



Using Serial Flash on the Xilinx Spartan-3E Starter Board

Version 9.2
October 18, 2007
Bryan H. Fletcher

Overview

The Xilinx Spartan-3E FPGA features the ability to configure from standard serial flash over a built-in Serial Peripheral Interface (SPI). Being general-purpose flash, the SPI serial flash can also be used for any other non-volatile storage that the user may need. One such non-volatile purpose is the storage of MicroBlaze processor application code for bootloading.

Objectives

The Xilinx Spartan-3E Starter Board features an STMicro M25P16 serial flash memory. This reference design demonstrates several aspects of using this serial flash and the Spartan-3E FPGA on the Xilinx Spartan-3E Starter Board, including:

- FPGA configuration over SPI
 - Store bitstream to serial flash
 - Configure FPGA from serial flash
- MicroBlaze test application utilizing the STMicro M25P16
 - Read manufacturer's ID
 - Perform a bulk erase
 - Write to all locations in the flash
 - Read from all locations in the flash
- MicroBlaze interactive user application utilizing the STMicro M25P16 for data program data storage
 - Read from a designated sector
 - Perform a sector erase
 - Write to a sector
- MicroBlaze bootloader application
 - Merge a configuration bitstream and binary MicroBlaze application
 - Store merged file to serial flash
 - Copy application image from serial flash to external memory
 - Run from external memory

© 2007 Avnet. All rights reserved. All trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Avnet is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Avnet makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Avnet expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Experiment Setup

Software Prerequisites

The software used to test this reference design is:

- WindowsXP 32-bit
- Xilinx ISE 9.2i with Service Pack 3¹
- Xilinx EDK 9.2 with Service Pack 1²

Hardware Prerequisites

The hardware setup used by this reference design includes:

- Computer with a minimum of 512MB RAM³
- Xilinx Spartan-3E Starter Board (Rev C or Rev D)
- Xilinx Parallel Cable IV (PC4) or Platform USB Cable with flyleads
- Serial Cable

Setup

- Jumper settings:
 - Install J30 jumper in position 3-4, which is M1 (M[2:0] = 101, Boundary Scan Mode). An additional jumper for M2 is needed later.
 - Install two jumpers on J11, positions 1-2 and 3-4.
 - Install a jumper on J9, positions 2-3 (3.3V)
 - Install jumpers on JP6 and JP7
 - All other jumpers NOT installed
- Cables:
 - Do one of the following:
 - Plug in a USB cable between the PC and the USB port (J18) on the board
 - Plug in the PC4 using flyleads to J28
 - Plug in the USB Platform cable using flyleads to J28
 - Plug in the serial cable between the PC RS232 port and the DCE RS232 connector (J9) on the board. (The DTE RS232 connector (J10) can also be used for the serial connection but the STDIN/STDOUT setting must be modified in the Software Platform Settings.)
- Files:
 - Unzip the Xil3S500E_Serial_Flash_v92.zip file to a folder of your choosing, making sure there are no spaces in the pathname.
- Project:
 - An XPS project is included with this reference design.

¹ ISE latest Service Pack is available at www.xilinx.com/download

² EDK latest Service Pack is available at www.xilinx.com/download

³ Refer to www.xilinx.com/ise/products/memory.htm

Experiment 1: Create and test a bitstream

This experiment uses a simple “Hello World” MicroBlaze design that prints to the UART. This experiment will ensure the software, jumpers, and cables are all installed properly. During this experiment, you will create a bitstream for this project and test it by downloading the bitstream directly to the FPGA via JTAG.

1. Launch Xilinx Platform Studio (XPS) by selecting **Start → Programs → Xilinx Platform Studio 9.2 → Xilinx Platform Studio**
2. Select **File → Open Project**. Browse to the `xil3S500E_Serial_Flash_v92` folder, select `system.xmp`, and click **Open**.
3. Select the *Applications* tab. Select the **hello_world** project for BRAM initialization by right-clicking on the project and selecting **Mark to Initialize BRAMs**. No other applications should be marked for BRAM initialization. The *Applications* tab should look like Figure 1.

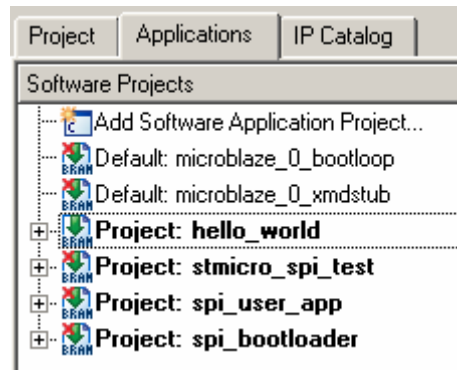


Figure 1 – Project: hello_world Selected for BRAM Initialization

4. Select **Device Configuration → Update Bitstream**. This will generate the MicroBlaze hardware platform, build the Board Support Package (BSP), compile the project, and initialize the bitstream with the application code. The result of this operation is an FPGA bitstream located at:
`xil3S500E_Serial_Flash_v92\implementation\download.bit`
5. Launch a HyperTerminal connected to the RS232 COM port with the settings 115200 bps, 8 data bits, 1 stop bit, no parity, and no flow control. Alternatively, double-click on the `com1_115200_8n1n.ht` file in the project directory to launch HyperTerminal with the appropriate settings.
6. Plug power into the Spartan-3E starter board. Turn the Power Switch to the ON position. The POWER LED should light.
7. In XPS, select **Device Configuration → Download Bitstream** which will download the `download.bit` file to the Spartan-3E FPGA. When the download is complete, the XC-DONE LED should light, and the HyperTerminal shows Hello World.

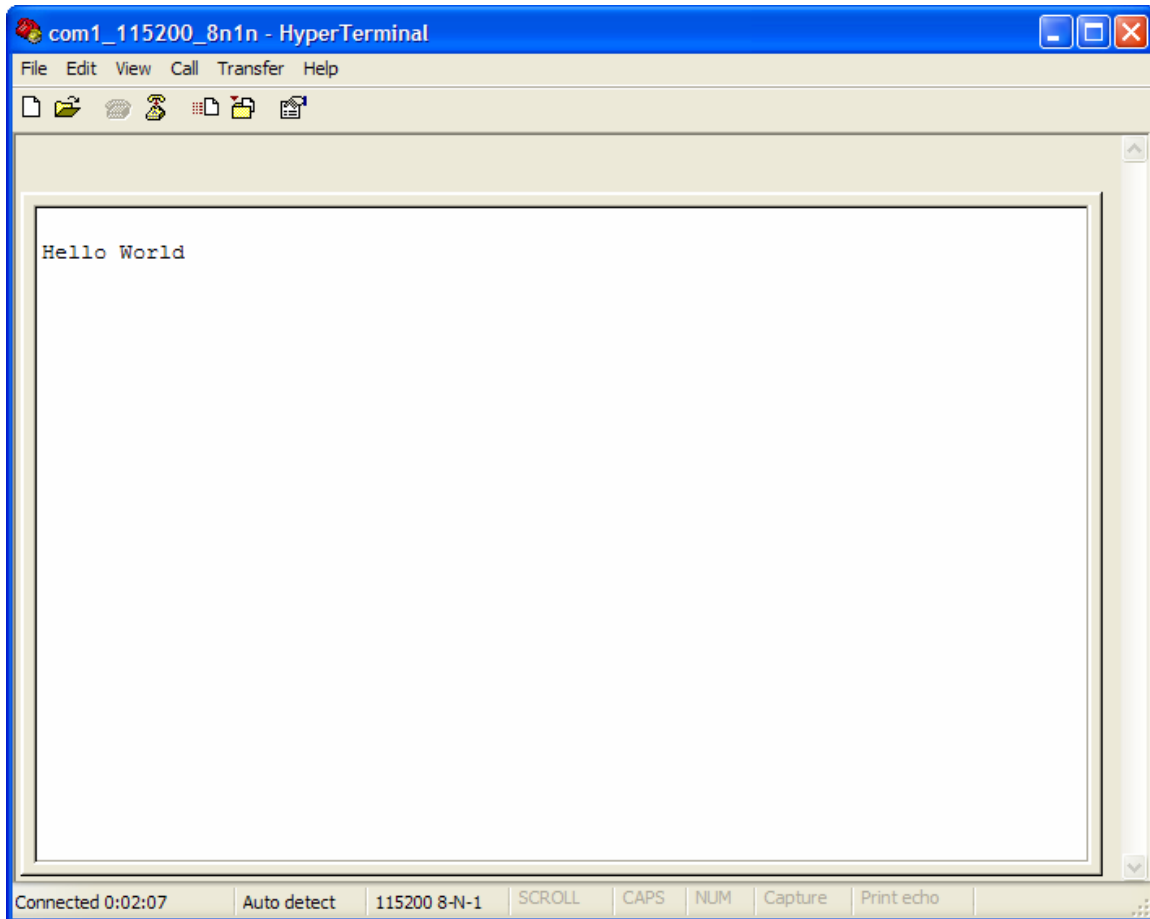


Figure 2 – Hello World Downloaded to FPGA

Experiment 2: Configure from serial flash

This experiment stores the bitstream from Experiment 1 in the serial flash and then configures the Spartan-3E FPGA from that flash. iMPACT is used in this experiment to generate the SPI image file and to burn the SPI using Direct SPI Configuration. Indirect SPI Configuration through iMPACT would be more convenient, but it is not supported for Spartan-3E at this time.

1. Turn Power OFF.
2. If plugged in previously, disconnect the USB cable from the board.

3. Plug either the PC4 or USB JTAG cable into the PC and the SPI connector (J12). If flyleads are used, match the signals as follows:

FLYLEADS	BOARD J12
VREF	VCC
GND	GND
TCK	SCK
TDO	SDO
TDI	SDI
TMS	SEL

4. Verify the MODE header (J30) is set to Boundary Scan mode (M1 jumper only – M[2:0] = 101)
5. Add a jumper to JP8 to ground the FPGA PROG pin.
6. Turn power ON.
7. Launch iMPACT. Create a new project in the project directory with a name of your choice. Click **OK**.

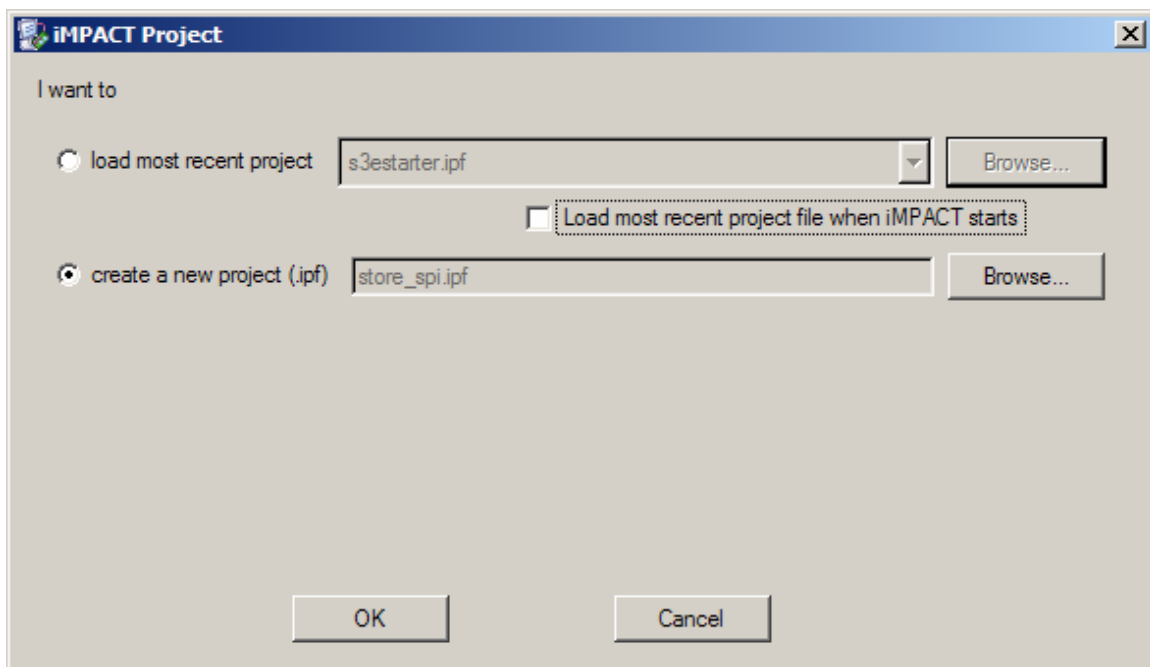


Figure 3 – Create New iMPACT Project

8. Select **Prepare a PROM File** and click **Next >**

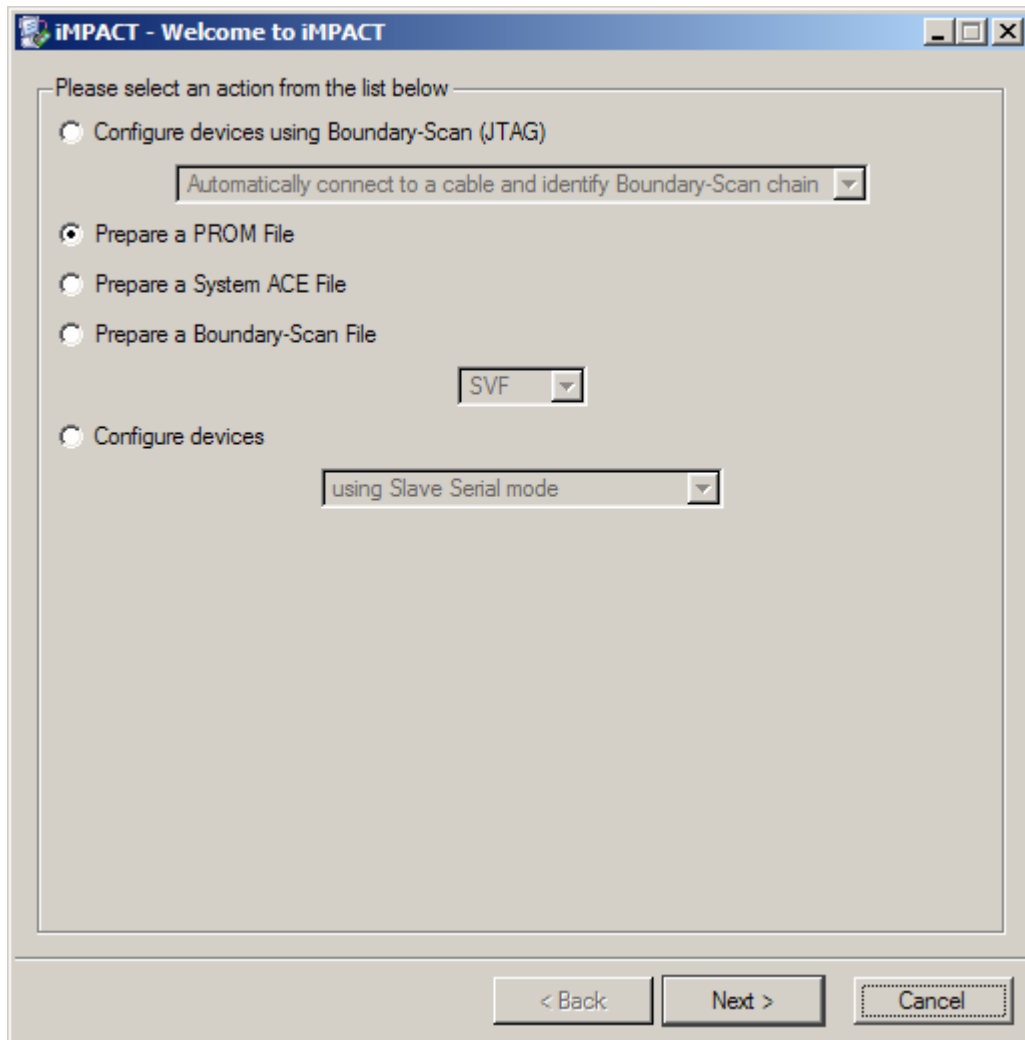
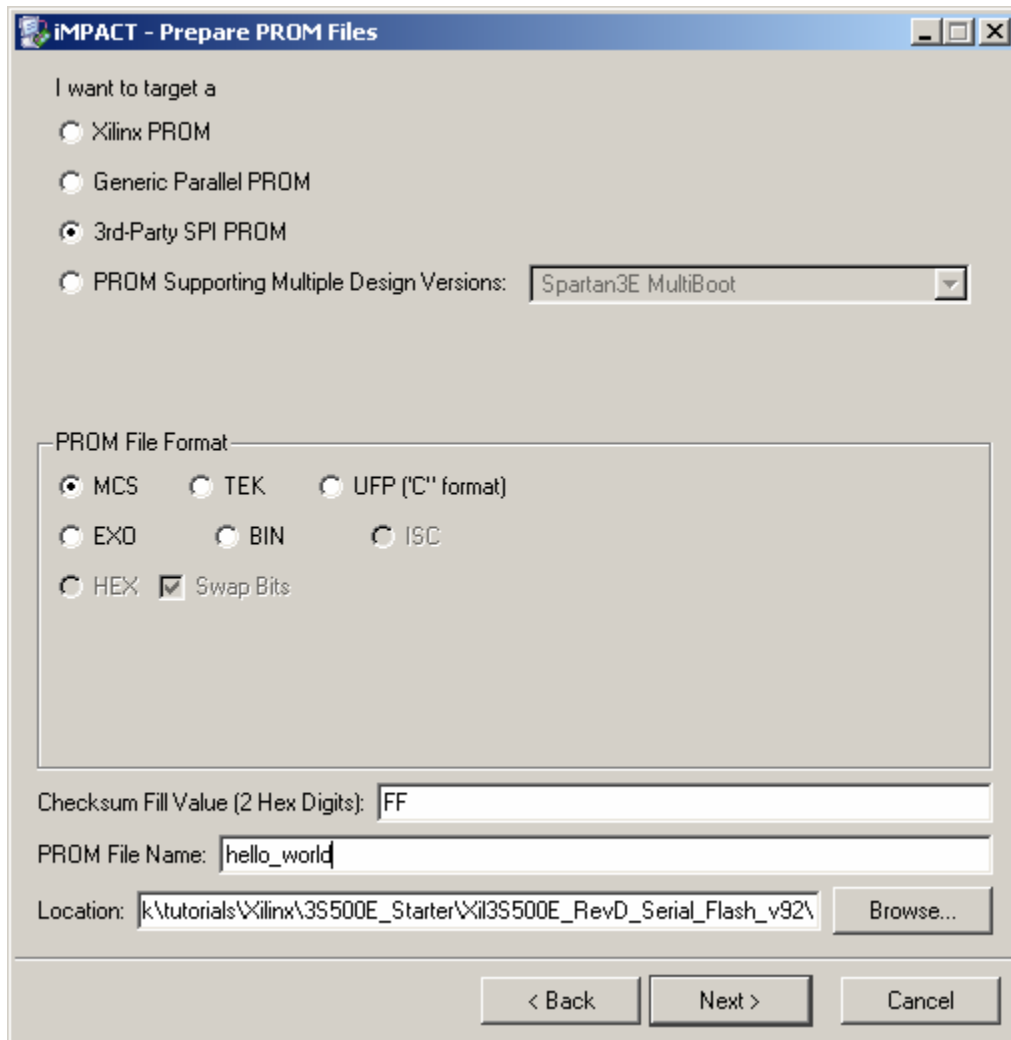


Figure 4 – Prepare a PROM File

9. Select 3rd-Party SPI PROM, give the PROM a File Name, and click Next >



The image shows a screenshot of the 'iMPACT - Prepare PROM Files' dialog box. The window has a title bar with the text 'iMPACT - Prepare PROM Files' and standard Windows window controls. Inside the dialog, there is a section titled 'I want to target a' with four radio button options: 'Xilinx PROM', 'Generic Parallel PROM', '3rd-Party SPI PROM' (which is selected), and 'PROM Supporting Multiple Design Versions:'. The last option has a dropdown menu showing 'Spartan3E MultiBoot'. Below this is a section titled 'PROM File Format' with several radio button options: 'MCS' (selected), 'TEK', 'UFP ("C" format)', 'EXD', 'BIN', 'ISC', 'HEX', and a checked checkbox for 'Swap Bits'. At the bottom of the dialog, there are three text input fields: 'Checksum Fill Value (2 Hex Digits):' with 'FF' entered, 'PROM File Name:' with 'hello_world' entered, and 'Location:' with 'k\tutorials\Xilinx\3S500E_Starter\Xil3S500E_RevD_Serial_Flash_v92\' entered. To the right of the 'Location' field is a 'Browse...' button. At the very bottom of the dialog are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 5 – Prepare 3rd-party SPI PROM File

10. The PROM density on this board is 16Mbits.

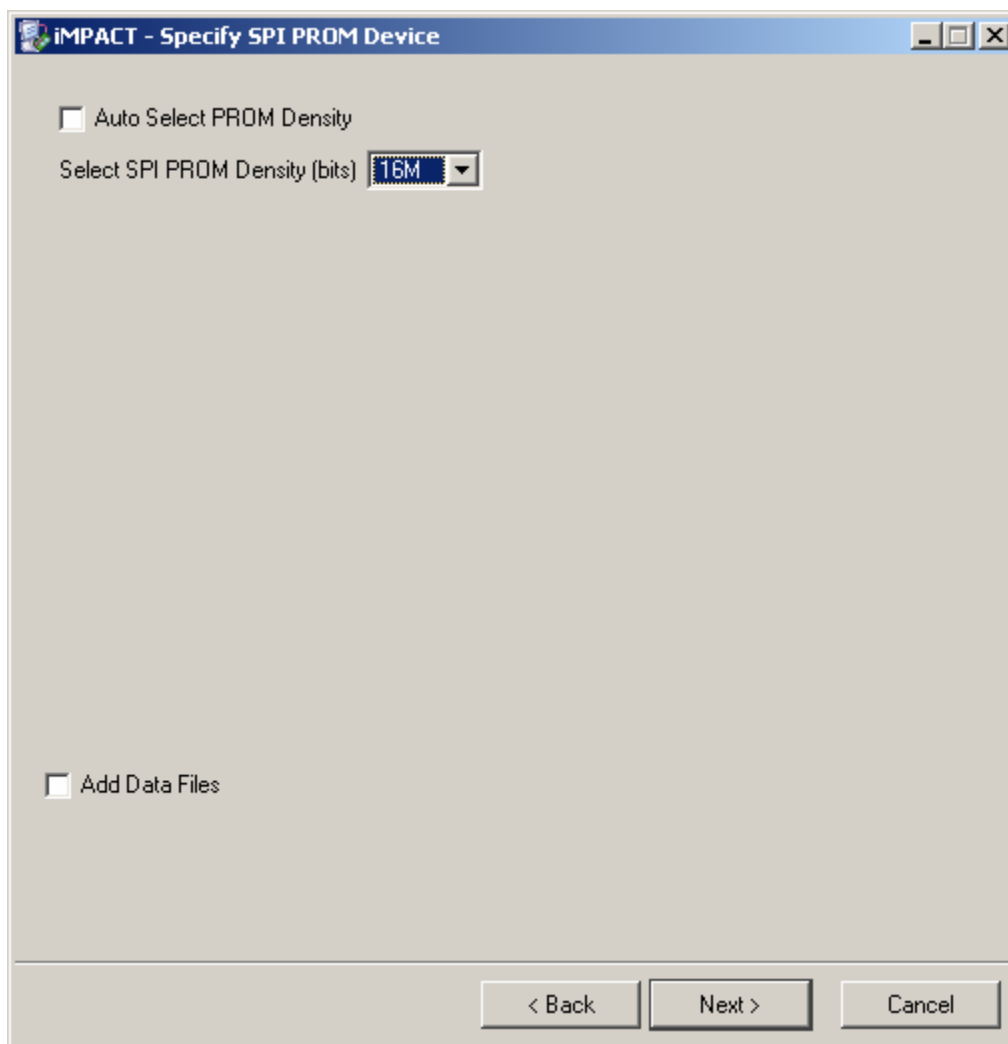


Figure 6 – SPI PROM is 16Mbits

11. Click **Finish** at the Summary.

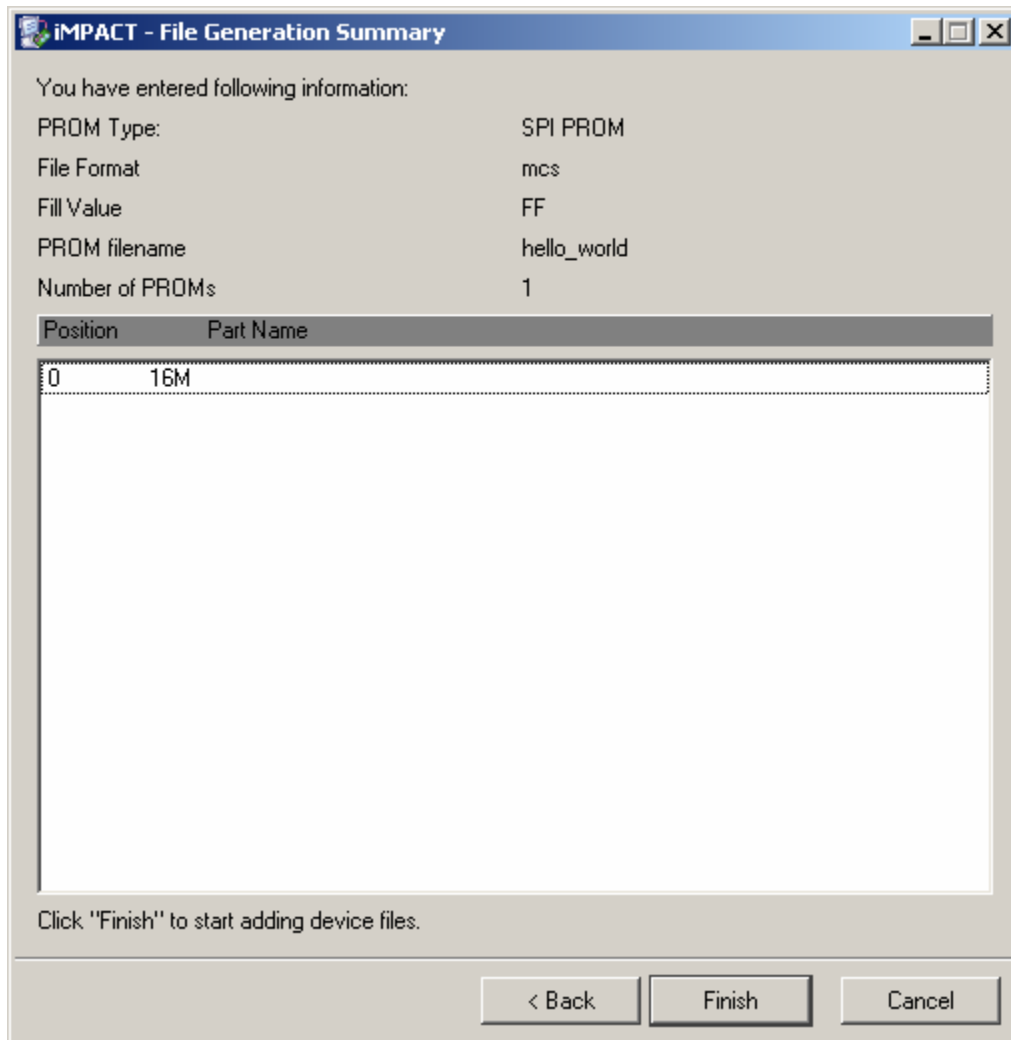


Figure 7 – File Generation Summary

12. Click OK to begin adding device files.

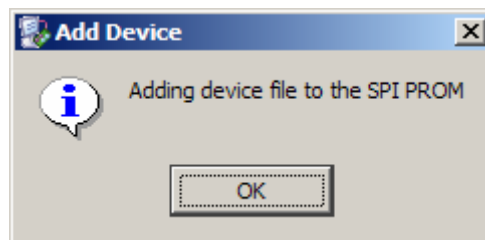


Figure 8 – Adding device files

13. Browse to `<project directory>/implementation/download.bit` and select it as the design file.
14. Click **OK** to acknowledge the warning about the startup clock changing to CCLK.

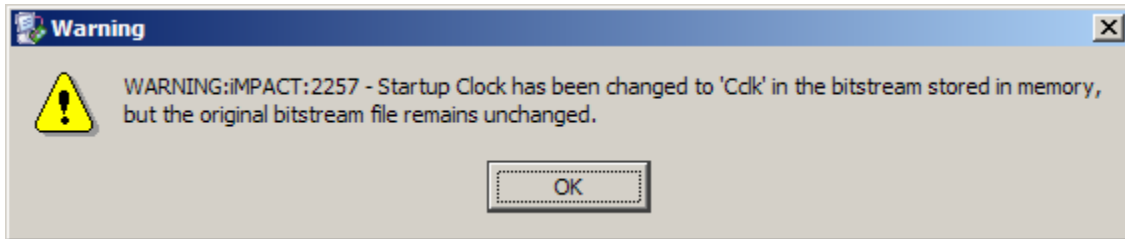


Figure 9 – Startup Clock Changed to CCLK

15. Click **No** since another device file will not be added.

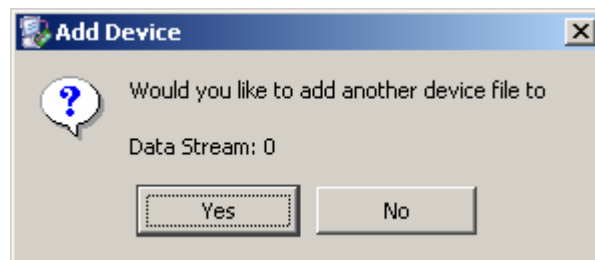


Figure 10 – No Other Device Files to be Added

16. Click **OK**.

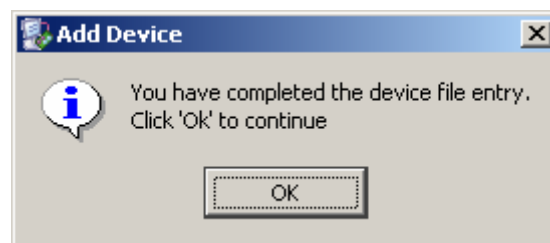


Figure 11 – Device Entry Complete

17. Select **Operations → Generate File**. A successful file generation is shown below.

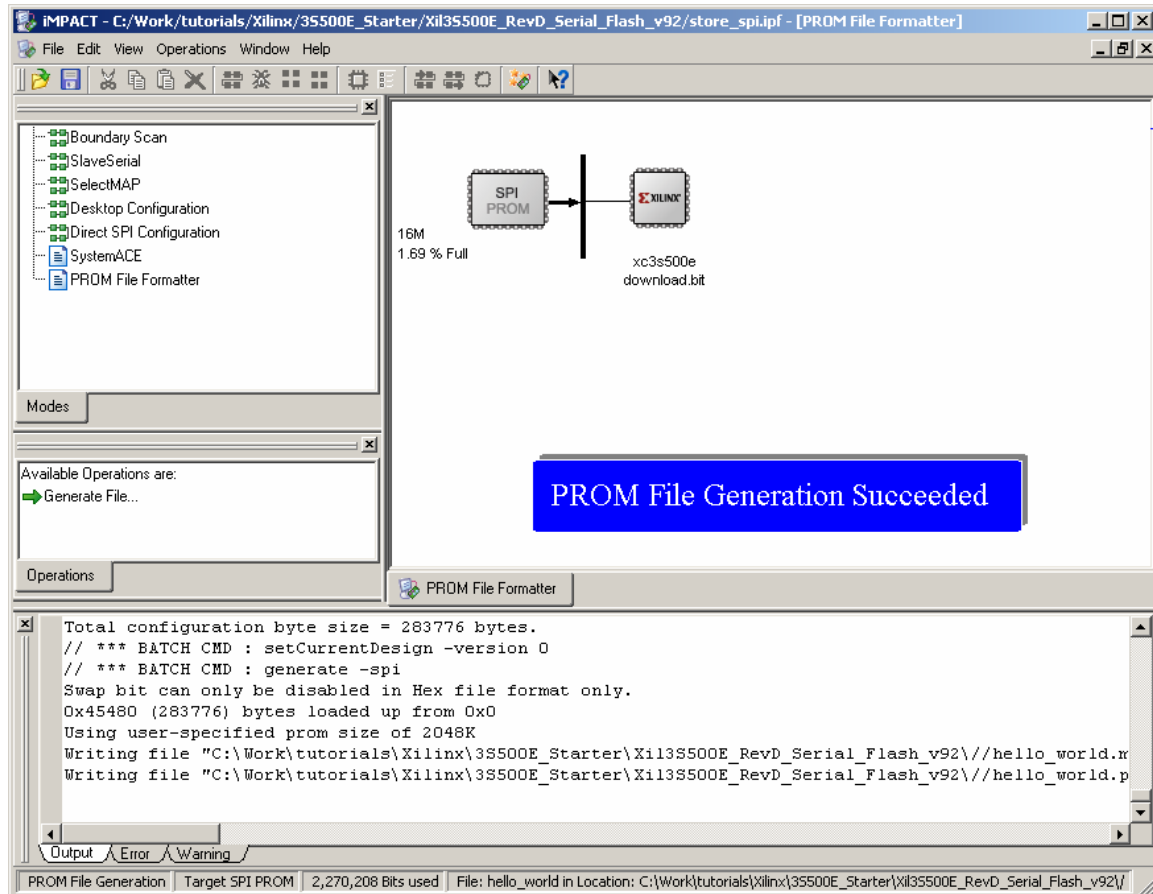


Figure 12 – SPI PROM File Generated

18. Double-click on the **Direct SPI Configuration** item in the *Modes* tab.
19. Right-click and select **Add SPI Device**.
20. Browse and select the previously generated MCS (hello_world.mcs).
21. Select M25P16 and click **OK**.

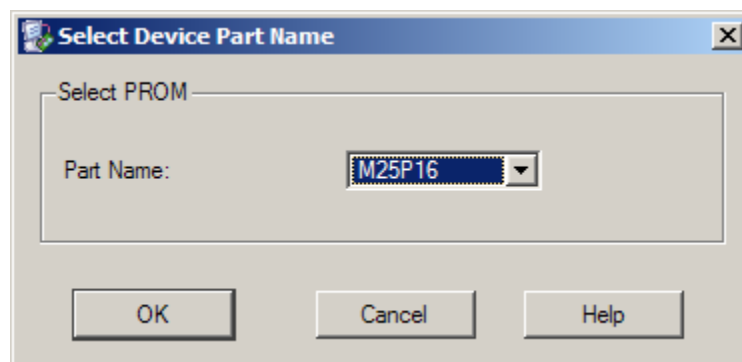


Figure 13 – Device is M25P16

22. In iMPACT, select **Operations** → **Program**.

23. Select **Verify** and **Erase** options and click **OK**.

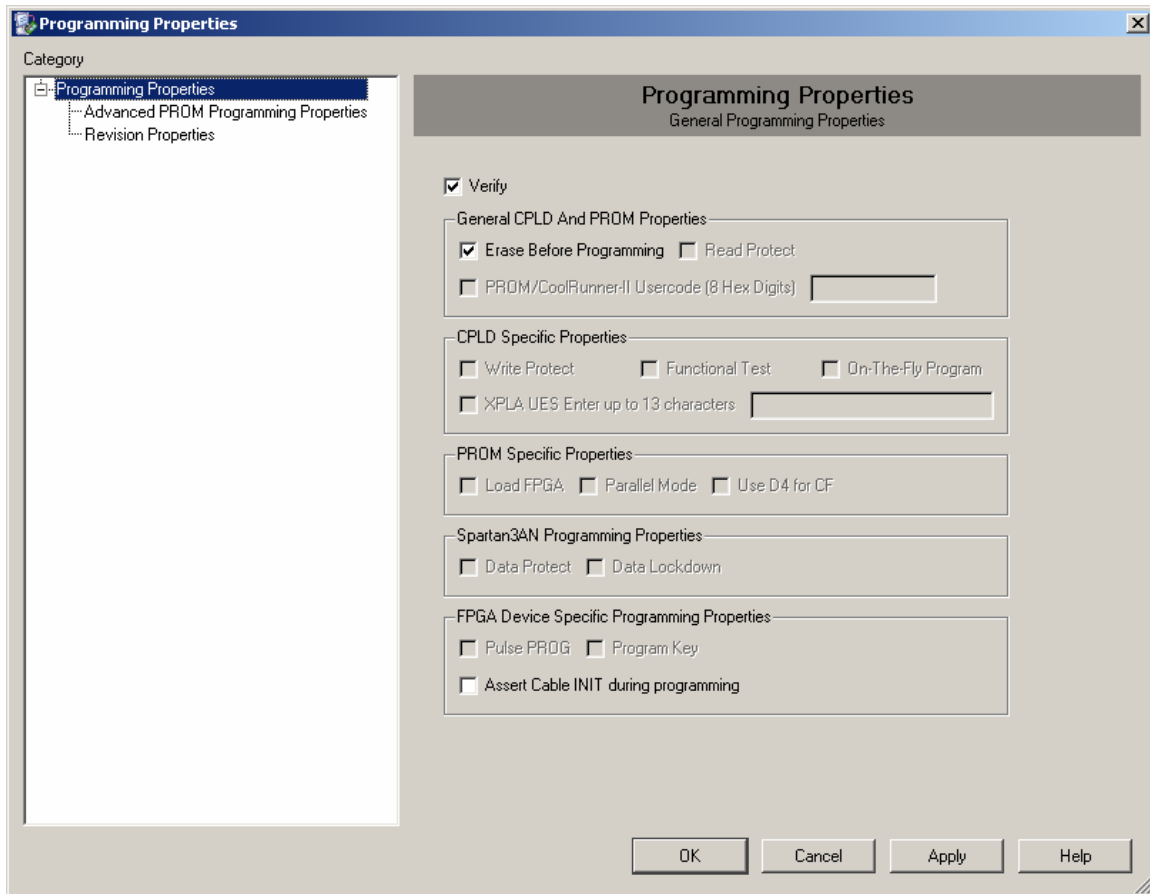


Figure 14 – Programming M25P16 Options

24. Click **OK**. PROG_B is being driven low by jumper JP8.

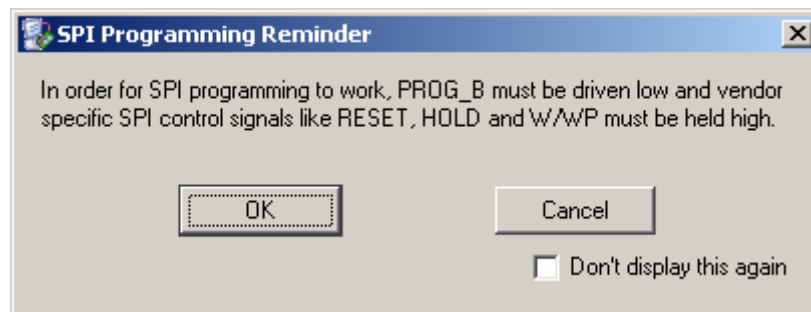


Figure 15 – SPI Programming Reminder

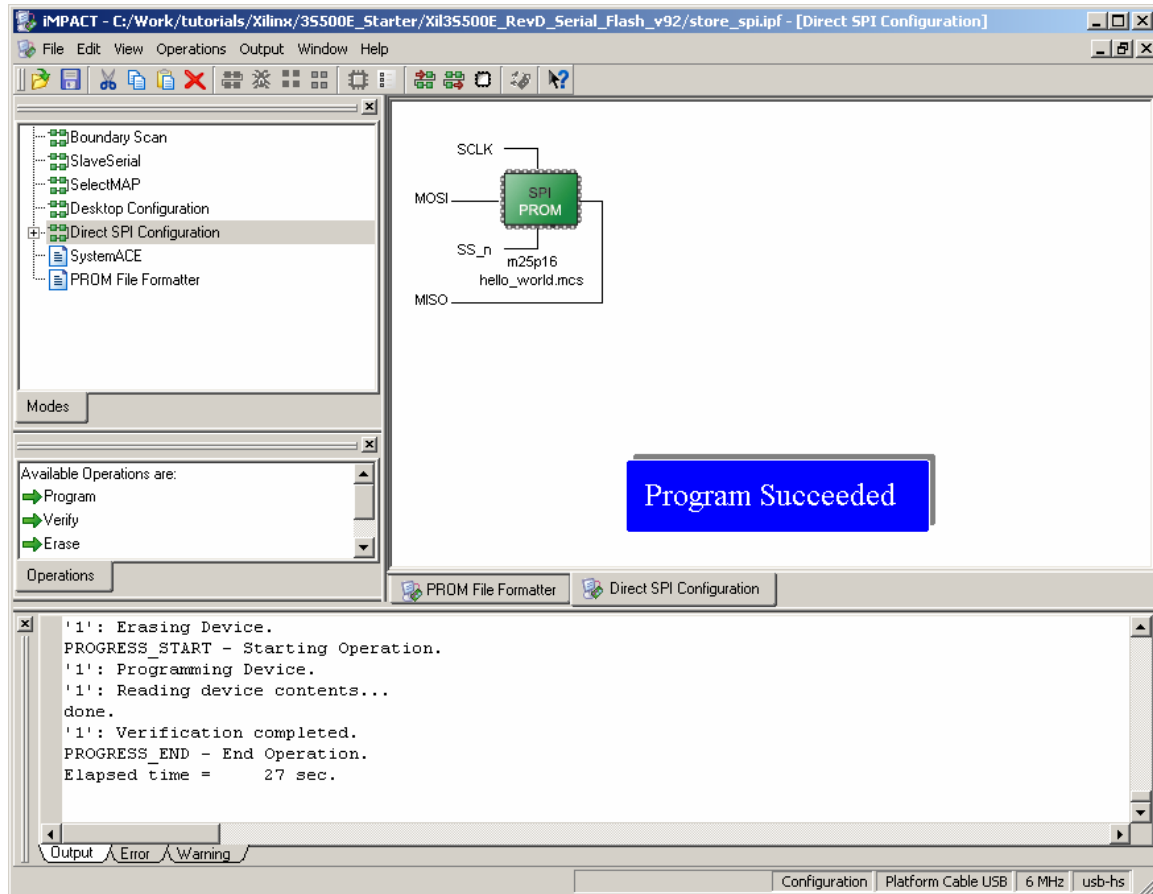


Figure 16 – Programming Completed Successfully

25. Turn power OFF.
26. Unplug the JTAG flyleads from J12 (required for proper SPI access by the FPGA).
27. Remove the jumper on JP8.
28. Add a jumper to M2 (J30.5-6), setting the MODE to SPI Configuration Mode ($M[2:0] = 0:0:1$).

Note that the Vendor Select pins for the FPGA are strapped by default on the Xilinx Spartan-3E Starter board to the proper value for the STMicro M25P16 flash, which is $VS[2:0] = 1:1:1$.

29. If previously closed, re-open the HyperTerminal (com1_115200_8n1n.ht)
30. Turn power on. DONE should light indicating the FPGA is configured.

You should see the same application that we saw previously in Experiment 1, but now the bitstream is stored and configured from low-cost, SPI Flash!

31. Turn power OFF.

Experiment 3: Exercise serial flash from MicroBlaze

A sample application to test the serial flash is included. This application uses the same driver code that is used in Experiment 4's user application. The results are shown in HyperTerminal. The test application does the following:

- Read the manufacturer's ID
- Perform a bulk erase
- Write to all locations in the flash
- Read from all locations in the flash

*NOTE: Because of the jumper placement on J11 (1-2 and 3-4), the serial flash select signal (ROM_CS) is tied to both CSO_B (U3) and SEL (R12). This jumper placement is necessary to allow either the programming cable (using the SEL signal) or the FPGA (using the CSO_B signal) to access the serial flash. However, SEL is an unused pin in the MicroBlaze design. The default for unused pins in ISE is to pull them down (see bitgen defaults), which causes a conflict with the SPI controller. Therefore, the following statement was previously added to the bitgen.ut file to prevent SEL from interfering with the serial flash operation:

-g UnusedPin:Pullnone

1. In XPS, mark the **stmicro_spi_test** for BRAM initialization and unmark all others, as shown in Figure 17.

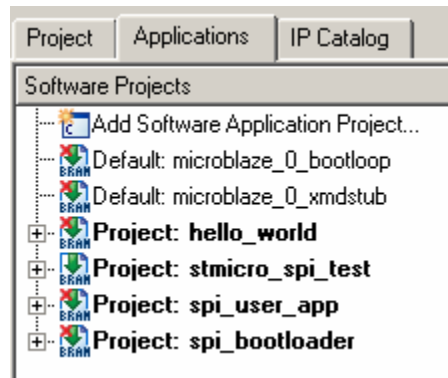


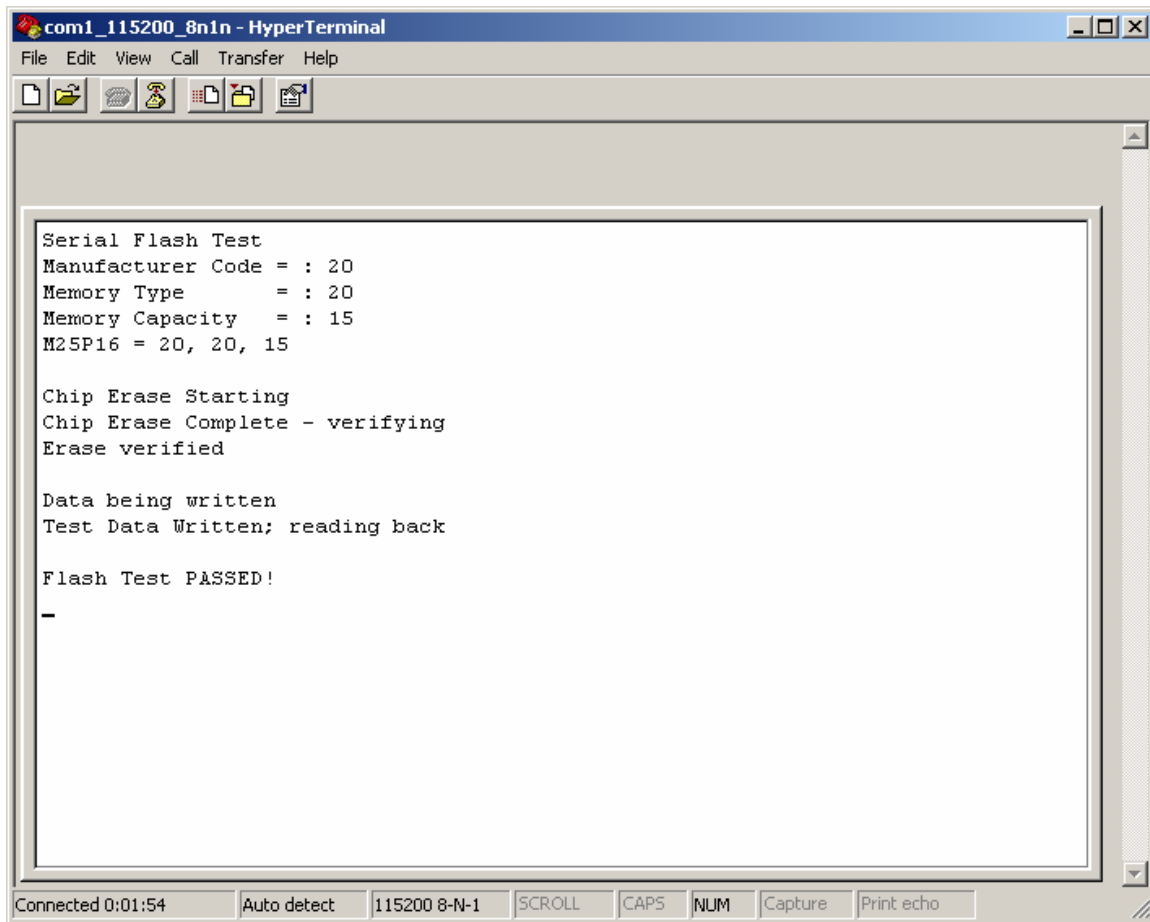
Figure 17 – Project: stmicro_spi_test Marked for BRAM Initialization

2. Take a moment to browse the code to become familiar with what the application is doing.
3. Select **Device Configuration → Update Bitstream** to compile this project and create a new `download.bit` bitstream with this application.

Before downloading, a few changes are made to the board.

4. Return the J30 jumpers to Boundary Scan Mode (M1 only installed).
5. Do one of the following:
 - a. Plug the PC4 or USB JTAG cable to the JTAG connector (J28).
 - b. Plug in the USB cable.
6. Turn power ON.
7. If previously closed, re-open the HyperTerminal (com1_115200_8n1n.ht)
8. Download the bitstream to the board.

The complete test takes less than two minutes to run. Results are shown in Figure 18.



```
com1_115200_8n1n - HyperTerminal
File Edit View Call Transfer Help

Serial Flash Test
Manufacturer Code = : 20
Memory Type      = : 20
Memory Capacity  = : 15
M25P16 = 20, 20, 15

Chip Erase Starting
Chip Erase Complete - verifying
Erase verified

Data being written
Test Data Written; reading back

Flash Test PASSED!
-

Connected 0:01:54  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Figure 18 – Serial Flash Test Results

9. Turn power OFF.

Experiment 4: Bootload MicroBlaze from serial flash

This experiment shows how a MicroBlaze application is stored in serial flash device and then bootloaded after configuration. The process to be completed in this experiment is:

- Configure the FPGA from serial flash with a bitstream containing a MicroBlaze hardware platform and BRAM contents initialized with a bootloader application.
- The bootloader accesses a pre-determined location in the serial flash and copies a stored user application from flash to DDR.
- The bootloader application then jumps to the user application in DDR and begins running.

The user application does the following:

- Reads and displays a 16-character location in serial flash and displays the user string.
- Prompts the user to enter a new string
- Accepts a new 16-character string from the UART
- Stores the string in serial flash
- Instructs you to reconfigure to show that your new string was stored in flash

This experiment uses the serial flash for three separate functions: FPGA configuration, MicroBlaze application bootloading, and user data storage.

1. Take a moment to browse through the code for the two applications that are used during this experiment:
 - **Project: spi_user_app.** A binary version of this application will be stored in serial flash and later bootloaded.
 - **Project: spi_bootloader.** This application is stored in BRAM and launches immediately after the MicroBlaze is configured.
2. In XPS, mark the **Project: spi_bootloader** for BRAM Initialization and unmark all others.

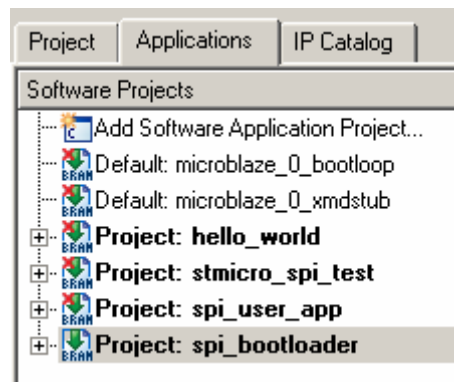


Figure 19 – SPI Bootloader Marked for BRAM Initialization

3. Right-click on **Project: spi_user_app** and select **Set Compiler Options...** Note that Program Start Address is set to 0x8C000000 which is the base address of the DDR memory. Click **Cancel** to close this window.

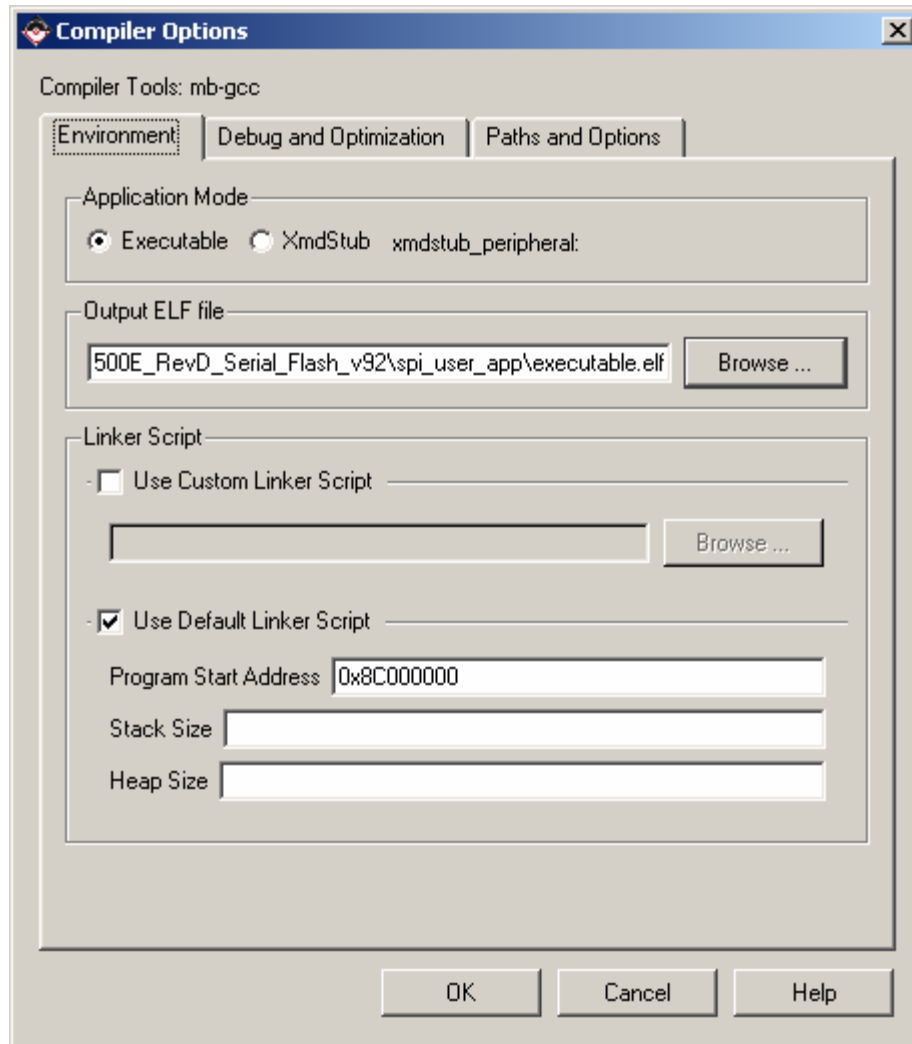


Figure 20 – spi_user_app Compiler Options

4. Right-click on **Project: spi_user_app** and select **Build Project**. This creates the ELF file: `xil3S500E_Serial_Flash_v92\spi_user_app\executable.elf`

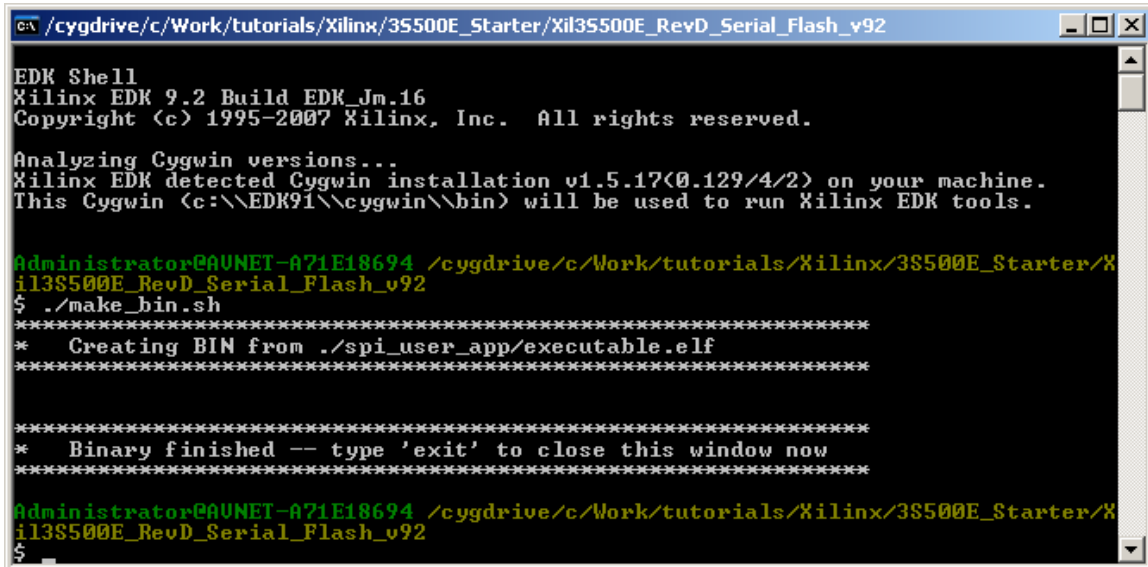
This application must be converted from ELF to binary format for use with the programming utilities.

5. Open a cygwin shell window by selecting **Project → Launch EDK Shell...**

A script is provided for ease of use. The script uses the mb-objcopy utility to convert `spi_user_app\executable.elf` to `FLASH_BURN\spi_user_app.b`. Options are included to exclude several initialization vectors that will be included as part of the bootloader. The syntax format to perform this on the command line of the shell window is:

```
mb-objcopy -O binary <options> <ELF file input> <binary file to output>
```

6. In the shell window, type the following: `./make_bin.sh <enter>`



```
c:\ /cygdrive/c/Work/tutorials/Xilinx/3S500E_Starter/Xil3S500E_RevD_Serial_Flash_v92
EDK Shell
Xilinx EDK 9.2 Build EDK_Jm.16
Copyright (c) 1995-2007 Xilinx, Inc. All rights reserved.

Analyzing Cygwin versions...
Xilinx EDK detected Cygwin installation v1.5.17(0.129/4/2) on your machine.
This Cygwin (c:\\EDK91\\cygwin\\bin) will be used to run Xilinx EDK tools.

Administrator@AUNET-A71E18694 /cygdrive/c/Work/tutorials/Xilinx/3S500E_Starter/X
il3S500E_RevD_Serial_Flash_v92
$ ./make_bin.sh
*****
* Creating BIN from ./spi_user_app/executable.elf
*****

*****
* Binary finished -- type 'exit' to close this window now
*****

Administrator@AUNET-A71E18694 /cygdrive/c/Work/tutorials/Xilinx/3S500E_Starter/X
il3S500E_RevD_Serial_Flash_v92
$
```

Figure 21 – Creating Binary of the SPI User Application

7. Type `'exit'` to close the command shell.

The file `spi_user_app.b` should be approximately 9 KB. If the options to exclude the initialization vectors are not included, this file will be over 500 MB, and the rest of this experiment will fail.

8. Browse to the `FLASH_BURN` directory. Make sure that `spi_user_app.b` is 9KB in size.

Now the bitstream with the hardware platform and bootloader application is created.

9. Select **Tools** → **Update Bitstream** to compile the **spi_bootloader** project and create a new `implementation/download.bit` bitstream with the bootloader application initialized into BRAM.

For this experiment, command-line utilities are used to program this information into the serial flash. The bitstream as well as the **spi_user_app** application binary must be stored in the flash.

10. Browse to the `Xil3S500E_Serial_Flash_v92\FLASH_BURN` directory

The script used in this experiment is called `Xil3SE_bootload.bat`. The user sets up four variables: `bitstream`, `application`, `spiPartName`, and `spi_offset`. These variables

are already set correctly for this project. This script accomplishes several things, including:

- Converts the bitstream to the MCS format
- Converts the user application binary to the MCS format
- Combines the bitstream MCS and the user application MCS into a single MCS.
- Erases, program, and verify the flash

11. In Explorer, right-click on `Xil3SE_bootload.bat` and select **Edit**.

Note that the **spi_user_app** will be stored at address 0x60000 in the serial flash. This is the beginning of Sector #6, which matches the location from which the bootloader application will read (see `BOOT_SECTOR` constant in the `spi_bootloader` project's `bootload.c` file).

12. Unplug the USB from the board.
13. Plug either the PC4 or USB JTAG into the PC and the flyleads into the SPI port (J12).
14. Make sure the MODE (J32) jumpers are in Boundary Scan Mode (M1 installed).
15. Add a jumper to JP8.
16. Turn power ON.
17. Double-click the `Xil3SE_bootload.bat` script. The files are converted and programmed into the flash. In the command window, make sure the device verification was successful:

'1': Verification completed.

18. Press any key to close the command window.
19. Turn off power to the board.
20. Remove the flyleads.
21. If closed, launch HyperTerminal (`com1_115200_8n1n.ht`).
22. Change the MODE (J32) to SPI (M1 and M2 installed, M0 uninstalled).
23. Turn board power on. The DONE light should go on, and you will see HyperTerminal display as shown in Figure 22.

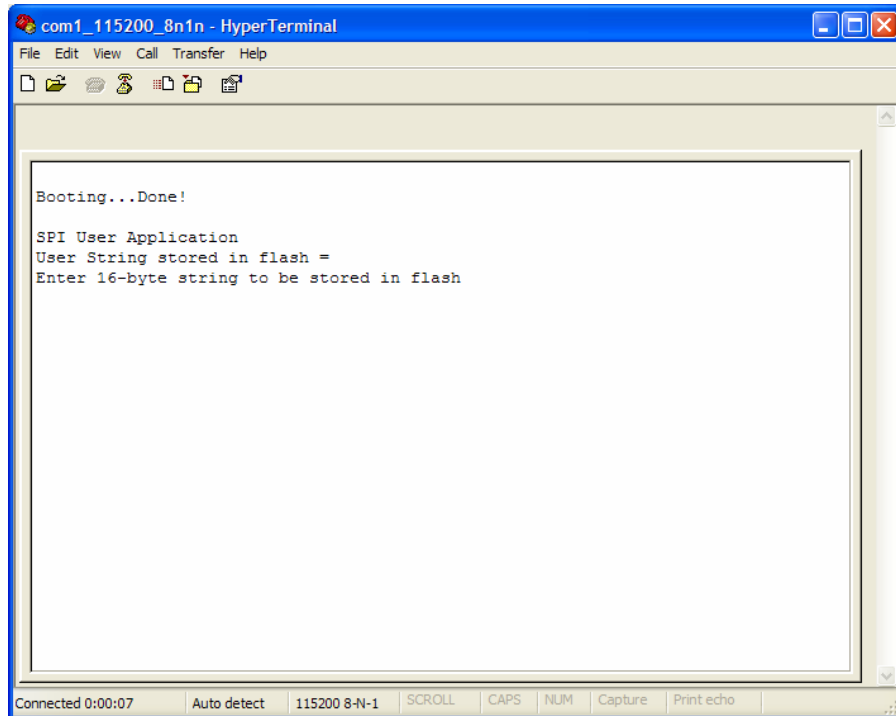


Figure 22 – Configured and Bootloaded from SPI Flash

24. Enter exactly 16 characters. For example, “0123456789abcdef” is exactly 16 characters. The data is then written in the SPI flash.

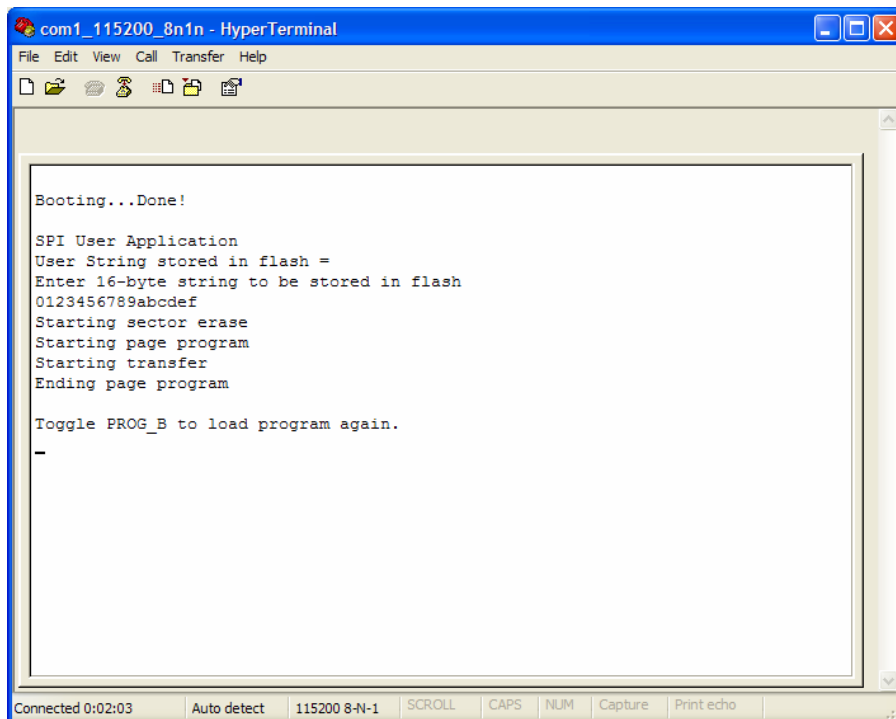
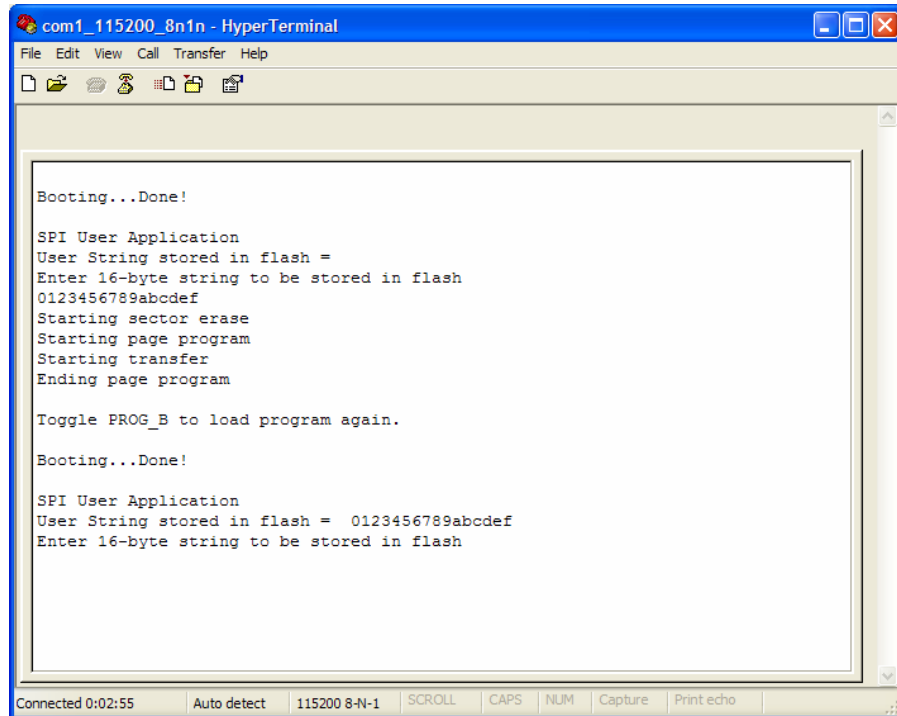


Figure 23 – Initial User String Written to Flash

25. Press the PROGRAM button. The FPGA will again configure from SPI flash. This time, however, the user application finds the string previously typed (see Figure 24).



```
com1_115200_8n1n - HyperTerminal
File Edit View Call Transfer Help

Booting...Done!

SPI User Application
User String stored in flash =
Enter 16-byte string to be stored in flash
0123456789abcdef
Starting sector erase
Starting page program
Starting transfer
Ending page program

Toggle PROG_B to load program again.

Booting...Done!

SPI User Application
User String stored in flash = 0123456789abcdef
Enter 16-byte string to be stored in flash

Connected 0:02:55 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Figure 24 – User String Retrieved from SPI Flash

26. Turn the board power off.

Revision History

Date	Version	Revision
02/23/06	8.1	Initial release.
07/24/07	9.1	Update to EDK 9.1.02. Convert over to iMPACT Direct SPI mode.
10/18/07	9.2	Update to EDK 9.2