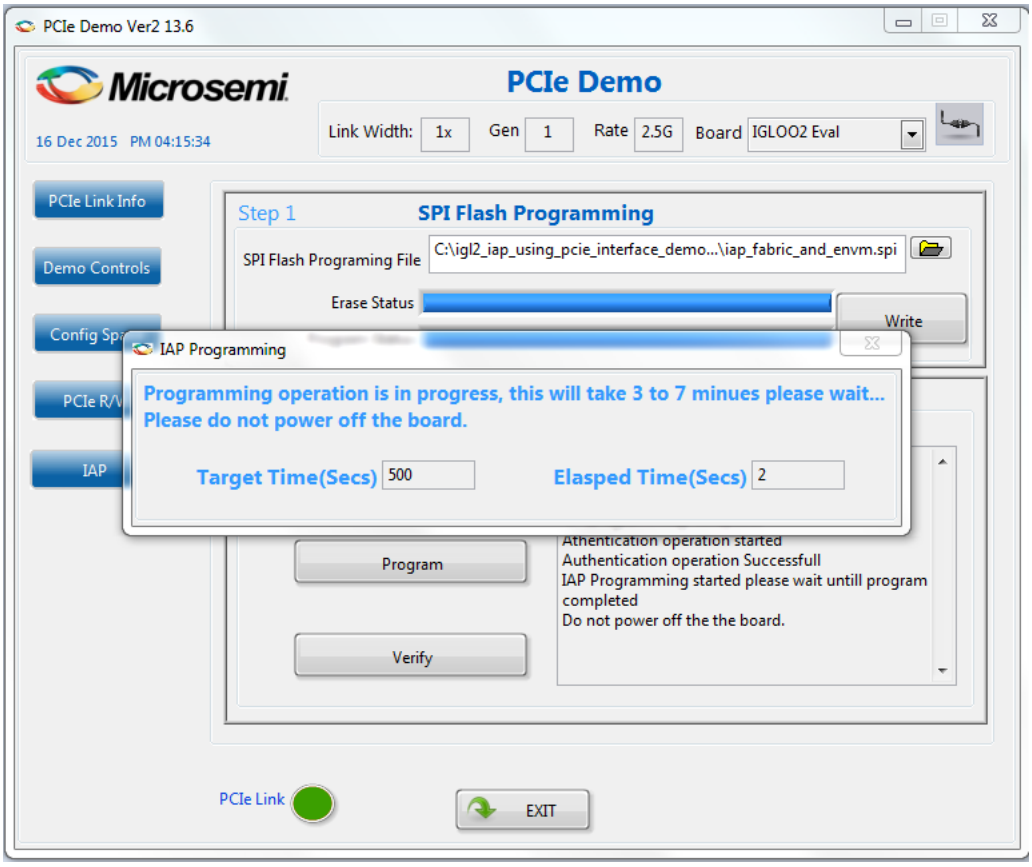
2. Click **Program** to reprogram the IGLOO2 device.

Figure 26 • **IAP Program**



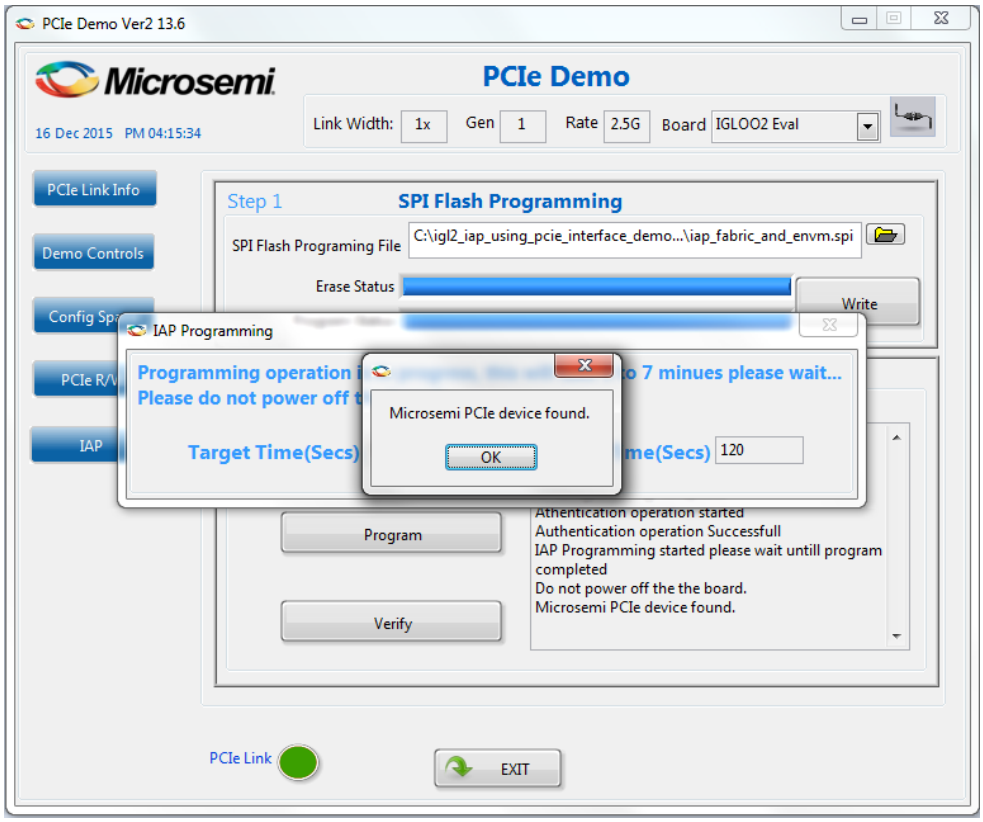
It takes a few minutes for the IAP service to complete and program the FPGA fabric and the eNVM. Figure 27 shows the IAP program status.

Figure 27 • **Programming Operation**



On completion of the IAP program, the GUI displays a **Microsemi PCIe Device Found** message, as shown in Figure 28. This indicates that the IGLOO2 device is programmed with the PCIe enabled design and the PCIe communication link status is restored to the active state. If the IGLOO2 device is programmed without the PCIe enabled design, the GUI displays a **No Microsemi PCIe Device Found** message. See Table 3 for more information on design type and programming results.

Figure 28 • **Microsemi PCIe Device Found Message in PCIe Demo GUI**



3.5.2.2 Programming Results

Table 3 shows the results for IAP Program operation mode for the sample programming files provided in the folder: *<download_folder>\m2gl_dg0681_liberov11p7_df\ sample_programming_files*. All the `.spi` files listed in Table 3 are not demonstrated.

Table 3 • **IAP Programming Results**

*.spi Programming File Name	Design Type	eNVM Programming Result	FPGA Fabric Programming Result
`iap_envm_only`	PCIe is not enabled	Incremental data pattern in eNVM (can be verified using SmartDebug)	NA
`iap_fabric_only`	PCIe is not enabled	NA	IGLOO2 LEDs 1 to 4 blinks
`iap_fabric_and_envm`	PCIe is not enabled	Incremental data pattern in eNVM (can be verified using SmartDebug)	IGLOO2 LEDs 1 to 4 blinks
`Pcie_iap_top`	PCIe is enabled	Retains original eNVM content	Retains original Fabric logic

Note: After the successful IAP program operation, the IGLOO2 Evaluation Kit must be reprogrammed with the original `pcie_iap_top.stp` file to try the IAP operation modes again if the design type is not PCIe enabled.

3.5.2.3 Verify Operation Mode

The following steps describe how to verify operation mode:

1. Click **Verify** to verify the IGLOO2 device FPGA fabric and eNVM contents. Figure 29 shows the **Verification operation is in progress** message.

Figure 29 • **IAP Verify Operation Status**

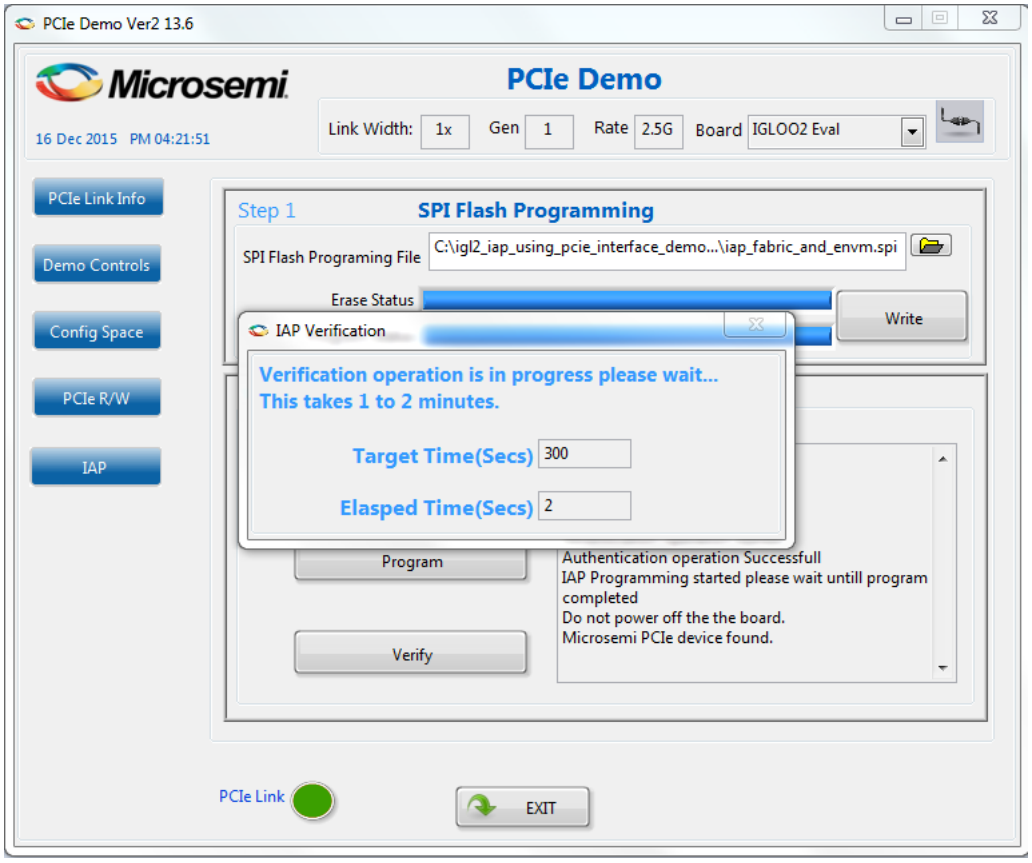
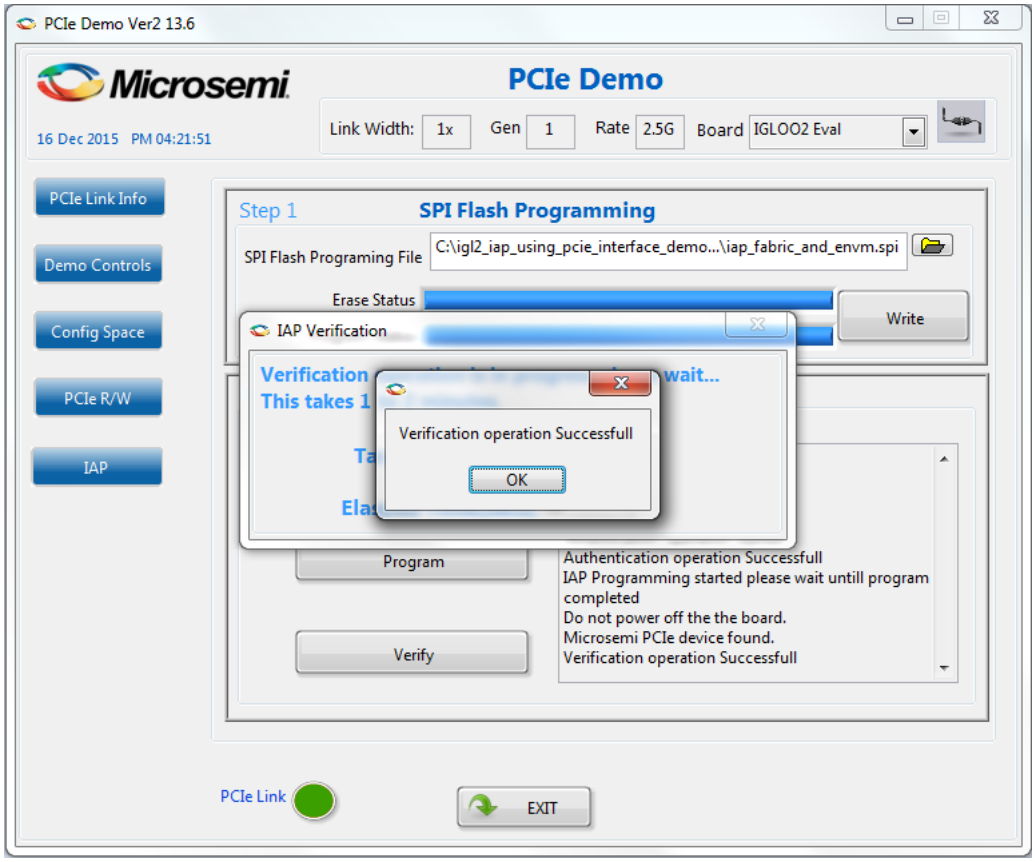


Figure 30 shows the **Verification operation Successful** message.

Figure 30 • **IAP Verification Success**



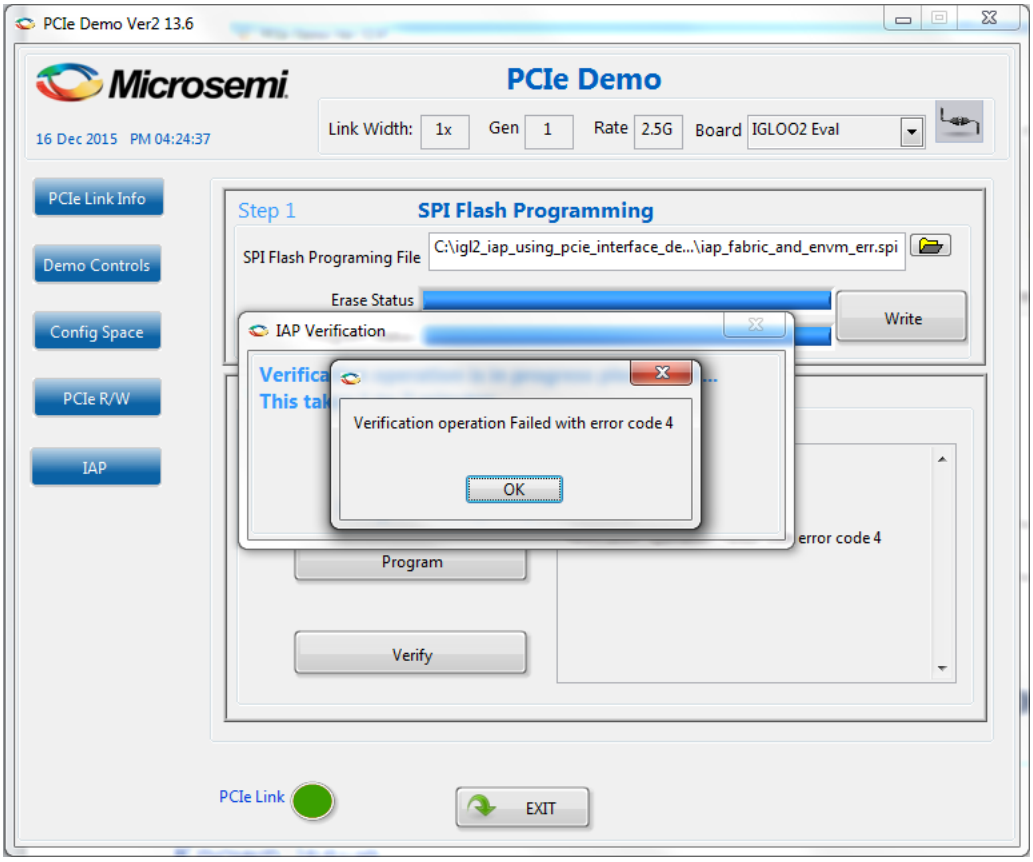
The verification operation is successful when the IGLOO2 device contents match the programming bitstream data stored in the SPI flash. If the verification fails, the GUI displays an error message with an error code. For information about error codes, see the "Appendix: Error Codes" section on page 42.

The sample programming files are at:
<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files

All the files do not pass the verification. Only the `pcie_iap_top.spi` file passes the verification operation as it matches the IGLOO2 device contents (`pcie_iap_top.stp`).

Figure 31 shows the verification error message.

Figure 31 • **IAP Verification Error Message**



3.6 Known Issue

After the successful completion of the two-step IAP, the LSRAM read and write access fails from the fabric path. This is a known silicon issue, which is documented in the *ER0198: IGLOO2 Device Errata*. The workaround for this problem is to reset the system after the IAP program operation. Microsemi recommends that this workaround is implemented for any design, which accesses LSRAM after IAP.
For more information about how to implement this workaround, see the "Appendix: Implementing Workaround to Access Fabric LSRAM after IAP/ISP Program Operation" section on page 48.

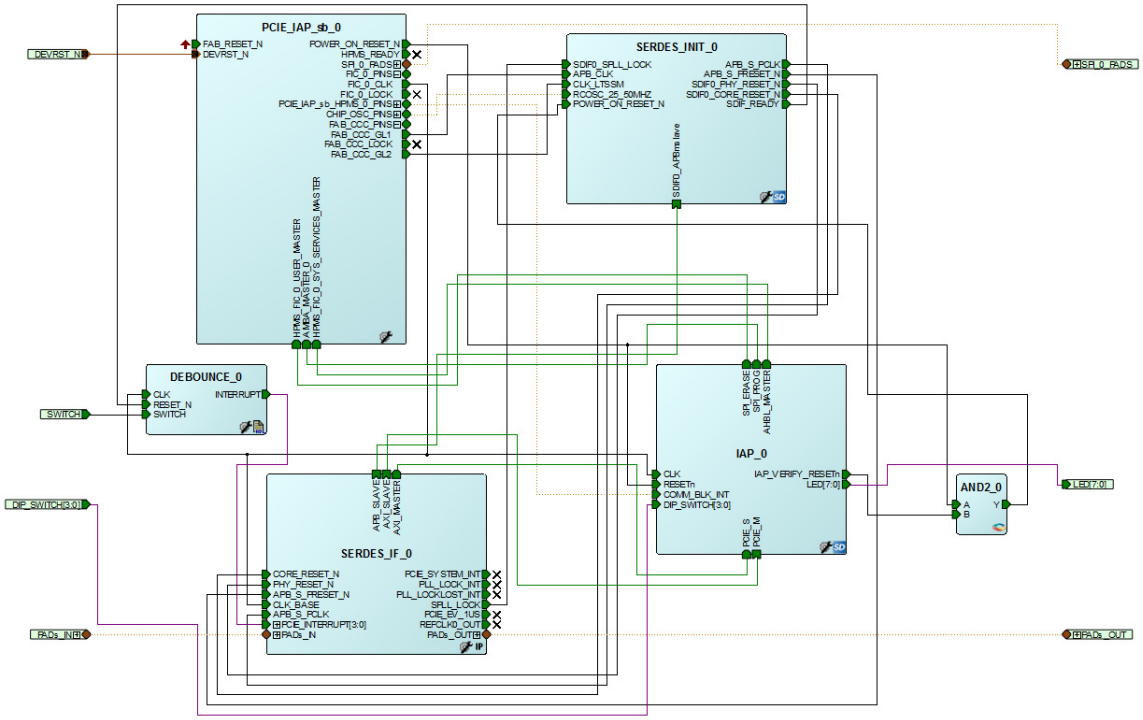
The design example provided in this demonstration implements the workaround for accessing the LSRAM after implementing the IAP program operation in the Libero software. The design files are available in the following location:
<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files\LSRAM_Workaround\PCIE_IAP_Tamper.rar

4 Appendix: Hardware Implementation

In this demo design, the following blocks are configured in the Libero hardware project:

- The SERDES_IF_0 in the IGLOO2 device is configured for PCIe 2.0, ×1 lane, and Gen2 rate.
- The AXI master interface of SERDES_IF_0 is enabled and connected to the controller logic in the fabric to initiate the IAP through PCIe. The AXI slave interface of SERDES_IF_0 is connected to the AXI master interface of the SPI flash loader logic to access the PCIe interface for the programming bitstream file.
- BAR0 and BAR1 are configured in the 32-bit memory mapped memory mode.
- The HPMS SPI_0 controller is enabled to access the external SPI flash memory.

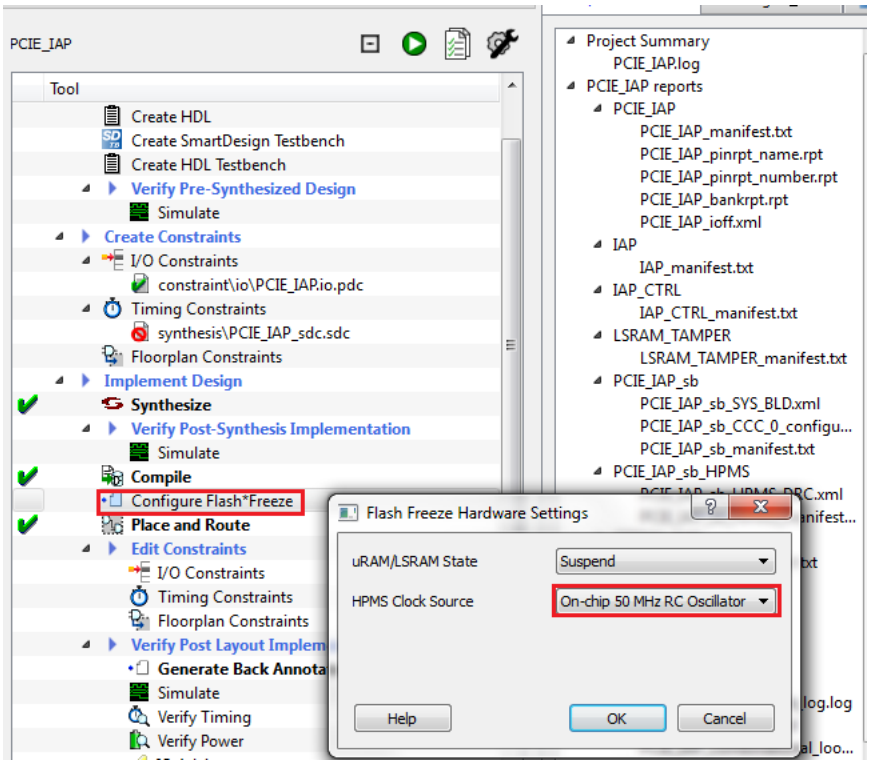
Figure 32 • **Demo Design Libero Top-Level Diagram**



4.1 Standby Clock Source Configuration

The standby clock source for the HPMS in F*F mode is configured to the on-chip 50 MHz RC Oscillator using the **Flash*Freeze Hardware Settings** dialog box in the Libero SoC software, as shown in Figure 33. A higher HPMS clock frequency is required in F*F mode to meet the SPI communication speed requirements.

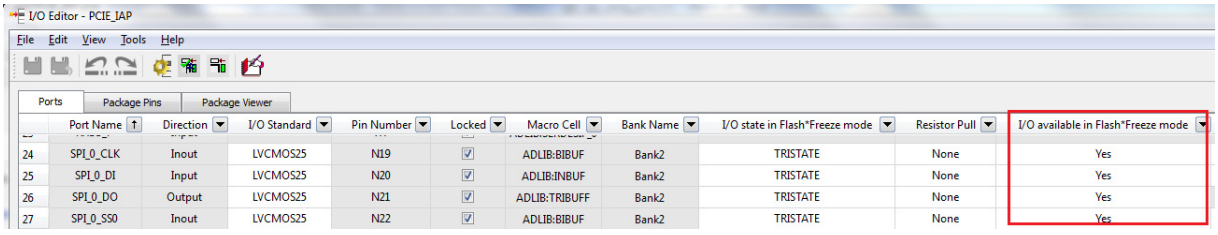
Figure 33 • **Flash*Freeze Hardware Settings**



4.2 Configuring I/Os for Flash*Freeze Mode

The FPGA fabric is not operational during the Program or Verify IAP operations as the device enters into Flash*Freeze (F*F) mode. On the IGLOO2 Evaluation Kit board, the SPI_0 is interfaced to the on-board SPI flash memory for loading the programming bitstream data to the SPI flash using the SPI interface. During F*F mode, the fabric and the I/Os are not available. Therefore, all the SPI_0 ports are configured using the I/O Editor to be available during F*F mode, as shown in Figure 34. **Commit and Check** the settings from the **File** menu after configuring the SPI_0 ports.

Figure 34 • **Configuring SPI_0 Ports Available During F*F**



5 Appendix: Error Codes

Table 4 lists the information about the error codes.

Table 4 • **Error Codes**

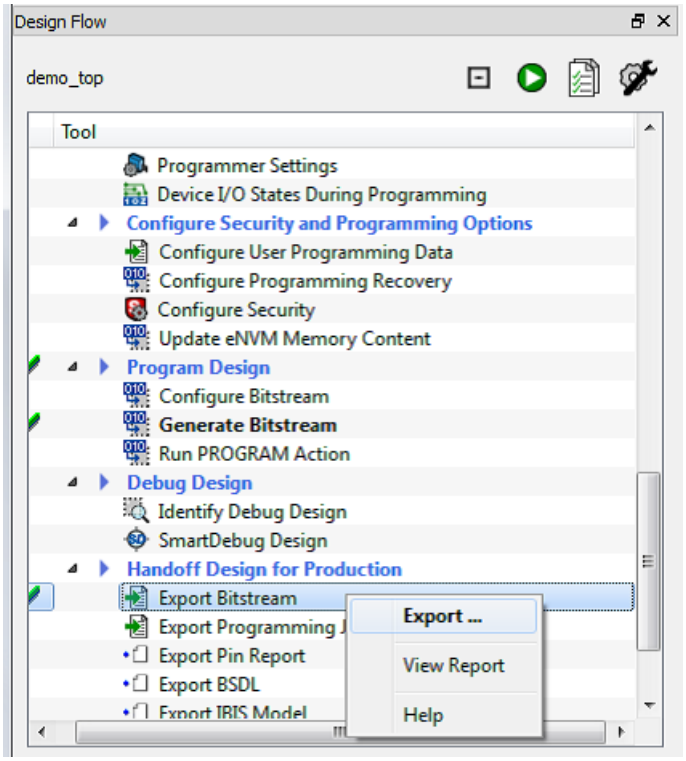
Error Code	Description
1	Device contents mismatch
2	Data is not supported
3	Invalid encryption key
4	Invalid file
5	Corrupted/ Invalid file
7	Corrupted/ Invalid file
8	Corrupted/ Invalid file
9	Invalid device capabilities
10	Invalid device ID
11	Bitstream not supported
12	Verification not allowed for input bitstream
127	Operation aborted
129	eNVM verification failed
130	Device secured
131	Programming mode not enabled

6 Appendix: Generating .spi Programming File using Libero

The following steps describe how to generate the `.spi` programming file using Libero software:

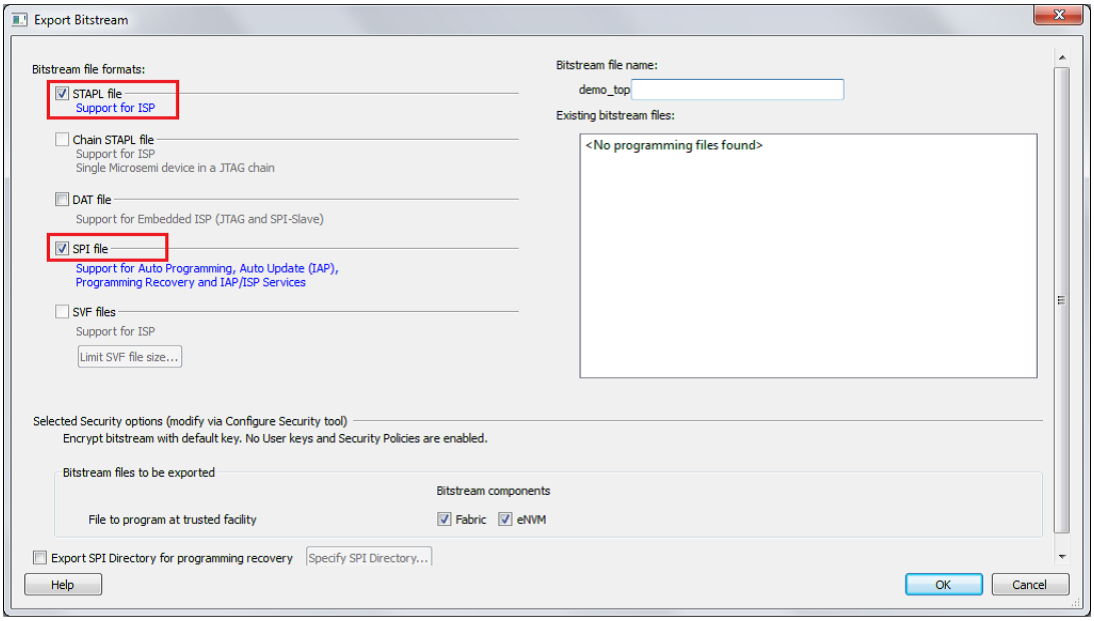
1. Launch the Libero SoC software to open a project for the `iap_fabric_and_envm.spi` programming file. The Libero design file is provided at *<download_folder>\m2gl_dg0681_liberov11p7_df\smaple_programming_files\fabric_and_envm*.
2. Right-click **Export Bitstream** under **Handoff Design for Production** in the **Design Flow** tab, and click **Export …**

Figure 35 • **Configuring Export Bitstream**



3. Select the **SPI file** check box in the **Export Bitstream** window and click **OK**.

Figure 36 • **Export Bitstream Window**



7 Appendix: IGLOO2 Evaluation Kit Board Setup for Laptop

Figure 37 shows how to line up the IGLOO2 Evaluation Kit PCIe connector with the adapter card slot.

Figure 37 • **Lining up the IGLOO2 Evaluation Kit Board**

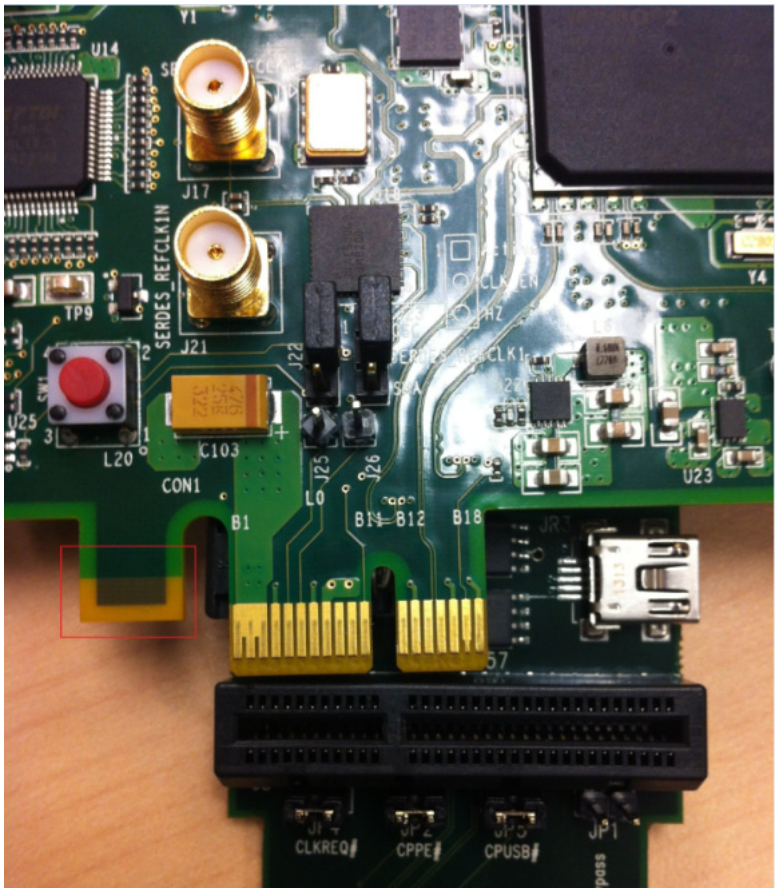
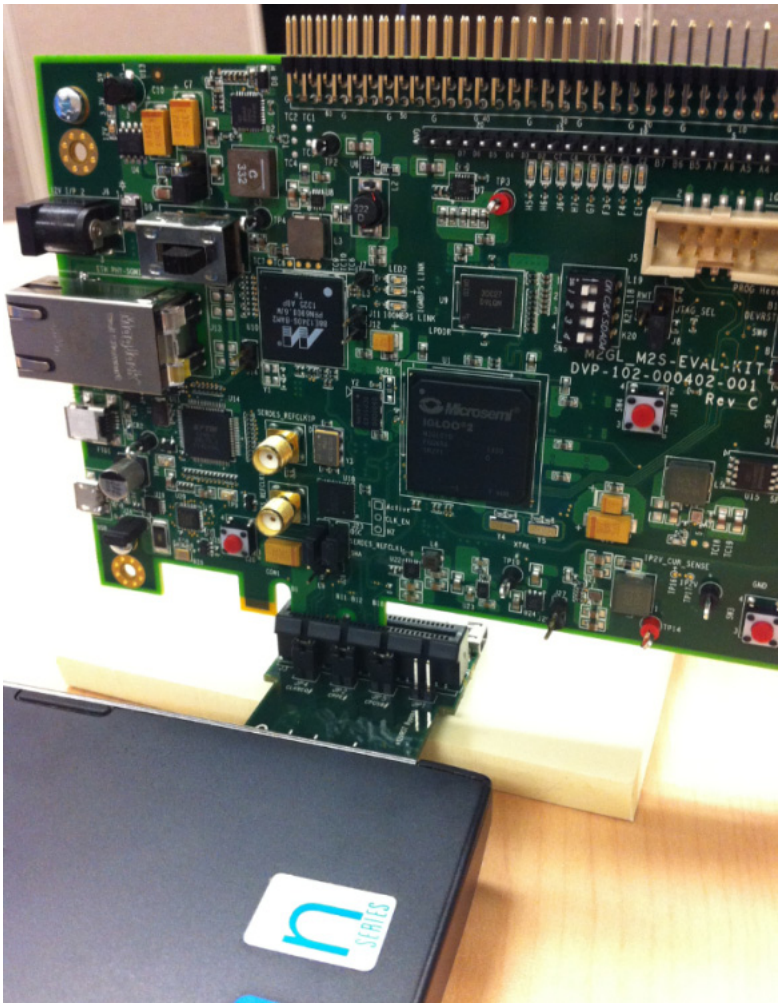


Figure 38 shows the IGLOO2 Evaluation Kit PCIe connector inserted into the PCIe adapter card slot.

Figure 38 • **Inserting the IGLOO2 Evaluation Kit PCIe Connector**

Figure 39 shows the IGLOO2 Evaluation Kit connected to the laptop.

Figure 39 • **IGLOO2 Evaluation Kit Connected to the Laptop**



8 Appendix: Implementing Workaround to Access Fabric LSRAM after IAP/ISP Program Operation

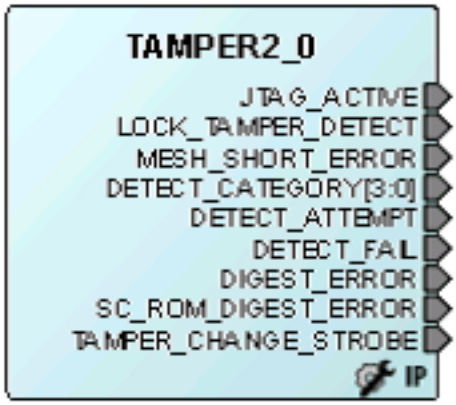
The LSRAM write and read accesses are denied after implementing the IAP program operation. The workaround for this problem is to apply System Reset after the IAP program operation. The following are the ways in which System Rest can be applied:

8.1 Option 1: Creating SmartDesign

The following steps describe how to apply System Reset:

1. Choose **File** > **New** > **SmartDesign**.
2. Enter the **Name** as **Dev_Restart_after_IAP_blk** in the **Create New SmartDesign** window.
3. Navigate to **Libero Catalog** and open **Tamper Macro**.
4. Drag the Tamper Macro (shown in Figure 40) available in the **Libero Catalog** to the **Dev_Restart_after_IAP_blk** SmartDesign canvas.

Figure 40 • **Tamper Macro – Before Configuration**



5. Select the **Enable RESET Function** check box in the **Configuring Tamper2_0** window, as shown in Figure 41.
6. Click **OK**. System Reset is enabled.

Figure 41 • **Tamper Macro Configuration Window**

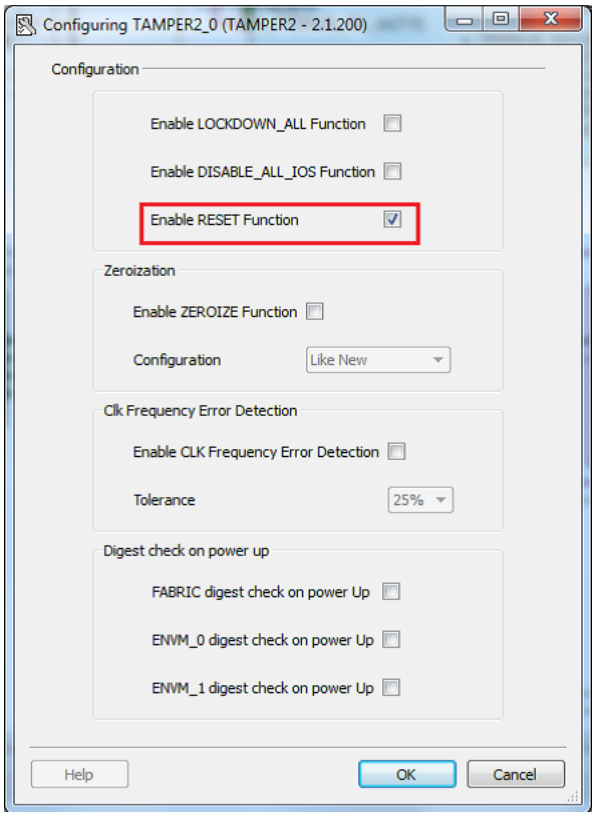
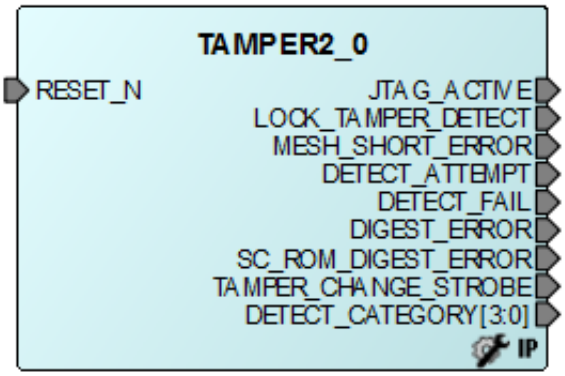


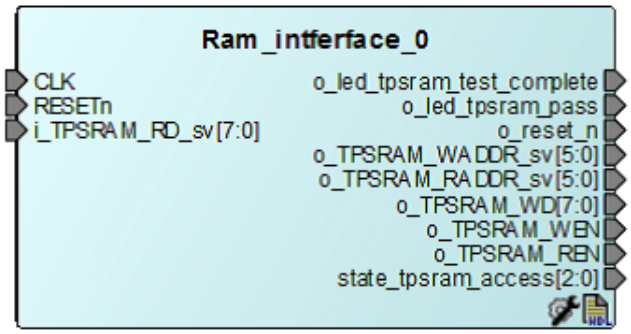
Figure 42 shows the TAMPER2_0 macro after configuration.

Figure 42 • **Tamper Macro – After Configuration**



7. Instantiate the **FSM Module** provided in the design files. This FSM logic performs three consecutive address writes to the two-port large SRAM with the known data pattern and then reads back data from the three consecutive address locations, to compare. If the read back data pattern does NOT match the written data pattern, the FSM asserts the RESET_N input to Tamper Macro, which in turn causes a System Reset. If the read back data pattern matches the written data pattern, the FSM does not do anything.
   Follow these steps to add the FSM logic to the PCIe IAP design:
   a. Choose **File** > **Import** > **HDL Source Files**.
   b. Browse to the **Ram_interface.v** file location in the design files folder <*download_-folder*>*\m2gl_dg0681_liberov11p7_df\sample_programming_files\LSRAM_Workaround\Source-files*
   c. Click the **Dev_Restart_after_IAP_blk** tab and drag the **Ram_interface** component from **Design Hierarchy** to the **Dev_Restart_after_IAP_blk** SmartDesign canvas. Figure 43 shows the **Ram_interface** component.

Figure 43 • **Ram_Interface FSM Component**

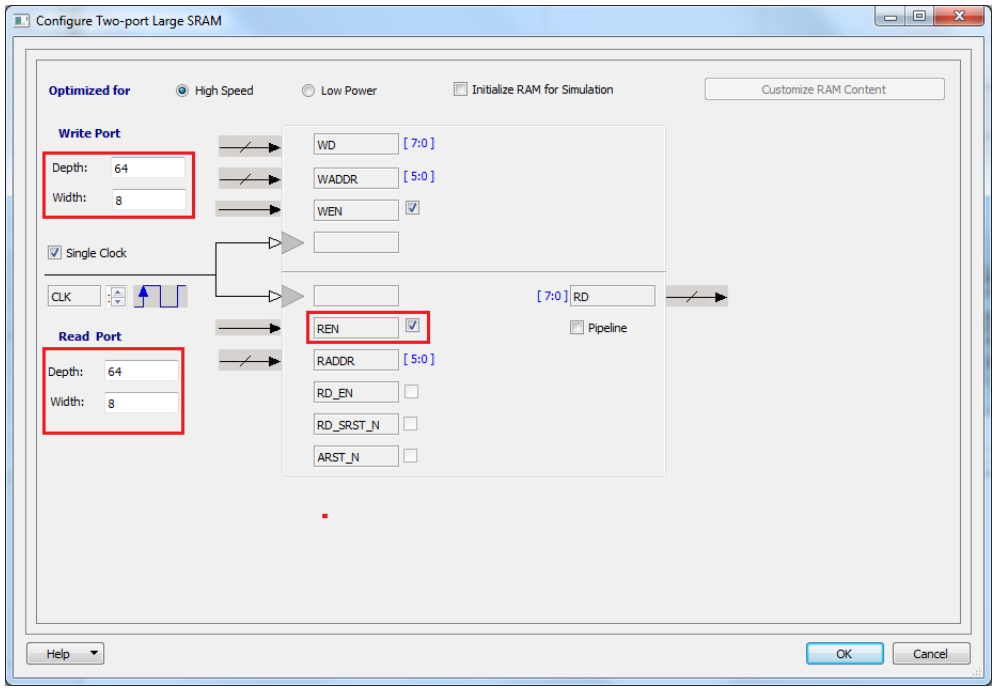


After the completion of the IAP programming, the System Controller asserts POWER_ON_RESET_N to the FPGA fabric. This triggers the RESETn signal and initiates the state machine in the FSM module.

8. Drag the **Two-Port Large SRAM (TPSRAM)** available in the **Libero Catalog** to the **Dev_Restart_after_IAP_blk SmartDesign** canvas.

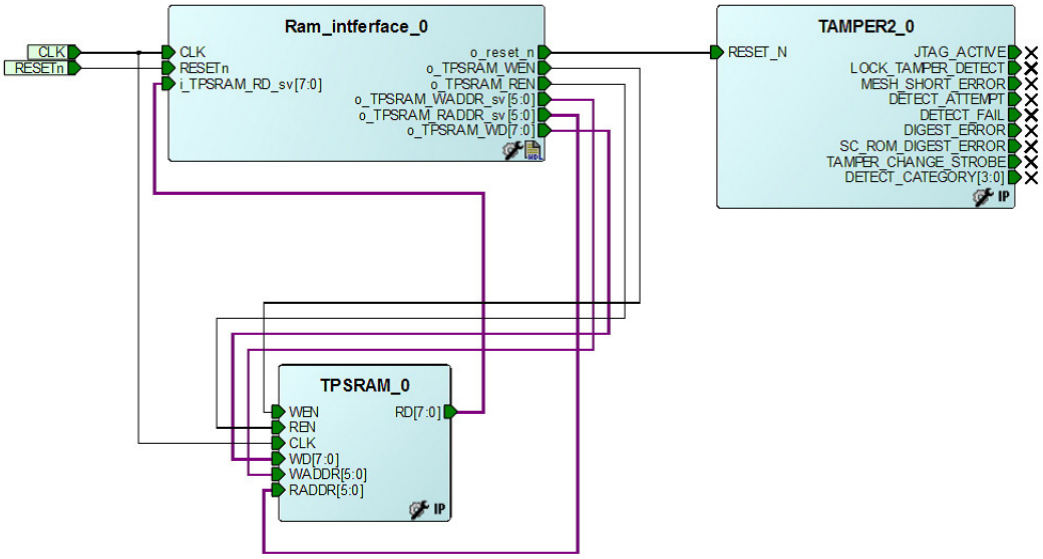
9. Configure the **TPSRAM** with the following settings:
   - Write Port
     - Depth: 64
     - Width: 8
   - Read Port
     - Depth: 64
     - Width: 8
   - Select **REN** check box

Figure 44 • **Two-Port SRAM Configurator Window**



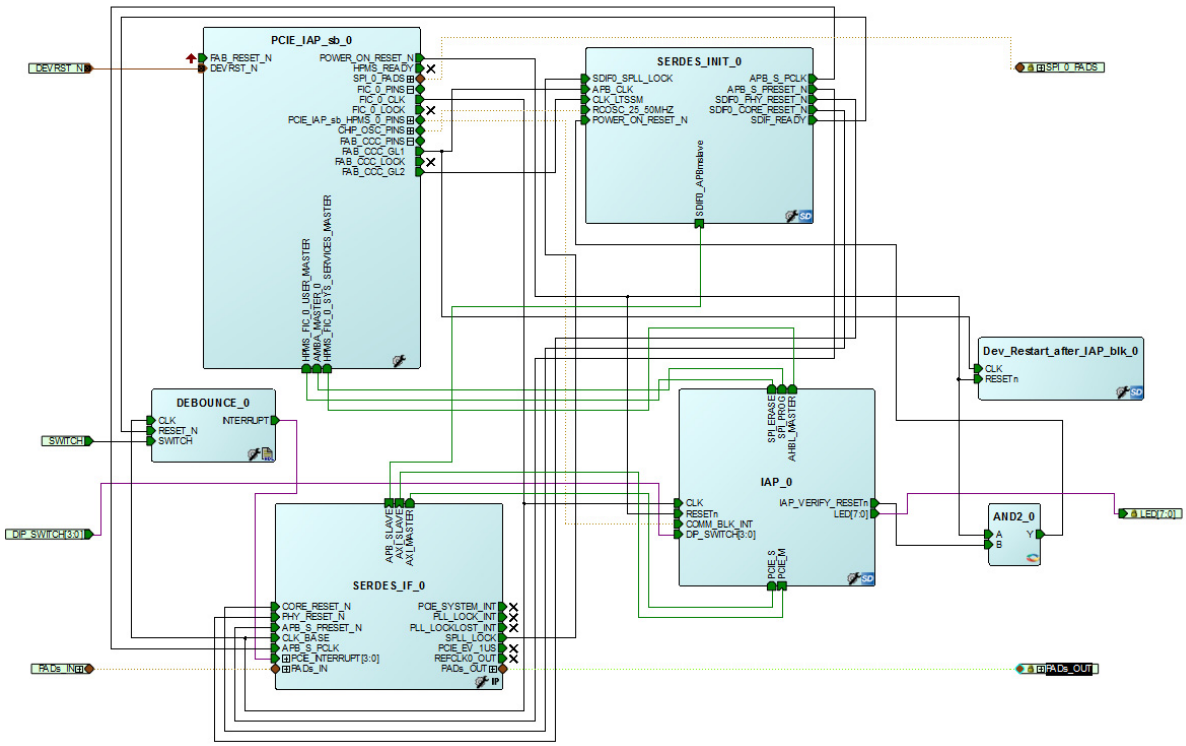
10. Connect Tamper Macro, FSM, and TPSRAM, as shown in Figure 45.

Figure 45 • **Dev_Restart_after_IAP_blk SmartDesign**



11. Click the **PCIE_IAP_top** tab and drag the **Dev_Restart_after_IAP_blk** component from the **Design Hierarchy** to the **PCIE_IAP_top SmartDesign** canvas.
12. Connect the blocks as shown in Figure 46 and generate **PCIE_IAP_top SmartDesign**. This completes the implementation of the workaround.

Figure 46 • **PCIe_IAP_top SmartDesign**



Note: This workaround is applicable for v11.7 software release or later, and must be implemented in the Libero design, which is used to generate the `.spi` programming file. Older versions of Libero may prune Tamper Macro during synthesis. To avoid pruning, one of the recommended options is to promote the DETECT_ATTEMPT signal of Tamper Macro to the top-level.

8.2 Option 2: Importing the .cxf File in Libero Design

Another option to implement this workaround is to import the `.cxf` file for SmartDesign Dev_Restart_after_IAP_blk. The `.cxf` file is provided with the design files and it has all the component instantiations and connections mentioned in "Option 1: Creating SmartDesign" section on page 48 from Step 1 to Step 6.

The following steps describe how to import the `.cxf` file:

1. Extract the files at *<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files\LSRAM_Workaround\PCIE_IAP_Tamper.rar*
2. Choose **File** > **Import** > **Others**.
3. Browse to the following **Dev_Restart_after_IAP_blk.cxf** file location in the design files folder: *<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files\LSRAM_Workaround\PCIE_IAP_Tamper\PCIE_IAP\component\work\Dev_Restart_after_IAP_blk*
4. Browse to the **Ram_interface.v** file location in the design files folder: *<download_folder>\m2gl_dg0681_liberov11p7_df\sample_programming_files\LSRAM_Workaround\Sourcefiles*.
5. Repeat Step 2 and Step 3 of Option 1 to instantiate **Dev_Restart_after_IAP_blk** in **PCIE_IAP_top** SmartDesign.

9 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.

9.1 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

9.2 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.

9.3 Technical Support

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

9.4 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.

9.5 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.

9.5.1 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.

9.5.2 My Cases

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

9.5.3 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.

9.6 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.