Platform Flash PROM User Guide

UG161 (v1.5) October 26, 2009

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/17/05	1.0	Initial Xilinx release.
08/23/07	1.1	• Added recommendation to use external revision selection pins for dynamic revision selection in "Selecting a Design Revision" in Chapter 3 • Added recommended revision control procedure in "Safe Update Model" in Chapter 3. • Added description of internal feedback path in "CLKOUT" in Chapter 4. • Changed recommended connections for the FPGA CS_B and RDWR_B to ground in SelectMAP schematics in Chapter 5, "Advanced and High-Speed Configuration Setups." • Updated iMPACT software flows to version 9.2i. • Added Appendix A, "References," a listing of references for application solutions and user guides.
01/15/08	1.2	• Hardwired external oscillator to FPGA CCLK in FPGA Slave SelectMAP mode in Figure 2-2, page 26, Figure 5-3, page 48, Figure 5-6, page 54, and Figure 5-8, page 58, where FPGA CCLK is tied to PROM CLK. • Corrected various figure note indices. • Updated text in Table 5-3, page 48, Table 5-5, page 52, Table 5-6, page 54, Table 5-7, page 56, Table 5-8, page 58, to match figures. • Updated URLs.
06/18/08	1.3	• Updated notes regarding the BUSY pin on page 97. • Update trademark notations. • Updated document template.
10/17/08	1.4	• Updated and renamed Figure 5-5, page 52 through Figure 5-8, page 58. • Renamed Table 5-5, page 52 and Table 5-6, page 54. • Revised and renamed Table 5-7, page 56 and Table 5-8, page 58. • Revised the first paragraph in "File Definitions," page 65. • Revised the paragraph after Figure 6-2, page 66. • Updated step 4, page 75 for programming a revisioned PROM. • Added section "Programming an XCFxxP with a Third Party Programmer," page 78, in Chapter 6. • Updated CLKOUT availability sentence in "CLKOUT," page 97.

Date	Version	Revision
10/26/09	1.5	• Updated for ISE® 11.3 software and included Spartan-6 and Virtex-6 FPGAs. Chapter 1: • Revised first two paragraphs in chapter. • Table 1-1, page 16: Changed lower bound on V_{CCO} for XCF08P, XCF16P, and XCF32P devices from 1.5V to 1.8V. Added note 1. • Updated Figure 1-2, page 17. • Added paragraph to end of "XCFxxP Family Members," page 17. • Added section "FPGAs and Compatible Platform Flash PROMs," page 17. Chapter 2: • In all tables, changed PROG_B to PROG_B/PROGRAM_B/$\overline{\text{PROGRAM}}$. • In Figure 2-1, page 24 and Figure 2-2, page 26, added a note for PROG_B. • Added two paragraphs to end of introduction to chapter. • In Table 2-1, page 24 and Table 2-2, page 27: ♦ Added rows for pins CEO, BUSY, EN_EXT_SEL, REV_SEL[1:0], and CLKOUT. ♦ Made changes to "Connect to" column in V_{CCO} row. ♦ Replaced "0V" with "Ground." • Revised first sentence of "Configuring in FPGA Master Serial Mode," page 23. • Added new first sentence and inserted note at the end of "Configuring in FPGA Slave Serial Mode," page 26. Chapter 3: • Revised second paragraph of "Safe Update Model," page 34. Chapter 4: • Revised "Why Use the Internal Decompressor or Oscillator?," page 39. • Revised last sentence of "Internal Oscillator," page 40. • Revised third and fourth paragraphs of "CLKOUT," page 40. Chapter 5: • In all tables, changed PROG_B to PROG_B/PROGRAM_B/$\overline{\text{PROGRAM}}$. • Added paragraph to end of introduction to chapter. • In Table 5-1, page 44, Table 5-2, page 46, Table 5-3, page 48, Table 5-4, page 50, Table 5-5, page 52, Table 5-6, page 54, Table 5-7, page 56, Table 5-8, page 58, Table 5-9, page 60, and Table 5-10, page 62: ♦ Made changes to "Connect to" column in V_{CCO} row. ♦ Replaced "0V" with "Ground." • In Figure 5-1, page 44, Figure 5-2, page 46, Figure 5-3, page 48, Figure 5-4, page 50, Figure 5-5, page 52, Figure 5-6, page 54, Figure 5-7, page 56, Figure 5-8, page 58, Figure 5-9, page 60, and Figure 5-10, page 62, added a note for PROG_B. • In Figure 5-2, page 46, Figure 5-3, page 48, Figure 5-4, page 50, Figure 5-7, page 56, Figure 5-8, page 58, and Figure 5-10, page 62, added a note for CS_B. Chapter 6: • Updated screen captures in Figure 6-4, page 68, Figure 6-5, page 69, Figure 6-6, page 69, Figure 6-11, page 72, Figure 6-12, page 73, Figure 6-14, page 74, Figure 6-16, page 75, and Figure 6-17, page 77. • Revised "File Description" column of Table 6-1, page 65. • Revised "Creating a Simple PROM File with iMPACT Software," page 67 and "Creating an (XCFxxP) Advanced PROM File with iMPACT Software," page 71. • Added first paragraph and note to beginning of "Programming an XCFxxP with a Third Party Programmer," page 78. • Added first paragraph and note to end of "Third-Party Device Programmers," page 90.

Date	Version	Revision
10/26/09	1.5 *(Cont'd)*	Chapter 7: • Table 7-1, page 96: Changed lower bound on V_{CCO} for XCFxxP family from 1.5V to 1.8V. • Revised"Regulating 3.3V Down to 1.8V for XCF08P/16P/32P Devices," page 94 and "Printed Circuit Board Considerations," page 98. • Revised"Using the PROM CF Pin to Initiate FPGA Configuration," page 95. • Added section "FPGA Design Considerations," page 99. • Revised"BUSY Pin," page 97, and note following section. Appendix A: • Added sections "Revision Managers for Safe Update Solutions," page 101 and "Device Reliability and Technology Information," page 102.

Table of Contents

Chapter 5: Advanced and High-Speed Configuration Setups

Chapter 6: PROM File Creation and Programming Flow

Chapter 7: Design Considerations

Appendix A: References

About This Guide

The *Platform Flash PROM User Guide* describes advanced features of the Platform Flash PROM family of devices. This guide also includes information on configuration setups, generation of the files used to program this PROM family, and other design considerations.

More information on the Platform Flash PROM family is available online in DS123, *Platform Flash PROM Data Sheet*.

Organization

This guide covers the following topics:

- Chapter 1, "Introduction to the Platform Flash PROM Family," provides a brief description of Platform Flash PROM family members and key features.
- Chapter 2, "Platform Flash PROM Quick Start," describes the two most popular and basic configuration mode setups: FPGA Master Serial mode and FPGA Slave Serial mode.
- Chapter 3, "XCFxxP Design Revisions," describes the design revision capability in the XCFxxP Platform Flash PROM family.
- Chapter 4, "XCFxxP Decompression and Clock Options," describes the internal oscillator and a built-in decompressor features in the XCFxxP family.
- Chapter 5, "Advanced and High-Speed Configuration Setups," describes advanced Platform Flash PROM configuration mode setups.
- Chapter 6, "PROM File Creation and Programming Flow," describes file extensions, file generation flows, and programming steps required when targeting the Platform Flash PROM family.
- Chapter 7, "Design Considerations," describes considerations for existing and new designs.
- Appendix A, "References," provides a complete listing of documents referenced in this user guide.

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/support/documentation/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

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

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**`lowpwr ={on|off}`**
Vertical bar |	Separates items in a list of choices	**`lowpwr ={on|off}`**

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis ...	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

Introduction to the Platform Flash PROM Family

This guide supplements DS123, *Platform Flash PROM Data Sheet* and Spartan® and Virtex® FPGA configuration user guides, giving specific information on Platform Flash PROM features and usage modes.

The Platform Flash PROM gives designers a memory storage solution that supports both Spartan and Virtex FPGA families. The Platform Flash PROM family provides non-volatile storage, as well as an integrated bitstream delivery mechanism for use with target FPGAs.

Family Description

The Xilinx® Platform Flash series of in-system programmable configuration PROMs is available in 1 Mb to 32 Mb densities. The Platform Flash PROM family, which includes both 3.3V XCFxxS and 1.8V XCFxxP members, was developed using a low-power, advanced CMOS NOR FLASH process. Family members operate over the full industrial temperature range (-40°C to +85°C), and members can be cascaded allowing for more storage capacity. This family supports the IEEE 1149.1 and IEEE 1532 standards.

XCF01S, XCF02S, and XCF04S (1 Mb, 2 Mb, and 4 Mb) features include:

- 3.3V supply voltage
- Serial FPGA configuration interface
- V020 package
- Convenient built-in power-up sequencing for FPGA data integrity

XCF08P, XCF16P, and XCF32P (8 Mb, 16 Mb, and 32 Mb) features include:

- 1.8V supply voltage
- Serial or parallel FPGA configuration interface
- Multiple design revisions for configuration
- Built-in data decompressor
- FS48 and VO48 packages
- Internal oscillator (CLKOUT)
- Convenient built-in power-up sequencing for FPGA data integrity

Feature Summary

Platform Flash PROM family members and their supported features are shown in Table 1-1. High-density Platform Flash PROM members (XCF08P, XCF16P, and XCF32P) have an advanced feature set.

Table 1-1: **Platform Flash PROM Feature Summary**

Device	Density	V_{CCINT}	V_{CCO} Range	V_{CCJ} Range	Packages	JTAG ISP Programming	Serial Configuration	Parallel Configuration	Design Revisioning	Compression
XCF01S	1 Mb	3.3V	1.8V – 3.3V	2.5V – 3.3V	VO20/VOG20	✓	✓			
XCF02S	2 Mb	3.3V	1.8V – 3.3V	2.5V – 3.3V	VO20/VOG20	✓	✓			
XCF04S	4 Mb	3.3V	1.8V – 3.3V	2.5V – 3.3V	VO20/VOG20	✓	✓			
XCF08P	8 Mb	1.8V	1.8V – 3.3V	2.5V – 3.3V	VO48/VOG48 FS48/FSG48	✓	✓	✓	✓(1)	✓
XCF16P	16 Mb	1.8V	1.8V – 3.3V	2.5V – 3.3V	VO48/VOG48 FS48/FSG48	✓	✓	✓	✓	✓
XCF32P	32 Mb	1.8V	1.8V – 3.3V	2.5V – 3.3V	VO48/VOG48 FS48/FSG48	✓	✓	✓	✓	✓

Notes:

1. XCF08P supports storage of a design revision only when cascaded with another XCFxxP PROM. See Chapter 3, "XCFxxP Design Revisions," for details.

XCFxxS Family Members

XCF01S, XCF02S, and XCF04S family members support Master Serial and Slave Serial configuration modes. An overview of control, data output, clock, and JTAG signals for these PROMs is shown in Figure 1-1.

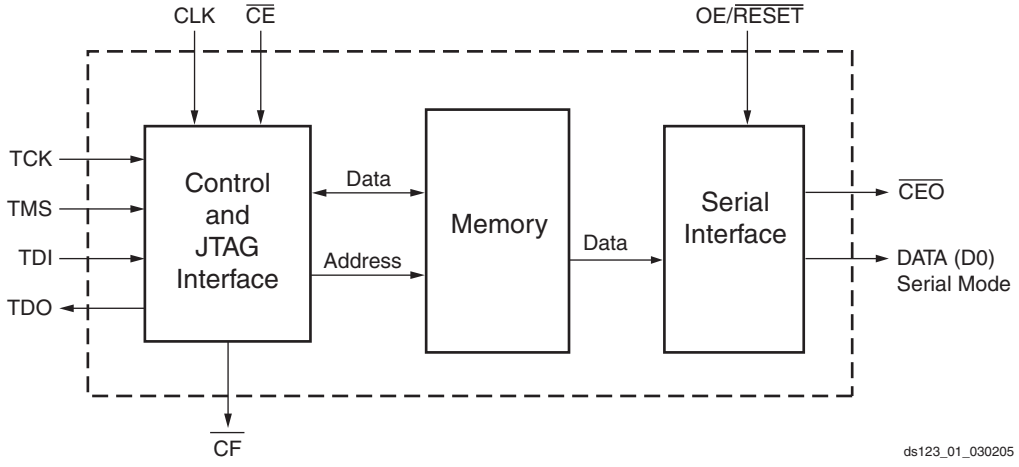


Figure 1-1: **XCF01S/02S/04S Platform Flash PROM Block Diagram**

XCFxxP Family Members

XCF08P, XCF16P, and XCF32P family members support SelectMAP (parallel), and Serial configuration modes. A block diagram of the control, data output, clock, and JTAG signals, as well as additional decompressor logic, is shown in Figure 1-2. The advanced features and higher density provided by members of this family offer enormous flexibility.

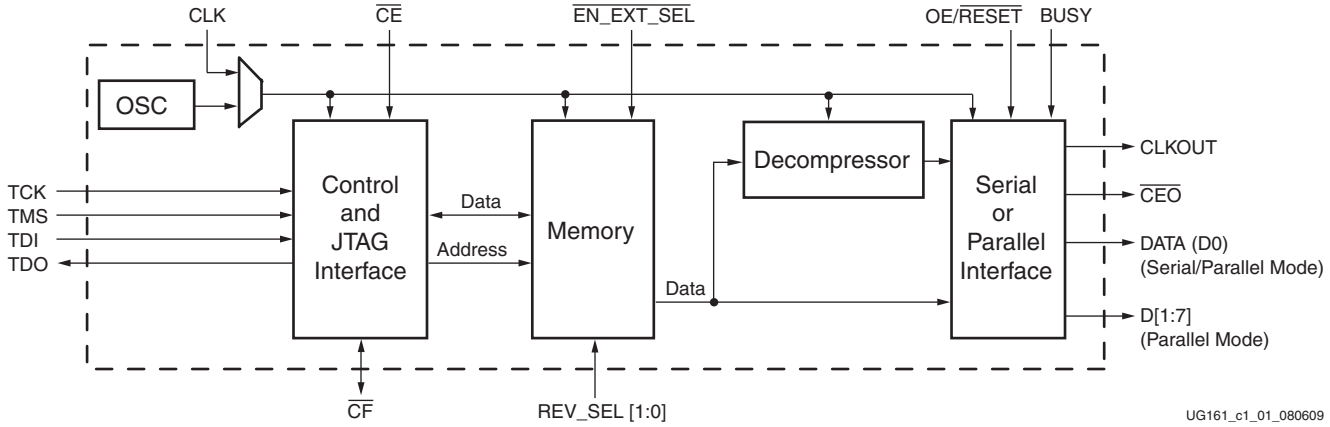


Figure 1-2: **XCF08P/16P/32P Platform Flash PROM Block Diagram**

See Chapter 4, "XCFxxP Decompression and Clock Options," for recommended use of the decompressor, CLKOUT, and internal oscillator features.

FPGAs and Compatible Platform Flash PROMs

The Platform Flash PROMs can configure Spartan and Virtex FPGAs. A reference list of FPGAs and Platform Flash PROMs is given in Table 1-2. For the FPGA bitstream length, refer to the corresponding FPGA configuration user guide.

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs**

FPGA	Platform Flash PROM(1)
Virtex-6 FPGAs	
XC6VLX75T	XCF32P
XC6VLX130T	XCF128X(2) or XCF32P+XCF16P
XC6VLX195T	XCF128X(2) or XCF32P+XCF32P
XC6VLX240T	XCF128X(2) or XCF32P+XCF32P+XCF08P
XC6VLX365T	XCF128X(2) or XCF32P+XCF32P+XCF32P
XC6VLX550T	See UG360, *Virtex-6 FPGA Configuration User Guide* for BPI flash.
XC6VLX760	See UG360, *Virtex-6 FPGA Configuration User Guide* for BPI flash.
XC6VHX250T	XCF128X(2) or XCF32P+XCF32P+XCF16P
XC6VHX255T	XCF128X(2) or XCF32P+XCF32P+XCF16P
XC6VHX380T	XCF128X(2)
XC6VHX565T	See UG360, *Virtex-6 FPGA Configuration User Guide* for BPI flash.
XC6VSX315T	XCF128X(2)
XC6VSX475T	See UG360, *Virtex-6 FPGA Configuration User Guide* for BPI flash.

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs *(Cont'd)***

FPGA	Platform Flash PROM(1)
Virtex-5 LX FPGAs	
XC5VLX30	XCF08P
XC5VLX50	XCF16P
XC5VLX85	XCF32P
XC5VLX110	XCF32P
XC5VLX155	XCF128X(2) or XCF32P+XCF08P
XC5VLX220	XCF128X(2) or XCF32P+XCF32P
XC5VLX330	XCF128X(2) or XCF32P+XCF32P+XCF16P
Virtex-5 LXT FPGAs	
XC5VLX20T	XCF08P
XC5VLX30T	XCF16P
XC5VLX50T	XCF16P
XC5VLX85T	XCF32P
XC5VLX110T	XCF32P
XC5VLX155T	XCF128X(2) or XCF32P+XCF16P
XC5VLX220T	XCF128X(2) or XCF32P+XCF32P
XC5VLX330T	XCF128X(2) or XCF32P+XCF32P+XCF16P
Virtex-5 SXT FPGAs	
XC5VSX35T	XCF16P
XC5VSX50T	XCF32P
XC5VSX95T	XCF128X(2) or XCF32P+XCF08P
XC5VSX240T	XCF128X(2) or XCF32P+XCF32P+XCF16P
Virtex-5 FXT FPGAs	
XC5VFX30T	XCF16P
XC5VFX70T	XCF32P
XC5VFX100T	XCF128X(2) or XCF32P+XCF08P
XC5VFX130T	XCF128X(2) or XCF32P+XCF16P
XC5VFX200T	XCF128X(2) or XCF32P+XCF32P+XCF08P
Virtex-5 TXT FPGAs	
XC5VTX150T	XCF128X(2) or XCF32P+XCF16P
XC5VTX240T	XCF128X(2) or XCF32P+XCF32P
Virtex-4 LX FPGAs	
XC4VLX15	XCF08P
XC4VLX25	XCF08P
XC4VLX40	XCF16P
XC4VLX60	XCF32P
XC4VLX80	XCF32P

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs *(Cont'd)***

FPGA	Platform Flash PROM[1]
XC4VLX100	XCF32P
XC4VLX160	XCF32P+XCF08P
XC4VLX200	XCF32P+XCF32P
Virtex-4 FX FPGAs	
XC4VFX12	XCF08P
XC4VFX20	XCF08P
XC4VFX40	XCF16P
XC4VFX60	XCF32P
XC4VFX100	XCF32P
XC4VFX140	XCF32P+XCF16P
Virtex-4 SX FPGAs	
XC4VSX25	XCF16P
XC4VSX35	XCF16P
XC4VSX55	XCF32P
Virtex-II Pro X FPGAs	
XC2VPX20	XCF08P
XC2VPX70	XCF32P
Virtex-II Pro FPGAs	
XC2VP2	XCF02S
XC2VP4	XCF04S
XC2VP7	XCF08P
XC2VP20	XCF08P
XC2VP30	XCF16P
XC2VP40	XCF16P
XC2VP50	XCF32P
XC2VP70	XCF32P
XC2VP100	XCF32P+XCF08P
Virtex-II FPGAs	
XC2V40	XCF01S
XC2V80	XCF01S
XC2V250	XCF02S
XC2V500	XCF04S
XC2V1000	XCF04S
XC2V1500	XCF08P
XC2V2000	XCF08P
XC2V3000	XCF16P
XC2V4000	XCF16P

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs *(Cont'd)***

FPGA	Platform Flash PROM(1)
XC2V6000	XCF32P
XC2V8000	XCF32P
Virtex-E FPGAs	
XCV50E	XCF01S
XCV100E	XCF01S
XCV200E	XCF02S
XCV300E	XCF02S
XCV400E	XCF04S
XCV405E	XCF04S
XCV600E	XCF04S
XCV812E	XCF08P
XCV1000E	XCF08P
XCV1600E	XCF08P
XCV2000E	XCF16P
XCV2600E	XCF16P
XCV3200E	XCF16P
Virtex FPGAs	
XCV50	XCF01S
XCV100	XCF01S
XCV150	XCF01S
XCV200	XCF02S
XCV300	XCF02S
XCV400	XCF04S
XCV600	XCF04S
XCV800	XCF08P
XCV1000	XCF08P
Spartan-6 FPGAs	
XC6SLX4	XCF04S
XC6SLX9	XCF04S
XC6SLX16	XCF04S
XC6SLX25	XCF08P
XC6SLX25T	XCF08P
XC6SLX45	XCF16P
XC6SLX45T	XCF16P
XC6SLX75	XCF32P
XC6SLX75T	XCF32P
XC6SLX100	XCF32P

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs *(Cont'd)***

FPGA	Platform Flash PROM[1]
XC6SLX100T	XCF32P
XC6SLX150	XCF32P+XCF08P[3]
XC6SLX150T	XCF32P+XCF08P[3]
Spartan-3A FPGAs	
XC3S50A	XCF01S
XC3S200A	XCF02S
XC3S400A	XCF02S
XC3S700A	XCF04S
XC3S1400A	XCF08P
Spartan-3A DSP FPGAs	
XC3SD1800A	XCF08P
XC3SD3400A	XCF16P
Spartan-3E FPGAs	
XC3S100E	XCF01S
XC3S250E	XCF02S
XC3S500E	XCF04S
XC3S1200E	XCF04S
XC3S1600E	XCF08P
Spartan-3 FPGAs	
XC3S50	XCF01S
XC3S200	XCF01S
XC3S400	XCF02S
XC3S1000	XCF04S
XC3S1500	XCF08P
XC3S2000	XCF08P
XC3S4000	XCF16P
XC3S5000	XCF16P
Spartan-IIE FPGAs	
XC2S50E	XCF01S
XC2S100E	XCF01S
XC2S150E	XCF02S
XC2S200E	XCF02S
XC2S300E	XCF02S
XC2S400E	XCF04S
XC2S600E	XCF04S
Spartan-II FPGAs	
XC2S15	XCF01S

Table 1-2: **Xilinx FPGAs and Compatible Platform Flash PROMs *(Cont'd)***

FPGA	Platform Flash PROM(1)
XC2S30	XCF01S
XC2S50	XCF01S
XC2S100	XCF01S
XC2S150	XCF01S
XC2S200	XCF02S

Notes:

1. If design revisioning or other advanced feature support is required, the XCFxxP can be used as an alternative to the XCF01S, XCF02S, or XCF04S.
2. For information about the XCF128X, see DS617, *Platform Flash XL High-Density Configuration and Storage Device.*
3. In certain cases, an FPGA bitstream can fit into fewer XCFxxP PROMs. See Chapter 4, "XCFxxP Decompression and Clock Options," for considerations.

Chapter 2

Platform Flash PROM Quick Start

The Platform Flash PROM family supports numerous FPGA configuration modes. This chapter highlights the two most popular basic configuration mode setups:

- FPGA Master Serial mode
- FPGA Slave Serial mode

These modes are the most popular because of their ease of use and the low pin count required to support them. The serial interface uses five signals in addition to power and ground: Chip Enable ($\overline{CE}$), Output Enable/Reset (OE/$\overline{RESET}$), Configuration Clock (CCLK), Data (D0), and Configuration Pulse ($\overline{CF}$).

The difference between FPGA Master Serial mode and FPGA Slave Serial mode is the clock source. In FPGA Master Serial mode, the FPGA generates a configuration clock that drives the PROM. In FPGA Slave Serial mode, the FPGA's configuration clock is driven from an external source, such as an oscillator.

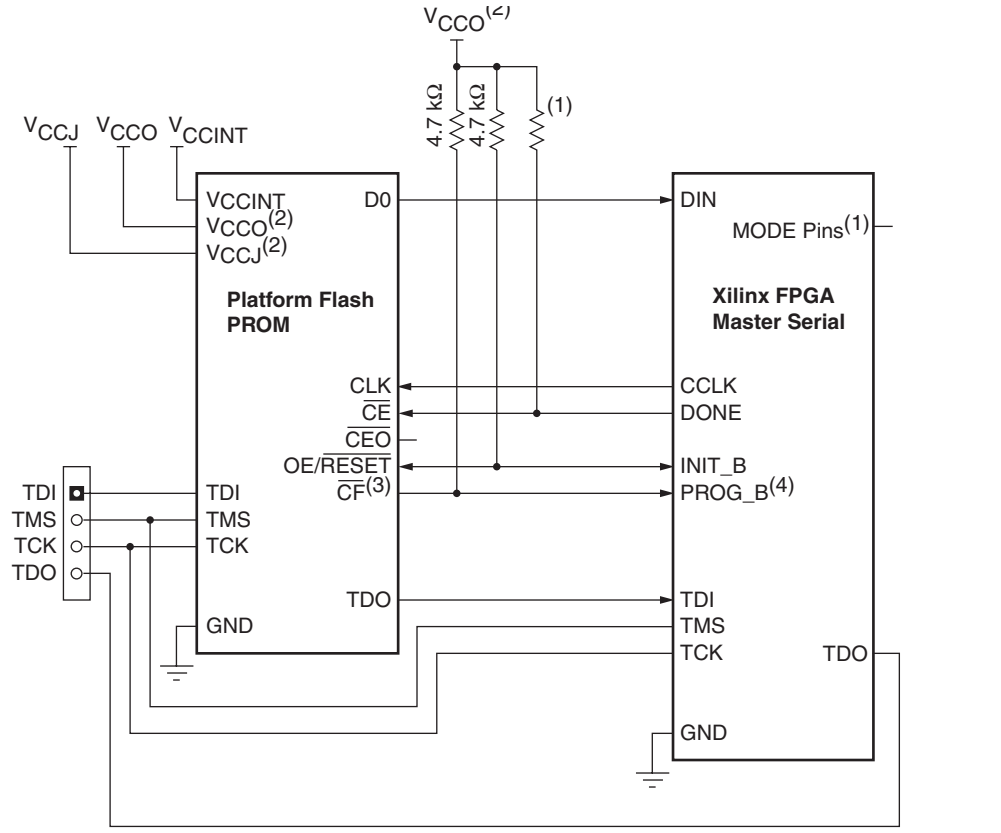
This chapter covers the basic pin connections and provides brief descriptions of FPGA Master Serial mode and FPGA Slave Serial mode. Refer to "Creating a Simple PROM File with iMPACT Software" in Chapter 6 for details on PROM file generation and description.

Serial interface features common to all FPGA families are discussed in this chapter. Each FPGA family can also have unique attributes in its serial configuration interface. Refer to the FPGA family's serial configuration interface section in its corresponding user guide for details of unique attributes.

Refer to Chapter 7, "Design Considerations," to ensure a robust configuration design. The information in the Design Considerations chapter supplements the recommendations provided in this chapter.

Configuring in FPGA Master Serial Mode

The most basic Master Serial configuration is shown in Figure 2-1, where a single PROM configures a single FPGA. Only a small number of signals are required to interface the PROM with the FPGA, and an external clock source is not required for configuration. In FPGA Master Serial mode, the FPGA generates the configuration clock. In this mode, data is available on the PROM Data (D0) pin when $\overline{CF}$ is High, and $\overline{CE}$ and OE are enabled. New data is available a short access time after each rising clock edge.



Notes:

1. For Mode pin connections and DONE pin pull-up value, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. For XCFxxS devices, the $\overline{CF}$ pin is an output pin. For XCFxxP devices, the $\overline{CF}$ pin is a bidirectional pin.
4. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_01_091509

Figure 2-1: **Basic FPGA Master Serial Configuration Setup**

Table 2-1 shows pin connections for FPGA Master Serial mode.

Table 2-1: **Pin Connections for Basic FPGA Master Serial Mode**

XCFxxS / XCFxxP Pins	Connect to	Comment
D0	FPGA DIN	PROM and FPGA Data signal.
CLK	FPGA CCLK	FPGA-generated clock signal.
$\overline{CE}$	FPGA DONE	See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$/INIT_B	4.7 KΩ pull-up resistor to V_{CCO} is required.

Table 2-1: **Pin Connections for Basic FPGA Master Serial Mode *(Cont'd)***

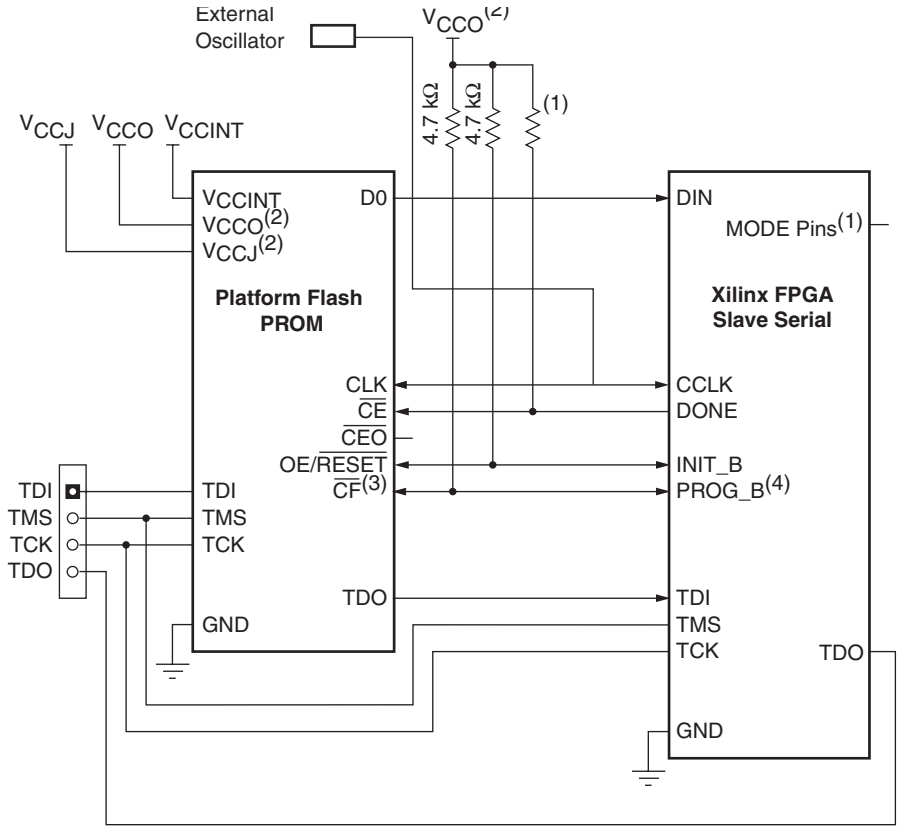
XCFxxS / XCFxxP Pins	Connect to	Comment
$\overline{CF}$	FPGA PROG_B/$\overline{PROGRAM}$_B/ $\overline{PROGRAM}$	XCFxxS: Open-drain output. XCFxxP: Bidirectional pin has open-drain output. 4.7 KΩ pull-up resistor to V_{CCO} is required.
CEO	Do Not Connect	Not used for single PROM setup. In a PROM daisy-chain, CEO drives CE of the downstream PROM.
BUSY	Do Not Connect	XCFxxP only. Not used during serial configuration. An internal pull-down keeps BUSY disabled.
EN_EXT_SEL	Ground	XCFxxP only. Default to select revision 0 for simple PROM setup.
REV_SEL[1:0]	Ground	XCFxxP only. Default to revision 0 for simple PROM setup.
CLKOUT	Do Not Connect	XCFxxP only.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	3.3V	XCFxxS: 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
	1.8V	XCFxxP: 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5V	XCFxxS: 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
	3.3/2.5/1.8V	XCFxxP: 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	XCFxxS: 0.1 µF decoupling capacitor to GND.
	3.3/2.5V	XCFxxP: 0.1 µF decoupling capacitor to GND.
GND	Ground	All ground (GND) pins must be connected.

XAPP986, *Bulletproof Configuration Guide for Spartan-3A FPGAs*, provides a reference design for Master-Serial mode configuration from a Platform Flash PROM.

Configuring in FPGA Slave Serial Mode

The most basic Slave Serial configuration is shown in Figure 2-2, where a single PROM configures a single FPGA. When the FPGA is in Slave Serial mode, the PROM and the FPGA are typically clocked by an external clock source, such as an oscillator. In FPGA Slave Serial mode, data is available on the PROM Data (D0) pin when $\overline{CF}$ is High, and both $\overline{CE}$ and OE are enabled. New data is available a short access time after each rising clock edge.

Note: For only Spartan®-6 FPGAs, the use of the Master Serial configuration mode with an external configuration clock is recommended instead of Slave Serial mode. See "External Configuration Clock for Master Modes" in UG380, *Spartan-6 FPGA Configuration User Guide* for information. The Spartan-6 FPGA can use an external clock source for its Master Serial mode, which provides the performance of the traditional Slave Serial mode with a re-usable and simplified clock scheme.



Notes:

1. For Mode pin connections and DONE pin pull-up value, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. For XCFxxS PROMs, the $\overline{CF}$ pin is an output pin. For XCFxxP PROMs, the $\overline{CF}$ pin is a bidirectional pin.
4. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_02_091509

Figure 2-2: **Basic FPGA Slave Serial Configuration Setup**

Table 2-2 shows pin connections for FPGA Master Serial mode.

Table 2-2: **Pin Connections for Basic FPGA Slave Serial Mode**

Pins	Connect to	Comment
D0	FPGA DIN	PROM and FPGA Data signal.
CLK	External CLK and FPGA CCLK	External CLK used to drive PROM data for FPGA configuration.
CE	FPGA DONE	See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
CEO	Do Not Connect	Not used for single PROM setup. In a PROM daisy-chain, CEO drives CE of the downstream PROM.
BUSY	Do Not Connect	XCFxxP only. Not used during serial configuration. An internal pull-down keeps BUSY disabled.
EN_EXT_SEL	Ground	XCFxxP only. Default to select revision 0 for simple PROM setup.
REV_SEL[1:0]	Ground	XCFxxP only. Default to revision 0 for simple PROM setup.
CLKOUT	Do Not Connect	XCFxxP only.
OE/$\overline{\text{RESET}}$	FPGA $\overline{\text{INIT}}$/INIT_B	4.7 KΩ pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{\text{PROGRAM}}$	XCFxxS: Open-drain output. XCFxxP: Bidirectional pin has open-drain output. 4.7 KΩ pull-up resistor to V_{CCO} is required.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	3.3V	XCFxxS: 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
	1.8V	XCFxxP: 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5V	XCFxxS: 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
	3.3/2.5/1.8V	XCFxxP: 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	XCFxxS: 0.1 µF decoupling capacitor to GND.
	3.3/2.5V	XCFxxP: 0.1 µF decoupling capacitor to GND.
GND	Ground	All ground (GND) pins must be connected.

Chapter 3

XCFxxP Design Revisions

The design revision capability in XCFxxP Platform Flash PROMs is supported in serial as well as SelectMAP (parallel) configuration mode. This section discusses design revision structural requirements, how to use design revisions, and how to select a particular design revision in a system solution.

For guidance on how to create a PROM programming file with design revision capability, see "Creating an (XCFxxP) Advanced PROM File with iMPACT Software" in Chapter 6.

Why Use Design Revisions?

The Platform Flash PROM design revision capability provides tremendous advantages. This feature gives Xilinx® FPGA systems greater flexibility by allowing onboard access to system design variations, either internally through a programmable bit setting or externally via pin usage.

Prototyping Advantage

During the prototyping stage, numerous design variations are often tested on a system. Multiple configurations for the system of Xilinx FPGAs can be stored in the Platform Flash PROM's design revisions. This gives designers a quick way to debug and test up to four variations of a target design without having to reprogram the PROM from an external source each time a minor update is tested.

Board Test Configuration or Safe Update

Design revisions can be used to store a diagnostic design for the system checkout in one revision and store the system application design in a second revision. This eliminates the need for an external hook-up for diagnostic patterns. The diagnostic pattern can be downloaded in seconds from the first revision in the onboard PROM, and there is no need to dispatch a technician to manually load diagnostic patterns on-site. This is not only a significant cost reduction but it also enables the system debug time to be minimized.

Utilizing design revisions can also ensure that a remote configuration update is done safely. When updating a PROM image remotely, an error in the FPGA image received results in an FPGA configuration failure and might require the system to be brought to the factory for an update. This could leave the customer with a system down situation. The Platform Flash PROM design revision feature decreases the possibility of a system failure during remote upgrades by targeting only one design revision for the new update. The original design revision can be left intact. If the system fails to respond correctly after the remote update, the system can automatically revert back to the original design for a robust update.

Multiple Applications for One Target System

Numerous applications can benefit from the ability to reconfigure the system without having to bring the system down or reprogram the onboard PROM. Design revisions can store various fixed DSP functions for an application. Additionally, a configuration file for processing data packets can be stored in one revision and a configuration file for processing both the data and voice packets stored in a second revision. Multiple revisions can also allow low-power products to be put in standby mode with one revision when not in use and to be reconfigured to the original function with another revision when enabled.

An additional advantage of design revisions is that a product's features can be set late in the manufacturing cycle based on demand. For example, systems required to support multiple geographic regions with different protocols or functions can be covered with a single Platform Flash PROM using design revisions. The specific region can be selected on a finished product simply by changing the external pin selection (REV_SEL[1:0]).

Protection from Incomplete or Corrupt Data

Many applications call for the update of just one of many design revisions stored in a PROM. An update process can reprogram just one select design revision in the PROM while leaving the remaining design revisions intact. The system selects the most appropriate design revision for FPGA configuration.

In some cases, a system can easily select a valid design revision for FPGA configuration; in other cases, the determination of a valid design revision is more difficult. In the easy case, some design revision areas are filled with valid bitstreams and other design revision areas are unused (blank). In the more difficult case for determining a valid bitstream, a design revision is only partially programmed with a bitstream.

The XCFxxP Platform Flash PROM supports IEEE Std 1532-style protection for partially programmed design revisions. IEEE Std 1532 specifies the DONE concept in which the programmable device must appear as if it is blank, unless its programming sequence is fully complete to a DONE state (similar to the FPGA DONE pin in which the FPGA I/Os remain in inactive state until the DONE signal goes High). When an appropriate programming sequence is used on the XCFxxP PROM, each data block in a design revision area appears blank unless the programming sequence is fully complete. This feature can protect against the appearance of a partially programmed design revision within an XCFxxP PROM.

What is a Design Revision?

A design revision contains a single configuration or version of the FPGA(s) target application. The Platform Flash PROM family members (XCF08P/XCF16P/XCF32P) have the ability to store up to four unique FPGA design revisions. The XCF32P PROM can store four unique revisions in a single PROM. The XCF16P and XCF08P PROMs can store up to four revisions when they are cascaded. Example revision combinations are described in the following sections.

Single XCF08P/16P/32P PROM Design Revision Options

A single revision is comprised of one or more 8 Mb memory blocks within the PROM(s). The size of the revisions can change each time the PROM is erased and reprogrammed. The smallest possible revision size is a single 8 Mb block. A revision can span over multiple 8 Mb blocks and over PROM boundaries. In addition, a single Platform Flash PROM can have a number of different design revision combinations (Figure 3-1).

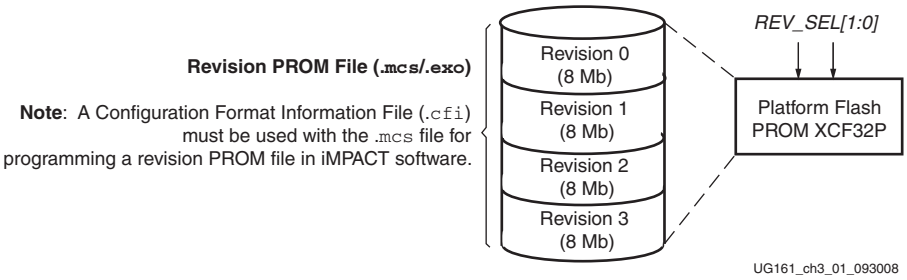


Figure 3-1: **Example XCF32P with Four Design Revisions**

Possible revision combinations for a single Platform Flash PROM are shown in Figure 3-2:

- A single 32 Mb PROM can store four 8 Mb memory blocks and, therefore, can store up to four separate design revisions: one 32 Mb design revision, two 16 Mb design revisions, three 8 Mb design revisions, or four 8 Mb design revisions.
- A single 16 Mb PROM can store two 8 Mb memory blocks and, therefore, can store up to two separate design revisions: one 16 Mb design revision or two 8 Mb design revisions.
- A single 8 Mb PROM can store only one 8 Mb design revision.

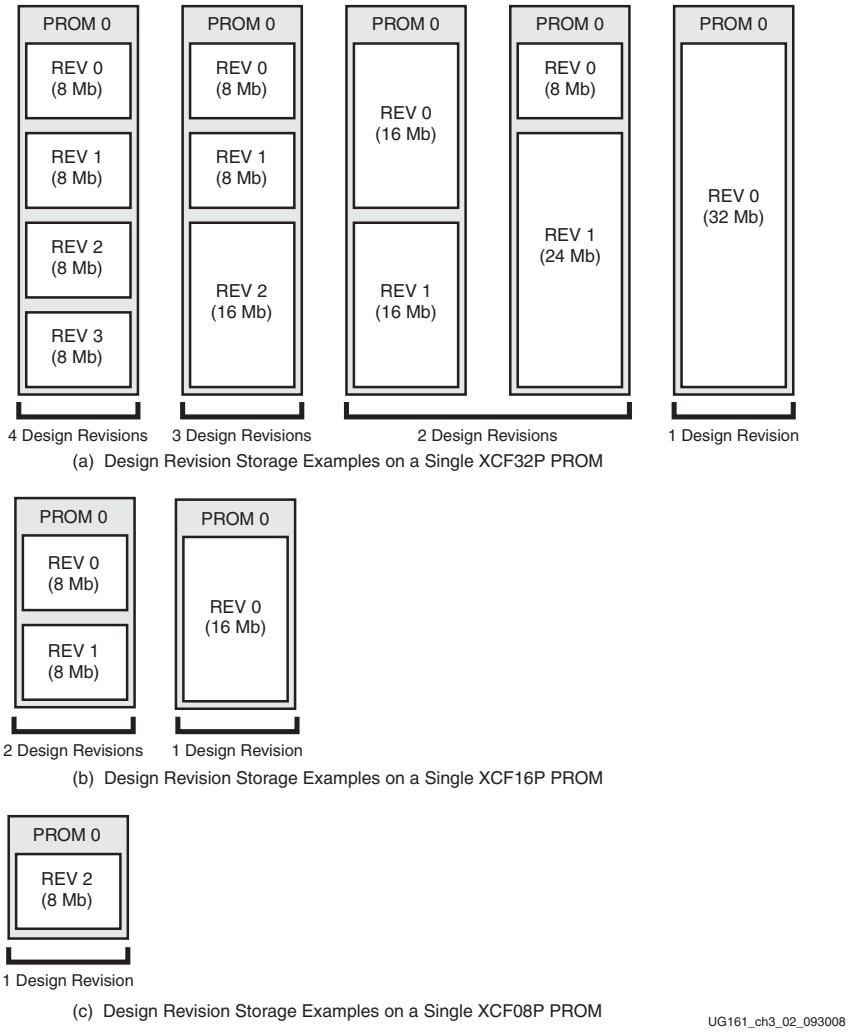


Figure 3-2: **Single-PROM Revision Storage Examples on XCF32P, XCF16P, or XCF08P**

Cascaded XCF08P/16P/32P PROM Design Revision Options

A single design revision can be larger than 32 Mb when using cascaded Platform Flash PROM devices. Larger design revisions can be split over several cascaded Platform Flash PROMs. Any combination of the XCF08P, XCF16P, or XCF32P PROMs can be cascaded together to give the desired density and revision size combinations.

- As shown in Figure 3-3a, two cascaded 32 Mb PROMs can store up to four separate design revisions: one 64 Mb design revision, two 32 Mb design revisions, three 16 Mb design revisions, four 16 Mb design revisions, and so on.
- As shown in Figure 3-3b, when cascading one 16 Mb PROM and one 8 Mb PROM, 24 Mb of space is available. Therefore, up to three separate design revisions can be stored: one 24 Mb design revision, two 8 Mb design revisions, three 8 Mb design revisions, and so on.

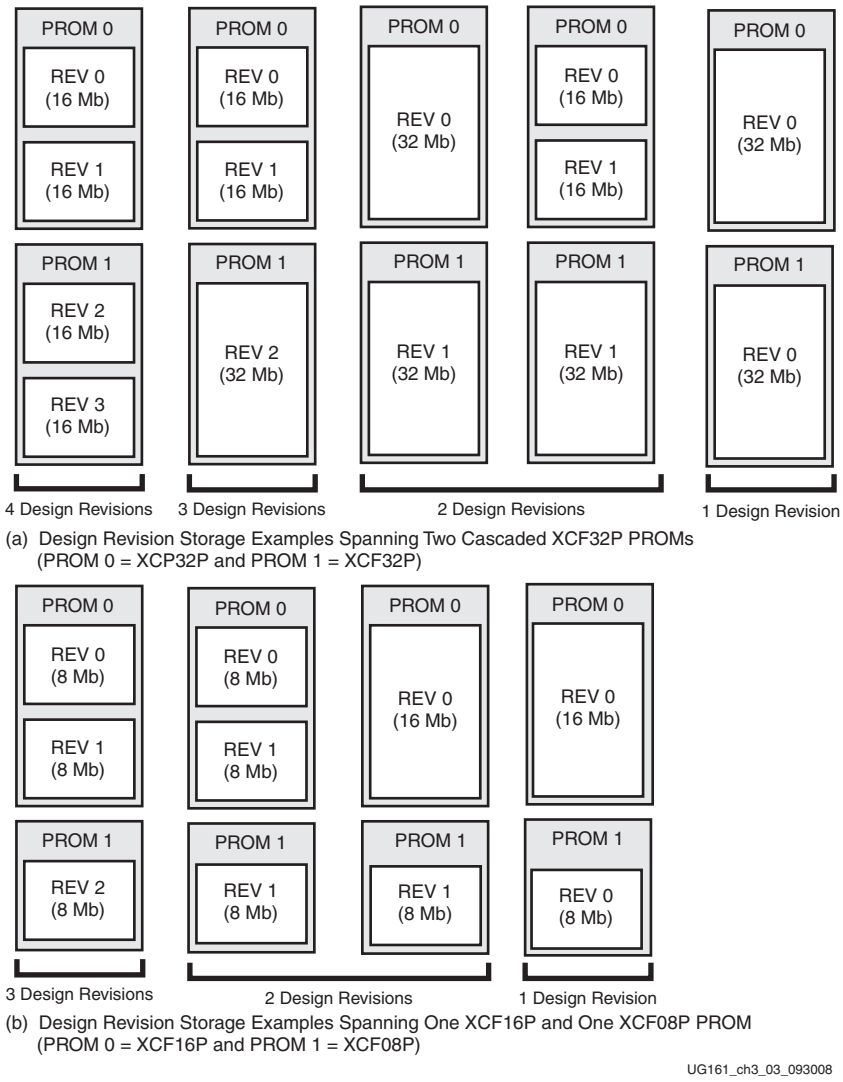


Figure 3-3: **Cascaded Design Revision Storage Examples**

Selecting a Design Revision

The design revision partitioning is handled automatically during file generation in the iMPACT software. During the PROM file (MCS or EXO) creation, each design revision is assigned a revision number:

Revision 0 = `00`, Revision 1 = `01`, Revision 2 = `10`, and Revision 3 = `11`

The iMPACT software also generates the PROM configuration format information (CFI) file, which contains the revision details required to correctly program the design revisions into the Platform Flash PROM. Refer to the "File Definitions" in Chapter 6 for details on the CFI file.

After programming, a particular design revision can be selected using the external REV_SEL[1:0] pins or using the internal programmable design revision control bits. The Enable External Select pin ($\overline{\text{EN\_EXT\_SEL}}$) determines if the external pins or internal bits are used to select the design revision. When the $\overline{\text{EN\_EXT\_SEL}}$ pin is Low, the design revision selection is controlled by the external revision select pins, REV_SEL[1:0]. When

the EN_EXT_SEL pin is High, the design revision selection is determined by the internal programmable revision select control bits. The internal bits are set when the iMPACT software or third-party programmer programs the PROM with the initial design revision data and operational settings (see "Programming a PROM with Design Revisions with iMPACT" in Chapter 6 for details).

When the XCFxxP PROM is used to store multiple revisions for a remote-update application, the general recommendation is to use the external revision select pins to dynamically select the active design revision instead of the internal revision selection control bits. Although internal revision select control bits can be in-system reprogrammed to dynamically select the active design revision, an unexpected interrupt during an in-system change to the internal control bits can leave the internal control bits in an undefined state. Safer dynamic revision selection is accomplished through a fixed revision control sequence applied to the external revision select control pins (see "Safe Update Model," page 34 for a safe control sequence).

During power up, the design revision selection inputs (pins or control bits) are sampled internally based on the EN_EXT_SEL setting. After power-up, when CE is asserted (Low) enabling the PROM inputs, the design revision selection inputs are sampled again after the rising edge of the CF pulse. The data from the selected design revision is then presented on the FPGA configuration interface.

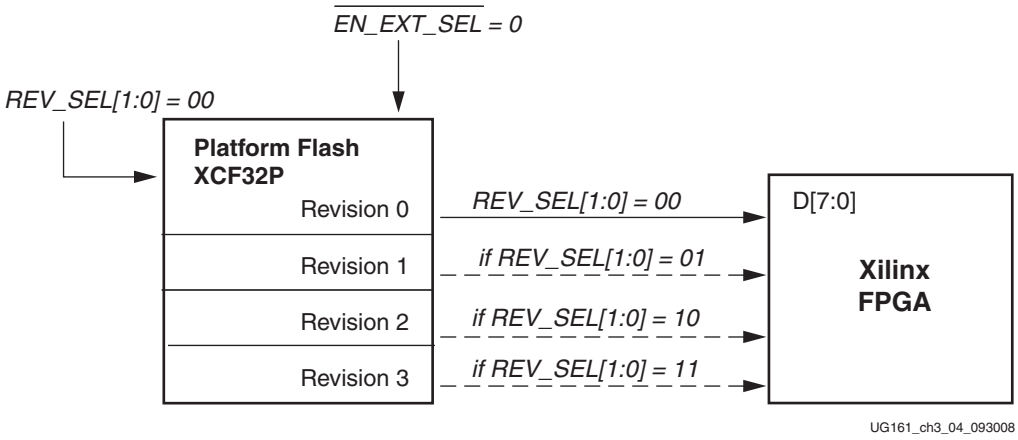


Figure 3-4: **Example Design Revision Selection with Revision 0 Selected by EN_EXT_SEL and Rev_SEL[1:0] Pins**

Design Revision Usage

This section discusses a Safe Update Model and how to reserve multiple design revisions for future updates. The Safe Update Model allows a designer to reserve revisions for future remote updates without affecting the original design revision in the PROM.

Safe Update Model

Many systems require a safe update model to ensure that the system can always be recovered. In the simple case, the safe update setup consists of two revisions: an original failsafe backup design in revision 0 and an updated location in revision 1. The first revision, revision 0, is in a write-protected area of the PROM that can be reverted to if a failure occurs, when updating the second revision with a new board configuration file. An example, fail-safe configuration procedure for an FPGA follows:

1. Attempt FPGA configuration from revision 1 at power-up.
2. Monitor the revision 1 configuration attempt for any failure to successfully configure the FPGA.
3. In the event of a failure, initiate an FPGA reconfiguration from the known-good revision 0.

The ability to update a system with a new design while ensuring that the original pattern is untouched can be accomplished with Platform Flash PROM design revisioning. XAPP972, *Updating a Platform Flash PROM Design Revision In-System Using SVF*, provides a recommended method for implementing a safe update system, details of the safe update programming sequences, and recommendations for update failure recovery.

This section summarizes the steps of a safe update solution.

Step 1:

Create a PROM file (MCS or EXO) and CFI file in the file generation mode of the Xilinx iMPACT software tool. Follow the *Design Revisions* enabled flow described in Chapter 6, "PROM File Creation and Programming Flow." For the simple Safe Update model, the user creates a PROM with two design revisions. Initially, the PROM is programmed with the original pattern in revision 0, and a copy is stored in revision 1 as a placeholder for future updates. In the Safe Update model, the original pattern should be write protected to prevent this design revision from inadvertently being modified. The PROM file is shown in Figure 3-5.

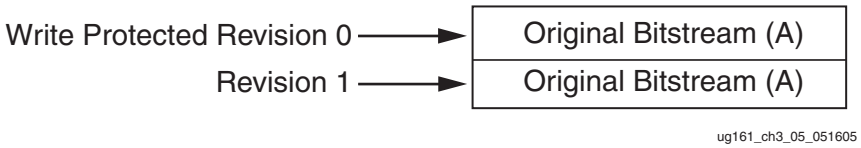


Figure 3-5: **Reserve Revision 1 for Future Updates**

Step 2:

After the number of revisions has been set up and the appropriate address locations defined, the user can then create a new PROM file in the future for a remote update of the revision 1 location. The user again creates a PROM file (MCS or EXO) and CFI file using the iMPACT software with the new bitstream.

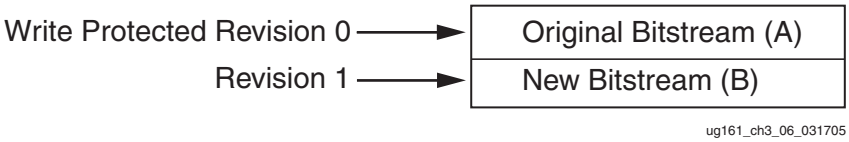


Figure 3-6: **Update Revision 1 with New Design and Maintain Original Version in Revision 0**

Refer to Chapter 6, "PROM File Creation and Programming Flow," for information on the software flow of creating a design-revisioned PROM file. See Appendix A, "References," for reference designs for configuration managers and in-system programming solutions for Platform Flash PROMs.

Reserving Revisions for Future Usage

Multiple revisions can be stored in 8/16/32 Mb Platform Flash PROMs and selected using the revision select pins. Up to four different revisions can be stored in the Platform Flash

PROMs to configure FPGAs in systems to support different standards or markets. This section shows how to manage multiple revisions.

Step 1:

Create an MCS and CFI file pair in the iMPACT software containing the maximum number of revisions planned for the PROM. If only one revision has been completely developed and tested but three more revisions are planned, then create an MCS file with the same bitstream specified for each revision (see Figure 3-7). Program the PROM by selecting program and verify for all the revisions in the iMPACT software. The FPGA(s) can now be configured with bitstream A using any REV_SEL[1:0] pin setting combination on the PROM.

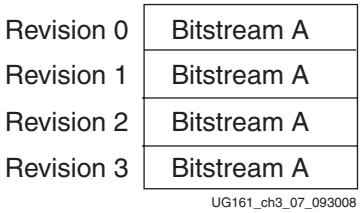


Figure 3-7: **Planning for Multiple Revisions**

Plan for multiple revisions by creating an MCS and CFI file pair with the maximum number of revisions. Temporarily use the same bitstream for the revision selections not yet completed. This serves as a placeholder for future design revision updates.

Step 2:

When the designer wants to update the PROM with a new FPGA bitstream in one of the design revisions, another PROM file (MCS or EXO) and CFI file pair is generated with the same number of planned revisions. This time, it has two unique bitstreams (see Figure 3-8).

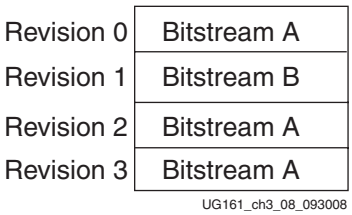


Figure 3-8: **New Bitstream Assigned to Revision 1**

Use bitstream A (or B) for the revision selections 2 and 3 that have not been completed as a temporary placeholder to set the addresses appropriately.

To replace revision 1 in the PROM (shown in Figure 3-8) with bitstream B without erasing and reprogramming the entire PROM use the iMPACT software. In the iMPACT software, right-click on the PROM to program, select the Program action, and select erase, program, and verify for revision 1. See details in Chapter 6, "PROM File Creation and Programming Flow." The PROM now contains bitstreams A and B. Bitstream A is selected by revisions 0, 2, and 3, and bitstream B by PROM revision selection 1.

Step 3:

When all desired FPGA bitstreams have been completed for the multiple design revisions, another PROM file (MCS or EXO) and CFI file pair must be generated (see Figure 3-9).

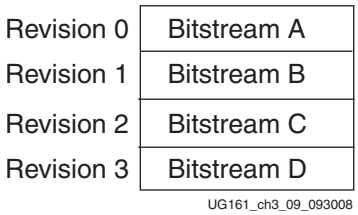


Figure 3-9: **New Bitstreams C and D Assigned to Revision Selections 2 and 3**

Replace revisions 2 and 3 in the PROM with bitstreams C and D, respectively. Right-click on the PROM, select the Program action again, and select revisions 2 and 3 to erase, program, and verify.

Bitstreams used to create the PROM files that update previously programmed bitstreams must be the same PROM block size (in other words, take up the same number of PROM blocks) or an error is generated during programming. The number of PROM blocks required by a bitstream can change if bitstream compression is used or if the FPGAs or number of FPGAs in the hardware has changed.

Using Write Protection

One or more bitstreams can be assigned to a revision and protected against inadvertent overwrite operations by checking the write-protect check box when programming. If write protect is not set, a revision in a PROM can be erased and reprogrammed. These operations are blocked and a warning message is displayed if these operations are attempted on write-protected revisions. A write-protected revision can only be erased if the entire device is erased.

The write-protection feature is useful when subsequent bitstream revision replacement/update operations are interrupted. The FPGAs in the system can always reboot from the write-protected revision(s).

Chapter 4

XCFxxP Decompression and Clock Options

In addition to the revision capability discussed in Chapter 3, XCFxxP Platform Flash PROMs have both an internal oscillator and a built-in decompressor. This chapter discusses these two features.

Why Use the Internal Decompressor or Oscillator?

Platform Flash PROM onboard decompression enables a user to store up to 50% more bits in the Platform Flash PROM. Because the compression ratio is dependent on the FPGA design and can vary between iterations of compiled FPGA bitstreams, PROM data compression is useful only when an FPGA design becomes final and the final FPGA design can be compressed to reduce the PROM size or number of cascaded PROMs on the production board. Xilinx® EasyPath™ FPGAs are prime candidates for compressed bitstreams. For more information on Xilinx EasyPath FPGAs, see:

http://www.xilinx.com/products/easypath/

Note: Encrypted bitstreams are not compressible.

The internal oscillator introduced in the Platform Flash PROM family provides the option to use the PROM as the configuration clock source generator. This oscillator can produce either a 20 MHz or a 40 MHz clock signal to be used for downloading configuration logic from the PROM to the FPGA.

The PROM's internal oscillator, and especially its 40 MHz option, are only appropriate when the PROM decompression feature is required. Otherwise, the FPGA master CCLK or an external oscillator is recommended for driving the PROM's CLK input.

The PROM decompressor outputs data at half the rate of the selected internal clock. In other words, the delivery of a bitstream from the PROM decompressor can require up to double the time as an uncompressed bitstream. If faster configuration is required, the FPGA's **`BitGen -g compress`** method can be used as an alternate compression method to the PROM compression. Combining **`BitGen -g compress`** with PROM compression does not result in a smaller bitstream than PROM compression alone.

Decompression

The XCF08P, XCF16P, and XCF32P Platform Flash PROMs have a built-in data decompressor that works with Xilinx compressed PROM files. PROM files compressed by Xilinx software can fit into a lower density (or lower cost) device. For example, a PROM file originally targeted for an XCF32P PROM can be compressed to fit into an XCF16P PROM. Compression rates vary, depending on several factors, including target device family and design contents. Encrypted bitstreams are not compressible.

Slave Serial and Slave SelectMAP (parallel) modes are supported for FPGA configuration when using the XCFxxP decompression feature. Because the PROM must control both the clock and the data during decompression, FPGA Master configuration modes are not supported.

The PROM decompresses the stored PROM file data before driving both the clock and the data onto the FPGA's configuration interface. When using decompression, the Platform Flash PROM clock output pin (CLKOUT) must be used as the clock signal for the configuration interface, to drive the target FPGA's configuration clock input pin (CCLK). As a controlled clock output, the CLKOUT signal has a reduced maximum frequency and remains Low when decompressed data is not ready. When decompression is enabled, the BUSY input is automatically disabled.

To utilize the XCFxxP decompression feature, the designer must compress the PROM file and program the XCFxxP as the Master device. A compressed PROM file is generated by selecting the Enable Compression option in the iMPACT software's PROM file generation mode. When the Enable Compression option is selected, a CFI file is created with the decompression option setting. The CFI file is read by the iMPACT software to program the decompression setting in the XCFxxP. The designer must also select the Master option in the programming software to ensure the FPGA is in a valid slave configuration mode for decompression. Refer to the "PROM File Creation and Programming Flow" in Chapter 6. Connections details are discussed in "Decompression Setups" in Chapter 5.

Internal Oscillator

An internal oscillator in the XCF08P, XCF16P, and XCF32P Platform Flash PROMs can be used to generate the configuration clock CLKOUT while the PROM supplies data to the FPGA configuration interface. The internal oscillator can be enabled during PROM programming and can be set to either 20 MHz or the default of 40 MHz. The PROM's internal oscillator, and especially its 40 MHz option, are only appropriate when the PROM decompression feature is required. For internal oscillator and related specifications, refer to DS123, *Platform Flash In-System Programmable Configuration PROMs Data Sheet*.

CLKOUT

The High-Density Platform Flash PROMs include the programmable option to enable the CLKOUT signal, which allows the PROM to provide a source synchronous clock aligned to the data on the configuration interface. The CLKOUT signal is derived from one of two clock sources: the CLK input pin or the internal oscillator. The input clock source is selected during the PROM programming sequence. Output data is available before the rising edge of CLKOUT.

The CLKOUT signal is enabled during programming and is active when $\overline{CE}$ is Low and OE/$\overline{RESET}$ is High. When disabled, the CLKOUT pin is put into a high-impedance state and should be pulled High externally to provide a known state.

When cascading Platform Flash PROMs with CLKOUT enabled and after completing its data transfer, the first PROM disables CLKOUT and drives the $\overline{CEO}$ pin Low, enabling the next PROM in the PROM chain. The next PROM begins driving the CLKOUT signal after that PROM is enabled and data is available for transfer. An external pull-up must be tied to CLKOUT to keep CLKOUT High during the transition period between cascaded PROMs.

The CLKOUT pin is recommended for use only when decompression is required. Example connections are shown in "Decompression Setups" in Chapter 5.

The CLKOUT pin is functionally an output pin with a special design consideration. When CLKOUT function is enabled, the derived clock signal from the selected clock source is driven out of the PROM through the CLKOUT pin for use as the configuration clock. The signal from the CLKOUT pin is returned for use by some internal PROM circuits. The board trace must be carefully designed to prevent reflections from causing glitches at the PROM CLKOUT pin that can propagate to the internal PROM circuits.

Chapter 5

Advanced and High-Speed Configuration Setups

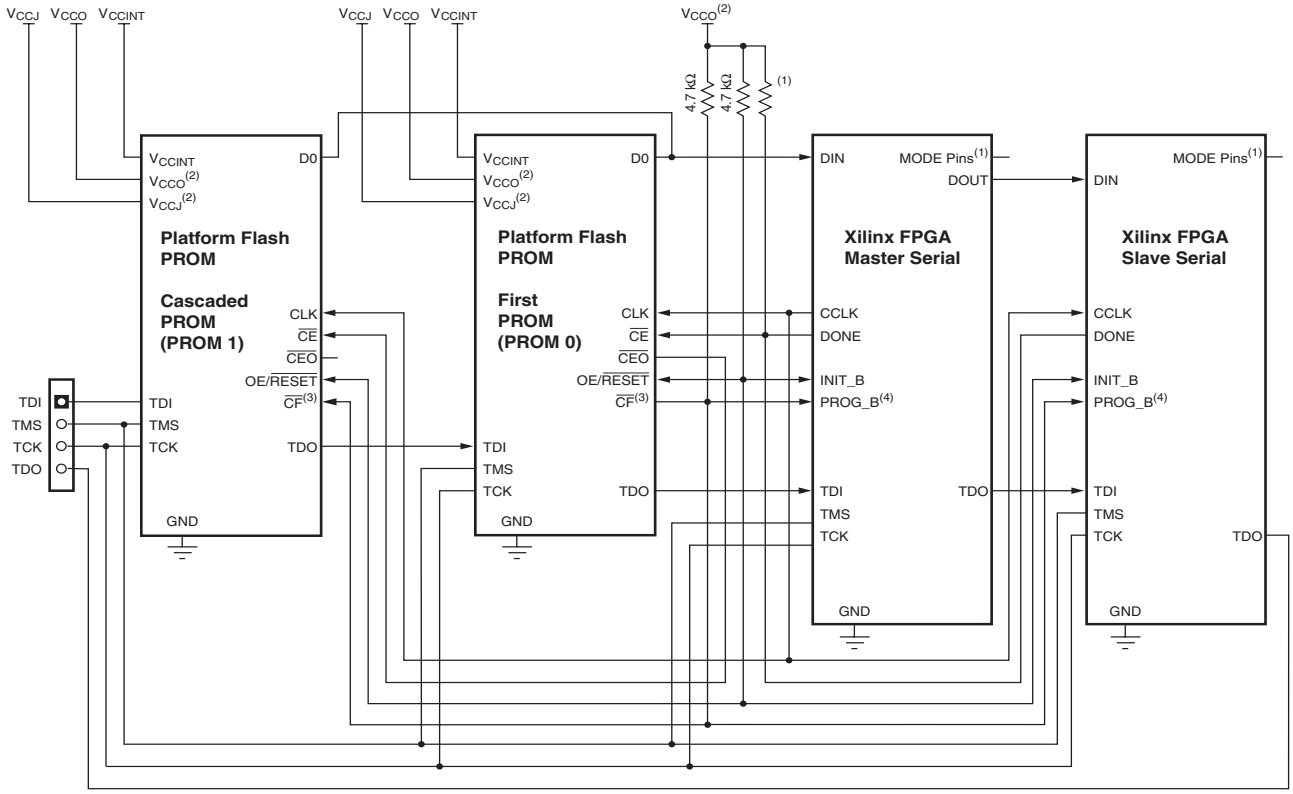
This chapter discusses advanced configuration setups available with the Platform Flash PROM family. In addition to high-speed configuration options, XCFxxP PROM members have a number of new features that allow a great deal of configuration setup flexibility. Each configuration setup is shown with a table describing required pin connections.

For details on PROM file generation, as well as file description details associated with setups that use the rich XCFxxP feature set (Design Revisions, CLK_OUT, or Decompression), refer to "Creating an (XCFxxP) Advanced PROM File with iMPACT Software" in Chapter 6.

Refer to Chapter 7, "Design Considerations," to ensure a robust configuration design. The information in the Design Considerations chapter supplements the recommendations provided in this chapter.

Multiple Device Serial Configuration Setup

The XCFxxS/XCFxxP devices both support the Serial configuration mode. These PROMs can be cascaded together to program a daisy-chain of Xilinx® FPGA devices. Figure 5-1 shows this advanced Serial configuration setup (see also Table 5-1).



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. For the XCFxxS, the $\overline{CF}$ pin is an output pin. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
4. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_01_091509

Figure 5-1: **Configuration Setup – Multiple FPGAs in FPGA Master/Slave Serial Mode**

Table 5-1: **Pin Connections for Configuring Multiple FPGAs in FPGA Master/Slave Serial Mode**

Pin	Connect to	Comment
XCFxxS and XCFxxP Pins		
D0	FPGA DIN	PROM and FPGA data signal.
CLK	FPGA CCLK	FPGA-generated clock signal.
CE	FPGA DONE	Single PROM or first PROM in a cascaded PROM chain. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$ /INIT_B	Pull-up resistor to V_{CCO} is required.

Table 5-1: **Pin Connections for Configuring Multiple FPGAs in FPGA Master/Slave Serial Mode *(Cont'd)***

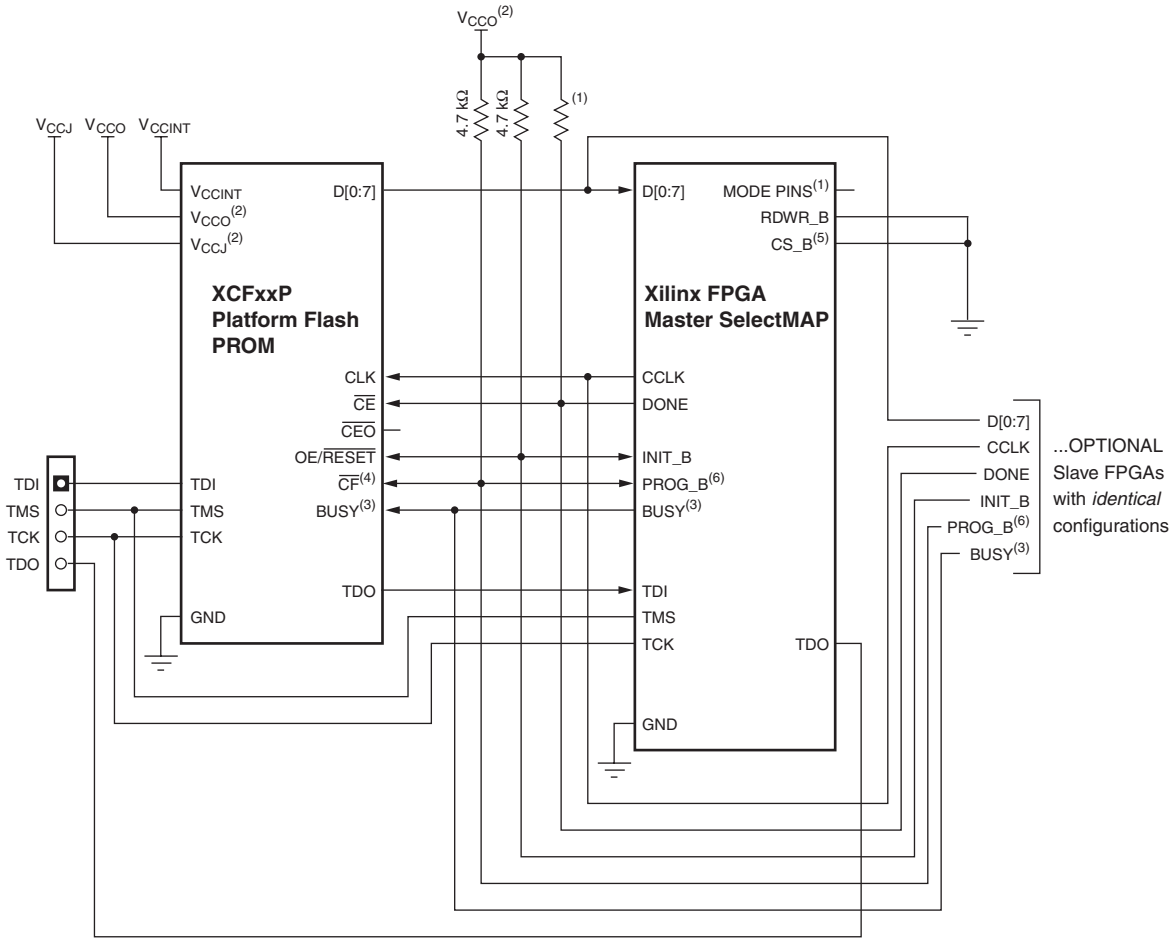
Pin	Connect to	Comment
CF	FPGA PROG_B/PROGRAM_B/ $\overline{\text{PROGRAM}}$	XCFxxS: Open-drain output. XCFxxP: Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	3.3V	XCFxxS 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
	1.8V	XCFxxP 0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5V	XCFxxS 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
	3.3/2.5/1.8V	XCFxxP 0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	XCFxxS 0.1 µF decoupling capacitor to GND.
	3.3/2.5V	XCFxxP 0.1 µF decoupling capacitor to GND.
GND	Ground	All ground (GND) pins must be connected.
XCFxxP Additional Pins		
D[1:7]	DNC	Parallel data option is not used.
BUSY	DNC	Parallel data option is not used.
CLKOUT	DNC	CLKOUT option is not used.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.

XCFxxP High-Speed SelectMAP (Parallel) Mode Setups

The XCFxxP members of the Platform Flash PROM family support the Xilinx FPGA byte wide configuration modes: SelectMAP mode for the Virtex® families and the parallel modes for the Spartan® FPGA families. This mode allows for faster throughput and configuration time. Three common high-speed setups are shown in this section with the necessary pin connections for each mode described.

FPGA Master SelectMAP (Parallel) Mode Setup

This setup shows a single Platform Flash PROM (XCFxxP) targeting a single Xilinx FPGA in Master SelectMAP (parallel) mode. This setup utilizes the Xilinx FPGA internal configuration clock source. See Figure 5-2 and Table 5-2.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high-frequency SelectMAP mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
5. CS_B is named CSI_B in some FPGA families.
6. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_02_091509

Figure 5-2: **XCFxxP High-Speed Setup - FPGA Master SelectMAP (Parallel) Mode**

Table 5-2: **XCFxxP Pin Connections for Configuring in FPGA Master SelectMAP (Parallel) Mode**

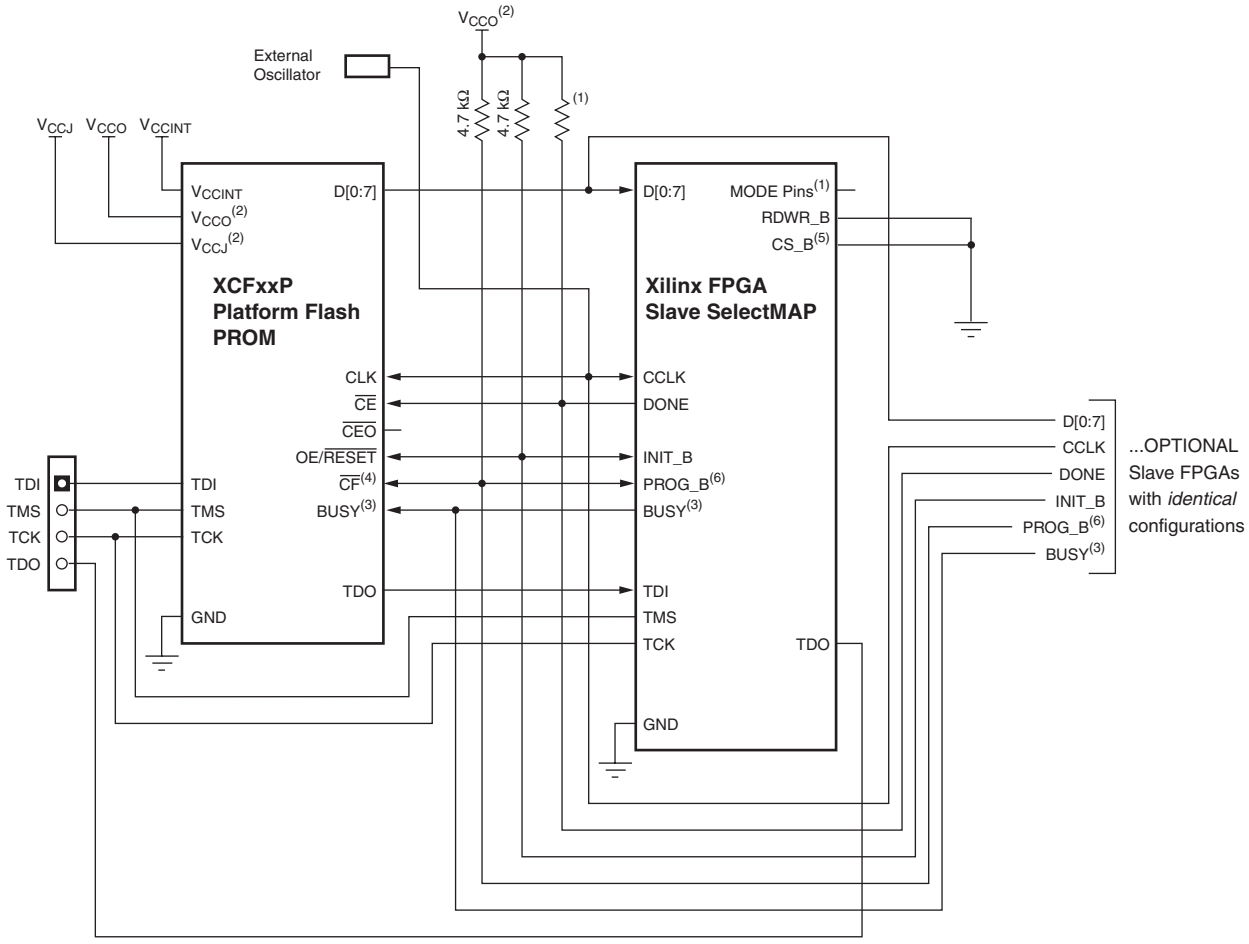
XCFxxP Pins	Connect to	Comment
D[0:7]	FPGA D[0:7]	PROM and FPGA data signal.
CLK	FPGA CCLK	FPGA-generated clock signal.
CLKOUT	DNC	CLKOUT option is not used.
CE	FPGA DONE	Single PROM or first PROM in a cascaded PROM chain. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.

Table 5-2: **XCFxxP Pin Connections for Configuring in FPGA Master SelectMAP (Parallel) Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	FPGA BUSY	Required for high-frequency SelectMap (parallel) mode. Refer to the FPGA data sheet.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.

FPGA Slave SelectMAP (Parallel) Mode Setup

This setup consists of a single high-density Platform Flash PROM (XCFxxP) targeting a single Xilinx FPGA in Slave SelectMAP (parallel) mode. This setup utilizes an external oscillator as a configuration clock source. See Figure 5-3 and Table 5-3.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high-frequency SelectMAP mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
5. CS_B is named CSI_B in some FPGA families.
6. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_03_091509

Figure 5-3: **XCFxxP High-Speed Setup – FPGA Slave SelectMAP (Parallel) Mode**

Table 5-3: **XCFxxP Pin Connections for Configuring in FPGA Slave SelectMAP (Parallel) Mode**

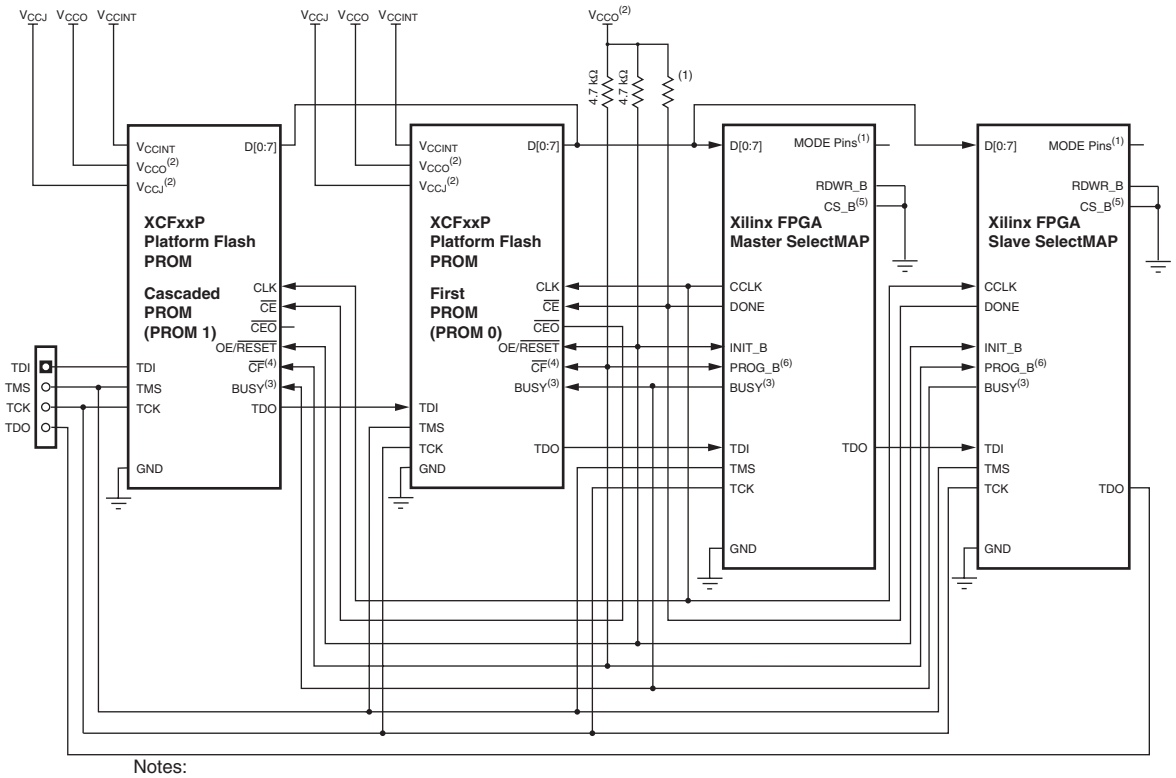
XCFxxP Pins	Connect to	Comment
D[0:7]	FPGA D[0:7]	PROM and FPGA data signals.
CLK	FPGA CCLK and External Oscillator	Configuration clock.
CLKOUT	DNC	CLKOUT option is not used.

Table 5-3: **XCFxxP Pin Connections for Configuring in FPGA Slave SelectMAP (Parallel) Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CE	FPGA DONE	Single PROM or first PROM in a cascaded PROM chain. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	FPGA BUSY	Required for high-frequency SelectMap (parallel) mode configuration. Refer to the FPGA data sheet.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.

FPGA Master/Slave SelectMAP Mode for Multiple FPGAs with Identical Patterns

This setup consists of a single high-density Platform Flash PROM (XCFxxP) targeting multiple Xilinx FPGAs in Master/Slave SelectMAP (parallel) mode. This setup utilizes the Master FPGA's internal oscillator as a configuration clock source. See Figure 5-4 and Table 5-4.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high-frequency SelectMAP mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
5. CS_B is named CSI_B in some FPGA families.
6. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_04_091509

Figure 5-4: **XCFxxP High-Speed Configuration Setup – Multiple FPGA Devices in Master/Slave SelectMAP (Parallel) Mode with Identical Patterns**

Table 5-4: **XCFxxP Pin Connections for Configuring Multiple FPGA Devices with Identical Patterns in Master/Slave SelectMAP (Parallel) Mode**

XCFxxP Pins	Connect to	Comment
D[0:7]	All FPGA D[0:7]	PROM and FPGA data signals.
CLK	All FPGA CCLK	FPGA-generated clock signal.
CLKOUT	DNC	CLKOUT option is not used.
CE	All FPGA DONE	Single PROM or first PROM in a cascaded PROM chain. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.

Table 5-4: **XCFxxP Pin Connections for Configuring Multiple FPGA Devices with Identical Patterns in Master/Slave SelectMAP (Parallel) Mode *(Cont'd)***

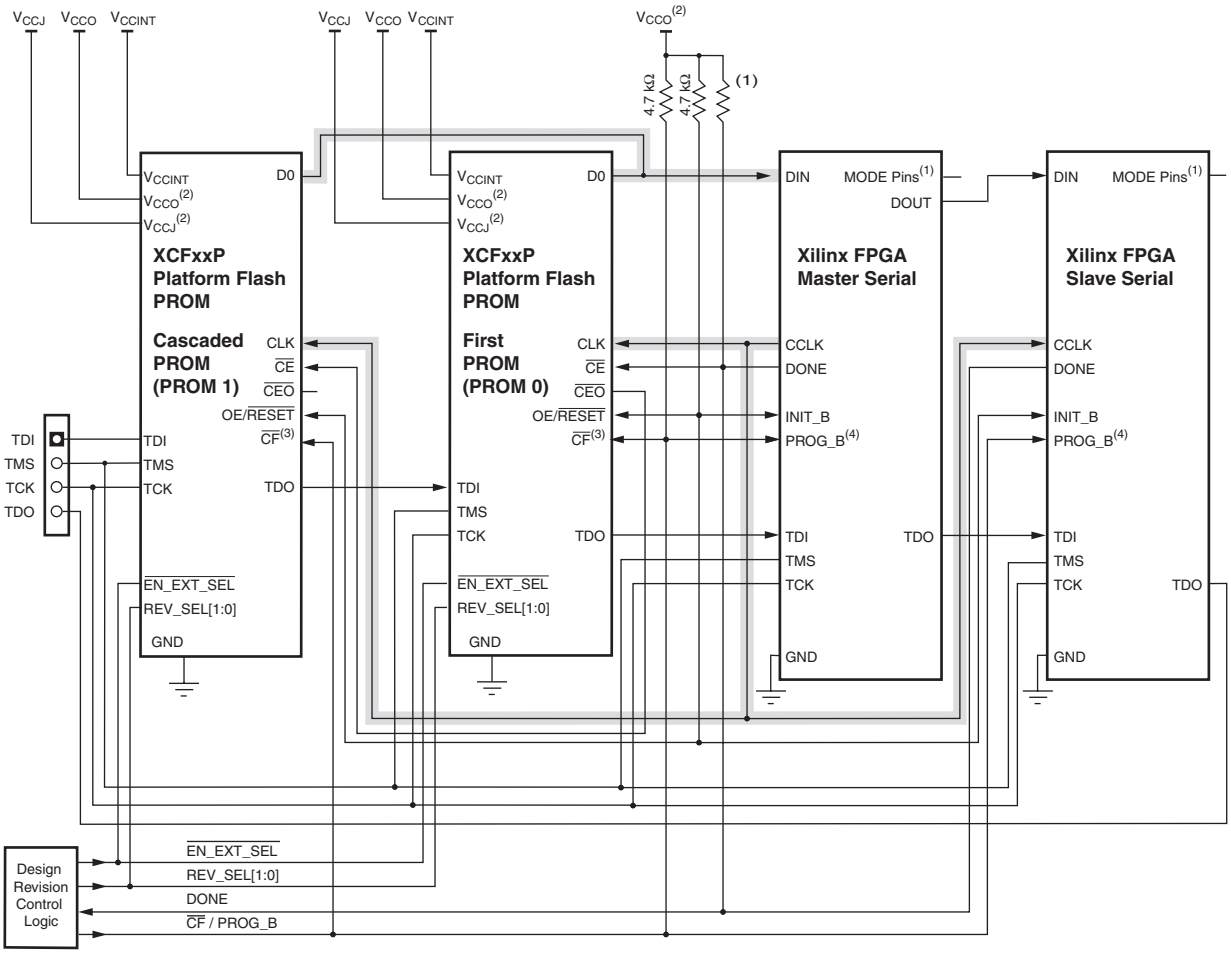
XCFxxP Pins	Connect to	Comment
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	All FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	All FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	All FPGA BUSY	Required for high-frequency SelectMap (parallel) mode configuration. Refer to the FPGA data sheet.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.

XCFxxP Advanced Feature Configuration Setups

The XCFxxP members have an advanced feature set that enables new configuration setups. The new features for these family members are: design revisions, an internal oscillator, and decompression capability. This section shows the configuration setups and pin connections required for using the advanced options.

Design Revision Setups

Four setups for using the design revision features in the Platform Flash PROM are described in the following sections. Setups for both Serial and SelectMAP (parallel) configuration modes are shown. See Figure 5-5 through Figure 5-8 and Table 5-5 through Table 5-8.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
4. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_05_091509

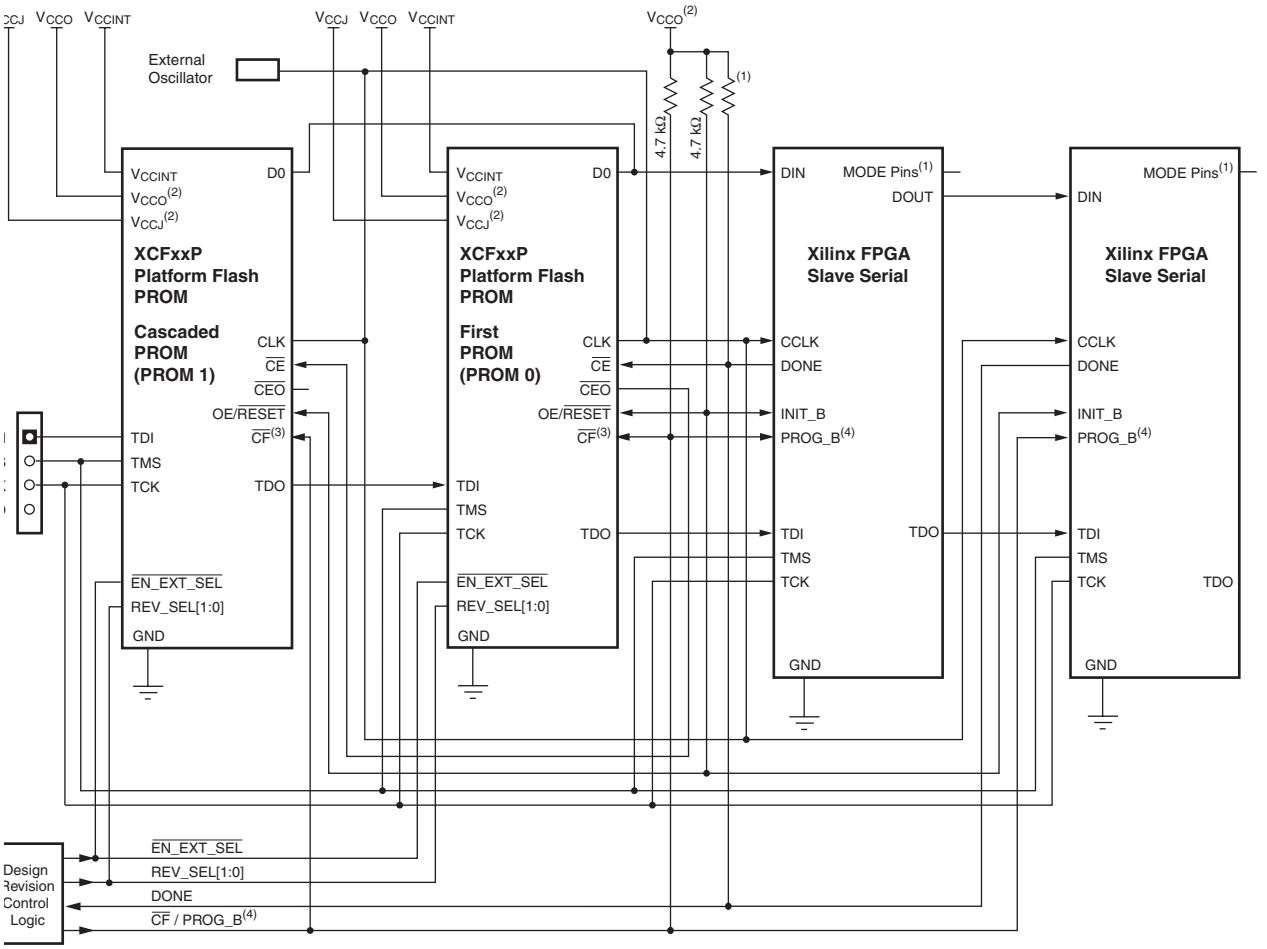
Figure 5-5: **XCFxxP Advanced Setup – Multiple PROMs with Design Revisions with FPGAs in Master/Slave Serial Mode**

Table 5-5: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGAs in Master/Slave Serial Mode**

XCFxxP Pins	Connect to	Comment
D0	First FPGA DIN	PROM and FPGA data signal.
D[1:7]	DNC	Parallel data option is not used.
CLK	All FPGA CCLK	Configuration clock.
CLKOUT	DNC	CLKOUT option is not used.

Table 5-5: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGAs in Master/Slave Serial Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CE	All FPGA DONE	First PROM. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	All FPGA $\overline{INIT}$ /INIT_B	Pull-up resistor to V_{CCO} is required.
CF	All FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	DNC	Parallel data option is not used.
EN_EXT_SEL	Design Revision (Ext /Int) Control Logic	`0` = Revision Control selection by external REV_SEL pins. `1` = Revision Control selection by internal REV_SEL bits.
REV_SEL[1:0]	Design Revision Selection Control Logic	Revision 0 = `00`, Revision 1 = `01` Revision 2 = `10`, Revision 3 = `11`
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
4. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_06_091509

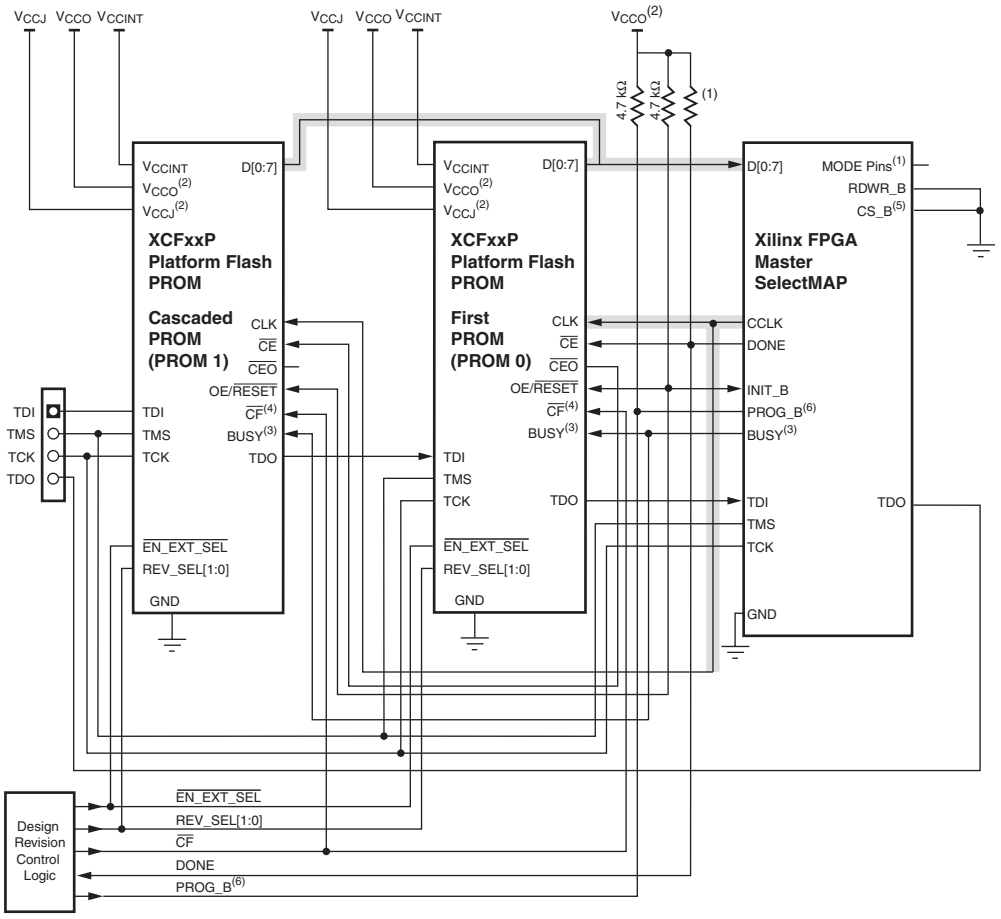
Figure 5-6: **XCFxxP Advanced Setup – Multiple PROMs with Design Revisions with FPGAs in Slave Serial Mode**

Table 5-6: **XCFxxP Pin Connections for Configuring Multiple PROMs with Design Revisions with FPGAs in Slave Serial Mode**

XCFxxP Pins	Connect to	Comment
D0	First FPGA DIN	PROM and FPGA data signal.
D[1:7]	DNC	Parallel data option is not used.
CLK	All FPGA CCLK and External Oscillator	Configuration clock.
CLKOUT	DNC	CLKOUT option is not used.

Table 5-6: **XCFxxP Pin Connections for Configuring Multiple PROMs with Design Revisions with FPGAs in Slave Serial Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CE	All FPGA DONE	First PROM. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	All FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	All FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	DNC	Parallel data option is not used.
EN_EXT_SEL	Design Revision (Ext /Int) Control Logic	`0` = Revision Control selection by external REV_SEL pins. `1` = Revision Control selection by internal REV_SEL bits.
REV_SEL[1:0]	Design Revision Selection Control Logic	Revision 0 = `00`, Revision 1 = `01` Revision 2 = `10`, Revision 3 = `11`
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high-frequency SelectMAP (Parallel) mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. For the XCFxxP, the CF pin is a bidirectional pin.
5. CS_B is named CSI_B in some FPGA families.
6. PROG_B is named PROGRAM_B or PROGRAM in some FPGA families.

UG161_c5_07_091509

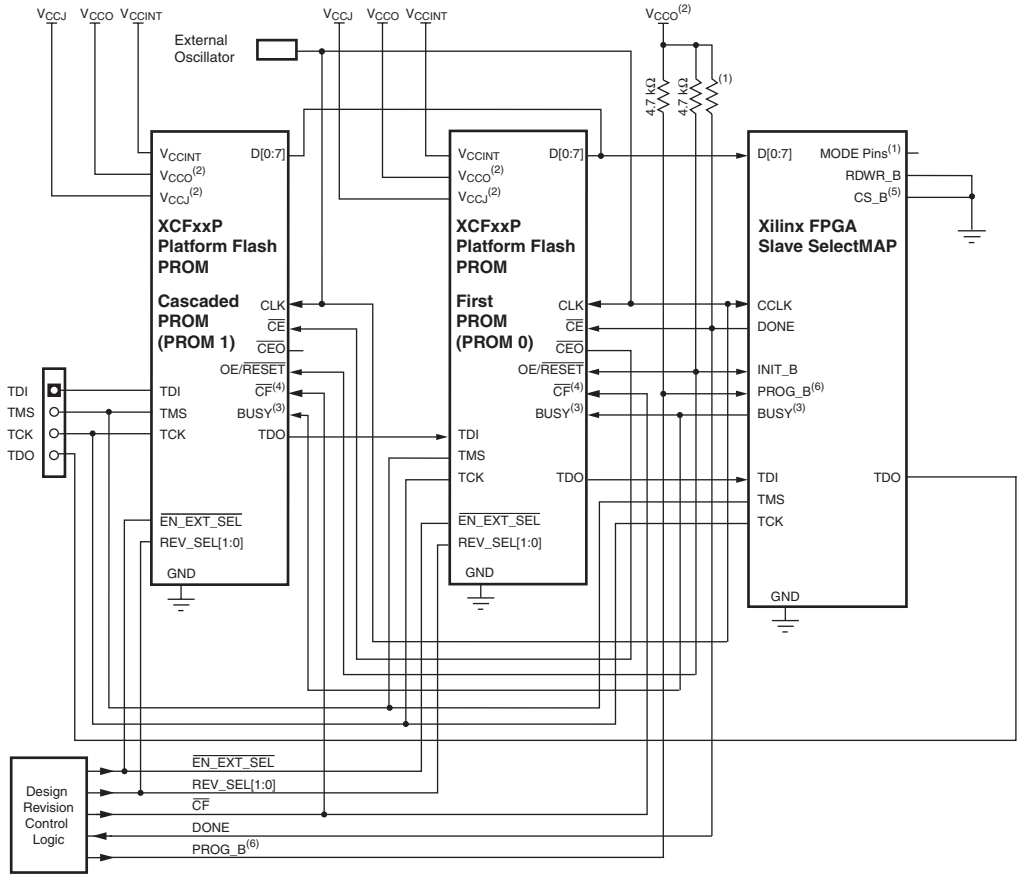
Figure 5-7: **XCFxxP Advanced Setup – Multiple PROMs with Design Revisions with FPGA in Master SelectMAP (Parallel) Mode**

Table 5-7: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGA in Master SelectMAP (Parallel) Mode**

Pins	Connect to	Comment
D[0:7]	FPGA D[0:7]	PROM and FPGA data signals.
CLK	FPGA CCLK	FPGA-generated clock signal.
CLKOUT	DNC	CLKOUT option is not used.

Table 5-7: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGA in Master SelectMAP (Parallel) Mode *(Cont'd)***

Pins	Connect to	Comment
CE	FPGA DONE	First. PROM. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	FPGA BUSY	Required for high-frequency SelectMap (parallel) mode configuration. Refer to the FPGA data sheet.
EN_EXT_SEL	Design Revision (Ext /Int) Control Logic	`0` = Revision Control selection by external REV_SEL pins. `1` = Revision Control selection by internal REV_SEL bits
REV_SEL[1:0]	Design Revision Selection Control Logic	Revision 0 = `00`, Revision 1 = `01` Revision 2 = `10`, Revision 3 = `11`
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.



Notes:

1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high frequency SelectMAP mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
5. CS_B is named CSI_B in some FPGA families.
6. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_c5_08_091509

Figure 5-8: **XCFxxP Advanced Setup - Multiple PROMs with Design Revisions with FPGA in Slave SelectMAP (Parallel) Mode**

Table 5-8: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGA in Slave SelectMAP (Parallel) Mode**

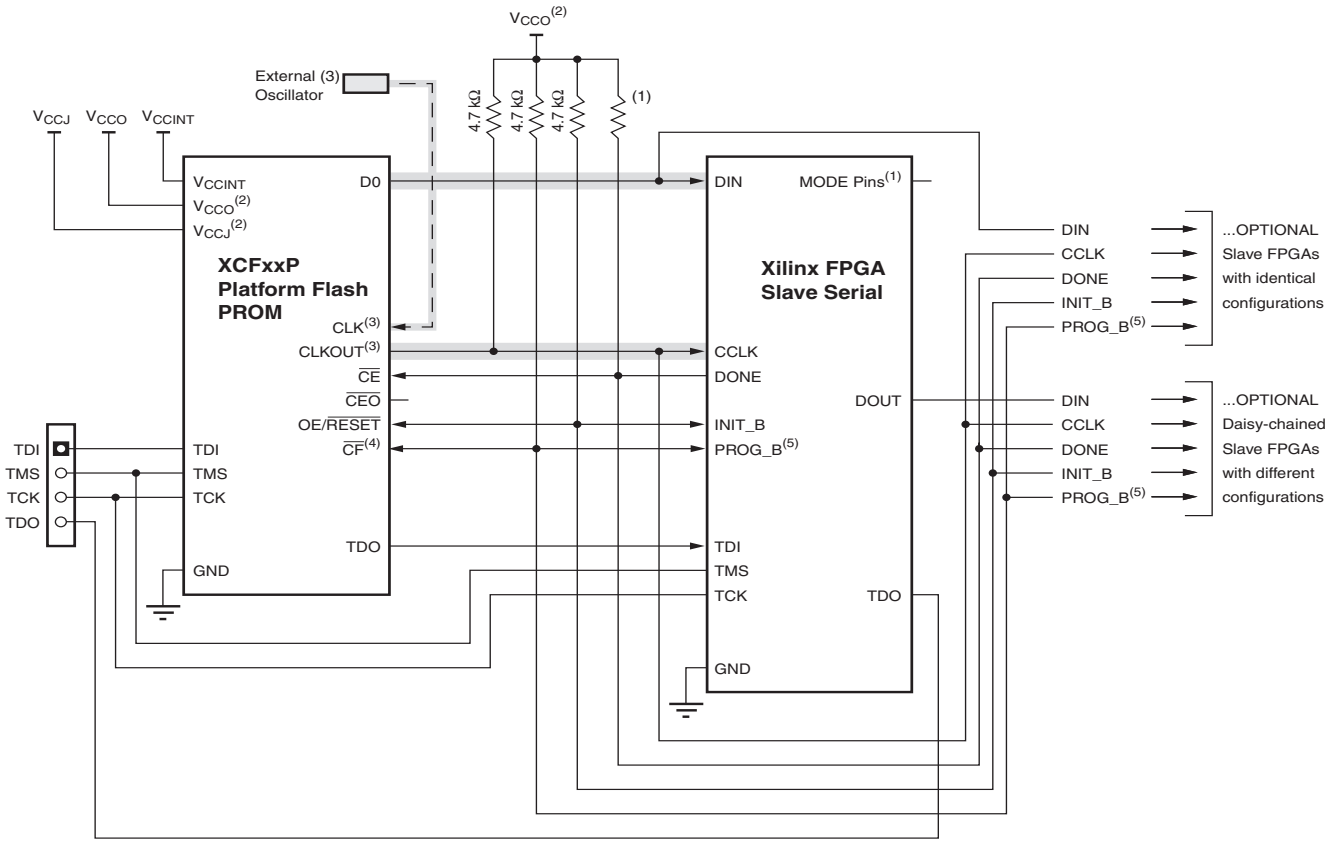
XCFxxP Pins	Connect to	Comment
D[0:7]	FPGA D[0:7]	PROM and FPGA data signals.
CLK	FPGA CCLK and External Oscillator	Configuration clock.
CLKOUT	DNC	CLKOUT option is not used.

Table 5-8: **XCFxxP Pin Connections for Multiple PROMs with Design Revisions with FPGA in Slave SelectMAP (Parallel) Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CE	FPGA DONE	First PROM. See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$/INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	FPGA BUSY	Required for high-frequency SelectMap (parallel) mode configuration. Refer to the FPGA data sheet.
EN_EXT_SEL	Design Revision (Ext /Int) Control Logic	0 = Revision Control selection by external REV_SEL pins. 1 = Revision Control selection by internal REV_SEL bits.
REV_SEL[1:0]	Design Revision Selection Control Logic	Revision 0 = 00, Revision 1 = 01 Revision 2 = 10, Revision 3 = 11
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.

Decompression Setups

The XCFxxP devices have a new decompressor feature. Only Slave Serial and Slave SelectMAP (parallel) configuration modes are supported for FPGA configuration when using an XCFxxP PROM programmed with a compressed bitstream. The Platform Flash PROM CLKOUT signal must be used for compression. See Figure 5-9 and Table 5-10.



Notes:
1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. When decompression is used, the decompressed data from the XCFxxP Platform Flash PROM must be clocked by CLKOUT. The CLKOUT signal can be sourced from the XCFxxP PROM's internal oscillator, or optionally, sourced from an external clock driving the CLK input pin. CLKOUT must be tied to a 4.7 KΩ resistor pulled up to V_{CCO}. If CLK is not used as an input, then CLK should be tied to ground.
4. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
5. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_09_091509

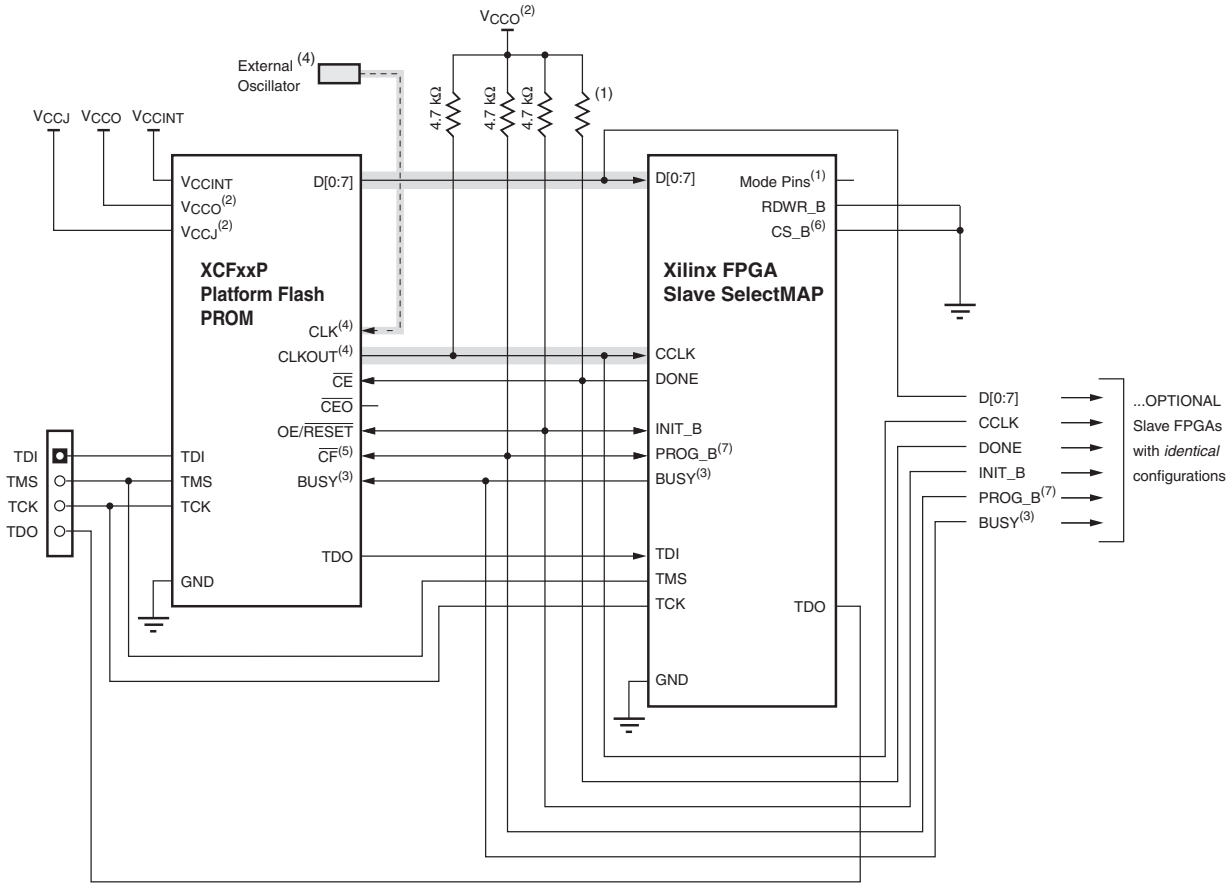
Figure 5-9: **XCFxxP Decompression Setup – FPGA Slave Serial Mode**

Table 5-9: **XCFxxP Pin Connections for Configuring with Decompression in FPGA Slave Serial Mode**

XCFxxP Pins	Connect to	Comment
D0	FPGA DIN	PROM and FPGA data signals.
D[1:7]	DNC	Parallel data option is not used.
CLK	External CLK	If CLKOUT with external clock option is used.
	GND	If CLKOUT with internal oscillator option is used.
CLKOUT	FPGA CCLK	CLKOUT is required for decompression. Pull-up resistor to V_{CCO} is required.

Table 5-9: **XCFxxP Pin Connections for Configuring with Decompression in FPGA Slave Serial Mode**

XCFxxP Pins	Connect to	Comment
CE	FPGA DONE	See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$ /INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	DNC	Parallel data option is not used.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.



Notes:

1. For Mode pin connections and DONE pin pull-up values, refer to the appropriate FPGA data sheet.
2. For compatible voltages, refer to the appropriate data sheet.
3. The BUSY pin is only available with the XCFxxP Platform Flash PROM, and the connection is only required for high frequency SelectMAP mode configuration. For BUSY pin requirements, refer to the appropriate FPGA data sheet.
4. When decompression is used, the decompressed data from the XCFxxP Platform Flash PROM must be clocked by CLKOUT. The CLKOUT signal can be sourced from the XCFxxP PROM's internal oscillator, or optionally, sourced from an external clock driving the CLK input pin. CLKOUT must be tied to a 4.7 KΩ resistor pulled up to V_{CCO}. If CLK is not used as an input, then CLK should be tied to ground.
5. For the XCFxxP, the $\overline{CF}$ pin is a bidirectional pin.
6. CS_B is named CSI_B in some FPGA families.
7. PROG_B is named PROGRAM_B or $\overline{PROGRAM}$ in some FPGA families.

UG161_ch5_10_091509

Figure 5-10: **XCFxxP Decompression Setup – FPGA Slave SelectMAP Mode**

Table 5-10: **XCFxxP Pin Connections for Configuring with Decompression in FPGA Slave SelectMAP Mode**

XCFxxP Pins	Connect to	Comment
D[0:7]	FPGA D[0:7]	PROM and FPGA data signals.
CLK	External CLK	If CLKOUT with external clock option is used.
	GND	If CLKOUT with internal oscillator option is used.
CLKOUT	FPGA CCLK	CLKOUT option is required for decompression. Pull-up resistor to V_{CCO} is required.

Table 5-10: **XCFxxP Pin Connections for Configuring with Decompression in FPGA Slave SelectMAP Mode *(Cont'd)***

XCFxxP Pins	Connect to	Comment
CE	FPGA DONE	See the FPGA data sheet for the recommended value for the DONE pull-up resistor to V_{CCO}. Do not directly drive the DONE LED status indicator.
	$\overline{CEO}$ of previous PROM in cascade chain	Cascaded PROM.
CEO	DNC	No cascaded PROM.
	$\overline{CE}$ of next PROM in cascade chain	Cascaded PROM.
OE/$\overline{RESET}$	FPGA $\overline{INIT}$ /INIT_B	Pull-up resistor to V_{CCO} is required.
CF	FPGA PROG_B/PROGRAM_B/ $\overline{PROGRAM}$	Bidirectional pin has open-drain output. Pull-up resistor to V_{CCO} is required.
BUSY	FPGA BUSY	Required for high-frequency SelectMap (parallel) mode configuration. Refer to the FPGA data sheet.
EN_EXT_SEL	DNC	Revision Control is not used.
REV_SEL[1:0]	DNC	Revision Control is not used.
TMS	JTAG TMS	JTAG mode select.
TCK	JTAG TCK	JTAG test clock.
TDI	JTAG CHAIN	JTAG serial data input.
TDO	JTAG CHAIN	JTAG serial data output.
V_{CCINT}	1.8V	0.1 µF decoupling capacitor to GND. All V_{CCINT} pins must be connected.
V_{CCO}	3.3/2.5/1.8V	0.1 µF decoupling capacitor to GND. All V_{CCO} pins must be connected.
V_{CCJ}	3.3/2.5V	0.1 µF decoupling capacitor to GND.
GND	Ground	All GND pins must be connected.

Chapter 6

PROM File Creation and Programming Flow

File Definitions

This chapter discusses the file extensions, the file generation flows, and programming steps required when targeting the Platform Flash PROM family. Figure 6-1 shows the three files typically associated with PROM file creation: the input bitstream (BIT), the output data (MCS), and the configuration format information (CFI) files. The CFI contains the output PROM information. Table 6-1 lists the usage of each file.

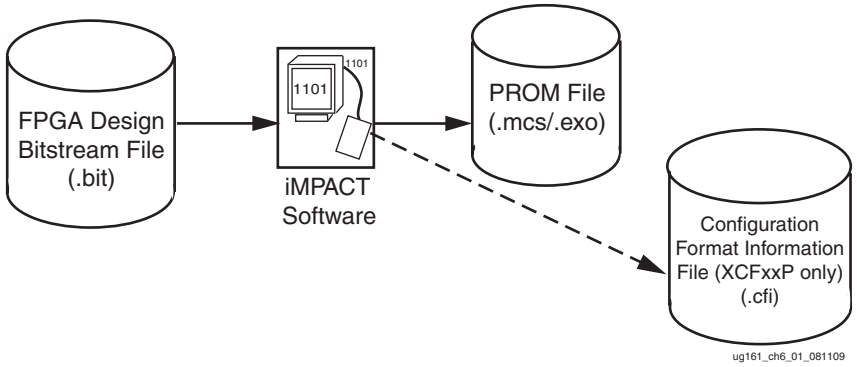


Figure 6-1: **Platform Flash PROM File Generation Flow**

Table 6-1: **File Extension Definitions**

File Extension	File Description
`.bit`	Bitstream binary design data file for each FPGA.
`.mcs/.exo`	PROM data file, industry standard format required for PROM programmers.
`.cfi`	Configuration format information file. The CFI file is generated with the same root name as the MCS or EXO file. The CFI file contains auxiliary information to enable the programming of special PROM options, including design revisions, CLKOUT, and decompression features. The revision start and end addresses are specified in the CFI file.

PROM File and Configuration Format Information File Usage

A standard PROM file is required for programming the Platform Flash PROMs and is generated from the design bitstream(s) for Xilinx® FPGAs. The PROM file is the only file

required for programming a Platform Flash PROM XCFxxS. The PROM file is the only file required for programming the higher density XCFxxP Platform Flash PROMs if they do not utilize any of the advanced features (Design Revisions, CLKOUT, or decompression).

An XCFxxP PROM utilizing the advanced features (design revisions, CLKOUT, or decompression) uses the PROM file (MCS or EXO) and an associated CFI file to program the PROM using iMPACT software or third-party programmers. The CFI file is automatically generated by iMPACT software when the advanced file generation flow is used. This flow is discussed in the following sections.

What Is a Configuration Format Information File?

The CFI file contains specific revision information used by iMPACT software during in-system erase, program, and verify operations. The CFI file is used to enable the XCFxxP internal clock (CLKOUT) and the decompression feature for programming.

The configuration format information file in Figure 6-2 gives the start and end blocks for each design revision used. In this example, there are four revisions, each span a single 8 Mb block which is indicated by the END statement.

```
# PROMGen: Xilinx PROM Generator
# Copyright © 1995-2005 Xilinx, Inc. All rights reserved.
DATE   07/01/2005 -11:05
SOURCE   revision.mcs
DEVICE   XCF32P
[Signature 0x########]
DEFAULT VERSION 0
BLOCK 0 VERSION 0 [END] [COMPRESSED] [0x########] [DCM|DCI]
BLOCK 1 VERSION 1 [END] [COMPRESSED] [0x########] [DCM|DCI]
BLOCK 2 VERSION 2 [END] [COMPRESSED] [0x########] [DCM|DCI]
BLOCK 3 VERSION 3 [END] [COMPRESSED] [0x########] [DCM|DCI]
```

ug161_ch6_03_080305

Figure 6-2: **Example CFI Generated by iMPACT Software**

The settings in the CFI file enclosed in brackets are optional. The optional settings include the SIGNATURE, DEFAULT VERSION, END, COMPRESSED, and DCM/DCI records. The SIGNATURE option allows the file to be used in place of the SIG file. The DEFAULT VERSION statement is another optional setting used to indicate the default design revision that is accessed at power-up. The internal default design revision selection is overridden by using the EN_EXT_SEL pin to enable the REV_SEL[1:0] pins. The [END] keyword indicates the last block a revision occupies. If the PROM is compressed, each block shows the [COMPRESSED] keyword in the CFI file. This keyword indicates to the programming software that the decompression setting in the PROM must be programmed. The [0x########] option represents the cumulative checksum for each revision. The [DCM|DCI] keyword is used when the associated bitstream for a revision has a Match_Cycle or Lock_Cycle set to anything other than No_Wait. This option indicates to the programming software that a free running clock option should be set during programming when using the CLKOUT feature.

If a design revision spanned two 8 Mb blocks rather than one, then version 0 is associated with both Block 0 and Block 1, indicated by the following statement in the CFI file:

```
BLOCK 0 VERSION 0 0x#########
BLOCK 1 VERSION 0 END 0x#########
```

PROM File Creation with iMPACT 11.1 (or Later) Software

The Xilinx iMPACT Programming and File Generation software is used to create a valid PROM file for the Platform Flash PROM Family. The basic flow takes an FPGA design bitstream input and converts it into a valid PROM file format (MCS/EXO). The basic modes discussed in Chapter 2, "Platform Flash PROM Quick Start" for XCFxxS/XCFxxP should follow the "Creating a Simple PROM File with iMPACT Software" section.

Designers accessing the advanced features of the XCFxxP device must generate the CFI file. This file enables the programming of the XCFxxP design revisions, CLKOUT, or decompression features. The section "Creating an (XCFxxP) Advanced PROM File with iMPACT Software" shows how to enable these advanced features.

Creating a Simple PROM File with iMPACT Software

The Xilinx iMPACT software is used to create a PROM file for programming. The flow used in the iMPACT software to generate the simple PROM file is described below.

Note: For more detailed information about the iMPACT software, refer to ISE® software online help. A complete set of the ISE 11 software documents, including a web version of the ISE 11 online help, are available at: http://www.xilinx.com/support/documentation/dt_ise11-1.htm.

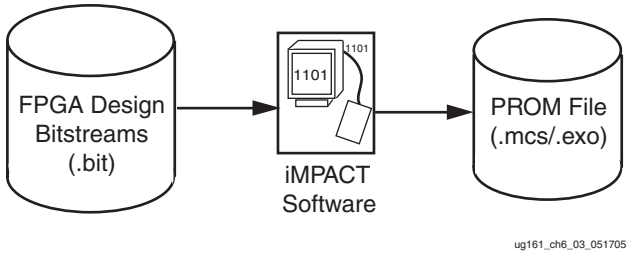


Figure 6-3: **Simple PROM File Generation Flow**

1. Start the iMPACT software, choose to create a new project in the New iMPACT Project dialog box, and select **Prepare a PROM File** in the Welcome to iMPACT dialog box (Figure 6-4).

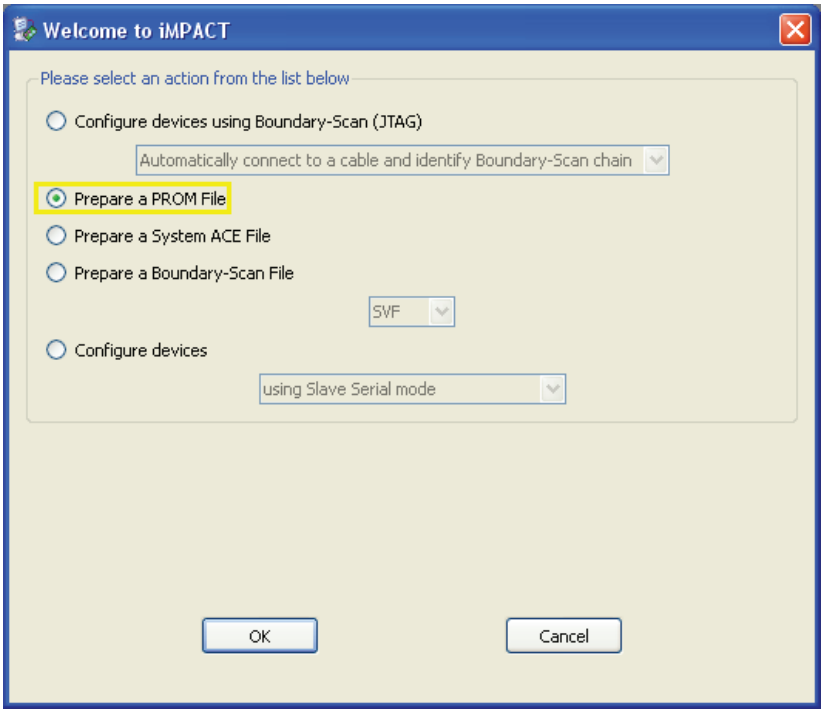
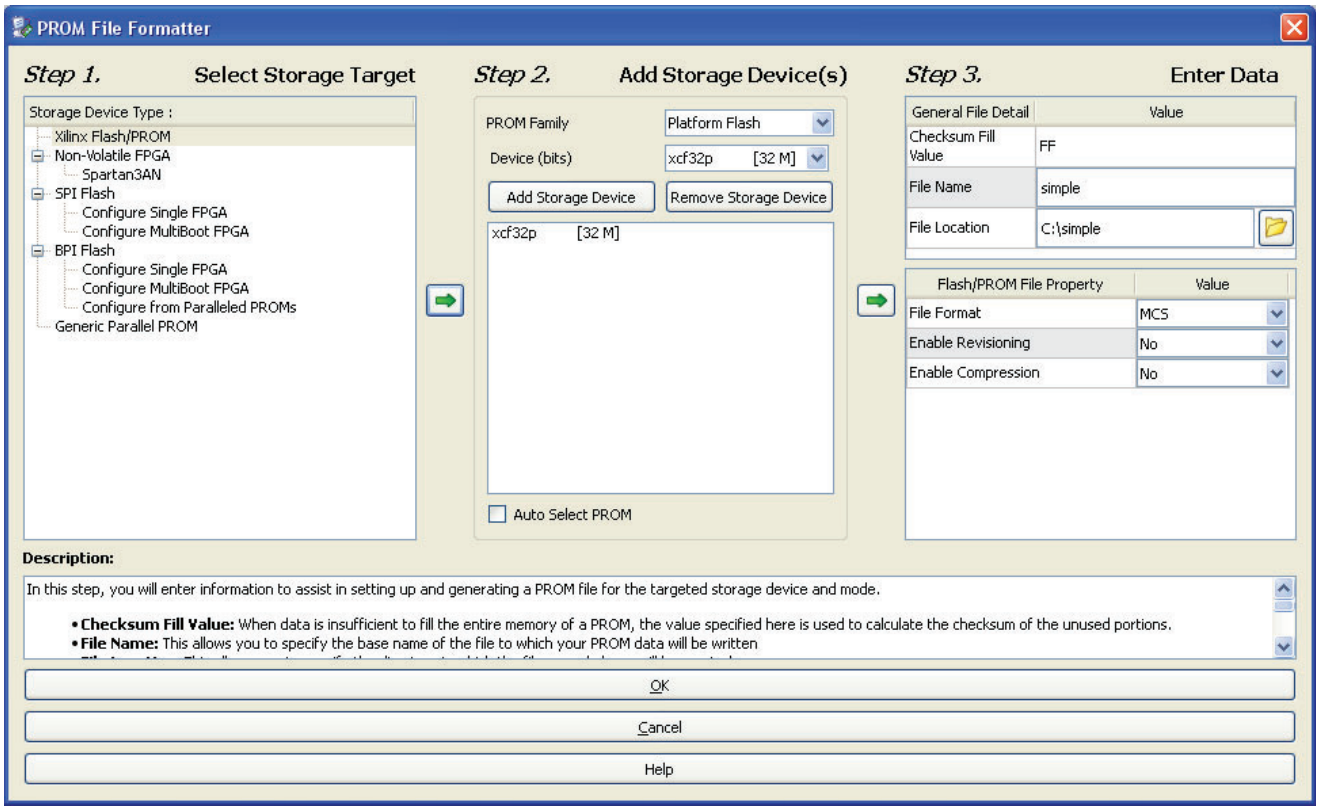


Figure 6-4: **Enter PROM Generation Mode in iMPACT Software**

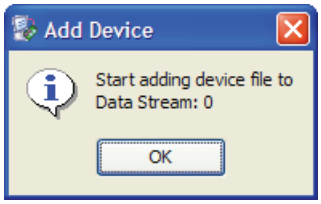
2. Click **OK** to display the PROM File Formatter dialog box (Figure 6-5).
   a. In the "Step 1. Select Storage Target" section:
      - In the Storage Device Type box, select the Xilinx Flash/PROM device type.
      - Click the green arrow button between the Step 1 and Step 2 sections to proceed to Step 2.
   b. In the "Step 2. Add Storage Device(s)" section:
      - In the PROM Family drop down, select Platform Flash.
      - In the Device (bits) drop down, select the appropriate Platform Flash PROM.
      - Click **Add Storage Device** to add the selected XCF32P PROM to the list of devices in which FPGA bitstreams are stored.
      - Click the green arrow button between the Step 2 and Step 3 sections to proceed to Step 3.
   c. In the "Step 3. Enter Data" section:
      - In the File Name box, enter the file name.
      - In the File Location box, set the location where the PROM file will be generated. Use the **Browse** button to navigate to the desired location.
      - In the File Format drop down, select **MCS**.
      - In the Enable Revisioning drop down, select **No**.
      - In the Enable Compression drop down, select **No**.



ug161_ch6_05_081109

Figure 6-5: **Select PROM Type and Output File Properties**

3. When the PROM File Formatter fields are set, click **OK** to confirm the PROM type and output file properties. The iMPACT software displays the Add Device dialog box (Figure 6-6).



ug161_ch6_06_081109

Figure 6-6: **Add Device Dialog Box**

4. Click **OK** to begin adding FPGA bitstreams for inclusion in the PROM file.
5. Click **Next** to verify the summary of the selections. Click **Next** if they are correct. Select **Back** at any stage to change your selections. At the **Add Device File** menu, select the appropriate FPGA bitstream(s).
6. Browse when prompted for the design input bitstream file(s) to be converted into the PROM file (Figure 6-7), and click **Open**. After adding the bitstream, the iMPACT software prompts the user for additional FPGA bitstream files to be added to the PROM file. If no further bitstream files are to be added to the PROM file, click **No**. After completing the Add Device procedure, double-click **Generate File** in the iMPACT Processes window to generate the PROM file.

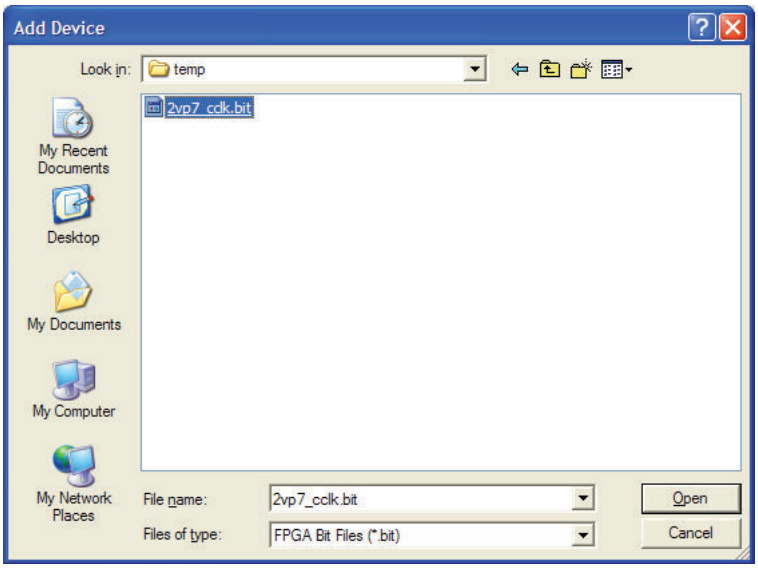


Figure 6-7: **Browse for FPGA Bitstream(s) Input to Create PROM File**

7. After each bit file is added, the user is prompted for additional bitstream files to be added to the PROM file. If no further bitstream files are to be added to the PROM file, select **No**. (Figure 6-8).

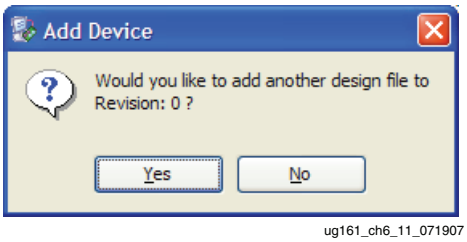
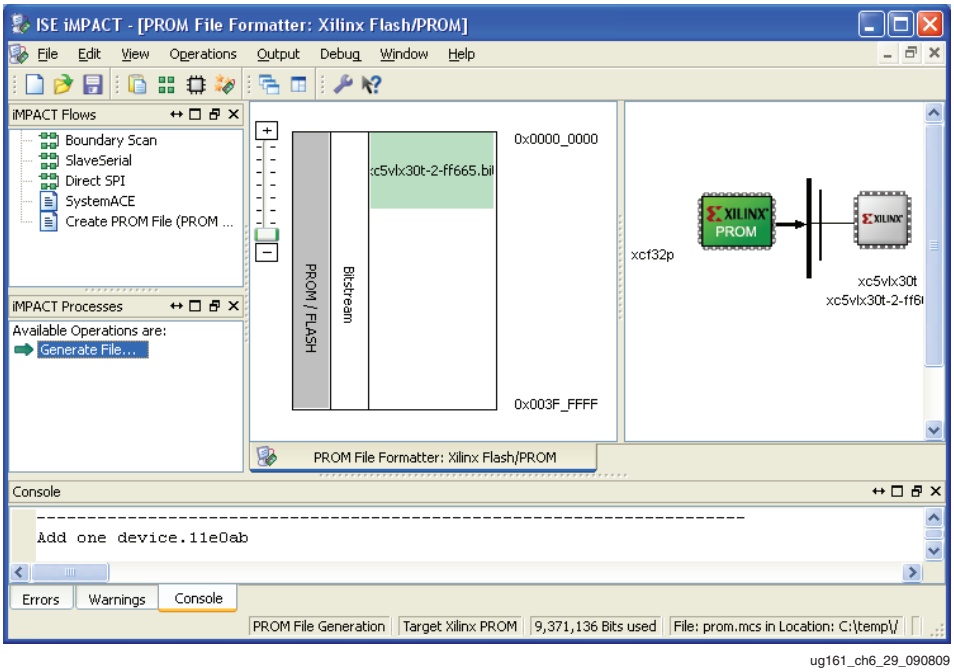


Figure 6-8: **Assign Associated FPGA Bitstreams to Multiple Revisions**

8. After completing the Add Device procedure, double-click **Generate File** in the iMPACT Processes window (Figure 6-9) to generate the PROM file.

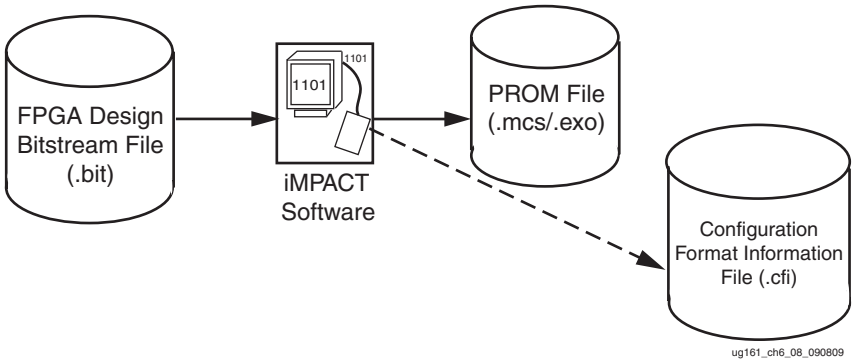


ug161_ch6_29_090809

Figure 6-9: **iMPACT Processes Window**

Creating an (XCFxxP) Advanced PROM File with iMPACT Software

When users first launch the iMPACT software tool, they are guided through a wizard for a choice of valid user options. The XCFxxP Advanced Feature PROM file generation flow varies slightly from the simple PROM generation model. The advanced flow includes the generation of the required CFI to enable the advanced feature set of this member of the Platform Flash PROM family (Figure 6-10).



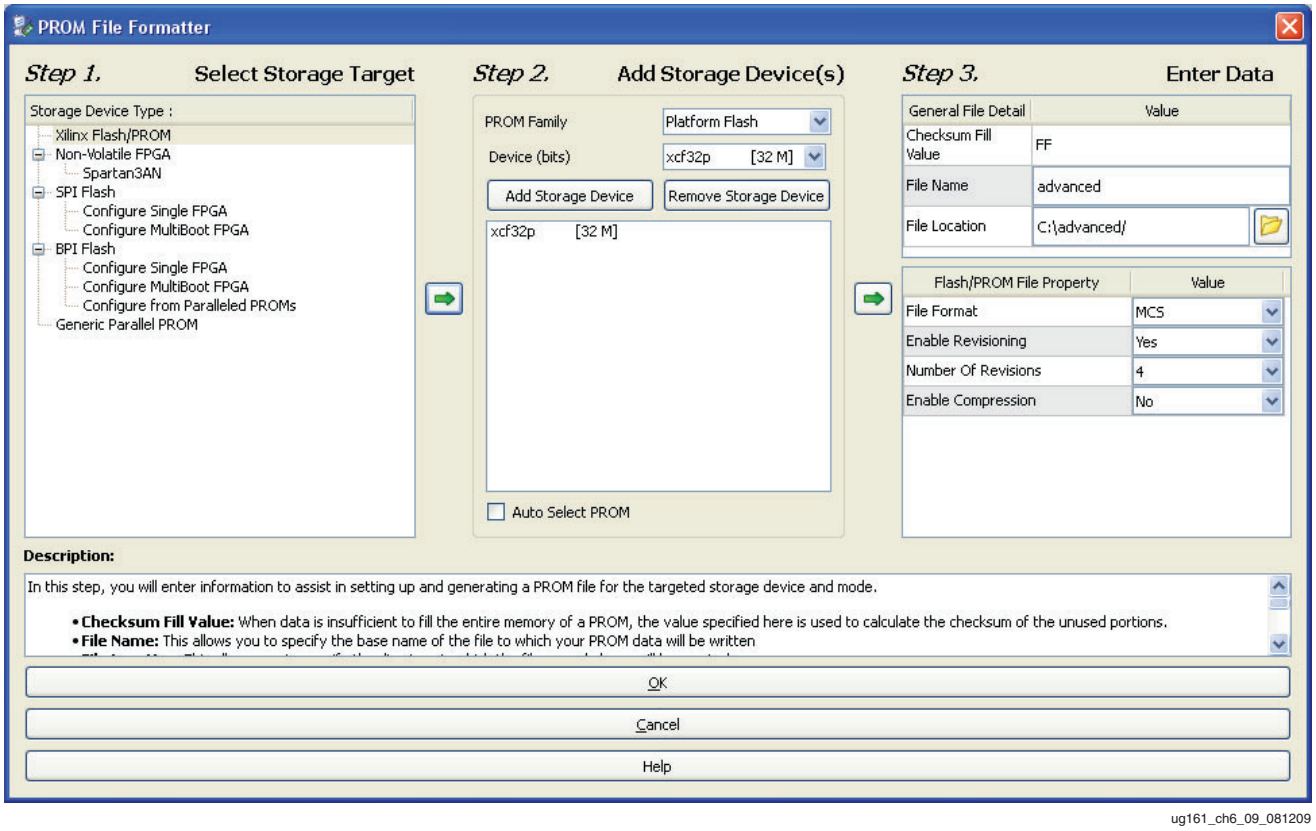
ug161_ch6_08_090809

Figure 6-10: **XCFxxP Advanced Feature PROM File Generation Flow**

1. Start the iMPACT software, choose to create a new project in the New iMPACT Project dialog box, and select **Prepare a PROM File** in the Welcome to iMPACT dialog box (Figure 6-4, page 68).
2. Click **OK** to display the PROM File Formatter dialog box (Figure 6-11).
   a. In the "Step1. Select Storage Target" section:
      - In the Storage Device Type box, select the Xilinx Flash/PROM device type.
      - Click the green arrow button between the Step 1 and Step 2 sections to proceed to Step 2.

b. In the "Step 2. Add Storage Device(s)" section
   - In the PROM Family drop down, select Platform Flash.
   - In the Device (bits) drop down, select the appropriate Platform Flash PROM.
   - Click **Add Storage Device** to add the selected XCF32P PROM to the list of devices in which FPGA bitstreams are stored. When multiple, cascaded PROMs are required, click **Add Storage Device** again to add additional cascaded PROMs to the list of PROM storage devices.
   - Click the green arrow button between the Step 2 and Step 3 sections to proceed to Step 3.

c. In the "Step 3. Enter Data" section:
   - In the File Name box, enter the file name.
   - In the File Location box, set the location where the PROM file will be generated. Use the **Browse** button to navigate to the desired location.
   - In the File Format drop down, select **MCS**.
   - In the Enable Revisioning drop down, select **Yes**.
   - Set **Number of Revisions** to 4 (assuming there are four FPGA design revisions).
   - In the Enable Compression drop down, select **No**.



ug161_ch6_09_081209

Figure 6-11: **PROM File Formatting for an XCF32P PROM with Revisions**

3. Upon finishing the PROM selection procedure, the iMPACT software automatically begins the procedure for adding the FPGA bitstreams to be stored in the selected PROMs. (Figure 6-12).

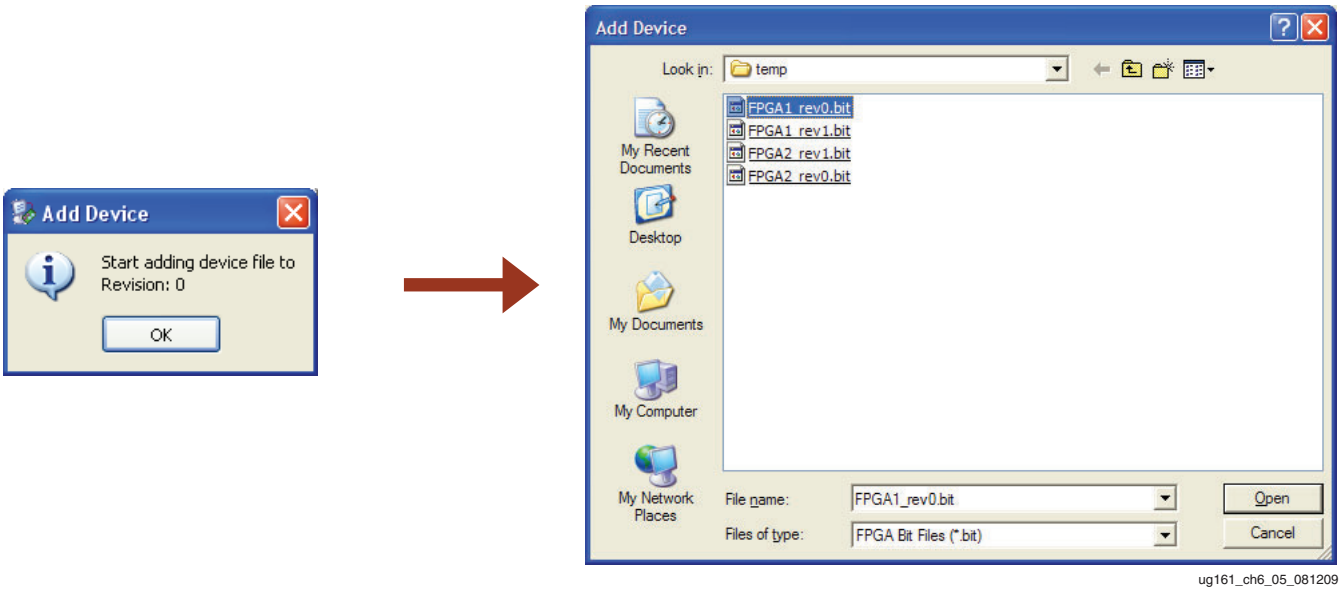


Figure 6-12: **Browse for FPGA Bitstream(s) Inputs to Create Revisioned PROM File**

4. After each bit file is added, the user is prompted for additional files for a chain of FPGAs to be targeted. If the configuration chain contains only a single FPGA, then select **No**. (Figure 6-13).

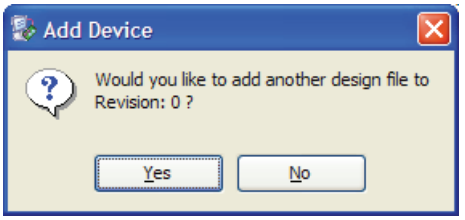
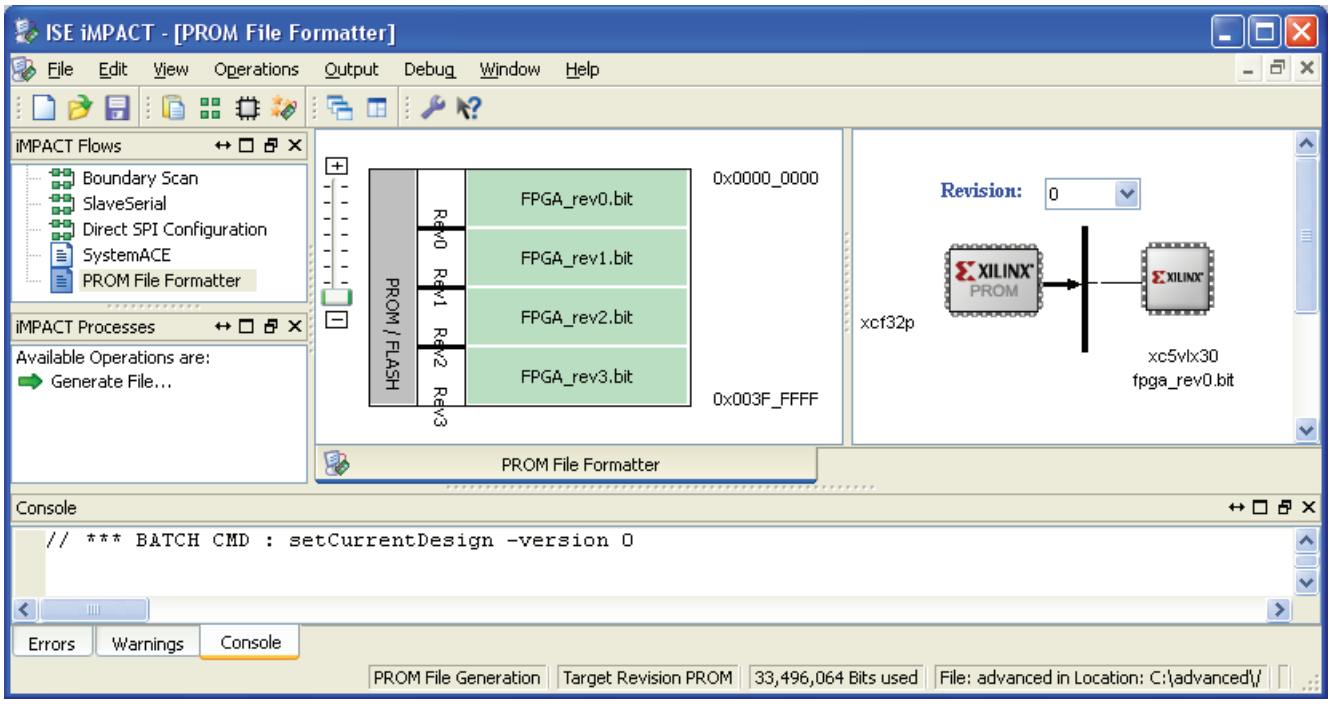


Figure 6-13: **Assign Associated FPGA Bitstreams to Multiple Revisions**

5. After the bit file(s) for Revision 0 are specified, the wizard prompts the user to select the bit files for each additional revision. In this example, with four revisions, bit file(s) need to be assigned for Revision 0, Revision 1, Revision 2, and Revision 3.
6. When the user specifies the FPGA bitstream(s) for each revision, the GUI displays the target chain of FPGAs and associated bitstream details, and the PROM usage summary (Figure 6-14).



ug161_ch6_12_081209

Figure 6-14: **Assigned Bitstreams to Design Revisions**

7. Select a specific revision (0-3) from the Revision drop-down list to toggle between design revisions. The window displays the targeted FPGA chain and associated bit files for that revision. Click **Revision (0-3)** for details on the chain composition and bit file assignments the targets.
8. In the iMPACT Processes box, double-click **Generate File...** to generate the MCS file and CFI file for the advanced PROM with revisions.

Programming a PROM with Design Revisions with iMPACT

The Xilinx iMPACT programming software can be used to program a design revisioned Platform Flash PROM in-system. To program the Platform Flash PROMs, the user must have both a PROM file (MCS/EXO) and a CFI file (Figure 6-15). The CFI file is generated automatically by iMPACT during the PROM file generation flow described in the "Creating an (XCFxxP) Advanced PROM File with iMPACT Software"section.

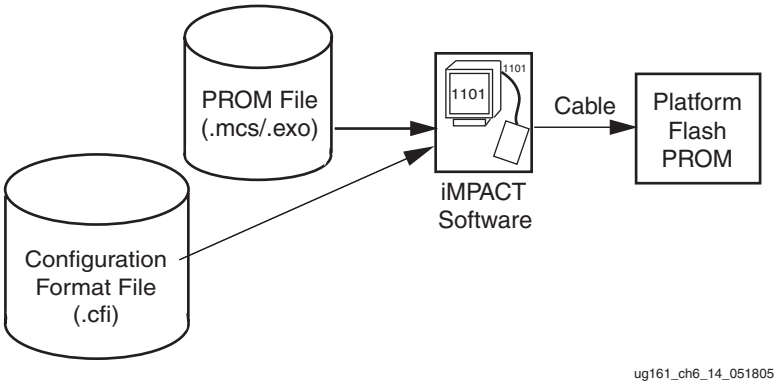


Figure 6-15: **Programming Revisioned Platform Flash PROM in iMPACT Configuration Mode**

To program a revisioned PROM in the iMPACT 11.1 (or later) software:

1. Start iMPACT, create a new project, and select **Configure devices using Boundary-Scan (JTAG)** in the wizard to select the Boundary-Scan mode. Choose the **Automatically connect to a cable and identify Boundary-Scan chain** option if iMPACT is connected via a Xilinx cable to the JTAG port of a powered board; or, select **Enter a Boundary-Scan chain manually** to manually define the Boundary-Scan chain (Figure 6-16).

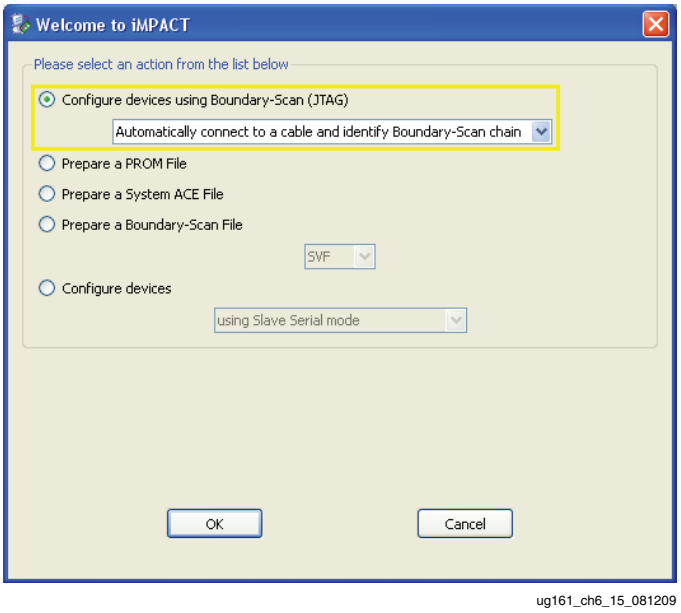


Figure 6-16: **Entering iMPACT Configuration Mode**

2. When the Boundary-Scan chain is defined, assign the PROM MCS file to the Platform Flash PROM in the chain.
3. Click on the target Platform Flash PROM to select it for programming.
4. Select **Operations** → **Program...** The programming properties should be set for the target Platform Flash PROM prior to programming. If the assigned MCS and CFI file pair was created with the revision feature enabled for the XCFxxP, then the individual revision properties are available in the programming properties (Figure 6-17).

Note: iMPACT only recognizes the advance, revision configuration for a PROM when the iMPACT software can find a CFI file with the same base file name as the PROM MCS file in the same directory as the PROM MCS file. Otherwise, when no matching CFI file is found, the iMPACT software assumes the advanced features of the XCFxxP PROM are not used and disables access to the Revision Properties in the Programming Properties dialog box.

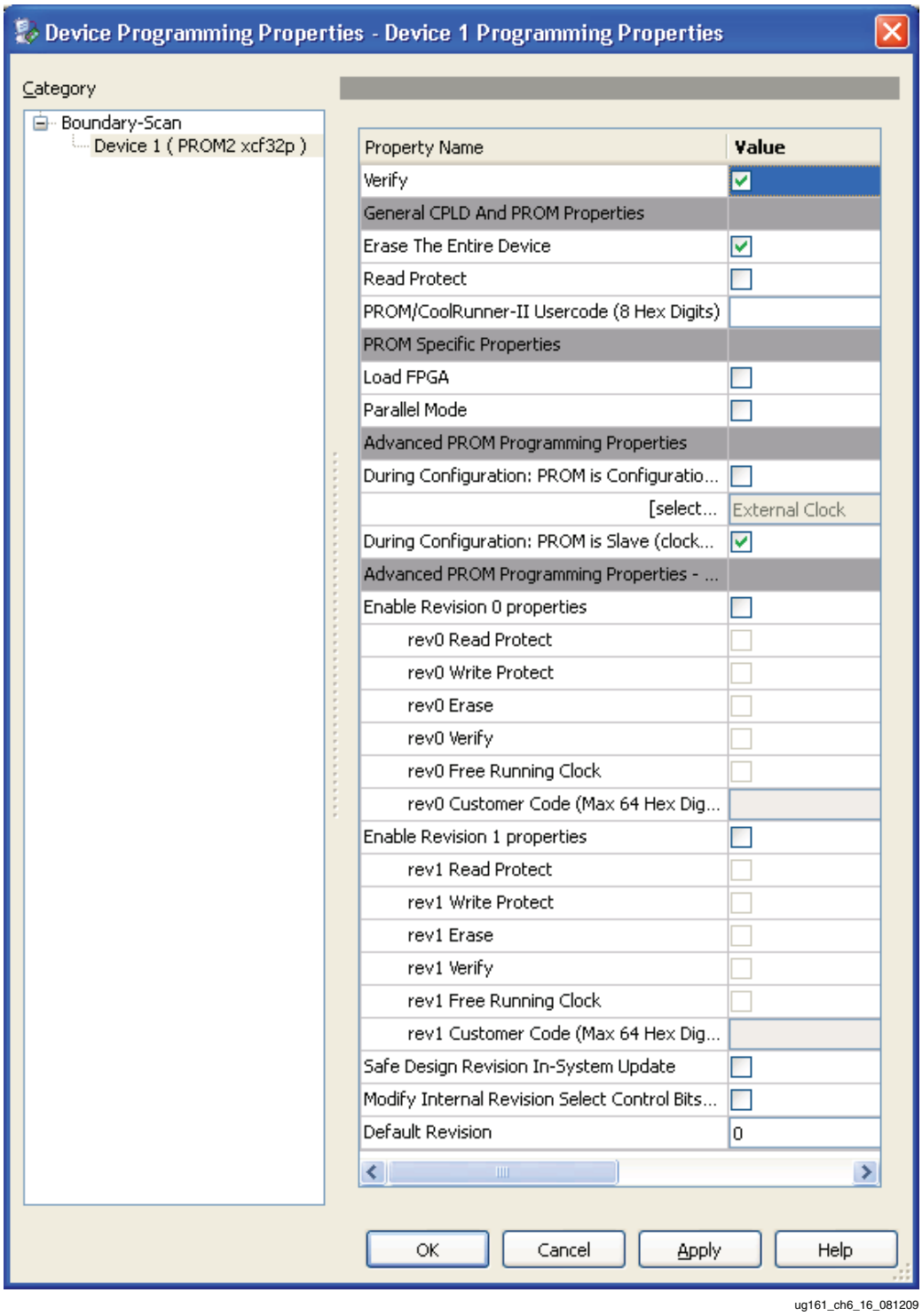


Figure 6-17: **Programming Options Menu for Platform Flash PROMs with Design Revisions**

5. Click **OK** at the bottom of the Programming Properties dialog to begin the programming operation on the target Platform Flash PROM.

The design revision feature gives the user the powerful capability for reprogramming. A single revision can be targeted for erase or in-system programming through JTAG. Targeting a single revision allows field updates or tests to be done with a known good system design intact in the remaining untouched revision(s). The Platform Flash PROM revisions can be selected with an external pin selection or internal registry settings as discussed in the previous section. The internal programmable register control bits are set with the Default Revision selection.

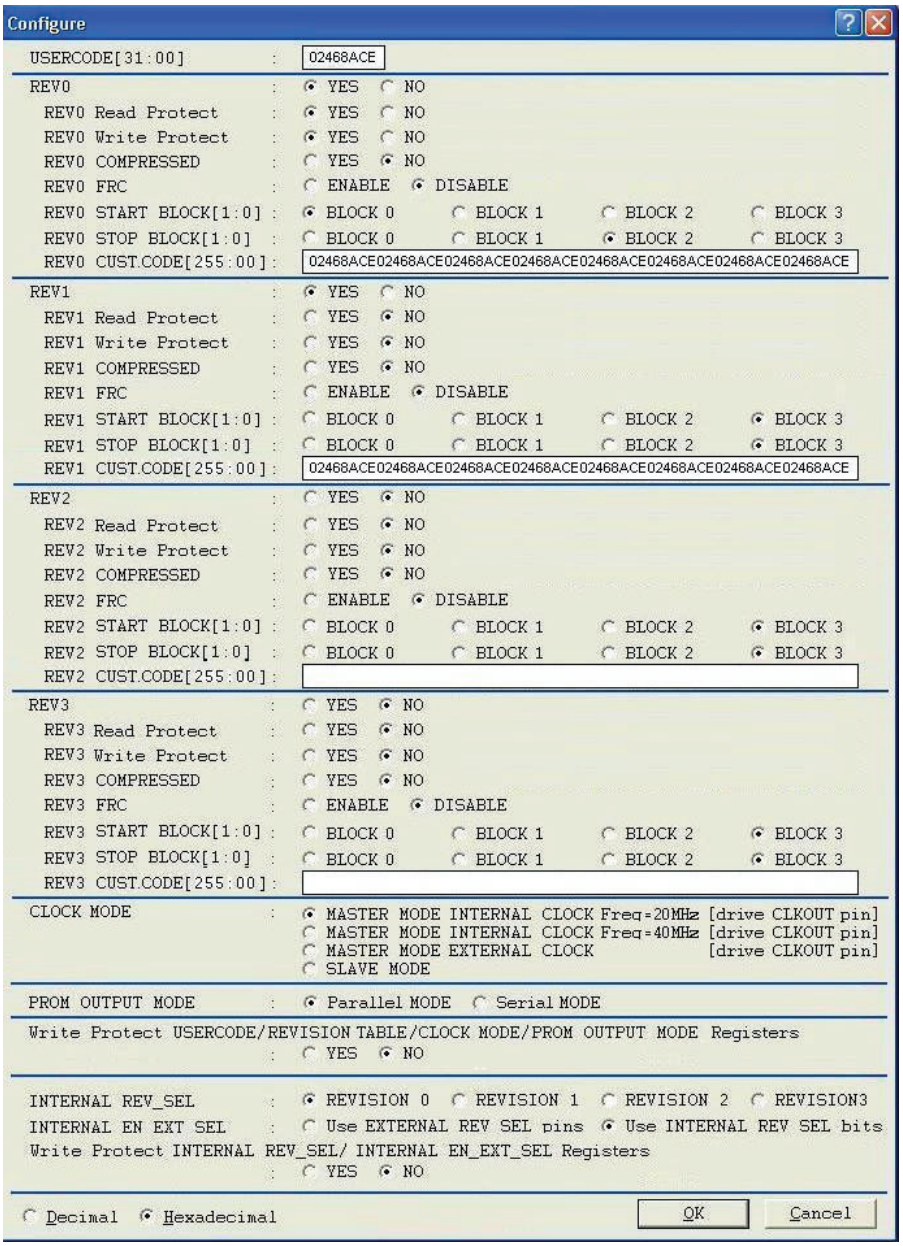
Programming an XCFxxP with a Third Party Programmer

A third-party programmer must program the FPGA design data into the PROM data array and must program the appropriate PROM option settings. A third-party programmer requires the PROM data in the form of a standard formatted PROM file, such as an MCS data file. See "PROM File Creation with iMPACT 11.1 (or Later) Software," page 67 for instructions on generating a standard PROM file that is properly formatted for a third-party programmer.

Note: An FPGA design BIT file is not an appropriate input data format for third-party programmers.

To enable the advanced features available in the XCFxxP Platform Flash PROM, the programming tool must also program the appropriate internal register options. As indicated in "Programming a PROM with Design Revisions with iMPACT," the iMPACT programming tool utilizes both the MCS file (containing data only) and the CFI file. The CFI file contains revision mapping information as well as other options specifically linked to the file generation flows. The CFI file helps to define the correct revision control settings which correspond to the associated MCS data file. In addition to the revision control options, several other options can be set at programming time. Options include the clocking options, serial or parallel data output, read protection, write protection, usercode, and customer code. The way in which options are presented on the user interface varies depending on the programming tool used. Typically, third party programmers do not utilize the CFI file to determine the required settings for revision control, so the user must be able to manually set these options. This section presents a general overview of all of the available programmable options. Refer to the individual programmer for details on how these options are arranged for the specific programmer.

When viewed as a general device, the options shown in Figure 6-18 are typically available at programming time.



UG161_c6_17_100108

Figure 6-18: **Configuration Options**

The XCFxxP memory space is composed of a set of one, two, or four 8 Mb data blocks (depending on the device size), where each data block or consecutive set of data blocks can be assigned to a particular revision. When cascading multiple XCFxxP PROMs, a revision can span across multiple PROMs. In addition to having up to four 8 Mb data blocks, each PROM has the following internal programmable registers:

- One user code register:
  - USERCODE[31:0]
- Up to four customer code registers (one per block):
  - DATA-CC[255:0] for Block 0

- DATA-CC[255:0] for Block 1
- DATA-CC[255:0] for Block 2
- DATA-CC[255:0] for Block 3

- Five general control registers:
  - DATA-BTC[31:0]: Design Revision Block Table Register (only DATA_BTC[27:0] are used)
  - DATA-CCB[15:0]: Customer Controlled Bits Register (only DATA_CCB[5:0] are used)
  - DATA-SUCR[15:0]: Special User Controlled Register (only SUCR[2:0] are used)
  - DATA-RDPT[15:0]: Read Protection Register (only DATA-RDPT[3:0] are used)
  - DATA-WRPT[15:0]: Write Protection Register (only DATA-WRPT[5:0] are used)

Usercode Register, USERCODE[31:0]

USERCODE is an optional 32-bit register that can be programmed to identify the design. See Figure 6-19.

Figure 6-19: **USERCODE Register**

Customer Code Registers, DATA-CC[255:0]

Programming the 256-bit customer code registers (DATA-CC) is optional.

There is one DATA-CC is associated with each data block. Typically, the customer code field is used to create a unique identifier for a particular revision by programming the DATA-CC for the first block in that revision. The customer code is erased when the corresponding block of data is erased. See Figure 6-20.

Figure 6-20: **Customer Code Registers**

Design Revision Block Table Register, DATA-BTC[31:0]

The Design Revision Block Table register (DATA-BTC) must be programmed to enable data access. The DATA-BTC register controls how each revision selection maps to the internal data blocks. The settings used to map the revision location during iMPACT file generation flow can be determined by the design revision options written into the corresponding CFI file. The default erased state for the DATA-BTC register causes the PROM to have no data associated with any of the four possible revision locations, so the DATA-BTC must be programmed in order to access any data in any FPGA configuration mode (see Figure 6-21).

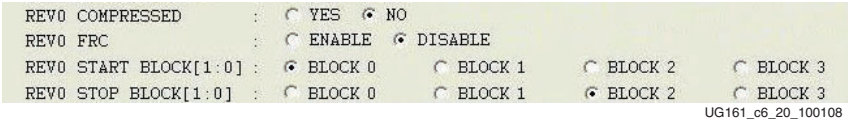


Figure 6-21: **Design Revisions**

Determining the DATA-BTC Settings with the CFI File

The CFI file is created by the iMPACT software automatically when compression or design revisioning is specified during PROM file generation. The CFI contains information about the PROM data formatting that can be used to determine the corresponding DATA-BTC settings. The CFI file specifies the following for each PROM:

- Up to four design revisions (set the SKIP bit in the DATA-BTC register for unused design revisions).
- Which 8 Mb data blocks are assigned to each design revision.
- A data compression flag for each design revision, if data was compressed during PROM file generation.
- FPGA DCM or DCI startup wait as specified in the FPGA bit file assigned to the revision. When an FPGA startup wait option is set and the FPGA is the configuration slave with the PROM driving CLKOUT, then the PROM clocking option should be set to use a free-running configuration clock.

Determining the DATA_BTC Settings without the CFI File

The CFI file is not always created if compression or design revisioning options are not specified during PROM file generation. As shown in the sample programming options in Figure 6-22, when using a PROM data file (MCS) with no CFI file, typically **REV0** is selected with all other revisions set to **NO**, and **REV0 COMPRESSION** is typically set to **NO**. By default, when no revision structure is specified (no CFI file), then Revision 0 can be set to start at block 0 and to end at the last 8 Mb block available on the PROM.

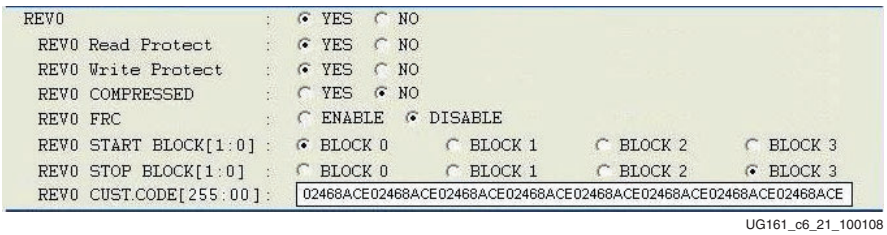


Figure 6-22: **Revision Structure**

DATA-BTC[31:0] Bit Assignments

Table 6-2 shows the bit assignments for the Design Revision Block Table register (DATA-BTC). Bits [31:28] are reserved and set to all 1s.

Table 6-2: **DATA-BTC[31:0] Bit Assignments**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Revision 3						Revision 2						Revision 1						Revision 0					
Reserved				Free-running clock				COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]	
1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

DATA-BTC[31:24]

The upper bits in the Design Revision Block Table register contain the free-running clock option switches.

- DATA-BTC[31:28] => Reserved (set to all 1s)
  - DATA-BTC[31:28] = 1111
- DATA-BTC[27] => Revision 3 free-running clock option
  Enable if DCI or DCM flagged for Revision 3 in the CFI file
  - 1 = Disable Revision 3 free-running clock
  - 0 = Enable Revision 3 free-running clock
- DATA-BTC[26] => Revision 2 free-running clock option
  Enable if DCI or DCM flagged for Revision 2 in the CFI file
  - 1 = Disable Revision 2 free-running clock
  - 0 = Enable Revision 2 free-running clock
- DATA-BTC[25] => Revision 1 free-running clock option
  Enable if DCI or DCM flagged for Revision 1 in the CFI file
  - 1 = Disable Revision 1 free-running clock
  - 0 = Enable Revision 1 free-running clock
- DATA-BTC[24] => Revision 0 free-running clock option
  Enable if DCI or DCM flagged for Revision 0 in the CFI file
  - 1 = Disable Revision 0 free-running clock
  - 0 = Enable Revision 0 free-running clock

DATA-BTC[23:0]

The lower bits in Design Revision Block Table register contain the unique settings for each revision (6 bits per revision). The settings include the SKIP bit (to indicate the revision is not used), the COMPRESSED bit, and fields to specify the blocks at which a particular revision starts and ends.

- DATA-BTC[23:18] => Revision 3 settings

- DATA-BTC[23]
  - 0 = COMPRESSED
  - 1 = NOT COMPRESSED
- DATA-BTC[22]
  - 0 = DON'T SKIP (Revision 3 used)
  - 1 = SKIP
- DATA-BTC[21:20]
  - STOP[1:0] = Block where Revision 3 stops (CFI [END] block)
- DATA-BTC[19:18]
  - START[1:0] = Block where Revision 3 starts

- DATA-BTC[17:12] => Revision 2 settings
  - DATA-BTC[17]
    - 0 = COMPRESSED
    - 1 = NOT COMPRESSED
  - DATA-BTC[16]
    - 0 = DON'T SKIP (Revision 2 used)
    - 1 = SKIP
  - DATA-BTC[15:14]
    - STOP[1:0] = Block where Revision 2 stops (CFI [END] block)
  - DATA-BTC[13:12]
    - START[1:0] = Block where Revision 2 starts
- DATA-BTC[11:6] => Revision 1 settings
  - DATA-BTC[11]
    - 0 = COMPRESSED
    - 1 = NOT COMPRESSED
  - DATA-BTC[10]
    - 0 = DON'T SKIP (Revision 1 used)
    - 1 = SKIP
  - DATA-BTC[9:8]
    - STOP[1:0] = Block where Revision 1 stops (CFI [END] block)
  - DATA-BTC[7:6]
    - START[1:0] = block where Revision 1 starts
- DATA-BTC[5:0] => Revision 0 settings
  - DATA-BTC[5]
    - 0 = COMPRESSED
    - 1 = NOT COMPRESSED
  - DATA-BTC[4]
    - 0 = DON'T SKIP (Revision 0 used)
    - 1 = SKIP
  - DATA-BTC[3:2]

- STOP[1:0] = Block where Revision 0 stops (CFI [END] block)

♦ DATA-BTC[1:0]

- START[1:0] = Block where Revision 0 starts

Example 1: Determining the DATA-BTC Settings from a CFI File with One Revision

Figure 6-23 shows the CFI file contents with one revision (Revision 0).

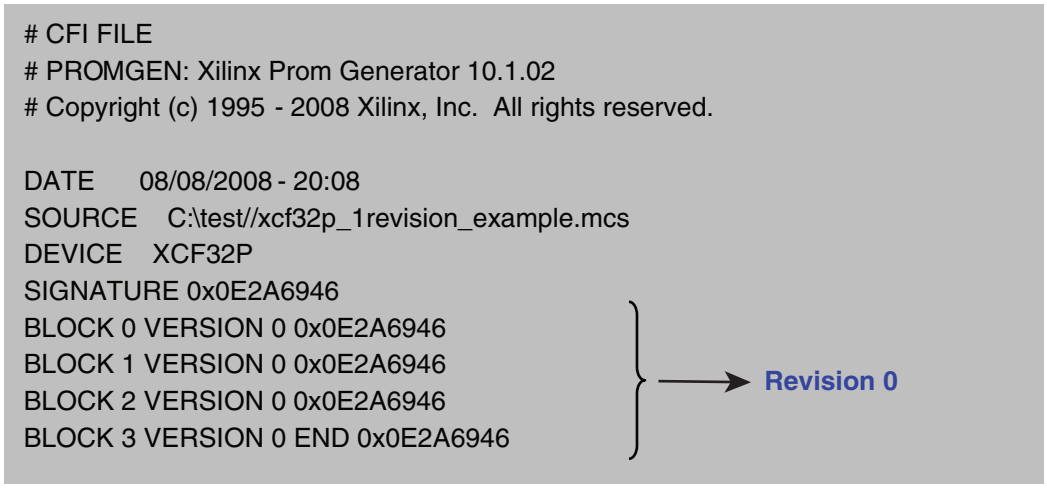


Figure 6-23: **Example 1: CFI File with One Revision**

This CFI file indicates that DATA-BTC[31:0] should be set to `FFFFFFECh` (see Table 6-3).

Table 6-3: **DATA-BTC[31:0] Bit Assignments for Example 1**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Revision 3						Revision 2						Revision 1						Revision 0					
Reserved				Free-running clock = OFF				COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0

These bit assignments indicate the following:

- One design revision is used:
  - ♦ DATA-BTC[5:0] (Revision 0)
  - ♦ All DATA-BTC bits for the three unused design revisions are set to 1.
- Revision 0 covers Block 0 - Block 3.
- Compression is not used.
- FPGA DCM or DCI start-up wait is not used.

Example 2: Determining the DATA-BTC Settings from a CFI File with Two Revisions

Figure 6-24 shows the CFI file contents with two revisions (Revision 0 and Revision 1).

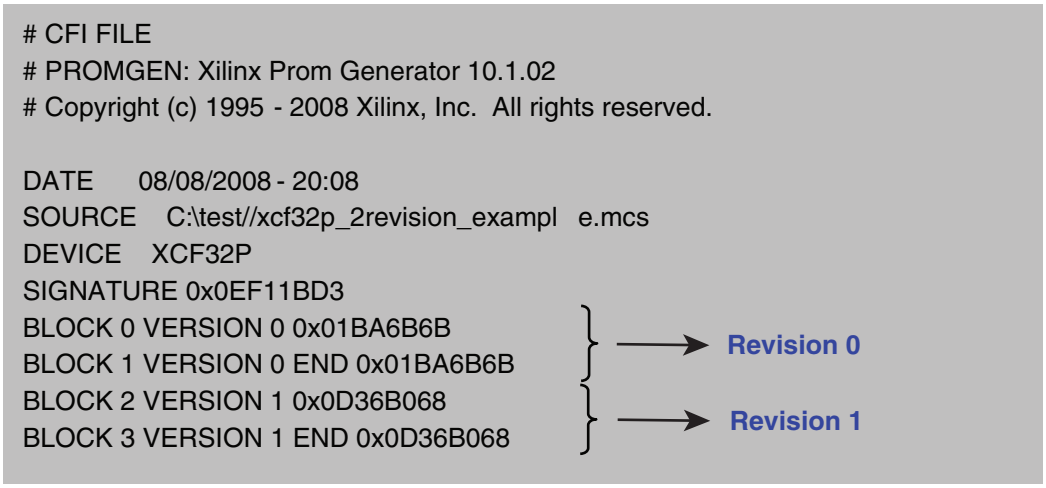


Figure 6-24: **Example 2: CFI File with Two Revisions**

This CFI file indicates that DATA-BTC[31:0] should be set to `FFFFFBA4h` (see Table 6-4).

Table 6-4: **DATA-BTC[31:0] Bit Assignments for Example 2**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Revision 3						Revision 2						Revision 1						Revision 0					
Reserved				Free-running clock = OFF				COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0

These bit assignments indicate the following:

- Two design revisions are used:
  - DATA-BTC[5:0] (Revision 0) and DATA-BTC[11:6] (Revision 1)
  - All DATA-BTC bits for the two unused design revisions are set to 1.
- Revision 0 covers Block 0 – Block 1, and Revision 1 covers Block 2 – Block 3.
- Compression is not used.
- FPGA DCM or DCI start-up wait is not used.

Example 3: Determining the DATA-BTC Settings from a CFI File with Four Revisions

Figure 6-25 shows the CFI file contents with four revisions (Revision 0 through Revision 3).

```
# CFI FILE
# PROMGEN: Xilinx Prom Generator 10.1.02
# Copyright (c) 1995 - 2008 Xilinx, Inc.  All rights reserved.

DATE      08/08/2008 - 20:08
SOURCE    C:\test//xcf32p_4revision_example.mcs
DEVICE    XCF32P
SIGNATURE 0x3BF22259
BLOCK 0 VERSION 0 END COMPRESSED 0x0FEB9F88 DCM DCI ──► Revision 0
BLOCK 1 VERSION 1 END COMPRESSED 0x0E14A362 DCM DCI ──► Revision 1
BLOCK 2 VERSION 2 END COMPRESSED 0x0E14A362 DCM DCI ──► Revision 2
BLOCK 3 VERSION 3 END COMPRESSED 0x0FDD3C0D DCM DCI ──► Revision 3
```

UG161_c6_24_100108

Figure 6-25: **Example 3: CFI File with Four Revisions**

This CFI file indicates that DATA-BTC[31:0] should be set to `F03CA140h` (see Table 6-5).

Table 6-5: **DATA-BTC[31:0] Bit Assignments for Example 3**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Revision 3						Revision 2						Revision 1						Revision 0					
Reserved				Free-running clock = ON				COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]		COMPRESSED	SKIP	STOP[1:0]		START[1:0]	
1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0

These bit assignments indicate the following:

- Four design revisions are used:
  - DATA-BTC[5:0] (Revision 0), DATA-BTC[11:6] (Revision 1), DATA-BTC[17:12] (Revision 2), and DATA-BTC[23:18] (Revision 3)
- Revision 0 only covers Block 0, Revision 1 only covers Block 1, Revision 2 only covers Block 2, and Revision 3 only covers Block 3.
- Compression is used.
- FPGA DCM or DCI start-up wait is used.

Customer Controlled Bits Register, DATA-CCB[15:0]

The Customer Controlled Bits register (DATA-CCB) controls the following PROM options: serial or parallel data output mode and configuration clock source. The erased state for the DATA-CCB register causes the PROM to default to the slave clock and serial data output settings, corresponding to the FPGA Master Serial configuration mode. If the XCFxxP PROM is not intended to configure the FPGA is Master Serial mode, then this register is required to be programmed.

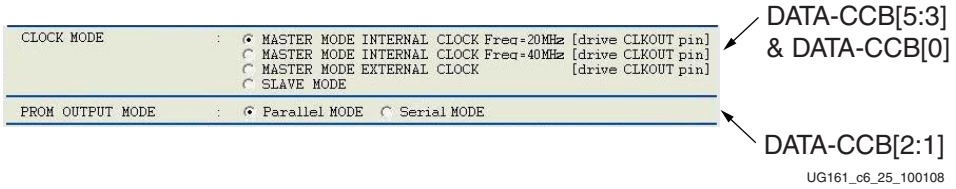


Figure 6-26: **Customer Controlled Bits Register**

- DATA-CCB[15:6] => Reserved (set to all 1s)
- DATA-CCB[5:0]: Select clock source and data output mode
  - DATA-CCB[5:4] => Internal oscillator frequency
    - `11` = 40 MHz (default)
    - `01` = 20 MHz
  - DATA-CCB[3] => Enable PROM as master (driving CLKOUT)
    - `1` = PROM is slave (FPGA in master mode)
    - `0` = PROM is master (FPGA in slave mode)
  - DATA-CCB[2:1] => Serial or parallel data output mode
    - `11` = Serial (1-bit) data output
    - `00` = Parallel (8-bit) data output
  - DATA-CCB[0] => Select PROM clock source
    - `1` = External clock source
    - `0` = Internal oscillator

If the FPGA is in **Master Serial mode** with the PROM as the clock slave, then DATA-CCB[7:0] is set to `FFh`.

If the FPGA is in **Master SelectMAP mode** with the PROM as the clock slave, then DATA-CCB[7:0] is set to `F9h`.

If the FPGA is in **Slave Serial mode** with the PROM as the clock master driving CLKOUT, then:

- Using the internal oscillator at 20 MHz, DATA-CCB[7:0] = `D6h`
- Using the internal oscillator at 40 MHz, DATA-CCB[7:0] = `F6h`
- Using an external clock supplied on the PROM CLK input, DATA-CCB[7:0] = `F7h`

If the FPGA is in **Slave SelectMAP mode** with the PROM as the clock master driving CLKOUT, then:

- Using the internal oscillator at 20 MHz, DATA-CCB[7:0] = `D0h`
- Using the internal oscillator at 40 MHz, DATA-CCB[7:0] = `F0h`
- Using an external clock supplied on the PROM CLK input, DATA-CCB[7:0] = `F1h`

Note: If compression is used then the PROM must be the configuration clock master, and the FPGA must be the configuration clock slave.

Special User Controlled Register, DATA-SUCR[15:0]

The Special User Controlled Register (DATA-SUCR) settings specify the internal bits which control revision selection when the external EN_EXT_SEL pin is High. The default erased state for the DATA-SUCR register causes the PROM to default to selecting Revision 3 (11b). However, this default internal revision selection is overridden by the external revision select pins (REV_SEL[1:0]) when the external pins are enabled by holding the EN_EXT_SEL pin is Low.

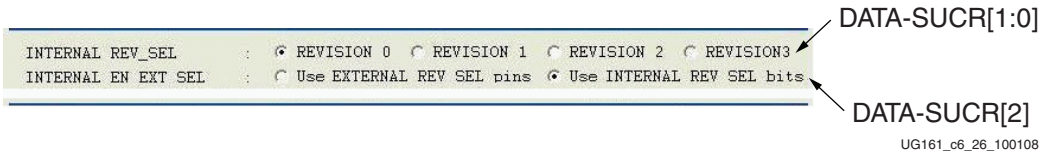


Figure 6-27: **Special User Controlled Register**

- DATA-SUCR[15:3] => Reserved (set to all 1s)
- DATA-SUCR[2:0] => Internal revision control bits
  - DATA-SUCR[2] => Internal EN_EXT_SEL bit
    - 0 = Use external REV_SEL[1:0] pins
    - 1 = Use internal REV_SEL[1:0] bits
  - DATA-SUCR[1:0] => Internal REV_SEL[1:0] bits
    - 00 = Revision 0
    - 01 = Revision 1
    - 10 = Revision 2
    - 11 = Revision 3

Driving the EN_EXT_SEL pin Low overrides all of the internal revision selection bit settings in the DATA-SUCR register and forces the PROM to use the external REV_SEL[1:0] pins to select the design revision. However, if the EN_EXT_SEL pin is High, then the following DATA-SUCR register settings affect how the revision is selected:

- SUCR[2] = 0 = Enable the external REV_SEL[1:0] pins

  When SUCR[2] = 0, the internal REV_SEL bits (SUCR[1:0]) are ignored by the PROM. The PROM uses the external REV_SEL[1:0] pins to control revision selection.

  Can set SUCR[2:0] = 8h, 9h, Ah, Bh
- SUCR[2] = 1 = Enable the internal REV_SEL[1:0] bits
  - SUCR[3:0] = Ch = Select Revision 00 (Revision 0)
  - SUCR[3:0] = Dh = Select Revision 01 (Revision 1)
  - SUCR[3:0] = Eh = Select Revision 10 (Revision 2)
  - SUCR[3:0] = Fh = Select Revision 11 (Revision 3)

Read Protection Register, DATA-RDPT[15:0]

The Read Protection Register (DATA-RDPT) is used to protect the data contents in the PROM from a JTAG read operation. Read protection can only be disabled by erasing the associated data block. By default, read protection is not enabled until the DATA-RDPT register is programmed.

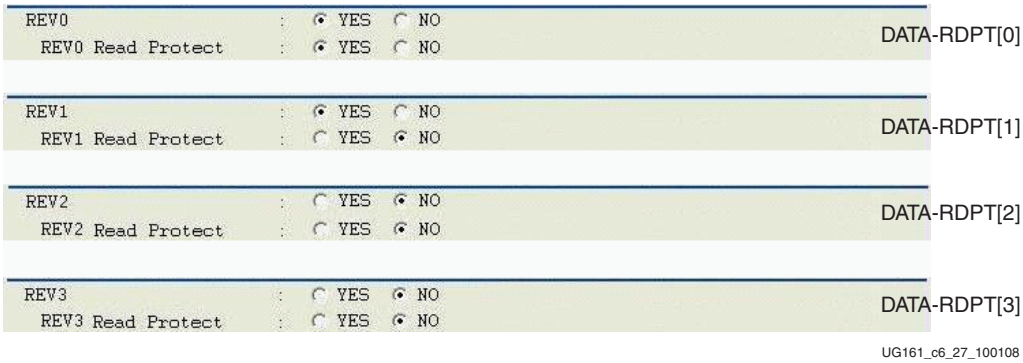


Figure 6-28: **Read Protection Register**

- DATA-RDPT[15:4] => Reserved (set to all 1s)
- DATA-RDPT[3:0]: Can be set for each 8 Mb data block (YES = 0 = Read protect)
  - DATA-RDPT[3] = Block 3
  - DATA-RDPT[2] = Block 2
  - DATA-RDPT[1] = Block 1
  - DATA-RDPT[0] = Block 0

Write Protection Register, DATA-WRPT[15:0]

The Write Protection Register (DATA-WRPT) is used to protect the data and register contents in the PROM from an inadvertent JTAG write operation. Write protection can be disabled by sending an unlock command prior to erasing the associated data block or control register. By default, write protection is not enabled until the DATA-WRPT register is programmed.

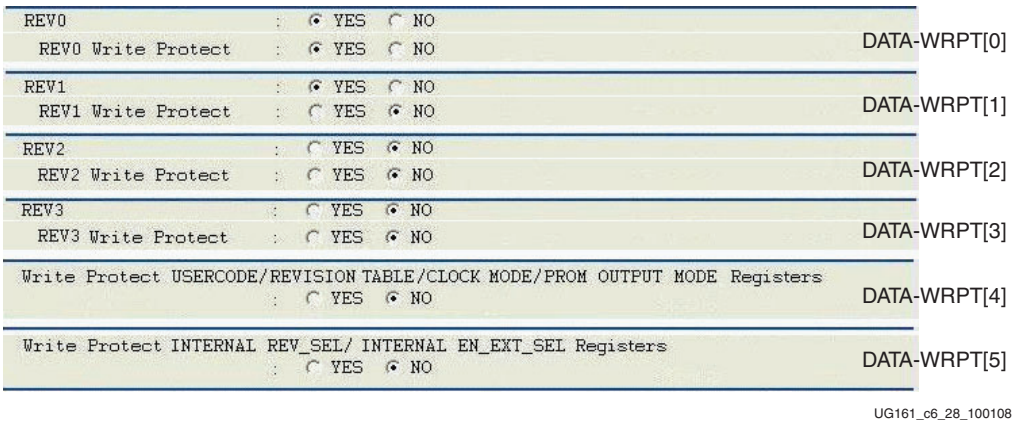


Figure 6-29: **Write Protection Register**

- DATA-WRPT[15:6] => Reserved (set to all 1s)
- DATA-WRPT[5:0]: can be set for each 8 Mb data block and for the control blocks (YES = 0 = write protect)
  - DATA-WRPT[5] = SUCR
  - DATA-WRPT[4] = USERCODE, BTC, and CCB
  - DATA-WRPT[3] = Block 3
  - DATA-WRPT[2] = Block 2
  - DATA-WRPT[1] = Block 1
  - DATA-WRPT[0] = Block 0

Production Programming Solutions

For production, Platform Flash PROM programming is supported in variety of programming methods and platforms. Third-party programmer vendors support the Platform Flash PROM for traditional, off-board programming. The standard JTAG port on the Platform Flash PROM also enables the advantages of in-system programming which include reduction of parts inventory overhead and update of designs late in the manufacturing process. Platform Flash PROM in-system programming is supported in production Boundary-Scan tools and automated test equipment.

Third-Party Device Programmers

Many third-party device programmer vendors support the Platform Flash PROM on their device and gang programmers. A partial list of vendors supporting the Platform Flash PROM is available at

http://www.xilinx.com/support/programr/dev_sup.htm

A third-party programmer must program the FPGA design data into the PROM data array and must program the appropriate PROM option settings. A third-party programmer requires the PROM data in the form of a standard formatted PROM file, such as an MCS data file. See "PROM File Creation with iMPACT 11.1 (or Later) Software," page 67 for instructions on generating a standard PROM file that is properly formatted for a third-party programmer. See "Programming an XCFxxP with a Third Party Programmer," page 78for instructions specific to the XCFxxP PROM.

Note: An FPGA design BIT file is not an appropriate input data format for third-party programmers.

Boundary-Scan Tools and Automated Test Equipment

The iMPACT software can generate standard serial vector format (SVF) files or JEDEC standard programming language and test (STAPL) files for programming the Platform Flash PROM in-system through the board JTAG chain. See Appendix A, "References" for further information regarding SVF and STAPL.

Most third-party Boundary-Scan tools accept SVF or STAPL for in-system programming. SVF-based in-system programming solutions exist for bed-of-nails, automated test equipment. For a list of Boundary-Scan tool vendors and automated test equipment, see

http://www.xilinx.com/products/design_resources/config_sol/resource/isp_ate.htm.

Note: For automated test equipment, a Platform Flash PROM programming solution can require hundreds of millions of test vectors to implement. Check the capacity of the automated test equipment when considering this solution.

Embedded In-System Programming Solutions

The in-system reprogrammability of the Platform Flash PROM can be leveraged for a system-level advantage. Embedded in-system programming solutions have been used for years with CPLDs and XC18V00 PROMs. The embedded solutions are available in the form of reference C code for an embedded processor or in the form of HDL code for a logic-only implementation. These embedded solutions also support the Platform Flash PROM.

The embedded solutions are similarly comprised of two steps:

1. The iMPACT software can generate the JTAG programming sequence for the Platform Flash PROM in the form of a standard SVF file.
2. An embedded executor executes a given SVF-based file to reprogram a Platform Flash PROM in system.

See XAPP058, *Xilinx In-System Programming Using an Embedded Microcontroller,* and XAPP424, *Embedded JTAG ACE Player* for details on the embedded programming solutions.

XILINX®

Chapter 7

Design Considerations

The Platform Flash PROM family density, cost, package size, and feature advantages have made this family a popular configuration solution for both new and existing designs.

For designers with existing designs, the "Migration from Legacy PROMs" section discusses package, voltage, and feature considerations, as well as differences between previous Xilinx® PROM families and the feature rich, cost saving Platform Flash PROMs. For new system designs, the "Reset and Power On Reset Activation" section gives suggestions to ensure a robust configuration setup is in place. Configuration is straightforward when the proper consideration to the setup is given.

Reset and Power On Reset Activation

At power up, the device requires the V_{CCINT} power supply to monotonically rise to the nominal operating voltage within the specified V_{CCINT} rise time. If the power supply cannot meet this requirement, the device might not perform power-on-reset properly. During the power up sequence, the PROM holds OE/$\overline{\text{RESET}}$ Low. After the required supplies have reached their respective Power On Reset (POR) thresholds, the OE/$\overline{\text{RESET}}$ release is delayed (TOER minimum) to allow more margin for the power supplies to stabilize before initiating configuration. The OE/$\overline{\text{RESET}}$ pin is connected to an external pull-up resistor and also to the target FPGA's $\overline{\text{INIT}}$ pin. For systems utilizing slow rising power supplies, an additional power monitoring circuit can be used to delay the target configuration until the system power reaches minimum operating voltages by holding the OE/$\overline{\text{RESET}}$ pin Low. When OE/$\overline{\text{RESET}}$ is released, the FPGAs $\overline{\text{INIT}}$ pin is pulled High allowing the FPGA's configuration sequence to begin. If the power drops below the POR threshold (VCCPOR), the PROM resets, and OE/$\overline{\text{RESET}}$ is again held Low until after the POR threshold is reached. OE/$\overline{\text{RESET}}$ polarity is not programmable.

For a fully powered Platform Flash PROM, a reset occurs whenever OE/$\overline{\text{RESET}}$ is asserted (Low) or $\overline{\text{CE}}$ is deasserted (High). The address counter is reset, $\overline{\text{CEO}}$ is driven High, and the remaining outputs are placed in a high-impedance state.

Notes:

1. The XCFxxS PROM only requires V_{CCINT} to rise above its POR threshold before releasing OE/$\overline{\text{RESET}}$.
2. The XCFxxP PROM requires both V_{CCINT} to rise above its POR threshold and for V_{CCO} to reach the recommended operating voltage level before releasing OE/$\overline{\text{RESET}}$.

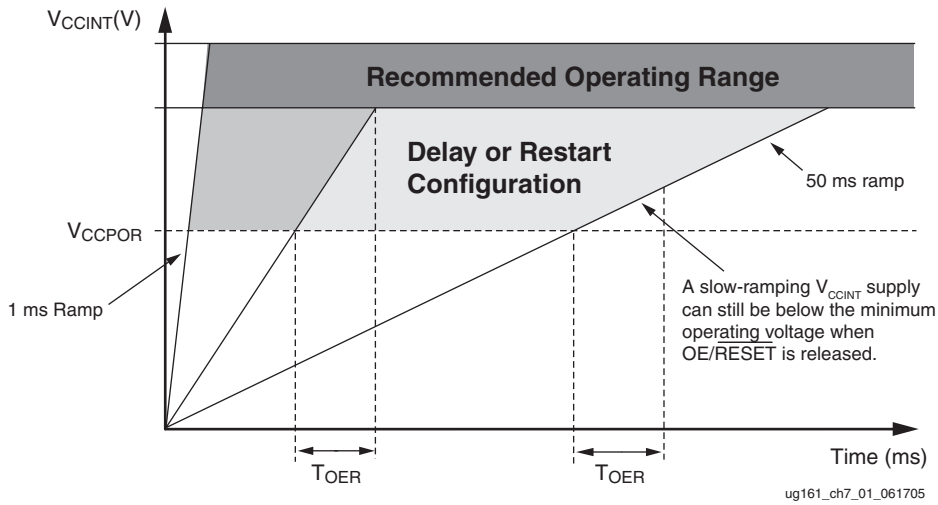


Figure 7-1: **Platform Flash PROM Power On Reset Threshold**

Regulating 3.3V Down to 1.8V for XCF08P/16P/32P Devices

Frequently, the power voltage applied to a board is higher than the nominal 1.8V V_{CCINT} level required by the XCFxxP Platform Flash PROMs. In these situations, power ICs are commonly used to perform the required DC-to-DC conversion of the power voltage. These devices, known as regulators, take an unregulated input voltage and provide a regulated output voltage independent of input voltage variations or output current fluctuations.

Voltage regulators from different vendors can be used to regulate 3.3V or 2.5V down to the XCFxxP V_{CCINT} of 1.8V.

The designer chooses the regulator that is most applicable to the design. A good regulator choice is one that is cost-efficient, power-savvy, and small in form factor. Application note XAPP389, *Powering CoolRunner™-II CPLDs*, provides an explanation of different regulator types and presents some typical circuits to highlight currently available commercial regulators.

For a list of companies that supplies power regulators, see the Xilinx Power Solutions website at:

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

PROM JTAG Boundary-Scan Chain

The Platform Flash ISP PROMs support the IEEE 1149.1 Boundary-Scan Standard connections. The JTAG signals on the Platform Flash PROM family are used for programming and can optionally provide Boundary-Scan test access with third party Boundary-Scan tools. The user can make a decision to separate the PROM JTAG Boundary-Scan chain when using the Platform Flash PROMs. The JTAG CONFIG command pulses the $\overline{CF}$ pin connected to the FPGA PROG_B/PROGRAM_B/PROGRAM pin on some FPGAs which causes the JTAG TAP controller to reset. Refer to the FPGA data sheets or user guides for proper FPGA operation.

Driving an LED with the DONE Pin

Directly driving an LED with the DONE pin is not recommended. The direct connection from DONE to an LED provides less than optimal signals to the $\overline{CE}$ input of the PROM. An alternate method of driving an LED is shown in Figure 7-2. The LED is driven with a 330Ω pull-up resistor and the DONE output with a 1.5 kΩ pull-up resistor. The interconnection from LED to DONE is by a silicon diode. This circuit meets all specifications (relying on the known fact that the forward voltage of a Si-diode is lower than that of an LED).

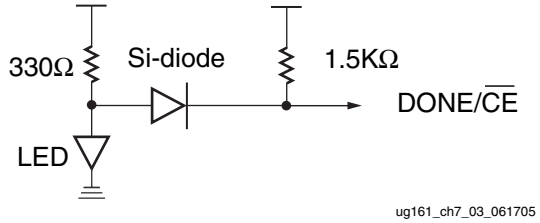


Figure 7-2: **Driving an LED with the DONE Pin**

Using the PROM CF Pin to Initiate FPGA Configuration

The Platform Flash PROM can initiate an FPGA configuration. When a JTAG CONFIG instruction is updated into the Platform Flash PROM, the Platform Flash PROM temporarily drives its $\overline{CF}$ pin Low followed with a very brief drive to a High. Combined with an external pull-up resistor on the $\overline{CF}$ pin, this generates a High-Low-High pulse out of the $\overline{CF}$ pin that can be used to pulse the FPGA $\overline{PROGRAM}$ (or PROG_B/PROGRAM_B) pin. The pulse to the FPGA PROG_B/PROGRAM_B/PROGRAM pin initiates the FPGA configuration sequence.

For the XCFxxP Platform Flash PROM, the $\overline{CF}$ pin has the added functionality of resetting the design revision selected when a new configuration sequence is initiated. The XCFxxP $\overline{CF}$ pin must always be connected to an external pull-up resistor to prevent the $\overline{CF}$ pin from floating to a Low which holds the XCFxxP PROM in reset. When the XCFxxP PROM stores multiple design revisions, the $\overline{CF}$ pin must be connected to the FPGA PROG_B/PROGRAM_B/PROGRAM pin to ensure that the PROM is reset to the selected design revision whenever the FPGA configuration is initiated via a pulse to the FPGA PROG_B/PROGRAM_B/PROGRAM pin.

For the XCFxxS Platform Flash PROM, the $\overline{CF}$ pin is a output-only pin and can be left unconnected if the function is not needed.

Migration from Legacy PROMs

When designing a system for migration between Xilinx PROM families, care must be given to the selecting package layout, clock frequency, and voltage differences (Table 7-1).

Table 7-1: **Platform Flash PROM Migration Considerations**

PROM Family	Density	V_{CCINT}	V_{CCJ}	V_{CCO}	Package	JTAG ISP Prog	Serial Config	Parallel Config	Max CLK Freq
XCFxxS	1 – 4 Mb	3.3V	2.5V – 3.3V	1.8V – 3.3V	VO20	Yes	Yes	–	33 MHz
XCFxxP	8 – 32 Mb	1.8V	2.5V – 3.3V	1.8V – 3.3V	VO48 FS48	Yes	Yes	Yes	40 MHz
XC18Vxx	512K – 1 Mb	3.3V	–	2.5V –3.3V	SO20 PC20 VQ44	Yes	Yes	Yes	33 MHz
	2 – 4 Mb	3.3V	–	2.5V –3.3V	VQ44 PC44	Yes	Yes	Yes	20 MHz
XC17Vxx	1.6 – 16 Mb	3.3V	–	–	VO8 SO20 PC20 VQ44 PC44	–	Yes	8–16 Mb	15 MHz
XC17SxxA	197K – 1.8 Mb	3.3V	–	–	PD8 VO8 SO20 VQ44	–	Yes	–	10 MHz
XC17xxE/L	65K – 4 Mb	3.3V 5.0V	–	–	PD8 VO8 SO8 SO20 PC20 VQ44 PC44	–	Yes	–	15 MHz
XC17Sxx	53K – 1 Mb	3.3V 5.0V	–	–	PD8 VO8 SO8 SO20	–	Yes	–	10 MHz

CLK Frequency Differences

The maximum clock (CLK) frequency of all families under the migration path must be considered. The chosen CLK frequency is important in systems that use a fixed external CLK source. On the other hand, if the FPGA Master-Serial mode or Master-SelectMAP mode is used, the configuration CLK frequency is user selectable by the Xilinx BitGen software's ConfigRate option. Thus, the configuration rate in the Master configuration modes can be adjusted late in the design cycle to suit the PROM family. The Platform Flash PROM family supports faster maximum configuration CLK frequencies than the XC1700 series of One Time Programmable (OTP) PROMs. Table 7-1 shows the maximum configuration CLK frequencies for each PROM family.

Voltage Supply and I/O Differences

The XCFxxS Platform Flash PROMs (1 Mb to 4 Mb) voltage supply is 3.3V. The XCFxxS PROM includes a separate supply (V_{CCO}) for its output buffer that can be powered at 1.8V to 3.3V. The families of the XC1700 series OTP PROMs span the 3.3V and 5V supply ranges. The XC18V00 ISP PROM is a 3.3V PROM and includes a separate supply (V_{CCO}) for its output buffers that can be powered at 2.5V or 3.3V.

For the Virtex® Series, Virtex-II FPGAs, and Spartan®-II/IIE FPGAs, the FPGA V_{CCO} pins for banks with configuration pins are recommended to be connected to 3.3V for I/O compatibility with the Xilinx PROMs.

The I/Os on each reprogrammable Platform Flash PROM are fully 3.3V tolerant. This allows 3V CMOS signals to connect directly to the inputs without damage. The core power supply (V_{CCINT}), JTAG pin power supply (V_{CCJ}), output power supply (V_{CCO}), and

external 3V CMOS I/O signals can be applied in any order. Additionally, for the XCFxxS PROM only, when V_{CCO} is supplied at 2.5V or 3.3V and V_{CCINT} is supplied at 3.3V, the I/Os are 5V tolerant. This allows 5V CMOS signals to connect directly to the inputs on a powered XCFxxS PROM without damage. Failure to power the PROM correctly while supplying a 5V input signal can result in damage to the XCFxxS device.

Special Signals

This section lists the differences in the PROM connections and signal operation from other Xilinx PROM families. These signals need to be considered when migrating from an older Xilinx PROM family to the Platform Flash PROM family.

BUSY Pin

The set of configuration signals for the Virtex series SelectMAP configuration mode includes the BUSY signal. When asserted High, the BUSY signal indicates that the PROM data source should hold the current data byte until BUSY is deasserted. The XCFxxP Platform Flash PROMs support the BUSY signal, but the XCFxxS PROMs do not support the BUSY signal. See the appropriate FPGA configuration user guide for the existence of the FPGA BUSY pin and the behavior of the BUSY pin during configuration.

Note: When configuring Virtex-II FPGAs or Virtex-II Pro FPGAs in SelectMAP mode from an encrypted bitstream, the XCFxxP PROM BUSY pin must monitor the FPGA BUSY pin. Virtex-II FPGAs and Virtex-II Pro FPGAs can assert their BUSY pins at lower CCLK frequencies than specified for non-encrypted bitstreams. See the Xilinx Answer Records for further information regarding the Virtex-II FPGA or Virtex-II Pro FPGA BUSY pin and encrypted bitstreams.

$\overline{CF}$ Pin

The Platform Flash PROMs support the $\overline{CF}$ pin. See "Using the PROM CF Pin to Initiate FPGA Configuration," page 95 for details on the Platform Flash PROM $\overline{CF}$ pin.

Design Revisioning Selection Pins ($\overline{EN\_EXT\_SEL}$, REV_SEL[0:1])

The XCFxxP Platform Flash PROMs include the $\overline{EN\_EXT\_SEL}$, REV_SEL0, and REV_SEL1 input pins used for design revisioning. These pins can be left floating or tied High externally if design revisioning is not utilized by the design. Design revisioning is not available in the XC1700 series OTP PROMs, the XC18V00 PROMs, or the XCFxxS Platform Flash PROMs.

CLKOUT

The XCFxxP Platform Flash PROMs also include the CLKOUT pin, which supports the advanced clocking options available for these devices. This output pin can be left unconnected when not utilized by the design. The CLKOUT pin is not available in the XC1700 series OTP PROMs, the XC18V00 PROMs, or the XCFxxS Platform Flash PROMs.

JTAG Port (TCK, TMS, TDI, and TDO)

The Platform Flash ISP PROMs support the IEEE 1149.1 Boundary-Scan Standard connections to the Platform Flash ISP PROM's JTAG port. The JTAG port is required to access the PROM for ISP programming and can optionally provide Boundary-Scan test access. The JTAG port is also available with the XC18V00 ISP PROMs. In-system programming via JTAG is not available in the XC1700 series OTP PROM family.

Package Migration from 17Vxx/18Vxx to XCF01S/02S/04S Devices

There are no common packages between the Legacy PROMs and the Platform Flash PROMs. Figure 7-3 shows a PC board layout migration from the Legacy PROM VQ44 package to the Platform Flash PROM VO20 package.

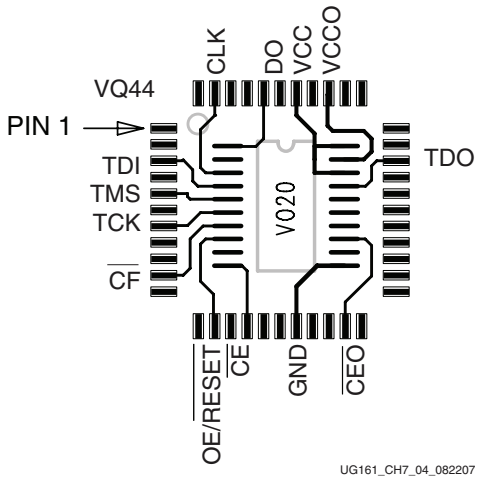


Figure 7-3: **Conversion from Legacy PROM VQ44 to Platform FLASH VO20**

Printed Circuit Board Considerations

Proper attention to printed circuit board (PCB) design can ensure a robust configuration circuit. It is highly recommended that users review the PCB Design Checklist at:

http://www.xilinx.com/products/design_resources/signal_integrity/si_pcbcheck.htm

The PCB Design Checklist provides a valuable set of fundamental PCB design considerations and guidelines. The considerations include PCB layout, power and ground planes, bypass capacitors, signal paths, and JTAG. The PCB Design Checklist provides guidelines for signal paths including a recommendation to simulate long or critical signal paths for transmission line effects. Transmission line effects can result in undesirable glitches on clock edges or edge-triggered resets such as the JTAG TCK clock, the configuration clock to or from the PROM CLK input or PROM CLKOUT output, PROM CE input, and PROM OE/RESET input. Transmission line effects can also result in unacceptable overshoot on any signal including PROM data outputs. Glitches and overshoot can be treated with a variety of termination techniques.

The JTAG bus provides an important programming and debug path to the Platform Flash PROM. The PCB design for the JTAG bus is critical. PCB design considerations for the JTAG bus are found in XAPP104, *A Quick JTAG ISP Checklist*.

Special PCB design consideration is required for the Master FPGA CCLK signal. For some FPGA families, the FPGA's internal configuration logic always sources the configuration clock from the FPGA CCLK pin in all configuration modes except the JTAG configuration mode. In the case of a master configuration mode, the applicable FPGAs drive a clock to the CCLK pin and use the clock from the CCLK pin to drive the internal configuration logic. For a PCB design with an applicable FPGA in a master configuration mode, good signal integrity must be ensured at the FPGA CCLK pin as well as the PROM CLK pin. Otherwise, glitches on the edges at the FPGA CCLK pin affect the FPGA's internal configuration logic. FPGA families that always source the configuration clock from the

CCLK pin include all Virtex FPGA families and the Spartan-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGA families.

The Spartan-3A, Spartan-3AN, Spartan-3A DSP, and Spartan-6 FPGA families are exempt from this special Master FPGA CCLK consideration because these FPGA families source the configuration clock directly from the internal master CCLK source rather than from the CCLK pin.

IBIS Models

IBIS models can be downloaded via the Xilinx Download Center at:

http://www.xilinx.com/support/download/index.htm

FPGA Design Considerations

After FPGA configuration, the FPGA can read user data from areas of the PROM that are not consumed by the FPGA bitstream. See XAPP694, *Reading User Data from Configuration PROMs*, for a reference on reading user data from a configuration PROM.

When the Platform Flash PROM is not used after FPGA configuration, these techniques can be applied to the FPGA design to ensure that the Platform Flash PROM and its signals remain in a quiescent, low-power state after the FPGA configuration process completes:

- When the FPGA is set to a master configuration mode and when parallel Thevenin termination is used on the FPGA CCLK signal, set the FPGA design to drive the CCLK pin to a valid logic High or Low. After the FPGA master mode configuration process completes, the CCLK pin goes to a high-impedance state by default. In a high-impedance state, a parallel Thevenin termination scheme can allow the CCLK signal to float to an intermediate state. The FPGA CCLK is driven to a valid logic High or Low to avoid the intermediate input signal level on the CCLK. For Virtex FPGA families, the CCLK pin can be driven from the design through the STARTUP primitive. (See the corresponding FPGA libraries guide for a description of the FPGA STARTUP primitive.) For most Spartan FPGA families, the CCLK pin becomes a standard, user-configurable I/O and can be driven directly from the design to a constant High or Low.
- For Spartan FPGAs, the FPGA design must explicitly put INIT_B pin into a high-impedance state. After configuration, the Spartan FPGA INIT_B pin becomes a user-configurable I/O. If left unused, the I/O is set to high-impedance with an internal pull-down resistor. See the BitGen **UnusedPin** option in UG628, *Command Line Tools User Guide*, for default settings of unused I/O pins.

  The strong internal pull-down resistor in the Spartan FPGA families can offset the typical, external pull-up resistor on the INIT_B signal resulting in an indeterminate signal level to the PROM OE/RESET input and can indicate a false configuration CRC error on the INIT_B pin. The design needs to explicitly make the INIT_B pin high-impedance to ensure that BitGen does not add an internal pull-down and that the typical, external pull-up can keep the INIT_B pin High. INIT_B can be explicitly put into high-impedance by instantiating an OBUFT primitive for the INIT_B pin with its 3-state control set to a constant High.

Appendix A

References

Best Practices for Platform Flash PROM Configuration

XAPP986, *Bulletproof Configuration Guide for Spartan®-3A FPGAs*

In-System Programming Solutions for Platform Flash PROM

ISE® Software Design Tools and iMPACT Software

ISE Software and iMPACT Software Manuals

Cables and Programming Solutions

Embedded, In-System Programming Solutions for Platform Flash PROM

XAPP058, *Xilinx In-System Programming Using an Embedded Microcontroller*

XAPP424, *Embedded JTAG ACE Player*

XAPP972, *Updating a Platform Flash PROM Design Revision In-System Using SVF*

XAPP975, *Low-Profile In-System Programming Using XCF32P Platform Flash PROMs*

Serial Vector Format (SVF) File

JEDEC Standard Test and Programming Language (STAPL) - JESD71

Using a Platform Flash PROM with an Embedded Processor

XAPP482, *MicroBlaze™ Platform Flash/PROM Boot Loader and User Data Storage*

XAPP938, *Dynamic Bus Mode Reconfiguration of PCI-X and PCI Designs*

XAPP483, *Multiple-Boot with Platform Flash PROMs*

Revision Managers for Safe Update Solutions

XAPP693, *A CPLD-Based Configuration and Revision Manager for Xilinx Platform Flash PROMs and FPGAs*

Storing and Retrieving Data from a Platform Flash PROM

XAPP694, *Reading User Data from Configuration PROMs*

XAPP544, *Using Xilinx XCF02S/XCF04S JTAG PROMs for Data Storage Applications*

Supplying Power to Platform Flash PROMs

XAPP389, *Powering CoolRunner-II CPLDs*

FPGA User Guides

UG002, *Virtex®-II Platform FPGA User Guide*

UG071, *Virtex-4 FPGA Configuration Guide*

UG191, *Virtex-5 FPGA Configuration User Guide*

UG332, *Spartan-3 Generation Configuration User Guide*

UG360, *Virtex-6 FPGA Configuration User Guide*

UG380, *Spartan-6 FPGA Configuration User Guide*

Device Reliability and Technology Information

UG116, *Device Reliability Report*