SPI FLASH/EEPROM PROGRAMMING USING FPGA AND JTAG

APPLICATION BRIEF

BY MICHAEL SONG
APPLICATION ENGINEER

By Michael Song – Senior Application Engineer

Michael has been an application engineer for ASSET InterTech for more than seven years. He is responsible for pre- and post-sales technical support and works directly with users of the ScanWorks® platform for embedded instruments. He also contributes to and conducts user training classes and assists users with test-related project development. Prior to joining ASSET, Michael was a test development engineer at the contract manufacturing firm, Jabil, and a test developer with Wistron. Fluent in two languages, Chinese and English, Michael received his Bachelor of Science Degree in Computer Science from Nanjing University of Aeronautics and Astronautics.

Table of Contents

Table of Figures/Tables

Overview

SPI flash memory can be programmed effectively through either of two methods: 1. accessing the SPI flash through the boundary-scan chain of a connected FPGA, or 2. accessing the SPI flash from IP (intellectual property, such as an embedded instrument) inserted in the FPGA. Both methods are executed from the boundary-scan test (BST) tool on the ScanWorks platform for embedded instruments.

This application brief describes and compares these two methods, using as an example the programming of 1MB of data into a Micron SPI Flash device (M25P80) from a Xilinx Spartan 6 FPGA (XC6SLX16-2CSG324C).

Details

Board Type:

- Xilinx Spartan 6 Demo Board

Flash Device:

- Type: M25P80
- Size: 1Mbyte (8Mbit)

FPGA Device:

- Type: XC6SLX16-2CSG324C
- Chain Length: 744

Software:

- ScanWorks Boundary-Scan Test Development, version 3.15.1.
- ScanWorks FPGA-Controlled Test Development, version 1.2

Hardware

- RIC-1000 Single-Port Remote Instrumentation Ethernet Controller

Methods

Boundary-Scan Chain Access Method

The first method of programming through the FPGA's boundary-scan chain is slower than programming from an embedded instrument IP in the connected FPGA. With the standard boundary-scan access method, the data to be programmed is scanned into the FPGA but simply passed through the device's internal boundary scan chain. It is then output to the targeted SPI memory device. (Figure 1)

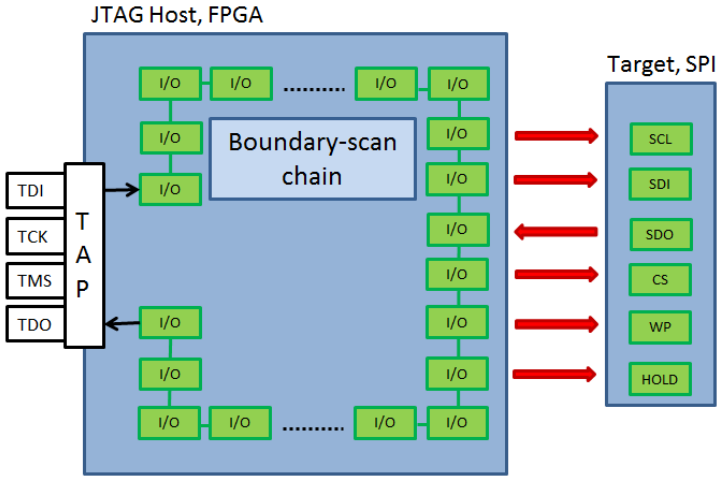

Figure 1: Boundary-scan chain access method

On most high-end FPGAs, the boundary-scan chain can typically be 1,000 cells or longer. All of the cells in the FPGA's boundary-scan chain must be scanned through before the data reaches the FPGA's SPI pins where it is output to the SPI memory. In fact, every change to the FPGA's SPI data pins or clock pin requires a new scan through the boundary-scan register on the device, resulting in slower programming speeds. The factors that limit the speed of this method are mostly the length of the boundary-scan scan chain in the FPGA and, to some degree, the speed at which the boundary-scan test clock (TCK) can be run on the circuit board.

SPI Master IP with FIFO Memory

With this method, a SPI Master embedded instrument or IP is programmed into the FPGA. The SPI Master includes a memory area implemented as FIFO memory (Figure 2). The SPI Master will read the data to be programmed into the connected SPI memory device(s) from the FPGA's memory area. The size of the FIFO will typically correspond to the page sizes of the SPI memory devices. After the boundary-scan tool has streamed a full page of data through the FPGA's boundary-scan Test Access Port (TAP) into the on-chip FIFO memory space, the SPI Master IP grabs the data and starts programming the connected SPI memory at the access speed of the memory.

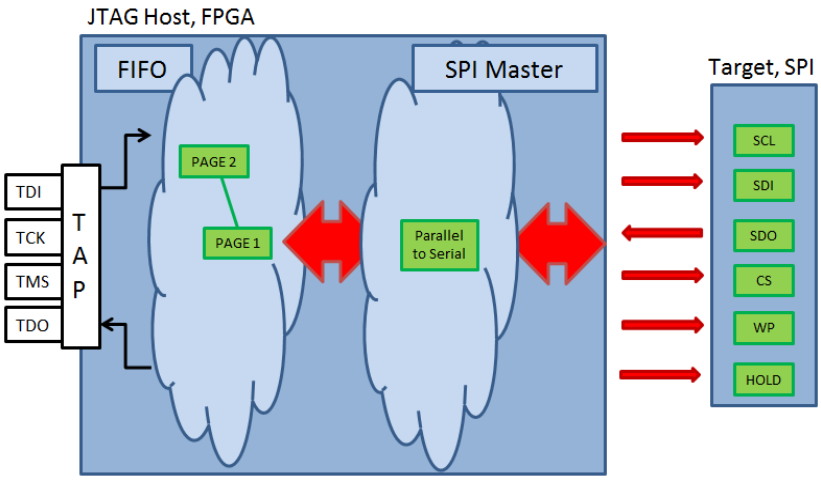

Figure 2: SPI master IP with FIFO memory

Very high programming speeds can be achieved with this method, making it suitable for any amount of programming data, any size FPGA and any size memory device. This method can usually approach the maximum programming speed of the SPI memory device.

Programming Time Comparison

The following comparison shows that the second method of inserting an embedded instrument IP into the FPGA is 190 times faster when it comes to programming SPI memory than the method based on access through the FPGA's boundary-scan chain. This analysis found that a blank chip could be programmed in just seven seconds with the second method. This faster method would save a tremendous amount of time and resources on the production floor. Table 1 includes comparison data on other functions besides programming, such as erase, blank check and verify.

Table 1: Programming performance. Times are in seconds.

	TCK	Data Size	Chain Length	FPGA Config.	Erase Program Chip	Blank Check	Program	Verify	Total
Boundary-Scan Test	10MHz	1MB	744	0	15	1391	1332	1383	4121
FPGA-Controlled Test	10 MHz	1MB	Embedded Instrument in the FPGA	4	15	5	7	5	36
Difference						**278x**	**190x**	**276x**	**114x**

Manufacturing Cost Savings

Assuming an average cost-per-minute in a manufacturing environment of $1, the cost savings from faster programming times can be considerable. For example, if 10,000 units are manufactured yearly and programming times are reduced from 22 minutes to 30 seconds, the savings amounts to $215,000 per year in this instance. Or, to state it another way, each unit can be programmed for just $0.50 instead of $22!

Summary

Because the FPGA IP method is much faster than the boundary-scan chain method, it will support the programming of SPI memories in the standard manufacturing flow.

ASSET has published a number of eBooks on the subject of in-system, in-line and parallel programming of memory. Check out these and other resources on our eResources page at: http://www.asset-intertech.com/eResources.