BPI Configuration Flash Design Guide

Red Rapids

797 North Grove Rd, Suite 101
Richardson, TX 75081
Phone: (972) 671-9570
www.redrapids.com

Red Rapids reserves the right to alter product specifications or discontinue any product without notice. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment. This product is not designed, authorized, or warranted for use in a life-support system or other critical application.

All trademark and registered trademarks are the property of their respective owners.

Table of Contents

List of Figures

1.0 Introduction

A byte peripheral interface (BPI) flash is used to store the FPGA bitstream that will be loaded automatically at power-up. This manual is directed at the FPGA developer that will load the FPGA or configuration flash with an application specific bitstream.

The latest product documentation and software is available for download from the Red Rapids website (www.redrapids.com).

1.1 Conventions

This manual uses the following conventions:

- Hexadecimal numbers are prefixed by "0x" (e.g. 0x00058C).
- *Italic* font is used for descriptive text found on GUI displays.
- Blue font is used for GUI button and menu option names.
- Active low signals are followed by _B, For example, RST_B.

☞	Text in this format highlights useful or important information.

!	Text shown in this format is a warning. It describes a situation that could potentially damage your equipment. Please read each warning carefully.

The following are some of the acronyms used in this manual.

- **BPI** Byte Peripheral Interface
- **BIT** File extension for FPGA bitstreams
- **MCS** File extension for flash PROM bitstreams
- **XDC** File extension for Xilinx design constraints

1.2 Revision History

Version	Date	Description
R02	6/25/2018	Added reference to different BPI flash device part numbers.
R01	11/23/2014	Replaced ISE references with Vivado references.
R00	5/8/2014	Initial release.

2.0 Hardware Architecture

Figure 2-1 illustrates the connection between the FPGA and the BPI flash component. Maximum configuration speed is achieved using a 16-bit data interface with the synchronous read mode feature of the flash. A 100 MHz crystal oscillator is connected to the external master configuration clock (EMCCLK) pin of the FPGA. This clock source drives the internal configuration engine of the FPGA to load the bitstream at power-up.

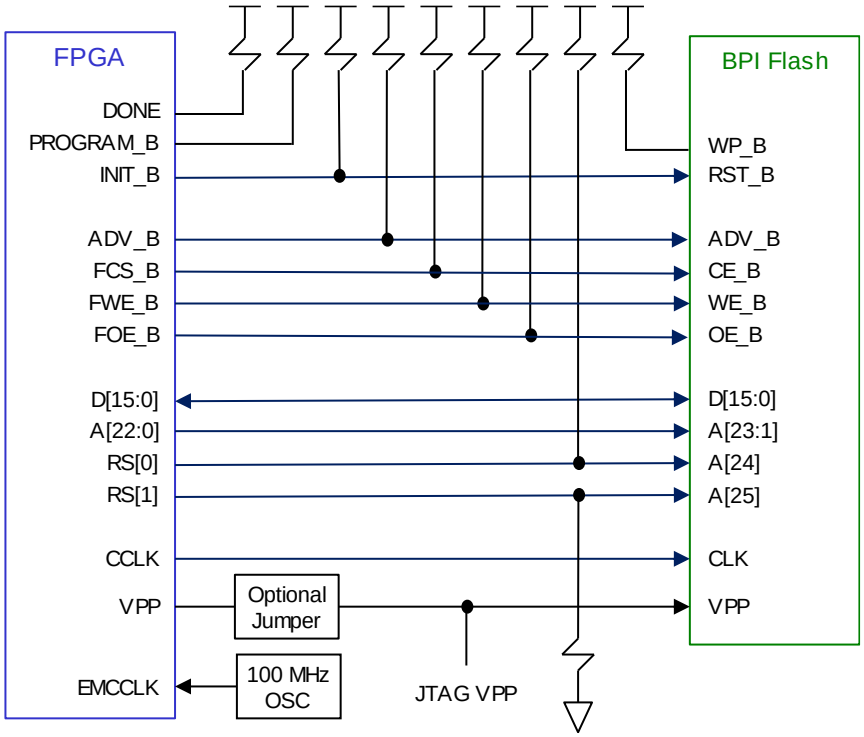

Figure 2-1 BPI Configuration Hardware Interface

The write protect (WP) pin of the flash is pulled up to disable block lock down. The program and erase operations of the flash can be enabled or disabled through the programming voltage (VPP) pin. The VPP input is always driven high when the JTAG programming cable is physically connected to the board. This allows the flash to be programmed indirectly through the FPGA using a Xilinx JTAG programming cable and the iMPACT™ configuration tool. An optional hardware jumper also allows the FPGA direct control of the programming voltage. The jumper can be eliminated as a product build option if a customer wants to disable flash programming from the FPGA application logic for security purposes.

The flash is large enough to store multiple bitstreams. The exact number depends on the size of the flash. The Xilinx multiboot and fallback features are supported through the FPGA RS[1:0] pin connections to the upper address bits of the flash. Refer to the Xilinx Configuration User Guide for further details.

The part number of the BPI flash device can be obtained from the model number options decoder for a specific product.

2.1 Device Configuration Bitstream Settings

There are two device configuration bitstream settings that must be changed from the default values to accommodate the BPI flash.

Setting	Default	BPI Value
BITSTREAM.CONFIG.BPI_SYNC_MODE	Disable	Type1
BITSTREAM.CONFIG.EXTMASTERCCLK_EN	Disable	div-1

These settings are included in the Xilinx Design Constraints (XDC) file supplied with each Red Rapids product. Additional settings must be changed to support multiboot and fallback configurations. Refer to the Xilinx Configuration User Guide for further details.

3.0 MCS File Generation

Before the flash can be programmed, an MCS file must be created from the BIT file produced by Vivado™. Launch the Vivado™ Hardware Manager as shown in Figure 3-1 and enter the following command into the Tcl Console:

write_cfgmem -format mcs -interface bpix16 -size 64 –loadbit
"up 0x0 SigFPGA_Top.runs/impl_1/SigFPGA_Top.bit" -file SigFPGA_Top.mcs

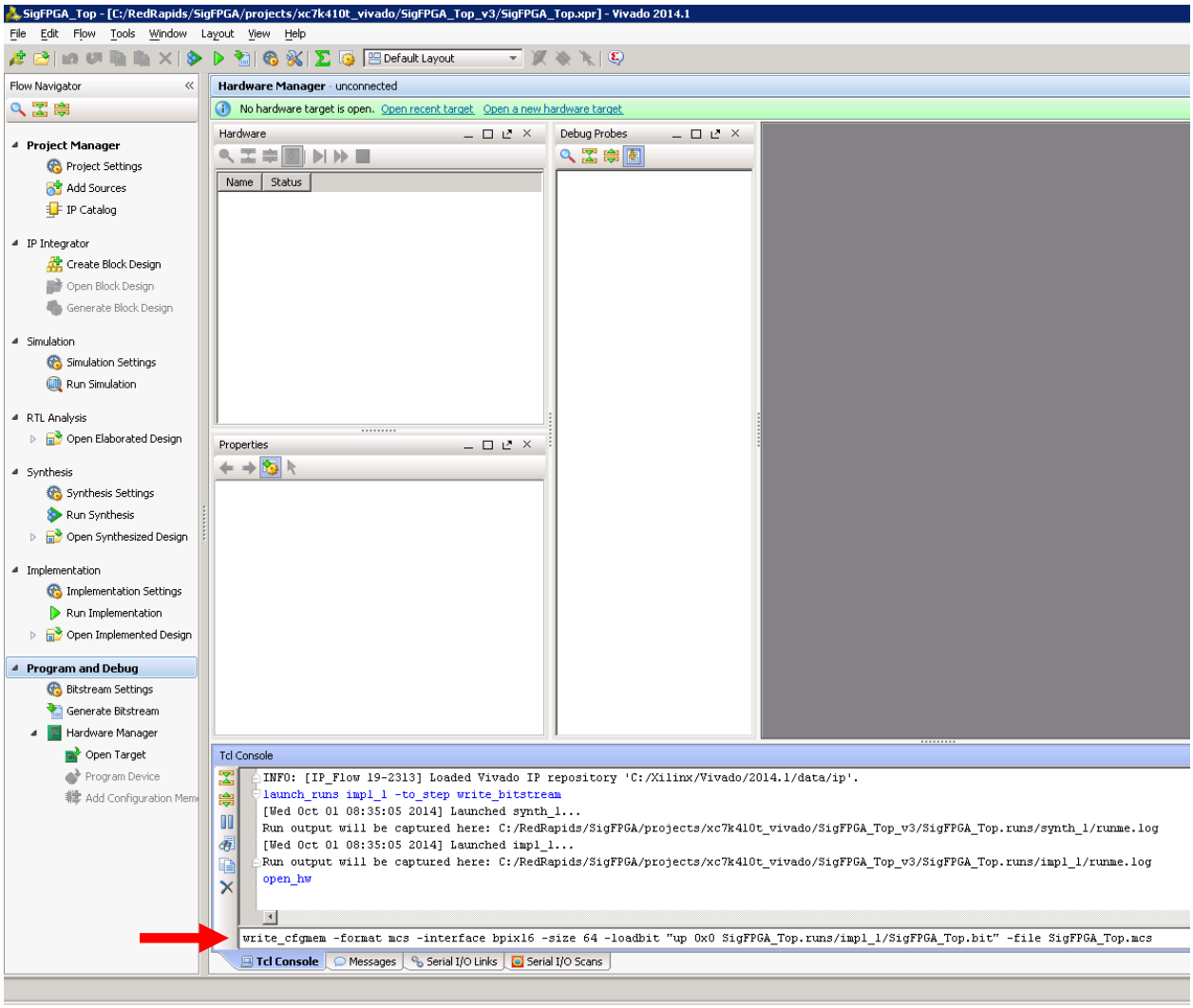

Figure 3-1 MCS File Generation From Vivado™ Hardware Manager

The Tcl command assumes that Vivado™ is executed from the root of the project directory and the BIT file is located in subdirectory SigFPGA_Top.runs/impl_1. The resulting MCS file will be created at the root of the project directory. Refer to the Xilinx Vivado Design Suite User Guide: Programming and Debugging for further details.

4.0 JTAG Programming With Vivado™

The Vivado™ tool is used to directly program the FPGA from a BIT file or indirectly program the BPI flash from an MCS file. A Xilinx programming cable is needed to connect the host computer to the JTAG connector on the board. The Xilinx programmer plugs into a USB port on the host computer and a cable provided by Red Rapids connects the programmer to the JTAG port on the product.

Launch the Vivado™ Hardware Manager with the Xilinx programmer connected to the board. Open a hardware target from the Hardware Manager. This will cause the JTAG cable to scan the chain and identify the FPGA device as shown in Figure 4-1.

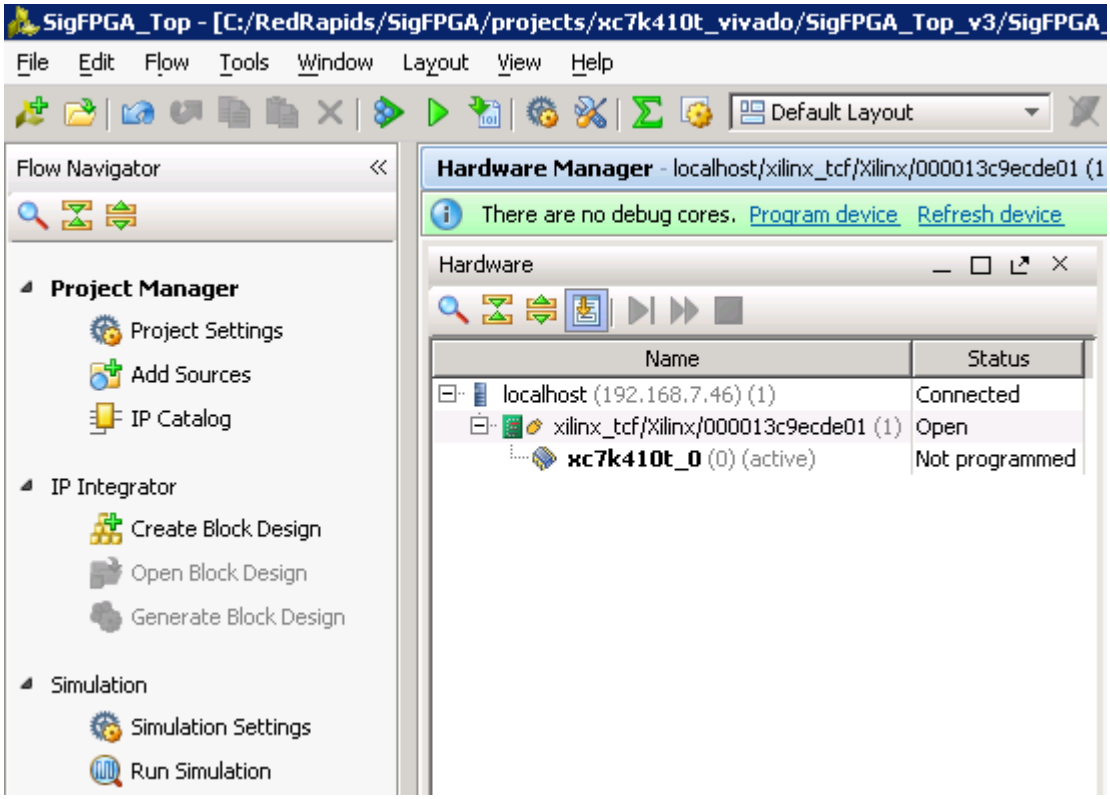

Figure 4-1 Initialize JTAG Chain

Right click the FPGA device name and select Program Device… to directly load a BIT file to the FPGA. A *Program Device* window will open as shown in Figure 4-2. The location of the Bitstream file and any Debug Probes file are entered in this window. The default values will point to the correct location if Vivado™ is executed from the root of the project directory. Click Program to continue.

The device status will change from *Not Programmed* to *Programmed* if the BIT file is loaded successfully.

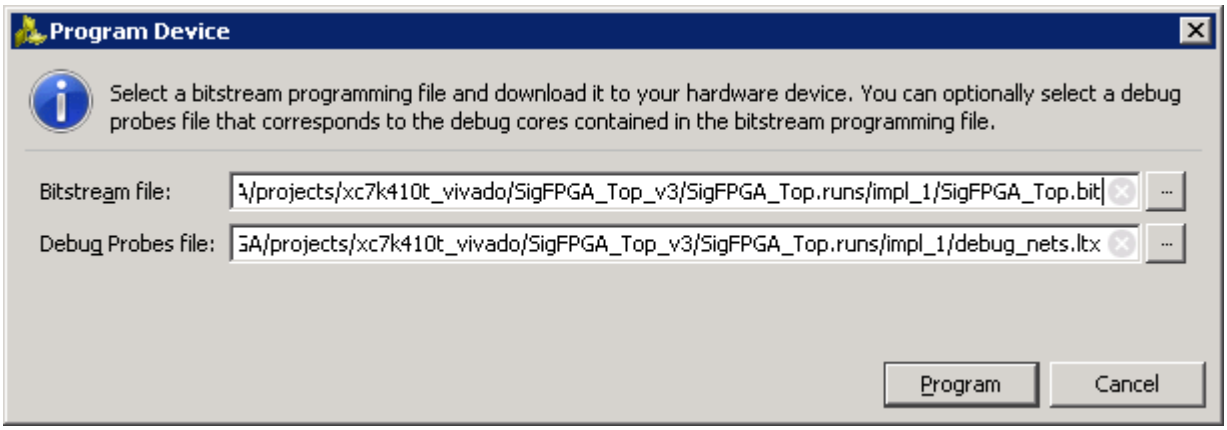

Figure 4-2 Program Device Window

Right click the FPGA device name and select Add Configuration Memory Device… to attach the flash to the FPGA. An *Add Configuration Memory Device* window will open as shown in Figure 4-3. Select the mt28gu512aax1e-bpi-x16 or mt28gu256aax1e-bpi-x16 device name and click OK to continue.

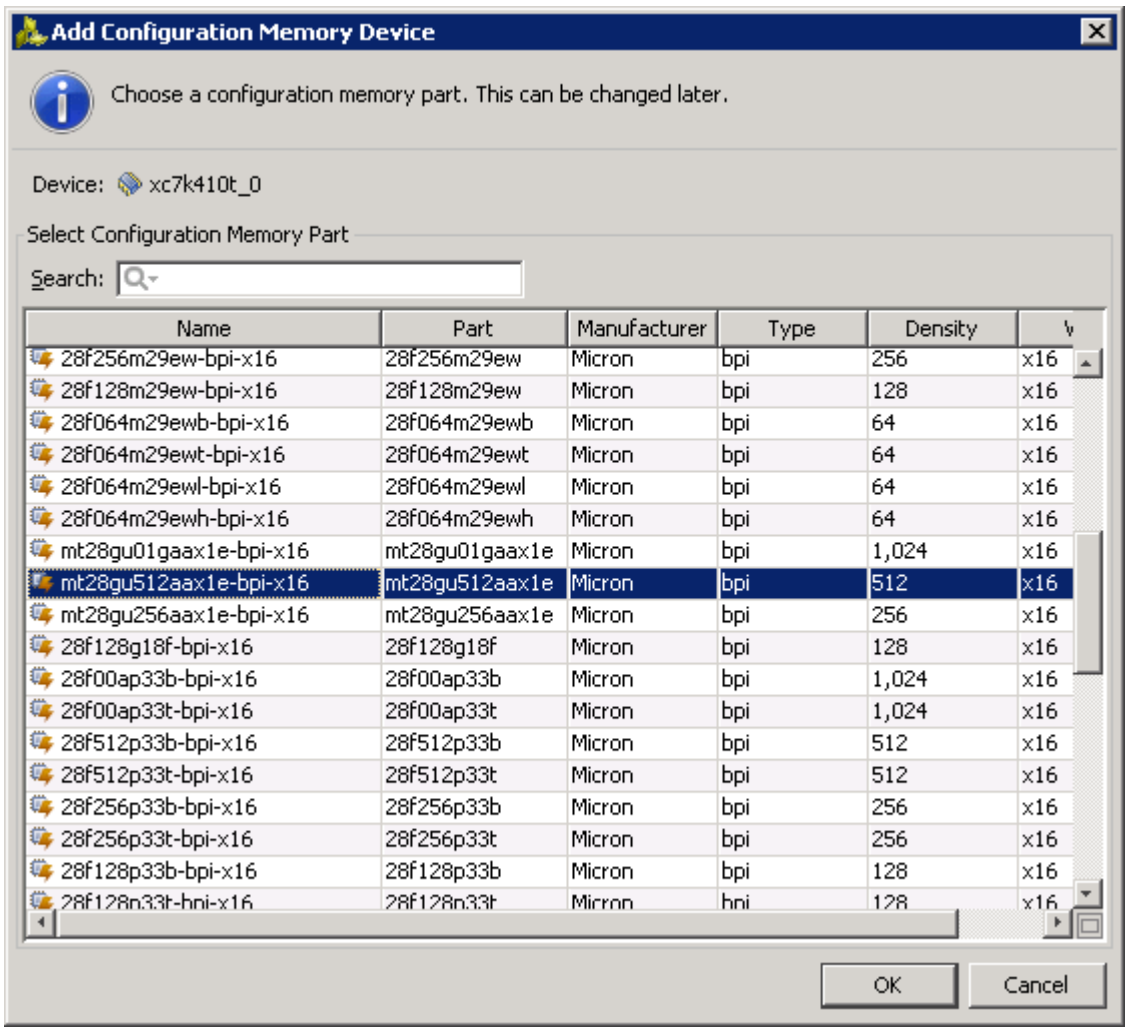

Figure 4-3 Add Configuration Memory Device Window

The flash device will now be listed as an attachment to the FPGA device in the Hardware Manager as shown in Figure 4-4.

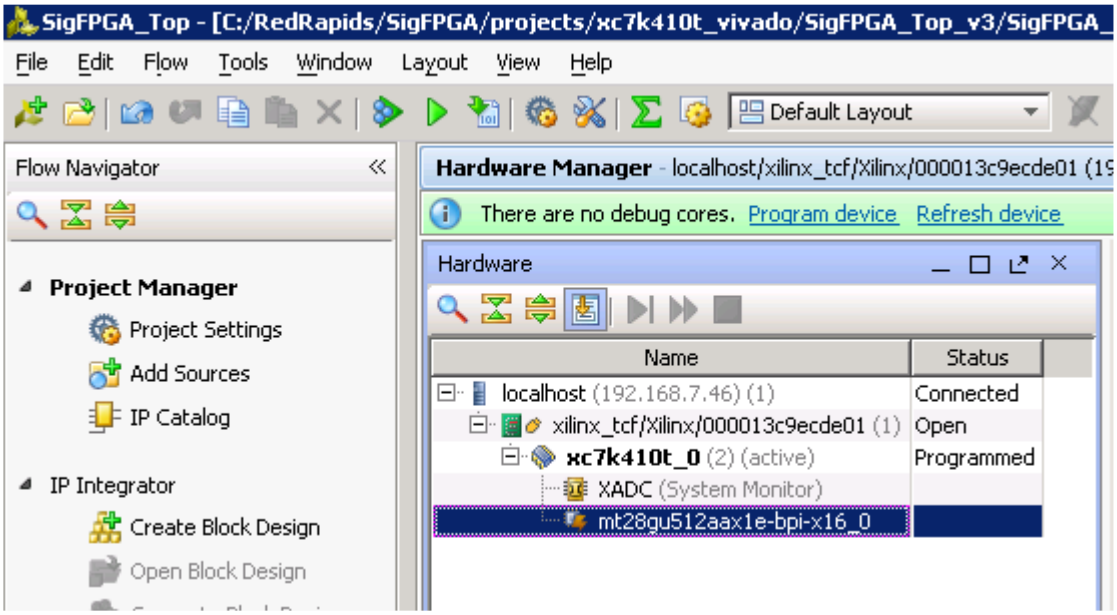

Figure 4-4 Add Configuration Memory Device Window

Right click the flash device name and select Program Configuration Memory Device… to indirectly load an MCS file to the flash. A *Program Configuration Memory Device* window will open as shown in Figure 4-5. The location of the MCS file must be entered into the Configuration File: dialog box, click OK to continue.

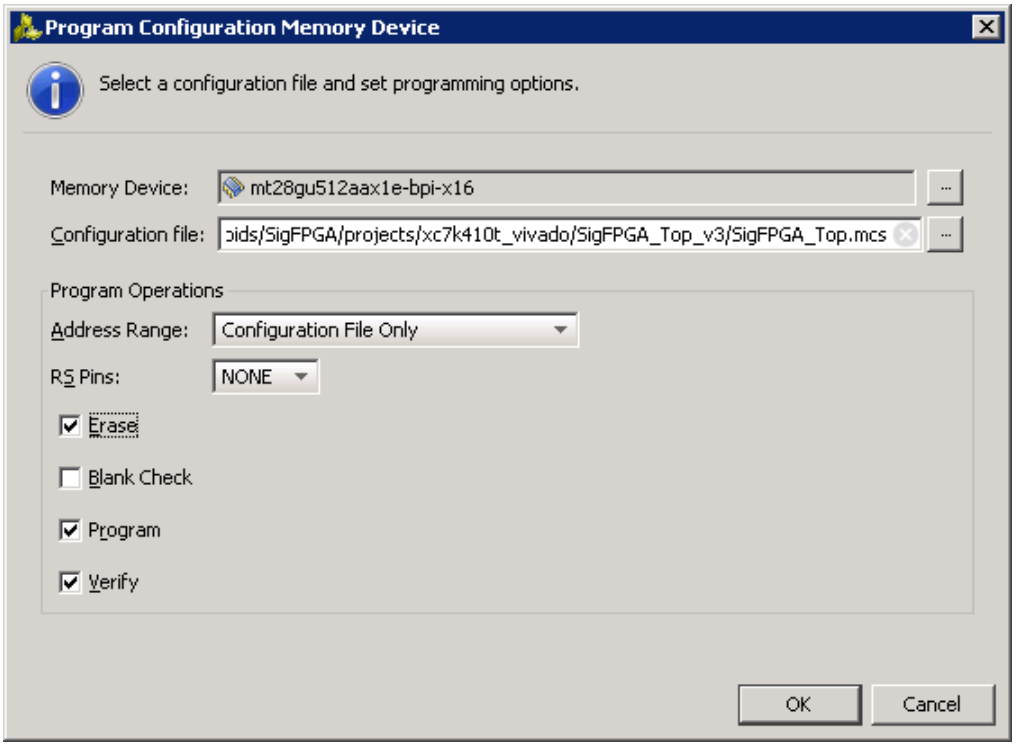

Figure 4-5 Program Configuration Memory Device Window

A sequence of three status windows will be displayed as the flash is programmed. Each window displays progress toward the three steps of erasing, programming, and verifying the flash contents. All three windows are shown in Figure 4-6.

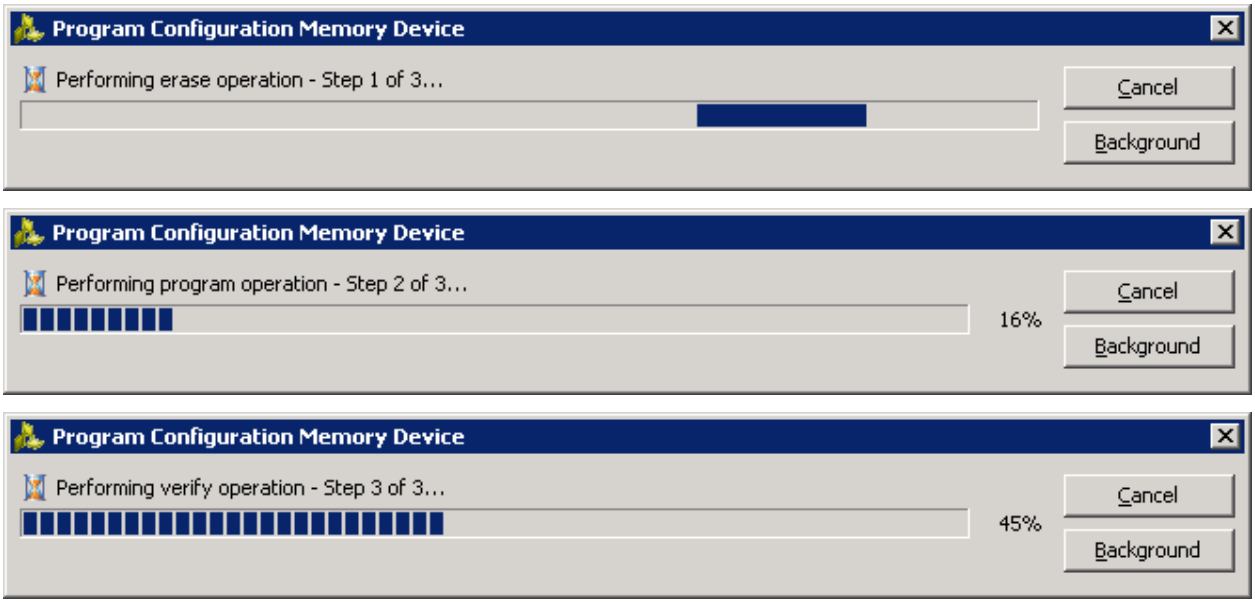

Figure 4-6 Three Flash Programming Status Windows