XAPP986 (v1.0.2) November 12, 2007

Bulletproof Configuration Guide for Spartan-3A FPGAs

Author: Jameel Hussein

Summary

Platform Flash PROMs give designers a no-compromises memory storage solution supporting the newest generation of Xilinx FPGAs. This application note is a condensed version of the best practices for configuration of Platform Flash PROMs and serves as a guide for a basic configuration setup. The focus of this document is on Spartan™-3A FPGAs.

Platform Flash PROMs

The Platform Flash PROM family provides non-volatile storage, as well as an integrated bitstream delivery mechanism. In addition, this family of PROMs meets the increased bitstream storage requirements of larger Xilinx FPGAs and is compatible with all Xilinx FPGA families. Platform Flash PROMs combine a small footprint, low-cost package with high density, making them an ideal configuration solution for both Spartan and Virtex™ FPGA families.

Introduction to Master Serial Mode

The Platform Flash PROM family supports numerous FPGA configuration modes. This application note highlights one of the most popular configuration mode setups: FPGA Master Serial mode.The Master Serial Mode is popular because of its ease of use and the low-pin-count requirements.

Supported Platform Flash PROMs

Table 1: **Number of Bits to Program a Spartan-3A FPGA and Smallest Platform Flash PROM**

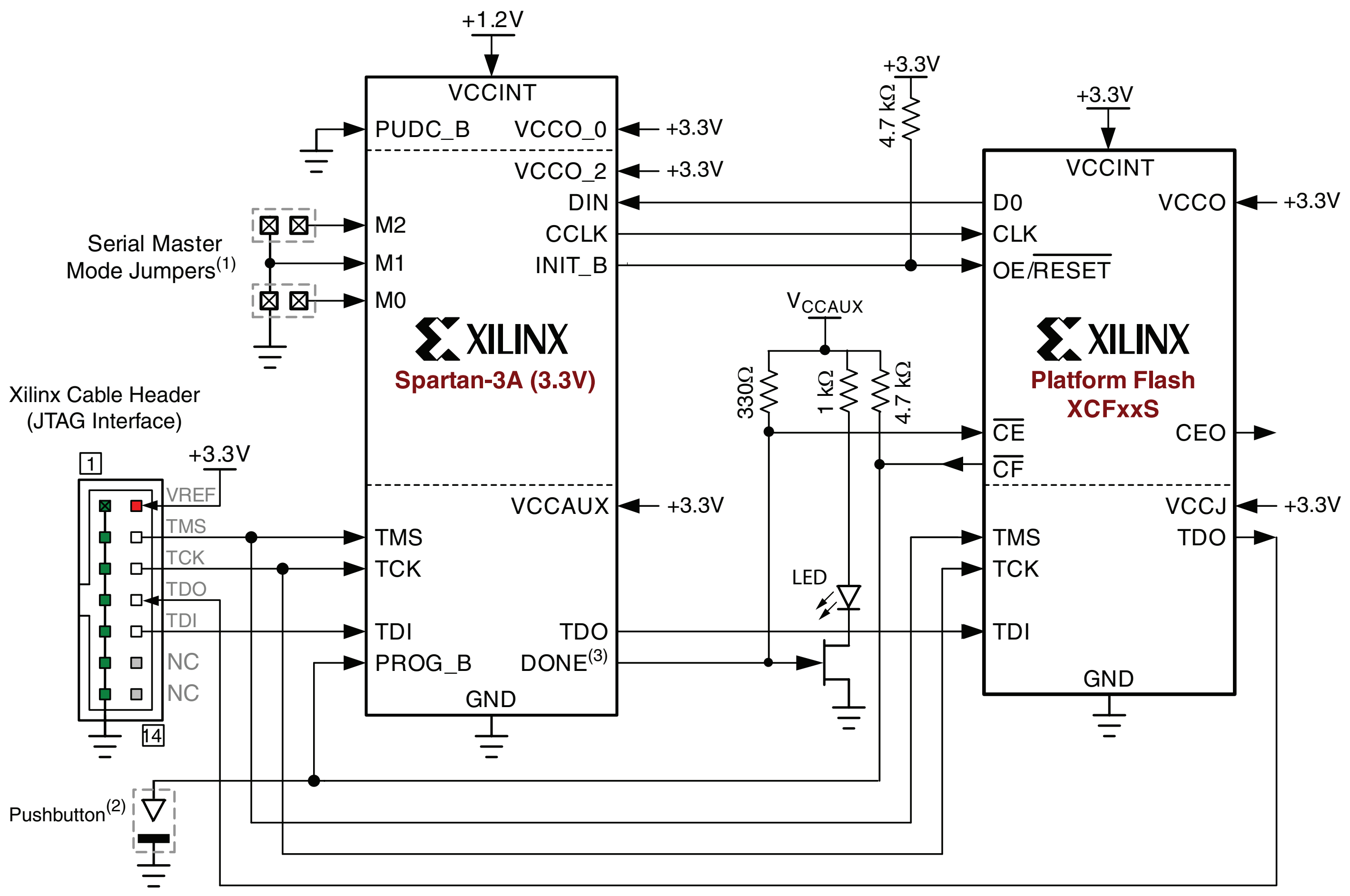
FPGA	Number of Configuration Bits	Smallest Possible Platform Flash PROM(1)
XC3S50A	437,312	XCF01S
XC3S200A	1,196,128	XCF02S
XC3S400A	1,886,560	XCF02S
XC3S700A	2,732,640	XCF04S
XC3S1400A	4,755,296	2 x XCF04S or 1 x XCF08P

Notes:
1. For more details regarding Platform Flash PROMS, refer to [Ref 3].

Hardware Connections for Master Serial Mode

Master Serial configuration mode (Figure 1) requires a small number of signals to interface the PROM with the FPGA. using only five signals in addition to power and ground (Table 2): Chip Enable ($\overline{\text{CE}}$), Output Enable/Reset (OE/$\overline{\text{RESET}}$), Configuration Clock (CCLK), Data (D0), and Configuration Pulse ($\overline{\text{CF}}$). Moreover, an external clock source is not required for configuration.

In FPGA Master Serial mode, the FPGA supplies the CCLK output clock from its internal oscillator to the attached Platform Flash PROM. In this mode, data is available on the PROM Data (D0) pin when $\overline{\text{CF}}$ is High, and $\overline{\text{CE}}$ and OE are enabled. New data is available after each rising clock edge.



Notes:
1. The jumpers on the Mode Pins are enhancements for prototyping and debugging, to allow changing from Master Serial Mode, [000] to JTAG mode [101].
2. Pushbutton on PROG_B is an enhancement for prototyping and debugging.
3. The LED on DONE is an enhancement for prototyping and debugging.
4. Supply and pull-up resistor value for the LED cn vary depending on LED vendor.

X986_01_062207

Figure 1: **Master Serial Mode Using Platform Flash PROM (Spartan-3A FPGA, V_{CCAUX} = 3.3V)**

Table 2: **Spartan-3A Serial Master Mode Connections**

Pin Name	FPGA Direction	Description	During Configuration	After Configuration
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
M[2:0]	Input	**Mode Select.** Selects the FPGA configuration mode.	M2=0, M1=0, M0=0.Sampled when INIT_B goes High. M[2:0] has internal pull-ups if not connected.	User I/O
DIN	Input	**Serial Data Input**.	Receives serial data from PROM's D0 output.	User I/O
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 CLK clock input.	User I/O
DOUT	Output	**Serial Data Output.**	Not used in single-FPGA designs; DOUT is pulled up and not actively driving. In a daisy-chain configuration, this pin connects to DIN input of the next FPGA in the chain.	User I/O
INIT_B	Open-drain bidirectional I/O	**Initialization Indicator.** Active Low. Goes Low at start of configuration during Initialization memory clearing process. Released at and of memory clearing, when mode select pins are sampled. Requires external 4.7 kΩ pull-up resistor to V_{CCO_2}.	Connects to PROM's OE/$\overline{\text{RESET}}$ input. FPGA clears PROM's address counter at start of configuration, enables outputs during configuration. PROM also holds FPGA in Initialization state until PROM reaches Power-On Reset (POR) state. If CRC error detected during configuration, FPGA drives INIT_B Low.	User I/O. If unused in the application, drive INIT_B High 3-stated to avoid contention with the external pull-up.
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 V_{CCAUX} (3.3V).	Connects to PROM's chip enable ($\overline{\text{CE}}$) input. Enables PROM during configuration. Disables PROM after configuration.	Pulled High via external pull-up. When High, indicates that the FPGA successfully configured.
PROG_B	Input	**Program FPGA.** Active Low. Asserted Low for 300 ns or longer, forces the FPGA to restart its configuration process by clearing configuration memory and resetting the DONE and INIT_B pins once PROG_B returns High. Requires external 4.7 kΩ pull-up resistor to V_{CCAUX} (3.3V).	Must be High during configuration to allow configuration to start. Connects to PROM's $\overline{\text{CF}}$ pin, allowing JTAG PROM programming algorithm to reconfigure the FPGA.	Drive PROG_B Low and release to reconfigure the FPGA.

Table 3: **Pin Connections for FPGA Master Serial Mode**

XCFxxS Pins	Connects to FPGA Pin	Comment
D0	DIN	PROM and FPGA Data signal.
CLK	CCLK	FPGA generated clock signal.
$\overline{\text{CE}}$	DONE	See FPGA data sheet for recommended value for DONE pull-up resistor to V_{CCAUX}. Do not directly drive DONE LED status indicator.
OE/$\overline{\text{RESET}}$	INIT_B	4.7 kΩ pull-up resistor to V_{CCAUX} is required.
$\overline{\text{CF}}$	PROG_B	Open drain output. 4.7 kΩ pull-up resistor to V_{CCAUX} is required.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG clock. If not using in-system programming or Boundary-Scan, tie to GND with resistor.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
VCCINT	3.3V	0.1 μF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
VCCO	3.3V	0.1 μF decoupling capacitor to GND. All V_{CCO} pins must be connected.
VCCJ	3.3V	0.1 μF decoupling capacitor to GND.
GND	0V	Ground, all GND pins must be connected.

Voltage Compatibility

The PROM's VCCINT supply pin must be tied to 3.3V for the serial XCFxxS Platform Flash PROMs. The FPGA's VCCO_2 supply input and the Platform Flash PROM's VCCO supply input must be the same voltage of +3.3V (refer to [Ref 5] for additional information).

Special Hardware Requirements

Buffer on JTAG Interface

The JTAG header enables configuration and verification through the JTAG interface of both the FPGA and PROM. Spartan-3A FPGAs and the Platform Flash PROMs both have a four-wire IEEE 1149.1/1532 JTAG port. These devices share the TCK clock input and the TMS mode select input. The devices can connect in either order on the JTAG chain with the TDO output of one device feeding the TDI input of the following device in the chain. The TDO output of the last device in the JTAG chain drives the JTAG connector. The JTAG interface on the FPGA is powered by the V_{CCAUX} supply. Consequently, the PROM's VCCJ supply input must also be 3.3V. A buffer is required between the JTAG header and the FPGA to improve signal integrity on the JTAG interface in some cases. For long traces or for JTAG scan chains with more than three devices, it is recommended to buffer TMS and TCK. Terminate the JTAG buffer inputs to avoid floating inputs when then the cable is not connected. Apply good clock routing and termination to the JTAG TCK trace to ensure clean clock edges.

LED on DONE Pin

DONE goes High when the FPGA is successfully configured. An easy check for completion is to install an LED driven by the DONE Pin (the LED stays lit during FPGA operation). Refer to Figure 1 for more details on how to connect an LED.

Push Button on PROG_B Pin

When PROG_B Pin is asserted Low, it forces the FPGA to restart its configuration. During prototyping, having the ability to manually restart the FPGA configuration can be useful. This capability can be easily achieved by installing a push button to pulse PROG_B to ground (refer to Figure 1 for more details).

CCLK Termination

For larger boards with long traces, signal integrity can become an issue. For example, poor signal integrity on CCLK can be detrimental to configuration. For boards with long CCLK traces, termination should be added to CCLK to improve signal integrity.

Mode Pin Jumpers

During configuration via JTAG, it is important to reduce any contention on the configuration signals. It is suggested to use the jumpers on the MODE pins to place the FPGA in JTAG mode to configure through JTAG. The MODE pins themselves have internal pull-ups.

Note: The MODE pin setting for JTAG is `101`; for Master Serial, `000`.

In-System Programming Support

The following sections cover the Xilinx ISE™ software flows necessary for programming a Platform Flash PROM during prototyping:

- Preparing a Platform Flash PROM file flow
- iMPACT in-system programming flow for Platform Flash PROM

Generating a Platform Flash File

Before converting a FPGA bitstream into a MCS-formatted PROM file, the designer must verify that the bitstream was generated with the option:

```
bitgen -g StartupClk:Cclk
```

This option ensures proper FPGA functionality by synchronizing the startup sequence to the internal FPGA clock, and then DONE goes High upon successful configuration.

Preparing a Platform Flash MCS File Using the ISE PROMGen Command-Line Software

The Xilinx ISE PROMGen software takes a bitstream (`.bit`) file as input and with the appropriate options, generates a memory image file for the data array of a Platform Flash 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. An example PROMGen software command-line to generate an MCS-formatted file for a 4-Mb (4096-kb) Platform Flash PROM is:

```
promgen -w -p mcs -c FF -o PFP.mcs -u 0 bitfile.bit -x xcf04s
```

The **`-w`** option is required to overwrite an existing output file. The **`-p mcs`** option specifies Intel Hex (`.mcs`) output file format.The **`-c FF`** option calculates a 32-bit checksum for the PROM file with the fill value of `0xFF`. The **`-o PFP.mcs`** specifies the output to the `PFP.mcs` file. The **`-u 0`** option specifies the data to start at address zero of the `.bit` file and fill the data array in the up direction. The `bitfile.bit` is the input bitstream file. The **`-x xcf04s`** specifies the target Platform Flash device as the XCF04S PROM.

Preparing a Platform Flash PROM MCS File Using the ISE iMPACT GUI

The ISE iMPACT 9.1i (or later) software integrates PROM file formatting and in-system programming features behind an intuitive graphical user interface. 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, a Platform Flash PROM file can be generated from a Xilinx FPGA bitstream through a simple ten-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 MCS formatted Platform Flash PROM file in the MCS file format for a 4-Mb XCF04S. The demonstrated process takes the `bitfile.bit` FPGA bitstream file as input and generates a PROM file named, `PFP_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 2). 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.

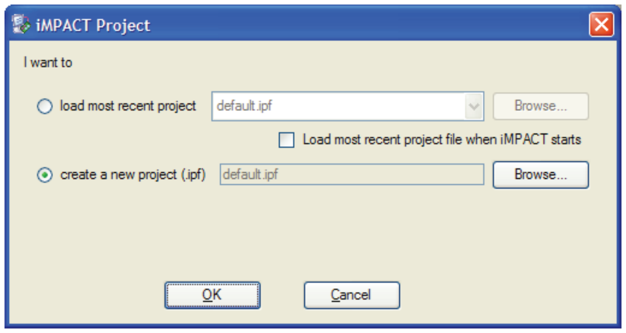


Figure 2: **Create a new Project for PROM File Generation**

Step 2: Choose to Prepare a PROM File

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

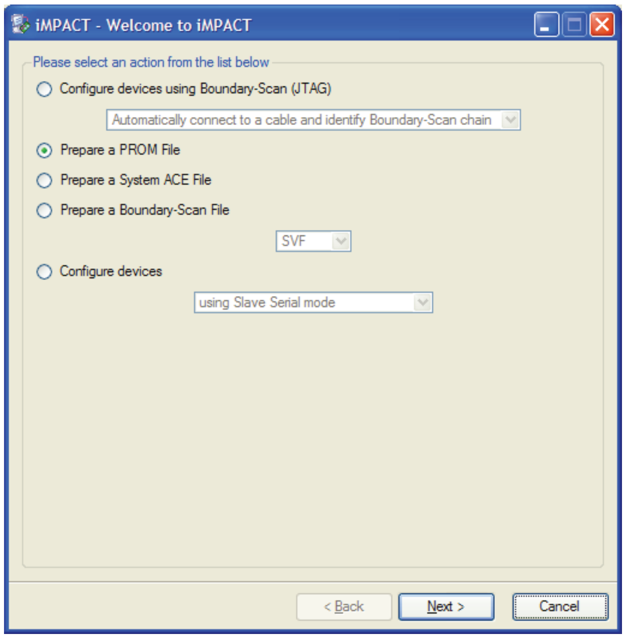
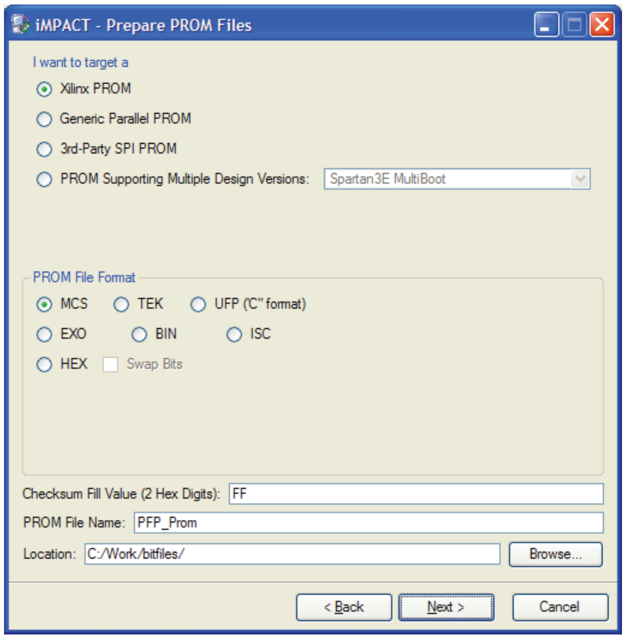

iMPACT - Welcome to iMPACT
Please select an action from the list below
Configure devices using Boundary-Scan (JTAG)
Automatically connect to a cable and identify Boundary-Scan chain
[✓ Prepare a PROM File]
Prepare a System ACE File
Prepare a Boundary-Scan File
SVF
Configure devices
using Slave Serial mode
< Back
Next >
Cancel
X986_03_032407

Figure 3: **Choose to Prepare a PROM File**

Step 3: Specify the Output 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 4). Choose to target the "Xilinx PROM" type. Select the "MCS" PROM file format. Specify the PROM file name to be `PFP_Prom` (to the PFP_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 file.



I want to target a
[✓ Xilinx PROM]
Generic Parallel PROM
3rd-Party SPI PROM
PROM Supporting Multiple Design Versions: Spartan3E MultiBoot

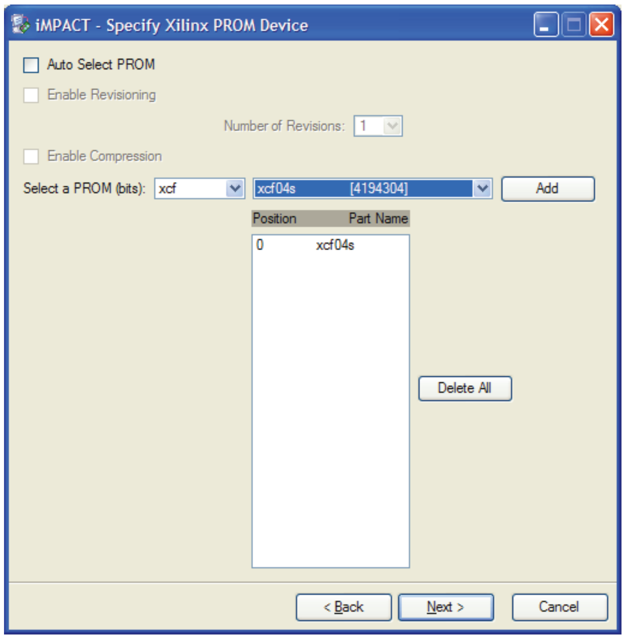
PROM File Format
[✓ MCS] TEK UFP ("C" format)
EXO BIN ISC
HEX Swap Bits

Checksum Fill Value (2 Hex Digits): FF
PROM File Name: PFP_Prom
Location: C:/Work/bitfiles/

Figure 4: **Specify the Output Platform File Options**

Step 4: Specify a Platform Flash PROM Device

The fourth step of the process is to select which Platform Flash PROM (Figure 5). From the "Select a PROM" drop-down list, choose the **xcf** Platform Flash PROM family and the XCF04S 4-Mb device. Click **Add** to add the XCF04S PROM to the list of devices that the PROM file is created for. Click **Next** to proceed to step 5 of the process.



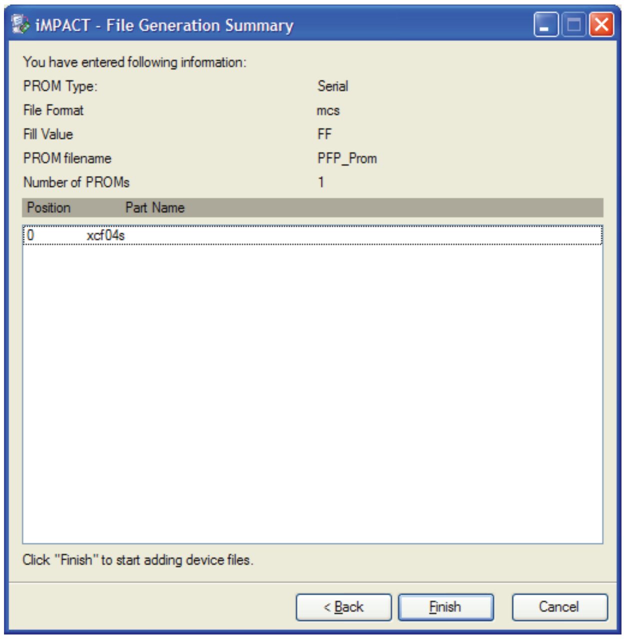
Auto Select PROM
Enable Revisioning
Number of Revisions: 1
Enable Compression
Select a PROM (bits): xcf xcf04s [4194304]

Position	Part Name
0	xcf04s

Figure 5: **Specify the Output Platform File Options**

Step 5: Summary of Platform Flash PROM File Selections

The fifth step in the process displays a summary of the options selected from the prior steps in the process (Figure 6). 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 `pfp_prom` for XCF04S. Click **Finish** to complete the wizard and proceed to step 6 of the process.



You have entered following information:

PROM Type:	Serial
File Format	mcs
Fill Value	FF
PROM filename	PFP_Prom
Number of PROMs	1

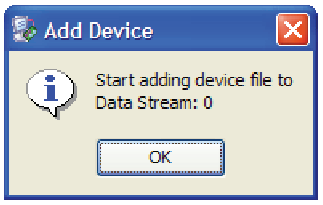
Position	Part Name
0	xcf04s

Click "Finish" to start adding device files.

Figure 6: **Summary of Platform Flash Selections**

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

After the iMPACT project wizard is finished, the iMPACT Platform Flash PROM generation project is set to generate a specific PROM file with specific parameters. At this stage in the process, the PROM file memory image is empty. The sixth step in the process 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 Platform Flash PROM memory image. Click **OK** in the Add Device notification dialog box (Figure 7) to proceed to step 7 of the process.

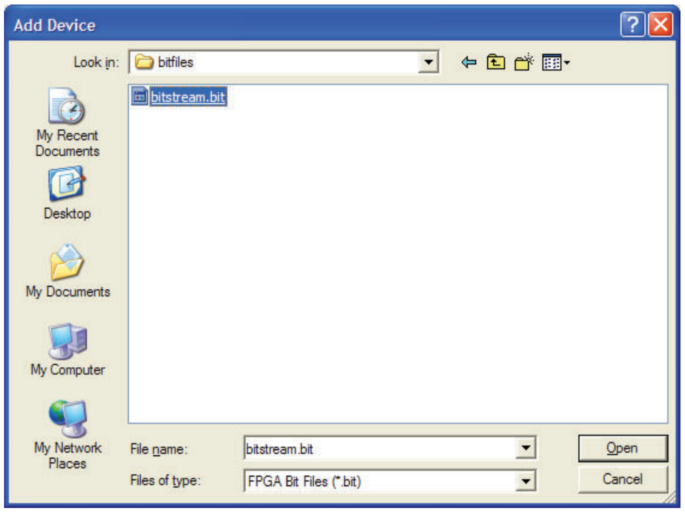


Start adding device file to Data Stream: 0

Figure 7: **Add Device Notification Dialog Box**

Step 7: Select the FPGA Bitstream File to Add to the Platform Flash 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 Platform Flash PROM memory image (Figure 8). Select the FPGA bitstream file to be written to the Platform Flash PROM. Click **Open** in the browser to add the selected FPGA bitstream to the Platform Flash PROM memory image, and proceed to step 8.



File name: bitstream.bit

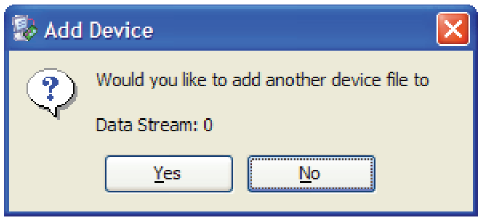
Files of type: FPGA Bit Files (*.bit)

X986_08_032407

Figure 8: **Add Device File Browser**

Step 8: Automated Notification to Add Another Device File to the Platform Flash PROM File

The Step 8 in the process is an automatic notification that the next step is to add another device to the Platform Flash PROM memory image. Click **NO** in the Add Device notification dialog box (Figure 9). Proceed to step 9.



Would you like to add another device file to

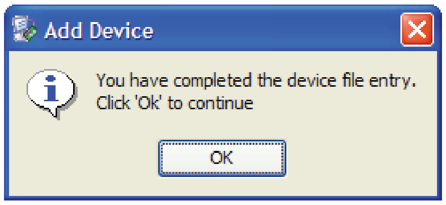
Data Stream: 0

X986_09_032407

Figure 9: **Add Another Device Notification Dialog Box**

Step 9: Automated Notification to the Device File Entry is Completed

The Step 9 in the process is an automatic notification that the device file entry is completed. Click **OK** in the Add Device notification dialog box (Figure 10). This action completes the automated iMPACT process for preparing a Platform Flash PROM file to be generated. Proceed to step 10 to generate the Platform Flash PROM file.



You have completed the device file entry.
Click 'Ok' to continue

X986_10_032407

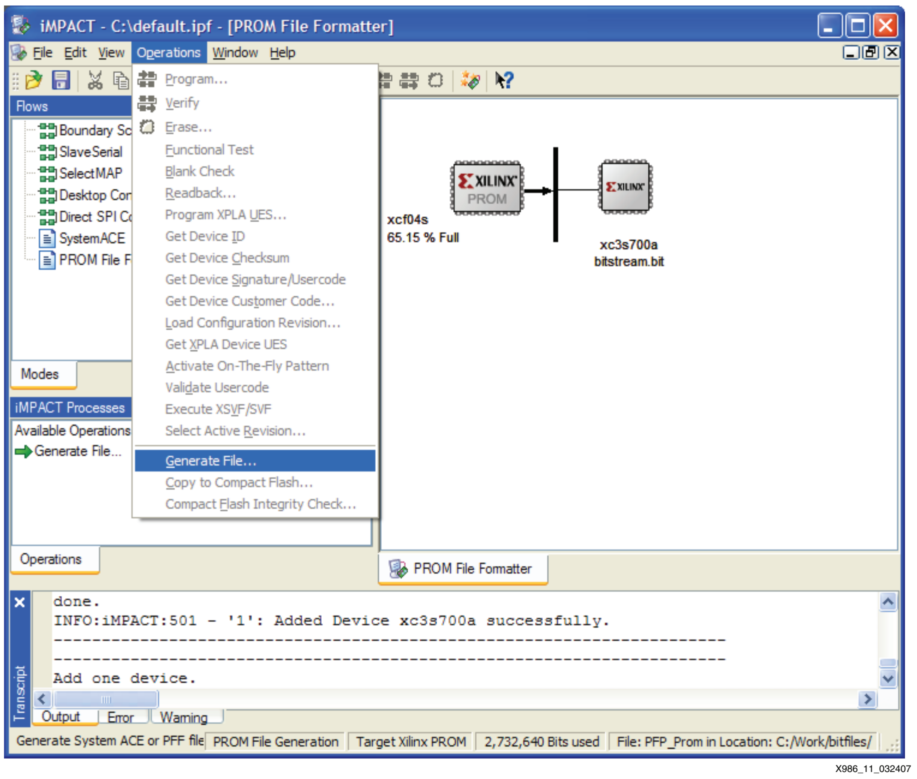
Figure 10: **Device File Entry Complete Notification Dialog Box**

Step 10: iMPACT Generate File Operation

The tenth and final step in the process is to generate the PROM file. Under the iMPACT **Operations** menu, invoke the **Generate File...** menu item (Figure 11). Once invoked, the **Generate File...** menu item causes iMPACT to generate the specified Platform Flash PROM file. iMPACT reports a "PROM File Generation Succeeded" message after successful generation of the Platform Flash PROM file.

After the **Generate File...** operation has completed, the resulting `pfp_prom.mcs` file is available in the specified location. The `pfp_prom.mcs` file can be used in any of the supported programming solutions to program the Platform Flash PROM XCF04S with the specified FPGA bitstream contained within the Platform Flash PROM file.

Save the iMPACT Platform Flash PROM generation project for quick re-generating the Platform Flash PROM file whenever the FPGA bitstream design is revised. To re-generate a Platform Flash PROM file, re-open the saved iMPACT project, and invoke the **Generate File...** operation. iMPACT generates a revised Platform Flash 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 MCS-formatted Platform Flash PROM file. The designer is prompted to name his project and select the option "Prepare a PROM File", following the steps 1–10 outlined above to generate a new Platform Flash PROM File.



```
done.
INFO:iMPACT:501 - '1': Added Device xc3s700a successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
Add one device.
```

Figure 11: **Generate File Menu**

Using iMPACT Software to In-System Program Platform Flash PROMs

In prototype applications, the ISE iMPACT software can be used to program Xilinx Platform Flash devices in-system with a memory image from a given MCS PROM file. The iMPACT software can program Platform Flash PROMs using a simple seven 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 Platform Flash XCF04S PROM. The demonstrated process takes the `PFP_prom.mcs` MCS PROM file (generated in the "Preparing a Platform Flash PROM MCS File Using the ISE iMPACT GUI") as input, erases the PROM, programs the PROM file contents into the XCF04S PROM, and verifies the PROM contents against the given PROM file contents.

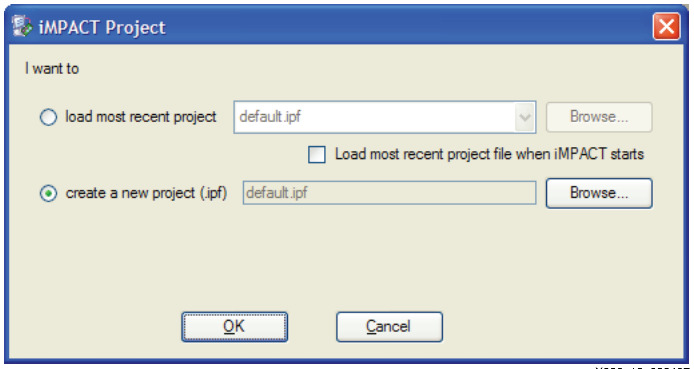
Step 1: Setup Hardware for In-system Programming

The first step of the programming process proper setup of the hardware for in-system programming of the Platform Flash PROM:

- **Proper Xilinx cable connection:** The Xilinx cable must be properly connected to the computer and to the JTAG header of the target FPGA (see Figure 1, page 2 for hardware connections from the Xilinx cable to the JTAG header of the target FPGA).
- **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 Platform Flash PROM.

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

After launching the iMPACT software, the iMPACT Project dialog box is displayed (Figure 12). 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.


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

Figure 12: **Create a New Project for PROM file Generation**

Step 3: Configure Devices Using Boundary-Scan

The second step of the process begins with the iMPACT project wizard. The first dialog box of the wizard displays the available kinds of projects that can be created (Figure 13). Choose the "Configure Devices using Boundary-Scan" option from radio Buttons. And select the "Automatically Connect to a cable and identify Boundary-Scan chain." Click **Finish** to complete the new project setup process. At the completion of this process, iMPACT initializes the JTAG chain and prompts the user for the configuration file for the FPGA (assuming that the FPGA is the first device in the chain and the Platform Flash PROM is the second).

Note: This step assumes that the user has a cable correctly connected to the board, and that the board is powered.

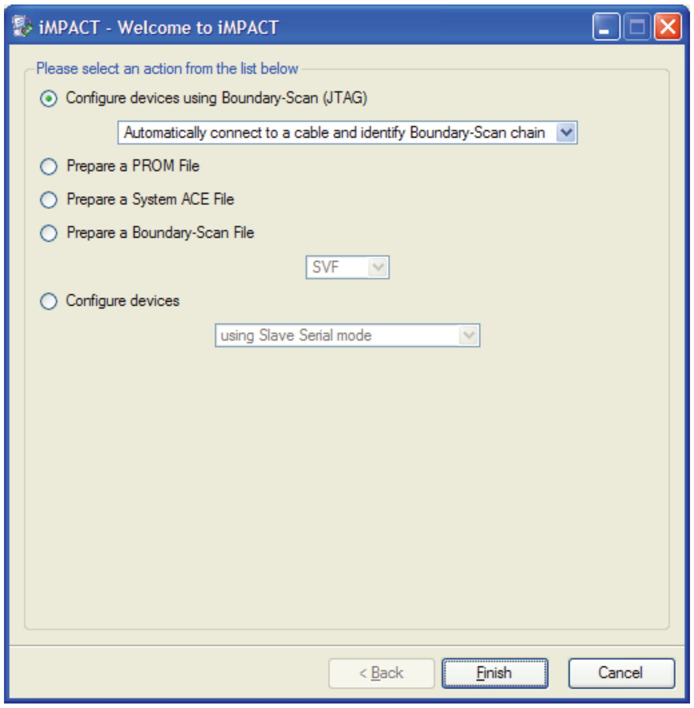

iMPACT - Welcome to iMPACT
Please select an action from the list below
[✓ Configure devices using Boundary-Scan (JTAG)]
Automatically connect to a cable and identify Boundary-Scan chain
Prepare a PROM File
Prepare a System ACE File
Prepare a Boundary-Scan File
SVF
Configure devices
using Slave Serial mode
< Back
Finish
Cancel
X986_13_032407

Figure 13: **Configure Devices Using Boundary-Scan**

Step 4: Assign 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 a FPGA configuration file (Figure 14). Click **Bypass** for the FPGA as there is no need at this time to add a configuration file for the FPGA. After clicking **Bypass** proceed to the next step.

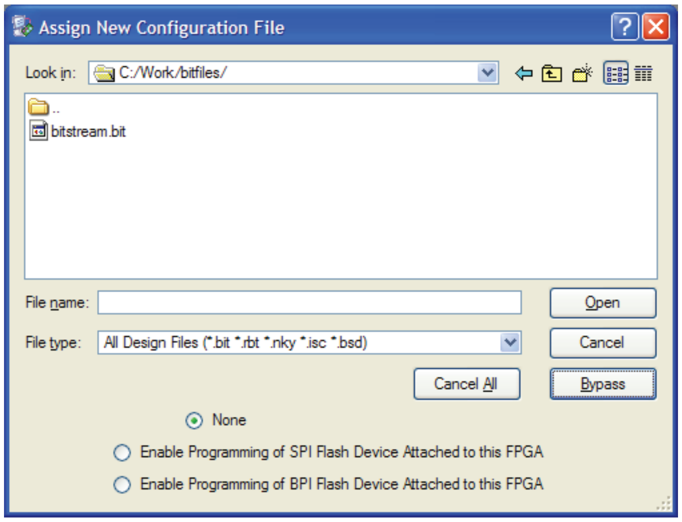
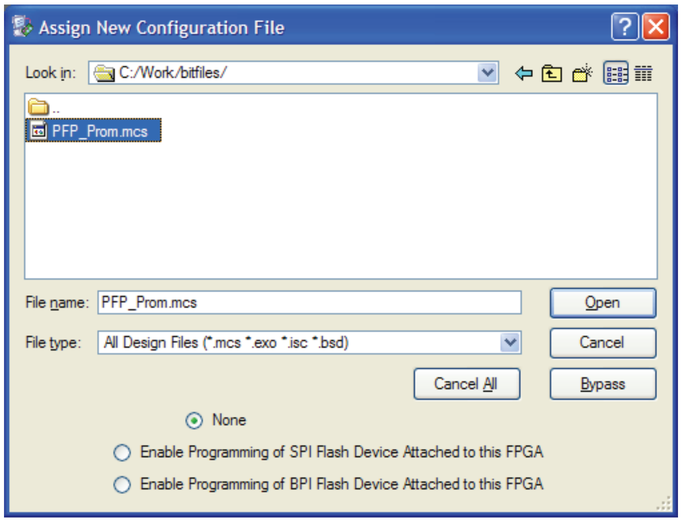

Assign New Configuration File
Look in: C:/Work/bitfiles/
..
bitstream.bit
File name:
File type: All Design Files (*.bit *.rbt *.nky *.isc *.bsd)
Open
Cancel
Cancel All
Bypass
[✓ None]
Enable Programming of SPI Flash Device Attached to this FPGA
Enable Programming of BPI Flash Device Attached to this FPGA
X986_14_032407

Figure 14: **Assign new Configuration File for FPGA**

Step 5: Add a PROM File

After clicking **Bypass** for the FPGA, iMPACT automatically leads into the third step of the process. iMPACT automatically displays a file browser window to select a PROM file for programming into the Platform Flash device (Figure 15). Choose the `PFP_prom.mcs` file, and click **Open**.



Look in: C:/Work/bitfiles/

File name: PFP_Prom.mcs

File type: All Design Files (*.mcs *.exo *.isc *.bsd)

[✓ None]

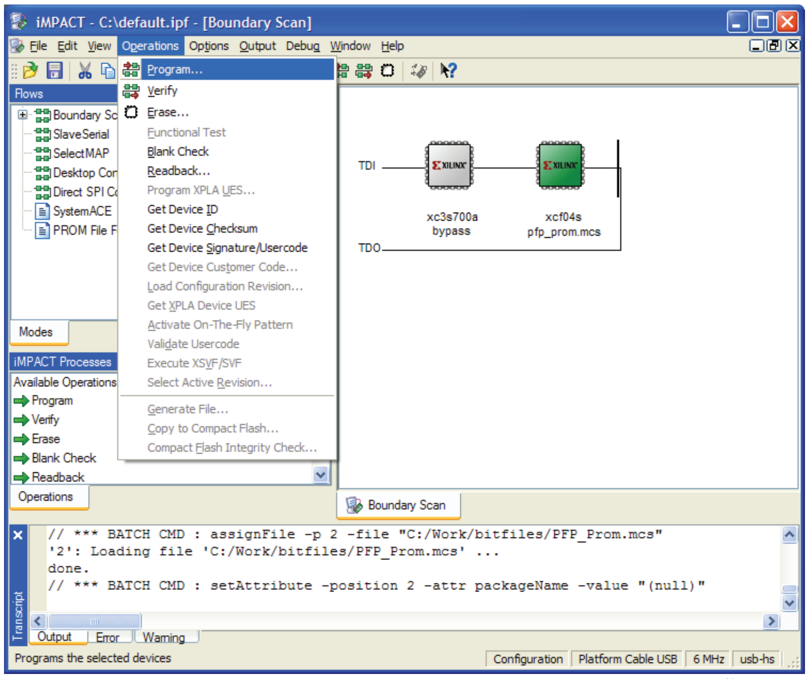
Enable Programming of SPI Flash Device Attached to this FPGA

Enable Programming of BPI Flash Device Attached to this FPGA

Figure 15: **Add a Platform Flash PROM File**

Step 6: Invoke the iMPACT Program Operation

The sixth step of the process programs the target Platform Flash PROM with the selected PROM file contents. Ensure the Platform Flash PROM icon in the iMPACT window is selected by left-clicking on the Platform Flash PROM icon (the Platform Flash PROM icon is highlighted in green when selected). Select **Operations** → **Program** to begin programming (Figure 16).



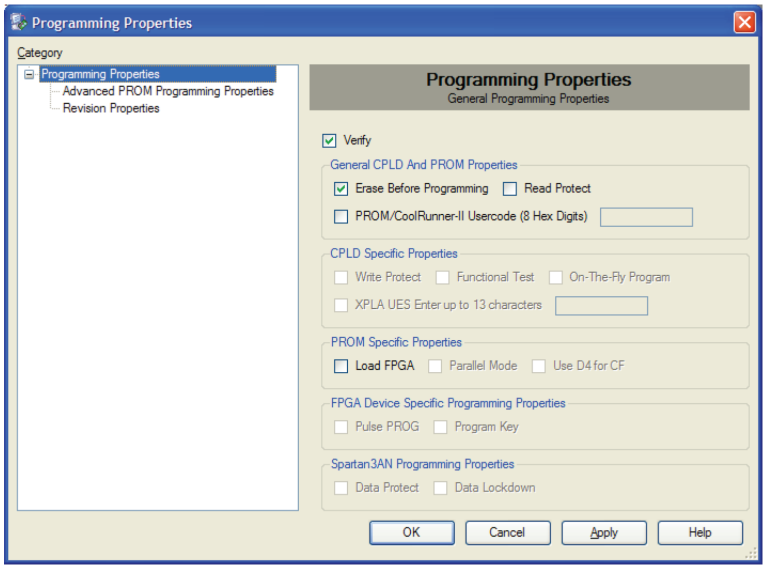
```
// *** BATCH CMD : assignFile -p 2 -file "C:/Work/bitfiles/PFP_Prom.mcs"
'2': Loading file 'C:/Work/bitfiles/PFP_Prom.mcs' ...
done.
// *** BATCH CMD : setAttribute -position 2 -attr packageName -value "(null)"
```

Programs the selected devices

Figure 16: **Program Menu**

Step 7: Select iMPACT Programming Properties

In response to the invocation of the Program operation, iMPACT presents the Programming Properties dialog box (Figure 17). The seventh 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 Platform Flash PROM. Click **OK** to begin the erase, program, and verify operations.



[✓] 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

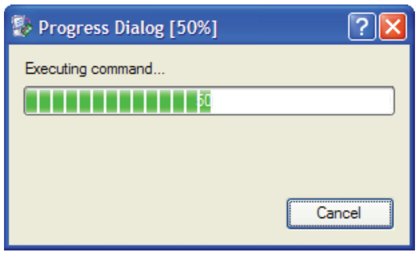
Spartan3AN Programming Properties

[ ] Data Protect [ ] Data Lockdown

X986_17_032407

Figure 17: **Programming Properties Dialog Box**

At the start of the programming operation, iMPACT displays a Progress Dialog box as it progresses through the in-system erase, program, and verify operations (Figure 18). Depending on the size of the Platform Flash PROM, size of the 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.



Executing command...

X986_18_032407

Figure 18: **Progress Dialog Box**

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

Save the iMPACT Platform Flash Programming project for quickly reprogramming the Platform Flash PROM whenever the PROM file is revised. To reprogram the Platform Flash PROM, re-open 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 Platform Flash PROM, assuming the revised PROM file is located in the same location as the original PROM file. After the progress reaches 100%, a "Program Successful" message is displayed.

Note: Pulsing PROG_B with the push button is a good way to confirm programming is successful. The LED on DONE should turn on if programming and configuration are successful.

Programming Time Expectations

The user can expect a programming time of approximately 23 seconds for a 4-Mbit device. Erase on the same device takes approximately 6 seconds. A full erase, program, verify operation takes approximately 32 seconds. All times can vary slightly due to bitstream size and cable speed and/or system speed.

References

1. DS123, *Platform Flash In-System Programmable Configuration PROMs.*
2. DS529, *Spartan-3A FPGA Family Data Sheet.*
3. UG161, *Platform Flash User Guide.*
4. UG332, *Spartan-3 Generation Configuration User Guide.*
5. XAPP453, *The 3.3V Configuration of Spartan-3 FPGAs.*

Conclusion

Following the steps exactly as described in this application note should provide a *bulletproof* configuration solution for the covered devices and setup. This application note is a condensed version of the best practices for configuration and serves mainly as a guide for a basic configuration setup. Refer to [Ref 4], and other appropriate user guides and data sheets for more information.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/16/07	1.0	Initial Xilinx release.
06/22/07	1.0.1	Corrected value of pull-up resistor on DONE pin in Figure 1, page 2.
11/12/07	1.0.2	• Updated document template. • Updated URLs. • Corrected typos.

Notice of Disclaimer