XAPP586 (v1.2) January 31, 2014

Using SPI Flash with 7 Series FPGAs

Author: Arthur Yang

Summary

This application note describes the advantages of selecting a serial peripheral interface (SPI) flash as the configuration memory storage for the Xilinx 7 series FPGAs and the details for implementing the solution. This document includes the required connections between the FPGA and the SPI flash memory and the details necessary to select the proper SPI flash.

Programming the SPI Flash In-System provides details about using the ISE® Design Suite for in-system programming of the SPI flash via the FPGA. This allows for configuration flexibility during the debugging stages of development. The 7 series FPGAs can also be programmed in-system using the Vivado® Design Suite. More information on the Vivado tools can be found at www.xilinx.com. The designer should be familiar with UG470, *7 Series FPGAs Configuration User Guide* that contains additional information on FPGA configuration and details on other configuration methods.

Introduction

This application note addresses the two flows shown in Figure 1:

- Indirect SPI flash programming using the ISE Design Suite iMPACT tools.
- SPI flash configuration that delivers the FPGA configuration bitstream stored in a SPI flash memory to the 7 series FPGAs.

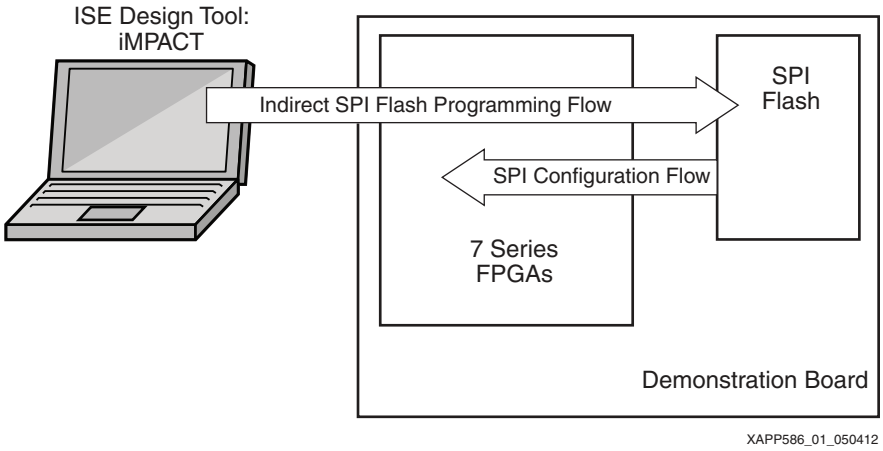


Figure 1: **SPI Flash Configuration and Indirect Programming Flows**

Xilinx FPGAs require that a configuration bitstream is delivered at power-up. The SPI flash memories use a 4-wire synchronous serial data bus. The SPI flash configuration requires only four pins, which allows 1- or 2-bit data width for delivery of the configuration bitstream. Newer SPI flash devices offer the option to use six pins to enable 4-bit data width, thereby decreasing configuration time appropriately. FPGA configuration via the SPI interface is a very low pin count configuration solution and many vendors have devices in a large range of density options.

Other options for FPGA configuration, such as a byte peripheral interface (BPI) parallel NOR flash, supports a wider configuration data bus that allows for faster configuration at power-up, however this mode requires a minimum of 25 pins.

Because parallel NOR flash devices have higher density options than SPI flash, BPI flash should be considered if the application requires large amounts of nonvolatile data storage or if several FPGA bitstreams need to be stored.

Xilinx also provides the ability to program the SPI flash in-system using the existing configuration connections between the SPI flash and the FPGA. The Xilinx iMPACT programming tool uses JTAG to configure the FPGA to enable a path between the configuration cable and the SPI flash. This allows design flexibility in a lab environment to easily program new configuration bitstreams into the SPI flash without removing the flash from the board and using an external desktop programmer.

The sections in this document are:

- SPI Flash Basics: Review of the SPI flash pin functions and device features.
- SPI Flash Configuration Interface: Details on the FPGA configuration interface with the SPI flash.
- SPI Flash Configuration Time: Details the steps for determining the maximum clock frequency.
- SPI Flash Configuration Options: Describes the options for generating the bitstream.
- Preparing the SPI Flash Programming File: Provides instructions to generate a SPI flash data file.
- Programming the SPI Flash In-System: Provides instructions to program the SPI flash.

SPI Flash Basics

This section reviews the SPI flash pins and their connections to 7 series FPGAs. Details about the SPI flash configuration options, such as density selection, data width, and FPGA configuration time, are also included.

Figure 2 shows the basic connectivity between 7 series FPGAs and the SPI flash with a x1 data width. The read and address instructions are sent from the FPGA to the SPI flash via the master-out-slave-in (MOSI) pin. The data is returned from the SPI flash via the master-in-slave-out (MISO) pin. SCK is the clock pin and SS is the active-Low slave select pin. A x2 data width has the same connections, however the MOSI becomes bidirectional and is used as an additional data pin.

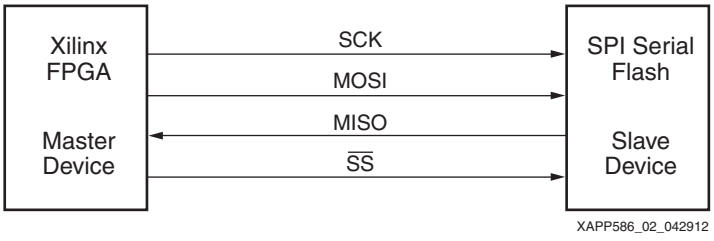


Figure 2: **Basic SPI Flash to FPGA Connections—x1 Data Width**

In addition to the pins described above, the SPI flash can have additional pins that can be used to control other special functions. These additional pins can vary with the SPI flash vendor, however two common special function pins are *hold* and *write protect*. Newer SPI flash devices enable these hold and write protect pins with a dual function of additional data output pins to increase the data bus up to 4 bits.

Table 1: **SPI Flash Pin Names**

Pin Names Used in This Document	Alternate Pin Names Used by Other Vendors	Pin Function
SCK	C, CLK	Clock for SPI flash instructions and data
MOSI	DQ0, DI, SI, IO0	Master out; slave in. Can be an additional data pin in x2 or x4 output modes
MISO	DQ1, IO1, SO, DO	Master in; slave out
$\overline{SS}$	S/, CS/	Slave select
$\overline{HOLD}$	DQ3, IO3	Hold or pause without deselecting the device. Can be an additional data pin in x4 output mode.
$\overline{W}$	DQ2, WP/, IO2	Write protect portions of the SPI flash memory. Can be an additional data pin in x4 output mode.

Selecting a SPI Flash

The first criteria in selecting a SPI flash is density. For many designs this means selecting a flash device that is large enough to store the configuration bitstream of the target FPGA. For some designs, other considerations narrow the options of which flash to use, such as the need to store multiple bitstreams, or have a daisy chain of FPGAs to be configured, or configuration speed.

The minimum density required is always the size of the FPGA configuration bitstream. See UG470, *7 Series FPGAs Configuration User Guide* for details. If the design requires multiple bitstreams, multiply the size of the bitstream accordingly. The Xilinx tools allow bitstream compression, however it is not recommended to rely on compression when determining SPI flash size because compression varies greatly with the user's design and is not predictable.

Some designs require the FPGA to configure in a specified amount of time. In this case, the designer should consider using a SPI flash that allows for the fastest read-clock rate and ensuring the support of x4 data width read operations (sometimes called quad output fast read in SPI flash data sheets).

The designer also must consider the I/O voltage compatibility. The Artix™-7 and Kintex™-7 families support configuration I/O voltages up to 3.3V and the Virtex®-7 family supports up to 1.8V. The SPI flash vendors generally use the same voltage supply for the core voltage and the I/O voltage. However, some vendors can use a separate I/O voltage pin. The differences in the type of voltage supplies affect the ability to use different vendors as a second source.

The list of devices that are tested and supported by Xilinx tools can be found at:

http://www.xilinx.com/cgi-bin/docs/rdoc?v=latest_ise;d=isehelp_start.htm;a=pim_c_introduction_indirect_programming.htm

SPI Flash Configuration Interface

Figure 3 shows the pins of the FPGA required for SPI flash configuration. Many of these pins are also required for other configuration methods and are not specific to SPI flash configuration.

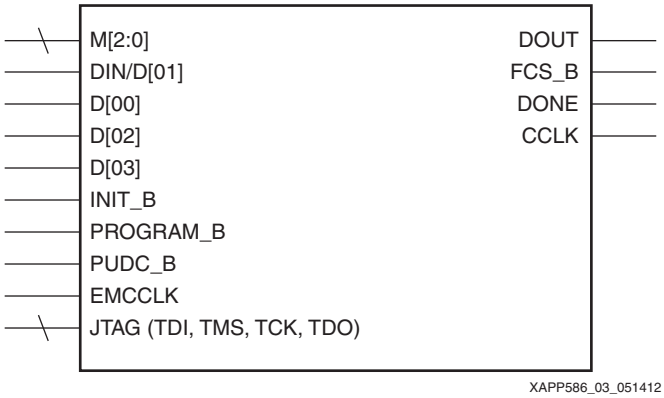


Figure 3: **FPGA SPI Flash Configuration Interface Block Diagram**

Table 2 details the functions of the FPGA pins during SPI flash configuration. In addition to the pins mentioned in the SPI Flash Basics section, other configuration interface signals are shown which give status information and control of FPGA configuration.

Table 2: **SPI Flash Configuration Pins**

FPGA Pin Name	FPGA Direction	Dedicated or Dual Purpose	Description
M[2:0]	Input	Dedicated	Determines the FPGA configuration mode. M[2:0] = 001 for master SPI flash mode. Connect each mode pin either directly, or via a 1 kΩ (or stronger) resistor, to VCCO_0 or GND.
DIN/D[01]	Input	Dual-Purpose	Receives data from the SPI flash MISO pin. In x1 mode, this is the only data input pin to the FPGA.
D[00]	Input/Output	Dual-Purpose	At the start of FPGA configuration, this pin drives the SPI flash's MOSI pin and delivers a read instruction and the address. In x1 mode, this pin is output only. In x2 and x4 data width modes, this pin is bidirectional and receives data from the SPI flash.
D[02]	Input	Dual-Purpose	Receives data bit 2 from the SPI flash in x4 data width mode.
D[03]	Input	Dual-Purpose	Receives data bit 3 from the SPI flash in x4 data width mode.
INIT_B	Bidirectional, Input, Output, Open-drain	Dedicated	Driven Low during FPGA power-up, indicating the FPGA is performing self-initialization prior to initiating configuration. After self-initialization is complete, and before the mode pins are sampled, this pin can be externally driven Low to delay configuration. After mode pins are sampled, INIT_B becomes open drain. During configuration bitstream loading, this pin acts as an indicator for CRC error.
PROGRAM_B	Input	Dedicated	Active-Low asynchronous full-chip reset.
PUDC_B	Input	Dual-Purpose	Controls I/O (except bank 0 dedicated I/Os) pull-up resistors during configuration. This pin must be externally terminated. 0 = Pull-up resistors during configuration 1 = 3-state output during configuration
EMCCLK	Input	Dual-Purpose	An input for supplying an external configuration clock (optional). This clock is then internally routed to the CCLK FPGA pin.
CCLK	Input/Output	Dedicated	Initial configuration clock source for all configuration modes except JTAG. Drives the SCK pin of the SPI flash.
FCS_B	Output	Dual-Purpose	Drives the SPI flash SS/ pin Low during configuration to enable the SPI flash.

Table 2: **SPI Flash Configuration Pins *(Cont'd)***

FPGA Pin Name	FPGA Direction	Dedicated or Dual Purpose	Description
DOUT	Output	Dual-Purpose	Only used in x1 SPI flash configuration modes when daisy chaining multiple FPGAs. See UG470, *7 Series FPGAs Configuration User Guide* for more information on daisy-chain configuration.
DONE	Output/ Open-Drain	Dedicated	Active-High signal indicating configuration is complete. 0 = FPGA not configured 1 = FPGA configured
CFGBVS	Input	Dedicated	For the Artix-7 and Kintex-7 families, this pin determines the voltage standard supported in the configuration I/O banks. For the Virtex-7 family, the only allowable configuration voltage is 1.8V or less. See the Configuration Bank Voltage Select section in UG470, *7 Series FPGAs Configuration User Guide* for additional information. 1 = 2.5V or 3.3V 0 = 1.8V or less
TDI	Input	Dedicated	JTAG test data input port(1)
TMS	Input	Dedicated	JTAG test mode select input port(1)
TCK	Input	Dedicated	JTAG test clock input port(1)
TDO	Output/ open-drain	Dedicated	JTAG test data out port(1)

Notes:

1. JTAG pins are optional for the SPI flash configuration interface, but are required for indirect SPI flash programming.

Figure 4 and Figure 5 are nearly identical. Figure 4 illustrates the connections for a SPI flash configuration solution in x1 or x2 data width mode. Figure 5 illustrates the connections for a SPI flash configuration solution in x4 data width mode. The only difference between Figure 4 and Figure 5 is the handling of the $\overline{HOLD}$ and the $\overline{W}$ pins, which are terminated in x1 and x2 data width modes and are connected to the 7 series FPGAs in x4 data width mode.

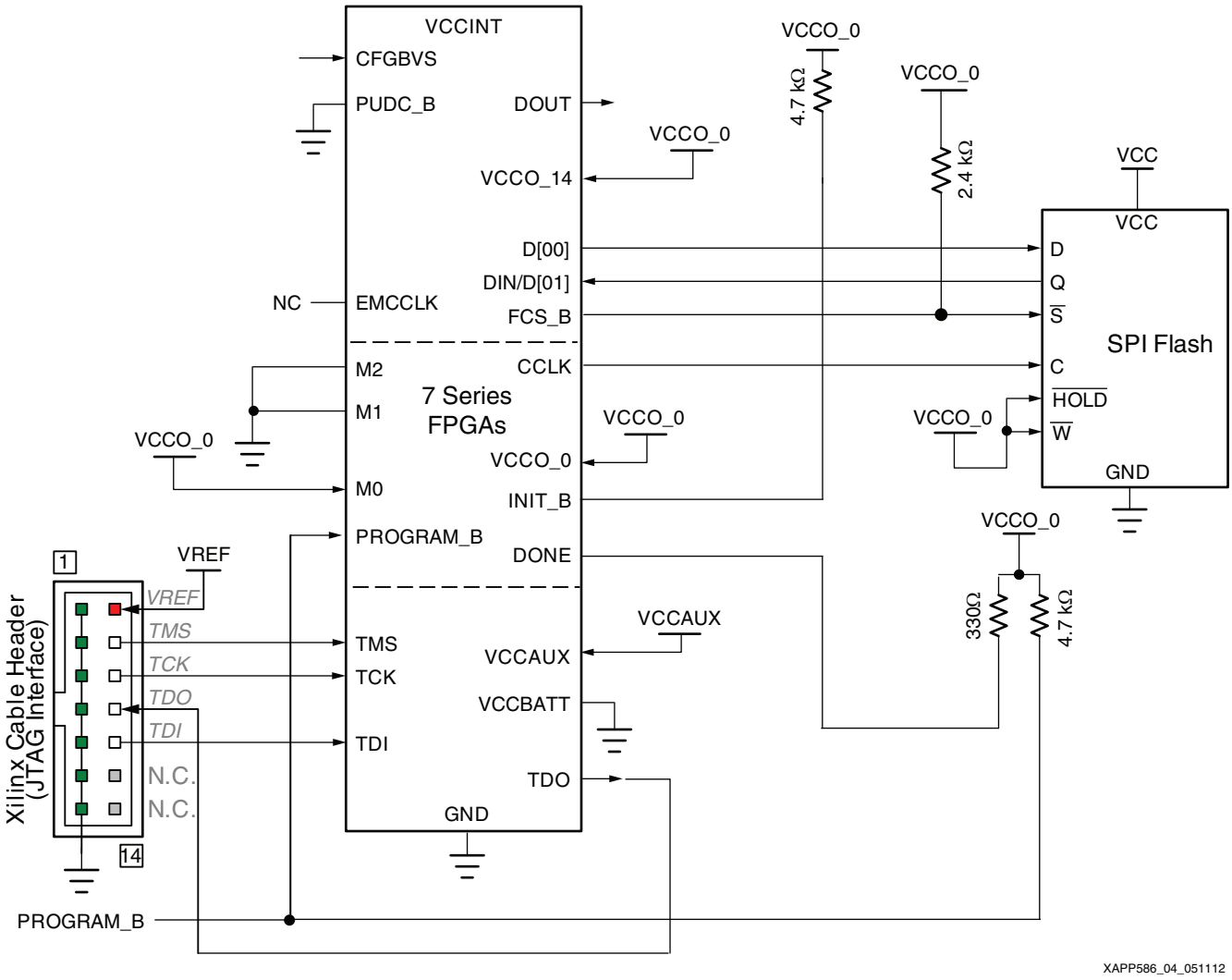


Figure 4: **SPI Flash x1/x2 Configuration Schematic**

Notes relevant to Figure 4, SPI flash x1/x2 configuration schematic:

1. DONE is by default an open-drain output. An external pull-up resistor is recommended.
2. INIT_B is a bidirectional open-drain pin. An external pull-up resistor is required.
3. CCLK signal integrity is critical.
4. Series resistors should be considered for the datapath from the SPI flash to the FPGA to minimize overshoot. The proper resistor value can be determined from simulation.
5. The VCCO supply of the 7 series FPGAs must be compatible with the VCC of the SPI flash.
6. VCCBATT is the power source for the AES key stored in SRAM. For details about AES encryption, see UG470, *7 Series FPGAs Configuration User Guide*.

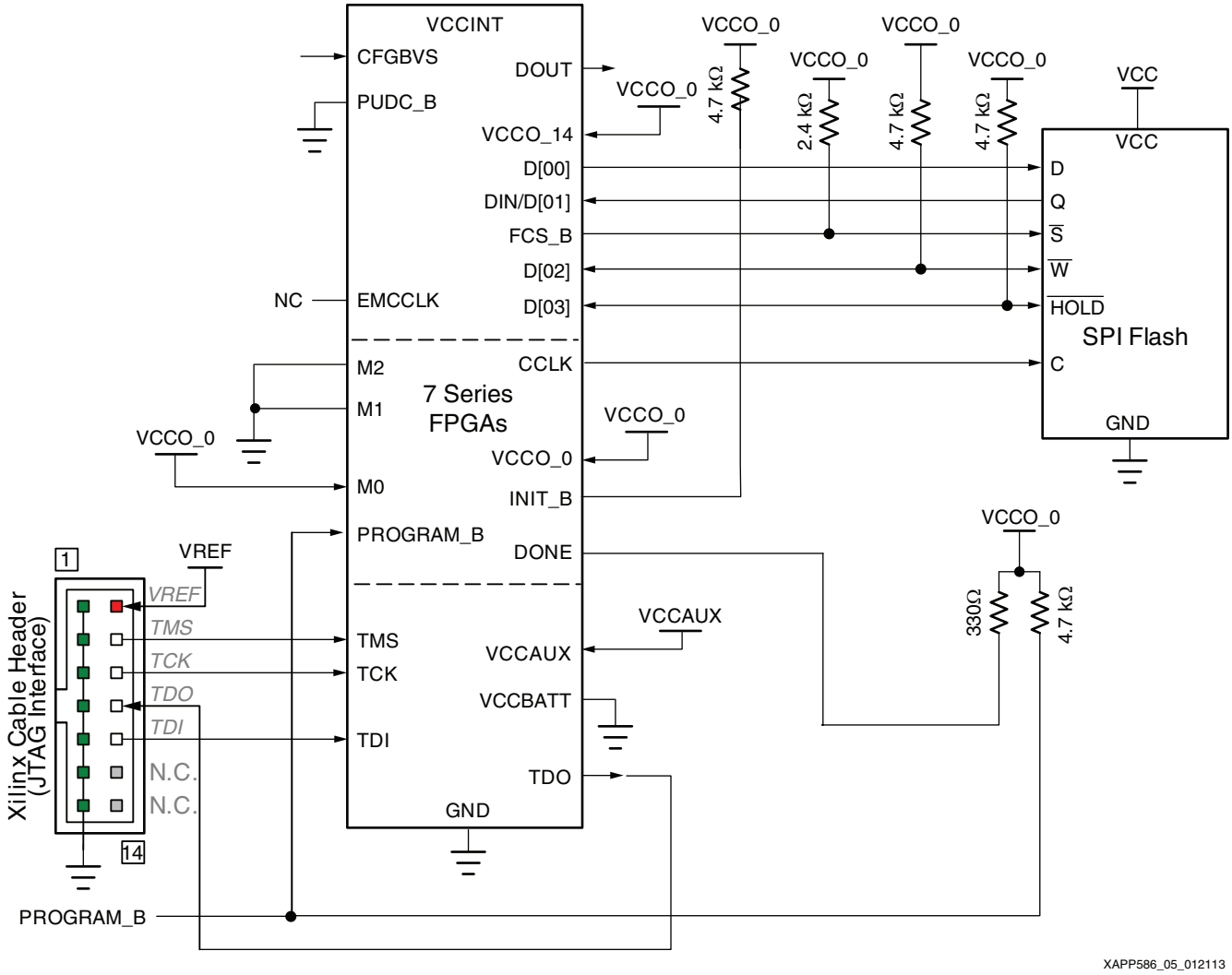


Figure 5: **7 Series FPGA SPI Flash x4 Configuration Schematic**

Notes relevant to Figure 5 SPI flash x4 configuration schematic:

1. DONE is by default an open-drain output. An external pull-up resistor is recommended.
2. INIT_B is a bidirectional open-drain pin. An external pull-up resistor is required.
3. CCLK signal integrity is critical.
4. Series resistors should be considered for the datapath from the SPI flash to the FPGA to minimize overshoot. The proper resistor value can be determined from simulation.
5. The VCCO supply of the 7 series FPGAs must be compatible with the VCC of the SPI flash.
6. VCCBATT is the power source for the AES key stored in SRAM. For details about AES encryption, see UG470, *7 Series FPGAs Configuration User Guide*.

Power-On Considerations for SPI Flash

At power-on, a race condition exists between the FPGA and the SPI flash. After the FPGA completes a self-initialization, it transmits a read command to the SPI flash to retrieve the configuration data, at which time the SPI flash must be ready to respond to this command. Refer to the specifications in the SPI flash data sheets for the time required for the flash to complete its own self-initialization (see the References section for the list of SPI flash data

sheets). In general the self-initialization time (also called power-on reset) of the 7 series FPGAs is an order of magnitude (milliseconds) more than the SPI flash (hundreds of microseconds), however the designer should evaluate the time. This is certainly more important if the SPI flash and the FPGA are on different supply rails.

Configuring the FPGA from SPI Flash

After the FPGA finishes self-initialization, INIT is released and the FPGA samples the mode pins (M[2:0]) to determine which configuration mode to use. With the mode pins M[2:0] = 001, the FPGA then begins to output clocks on CCLK at a frequency of approximately 3 MHz. Shortly afterwards, FCS_B drives Low, followed by the OPCODE for a x1 fast-read instruction and address on the D[00] pin as shown in Figure 6.

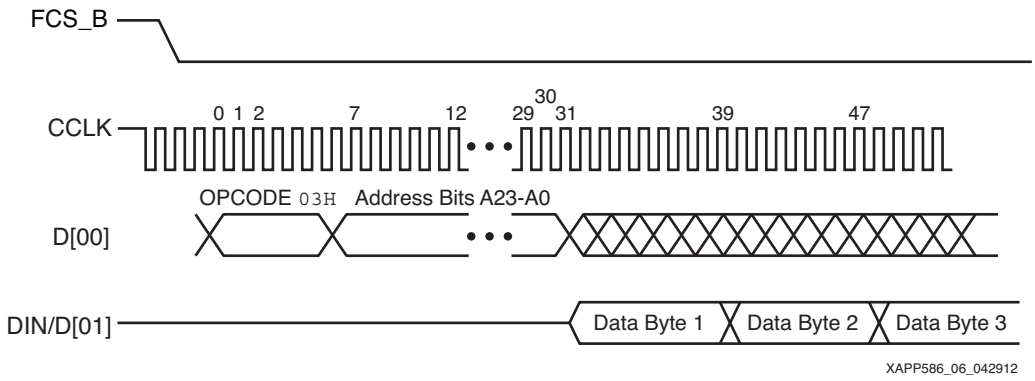


Figure 6: **7 Series FPGA SPI Flash Timing**

Data is initially transmitted from the SPI flash to the FPGA in x1 mode. The commands to switch to an external clock, x2 or x4 bus width, or other options are all contained within the early portion of the bitstream. After reading these options, the FPGA makes mid-configuration adjustments.

The default behavior is for data to be output from the SPI flash on the falling edge of CCLK and captured by the FPGA on the rising edge of CCLK. The default behavior can be changed to capture on the falling edge by enabling the SPI_FALL_EDGE BitGen option.

SPI Flash Configuration Time

Equation 1 should be used to determine the maximum frequency that the SPI flash can safely operate and still deliver the bitstream reliably. The SPI flash delivers data on the falling edge of the clock. The default for 7 series FPGAs is to capture data on the rising edge of the clock. For the equation below to be true, it is assumed that the SPI flash configuration option that enables the FPGA to capture data on the falling edge is enabled (-g spi_fall_edge:yes). This allows the full clock cycle to be utilized, therefore higher frequencies can be reached.

Equation 1 describes the SPI flash configuration frequency calculation.

$$\text{Maximum configuration clock frequency} = \frac{1}{\text{Flash clock to out}(T_{SPITCO}) + \text{FPGA data setup}(T_{SPIDDC})} \quad \textit{Equation 1}$$

In the 7 series FPGAs, the frequency tolerance of the internal oscillator (fMCCKTOL) is significant. If minimum configuration time is critical, it is recommended the designer use an external clock (EMCCLK).

After the optimum configuration rate is determined, the designer needs to divide the total bitstream size by the configuration rate to determine the total configuration time in x1 mode. If using x2 or x4 data widths, divide by the width.

Configuration Clock Calculation Example

This section demonstrates the steps to determine the maximum operating configuration frequency for the SPI flash solution.

The SPI flash selected is N25Q128A13 and the target is a Kintex-7 XC7K325T FPGA.

The SPI flash clock to out, per the SPI flash data sheet, has multiple values depending on VCC and the capacitance on the output pin.

These values are as fast as 6.5 ns and as slow as 8 ns. In this example, the value of 7.65 ns represents the SPI flash operating at the high end of the operating voltage with a load of 30 pF or less.

The FPGA setup time for a Kintex-7 XC7K325T FPGA is 3.0 ns (for the current value, refer to DS182, *Kintex-7 FPGAs Data Sheet*).

1 / (7.65 ns + 3.0 ns) yields a clock frequency of ~93.897 MHz.

The designer should consider using the FPGA's internal oscillator and the closest value to 93.9 MHz is 66 MHz. However, the frequency tolerance (fMCCKTOL) for the XC7K325T is ±50% (for the current value, refer to DS182, *Kintex-7 FPGAs Data Sheet*) so this clock frequency could potentially be (66 MHz x 1.5) = 99 MHz, which would be too fast for the calculated maximum.

The next fastest configuration rate is 50 MHz, which has a maximum frequency of (50 MHz x 1.5) = 75 MHz. This rate is well below the calculated maximum and nominally operates at 50 MHz, which is well below the desired 93.9 MHz.

The bitstream of a Kintex-7 XC7K325T FPGA is 91,548,896 bits.

91,548,896 / 50,000,000 = 1.83 seconds to configure XC7K325T in x1 data width @ 50 MHz

Assume that an 80 MHz oscillator was already on the board for another application or device, and so can serve a dual purpose as the clock for the FPGA configuration.

91,548,896 / 80,000,000 = 1.144 seconds to configure XC7K325T x1 data width @ 80 MHz

1.144 / 4 = 286 ms to configure the XC7K325T FPGA in SPI flash x4 data width

Design Considerations for the External Master Clock

When using the external master clock (EMCCLK) as a configuration clock source, EMCCLK must be included in the user's design. Not doing so results in the FPGA failing to complete the start-up sequence. No special design requirements are needed when using the internal oscillator for FPGA configuration.

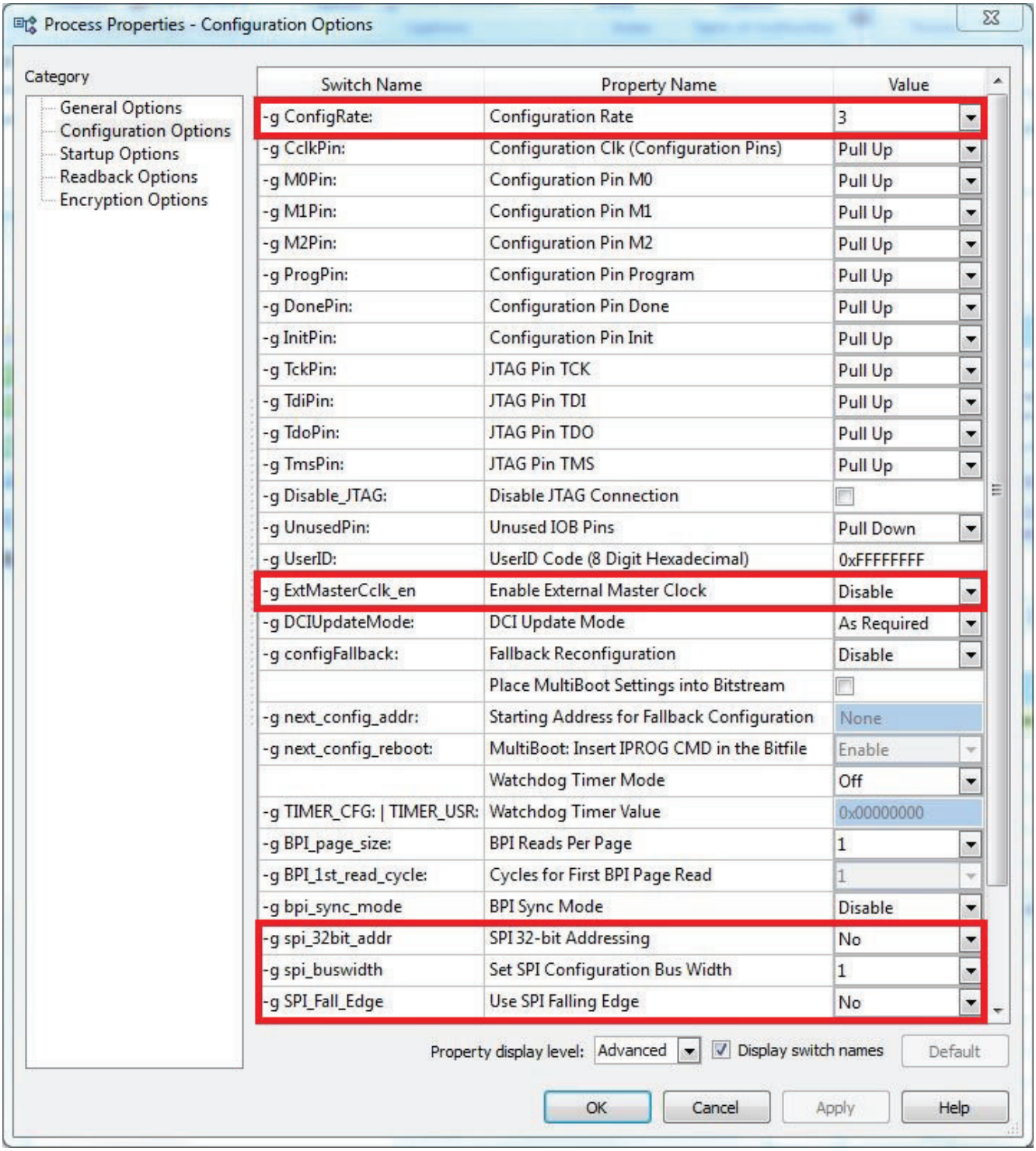
SPI Flash Configuration Options

Table 3 lists the BitGen options necessary to properly generate a configuration bitstream that is compatible with the SPI flash. These options are available through the Generate Programming File properties in the ISE tools shown in Figure 7. If the option is unspecified, the default value listed first and displayed bold text is used.

Table 3: **BitGen Configuration Options**

BitGen Option	Description
-g spi_buswidth: **1**|2|4	Selects the data width to be used when reading from the SPI flash.
-g spi_32bit_addr: **No**|Yes	Set to Yes when targeting a SPI flash 256 Mb or larger. This instructs the FPGA to transmit a larger address space required for larger flash devices.
-g SPI_Fall_Edge: **No**|Yes	Set to Yes when trying to achieve higher configuration speeds. Yes = capture data on the falling edge of the clock No = capture data on the rising edge of the clock
-g ConfigRate: **3**|6|9|12|16|22|26|33|40|50|66	Sets the approximate configuration clock frequency driven by the 7 series FPGAs to the SPI flash when using the internal oscillator (in MHz). The actual values for this option can be seen in UG628, *Command Line Tools User Guide* and might vary depending on the selected FPGA.
-g ExtMasterCclk_en: **Disable** |div-8|div-4|div-2|div-1	Instructs the FPGA to use the clock signal on EMCCLK as the configuration clock instead of the internal oscillator. Select div-1 to use the clock on EMCCLK at the same frequency and the other div options to divide down the EMCCLK clock by the appropriate value prior to output on CCLK.

To access these options from the ISE tool, right-click **Generate Programming File** then select **Process Properties > Configuration Options**. Alternatively, from the Process pull-down on the menu bar, select **Process Properties**, then select **Configuration Options**. From the Property display level pull-down at the bottom of the window, select **Advanced** to see all the options.



XAPP586_07_042912

Figure 7: **BitGen Configuration Options**

Preparing the SPI Flash Programming File

The Xilinx ISE tools takes the FPGA bitstream (.bit) and generates a PROM file (.mcs) that is then used to program the SPI flash. PROMGen (the program that performs this task) is located within the iMPACT programming tool under the Create PROM File flow. Selecting this flow walks the user through the options for generating a file. The user can also generate the PROM file by using the command line and the underlying program PROMGen.

Table 4: **PROMGen Options**

Option	Description
-spi	Used to maintain the correct bit ordering required to configure the FPGA from a SPI flash device.
-p mcs|exo	PROM output file format. Commonly accepted PROM file formats include Intel Hex (.mcs) and Motorola Hex (.exo). Only MCS is supported for Xilinx iMPACT indirect programming flows.
-s <size>	Specifies the SPI flash size in kilobytes. The SPI flash size must be a power of two for this option and the default setting is 64 KB.
-u <address>	Loads the bitstream starting at the specified address in an upward direction. If not entered, the default setting is at address 0. Most designs use address 0.
-o <filename>	Specifies the output file name.

The following command line example demonstrates how the options are used:

```
Promgen -spi -p mcs -o spi_flash.mcs -s 16384 -u 0 design.bit
```

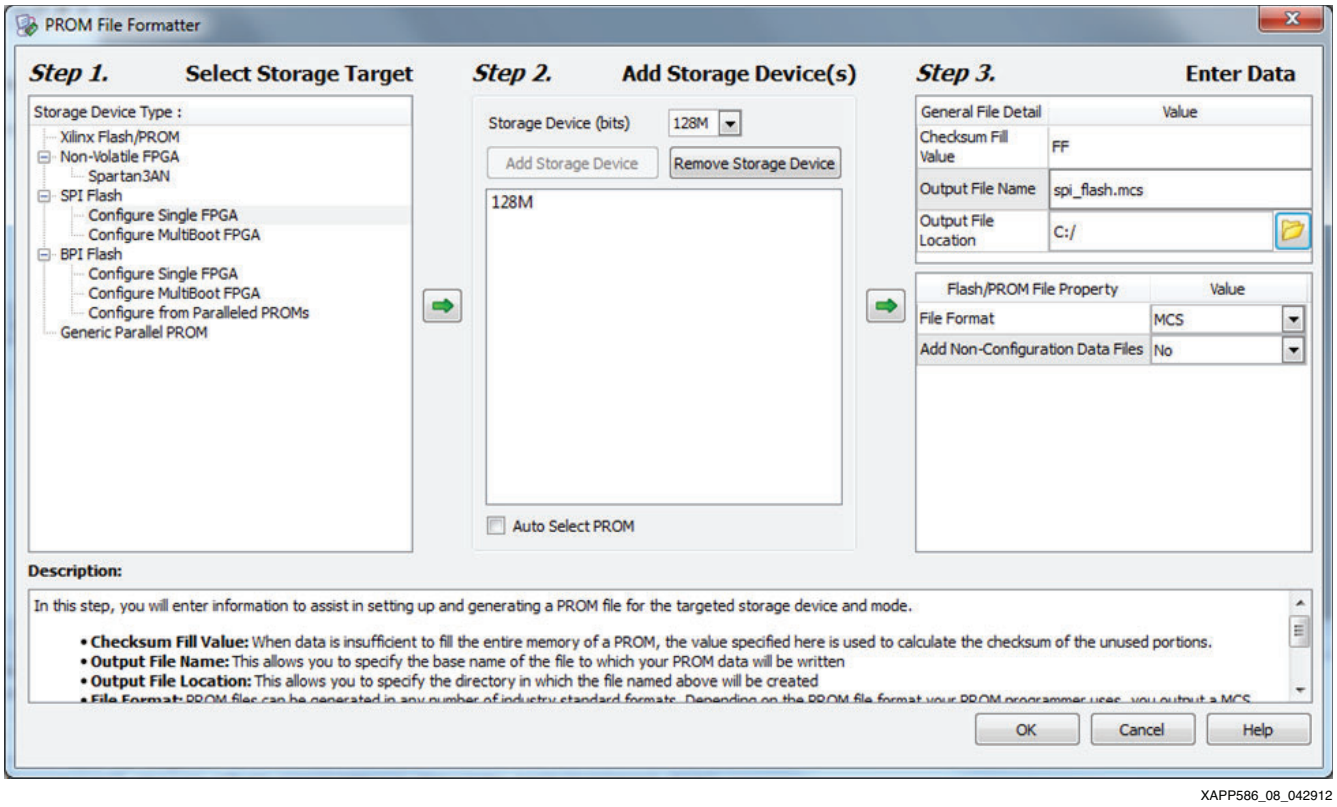
The example command line instructs PROMGen to:

- Create a file with the bit ordering for a SPI flash, using the MCS file format, with the output file name `spi_flash.mcs`
- Select a 128 Mb flash target (16384 x 1024 bytes x 8 bits = 134,217,728, which is the actual size of a 128 Mb SPI flash).

Launch the ISE iMPACT tool by starting the Configure Target Device process. Figure 8 shows the GUI interface for PROMGen.

The GUI steps to generate the flash file are:

1. Open iMPACT and select **Create PROM File (PROM File Formatter)** in the upper left box. Figure 8 shows the GUI interface for PROMGen.
2. Use the wizard that appears to generate the PROM file:
   a. Under Step 1 select **Storage Target** and under SPI Flash select **Configure Single FPGA**. Then select the green arrow that goes to Step 2.
   b. Under Step 2 Add Storage Device, from the Storage Device (bits) pull-down menu, select the proper SPI flash density (128 Mb is the example shown in Figure 8), and click **Add Storage Device**. Then select the green arrow that goes to Step 3.
   c. Under Step 3 Enter Data, type the output file name, browse to the location to place the file, ensure the file format is MCS, then click **OK**.



XAPP586_08_042912

Figure 8: **PROM File Formatter Flow**

Next, a dialog appears that provides instructions to begin adding configuration bitstream files.

3. Select **OK** and add the target BIT file.
4. For a single design image, click **No** at the next prompt, which is used for multiple designs.
5. Click **OK** to confirm completion of the design file entry.
6. Double-click **Generate File** to create the MCS flash programming file. The console log displays the files generated.

Programming the SPI Flash In-System

The iMPACT programming tool offers the ability to program the SPI flash in-system utilizing the existing connections between the SPI flash and the FPGA. This is referred to as *indirect programming* because the SPI flash is programmed through the FPGA, not directly from iMPACT. Also needed to perform indirect programming is access to the JTAG pins of the FPGA, a JTAG programming cable such as the Xilinx Platform Cable USB II, and a computer that has the iMPACT programming tool installed. Using this setup, the SPI flash can be reprogrammed without removing it from the board, which is extremely useful for debugging in a lab environment.

Note: The indirect programming solution is not intended for high-volume production. For production programming, consider solutions from BP Microsystems or Data I/O.

The first step to programming the SPI flash in-system requires that the 7 series FPGAs be first loaded with an interface design that bridges the programming cable to the SPI flash. This step clears the contents of the FPGA. This SPI interface design leaves the unused FPGA pins floating. The user should be aware of this and ensure that this does not have any undesired effects on other devices attached to the FPGA. For example, certain I/O pins might need to remain Low or High during SPI flash programming, and these pins would normally be pulled Low or High by the final FPGA design. In such cases, the designer might need to add external

pull-down or pull-up resistors on these pins to ensure correct behavior during SPI programming.

The following demonstration targets the Kintex-7 XC7K325T FPGA and the Micron N25Q128A13 SPI flash present on the KC705 Kintex-7 FPGA evaluation board.

Ensure the board is powered and the USB cable is attached. In the case of the KC705 board, a standard USB-to-micro-USB cable is used instead of a Platform Cable USB II. In most cases, there is no specialized hardware on the board to handle the PC-to-JTAG interface, therefore a Platform Cable USB II is required.

Figure 9 shows the GUI interface for the boundary-scan mode.

1. Start iMPACT by double clicking the **Configure Target Device** process and then select **Boundary Scan** in the upper left box (Figure 9).
2. Select the **Initialize Chain** icon to identify JTAG devices on the board.
3. The JTAG chain populates on the screen. The designer can choose to assign a bitstream to the FPGA because it does not matter for purposes of programming the SPI flash. In the example below the file `toplevel.bit` is assigned to the Kintex-7 FPGA.

   Note: A box with SPI/BPI? appears above the FPGA. This indicates that no PROM file is currently assigned to any attached flash device to this FPGA.

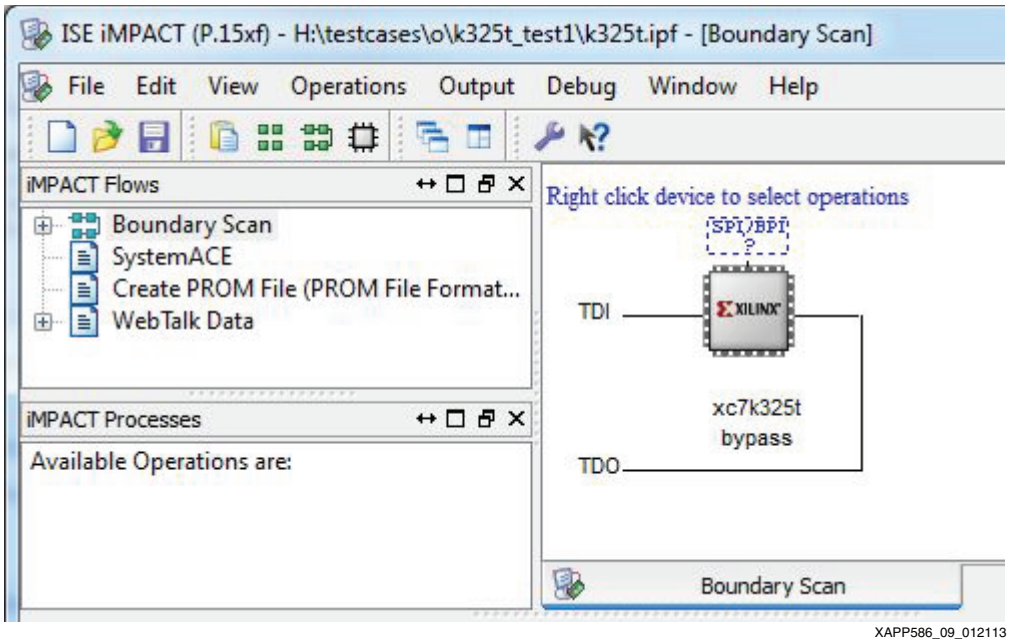


Figure 9: **iMPACT JTAG Chain**

4. Right-click **SPI/BPI** and select **Add SPI/BPI flash** (Figure 10). This step brings up an Explorer window. Navigate to the MCS file and select it.
5. iMPACT then asks which type of flash device is the target. Select the appropriate device in the pull-down menus.

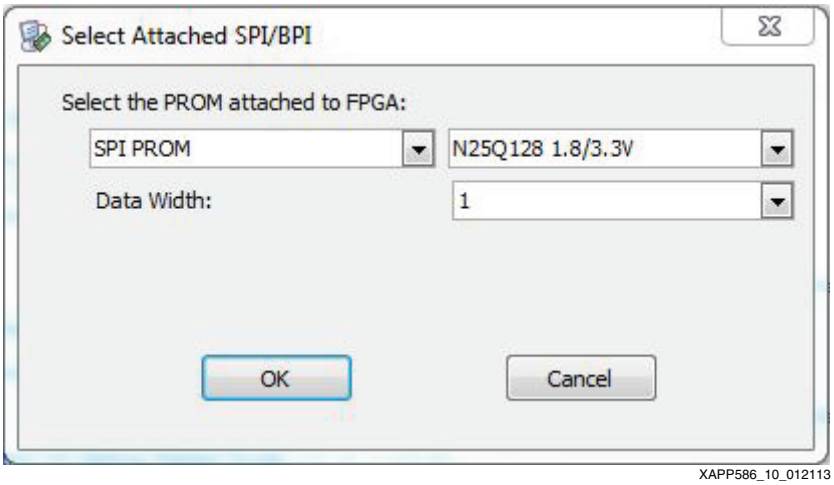


Figure 10: **iMPACT SPI Flash Selection**

The image on the right side of the window now changes from a dotted box labeled SPI/BPI? to a flash device image, indicating that a SPI flash device has been attached (Figure 11).

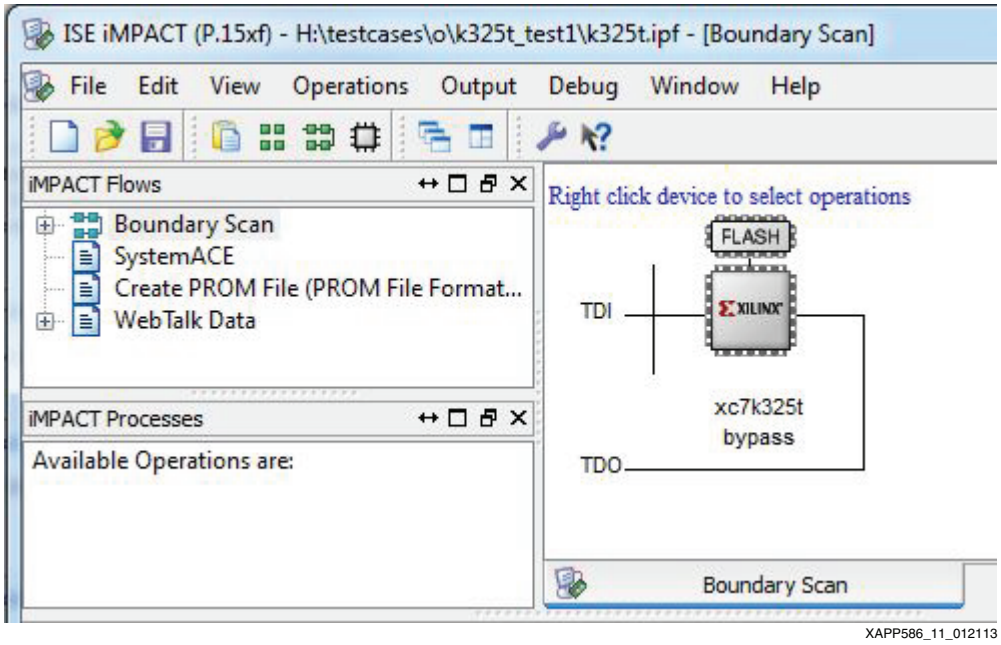


Figure 11: **iMPACT JTAG Chain with the SPI Flash Assigned**

6. Right-click **Flash** and select **Set Programming Properties** (Figure 12).

   From here, operations such as Erase and Verify are enabled by default. This step executes all operations **Erase > Program > Verify** when the designer programs the SPI flash. These instructions can be done individually, if desired.

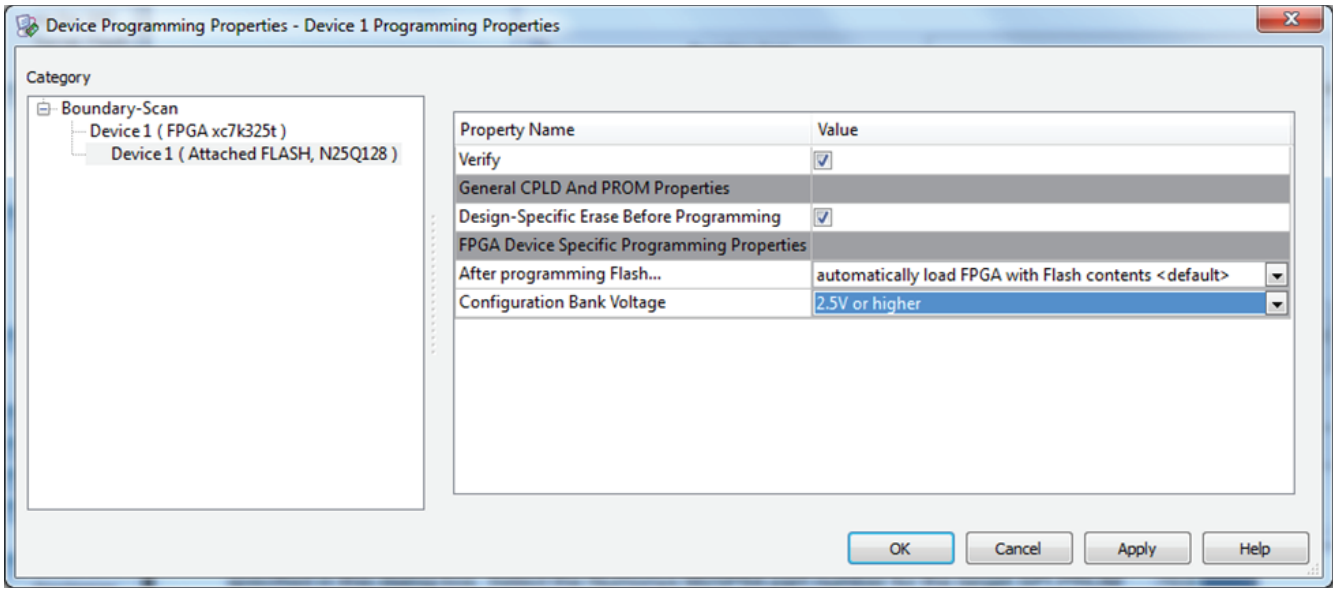


Figure 12: **iMPACT SPI Flash Programming Properties**

iMPACT Indirect Flash Operation Time Estimates

Using iMPACT 14.7 with the KC705 evaluation board, the operation times are provided. These times are not guaranteed values and are for reference only.

Kintex-7 XC7K325T FPGA's single bitstream (~91 Mb):

- Design-specific erase time: 190 seconds
- Program time: 440 seconds
- Verify time: 220 seconds

Conclusion

The SPI flash is a low-pin count and simple solution for configuring 7 series FPGAs. Support of indirect programming enhances ease of use by allowing in-system programming updates of the SPI flash by reusing connections already required for the configuration solution. Although some other configuration options permit faster configuration times or higher density, the SPI flash solution offers a good balance of speed and simplicity.

References

These documents provide supplemental material useful with this design:

1. UG470, *7 Series FPGAs Configuration User Guide*
2. DS180, *7 Series FPGAs Overview*
3. DS181, *Artix-7 FPGAs Data Sheet*
4. DS182, *Kintex-7 FPGAs Data Sheet*
5. DS183, *Virtex-7 FPGAs Data Sheet*
6. UG883, *Kintex-7 FPGA KC705 Evaluation Kit Getting Started Guide*
7. Xilinx ISE Design Suite Documentation: http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/design_tools/ise_design_suite.html
8. UG628, *Command Line Tools User Guide*
9. Micron Technology, Inc. for SPI flash data sheets: http://www.micron.com

10. Spansion, Inc. for SPI flash data sheets: http://www.spansion.com
11. Winbond Electronic Corp. site for SPI flash data sheets: http://www.winbond.com

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
05/30/12	1.0	Initial Xilinx release.
02/01/13	1.1	Updated pin functions in Table 1. Updated SPI flash and 7 series FPGAs blocks in Figure 5. Updated GUI steps to generate flash file in Preparing the SPI Flash Programming File. Updated Figure 9, Figure 10, and Figure 11. Updated Programming the SPI Flash In-System.
01/31/14	1.2	Added Vivado Design Suite support to second paragraph of Summary. Updated Kintex-7 FPGA bitstream number and calculations in Configuration Clock Calculation Example. Updated description of M[2:0] in Table 2. In iMPACT Indirect Flash Operation Time Estimates, updated iMPACT software version and operation times. Updated URLs for ISE Design Suite documentation and *Command Line Tools User Guide* in References.

Notice of Disclaimer