

DG0635
Demo Guide
Implementing Programming Recovery and
In-Application Programming Features Using
Ethernet Interface for SmartFusion2 Devices
- Libero SoC v11.8 SP1



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

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

© 2017 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.

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, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 3.0	1
1.2	Revision 2.0	1
1.3	Revision 1.0	1
2	Implementing Programming Recovery and In-Application Programming Features Using Ethernet Interface for SmartFusion2 Devices	2
2.1	Design Requirements	4
2.2	Demo Design	4
2.2.1	Demo Design Features Summary	5
2.2.2	Demo Design Description	5
2.2.3	Hardware Implementation	9
2.2.4	SoftConsole Firmware Project	10
2.3	Setting Up the Demo Design	12
2.3.1	Board Setup Snapshot	13
2.4	Running the Demo Design	14
2.5	Known Issue 1	22
2.6	Known Issue 2	22
3	Appendix 1: Board Setup for Running the Demo	23
4	Appendix 2: SmartFusion2 Security Evaluation Kit Board Jumper Locations	24
5	Appendix 3: Enable TFTP Client	25
6	Appendix 4: Running the Design in Static IP Mode	29
7	Appendix 5: Implementing Workaround to Reset the Device after Programming Recovery	32
8	Appendix 6: Implementing Workaround to Access Fabric LSRAM after IAP/ISP Program Operation	36
8.1	Changes Required in the Libero Design	36
8.1.1	Option 1: Creating SmartDesign	36
8.1.2	Option 2: Importing the .cxf file in the Libero Design	40

Figures

Figure 1	SmartFusion2 Demo Design	3
Figure 2	Directory Structure	4
Figure 3	Configure Programming Recovery - Enable Programming Recovery	5
Figure 4	Export Bitstream	6
Figure 5	SPI Directory	7
Figure 6	TFTP Server Application - Block Diagram	8
Figure 7	Libero SmartDesign	9
Figure 8	Flash*Freeze Hardware Settings	10
Figure 9	Configuring SPI_0 Ports Available During F*F	10
Figure 10	SoftConsole Project Explorer Window	11
Figure 11	Device Manager Window	13
Figure 12	FlashPro—New Project Dialog Box	14
Figure 13	Configured FlashPro Project Window	15
Figure 14	FlashPro Program Passed Window	16
Figure 15	PuTTY Window	16
Figure 16	Erasing the SPI Flash Memory Location [0x0 - 0xFFFF]	17
Figure 17	Transfer SPI Directory	17
Figure 18	SPI Directory Transferred Successfully Window	17
Figure 19	Erase the SPI Flash Memory [0x1000 - 0x3FFFFFF]	18
Figure 20	Transferring Golden Image	18
Figure 21	Successful Transfer of Golden Image Window	18
Figure 22	Erase the SPI Flash Memory 0x400000 - 0x7FFFFFF	19
Figure 23	Transferring IAP Image	19
Figure 24	IAP Image Transferred Successfully Window	19
Figure 25	IAP Application Options	20
Figure 26	IAP Program	20
Figure 27	IAP Verify Option	20
Figure 28	FlashPro PROGRAM FAILED Error Message Window	21
Figure 29	Golden Image Transferred Successfully Window	21
Figure 30	SmartFusion2 Security Evaluation Kit Setup	23
Figure 31	SmartFusion2 Security Evaluation Kit Board Jumper Locations	24
Figure 32	Control Panel - Programs and Features	25
Figure 33	Selecting TFTP Client from Windows Features	25
Figure 34	System and Security Window	26
Figure 35	Allowed Programs Window	26
Figure 36	Add a Program Window	27
Figure 37	Selecting the TFTP Executable File	27
Figure 38	TFTP.EXE Path in Add a Program Window	28
Figure 39	Selecting Trivial File Transfer Protocol App in Allowed Programs Window	28
Figure 40	Project Explorer Window SoftConsole Project	29
Figure 41	Properties for Demo_TFTP	30
Figure 42	Local Area Connection Properties Window	31
Figure 43	Internet Protocol Version 4 (TCP/IPv4) Properties	31
Figure 44	Tamper Macro	32
Figure 45	Tamper Macro Configuration Window	33
Figure 46	Tamper Macro	33
Figure 47	Clock_check HDL Component	34
Figure 48	Chip Oscillators Configurator	34
Figure 49	Program_Recovery_WA	34
Figure 50	Demo Smart Design	35
Figure 51	Tamper Macro	36
Figure 52	Tamper Macro Configuration Window	37
Figure 53	Tamper Macro	37
Figure 54	Ram_interface FSM Component	38

Figure 55	Two-Port LSRAM Configurator Window	38
Figure 56	Dev_Restart_after_IAP_blk_SmartDesign	39
Figure 57	Demo Smart Design	39

Tables

Table 1	Design Requirements	4
Table 2	SPI Flash Directory	6
Table 3	SmartFusion2 Security Evaluation Kit Jumper Settings	12

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 3.0

This document is updated for Libero SoC v11.8 SP1 release changes.

1.2 Revision 2.0

The following is a summary of the changes in revision 2.0 of this document.

- Updated the demo guide for Libero v11.8 software release changes.
- Information about Standby Clock Source Configuration and Flash*Freeze mode was added. see [Standby Clock Source Configuration](#), page 9, and [Configuring I/Os for Flash*Freeze Mode](#), page 10.

1.3 Revision 1.0

Revision 1.0 was the first publication of this document.

2 Implementing Programming Recovery and In-Application Programming Features Using Ethernet Interface for SmartFusion2 Devices

This demo design explains how to implement the Programming Recovery and In-Application Programming (IAP) features of the SmartFusion2 device. Programming Recovery feature allows the device to automatically recover from a power failure during a programming operation. When Programming Recovery option is enabled, the device is programmed with the Golden Image that is stored in the external flash memory.

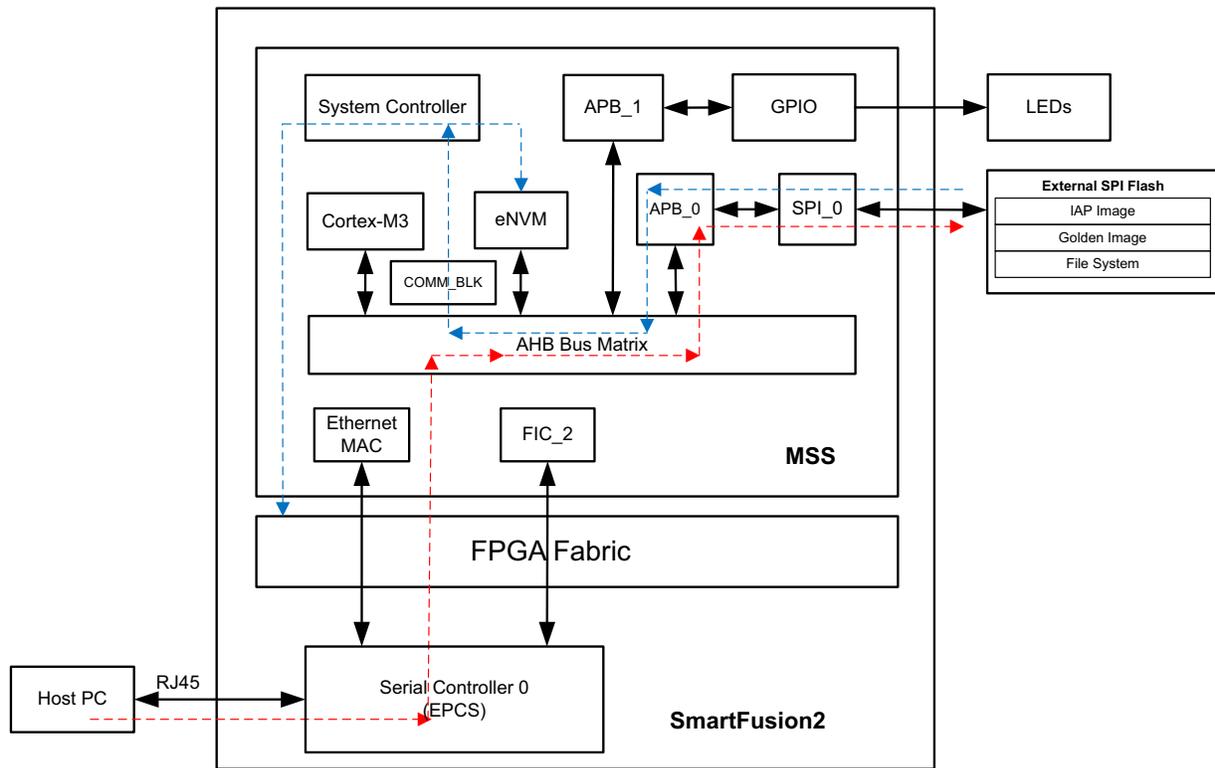
IAP is a SmartFusion2 device programming feature, which is used to reprogram the device for design iterations and field upgrades. By using the IAP feature, the application reprograms the flash components of the SmartFusion2 device. IAP is a two step process:

1. Used to transfer the image to SPI flash memory.
2. Does the programming of device.

There are different ways to transfer the IAP and Golden images from the Host PC to on-board external serial peripheral interface (SPI) flash. For example, SmartFusion2 supported microcontroller subsystem (MSS) peripherals such as universal asynchronous receiver/transmitter (UART), universal serial bus (USB), peripheral component interconnect express (PCIe), and Ethernet interfaces are used to load the images to the SPI flash memory. In this demo trivial file transfer protocol (TFTP)/Ethernet interface is used for loading SPI flash memory.

The following figure shows the data flow of the design. The arrows that are highlighted in red show the data flow between the Host PC and on-board external SPI flash memory using the Ethernet interface. The ARM Cortex-M3 processor copies the programming data from the Host PC to the SPI flash using the Ethernet interface. The arrows that are highlighted in blue show the system controller reading the data from external SPI flash memory to program the SmartFusion2 device. In this demo design, the SPI flash images contain information to program both embedded nonvolatile memory (eNVM) and FPGA fabric.

Figure 1 • SmartFusion2 Demo Design



- - - - - **Step1** Transferring data bitstream from Host PC to external Flash through Ethernet interface
- - - - - **Step2** System controller reads data bitstream from External flash to program the SmartFusion2 device

For more information about Program Recovery and IAP features of SmartFusion2, refer to the [UG0451:IGLOO2 and SmartFusion2 Programming User Guide](#).

For more information about SPI and Ethernet Interfaces, refer to the [UG0331:SmartFusion2 Microcontroller Subsystem User Guide](#).

2.1 Design Requirements

The following table lists the hardware, software, and IP requirements for this demo design.

Table 1 • Design Requirements

Design Requirements	Description
Hardware Requirements	
SmartFusion2 Security Evaluation Kit: – FlashPro4 programmer – 12 V adapter – USB A to Mini-B cable	Rev E or later
RJ45 Cable (not available in the kit)	
Host PC or Laptop	Windows 64-bit Operating System
Software Requirements	
Libero [®] System-on-Chip (SoC) for viewing the design files	v11.8 SP1
FlashPro programming software	v11.8 SP1
SoftConsole	v4.0
Host PC Drivers	USB to UART drivers
One of the following serial terminal emulation programs: – HyperTerminal – TeraTerm – PuTTY	

2.2 Demo Design

The demo design files are available for download from the following path in the Microsemi website:

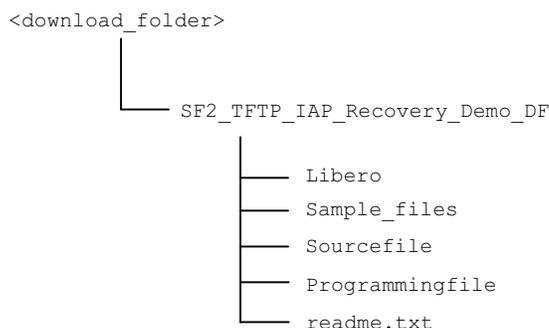
http://soc.microsemi.com/download/rsc/?f=m2s_dg0635_libero11p8_sp1_df

Design files include:

- Libero
- Sample_files
- Sourcefile
- Programmingfile
- Readme.txt file

The following figure shows the top-level structure of the design files. Refer to `readme.txt` file for the complete directory structure.

Figure 2 • Directory Structure



2.2.1 Demo Design Features Summary

The following are the demo design features:

- Program Recovery is enabled and to demonstrate this feature, Golden Image is loaded into the SPI flash memory. When a power failure scenario is created, the demo design programs the device with the Golden Image after a power failure during a programming operation.
- IAP feature is implemented. In this Demo Design following IAP services are implemented.
 - Authenticate
 - Program
 - Verify
- Transfer of programming files from the Host PC to an external SPI flash is done through Ethernet Interface using the TFTP application.
- Both eNVM and FPGA Fabric are programmed during Programming Recovery and IAP.
- LED blinking patterns and serial terminal messages are different for Golden and IAP images to indicate Programming Recovery and IAP are applied correctly.

2.2.2 Demo Design Description

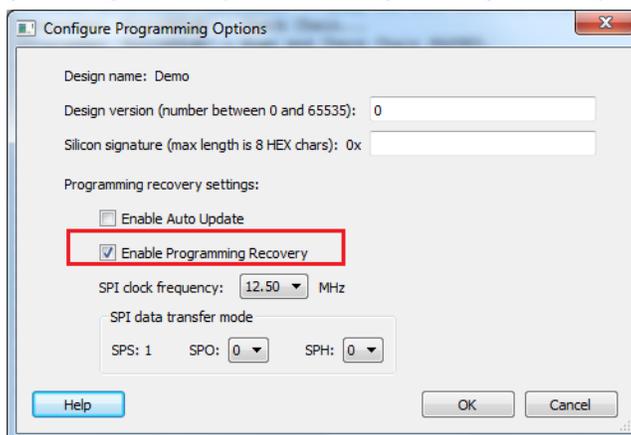
- The demo design comprises of the following features:
 - Programming Recovery
 - IAP
 - TFTP Server Application

2.2.2.1 Programming Recovery

Programming recovery, if enabled, automatically recovers from a power failure during a programming operation. The Programming Recovery option is enabled using the Libero SoC software. To enable the Programming Recovery feature, open the Libero Project and select **Configure Programming Recovery** option in **Design Flow** window.

In the **Configure Program Recovery** window, select **Enable Programming Recovery** check box as shown in the following figure.

Figure 3 • Configure Programming Recovery - Enable Programming Recovery



Based on the content provided in the Recovery programming file, three types of programming are possible:

- **eNVM programming:** Contains only eNVM content.
- **FPGA Fabric programming:** Contains only the FPGA fabric content.
- **eNVM and FPGA Fabric programming:** Contains both the FPGA fabric and eNVM content.

The Program Recovery image provided with design files in this demo has both eNVM and fabric content.

The Programming Recovery option requires an external SPI flash to be connected to MSS SPI_0. External SPI flash needs to be loaded with a SPI directory, Golden Image. The SPI directory provides the address of the Golden image and its design version as listed in the following table.

Table 2 • SPI Flash Directory

Offset	Name	Size (Bytes)	Description
0	GOLDEN_IMAGE_ADDRESS	[3:0]	Address where the golden image starts.
4	GOLDEN_IMAGE_DESIGNVER	[1:0]	Design version of the golden image.

Create a directory in the SPI flash to provide the address for Golden Image and design version. On the SmartFusion2 Security Evaluation Kit board, the flash memory size is 8 MB. Depending on the size of the programming file, the SPI flash memory is configured in the SoftConsole project as follows:

- 0x0 - 0xFFFF is for loading SPI Directory
- 0x1000 - 0x3FFFFFF is for loading Golden Image
- 0x400000 - 0x7FFFFFF is for loading IAP Image

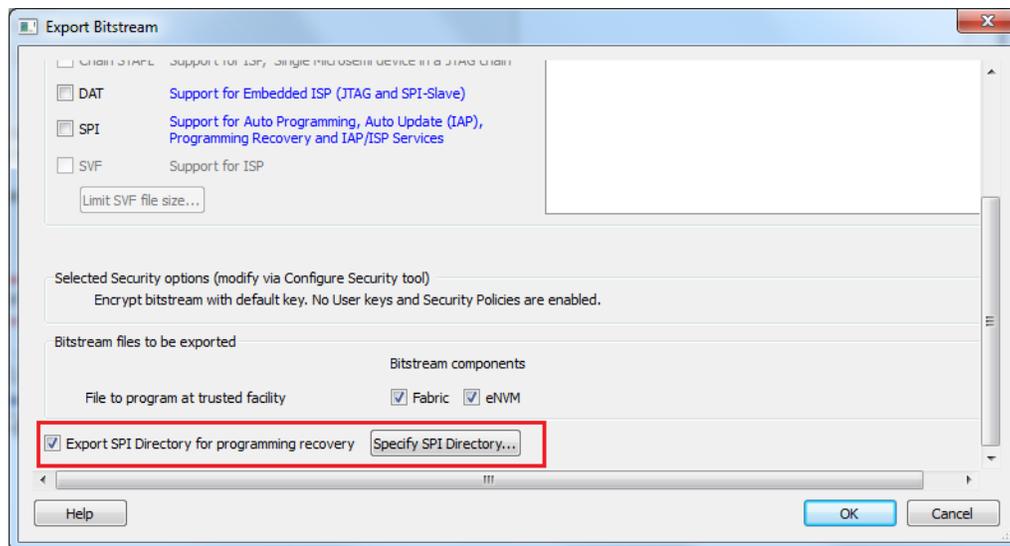
Note: All these addresses are configured in the SoftConsole project and same address needs to be entered while creating the SPI directory. Using TFTP/Ethernet SPI Directory, Golden image, and IAP images are transferred to SPI addresses 0x0, 0x1000, and 0x400000 respectively.

2.2.2.1.1 Creating SPI Directory

The following steps describe how to create an SPI directory:

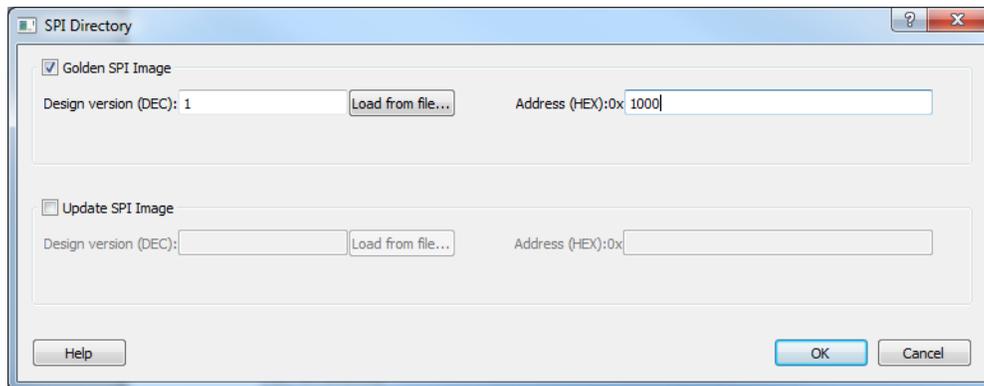
1. To create SPI directory, open Libero project and select **Export Bit Stream** from **Design Flow** window. The **Export Bitstream** window is displayed as shown in the following figure.
2. Select **Export SPI Directory for programming recovery** check-box and click **Specify SPI Directory**.

Figure 4 • Export Bitstream



The **SPI Directory** window is displayed as shown in the following figure. Enter the design version and address for Golden Image. Golden image address, which is configured in the SoftConsole project needs to be entered in the SPI Directory window.

Figure 5 • SPI Directory



Design version size is 16-bit. Address indicates the starting address of the Golden image and address field size is 32-bit. Using the TFTP/Ethernet SPI Directory, Golden Image, and IAP images are transferred to SPI addresses 0x0, 0x1000, and 0x400000 respectively. Golden and IAP images are provided with the `.spi` file extension.

The system controller reads the Golden Image address from SPI directory and programs the Golden Image available at 0x1000, in case of power failure during programming operation.

2.2.2.2 IAP

The IAP feature in SmartFusion2 devices is a two-step process.

Step1: Loading SPI Flash with Programming Image

The Cortex-M3 processor receives the bitstream data from the Host PC through the Ethernet interface and writes to the external SPI flash memory connected to the MSS SPI_0 controller.

Step 2: Initiating the IAP Service

The bitstream data is first verified by requesting the AUTHENTICATE IAP service from the system controller.

The system controller reads the bitstream data from 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 Cortex-M3 processor initiates a PROGRAM IAP system controller system service. The system controller fetches the bitstream data from the SPI flash and programs the flash components of the SmartFusion2 device. Programming can be done for the FPGA fabric, eNVM, or both FPGA fabric and eNVM.

The system controller executes the IAP services in the following modes:

Authenticate: 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 the Authenticate Operation mode. The SmartFusion2 device application must commit the bitstream for programming, only after successful authentication and the integrity of the bitstream is validated.

Program: Depending on the programming bitstream, system controller IAP service programs the following:

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

IAP Image provided with design files in this demo has both eNVM and fabric content.

Verify: System controller IAP service verifies the content of the SmartFusion2 device against the programming bitstream data that is stored in the SPI flash memory.

Note: The FPGA fabric is not operational during Program or Verify operations as the device enters into the Flash*Freeze (F*F) mode.

2.2.2.3 TFTP Server Application

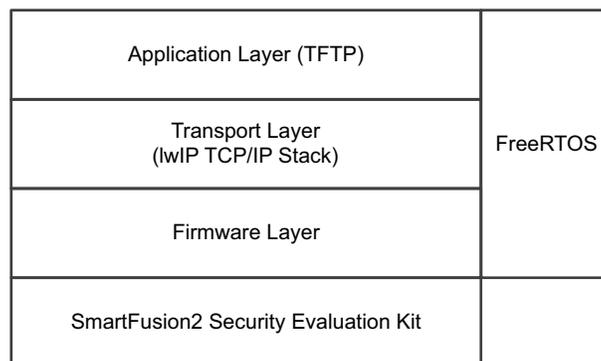
To transfer the Programming Images (.spi format) and load them to the SPI flash memory, the TFTP is used. The TFTP server application is implemented in the firmware project to transfer the SPI images from the Host PC to the external flash memory that is, available on the SmartFusion2 Security Evaluation Kit board.

The TFTP server application has the following layers:

- Application Layer
- Transport Layer
- Firmware Layer

The following figure shows the block diagram of the TFTP server application on the SmartFusion2 device used in this demo design.

Figure 6 • TFTP Server Application - Block Diagram



2.2.2.3.1 Application Layer

The TFTP protocol is implemented in the application layer. TFTP is used to transfer the files between client and server. A file transfer is initiated by the client issuing a request to read or write a particular file on the server.

The TFTP client (Host PC) transfers the file using TFTP PUT command to the SmartFusion2 device (TFTP server). Transferred files are stored in the external flash memory connected to the MSS SPI_0 on the SmartFusion2 Security Evaluation Kit board.

2.2.2.3.2 Transport Layer (lwIP TCP/IP Stack)

The lwIP stack is suitable for the embedded systems because of its less resource usage. It is used with or without the operating system. The lwIP consists of the actual implementations of the IP, ICMP, UDP, and TCP protocols, as well as the support functions such as buffer and memory management.

For more information on the design and implementation, refer to www.sics.se/~adam/lwip/doc/lwip.pdf.

2.2.2.3.3 FreeRTOS and Firmware Layer

FreeRTOS is an open source real time operating system kernel. FreeRTOS is used in this demo to prioritize and schedule the tasks. For more information about FreeRTOS and the latest source code, refer to <http://www.freertos.org>.

The firmware provides the software driver implementation to configure and control the following MSS components:

- Ethernet MAC
- MMUART

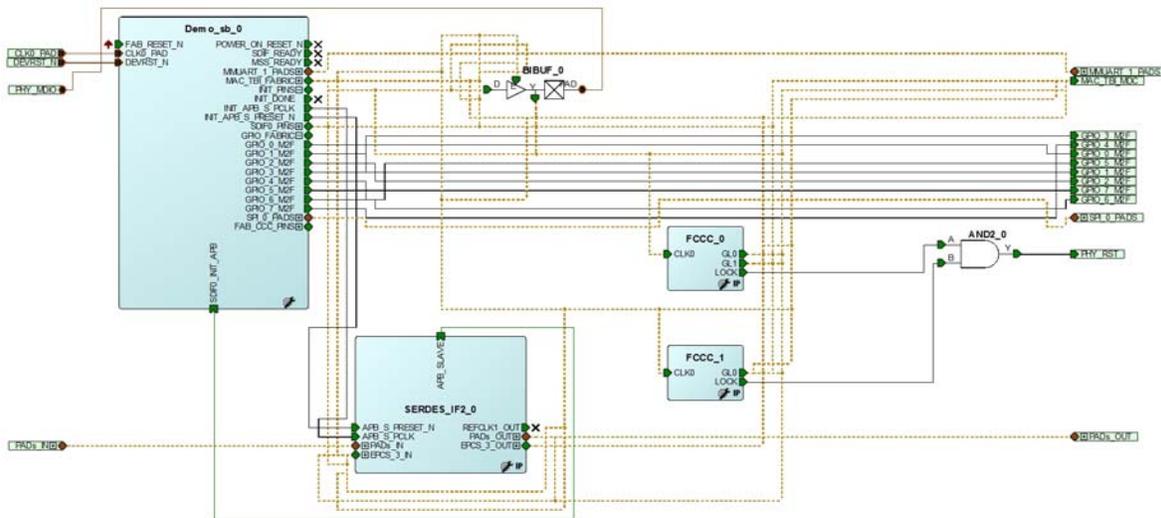
- GPIO
- SPI
- RTC
- System services

2.2.3 Hardware Implementation

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

- MSS GPIO block is enabled and configured as: GPIO_0 to GPIO_7 as outputs and connected LEDs.
- M3_CLK clock is configured to 100 MHz.
- MSS SPI_0 controller is enabled to access the external SPI flash memory.
- MMUART1 is enabled for RS-232 communication on the SmartFusion2 Security Evaluation Kit board.
- The MSS TSEMAC is configured for the ten-bit interface (TBI) operation.
- The SERDES_IF_2 in the SmartFusion2 device is configured for EPCS Lane3.

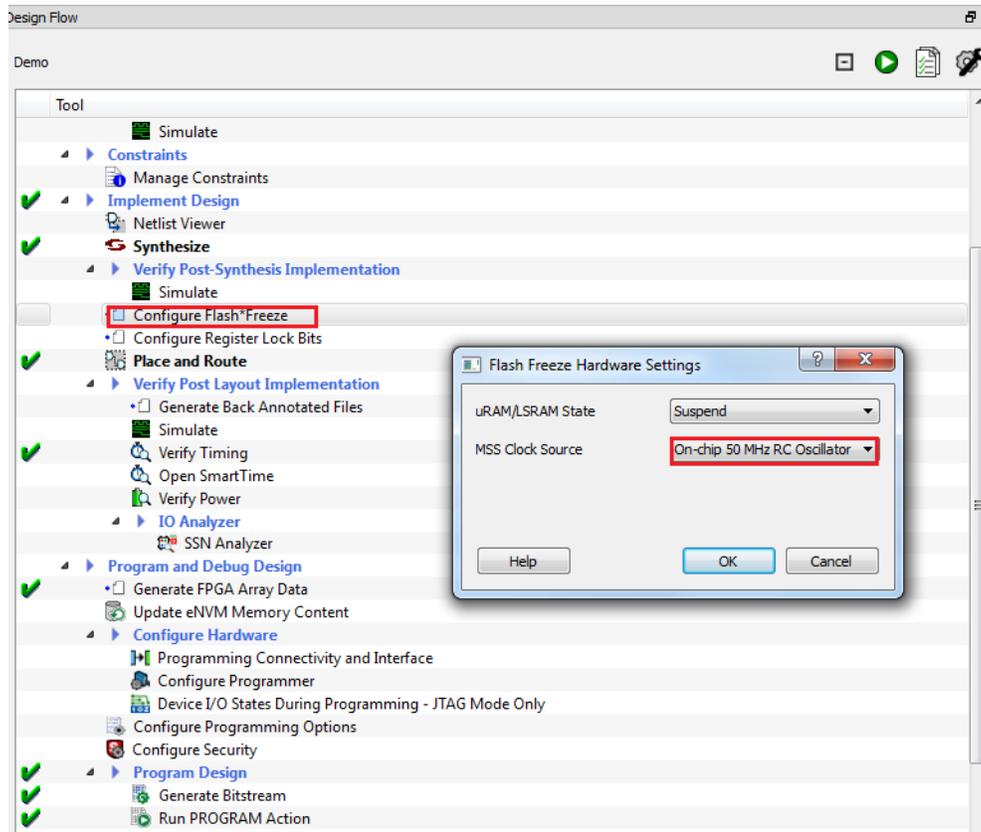
Figure 7 • Libero SmartDesign



2.2.3.1 Standby Clock Source Configuration

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

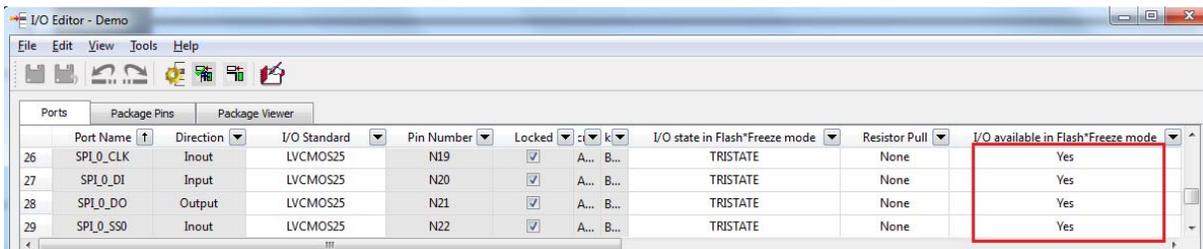
Figure 8 • Flash*Freeze Hardware Settings



2.2.3.2 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 the Flash*Freeze (F*F) mode. On the SmartFusion2 Security 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 the F*F mode, the fabric and I/Os are not available. Therefore, all the SPI_0 ports are configured using the I/O Editor to be available during the F*F mode, as shown in the following figure. Commit and Check the settings from the File menu after configuring the SPI_0 ports.

Figure 9 • Configuring SPI_0 Ports Available During F*F



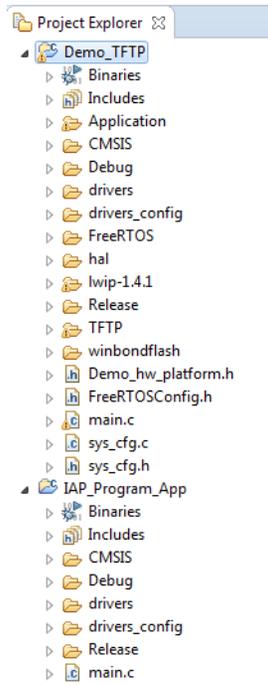
2.2.4 SoftConsole Firmware Project

The following stacks are used for this demo design:

- lwIP TCP/IP stack version 1.4.1
- FreeRTOS

The following figure shows the SoftConsole software directory structure of the demo design.

Figure 10 • SoftConsole Project Explorer Window



The SoftConsole workspace consists of the following projects.

- **Demo_TFTP:** Contains TFTP server application using LWIP and FreeRTOS. The SoftConsole project transfers only the SPI directory, Golden, and IAP Images to the SPI Flash memory using Ethernet Interface.
- Contains all the firmware and hardware abstraction layers that correspond to the hardware design.
- **IAP_Program_app:** Copies the IAP program code to the eSRAM and runs it from embedded SRAM (eSRAM).

2.3 Setting Up the Demo Design

The following steps describe how to setup the hardware demo for the SmartFusion2 Security Evaluation Kit board:

1. Connect the jumpers on the SmartFusion2 Security Evaluation Kit board, as shown in the following figure. The following figure shows the jumper settings.

Table 3 • SmartFusion2 Security Evaluation Kit Jumper Settings

Jumper	Pin From	Pin To	Comments
J22, J23, J24, J8, J3	1	2	Default These are the default jumper settings of the SmartFusion2 Security Evaluation Kit board. Ensure that these jumpers are set properly.

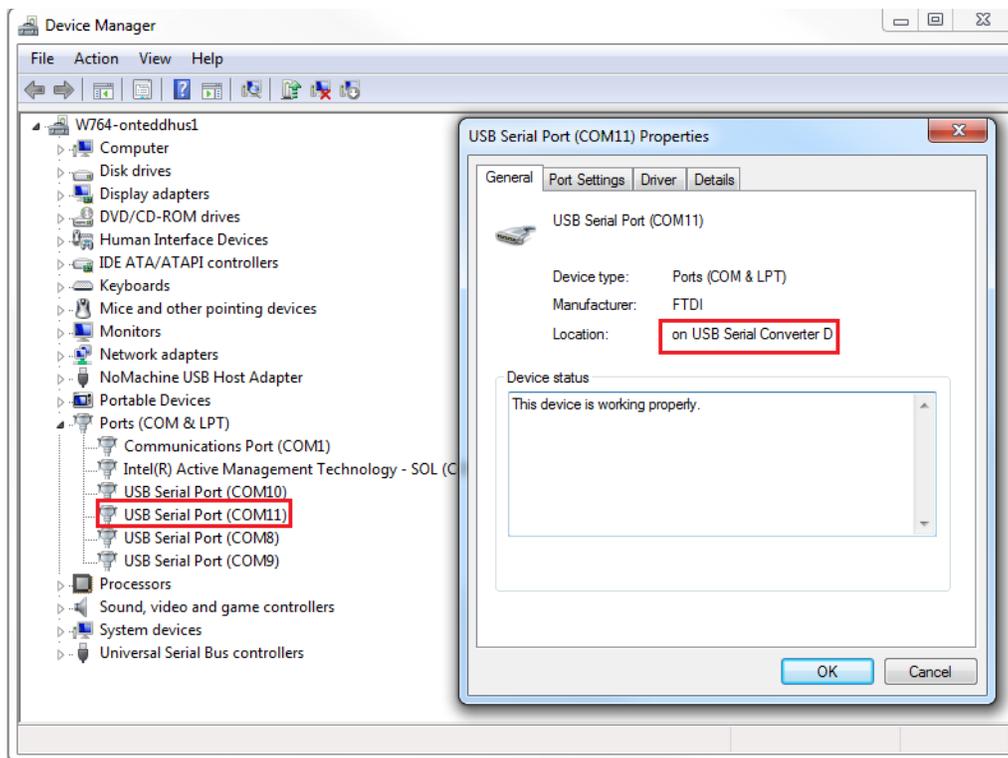
[Appendix 2: SmartFusion2 Security Evaluation Kit Board Jumper Locations](#), page 24 provides the SmartFusion2 Security Evaluation Kit silk screen to identify the jumper locations on the board.

Note: Ensure that the power supply switch, **SW7** is switched OFF while connecting the jumpers on the SmartFusion2 Security Evaluation Kit board.

2. Connect the Host PC to the **J18** connector using the USB Mini-B cable. The USB to UART bridge drivers are automatically detected.
3. From the detected four COM ports, right-click any one of the COM ports and select **Properties**. The selected COM port properties window is displayed, as shown in the following figure.

Ensure to have the Location as **on USB Serial Converter D** in the **Properties** window, as shown in the following figure.

Figure 11 • Device Manager Window



4. Install the USB driver, if the USB drivers are not detected automatically.
5. Install the FTDI D2XX driver for serial terminal communication through the FTDI mini USB cable. Download the drivers and installation guide from:
www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip
6. Connect the FlashPro4 programmer to the J5 connector of the SmartFusion2 Security Evaluation Kit board.
7. Connect the power supply to the J6 connector of the SmartFusion2 Security Evaluation Kit.
8. This design example can be run in both Static IP and Dynamic IP modes. By default, the programming files are provided for Dynamic IP mode.
 - For Static IP, connect the Host PC to the J13 connector of the SmartFusion2 Security Evaluation Kit board using an RJ45 cable.
 - For Dynamic IP, connect any one of the open network ports to the J13 connector of the SmartFusion2 Security Evaluation Kit board using an RJ45 cable.

2.3.1 Board Setup Snapshot

Snapshots of the SmartFusion2 Security Evaluation Kit board with all the setup made is given in [Appendix 1: Board Setup for Running the Demo](#), page 23.

2.4 Running 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_dg0635_liberov11p8_sp1_df
2. Switch **ON** the SW7 power supply switch.
3. Start any serial terminal emulation program such as:
 - HyperTerminal
 - PuTTY
 - TeraTerm

Note: In this demo PuTTY is used.

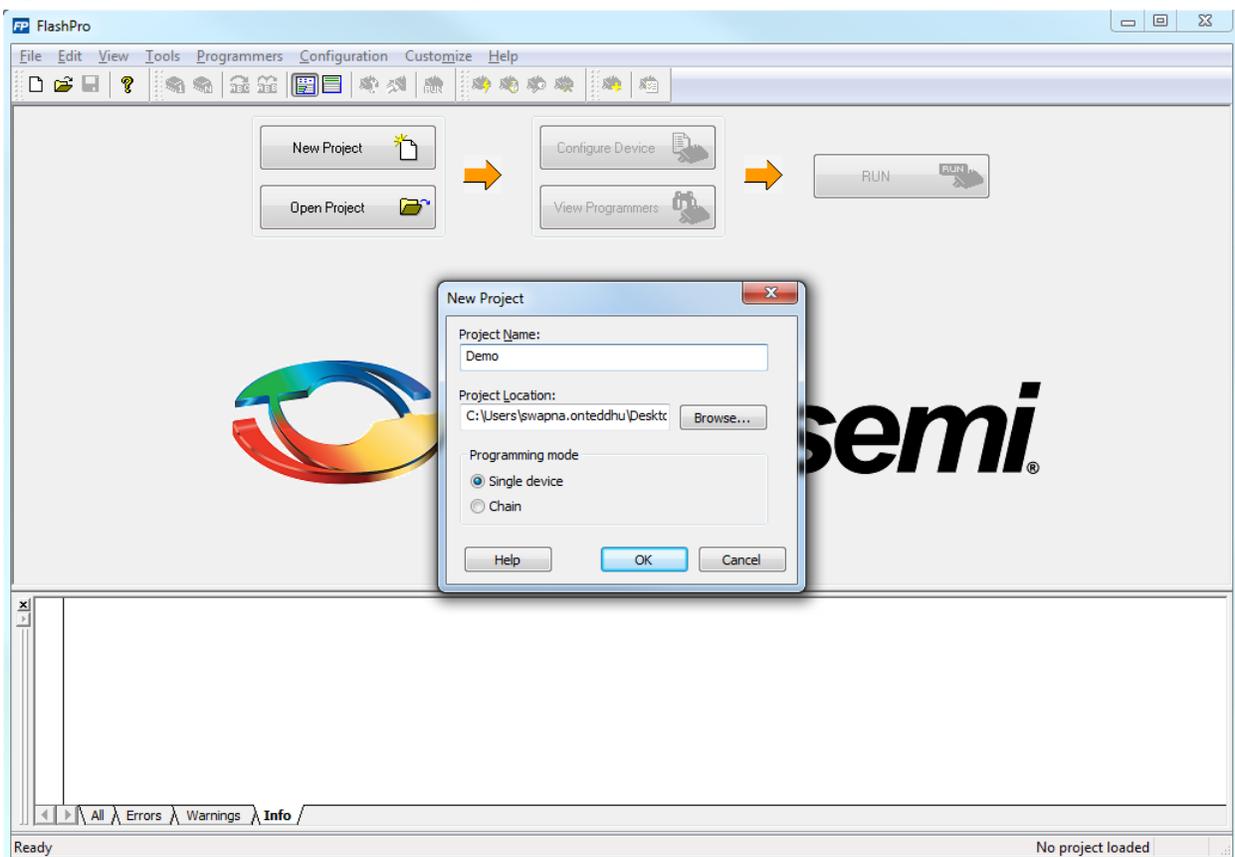
The configuration for the program is:

- Baud Rate: 115200
- Eight data bits
- One stop bit
- No Parity
- No flow control

For more information about how to configure the serial terminal emulation programs, refer to the [Configuring Serial Terminal Emulation Programs Tutorial](#).

4. Launch the FlashPro software.
5. Click **New Project**.
6. In the **New Project** window, enter the Project Name.

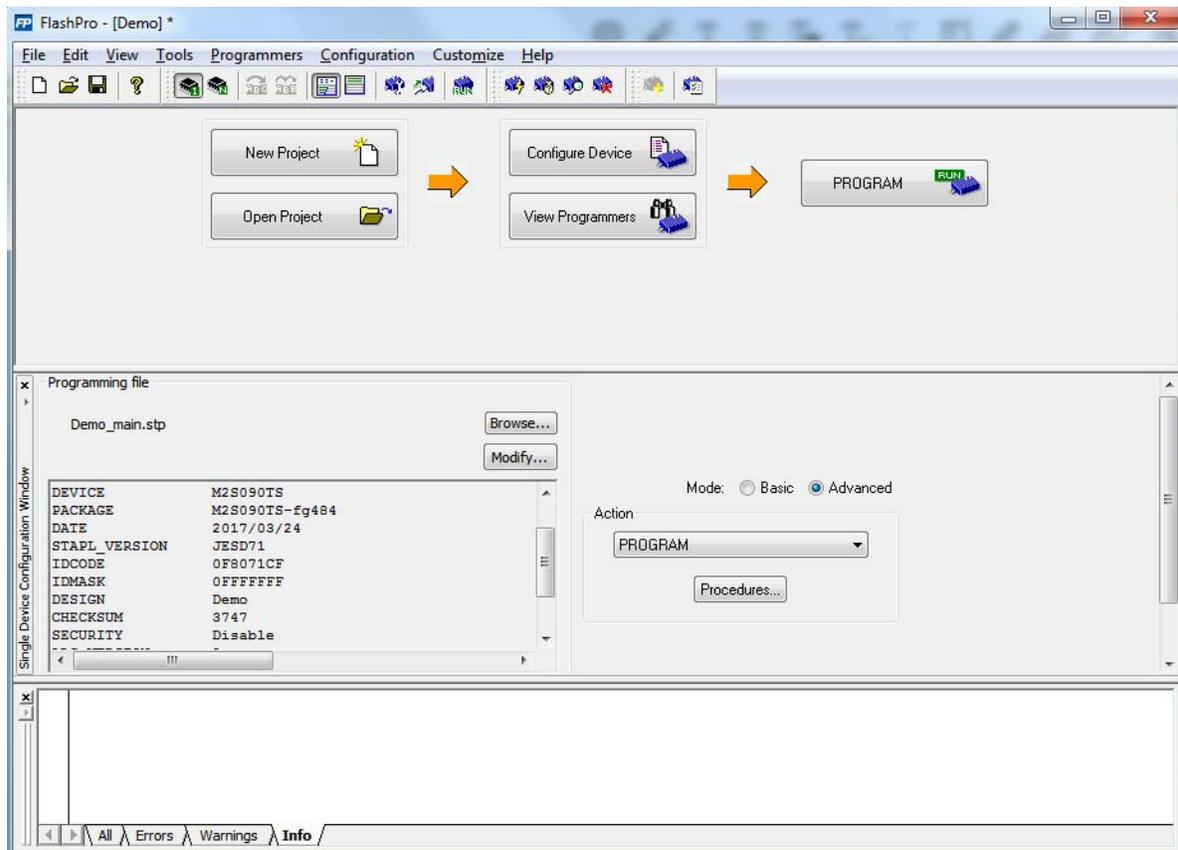
Figure 12 • FlashPro—New Project Dialog Box



7. Click **Browse** and navigate to the location where the project needs to be saved.
8. Select **Single device** as the **Programming mode**.
9. Click **OK** to save the project.

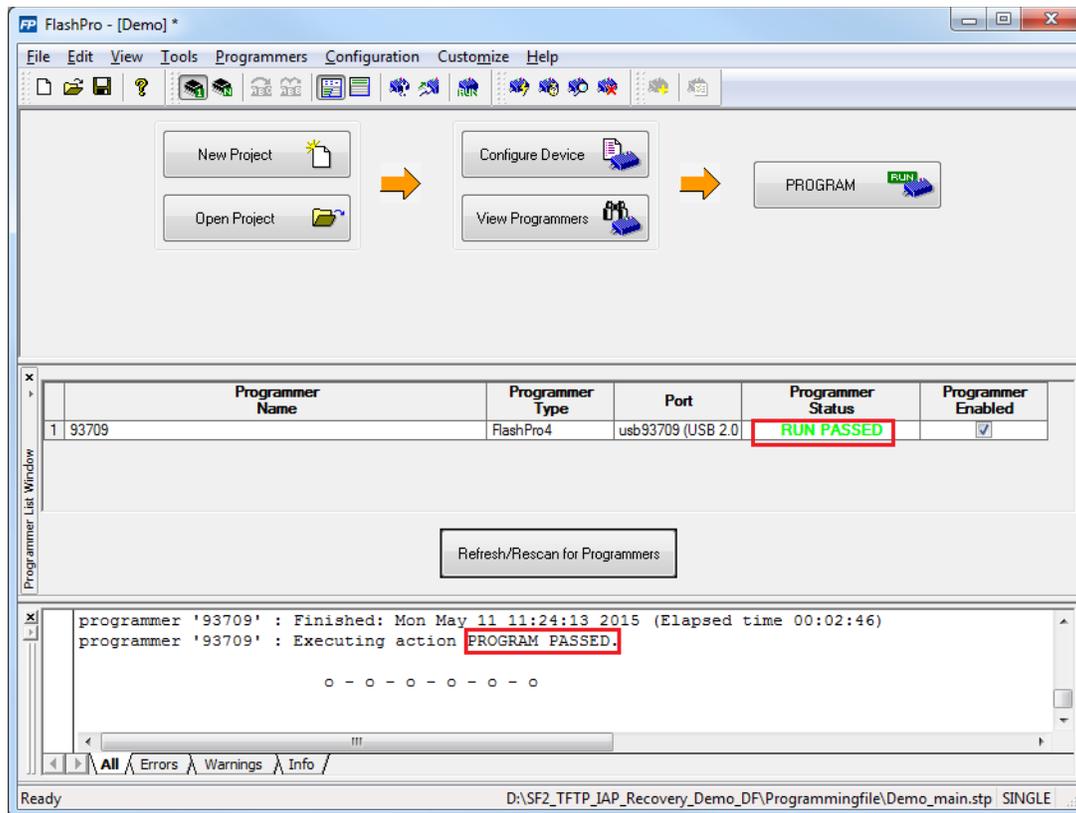
10. Click **Configure Device**.
11. Click **Browse** and navigate to the location where the `Demo_main.stp` file is located, and select the file. The default location is:
<download_folder>\SF2_TFTP_IAP_Recovery_Demo_DF\ProgrammingFile\Demo_main.stp. The required programming file is selected and is ready to be programmed in the device.
12. Click **Open** in **Load Programming File** dialog.

Figure 13 • Configured FlashPro Project Window



13. Click **PROGRAM** to start programming the device. Wait until the Programmer Status is changed to **RUN PASSED**, as shown in the following figure.

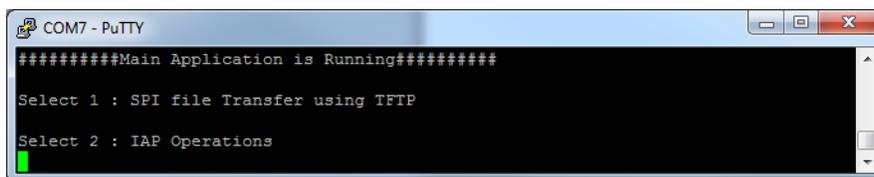
Figure 14 • FlashPro Program Passed Window



Note: The demo can be run in both Static and Dynamic IP modes. To run the design in Static IP mode, refer to [Appendix 4: Running the Design in Static IP Mode](#), page 29.

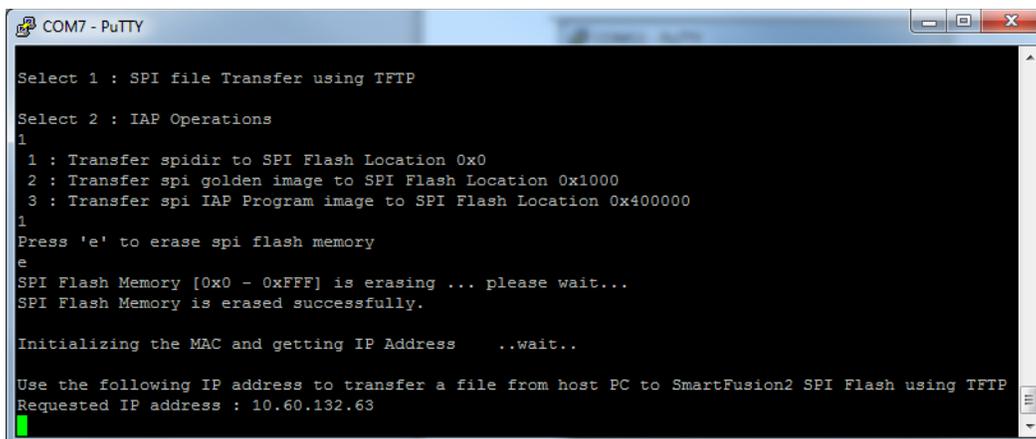
PuTTY displays the message as shown in the following figure.

Figure 15 • PuTTY Window



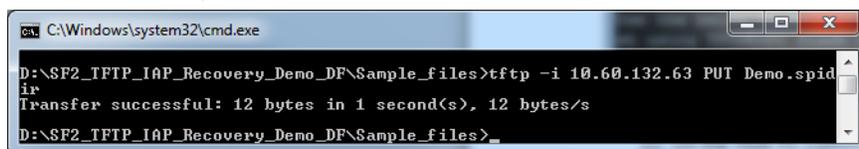
14. Enable TFTP client in the Host PC. To enable the TFTP client in Host PC, refer to [Appendix 3: Enable TFTP Client](#), page 25.
15. Enter **1** to initiate SPI directory transfer.
16. Enter **e** to erase the SPI flash memory location (0x0 – 0xFFFF).

Figure 16 • Erasing the SPI Flash Memory Location [0x0 - 0xFFFF]



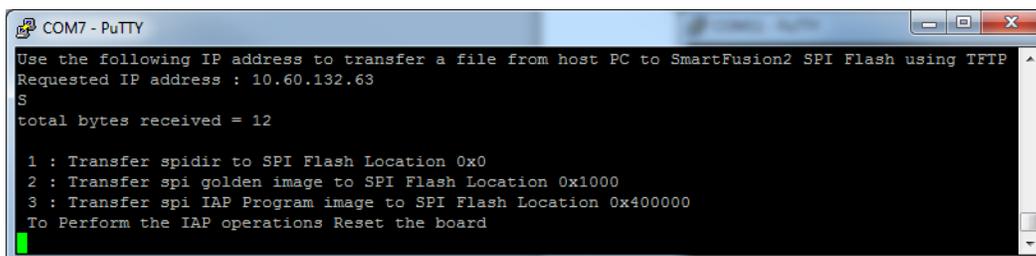
17. After completion of SPI flash erase operation, the Ethernet link is up and the IP address is displayed on the PuTTY terminal. The LED 2 on the SmartFusion2 Security Evaluation Kit board starts blinking.
18. On the Host PC command prompt, browse to the folder:
<downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sample_files
19. Type the following command to transfer the SPI Directory as shown in the following figure.
tftp -i 10.60.132.63 PUT Demo.spidir

Figure 17 • Transfer SPI Directory



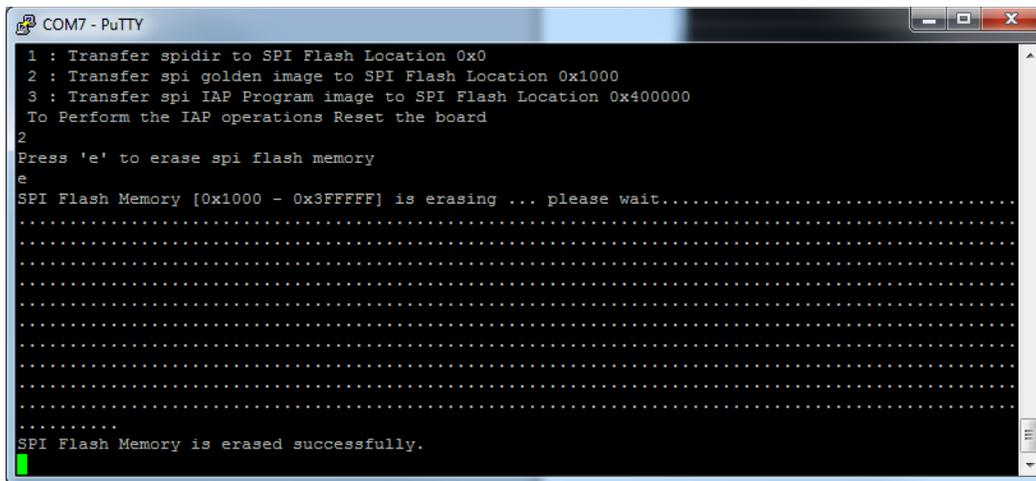
20. Wait until total bytes received message is displayed on the PuTTY terminal, to ensure that the SPI directory TFTP transferred to SPI Flash.

Figure 18 • SPI Directory Transferred Successfully Window



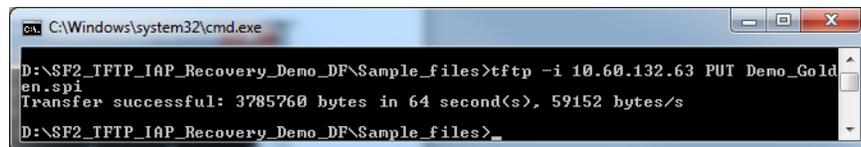
Select Option 2 and enter e to erase the SPI flash memory location [0x1000 – 0x3FFFFFF] for transferring the Golden Image to the address 0x1000.

Figure 19 • Erase the SPI Flash Memory [0x1000 - 0x3FFFFFF]



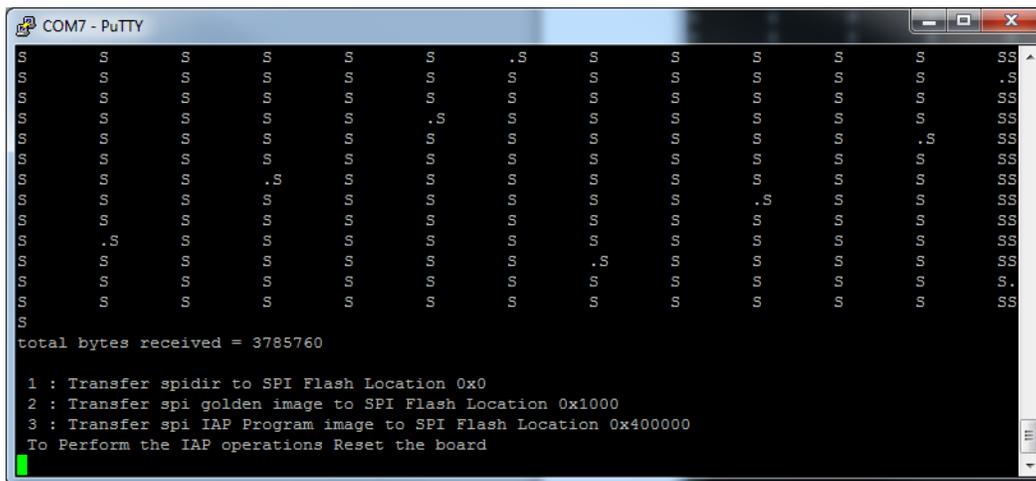
1. On the Host PC command prompt, browse to the folder:
`<downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sample_files`
2. Type the following command to transfer the Golden Image to 0x1000 memory location of SPI Flash.
`tftp -i 10.60.132.63 PUT Demo_Golden.spi`

Figure 20 • Transferring Golden Image



Wait until “total bytes received” message is displayed on the PuTTY terminal, to ensure Golden Image TFTP transfer to SPI Flash is completed.

Figure 21 • Successful Transfer of Golden Image Window

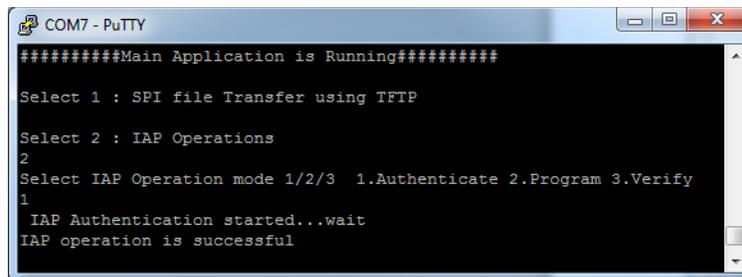


Select **3** to start the IAP program image transfer. Select **e** to erase the SPI flash memory [0x400000 - 0x7FFFFFFF] for transferring the IAP image to the address 0x400000.

To run the IAP operations, select **SW6** to reset the SmartFusion2 Security Evaluation Kit. Select **2** to run the IAP operations.

1. Select **1** to perform the IAP authentication and wait for an IAP successful message.

Figure 25 • IAP Application Options



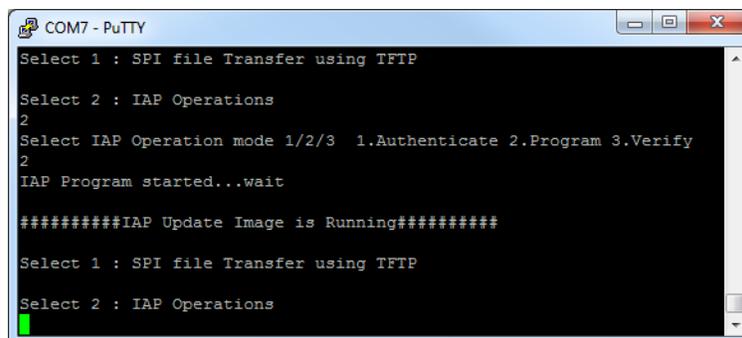
```
COM7 - PuTTY
#####Main Application is Running#####
Select 1 : SPI file Transfer using TFTP
Select 2 : IAP Operations
2
Select IAP Operation mode 1/2/3  1.Authenticate 2.Program 3.Verify
1
IAP Authentication started...wait
IAP operation is successful
```

2. Select **2** to perform the IAP Operations.
3. Select **2** to run the IAP Program.

Note: During IAP Program process, LEDs (H5, H6, J6, H7, F3, F4, and E1) on the SmartFusion2 Security Evaluation Kit are turned OFF.

4. After successful execution of the IAP Program option, the PuTTY terminal displays a message “IAP Update Image is Running” and also ensure LEDs (H5, H6, J6, and H7) on the SmartFusion2 Security Evaluation Kit are ON.

Figure 26 • IAP Program

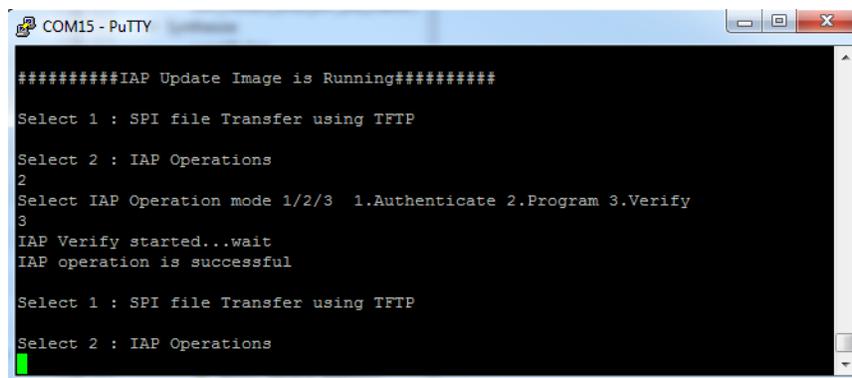


```
COM7 - PuTTY
Select 1 : SPI file Transfer using TFTP
Select 2 : IAP Operations
2
Select IAP Operation mode 1/2/3  1.Authenticate 2.Program 3.Verify
2
IAP Program started...wait
#####IAP Update Image is Running#####
Select 1 : SPI file Transfer using TFTP
Select 2 : IAP Operations
```

5. Select **2** to perform IAP operations.
6. Select **3** to perform IAP Verify.

Note: During IAP Verify process, LEDs (H5, H6, J6, H7, F3, F4, and E1) on the SmartFusion2 Security Evaluation Kit board are turned OFF.

Figure 27 • IAP Verify Option



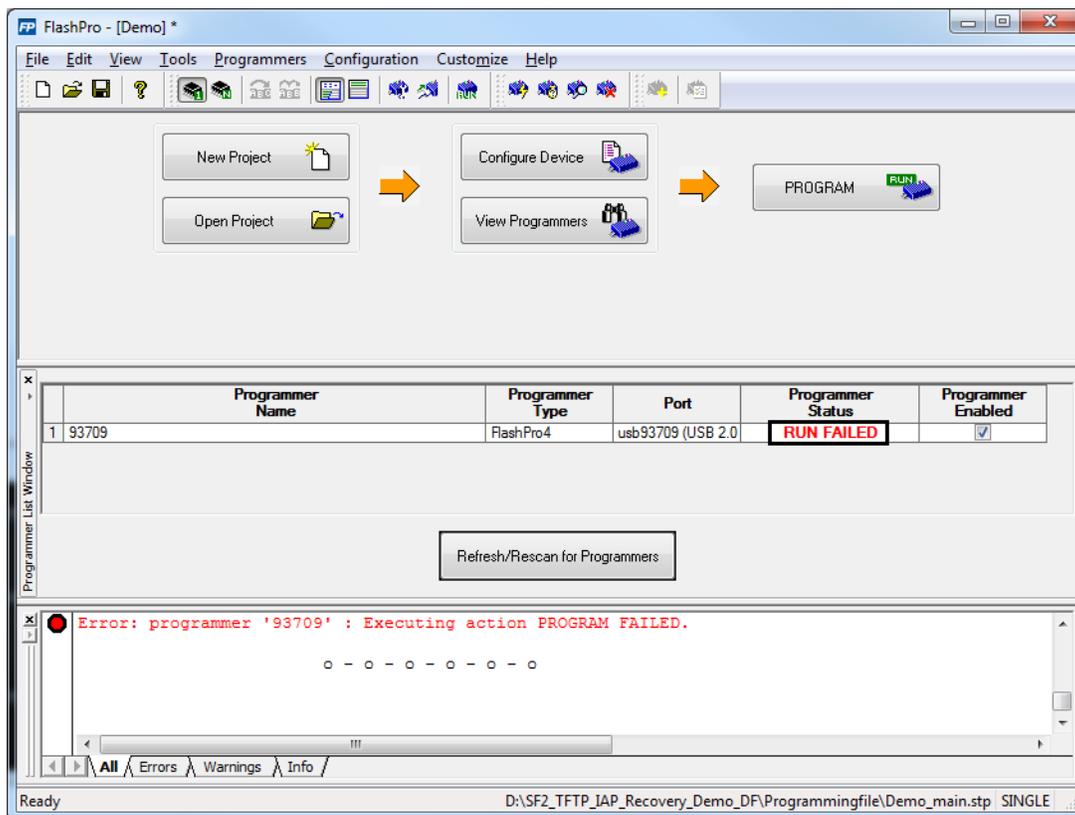
```
COM15 - PuTTY
#####IAP Update Image is Running#####
Select 1 : SPI file Transfer using TFTP
Select 2 : IAP Operations
2
Select IAP Operation mode 1/2/3  1.Authenticate 2.Program 3.Verify
3
IAP Verify started...wait
IAP operation is successful
Select 1 : SPI file Transfer using TFTP
Select 2 : IAP Operations
```

7. After successful execution of the IAP Verify option, ensure LEDs (H5, H6, J6, and H7) on the SmartFusion2 Security Evaluation Kit board are ON.

In order to run the Programming Recovery option, create a scenario where the programming operation meets a power failure. One of the ways to perform this task is to start programming the device using the FlashPro programming tool. Ensure to switch OFF the power supply to the board before programming operation is completed by FlashPro programmer. After the power failure situation is created, then the FlashPro programmer can be removed. The SmartFusion2 system controller performs Programming Recovery operation, as described below:

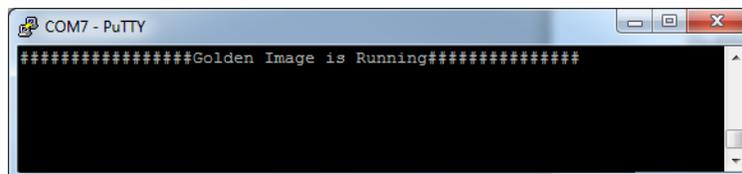
1. Run FlashPro software and select the same `Demo_main.stp` file provided in the Programming File folder.
2. After selecting the `Demo_main.stp` file for programming in FlashPro, click **PROGRAM**.
3. While the programming is going ON, observe LEDs (H5, H6, J6, H7, F3, F4, and E1) are turned OFF on the SmartFusion2 Security Evaluation Kit board and after seeing LEDs turn OFF, immediately switch OFF the board using SW7.
4. FlashPro programmer must stop programming the device and display an error message, as shown in the following figure.

Figure 28 • FlashPro PROGRAM FAILED Error Message Window



5. Switch ON the SmartFusion2 Security Evaluation Kit board and wait for approximately 4 to 5 minutes for the program recovery. After successful program recovery, the PuTTY terminal displays a message **Golden Image is Running** and ensure the LEDs (H5, J6, G7, F4) are ON.

Figure 29 • Golden Image Transferred Successfully Window



2.5 Known Issue 1

The system controller is unable to change the oscillator frequency after a successful Programming Recovery. The oscillator frequency is 25 MHz instead of 50 MHz. This is a known silicon issue, which is documented in the [ER0196: SmartFusion2 Device, Errata](#).

After Recovery Operation, junk messages are displayed on the serial terminal. The workaround is to apply a system reset after a successful recovery. Microsemi recommends that this workaround is implemented for any design, which uses the Programming Recovery. For more information about how to implement this workaround, refer to [Appendix 5: Implementing Workaround to Reset the Device after Programming Recovery](#), page 32.

The design example provided in this demo implements the workaround for Programming Recovery issue and the design files are available in the following location:

<downloadfolder>SF2_TFTP_IAP_Recovery_Demo_DF\Sample_files\Recovery_WA

The same workaround is also implemented in the Program Recovery Image that is, `Demo_Golden spi` which is demonstrated in [Running the Demo Design](#), page 14.

2.6 Known Issue 2

After the IAP or ISP program execution, read and write access to LSRAM is not allowed. This is a known Silicon issue, which is documented in the [ER0196: SmartFusion2 Device, Errata](#).

The workaround for this problem is to apply the system reset after IAP or ISP program operation. Microsemi recommends that this workaround is implemented for any design, which accesses LSRAM after IAP or ISP.

For more information about how to implement this workaround, refer to [Appendix 6: Implementing Workaround to Access Fabric LSRAM after IAP/ISP Program Operation](#), page 36 to resolve this issue.

The design example provided in this demonstration implements the workaround for accessing LSRAM after implementing the IAP or ISP program operation in the Libero software, and the design files are available in the following location:

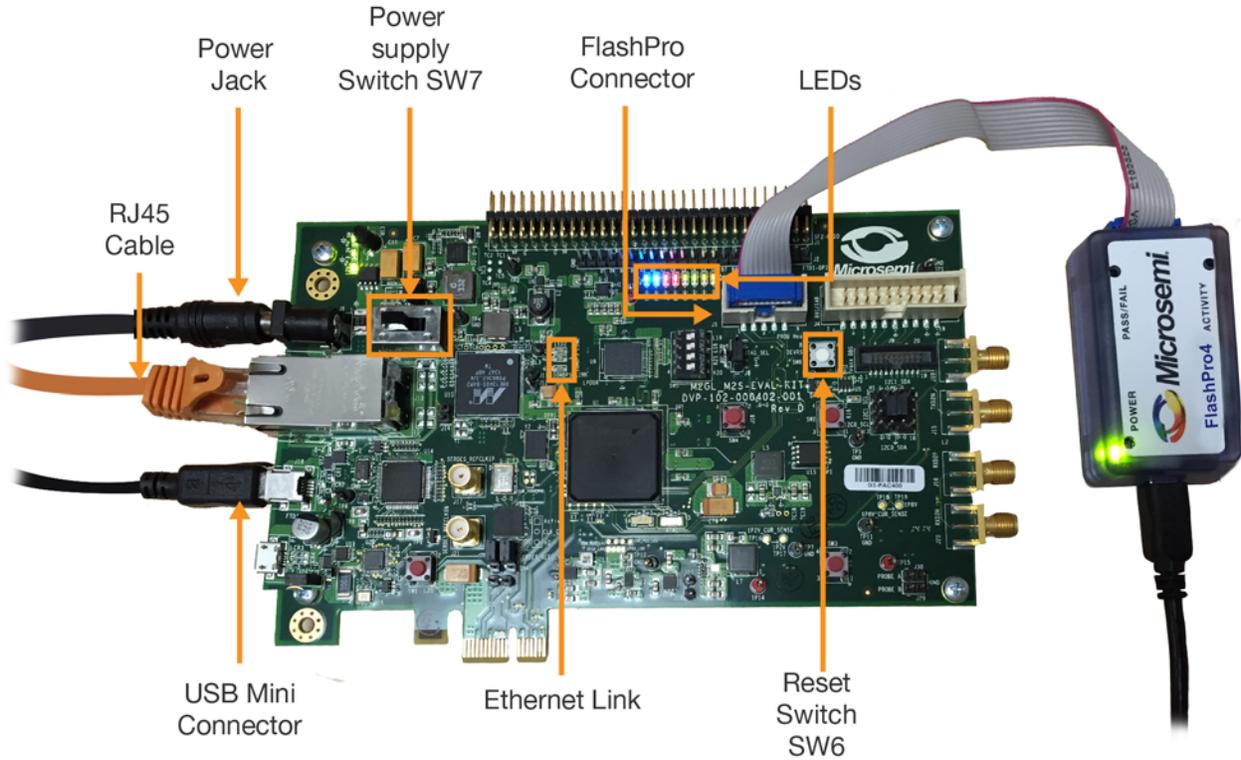
<downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sample_files\LSRAM_WA

The same workaround is also implemented in the IAP Image that is, `Demo_IAPImage spi`, which is demonstrated in [Running the Demo Design](#), page 14.

3 Appendix 1: Board Setup for Running the Demo

The following figure shows the board setup for running the demo on the SmartFusion2 Security Evaluation Kit.

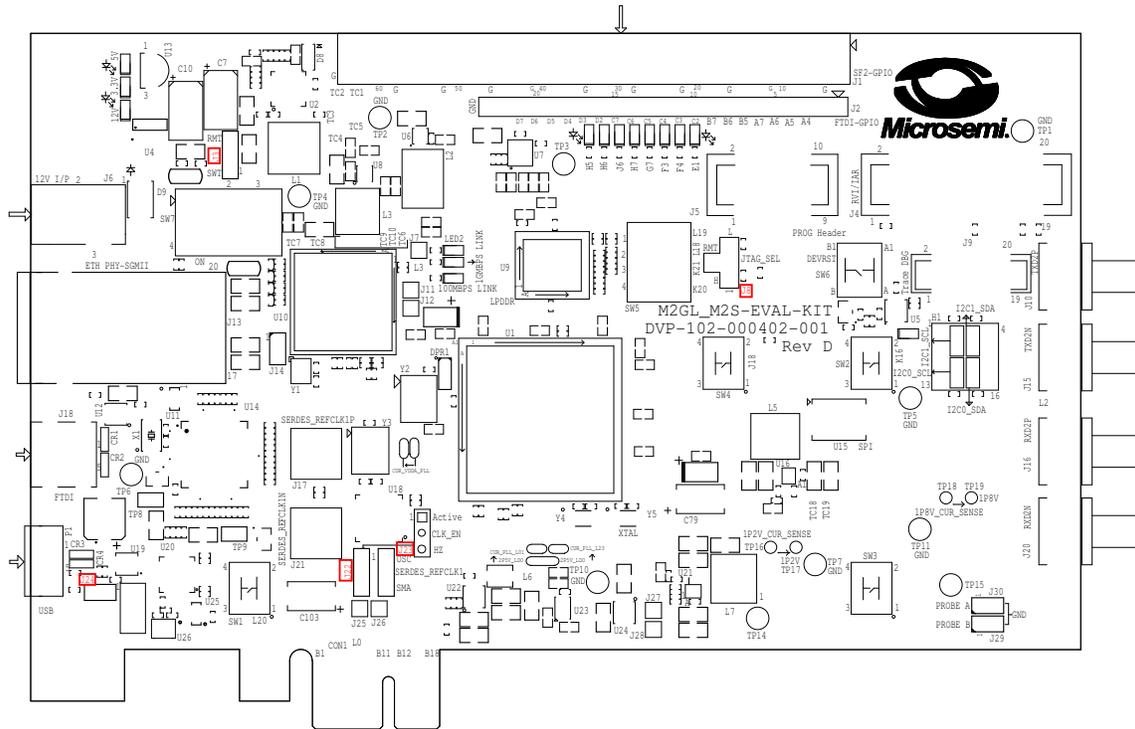
Figure 30 • SmartFusion2 Security Evaluation Kit Setup



4 Appendix 2: SmartFusion2 Security Evaluation Kit Board Jumper Locations

The following figure shows the jumper locations on the SmartFusion2 Security Evaluation Kit board.

Figure 31 • SmartFusion2 Security Evaluation Kit Board Jumper Locations



Note: Jumpers highlighted in red (J22, J23, J24, J3, J8) are set by default.

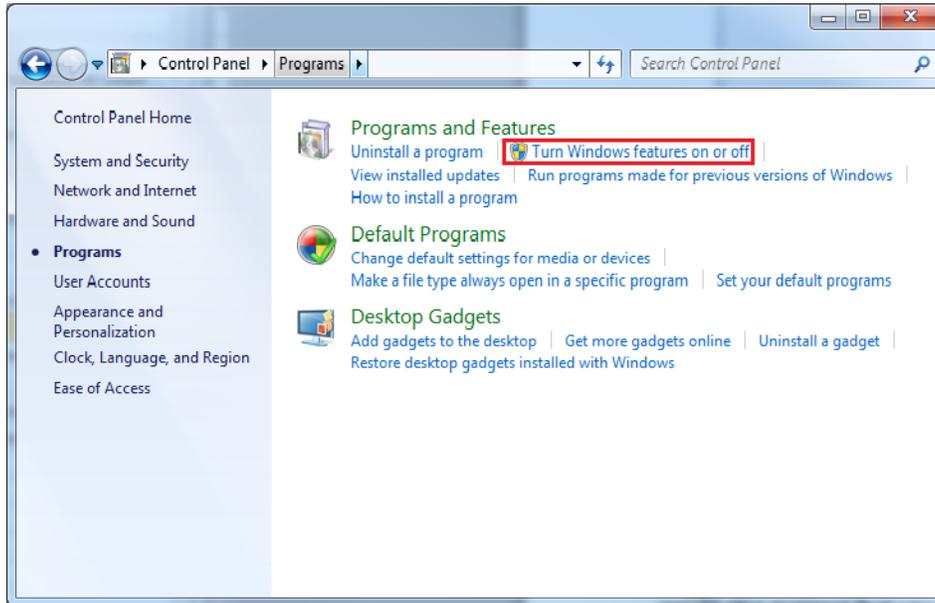
Note: The location of the jumpers in the preceding figure are searchable.

5 Appendix 3: Enable TFTP Client

The following steps describe how to enable TFTP client:

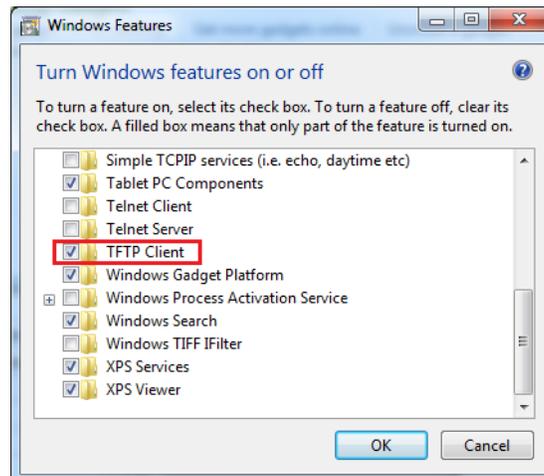
1. Navigate to **Control Panel > Programs**. Click **Turn Windows Features On or Off** as shown in the following figure.

Figure 32 • Control Panel - Programs and Features



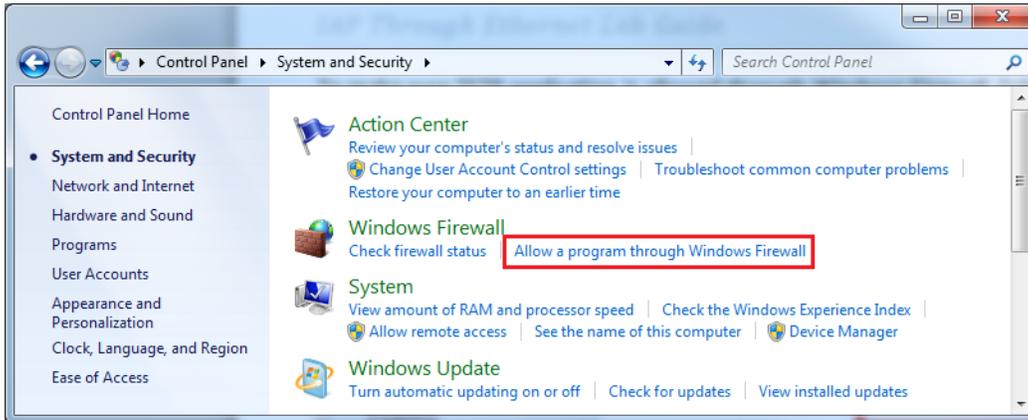
2. Select the **TFTP Client** check box from Windows Features as shown in the following figure.

Figure 33 • Selecting TFTP Client from Windows Features



- Browse through **Control Panel > System and Security**, click **Allow a program through Windows Firewall**.

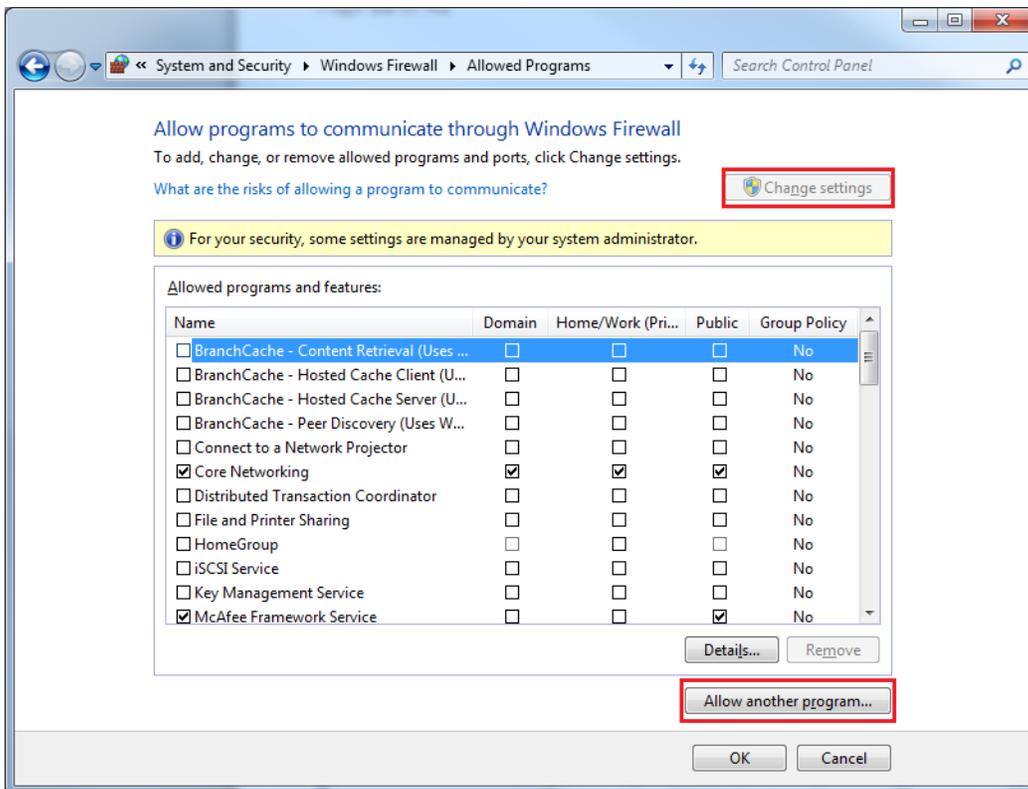
Figure 34 • System and Security Window



Note: If the System and Security option is not available, then enter the firewall in the search window to perform Step 3.

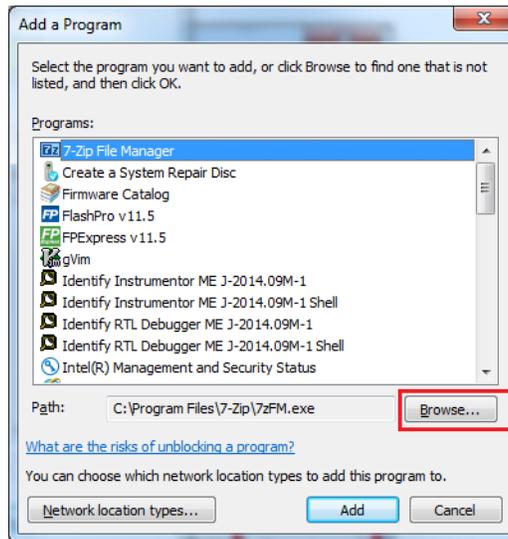
- Click **Change settings** and select **Allow another program...**

Figure 35 • Allowed Programs Window



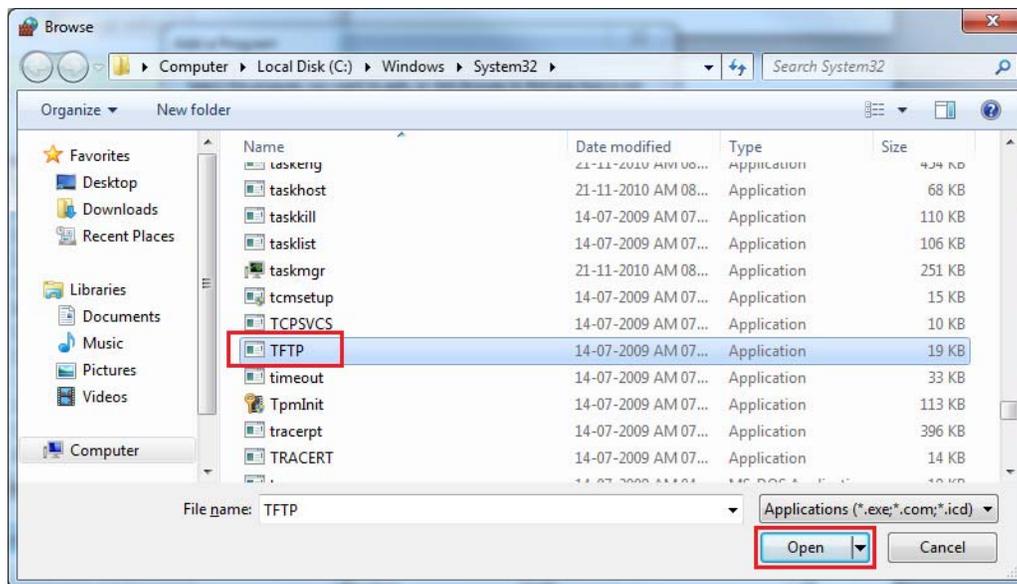
- The Add a Program window is displayed and click **Browse...**

Figure 36 • Add a Program Window



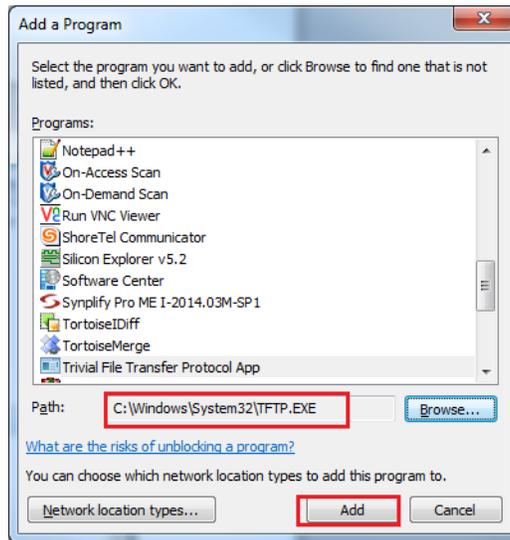
- Browse through `C:\ -> Windows -> System32`, select **TFTP.exe**, and click **Open**.

Figure 37 • Selecting the TFTP Executable File



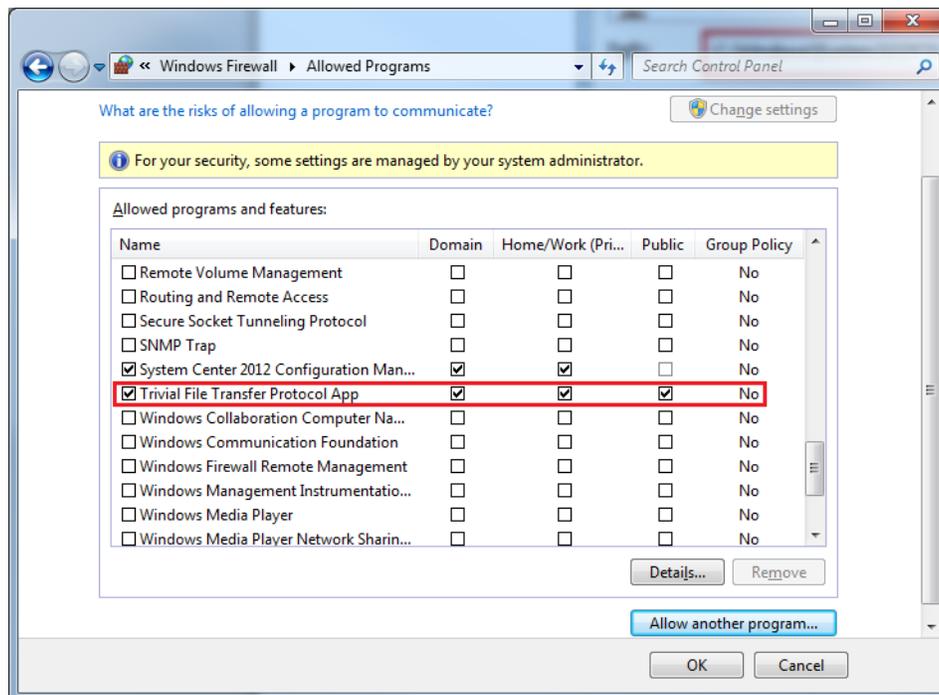
- Ensure that the TFTP.EXE path (C:\Windows\System32\TFTP.EXE) is selected correctly and click **Add**.

Figure 38 • TFTP.EXE Path in Add a Program Window



- Ensure that the **Trivial File Transfer protocol App** is added and select all the check boxes (Domain, Home/Work, Public) as shown in the following figure.

Figure 39 • Selecting Trivial File Transfer Protocol App in Allowed Programs Window



- Click **OK**.

6 Appendix 4: Running the Design in Static IP Mode

A static IP address does not change until it is explicitly changed and it is a permanent address assigned to a device to access the Internet. A device using a static IP address must be reconfigured each time it switches the network. A dynamic IP address is a temporary address that is assigned each time a computer or device accesses the Internet. The Dynamic IP address requires a DHCP-capable router.

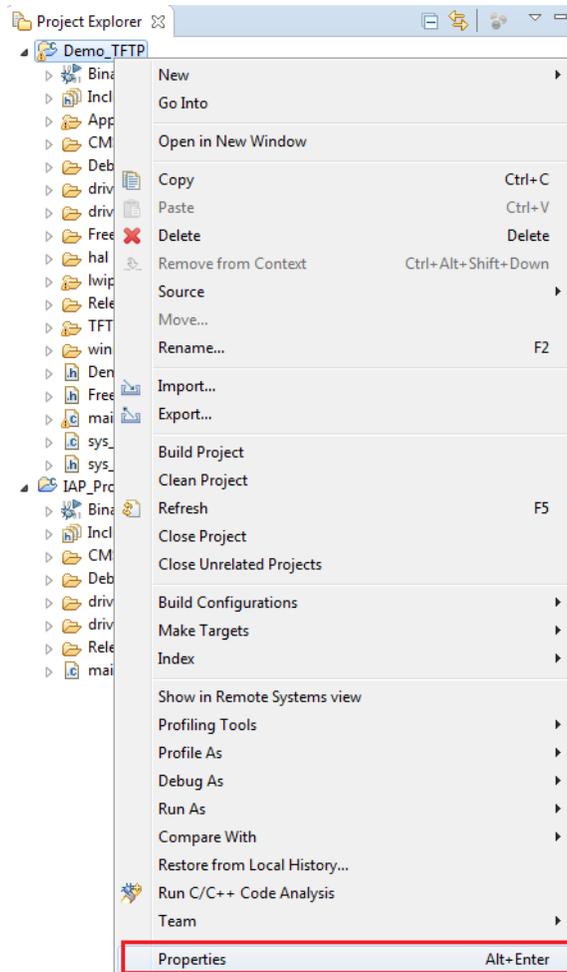
Note: In this demo, the design files are provided with the dynamic IP address settings.

The following steps describe how to run the design in Static IP mode:

1. Click **Start > Programs > Microsemi SoftConsole v4.0 > Microsemi SoftConsole v4.0** to open SoftConsole IDE.
2. Browse to the project Location `<downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DFLibero\Demo\SoftConsole4.0`.

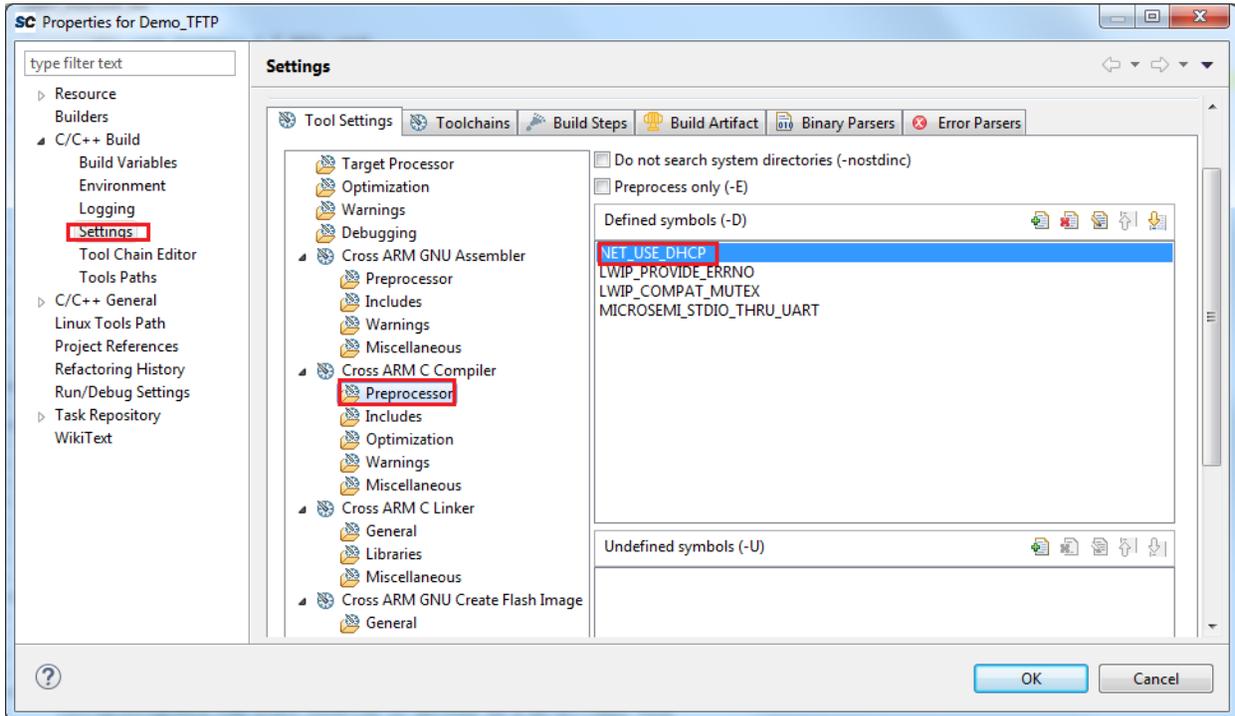
To run the design in Static IP mode, right-click the **Demo_TFTP** project and select **Properties** as shown in the following figure.

Figure 40 • Project Explorer Window SoftConsole Project



- Remove the **NET_USE_DHCP** symbol in **Tool Settings** of the Properties for **Demo_TFTP** window as shown in the following figure.

Figure 41 • Properties for Demo_TFTP



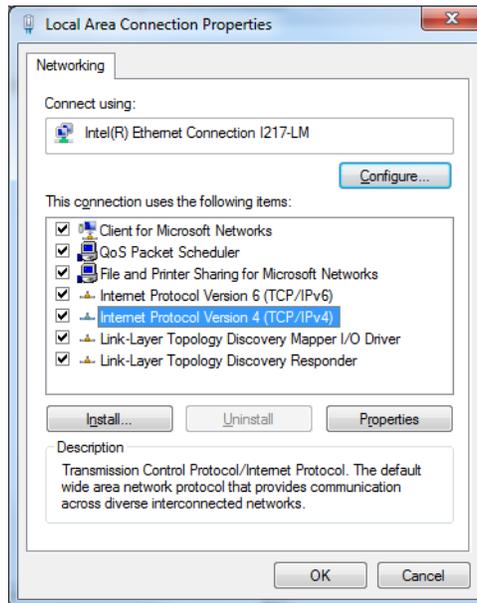
- Rebuild the SoftConsole Project. Load the design in to the eNVM.

Note: Programming file with static IP settings is available in the following path.
 <Downloadfolder>SF2_TFTP_IAP_Recovery_Demo_DF\Programmingfile\StaticIP\
 Demo_Main_static.stp.

- Navigate to Control Panel and enter **Network and Sharing Center** in search window.
- In **Network and Sharing Center** window, select **Change Adapter Settings**.
- Right-click on **Local Area Connection** and select **Properties**.
- If the device is connected in Static IP mode, the board static IP address is 169.254.1.23, then change the host TCP/IP settings to reflect the IP address.

The following figure shows Host PC TCP/IP settings.

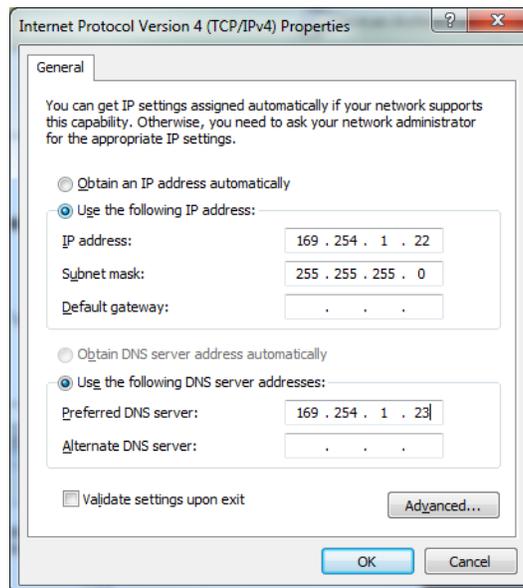
Figure 42 • Local Area Connection Properties Window



Update the static IP settings as shown in the following figure.

9. Click **OK**.

Figure 43 • Internet Protocol Version 4 (TCP/IPv4) Properties



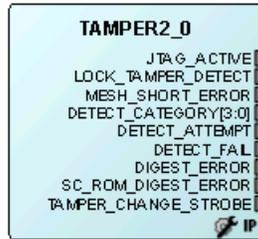
10. Run the demo with Static IP address 169.254.1.23 as described in the Running the Demo Section.

7 Appendix 5: Implementing Workaround to Reset the Device after Programming Recovery

The following changes are required in the Libero design.

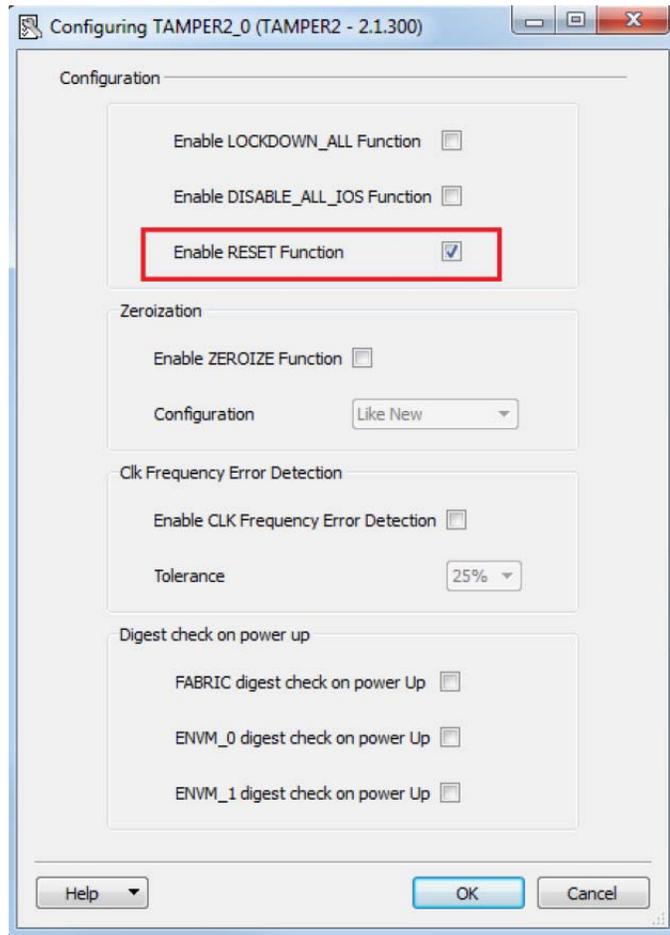
1. Select **File > New > SmartDesign**.
2. Enter the name as Program_Recovery_WA in the **Create New SmartDesign** window.
3. Browse to the Libero catalog to open **Tamper Macro**. The Tamper Macro resets the device when the RESET_N input port is connected to the logic 0.
4. From the Libero catalog, drag-and-drop **Tamper Macro** to the Program_Recovery_WA SmartDesign canvas, as shown in the following figure.

Figure 44 • Tamper Macro



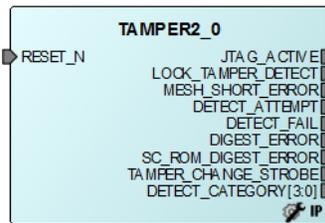
- Select the **Enable RESET** function check box in the **Configuring Tamper 2_0** window.
- Click **OK**. The System Reset option is enabled.

Figure 45 • Tamper Macro Configuration Window



The following figure shows the TAMPER2_0 macro after configuration.

Figure 46 • Tamper Macro



5. Instantiate the Clock_check HDL Module that is provided in the design files. The HDL module is a 25 bits ring counter and it counts the number of pulses coming into clock input pin and it is enabled by the logic high on the reset pin. The pulse output pin is asserted high only when any of the ring counter bits 10, 11, 12, 13, or 14 is high, and it occurs only at the negative edge of reset. So, this module generates output pulses only for 25 MHz and not for 50 MHz. Follow the steps to add Clock_check HDL module to Libero design.
6. Choose File > Import > HDL Source Files.
7. Browse to the following Clock_check.v file location in the design files folder:
 <downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sourcefiles

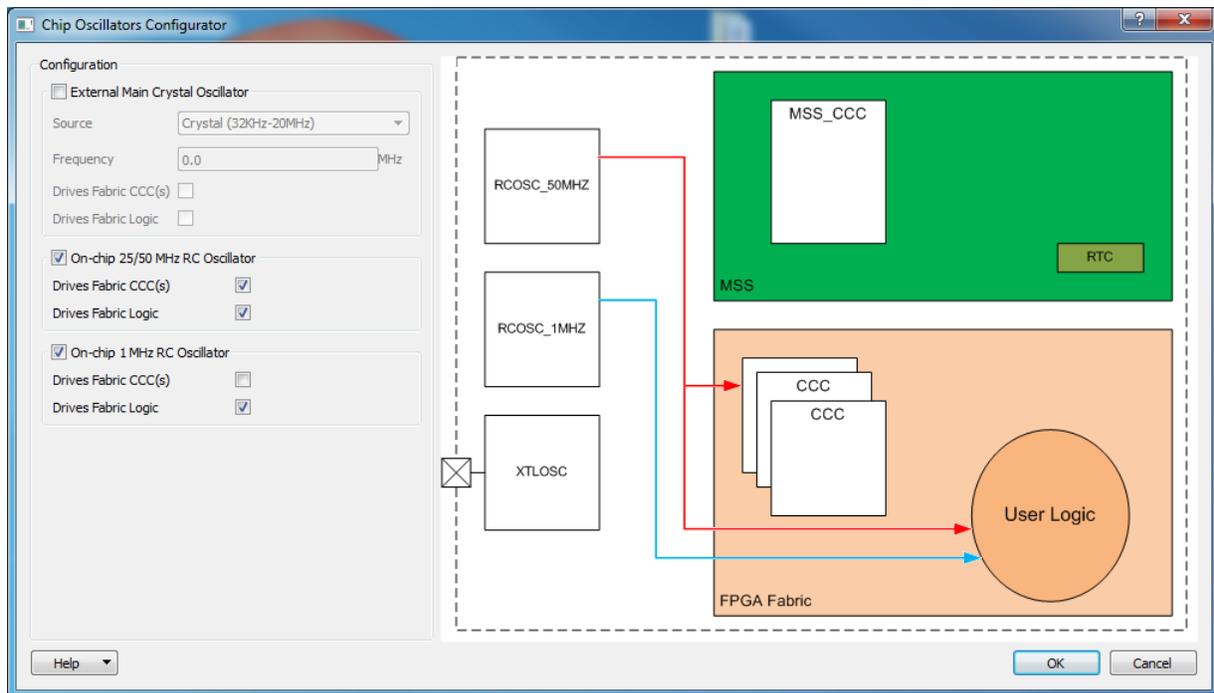
- Click **Program_Recovery_WA** tab and drag-and-drop the `clock_check.v` component from the from the Design Hierarchy to the Program_Recovery_WA SmartDesign canvas. The following figure shows the Clock_check HDL module.

Figure 47 • Clock_check HDL Component



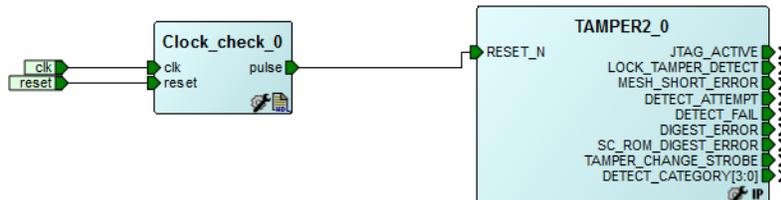
- Right-click **Demo_sb** under **Demo** in **Design Hierarchy** window and select **Open As SmartDesign**.
- Select **Demo_sb** tab and double-click **FABOSC_0 (On Chip Oscillator)**.
- Configure the FABOSC_0 with the following settings, as shown in the following figure.
 - On-chip 25/50 MHz RC Oscillator must be enabled to Drive Fabric Logic.

Figure 48 • Chip Oscillators Configurator



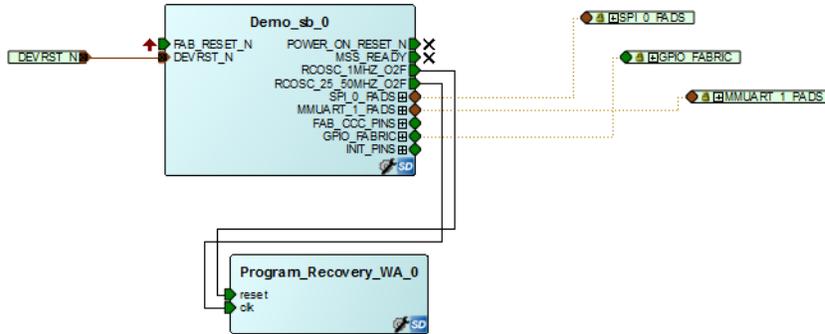
- Select **Demo** tab.
- Right-click **Demo_sb_0** and select **Update Instance** with the latest component.
- Click **Program_Recovery_WA** tab and make the connections, as shown in the following figure.

Figure 49 • Program_Recovery_WA



15. Click **Demo** tab and drag-and-drop the **Program_Recovery_WA** component from the Design Hierarchy to the Demo SmartDesign canvas.
16. Make the connection as shown in the following figure and generate Demo SmartDesign. This completes the implementation of the workaround.

Figure 50 • Demo Smart Design



Note: If the Auto Update feature is enabled and update image is available in the Flash memory, then after recovery, a default reset occurs due to the workaround. As a result, Update Image is programmed, if the Update Image version is greater than the Golden Image.

Note: This workaround only works for Mode 1.2 settings in the SPI images.

Reason: If the SPI images are set to Mode 1 V settings, the device might undergo permanent reset. As it continuously holds reset by the workaround since, the oscillator frequency is 25 MHz.

8 Appendix 6: Implementing Workaround to Access Fabric LSRAM after IAP/ISP Program Operation

The LSRAM write and read access are denied after implementing the IAP or ISP program operation. The workaround for this problem is to apply the system reset after IAP or ISP program operation.

8.1 Changes Required in the Libero Design

The changes that are required in the Libero Design are:

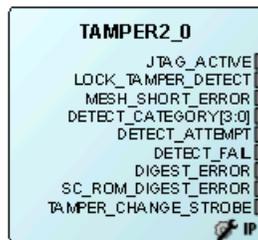
- Option 1: Creating SmartDesign
- Option 2: Importing the .cxf file in the Libero Design

8.1.1 Option 1: Creating SmartDesign

The following steps describe how to apply the system reset.

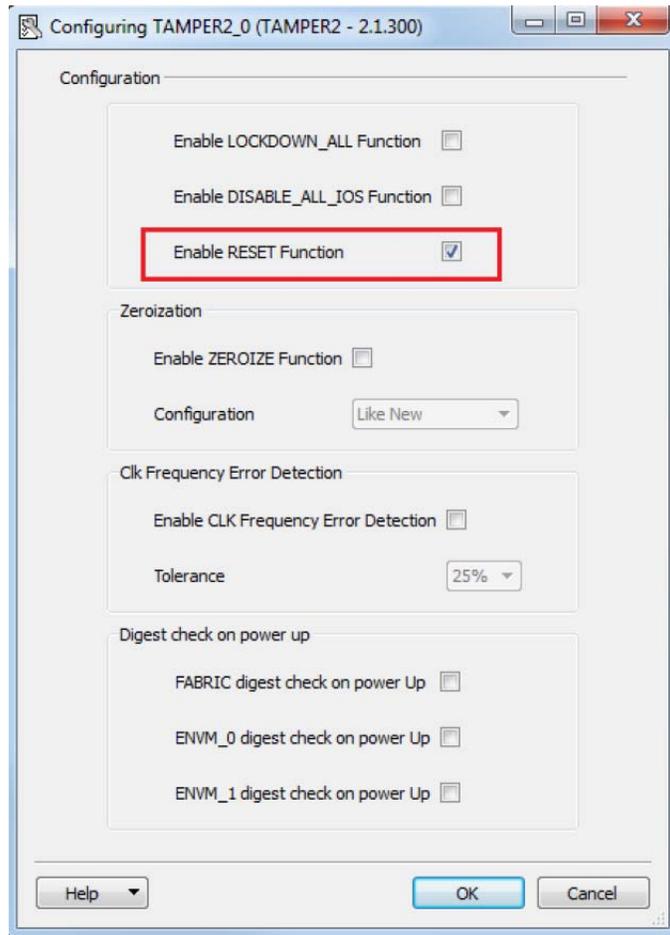
1. Select **File > New > SmartDesign**.
2. Enter the name as Dev_Restart_after_IAP_blk in **Create New SmartDesign** window.
3. Browse to the Libero catalog to open **Tamper Macro**. The Tamper Macro resets the device when the RESET_N input port is connected to the logic 0.
4. Drag-and-drop the **Tamper Macro** that is available in the Libero catalog to the Dev_Restart_after_IAP_blk SmartDesign canvas, as shown in the following figure.

Figure 51 • Tamper Macro



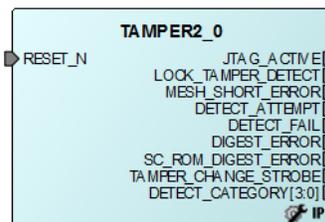
5. Click **Enable RESET** check box in the Configuring Tamper 2_0 window.
6. Click **OK**. The System Reset option is enabled.

Figure 52 • Tamper Macro Configuration Window



The following figure shows the TAMPER2_0 macro after configuration.

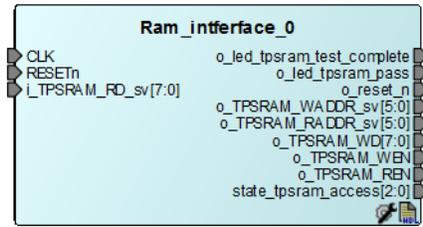
Figure 53 • Tamper Macro



7. Instantiate the FSM Module that is provided in the design files. This FSM Logic performs three consecutive address writes to the Two-Port LSRAM with the known data pattern and then reads back data from those three consecutive address locations for comparison. If the read back data pattern does not match with that of the written data pattern, then the FSM asserts the RESET_N input to the Tamper Macro, which in turn causes a system reset. If the read back data pattern matches with the written data pattern, then the FSM does not perform any action. Perform the following steps to add the FSM logic to the Libero design.
 - Select **File > Import > HDL Source Files**.
 - Browse to the following Ram_interface.v file location in the design files folder:
`<downloadfolder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sourcefiles`

- Click **Dev_Restart_after_IAP_blk** tab and drag-and-drop the `Ram_interface.v` component from the Design Hierarchy to the `Dev_Restart_after_IAP_blk` SmartDesign canvas. The following figure shows the `Ram_interface` component.

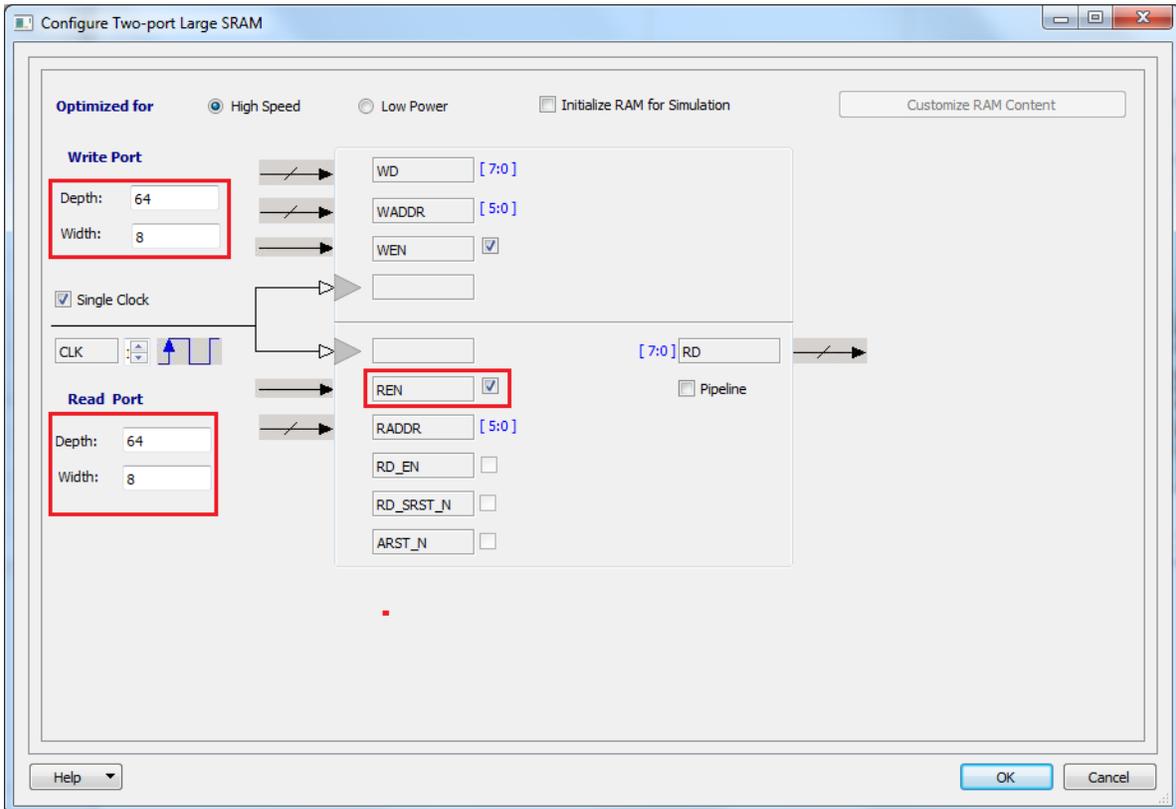
Figure 54 • Ram_interface FSM Component



After completion of 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.

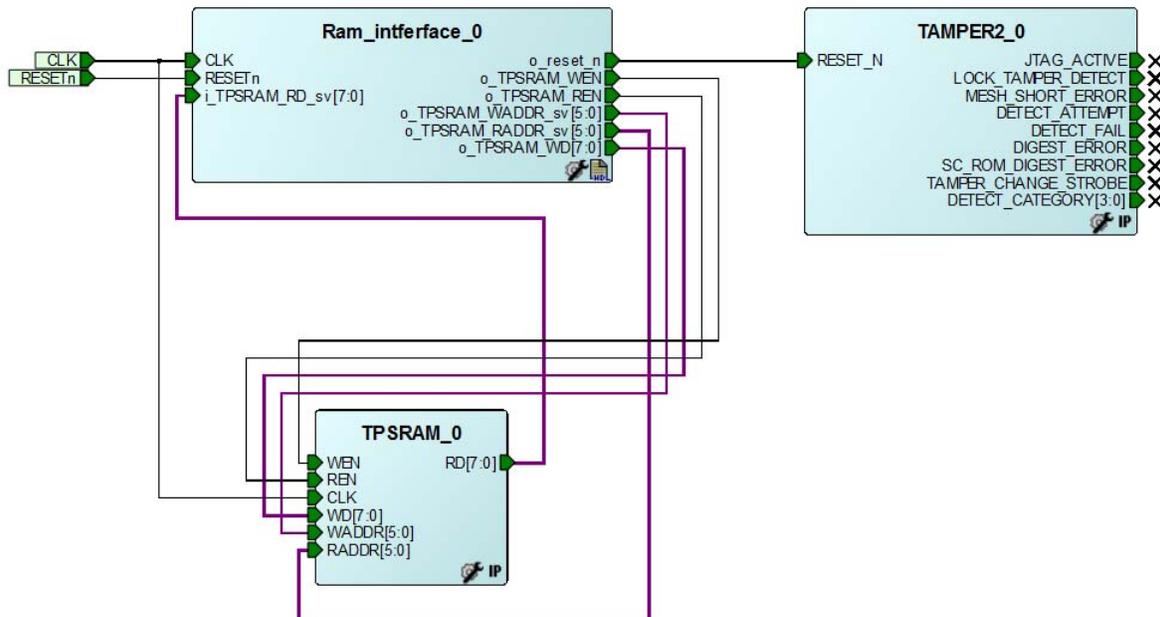
8. Drag-and-drop the Two-Port Large SRAM (TPSRAM) that is, available in the Libero catalog to the `Dev_Restart_after_IAP_blk` SmartDesign canvas. Configure the TPSRAM with the following settings:
 - Write Port
 - Depth: 64
 - Width: 8
 - Read Port
 - Depth: 64
 - Width: 8
 - Check **REN** check box

Figure 55 • Two-Port LSRAM Configurator Window



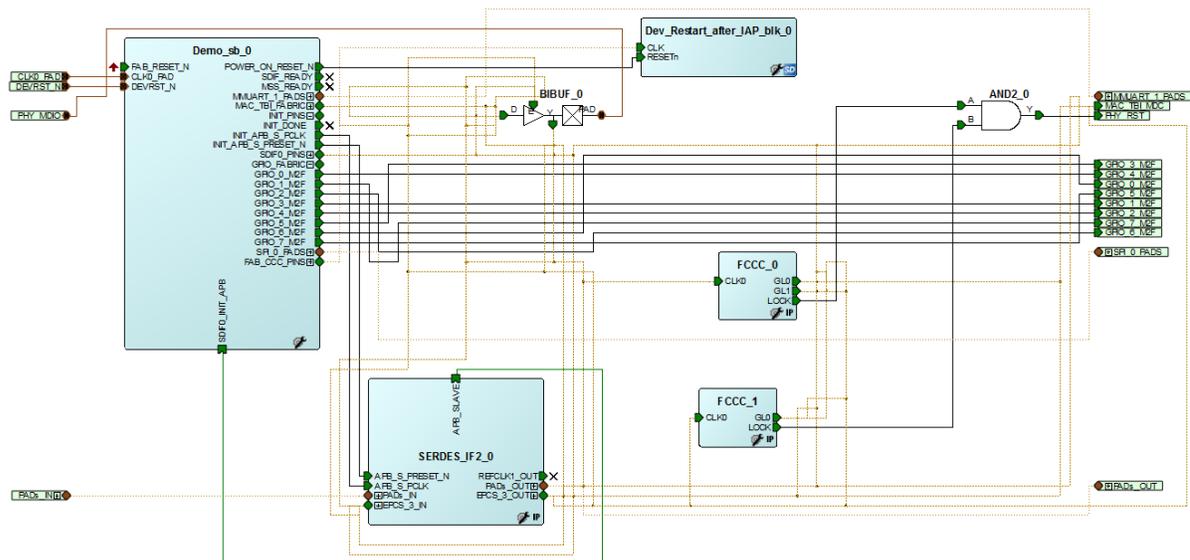
- Make the connections for Tamper Macro, FSM, and TPSRAM, as shown in the following figure.

Figure 56 • Dev_Restart_after_IAP_blk_SmartDesign



- Click **Demo** tab and drag-and-drop the **Dev_Restart_after_IAP_blk** component from the Design Hierarchy to the Demo SmartDesign canvas.
- Make the connection as shown in the following figure and generate the Demo SmartDesign. This completes the implementation of the workaround.

Figure 57 • Demo Smart Design



Note: This workaround is applicable only for v11.5 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 might prune Tamper Macro during Synthesis. To avoid pruning, one of the recommended options is to promote the DETECT_ATTEMPT signal to the top-level.

8.1.2 Option 2: Importing the .cxf file in the Libero Design

The alternative option to implement the workaround is to import the .cxf file for SmartDesign Dev_Restart_after_IAP_blk. This .cxf file is provided with the design files and it has all the component instantiations and connections as mentioned in [Option 1: Creating SmartDesign](#), page 36.

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

1. Extract files: `<download_folder>\SF2_TFTP_IAP_Recovery_Demo_DF\LSRAM_WA\Libero\Demo`
2. Select **File > Import > Others**.
3. Browse to the following Dev_Restart_after_IAP_blk.cxf file location in the design files folder:
`<download_folder>\SF2_TFTP_IAP_Recovery_Demo_DF\sample_files\LSRAM_WA\Libero\Demo\component\work\Dev_Restart_after_IAP_blk`
4. Browse to the following Ram_interface.v file location in the design files.
`<download_folder>\SF2_TFTP_IAP_Recovery_Demo_DF\Sourcefiles`
5. Repeat Step 7 and Step 8 to instantiate Dev_Restart_after_IAP_blk in Demo SmartDesign.