XAPP974 (v1.1.3) March 24, 2009

Indirect Programming of SPI Serial Flash PROMs with Spartan-3A FPGAs

Author: Jameel Hussein

Summary

This document describes the hardware setup, file generation flow, and software flow for programming an Numonyx M25Pxx SPI configuration PROM through the JTAG interface of a Spartan®-3A FPGA using iMPACT 9.1.01i. The software flows, including PROM file generation, are also covered.

Introduction

Similar to the traditional configuration memories, SPI serial flash memories must be loaded with the configuration data. SPI serial flash memories have a single interface for programming, but there are multiple methods to deliver the data to this interface. Four primary delivery methods exist to program an SPI serial flash through the SPI interface:

- Indirect in-system programming or ISP (iMPACT, JTAG tool vendor or custom solution)
- Direct in-system programming (SPI direct interface connect)
- Third-party programmers (off-board programming)
- Embedded processor (in-system programming)

Production programming is often accomplished via a third-party programmer or JTAG tool, and many distributors offer mass production gang programming. For prototyping, the iMPACT software, included in the ISE® development software tools, with a Xilinx® parallel cable or Platform Cable USB can program select SPI serial flash memories directly (refer to [Ref 4] for more details).

Unlike the Xilinx Platform Flash PROMs, which are in-system programmable through a standard JTAG interface, SPI flash devices require extra logic for SPI indirect in-system programming via JTAG with Xilinx software and cables. The extra logic is represented as a core residing inside the FPGA, which the iMPACT software uses as a bridge between the FPGA JTAG interface and the SPI PROM's SPI interface.

The following sections discuss the hardware connections required for the indirect in-system programming of SPI serial flash for prototype designs. The Xilinx software tool flows to generate an SPI formatted file and for programming select SPI serial flash memories is also covered.

SPI Basics

SPI serial flash memories use the Serial Peripheral Interface (SPI), a four-wire, synchronous serial data bus. This serial data link was pioneered as a serial communication interface between a microcontroller and its peripherals and is a popular interface in embedded and consumer markets. This interface can now also be used to configure Xilinx FPGAs.

An SPI system typically consists of a master device and a slave device (Figure 1, page 2). When using this four-signal interface to configure a Xilinx FPGA from an SPI serial flash, the FPGA is the master device and the SPI serial flash is the slave device.

The master FPGA device controls the timing via the SCK clock signal. Data is clocked out of the FPGA master and into the SPI serial flash slave on the MOSI signal after the select signal $\overline{SS}$ goes Low. During the same clock cycle, data is clocked out of the SPI serial flash slave and into the FPGA master using the MISO signal. Data is clocked out of each device on one edge and clocked into each device on the next opposite edge in the period.

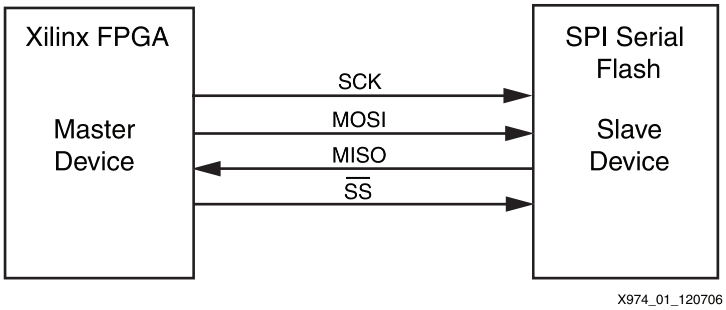


Figure 1: **Basic Block Diagram for SPI Configuration Mode**

In addition to the four-signal interface, each SPI serial flash vendor has unique control signals, such as write protect or hold, that need to be controlled appropriately during programming and configuration (refer to the appropriate vendor's SPI serial flash memory data sheet for additional details on the specific control signals).

A cross reference for the FPGA to SPI interface connections is provided in Table 1.

Table 1: **SPI Serial Flash Interface Connections and Pin Naming**

SPI Signals	SPI Serial Flash Pins[1]	FPGA Connection (Spartan-3A FPGA)	Signal Description
General SPI Signals			
MOSI	D	MOSI	Master Out Slave In is used by the master to specify the instruction to execute or to send data to the slave device.
MISO	Q	DIN	Master In Slave Out is used by the master to collect data transferred from the slave device.
$\overline{SS}$	S	CSO_B	Slave Select, active-Low signal; when driven High this signal is used to deselect the slave device and put MISO at high impedance.
SCK	C	CCLK	Serial Clock provides the timing for the serial interface.
Additional Vendor-Specific SPI Control Signals			
Write Protect	$\overline{W}$	Not required for FPGA configuration, but must be High to program or erase SPI serial flash. Optional connection to FPGA user I/O.	Write Protect protects select areas of memory against program or erase instructions.
Hold	$\overline{HOLD}$	Not required for FPGA configuration, but must be High during FPGA configuration and SPI erase or program. Optional connection to FPGA user I/O.	Hold is used to pause any serial communications with the device without deselecting the device.

Notes:

1. Numonyx M25Pxx SPI serial flash pin names are listed in this table with the most common vendor pin names. The subset of SPI control signals used by each vendor can vary. Refer to the vendor data sheet for specific pin information and descriptions.

Indirect ISP of SPI PROMs

The FPGA has JTAG test capabilities, which include the standard PRELOAD and EXTEST commands. When using these commands, it is possible to drive and sample the pins of the FPGA with the JTAG chain and thereby stimulate the pins of the SPI memory via the traces routed on the PCB. This method is supported by many third-party JTAG tool vendors, but it is often much slower than other programming methods. By using a core running in the FPGA that understands the JTAG protocol and converts the applied data to SPI bus relationships, the SPI memory can be programmed through the JTAG port of the FPGA.

During indirect SPI programming, the FPGA is reconfigured, and consequently, any running design in the FPGA is erased (refer to “Expectations,” page 17 for more details).

Requirements for iMPACT Indirect Programming Support

The following are required to successfully perform in-system programming on the attached SPI serial Flash PROM.

- A Xilinx programming cable:

  (http://www.xilinx.com/onlinestore/hardware/cable_compare.htm)

  Platform Cable USB

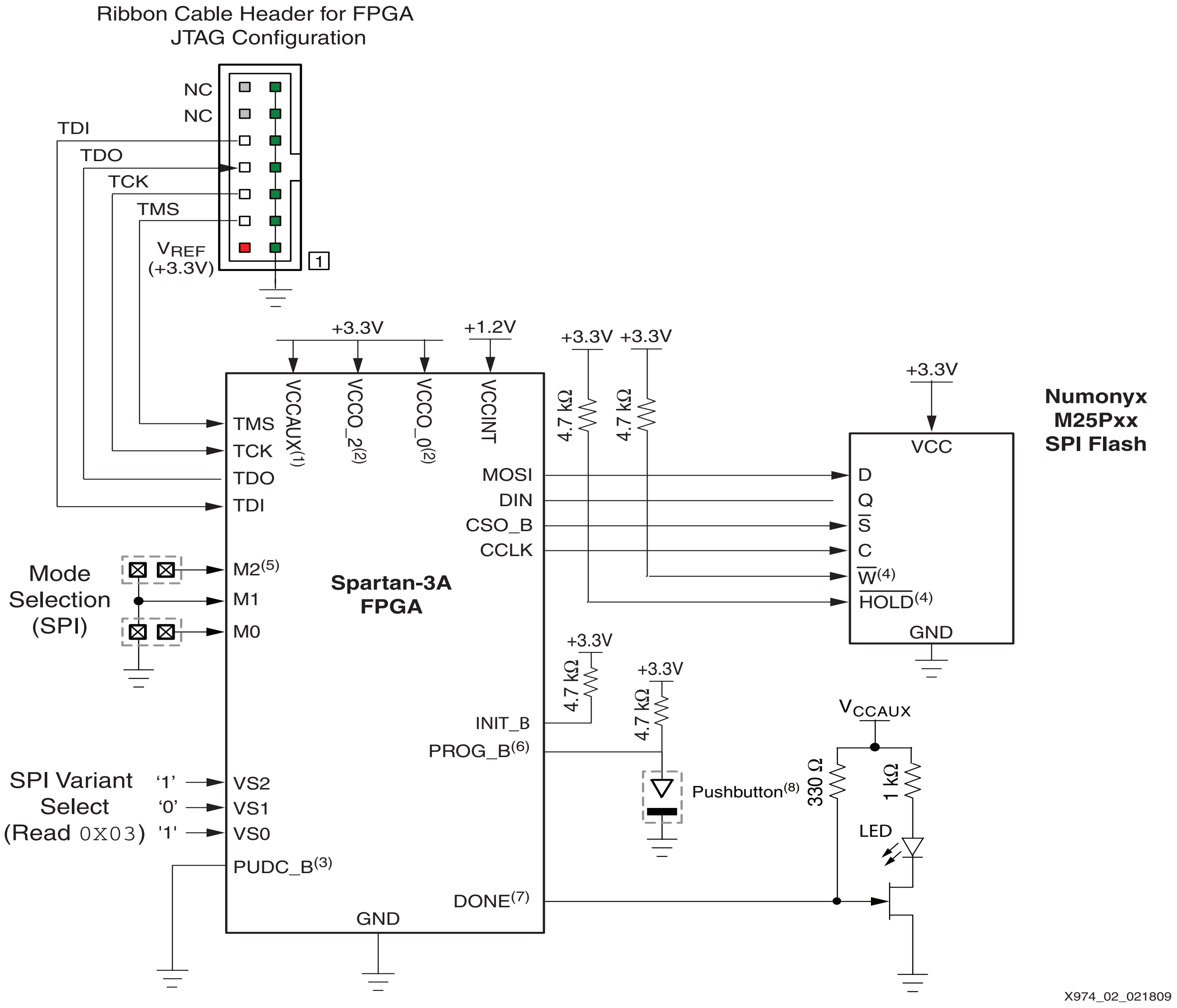
  (http://www.xilinx.com/support/documentation/data_sheets/ds593.pdf)

  Parallel Cable IV

  (http://www.xilinx.com/support/documentation/data_sheets/ds097.pdf)
- A compatible cable connector on-board
- Properly installed Xilinx ISE 9.1.01i software (or later)

Hardware and Connections for SPI Programming

A detailed SPI configuration setup is shown in Figure 2, page 4, where the Spartan-3A FPGA is the master and the Numonyx SPI serial flash is the slave. Power-up sequencing is especially important when configuring from SPI PROMs and care must be take to ensure successful configuration and programming. Refer to the "Master SPI Mode" section in [Ref 3] for more details. The configuration connections from the SPI serial flash to the FPGA are highlighted in this diagram. For information on the programming and configuration headers used by the Xilinx cables, refer to [Ref 1].



Notes:

1. VCCAUX supplies the dedicated Spartan-3A FPGA configuration pins: TMS, TDI, TDO, PROG_B, and DONE.
2. VCCO_2 supplies the voltage to the Spartan-3A FPGA configuration, dual-purpose pins: M[2:0], VS[2:0], INIT_B, CCLK, CSO_B, DIN, MOSI; and VCCO_0 supplies the dual-purpose pin: PUDC_B.
3. PUDC_B can be driven low to enable pull-ups on I/O. Refer to Table 2, page 5.
4. Control signals should be driven appropriately when programming the SPI serial flash. Signals such as the $\overline{\text{W}}$ and $\overline{\text{HOLD}}$ signals should be held High or inactive while programming the SPI serial flash. Refer to the vendor's data sheet for more details.
5. The jumper on the M[2] pin is an enhancement for prototyping and debugging, to allow changing from SPI Mode M[2:0] = `001` to JTAG mode M[2:0] = `101`.
6. Some board-level, power-on sequences can require an additional configuration delay method to ensure the SPI flash is ready to output the bitstream when the FPGA begins its power-on configuration sequence. See "Power-On Precautions if System 3.3V Supply is Last" in Sequence section in [Ref 3] for details and solutions.
7. The LED on DONE is an enhancement for prototyping and debugging.
8. Pushbutton on PROG_B is an enhancement for prototyping and debugging.

Figure 2: **Spartan-3A FPGA Configuration from Numonyx SPI Serial Flash Connection Diagram (Example for the Read Command `0x03`)**

Table 2: **FPGA SPI Configuration Signal Names and Descriptions**

Spartan-3A FPGA Pin Name	FPGA Direction During Configuration	Description	During Configuration	After Configuration
PROG_B	Input	**Program FPGA.** Active Low. When asserted Low, forces the FPGA to restart its configuration process by clearing configuration memory and by resetting the DONE and INIT_B pins. Requires external 4.7 kΩ pull-up resistor to 3.3V. If driving externally, use an open-drain or open-collector driver.	Must be High to allow configuration to start.	Dedicated. Drive PROG_B Low and release to reprogram FPGA.
INIT_B	Open-drain bidirectional I/O	**Initialization Indicator.** Active Low. Goes Low at start of configuration during initialization memory clearing process. Released at end of memory clearing, when mode and variant select pins are sampled. This signal requires an external 4.7 kΩ pull-up resistor to V_{CCO_2}.	Active during configuration. If SPI serial flash PROM requires > 2 ms to awake after powering on, hold INIT_B Low until PROM is ready. If a CRC error is detected during configuration, FPGA drives INIT_B Low.	User I/O. If unused in the application, drive INIT_B High.
M[2:0]	Input	**Mode Select.** Selects the FPGA configuration mode.	SPI mode M2=0, M1=0, M0=1. For JTAG mode for programming SPI Flash PROMS, set M[2:0] = `101`. Sampled when INIT_B goes High. M[2:0] has internal pull-ups if not connected.	User I/O.
VS[2:0]	Input	**Variant Select.** Instructs the FPGA how to communicate with the attached SPI serial flash PROM.	Valid setting options are shown in Figure 1, page 2. Must be at a valid setting when sampled as INIT_B goes High.	User I/O.
CSO_B	Output	**Chip Select Output.** Active Low.	Connects to the SPI serial flash PROM's Slave-Select input.	Drive CSO_B High after configuration to disable the SPI serial flash and reclaim the MOSI, DIN, and CCLK pins as user I/O. Optionally, reuse this pin and MOSI, DIN, and CCLK to continue communicating with SPI serial flash.
CCLK	Output	**Configuration Clock.** Generated by FPGA internal oscillator. Frequency controlled by **ConfigRate** bitstream generator option. If CCLK PCB trace is long or has multiple connections, terminate this output to maintain signal integrity.	Drives PROM's clock input.	User I/O.

Table 2: **FPGA SPI Configuration Signal Names and Descriptions *(Cont'd)***

Spartan-3A FPGA Pin Name	FPGA Direction During Configuration	Description	During Configuration	After Configuration
MOSI	Output	**Master Out Slave In.** Used by the master to specify the instruction to execute or to send data to the slave device.	FPGA sends SPI serial flash read commands and starting address to the PROM's serial data input.	User I/O.
DIN	Input	**Master In Slave Out.** Used by the master to collect data transferred from the slave device.	FPGA receives serial data from PROM's serial data output.	User I/O.
DONE	Open-drain bidirectional I/O	**FPGA Configuration Done.** Low during configuration. Goes High when FPGA successfully completes configuration. Requires external 330Ω pull-up resistor to 3.3V.	Low indicates that the FPGA is not yet configured.	Pulled High via external pull-up. When High, indicates that the FPGA successfully configured.
DOUT	Output	**Serial Data Output.**	Actively drives. Not used in single-FPGA configuration. In a daisy-chain configuration, this pin connects to DIN/D_IN input of the next FPGA in the chain. The downstream FPGA is now in Slave Serial mode.	User I/O.
PUDC_B	Input	**User I/O Pull-Up Control.** When Low during configuration, enables pull-up resistors in all I/O pins to respective I/O bank V_{CCO} input: 0: Pull-ups during configuration. 1: No pull-ups.	Drive at valid logic level throughout configuration.	User I/O.

Software Flows for SPI File Preparation and Programming

Preparing an SPI PROM File

This section provides guidelines and the software flow to create PROM files for an SPI PROM. Before converting an FPGA bitstream into an SPI-formatted PROM file, the designer must verify the bitstream was generated with the **`bitgen -g StartupClk:Cclk`** option. This option ensures proper FPGA functionality by synchronizing the startup sequence to the internal FPGA clock.

The ISE software tools, PROMGen or iMPACT, generate SPI-formatted PROM files from the FPGA bitstream. The SPI PROM serially outputs data bytes MSB first, while Xilinx PROMs output data LSB first. An SPI-formatted PROM file is bit-reversed within each byte from a standard Xilinx PROM file.

Preparing an SPI PROM File Using the ISE PROMGen Command-Line Software

The ISE PROMGen software takes an FPGA bitstream (`.bit`) file as input and, with the appropriate options, generates a memory image file for the data array of an SPI PROM. The output memory image file format is chosen through a PROMGen software command-line option. Typical file formats include Intel Hex (`.mcs`) and Motorola Hex (`.exo`).

The ISE PROMGen software utility is easily executed from a command-line (see Table 3 for ISE PROMGen software options used for SPI PROM file generation). An example PROMGen software command-line to generate an mcs-formatted file for a 64-Mb (8192-kB) SPI PROM is:

```
promgen -spi -p mcs -o spi_prom.mcs -s 8192 -u 0 bitfile.bit
```

The **`-spi`** option is required to ensure proper bit ordering within the SPI PROM file. The **`-p mcs`** option specifies Intel Hex (`.mcs`) output file format. The **`-o spi_prom.mcs`** specifies output to the `spi_prom.mcs` file. The **`-s 8192`** specifies a PROM file image size of 8192 kB. The **`-u 0`** option specifies the data to start at address zero and fill the data array in the up direction. The `bitfile.bit` file is the input bitstream file.

Table 3 list the various PROMGen options and the functions.

Table 3: **Example PROMGen SPI PROM File Options**

PROMGen Option	Description
`-spi`	Used to maintain the correct bit ordering required to configure the FPGA from an SPI serial flash device.
`-p <format>`	PROM output file format. Commonly accepted PROM file formats include Intel Hex (`.mcs`) and Motorola Hex (`.exo`).
`-s <size>`	Specifies the PROM size in kilobytes. The PROM size must be a power of two for this option, and the default setting is 64 kilobytes.
`-u <address>`	Loads the `.bit` file from the specified starting address in an upward direction. This option must be specified immediately before the input bitstream file.

Preparing an SPI PROM File Using the ISE iMPACT Graphical Software

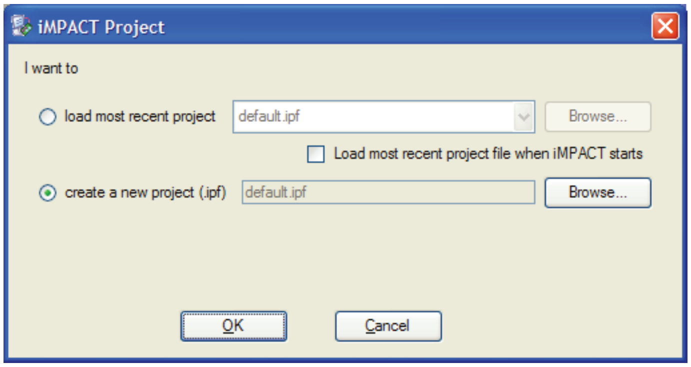
The ISE iMPACT 9.1.01i (or later) software integrates PROM file formatting and in-system programming features behind an intuitive graphical user interface (GUI). The PROMGen file formatting functionality is provided through a step-by-step wizard in the iMPACT software. The wizard steps through the output PROM file options and input bitstream selections. A final step is required for iMPACT to generate the PROM file.

In the iMPACT software, an SPI PROM file can be generated from a Xilinx FPGA bitstream through a simple eight step process. A prescribed sequence of dialog boxes (also known as a wizard) acts as a guide through most of the PROM file generation process.

The following section demonstrates the iMPACT software process for generating an SPI-formatted PROM file in the MCS file format for a 64 Mb SPI PROM. The demonstrated process takes the `bitfile.bit` FPGA bitstream file as input and generates a PROM file named `spi_prom.mcs`.

Step 1: Create a New Project for PROM File Generation

After launching the iMPACT software, the iMPACT project dialog box is displayed (Figure 3). Choose the "create a new project (`.ipf`)" option. Optionally, specify a project location via the **Browse…** button. Then, click **OK** to continue to step 2 in the process.



iMPACT Project

I want to

○ load most recent project   default.ipf

□ Load most recent project file when iMPACT starts

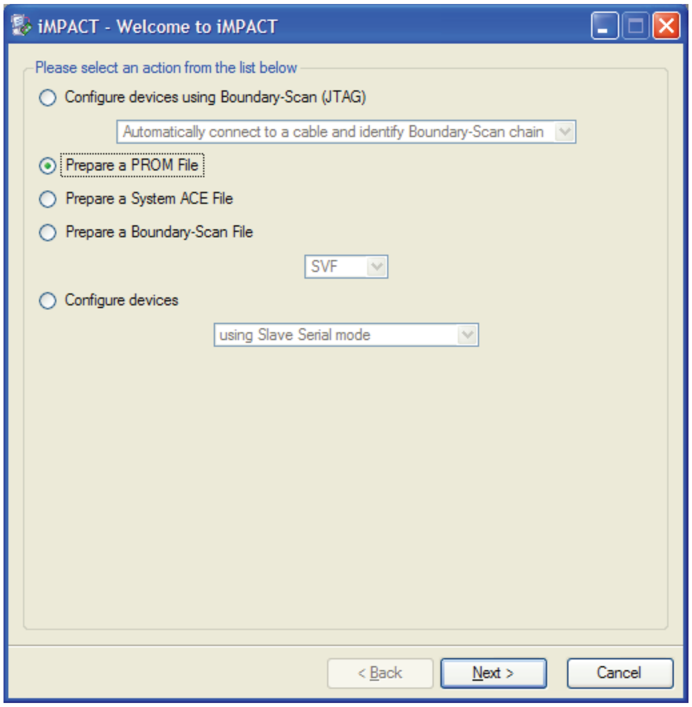
● create a new project (.ipf)   default.ipf

X974_03_032207

Figure 3: **Create a New Project for PROM File Generation**

Step 2: Choose to Prepare a PROM File

The first dialog box of the wizard displays the available kinds of projects that can be created (Figure 4). Check "Prepare a PROM File," and select **Next** to proceed to step 3 of the process.



iMPACT - Welcome to iMPACT

Please select an action from the list below

○ Configure devices using Boundary-Scan (JTAG)

Automatically connect to a cable and identify Boundary-Scan chain

● Prepare a PROM File

○ Prepare a System ACE File

○ Prepare a Boundary-Scan File

SVF

○ Configure devices

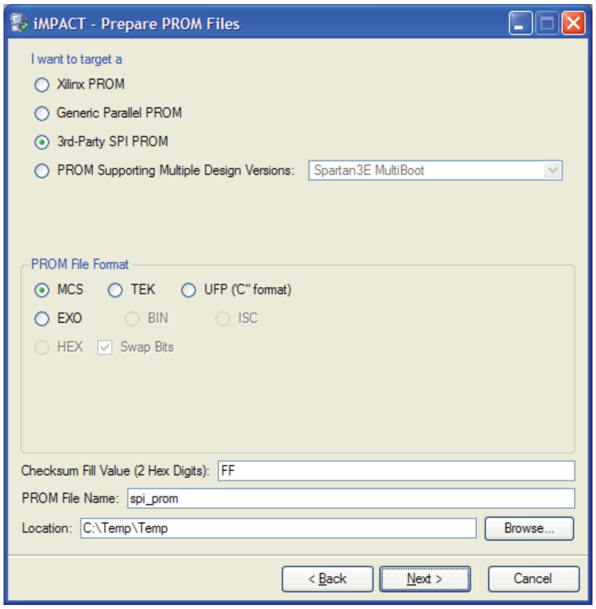
using Slave Serial mode

X974_04_032207

Figure 4: **Choose to Prepare a PROM File**

Step 3: Specify the Output SPI PROM File Options

The third step of the process is to specify the targeted PROM type, the PROM file format, and output file name and location (Figure 5). Choose to target the "3rd-Party SPI PROM" type. Select the "MCS" PROM file format. Maintain the default "Checksum Fill Value," which is a hexadecimal `FF` byte value. Specify the PROM file name to be `spi_prom` (to the spi_prom name, iMPACT automatically adds the `.mcs` file name extension corresponding to the chosen MCS PROM file format). Specify a desired directory location for the output `spi_prom.mcs` file. Click **Next** to continue to step 4.



iMPACT - Prepare PROM Files

I want to target a

○ Xilinx PROM
○ Generic Parallel PROM
● 3rd-Party SPI PROM
○ PROM Supporting Multiple Design Versions: Spartan3E MultiBoot

PROM File Format

● MCS ○ TEK ○ UFP ("C" format)
○ EXO ○ BIN ○ ISC
○ HEX ☑ Swap Bits

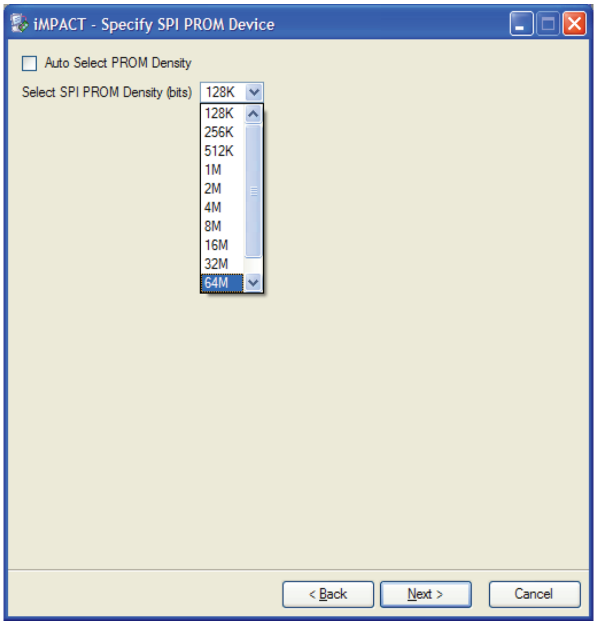
Checksum Fill Value (2 Hex Digits): FF
PROM File Name: spi_prom
Location: C:\Temp\Temp

X974_05_032207

Figure 5: **Specify the Output SPI PROM File Options**

Step 4: Specify an SPI PROM Density

The fourth step of the process is to specify the size of the target SPI PROM (Figure 6). From the “Select SPI PROM Density” drop-down list, choose the 64M value that matches the 64-Mb size of the targeted SPI PROM in this demonstration. Click **Next** to proceed to step 5 of the process.



iMPACT - Specify SPI PROM Device

□ Auto Select PROM Density

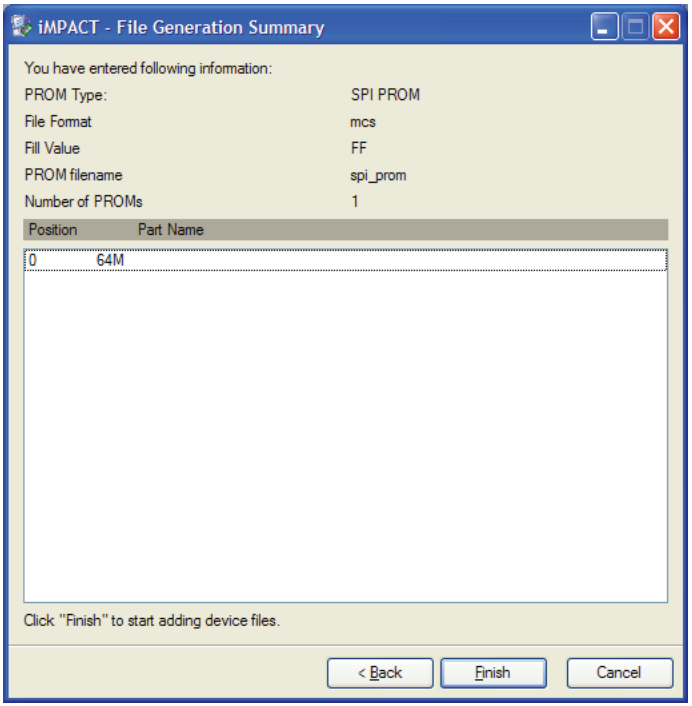
Select SPI PROM Density (bits) 128K

X974_06_032207

Figure 6: **Specify an SPI PROM Density**

Step 5: Summary of SPI PROM File Selections

The fifth step displays a summary of the options selected from the prior steps in the process (Figure 7). The summary shows that a PROM file in the `.mcs` file format with a fill value of hexadecimal `FF` is to be written to a file with a root name of `spi_prom` for a 64-Mb SPI PROM. Click **Finish** to complete the wizard and proceed to step 6 of the process.



You have entered following information:

PROM Type:	SPI PROM
File Format	mcs
Fill Value	FF
PROM filename	spi_prom
Number of PROMs	1

Position	Part Name
0	64M

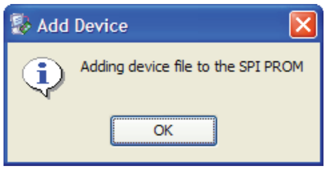
Click "Finish" to start adding device files.

X974_07_032207

Figure 7: **Summary of SPI PROM File Selections**

Step 6: Automated Notification to Add a Device File to the SPI PROM File

After the iMPACT project wizard is finished, the iMPACT SPI PROM generation project is set to generate a specific PROM file with specific parameters. At this stage, the PROM file memory image is empty. The sixth step is to add an FPGA bitstream to the PROM file memory image. This step begins immediately after completion of the iMPACT project wizard with an automatic notification that the next step is to add a device file to the SPI PROM memory image. Click **OK** in the Add Device notification dialog box (Figure 8) to proceed to step 7.



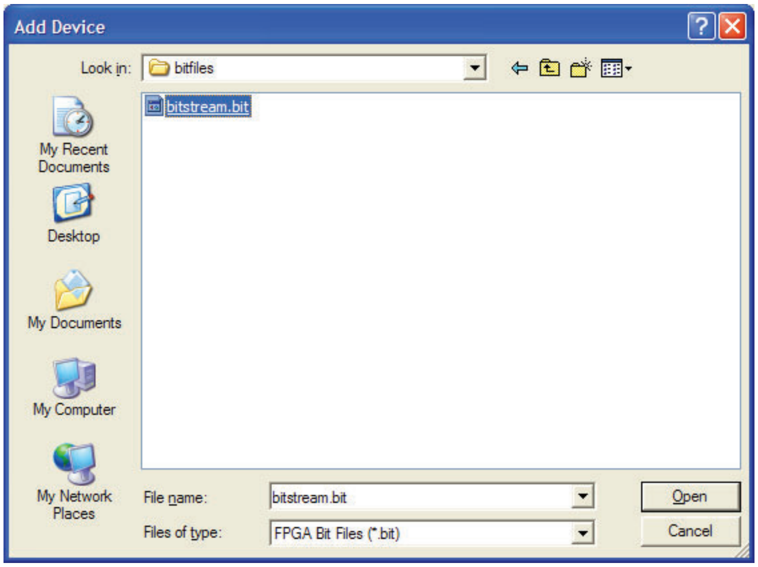
Adding device file to the SPI PROM

X974_08_032207

Figure 8: **Add Device Notification Dialog Box**

Step 7: Select the FPGA Bitstream File to Add to the SPI PROM Memory Image

After the Add Device notification, iMPACT automatically opens a file browser to select the FPGA bitstream (`.bit`) file to add to the SPI PROM memory image (Figure 9). Select the FPGA bitstream file to be written to the SPI PROM. Click **Open** in the browser to add the selected FPGA bitstream to the SPI PROM memory image. This action completes the automated iMPACT process for preparing an SPI PROM file to be generated. Proceed to step 8 to generate the SPI PROM file.



Look in: bitfiles

File name: bitstream.bit

Files of type: FPGA Bit Files (*.bit)

X974_09_032207

Figure 9: **Add Device File Browser**

Step 8: iMPACT Generate File Operation

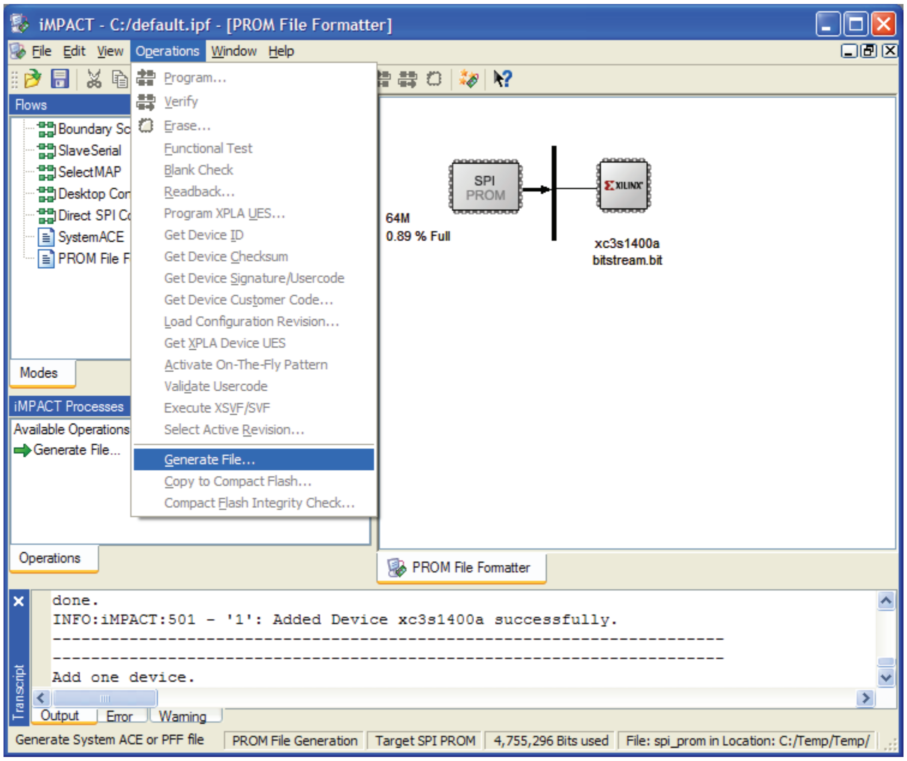
The eighth and final step is to generate the PROM file. Under the iMPACT Operations menu, invoke the **Generate File** menu item (Figure 10, page 12). Once invoked, the Generate File menu item causes iMPACT to generate the specified SPI PROM file.

iMPACT reports a "PROM File Generation Succeeded" message after successful generation of the SPI PROM file.

After the Generate File operation has completed, the generated `spi_prom.mcs` file is available in the specified location. The `spi_prom.mcs` file can be used in any of the supported programming solutions to program the SPI PROM with the specified FPGA bitstream contained within the SPI PROM file.

Save the iMPACT SPI PROM generation project for quick regenerating of the SPI PROM file whenever the FPGA bitstream design is revised. To regenerate an SPI PROM file, reopen the saved iMPACT project, and invoke the Generate File operation. iMPACT generates a revised SPI PROM file from the new version of the FPGA bitstream file, assuming the revised bitstream file is located in the same location as the original bitstream file.

If a project is not loaded when using the iMPACT GUI interface, a user is guided through the wizard steps each time to create a new SPI-formatted PROM file. The designer is prompted to name the project and select the option "Prepare a PROM File," following "Step 1: Create a New Project for PROM File Generation" through "Step 8: iMPACT Generate File Operation" to generate a new SPI file.



```
done.
INFO:iMPACT:501 - '1': Added Device xc3s1400a successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
Add one device.
```

X974_10_032207

Figure 10: **Generate File Menu**

Using the ISE iMPACT Software to In-System Program SPI PROMs

In prototype applications, the ISE iMPACT 9.1.01i (or later) software can be used to indirectly in-system program select SPI serial flash devices via JTAG with a memory image from a given SPI PROM file (see "Preparing an SPI PROM File," page 7 for instructions on the generation of an SPI PROM file).

Table 4 lists the selected SPI serial flash memories that can be programmed with iMPACT.

Table 4: **SPI Serial Flash Programming Capability with iMPACT**

SPI Serial Flash Vendor	Family[(1)]
Numonyx	M25P, M25PE, M45PE
Atmel	AT45DB

Notes:

1. Refer to the *iMPACT Software Manual* for SPI support.

The iMPACT software can program select SPI PROMs using a simple eight-step process. A prescribed sequence of dialog boxes (or wizard) acts as a guide through most of the iMPACT programming process.

The following section demonstrates the iMPACT software process for in-system programming a M25P64 (64 Mb) Numonyx SPI PROM. The demonstrated process takes the `spi_prom.mcs` SPI PROM file (generated in "Preparing an SPI PROM File") as input, erases the SPI PROM, programs the PROM file contents into the SPI serial flash device, and verifies the SPI PROM contents against the given SPI PROM file contents.

Step 1: Setup Hardware for In-System Programming

The first step of the programming process is the proper setup of the hardware for in-system programming of the SPI PROM. Check the following:

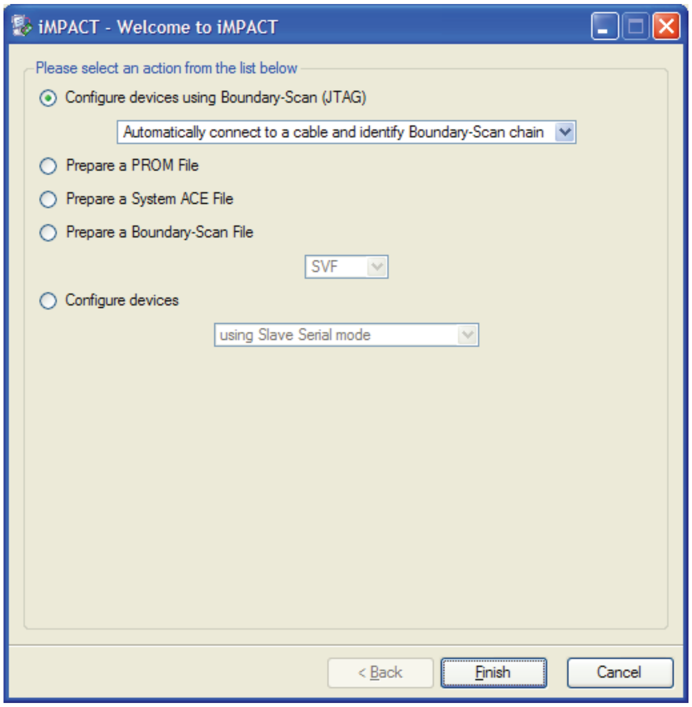
- **Proper Xilinx cable connection:** The Xilinx cable must be properly connected to the computer and to the JTAG bus of the FPGA, which is connected via the configuration SPI port to the SPI PROM (see Figure 2, page 4 for hardware connections from the Xilinx cable to the FPGA JTAG bus and from the FPGA SPI configuration port to the target SPI PROM).
- **Cable power:** If using the Xilinx Parallel Cable IV or Xilinx MultiPRO cable, then power must be applied to the cable.
- **Target system power**: Power must also be supplied to the target system containing the SPI PROM.

Step 2: Create a New Project for Indirect In-System Programming

After launching the iMPACT software, the iMPACT Project dialog box is displayed (Figure 3, page 8). Choose the "create a new project (`.ipf`)" option. Optionally, specify a project location via the **Browse…** button. Then, click **OK** button to continue to step 2 in the process.

Step 3: Configure Devices Using Boundary Scan

The second step begins with the iMPACT project wizard. The first dialog box of the wizard displays the available kinds of projects that can be created (Figure 11). Choose the "Configure Devices using Boundary Scan" option. Then, select the **Automatically Connect to a cable and identify Boundary Scan chain** item from the associated drop-down list box. Click **Finish** to complete the new project setup process. At the completion of this process, iMPACT initializes the JTAG chain and prompt the user for the configuration file for the FPGA.



iMPACT - Welcome to iMPACT

Please select an action from the list below

● Configure devices using Boundary-Scan (JTAG)

Automatically connect to a cable and identify Boundary-Scan chain

○ Prepare a PROM File

○ Prepare a System ACE File

○ Prepare a Boundary-Scan File

SVF

○ Configure devices

using Slave Serial mode

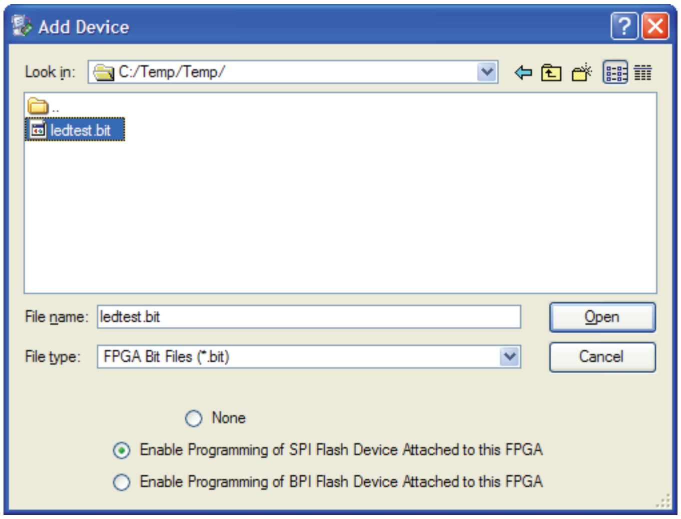
X974_11_032207

Figure 11: **Configure Devices Using the Indirect SPI Configuration Mode**

Step 4: Add a New Configuration File

After finishing the new project wizard, iMPACT automatically leads into the third step of the process. iMPACT automatically displays a file browser window to select an FPGA configuration file (Figure 12). Choose the `ledtest.bit` file, select "Enable Programming of SPI Flash Device Attached to this FPGA," and click **Open**.

Note: The file `ledtest.bit` is a placeholder for a generic bitstream for the FPGA. The user needs to select a bitstream to enable SPI indirect programming; however, the user does not need to configure the FPGA with this bitstream.


Add Device

Look in: C:/Temp/Temp/

ledtest.bit

File name: ledtest.bit

File type: FPGA Bit Files (*.bit)

○ None

● Enable Programming of SPI Flash Device Attached to this FPGA

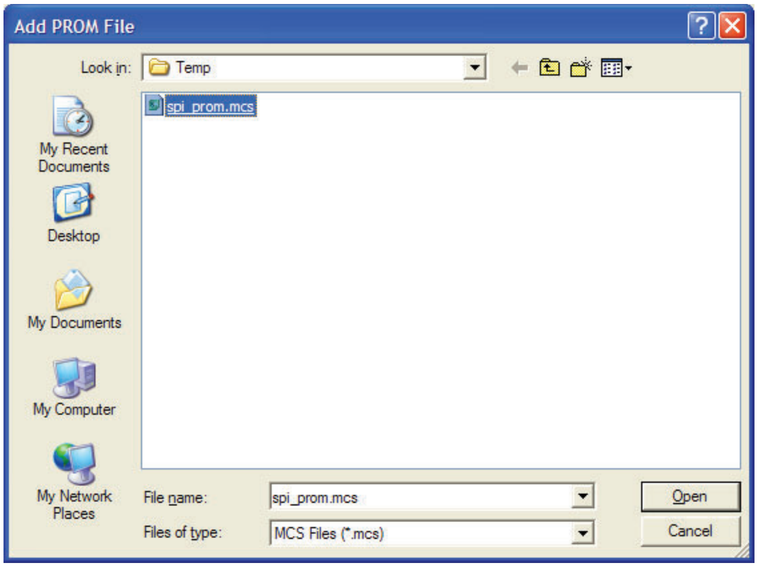
○ Enable Programming of BPI Flash Device Attached to this FPGA

X974_12_032207

Figure 12: **Assign New Configuration File**

Step 5: Add an SPI PROM File

After finishing the new project wizard, iMPACT automatically leads into the third step of the process. iMPACT automatically displays a file browser window to select an SPI PROM file for programming into the SPI serial flash device (Figure 13). Choose the `spi_prom.mcs` file, and click **Open**.


Add PROM File

Look in: Temp

spi_prom.mcs

File name: spi_prom.mcs

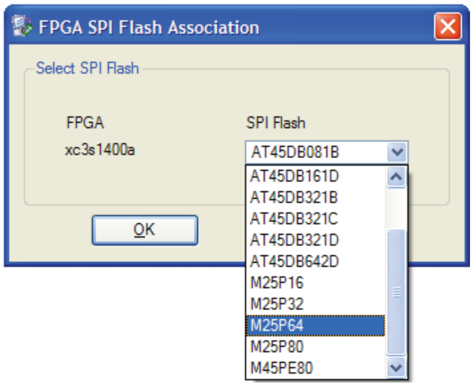
Files of type: MCS Files (*.mcs)

X974_13_032207

Figure 13: **Add an SPI PROM File**

Step 6: Select Numonyx M25P64 Device Part Number

After selecting the SPI PROM file to load, iMPACT displays the FPGA SPI Flash Association dialog box (Figure 14). The fourth step of the process requires the target type of SPI PROM to be specified in this dialog box. Select the Numonyx M25P64 part number for the target SPI PROM type used in this demonstration. Click **OK** to complete the SPI PROM programming setup.



FPGA: xc3s1400a

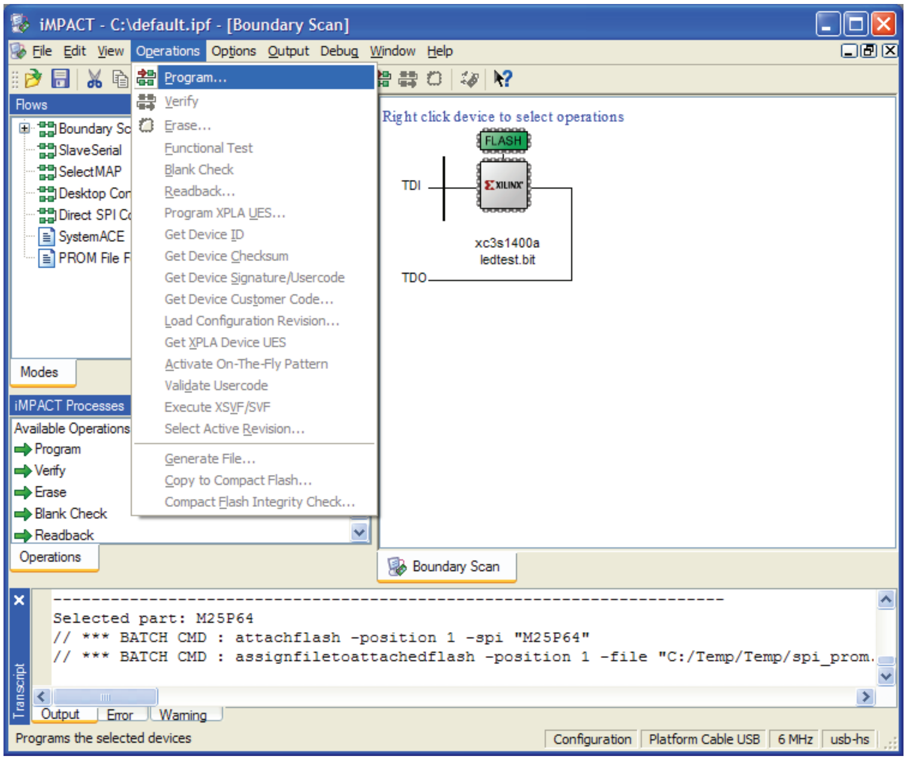
SPI Flash: AT45DB081B

X974_14_032207

Figure 14: **Select Device Part Name Dialog Box**

Step 7: Invoke the iMPACT Program Operation

The seventh step of the process programs the target SPI PROM with the selected SPI PROM file contents. Ensure the SPI PROM icon in the iMPACT window is selected by left-clicking on the SPI PROM icon (the SPI PROM icon is highlighted in green when selected). Select **Operations** → **Program** to begin programming (Figure 15).



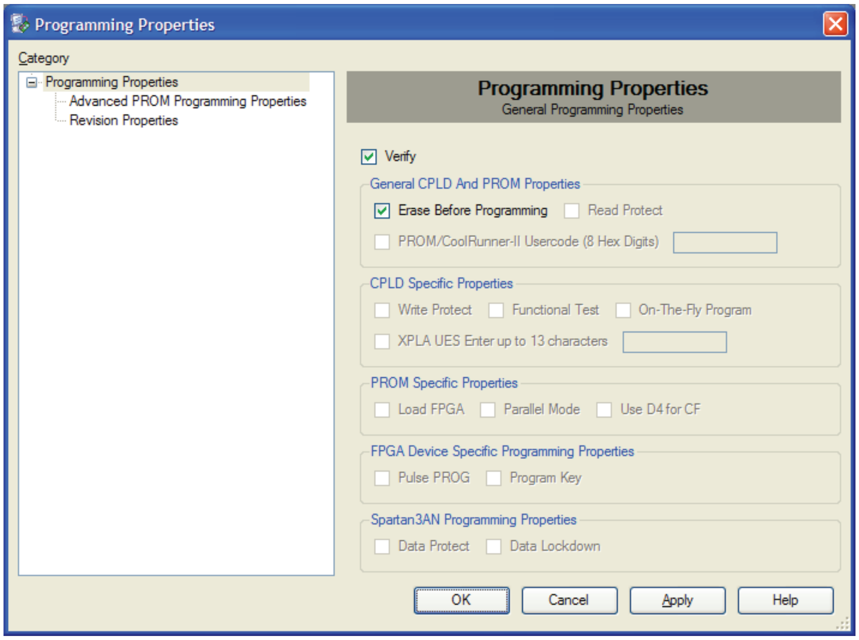
```
---------------------------------------------------------------------------
Selected part: M25P64
// *** BATCH CMD : attachflash -position 1 -spi "M25P64"
// *** BATCH CMD : assignfiletoattachedflash -position 1 -file "C:/Temp/Temp/spi_prom.
```

X974_15_032207

Figure 15: **Program Menu**

Step 8: Select iMPACT Programming Properties

In response to the invocation of the **Program** operation, iMPACT presents the Programming Properties dialog box (Figure 16). The eighth step of the process ensures the selection of proper programming properties. Ensure that both the **Verify** and the **Erase Before Programming** options are checked, ensuring proper programming of the SPI PROM. Click **OK** to begin the erase, program, and verify operations.



Programming Properties

Category

- Programming Properties
  - Advanced PROM Programming Properties
  - Revision Properties

Programming Properties
General Programming Properties

☑ Verify

General CPLD And PROM Properties

☑ Erase Before Programming □ Read Protect

□ PROM/CoolRunner-II Usercode (8 Hex Digits)

CPLD Specific Properties

□ Write Protect □ Functional Test □ On-The-Fly Program

□ XPLA UES Enter up to 13 characters

PROM Specific Properties

□ Load FPGA □ Parallel Mode □ Use D4 for CF

FPGA Device Specific Programming Properties

□ Pulse PROG □ Program Key

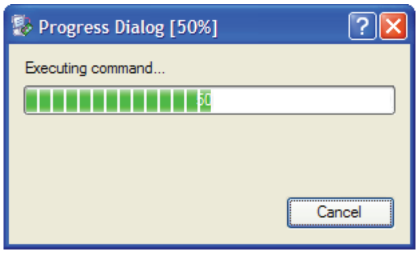
Spartan3AN Programming Properties

□ Data Protect □ Data Lockdown

X974_16_032207

Figure 16: **Programming Properties Dialog Box**

At the start of the programming operation, iMPACT automatically connects to the cable attached to the computer. Then, iMPACT displays a Progress Dialog box as it progresses through the in-system erase, program, and verify operations (Figure 17). Depending on the size of the SPI PROM, size of the SPI PROM file image, and speed of the cable configuration, the programming operation can take anywhere from a few seconds to a few minutes to complete.



Progress Dialog [50%]

Executing command...

X974_17_032207

Figure 17: **Progress Dialog Box**

At the end of a successful Program operation, iMPACT reports a "Program Succeeded" message.

Save the iMPACT project for quickly reprogramming the SPI PROM whenever the SPI PROM file is revised. To reprogram the SPI PROM, reopen the saved iMPACT project, and invoke the **Program** operation, ensure the selection of the **Erase** and **Verify Programming Properties**, and click **OK**. iMPACT reprograms the SPI PROM, assuming the revised SPI PROM file is located in the same location as the original SPI PROM file.

Expectations

Programming Time

The user can expect a programming time of approximately 60 seconds for a 64-Mbit device. Erase on the same device takes approximately 70 seconds. A full erase, program, and verify operation takes approximately 160 seconds. All times can vary slightly due to bitstream size, cable speed and/or system speed.

Impact of Indirect Programming on the Rest of the System

When using the JTAG port to program the SPI serial flash through the FPGA, the user must understand the behavior of the FPGA during this process and how it can affect other devices in the system. To access the SPI device through the JTAG port, a proprietary Xilinx JTAG-to-SPI core must be loaded into the FPGA, requiring the FPGA to be reconfigured and result in any existing logic being lost.

The core is integrated into the software and automatically selected and loaded by iMPACT. All core processes are preformed in the background, transparent to the user. The core is for iMPACT runtime use only — the source code is not available for designs.

During reconfiguration, the I/Os on the FPGA can be controlled through the PUDC_B enable pin. When this pin is Low, it activates internal pull-ups on all the I/Os. In addition, the user should ensure that any other devices on the SPI bus are disabled during in-system programming to reduce contention on the SPI bus. Also, if the FPGA I/O is controlling any other devices, pull-ups or pull-downs should be added to disable these device during programming.

Pull-Ups and Pull-Downs

Once the FPGA has been configured with the JTAG-to-SPI core, the unused I/Os are set to PULLUP, activating the internal pull-up on the I/O. The user can still pull down the I/O using a 1.1 kΩ resistor if dictated by system requirements.

Conclusion

The ability to program SPI devices through JTAG port of a Spartan-3A device with iMPACT can greatly increase the value of using Xilinx FPGAs in a system. While there are some limitations to this programming solution, it is one of the easiest to use in the industry.

References

Device

Xilinx documents

1. DS123, *Platform Flash In-System Programmable Configuration PROMs.*
2. DS529, *The Spartan-3A FPGA Family Data Sheet*.
3. UG332, *Spartan-3 Generation Configuration User Guide.*
4. XAPP951, *Configuring Xilinx FPGAs with SPI Serial Flash.*

Software

The Xilinx PROMGen and iMPACT software are available with the main Xilinx ISE Foundation software or with the downloadable Xilinx ISE WebPACK™ software packages.

ISE Foundation software:

http://www.xilinx.com/ise/logic_design_prod/foundation.htm

ISE WebPACK software:

http://www.xilinx.com/ise/logic_design_prod/webpack.htm

The Xilinx ISE software manuals are available at:

http://www.xilinx.com/support/software_manuals.htm

Hardware

Information regarding the Xilinx cables are found on the Xilinx Configuration Solutions website:

http://www.xilinx.com/products/design_resources/config_sol/

See the ISE iMPACT 9.1.01i (or later) software manuals for supported Xilinx cables.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/16/07	1.0	Initial Xilinx release.
09/28/07	1.1	• Updated document template. • Corrected pull-up resistor value for DONE in Figure 2, page 4. • Made other minor edits and corrections.
11/21/07	1.1.1	• Updated URLs.
01/06/09	1.1.2	• Updated Platform Cable USB reference on page 3.
03/24/09	1.1.3	• Replaced references to STMicroelectronics with Numonyx.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.