SmartFusion2 - Code Shadowing from SPI Flash to SDR Memory - Libero SoC v11.4

Demo Guide

Table of Contents

Code Shadowing from SPI Flash to SDR Memory - Libero SoC v11.4

Introduction

This demo design shows SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) device capabilities for code shadowing from serial peripheral interface (SPI) flash memory to single data rate (SDR) synchronous dynamic random access memory (SDRAM) and executing the code from SDR SDRAM.

Figure 1 shows the top-level block diagram for code shadowing from SPI flash to SDR demo.

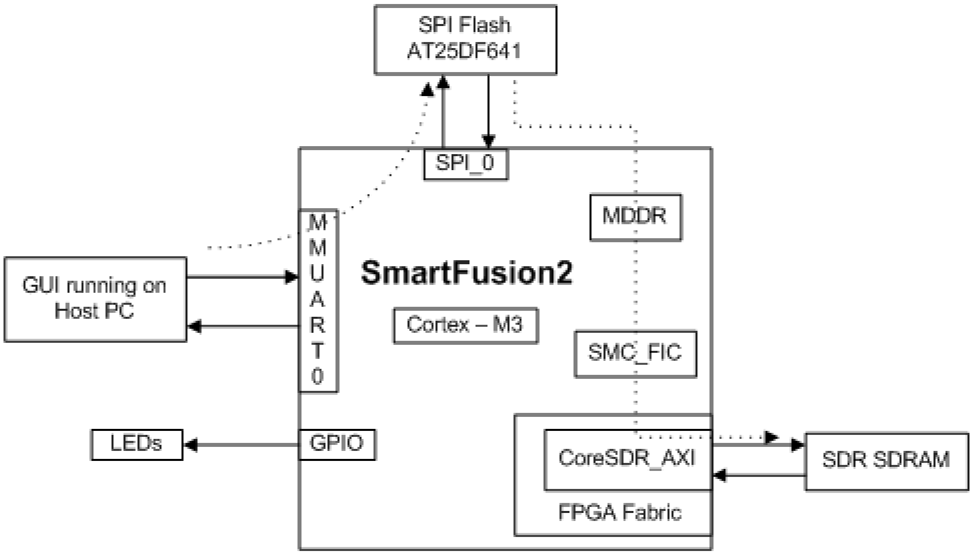


Figure 1. Top-level Block Diagram

Code shadowing is a booting method that is used to execute an image from external faster volatile memories (DRAM). Code shadowing is the process of copying the code from nonvolatile memory to volatile memory for execution. Code shadowing is required when the nonvolatile memory associated with the processor does not support random access to the code for execute-in-place or insufficient nonvolatile random access memory exists. In performance critical applications, execution speed can be improved by code shadowing where code is copied to higher throughput RAM for faster execution. SDR/double data rate (DDR) SDRAM memories are used in an application that has a large application executable image and requires higher performance. Typically, the large executable images are stored in nonvolatile memory such as NAND flash or SPI flash and copied to volatile memory such as SDR/DDR SDRAM memory at power up for execution.

SmartFusion2 SoC FPGA device integrates fourth generation flash-based FPGA fabric, ARM® Cortex®-M3 processor, and high performance communications interfaces on a single chip. SmartFusion2 SoC FPGA device has a soft memory controller fabric interface (SMC_FIC) which can be used to interface external bulk memories to the microcontroller subsystem (MSS) with a soft memory controller in the FPGA fabric. The Cortex-M3 processor can directly execute the instructions from external SDR memory through SDR controller in the FPGA fabric. The SmartFusion2 SoC FPGA cache controller and MSS DDR bridge handles the data flow for a better performance.

Design Requirements

Table 1 lists the design requirements.

Table 1. Design Requirements

Design Requirements	**Description**
Hardware Requirements	
SmartFusion2 Development Kit: • FlashPro4 programmer • USB to Mini USB cable • 12 V adapter	Rev C or later
Host PC or Laptop	Windows 64-bit Operating System
Software Requirements	
Libero® SoC	11.4
FlashPro Programming Software	11.4
Host PC driver	USB to UART drivers
Demo GUI	Microsoft .NET Framework 4 client
Operating system	Windows XP SP2 – 32-bit/64-bit Windows 7 – 32-bit/64-bit

Design Files

The design files for this demo can be downloaded from the Microsemi® website: http://soc.microsemi.com/download/rsc/?f=sf2_code_shadowing_spi_flash_sdr_memory_liberov11p4_dg_df

Design files include:

- Libero SoC project
- Programming files
- GUI executable
- Sample application images
- Linker scripts
- Readme file

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

Demo Design Description

This demo design implements a bootloader application to load the target application executable image into SPI flash from host PC and copies it to the SDR SDRAM for execution. This design provides host interface over SmartFusion2 SoC FPGA multi-mode universal asynchronous/synchronous receiver/transmitter (MMUART) to load the target application executable image into SPI flash connected to the MSS SPI0 interface. If the executable target application is available in SPI flash, the code shadowing from SPI flash to SDR SDRAM can be started. The bootloader program running from the embedded nonvolatile memory (eNVM) jumps to the target application in the SDR SDRAM once the target application image is copied to SDR SDRAM.

In this demo design, the external SDR SDRAM is interfaced with the SmartFusion2 SoC FPGA MSS through the SMC_FIC and a soft SDRAM controller (CoreSDR_AXI IP core) in the FPGA fabric. The CoreSDR_AXI IP core is interfaced with the MSS using AXI bus interface. This demo design is intended to be demonstrated on a SmartFusion2 SoC FPGA Development Kit. External component pin mappings are done according to the SmartFusion2 SoC FPGA Development Kit schematics.

The MSS and SDRAM controller (CoreSDR_AXI) in the FPGA fabric are operating at a clock frequency of 80 MHz which is derived using internal 25/50 MHz RC oscillator and fabric clock conditioning circuits (CCC). SDRAM controller timing parameters are set according to the system clock frequency of 80 MHz, as shown in Appendix A – SDR Configurations.

Figure 2 shows the block diagram of the demo design.

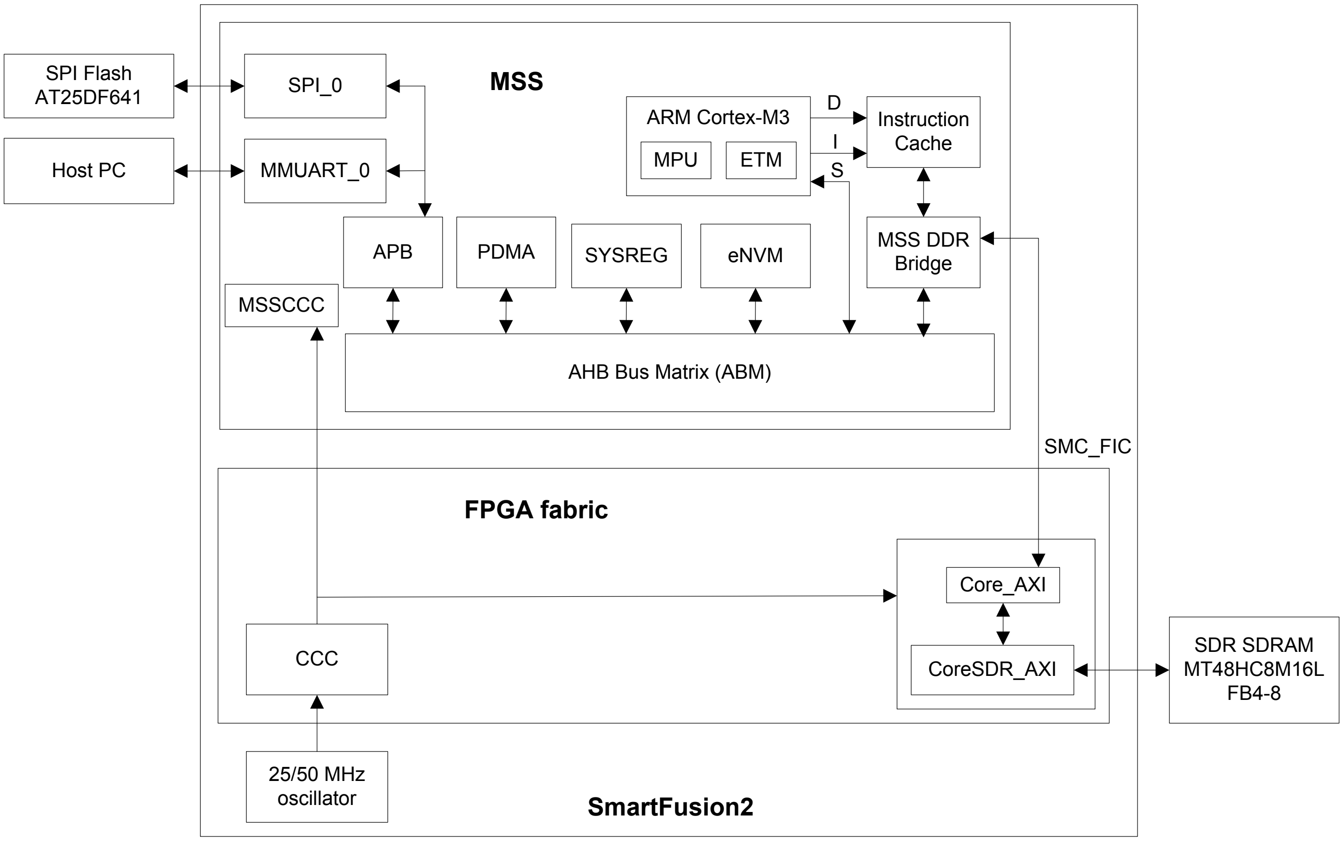


Figure 2. Code Shadowing from SPI Flash to SDR Memory Demo Block Diagram

This demo design involves the implementation of following tasks:

- Creating executable application image
- SPI flash loader to burn the application image into SPI flash memory
- Bootloader

Creating Application Image Targeting for SDR Memory

An image that can be executed from the SDR memory is required. To build the application image use the "production-execute-in-place-externalSDR.ld" linker description file that is included in the design files. This linker description file defines the SDRAM memory starting address as 0x00000000 since the bootloader application performs SDRAM memory remapping from 0xA0000000 to 0x00000000. This linked script creates an application image with instructions, data, and BSS sections in SDRAM whose starting address is 0x00000000. A simple light-emitting diode (LED) blinking, timer and switch based interrupt generation application image file is provided for this demo.

SPI Flash Loader

The SPI flash loader is implemented to load the on-board SPI flash memory with the executable target application image from the host PC through the MMUART_0 interface. The Cortex-M3 processor makes a buffer for the data coming over the MMUART_0 interface and initiates the peripheral DMA (PDMA) to write the buffered data into SPI flash through the MSS_SPI0.

Bootloader

This is the major part of the code shadowing demo which copies and boots the code from the SDR memory. The bootloader performs the following operations:

1. Copying the target application image from SPI flash memory to SDR SDRAM memory.
2. Remapping the SDRAM starting address from 0xA0000000 to 0x00000000 by writing to DDR_CR register.
3. Initializing the Cortex-M3 processor stack pointer as per the target application. The first location of the target application vector table contains the stack pointer value. The vector table of the target application is available starting from the address 0x00000000.
4. Loads the Program Counter (PC) to reset handler of the target application for executing the target application image from the SDR memory. Reset handler address of the target application is available in the vector table at the address 0x00000004.

The demo design architecture is described in Figure 3.

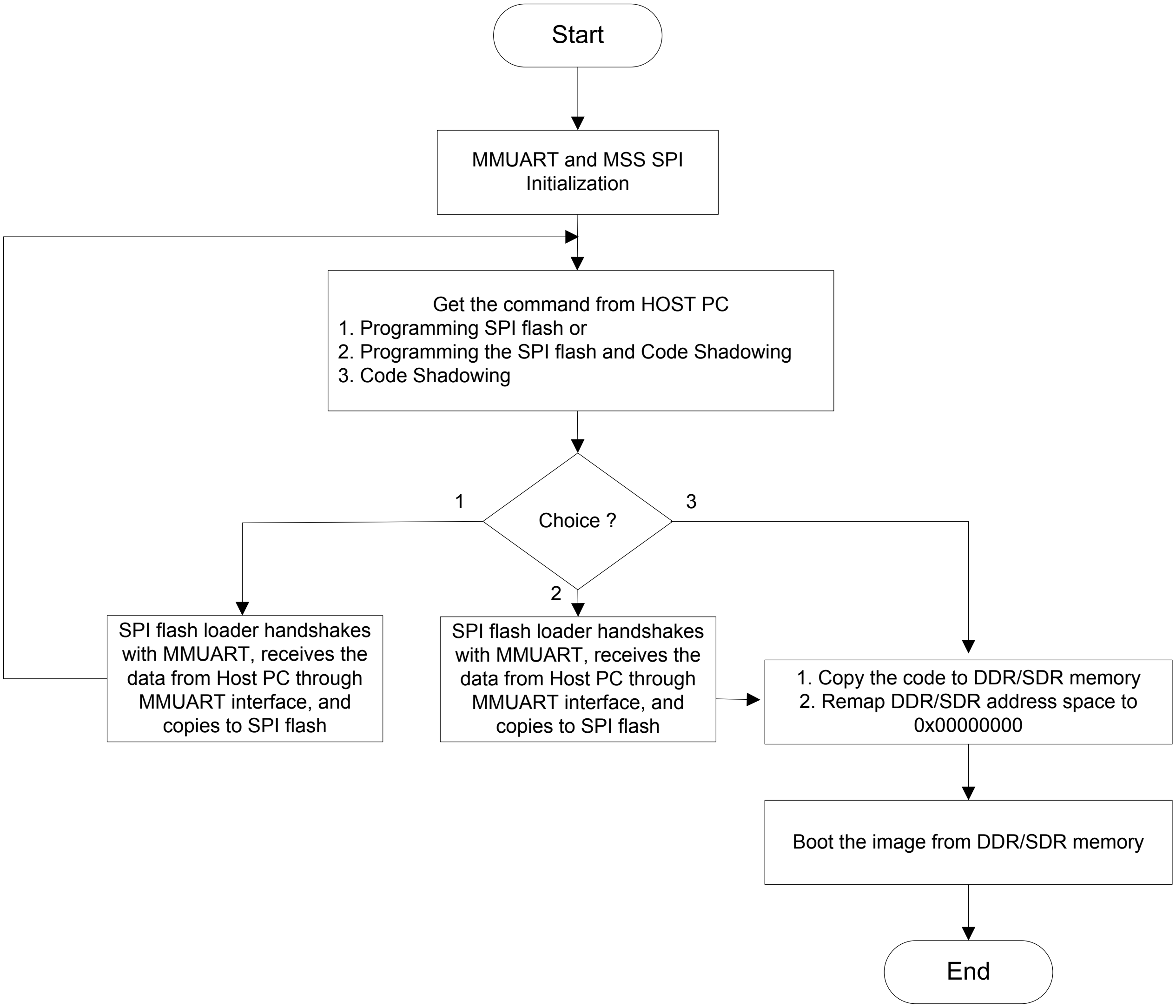

Start
MMUART and MSS SPI Initialization
Get the command from HOST PC
1. Programming SPI flash or
2. Programming the SPI flash and Code Shadowing
3. Code Shadowing
Choice ?
1
2
3
SPI flash loader handshakes with MMUART, receives the data from Host PC through MMUART interface, and copies to SPI flash
SPI flash loader handshakes with MMUART, receives the data from Host PC through MMUART interface, and copies to SPI flash
1. Copy the code to DDR/SDR memory
2. Remap DDR/SDR address space to 0x00000000
Boot the image from DDR/SDR memory
End

Figure 3. Design Flow

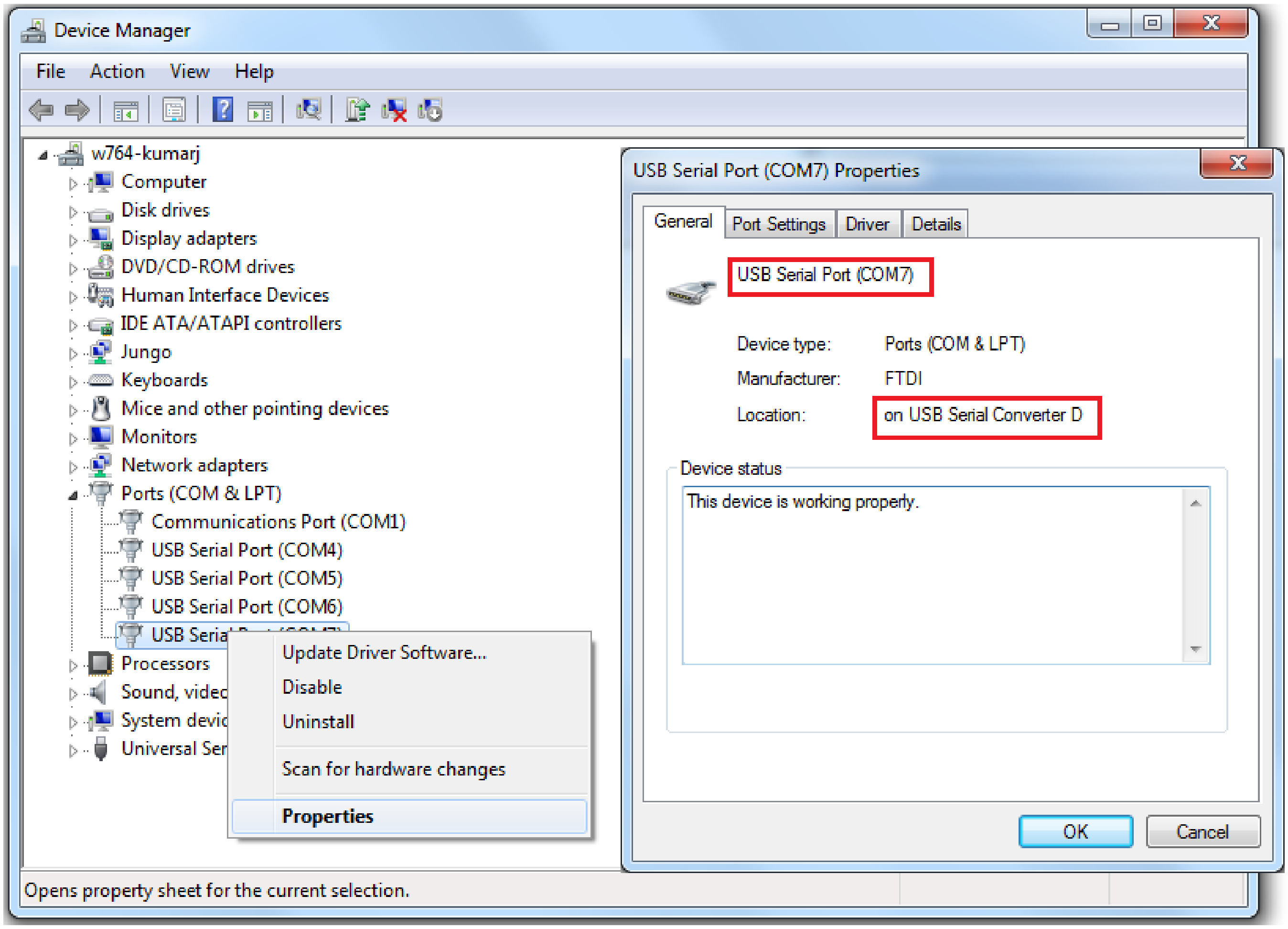
Running the Demo

The demo shows how to load the application image in the SPI flash and execute that application image from external SDR memories. This demo provides an example application image “sample_image_SDR.bin”. This image shows the welcome messages and timer interrupt message on the serial console and blinks the LED1 to LED8 on the SmartFusion2 Development Kit. To see the switch interrupt messages on the serial console, press **SW2** or **SW5** switch.

Demo Setup

1. Connect the FlashPro4 programmer to the J59 connector of SmartFusion2 SoC FPGA Development Kit.
2. Connect one end of the USB mini-B cable to the J24 connector provided on the SmartFusion2 SoC FPGA Development Kit. Connect the other end of the USB cable to the host PC.

Make sure that the USB to UART bridge drivers are automatically detected (can be verified in the Device Manager), as shown in Figure 4. From the detected four COM ports, select the one which location on its properties window should be as “on USB Serial Converter D”. Note the COM port number for serial port configuration.



USB Serial Port (COM7)

Device type: Ports (COM & LPT)

Manufacturer: FTDI

Location: on USB Serial Converter D

Device status

This device is working properly.

Opens property sheet for the current selection.

Figure 4. USB to UART Bridge Drivers

3. If USB to UART bridge drivers are not installed, download and install the drivers from www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.

4. Connect the jumpers on the SmartFusion2 SoC FPGA Development Kit, as shown in Table 2. While making the jumper connections the power supply switch SW7 on the board should be in **OFF** position.

Table 2. SmartFusion2 SoC FPGA Development Kit Jumper Settings

Jumper	Pin (from)	Pin (to)
J70, J93, J94, J117, J123, J142, J157, J160, J167, J225, J226, J227	1 (default)	2
J2	1 (default)	3
J23	2 (default)	3
J129, J133	2	3
For SPI to SPI Flash Connection		
J110, J118, J119, J121	1 (default)	2

5. Connect the power supply to J18 connector.

Figure 5 shows the board setup for running the code shadowing from SPI flash to SDR memory demo on SmartFusion2 SoC FPGA Development Kit.

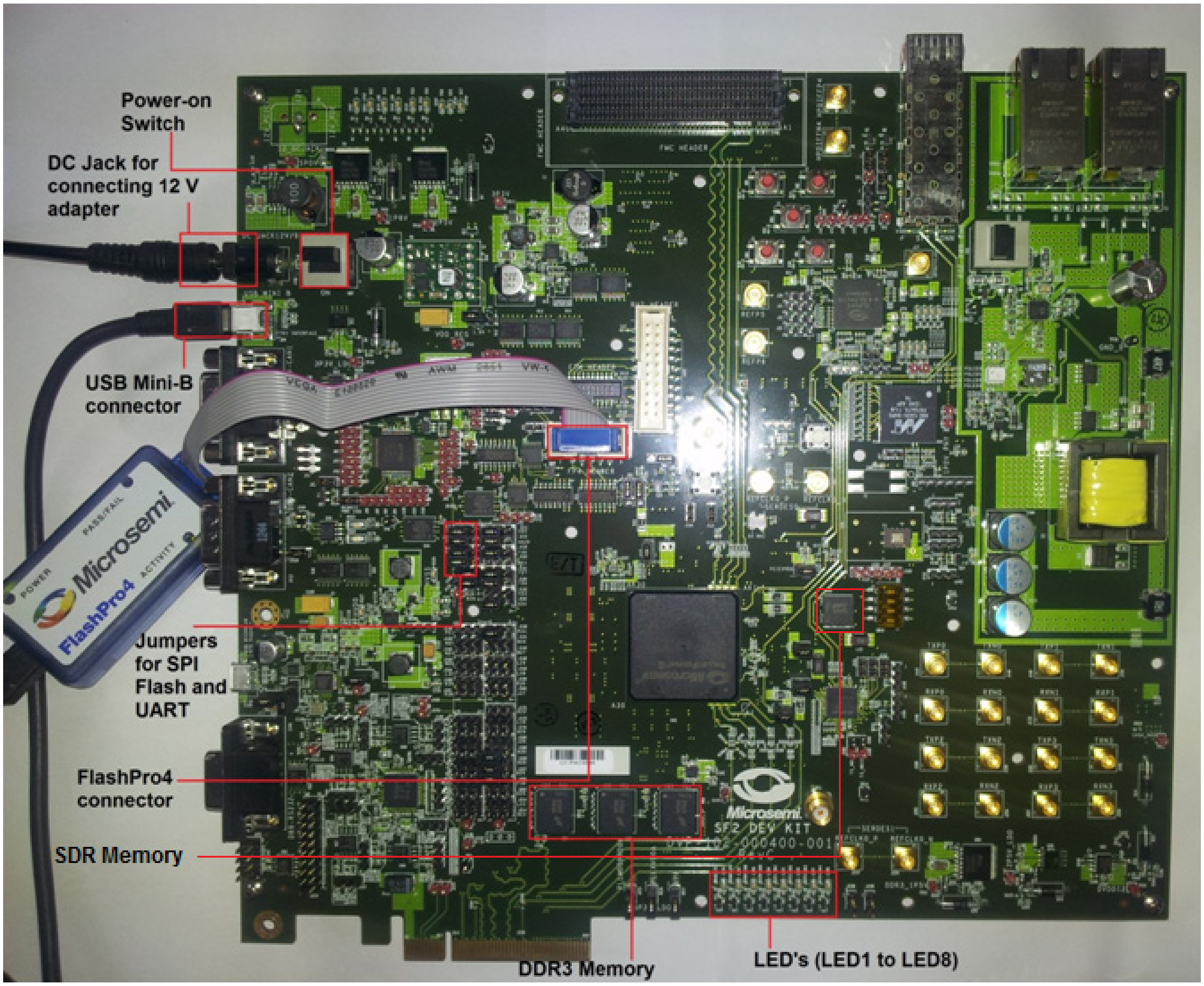
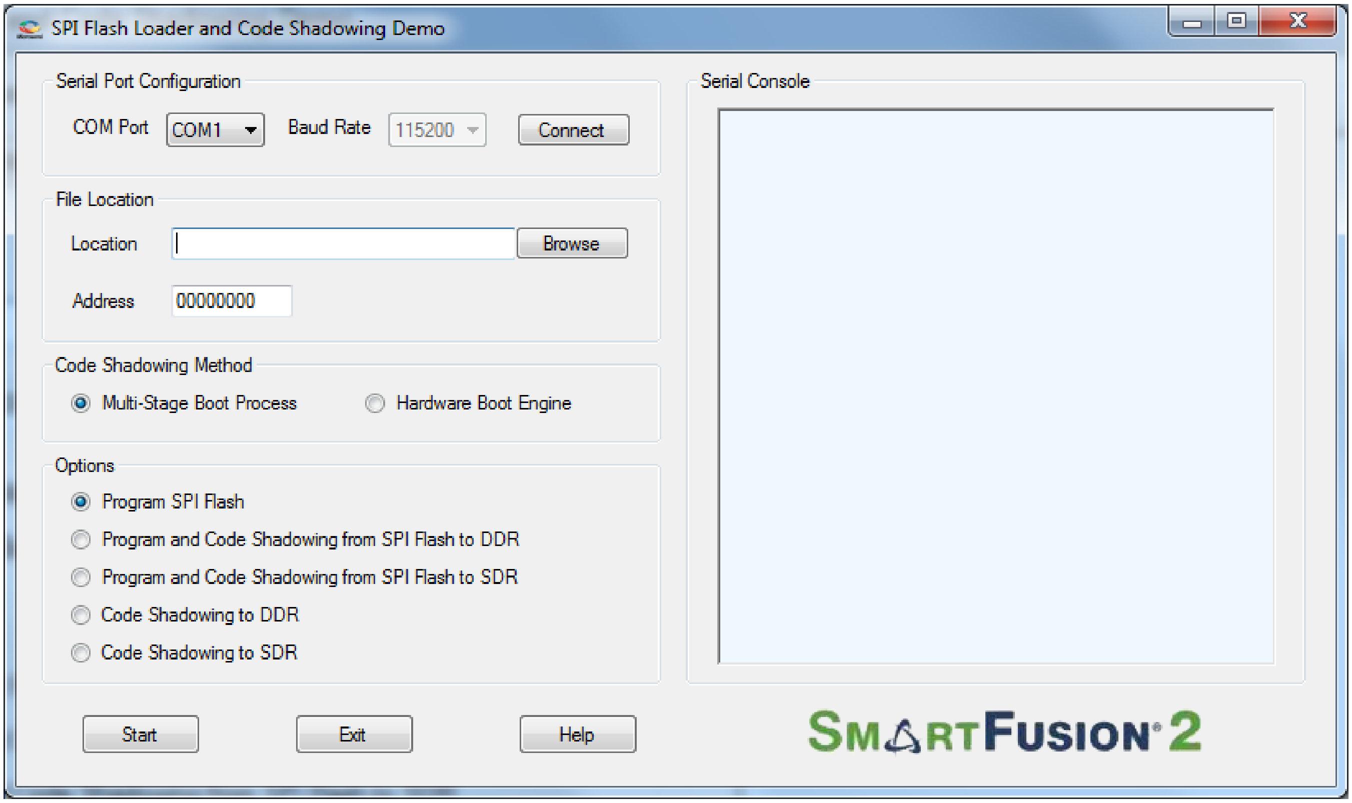


Figure 5. SmartFusion2 SoC FPGA Development Kit Setup

SPI Flash Loader and Code Shadowing Demo GUI

This is required to run the code shadowing demo. SPI Flash Loader and Code Shadowing Demo GUI is a simple graphic user interface that runs on the host PC to program the SPI flash and runs the code shadowing demo on the SmartFusion2 SoC FPGA Development Kit. UART is used as the underlining communication protocol between the host PC and SmartFusion2 SoC FPGA Development Kit. It also provides the serial console section to print the debug messages received from the application over the UART interface.

Figure 5 shows the SPI Flash Loader and Code Shadowing Demo GUI.



Serial Port Configuration

COM Port: COM1

Baud Rate: 115200

File Location

Location:

Address: 00000000

Code Shadowing Method

● Multi-Stage Boot Process ○ Hardware Boot Engine

Options

● Program SPI Flash

○ Program and Code Shadowing from SPI Flash to DDR

○ Program and Code Shadowing from SPI Flash to SDR

○ Code Shadowing to DDR

○ Code Shadowing to SDR

Figure 6. SPI Flash Loader and Code Shadowing Demo GUI

The GUI supports the following features:

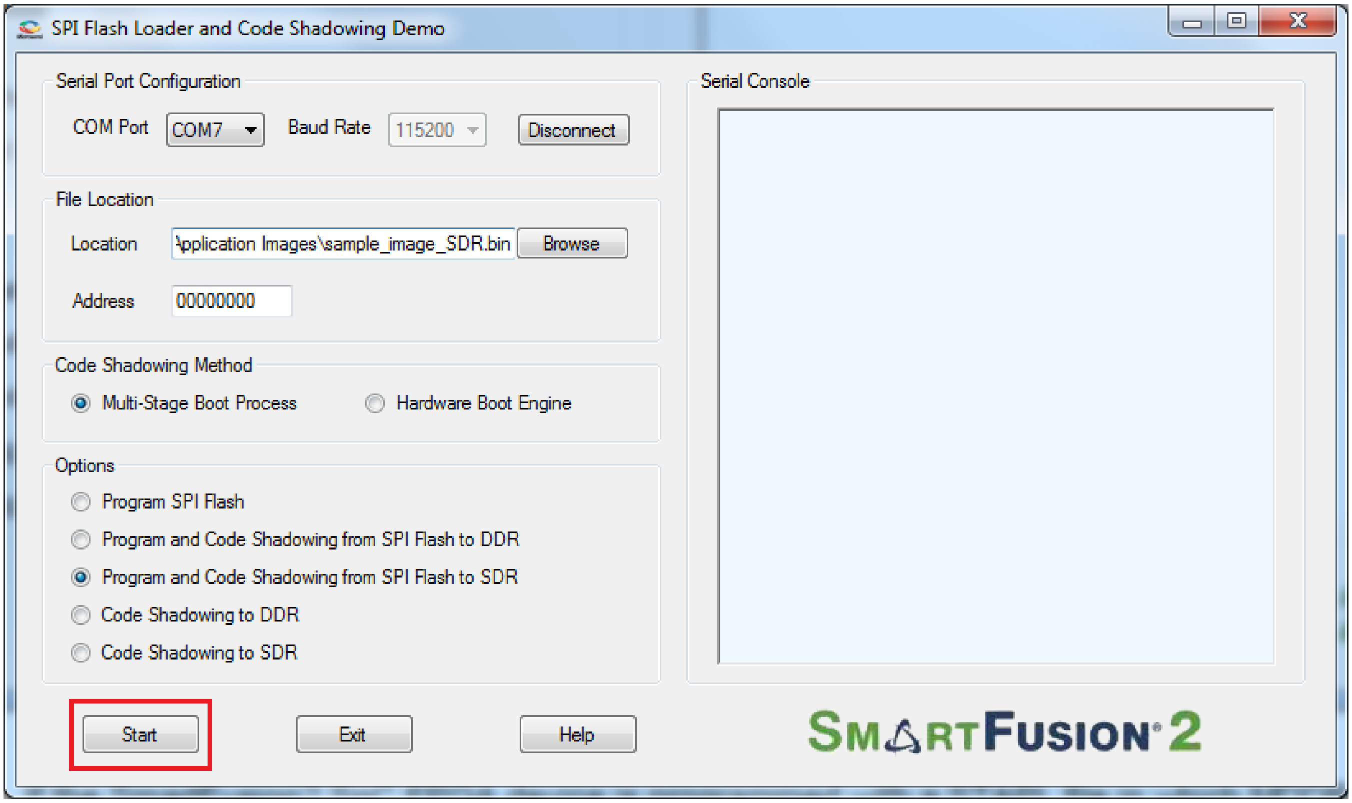
- **Program SPI Flash**: Programs the image file into the SPI flash.
- **Program and Code Shadowing from SPI Flash to DDR**: Programs the image file into the SPI flash, copies it to the DDR memory, and boots the image from DDR memory.
- **Program and Code Shadowing from SPI Flash to SDR**: Programs the image file into the SPI flash, copies it to the SDR memory, and boots the image from SDR memory.
- **Code Shadowing to DDR**: Copies the existing image file from the SPI flash to the DDR memory and boots the image from DDR memory.
- **Code Shadowing to SDR**: Copies the existing image file from the SPI flash to the SDR memory and boots the image from SDR memory.

Click **Help** for more information on the GUI.

Running the Design

1. Switch the power supply switch SW7 to **ON** position.
2. Program the SmarFusion2 SoC FPGA device with the programming file provided in the design files (SF2_CodeShadowing_SDR_DF\Programming Files\CodeShadowing_SDR.stp) using the FlashPro design software.
3. Launch the **SPI Flash Loader and Code Shadowing Demo** GUI executable file available in the design files (SF2_CodeShadowing_SDR_DF\GUI Executable\SF2_FlashLoader.exe). Select the appropriate COM port (to which the USB Serial drivers are pointed) from the **COM Port** drop-down list.
4. Click **Connect**. After establishing the connection, the **Connect** changes to **Disconnect**.
5. Click **Browse** to select the example target executable image file provided with the design files (SF2_CodeShadowing_SDR_DF/Sample Application Images/sampleimage_SDR.bin").

   Note: To generate the Bin files refer to Appendix-B – Generating Executable Bin File.
6. Keep the starting address of the SPI flash memory as default at 0x00000000.
7. Select the **Program and Code Shadowing from SPI Flash to SDR** option.
8. Click **Start** as shown in Figure 7 to load the executable image into SPI flash and code shadowing from SDR memory.



SPI Flash Loader and Code Shadowing Demo

Serial Port Configuration

COM Port: COM7 | Baud Rate: 115200 | Disconnect

File Location

Location: \pplication Images\sample_image_SDR.bin | Browse

Address: 00000000

Code Shadowing Method

● Multi-Stage Boot Process ○ Hardware Boot Engine

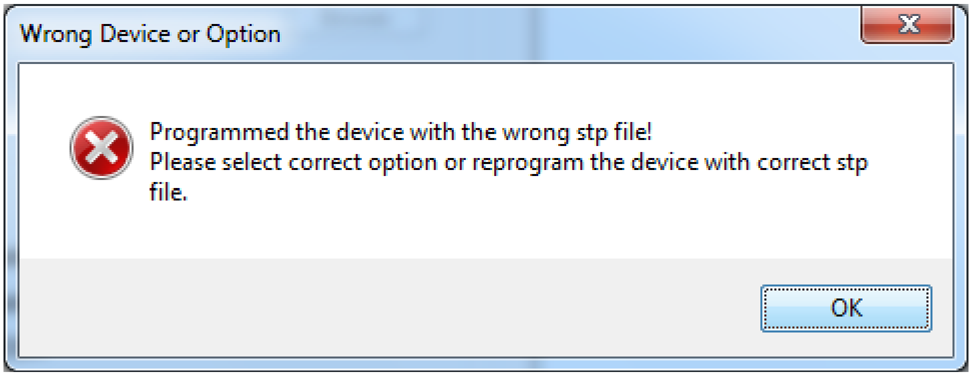
Options

○ Program SPI Flash
○ Program and Code Shadowing from SPI Flash to DDR
● Program and Code Shadowing from SPI Flash to SDR
○ Code Shadowing to DDR
○ Code Shadowing to SDR

Serial Console

Figure 7. Starting the Demo

9. If the SmartFusion2 SoC FPGA device is programmed with a STAPL file in which MDDR is not configured for SDR memory then it shows an error message, as shown in Figure 8.

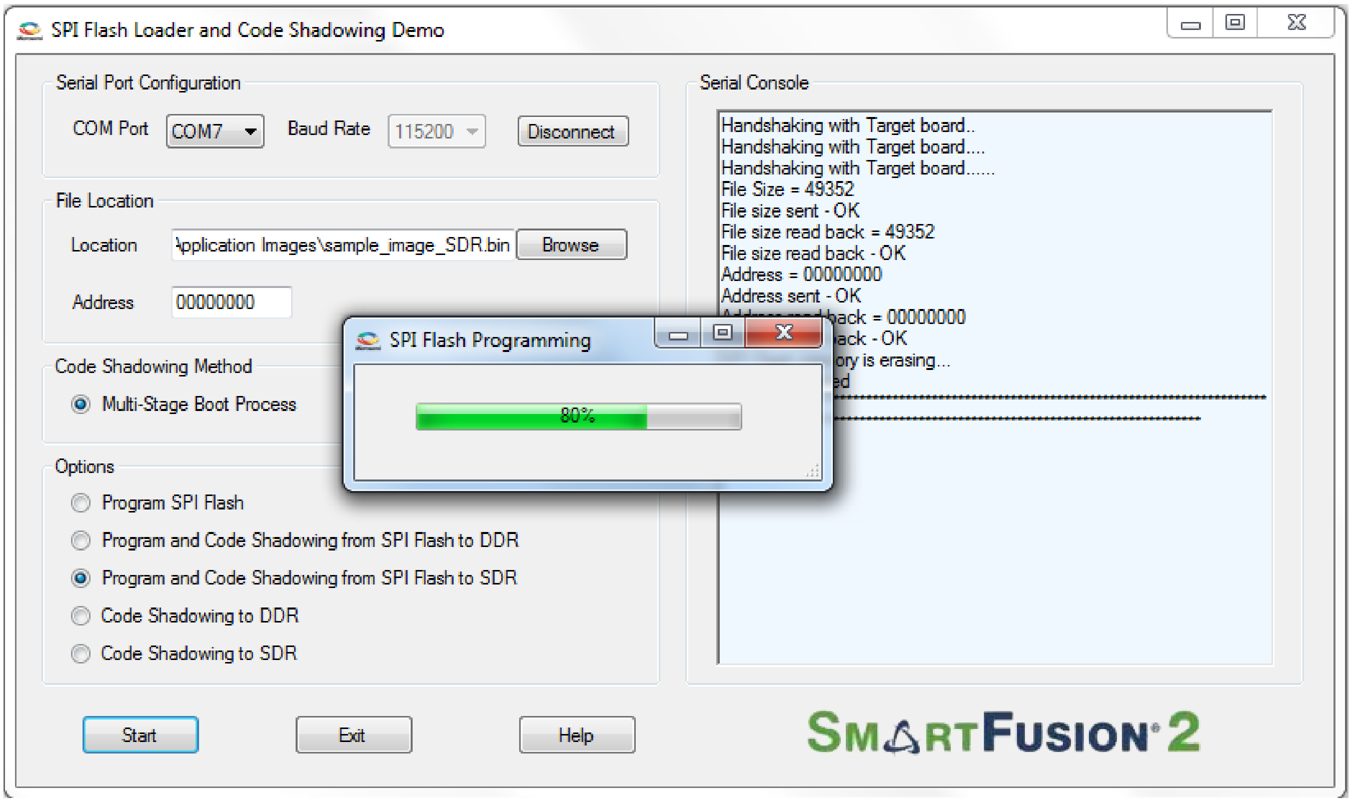


Wrong Device or Option

Programmed the device with the wrong stp file!
Please select correct option or reprogram the device with correct stp file.

Figure 8. Wrong Option

10. The Serial Console section on the GUI shows the debug messages, as shown in **Error! Reference source not found.**
11. The GUI starts programming SPI flash after successfully erasing the SPI flash. The GUI shows the status of SPI flash writing, as shown in Figure 9.



SPI Flash Loader and Code Shadowing Demo

Serial Port Configuration

COM Port: COM7
Baud Rate: 115200

File Location

Location: \pplication Images\sample_image_SDR.bin
Address: 00000000

Code Shadowing Method

● Multi-Stage Boot Process

Options

○ Program SPI Flash
○ Program and Code Shadowing from SPI Flash to DDR
● Program and Code Shadowing from SPI Flash to SDR
○ Code Shadowing to DDR
○ Code Shadowing to SDR

Serial Console

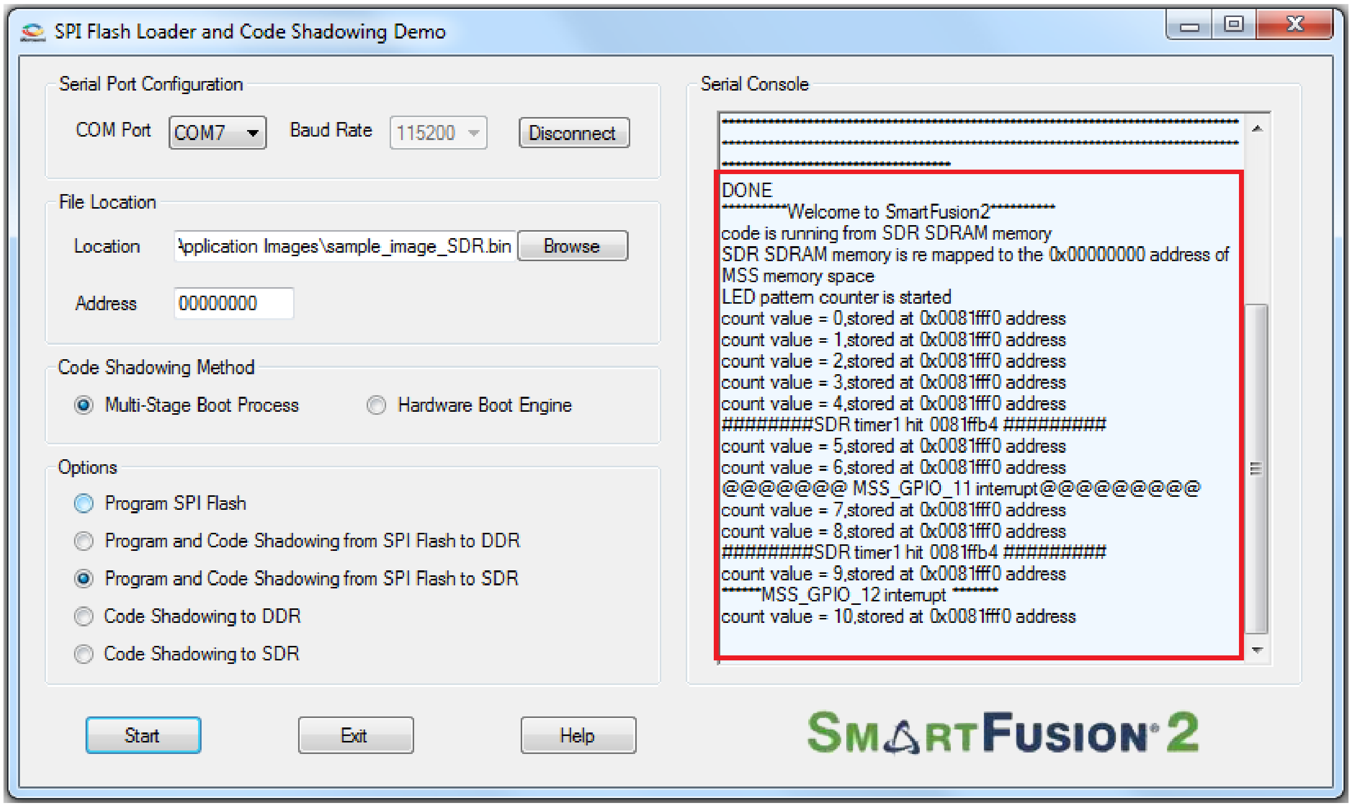
```
Handshaking with Target board..
Handshaking with Target board....
Handshaking with Target board......
File Size = 49352
File size sent - OK
File size read back = 49352
File size read back - OK
Address = 00000000
Address sent - OK
Address read back = 00000000
ack - OK
ory is erasing...
ed
..........................................................................
...............................................................
```

SPI Flash Programming

80%

Figure 9. SPI Flash Programming

12. On programming the SPI flash successfully, the boot loader running on SmartFusion2 SoC FPGA copies the application image from SPI flash to SDR memory and boots the application image. If the provided image sample_image_SDR.bin is selected, the serial console shows the welcome message and timer interrupt messages as shown in Figure 10. A running LED pattern is displayed on LED1 to LED8 on the SmartFusion2 SoC FPGA Development Kit. Press **SW2** and **SW5** switches to see switch interrupt messages on serial console.



Serial Port Configuration

COM Port: COM7

Baud Rate: 115200

File Location

Location: \pplication Images\sample_image_SDR.bin

Address: 00000000

Code Shadowing Method

● Multi-Stage Boot Process
○ Hardware Boot Engine

Options

○ Program SPI Flash
○ Program and Code Shadowing from SPI Flash to DDR
● Program and Code Shadowing from SPI Flash to SDR
○ Code Shadowing to DDR
○ Code Shadowing to SDR

Serial Console

```
****************************************************************************
****************************************************************************
**********************************
DONE
**********Welcome to SmartFusion2**********
code is running from SDR SDRAM memory
SDR SDRAM memory is re mapped to the 0x00000000 address of
MSS memory space
LED pattern counter is started
count value = 0,stored at 0x0081fff0 address
count value = 1,stored at 0x0081fff0 address
count value = 2,stored at 0x0081fff0 address
count value = 3,stored at 0x0081fff0 address
count value = 4,stored at 0x0081fff0 address
#########SDR timer1 hit 0081ffb4 ##########
count value = 5,stored at 0x0081fff0 address
count value = 6,stored at 0x0081fff0 address
@@@@@@@ MSS_GPIO_11 interrupt@@@@@@@@@
count value = 7,stored at 0x0081fff0 address
count value = 8,stored at 0x0081fff0 address
#########SDR timer1 hit 0081ffb4 ##########
count value = 9,stored at 0x0081fff0 address
******MSS_GPIO_12 interrupt *******
count value = 10,stored at 0x0081fff0 address
```

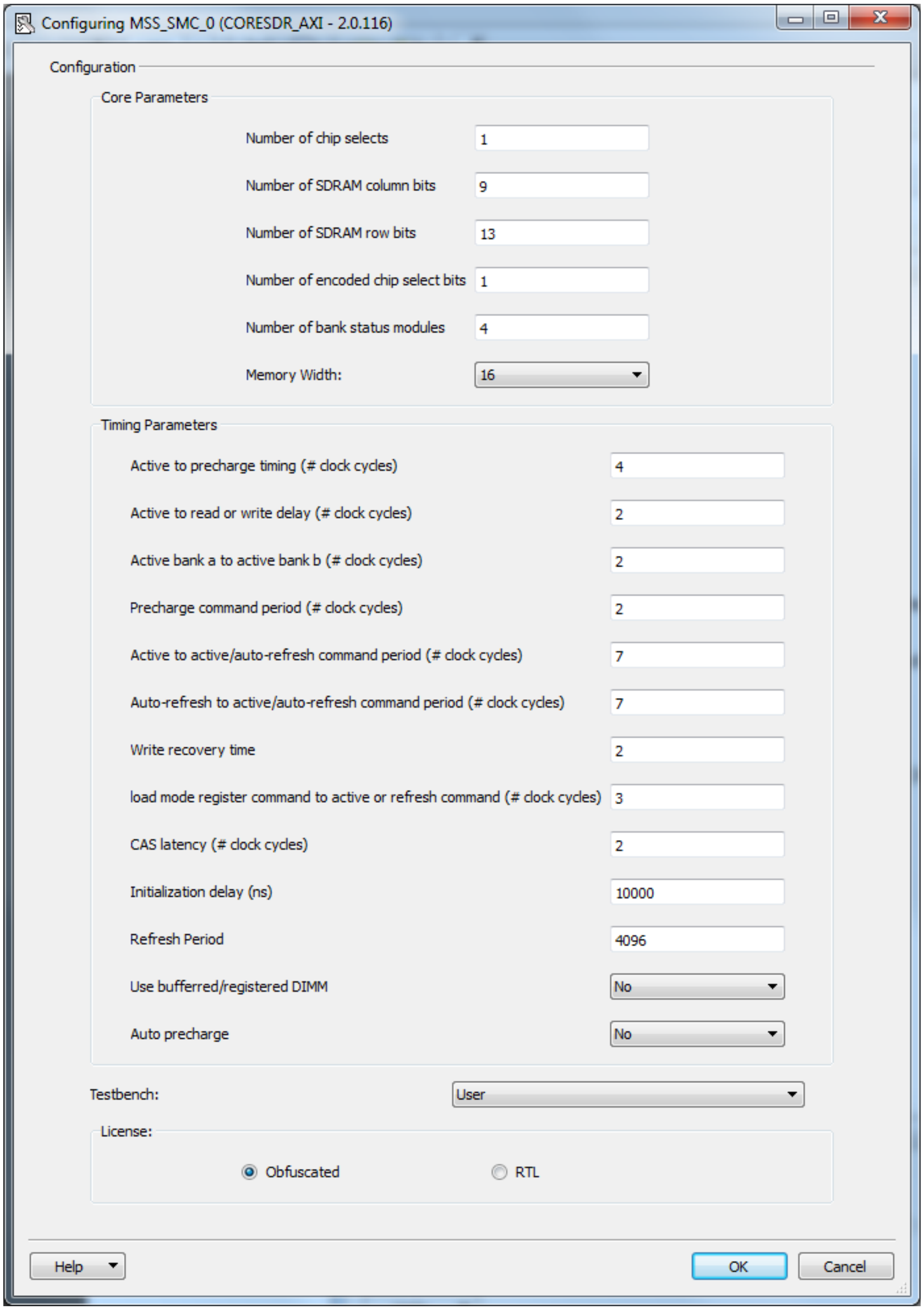
Figure 10. Running Target Application Image from SDR Memory

Conclusion

Th demo shows the capability of SmartFusion2 SoC FPGA device to interface with SDR memory and run the executable image from the SDR memory by shadowing code from SPI flash memory.

Appendix A - SDR Configurations

Figure 11 shows the configuration of CoreSDR_AXI controller to interface with the external SDR SDRAM at 80 MHz speed.



Configuring MSS_SMC_0 (CORESDR_AXI - 2.0.116)

Configuration

Core Parameters

Number of chip selects: 1

Number of SDRAM column bits: 9

Number of SDRAM row bits: 13

Number of encoded chip select bits: 1

Number of bank status modules: 4

Memory Width: 16

Timing Parameters

Active to precharge timing (# clock cycles): 4

Active to read or write delay (# clock cycles): 2

Active bank a to active bank b (# clock cycles): 2

Precharge command period (# clock cycles): 2

Active to active/auto-refresh command period (# clock cycles): 7

Auto-refresh to active/auto-refresh command period (# clock cycles): 7

Write recovery time: 2

load mode register command to active or refresh command (# clock cycles): 3

CAS latency (# clock cycles): 2

Initialization delay (ns): 10000

Refresh Period: 4096

Use bufferred/registered DIMM: No

Auto precharge: No

Testbench: User

License:

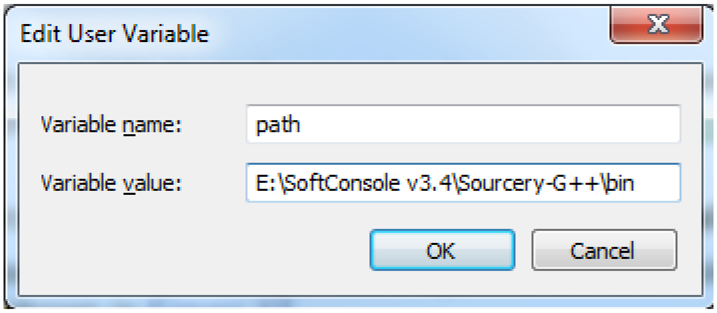
● Obfuscated ○ RTL

Figure 11. SDRAM Controller Configurations

Appendix B - Generating Executable Bin File

The executable bin file is required to program the SPI flash for running the code shadowing demo. To generate the executable bin file from “sample_image_SDR” SoftConsole project, follow the below steps:

1. Build the SoftConsole project with the linkerscript “production-execute-in-place-externalSDR.ld”.
2. Add the SoftConsole installation path, for example, C:\Microsemi\Libero_v11.4\SoftConsole\Sourcery-G++\bin, to the ‘Environment Variables’ as shown below.



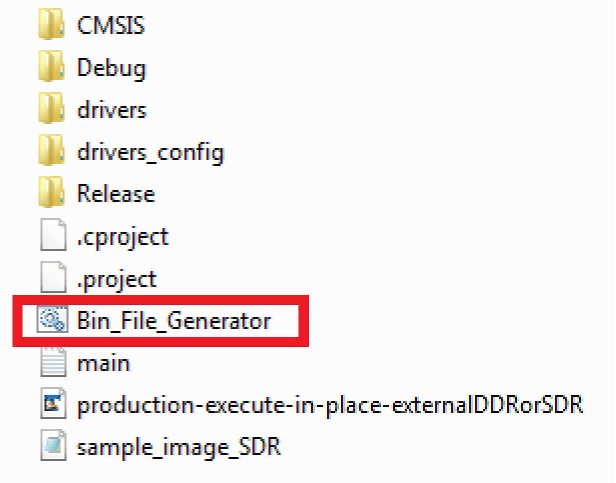
Edit User Variable

Variable name: path

Variable value: E:\SoftConsole v3.4\Sourcery-G++\bin

Figure 12. Adding SoftConsole Installation Path

3. Double-click batch file **Bin-File-Generator** in the SoftConsole/CodeShadowing_SDR_MSS_CM3/Sample_image_SDR folder, as shown below.



CMSIS
Debug
drivers
drivers_config
Release
.cproject
.project
Bin_File_Generator
main
production-execute-in-place-externalDDRorSDR
sample_image_SDR

Figure 13. Bin File Generator

4. The **Bin-File-Generator** creates “sample_image_SDR.bin” file.

List of Changes

Revision	Changes	Page
Revision 5 (September 2014)	Updated the document for Libero SoC 11.4 software release (SAR 60387).	NA
Revision 4 (April 2014)	Updated the document for Libero SoC 11.3 software release (SAR 56872).	NA
Revision 3 (November 2013)	Updated the document for Libero SoC 11.2 software release (SAR 52965).	NA
Revision 2 (June 2013)	Updated the document for Libero SoC 11.0 software release (SAR 47704).	NA
Revision 1 (March 2013)	Updated the document for Libero SoC 11.0 Beta SP1 software release (SAR 44873).	NA

Note: The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

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/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

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

Outside the U.S.

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

ITAR Technical Support

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

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs, and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.