Connecting Spansion® SPI Serial Flash to Configure Xilinx® FPGAs

Application Note

by Frank Cirimele and Jocelyn Carroue

1. Introduction

Xilinx FPGAs are programmable logic devices used for basic logic functions, chip-to-chip connectivity, signal processing, and embedded processing. These devices are programmed and configured using an array of SRAM cells that need to be re-programmed on every power-up. Several different methods of configuring FPGAs are normally used. They include programming by a microprocessor, JTAG port, or directly by a serial PROM or Flash. Spansion's SPI (Serial Peripheral Interface) Flash can be easily connected to Xilinx FPGAs in order to configure the FPGA at power-up. The SPI configuration mode is supported for Xilinx Virtex™-5 and Spartan™-3E, Spartan-3A, Spartan-3AN and Spartan-3A DSP FPGA families.

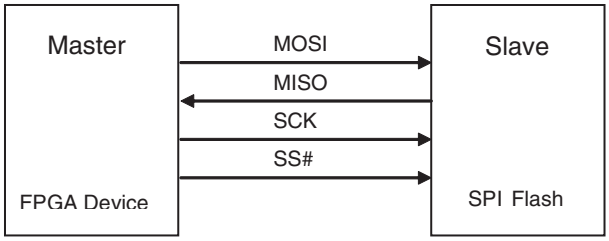
2. SPI Basics

Serial Peripheral Interface (SPI) is a simple 4-wire synchronous interface protocol which enables a master device and one or more slave devices to intercommunicate. The SPI bus consists of 4 signal wires:

- Master Out Slave In (MOSI) signal generated by the master (data to slave)
- Master In Slave Out (MISO) signal generated by the slave (data to master)
- Serial Clock (SCK) signal generated by the master to synchronize data transfers
- Slave Select (SS#) signal generated by master to select individual slave devices, also known as Chip Select (CS#) or Chip Enable (CE#)

Following SPI protocol, the master is assigned to the FPGA device and the slave to the SPI Flash device, as shown in Figure 2.1. Per this connection, the SPI Flash is available to configure the FPGA at power-up.

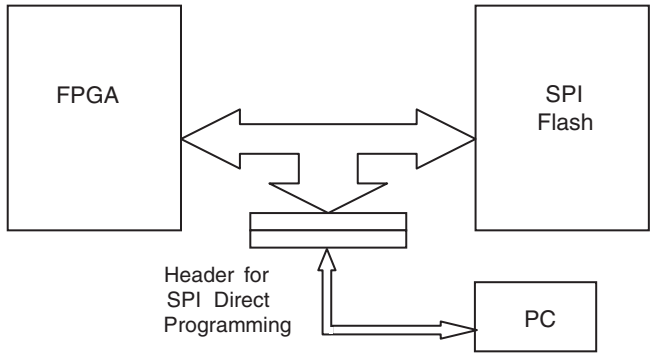
Figure 2.1 Direct Configuring FPGA Interface with SPI Flash



3. SPI Flash Connections to FPGAs

Figure 3.1 displays a simplified block diagram of the connection between SPI Flash and FPGA. It shows the configuration interface between FPGA and SPI Flash, as well as the header for direct programming the SPI Flash for configuration data updates from a PC or embedded host.

Figure 3.1 Block Diagram of Configuration Interface with Re-programming Capability



The following is a detailed explanation of connections between Xilinx FPGA and Spansion SPI Flash. The SPI Flash connections to the Virtex-5 and Spartan-3 family FPGA devices are shown in Figure 3.2 and Figure 3.3 respectively. Pin descriptions are listed in Table 3.1 on page 3.

The FPGA supplies the CCLK output from its internal oscillator to drive the clock input of SPI Flash. ISP Control is used to tri-state the FPGA SPI interface signals during In-System Programming to re-program the configuration data in the SPI Flash.

To read configuration data from the SPI Flash at power-up, the Xilinx FPGA must issue a read command to the SPI Flash, which it does automatically based on MODE being set to SPI and VS/FS being set to the correct type of read. The serial Flash has two read modes: normal read and fast read. Table 3.2 on page 4 lists the FPGA input logic levels to set up the SPI Master Mode and read configuration.

There are four power supply inputs to the Virtex-5 and Spartan FPGAs associated with configuration. The required voltages to be applied to these inputs and a description of the power supply usages in configuration are shown in Table 3.3 on page 4. There are a variety of FPGA sizes under each family covered in this application note. Table 3.3 on page 4 and Table 3.5 on page 5 list the smallest SPI Flash devices required for typical configuration bit requirements of different members of the Spartan and Virtex-5 families.

Figure 3.2 Virtex-5 Family FPGA Configuration from Spansion SPI Serial Flash Connection Diagram

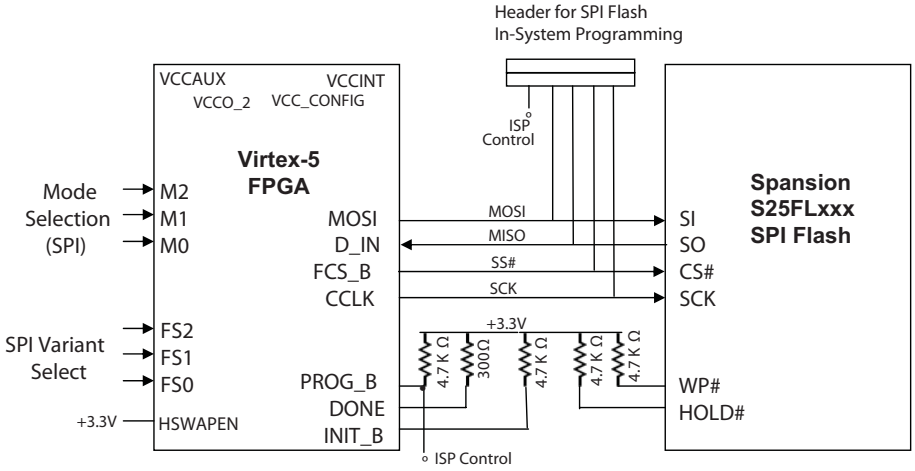


Figure 3.3 Spartan Family FPGA Configuration from Spansion SPI Serial Flash Connection Diagram

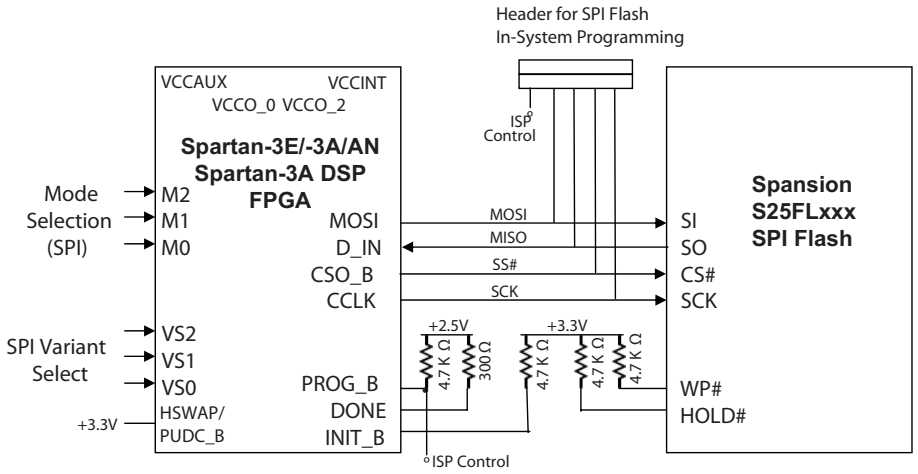


Table 3.1 Pin Descriptions for FPGA Configuration from SPI Flash

Pin Name	Type	FPGA Family	Description
M[2:0]	Input	Virtex5/ Spartan	**Mode Pins.** Selects FPGA configuration mode. M[2:0] = '001' defines Master SPI Mode.
FS[2:0]	Input	Virtex5	**Variant Select.** Instructs the Virtex-5 FPGA how to communicate with the attached SPI Flash device. FS[2:0] = '101' for Read FS[2:0] = '111' for Fast Read
VS[2:0]	Input	Spartan	**Variant Select.** Instructs the Spartan FPGA how to communicate with the attached SPI Flash device. VS[2:0] = '101' for Read VS[2:0] = '111' for Fast Read
HSWAPEN/ HSWAP/ PUDC_B	Input	Virtex5/ Spartan	**User I/O Pull-up Control During Configuration.** When Low during configuration, enables pull-up resistors in all I/O pins to respective I/O bank V_{CCO} input: 0 = Pull-ups during configuration 1 = Tri-stated during configuration
MOSI	Output	Virtex5/ Spartan	**Master Out Slave In.** Used by the FPGA to specify the instruction to execute or to send data to the SPI Flash device.
DIN	Input	Virtex5/ Spartan	**Data Input.** Used by the FPGA to collect data transferred from the SPI Flash device.
CCLK	Output	Virtex5/ Spartan	**Configuration Clock.** Provides the synchronous timing for the SPI interface.
CSO_B	Output	Spartan	**Chip Select Output.** Connects to the SPI Flash Chip Select input, and is Active Low.
FCS_B	Output	Virtex5	**Chip Select Output.** Connects to the SPI Flash Chip Select input, and is Active Low.
INIT_B	Open-drain bi-direct. I/O	Virtex5/ Spartan	**Initialization Indicator.** Goes active low during configuration. Requires an external 4.7 KΩ pull-up resistor for this signal.
DONE	Open-drain bi-direct. I/O	Virtex5/ Spartan	**FPGA Configuration Done.** Is Low during configuration and goes High when configuration is successfully completed. Requires external 300-330 KΩ pull-up resistor for this signal.
PROG_B	Input	Virtex5/ Spartan	**Program FPGA.** When asserted Low, it forces the FPGA to restart its configuration process by clearing configuration memory and resetting the DONE and INIT_B pins. Return High to allow configuration to start. Requires an external 4.7 KΩ pull-up resistor for this signal. If driving externally, use an open-drain or open-collector driver.

Table 3.2 FPGA Mode and Read Settings

FPGA	Function	Signal	Applied Logic Level
Virtex-5/ Spartan	Master SPI Mode	M2	0
		M1	0
		M0	1
Virtex-5	Normal Read/ Fast Read Selection	FS2	1 / 1
		FS1	0 / 1
		FS0	1 / 1
Spartan	Normal Read/ Fast Read Selection	VS2	1 / 1
		VS1	0 / 1
		VS0	1 / 1

Table 3.3 Configuration Power Supplies

FPGA	Power Supply	Applied Voltage	Description
Virtex-5	VCCAUX	+2.5V	Voltage for configuration logic
	VCCO_2	+3.3V	Voltage for: FS[2:0], FCS_B, MOSI
	VCC_CONFIG	+3.3V	Voltage for: M[2:0], HSWAPEN, PROG_B, DONE, INIT_B, D_IN
	VCCINT	+1.0V	Internal core voltage
Spartan	VCCAUX	+2.5V or +3.5V(1)	Voltage for: PROG_B, DONE
	VCCO_0	+3.3V	Voltage for: HSWAP
	VCCO_2	+3.3V	Voltage for: M[2:0], VS[2:0], INIT_B, CCLK, CSO_B, D_IN, MOSI
	VCCINT	+1.2V	Internal core voltage

Note:
1. Spartan-3A/AN/ADSP allows either 2.5V or 3.5V for VCCAUX.

Table 3.4 SPI Flash Selection for Spartan Family FPGA Devices

Family	FPGA	# of Configuration Bits	Minimum SPI Flash Device (Spansion part number)
Spartan-3A/3AN	XC3S50A/AN	437,312	512 Kb (S25FL040A)
	XC3S200A/AN	1,196,128	2 Mb (S25FL040A)
	XC3S400A/AN	1,886,560	2 Mb (S25FL040A)
	XC3S700A/AN	2,732,640	4 Mb (S25FL040A)
	XC3S1400A/AN	4,755,296	8 Mb (S25FL008A)
Spartan-3A DSP	XC3SD1800A	8,197,280	8 Mb (S25FL008A)
	XC3SD3400A	11,718,304	16 Mb (S25FL016A)
Spartan-3E	XC3S100E	581,344	1 Mb (S25FL040A)
	XC3S250E	1,353,728	2 Mb (S25FL040A)
	XC3S500E	2,270,208	4 Mb (S25FL040A)
	XC3S1200E	3,841,184	4 Mb (S25FL040A)
	XC3S1600E	5,969,696	8 Mb (S25FL008A)

Table 3.5 SPI Flash Selection for Virtex-5 Family FPGA Devices

FPGA	# of Configuration Bits	Minimum SPI Flash Device (Spansion Part Number)
XC5VLX30	8,374,016	8 Mb (S25FL008A)
XC5VLX50	12,556,672	16 Mb (S25FL016A)
XC5VLX85	21,845,632	32 Mb (S25FL032A)
XC5VLX110	29,124,608	32 Mb (S25FL032A)
XC5VLX220	53,139,456	64 Mb (S25FL064A)
XC5VLX330	79,704,832	128 Mb (S25FL128P)
XC5VLX30T	9,371,136	16 Mb (S25FL016A)
XC5VLX50T	14,052,352	16 Mb (S25FL016A)
XC5VLX85T	23,341,312	32 Mb (S25FL032A)
XC5VLX110T	31,118,848	32 Mb (S25FL032A)
XC5VLX330T	82,696,192	128 Mb (S25FL128P)

4. Other Considerations

There are three other considerations related to connecting SPI Flash to a Xilinx FPGA that are pertinent to this application note. First, since the power-up timing and voltage thresholds are different for both FPGAs and SPI serial Flash devices, it is important to carefully review the differences to insure compatibility between devices on power-up. Second, review the control signal that is required to enable direct SPI Flash programming in-system is described. Third, discuss the components contributing to the fastest configuration time from Spansion SPI Flash to Xilinx FPGA.

4.1 Applying Voltages at Power-On

A race condition can exist between the SPI Flash and FPGA at power-on. After completing its power-on reset sequence, the FPGA sends a read command to the SPI Flash to acquire the configuration data bitstream. If the SPI Flash has not yet completed its own power-on reset sequence, then it is not ready to respond to the FPGA read command, which is issued only once. Under this scenario, the FPGA does not configure.

The FPGA waits for its three power supplies -VCCINT, VCCAUX and VCCO_2 - to reach their individual power-on thresholds before starting the configuration process. In many applications, VCCO_2, which supplies +3.3V to both FPGA and SPI Flash, is valid before the FPGA's other two power supply inputs (VCCINT and VCCAUX), and consequently, there may be no issue. However, since the SPI Flash minimum voltage threshold is much higher than the VCCO_2 threshold and SPI Flash has an additional delay after it reaches its minimum voltage before SPI Flash is available for read operations, a careful analysis must be completed to insure timing compatibility between FPGA and SPI Flash.

Let's focus on each device. After the three FPGA voltages reach their Power On Reset thresholds, the FPGA starts its configuration process:

- Clears its internal configuration memory (which takes approximately 1 millisecond)
- De-asserts INIT_B, and selects SPI Flash
- Sends the appropriate read command to start configuration bitstream from SPI Flash

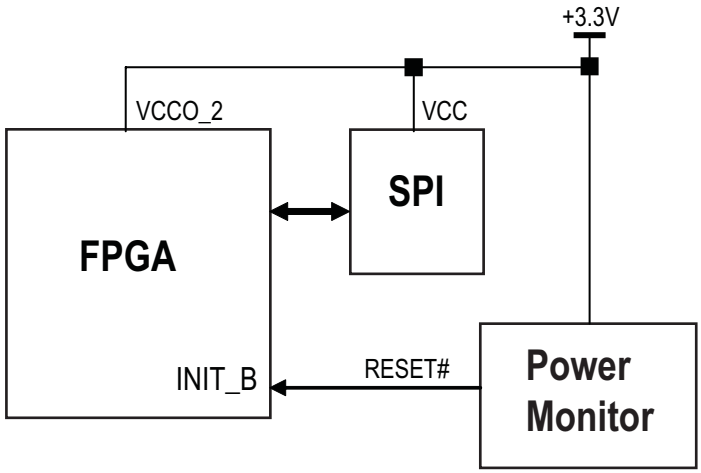
After Spansion SPI Flash reaches its minimum voltage, a power-up time delay (t_{PU}) must be added to the SPI Flash power-on before it is available for read operations. For current Spansion SPI Flash offerings, the minimum values for time delay t_{PU} are 10 and 15 milliseconds.

The power-on time difference is considerable between the FPGA and SPI Flash devices, which requires a circuit solution to guarantee power-on compatibility between FPGA and SPI Flash. The following solutions by Xilinx are candidates for the designer:

1. Use external control to hold the INIT_B or PROG_B pin Low until SPI Flash has powered up reliably and is ready to accept commands. For this solution, use an open-drain or open-collector output when driving INIT_B or PROG_B pins. One example of external control is using a Power Monitor Supervisor device as seen in Figure 4.1, and there are many available devices on the market from companies such as TI and Analog Devices.

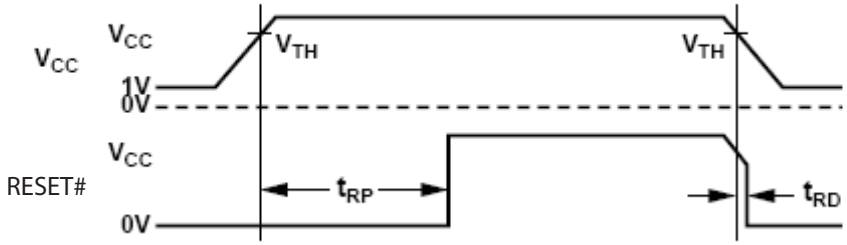
   This is Spansion's recommended method, due to its tolerance to temperature and voltage variations.

Figure 4.1 Power-on Reset Using Power Monitor Supervisor



One example of the Power Monitor Supervisor is the Analog Devices ADM6384x27D2. The RESET# signal from this device is held Low until its power supply voltage reaches 2.7 V + 20 ms (t_{RP} time), which is shown in Figure 4.2.

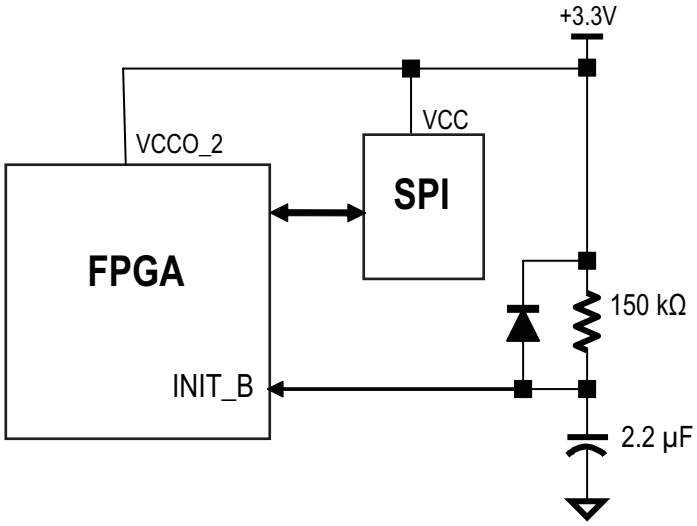
Figure 4.2 RESET# Signal at Power-on for ADM6384x27D2



2. Attach R-C delay circuit to the INIT-B pin as seen in Figure 4.3, which forces the FPGA to wait for a preselected period of time after its memory clearing process before allowing the FPGA to continue its configuration process. Minimum INIT_B threshold voltage is used in determining the R-C component values. In this case, the voltage input to INIT_B is 0.8 V. Using the component values recommended by Xilinx. In Figure 4.3, the delay to configuration start is 90 milliseconds.

   Note that this method is highly susceptible to temperature and voltage conditions. This method is not recommended by Spansion.

Figure 4.3 Power-on Reset Using R-C Delay Circuit



4.2 Direct Programming SPI Flash

During production, the Spansion SPI Flash is loaded with configuration data via a third-party programmer. After installation in the final product, the off-board programmed SPI Flash configures the Xilinx FPGA at every power-up. However, if a change in configuration data is planned, the SPI Flash requires reprogramming on-board. Figure 3.2 on page 2 and Figure 3.3 on page 3 shows the ribbon cable header for direct SPI Flash programming In-System Programming (ISP). ISP programmer selection dictates header configuration. In order to allow an external source to directly program the SPI Flash through this header, the PROG_B input signal, shown as ISP Control, must be held Low to force the FPGA I/O pins into a high-Z state during programming, thereby eliminating any interference from the FPGA I/O pins to the SPI Flash reprogramming. Direct programming is available through a Xilinx software utility (XSPI), which allows ID check to be skipped for non-supported SPI Flash devices. For more details, go to http://www.xilinx.com/support/answers/29578.htm

4.3 Fastest Configuration Time

The question arises on what is the fastest time that the Spansion SPI Flash device can configure a Xilinx FPGA. The answer to this question involves several components associated with the timing of this interface, specifically:

- SPI Flash Timing — Setup and hold times for the FPGA DIN serial input
- FPGA Timing — Delay and hold times of Spansion SPI Flash SO data output
- Configuration Clock — Maximum speed of CCLK from the FPGA

Figure 4.4 Configuration Timing Interface between Xilinx FPGA and Spansion SPI Flash

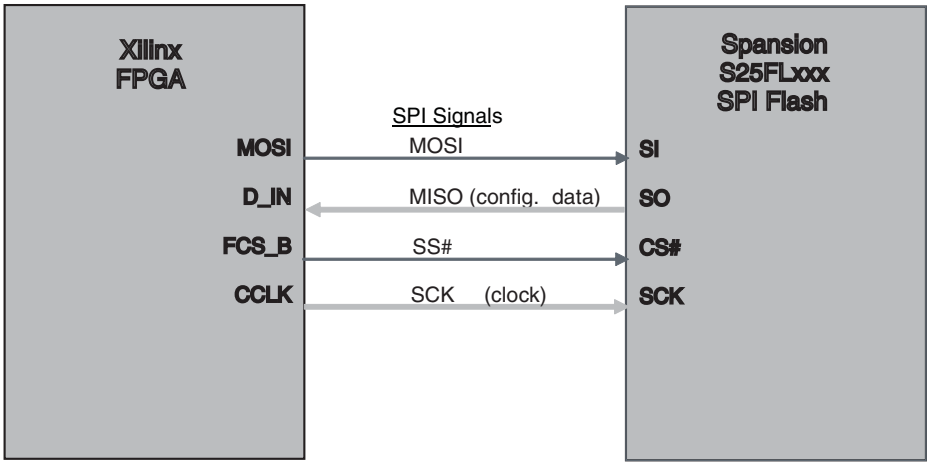
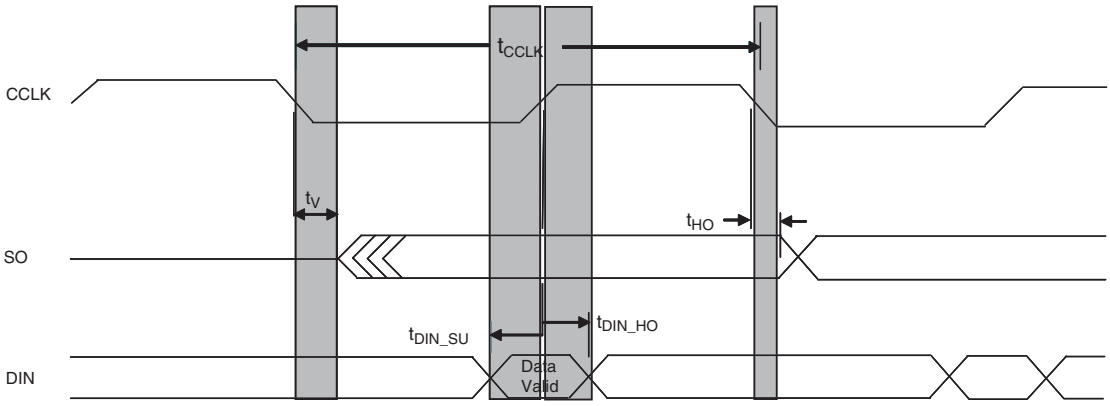


Figure 4.4 shows the two signal paths of the SPI interface involved in passing configuration data to the FPGA at power-on. The FPGA provides the configuration clock (CCLK) speed to transmit the data, while the configuration data passes from SPI Flash SO output to the D_IN input of the FPGA.

The DIN input setup and hold times determine the minimum input data pulse width that the FPGA requires from the SPI Flash for each configuration data bit it receives. Its complement on the SPI Flash side, the delay and hold times of SPI Flash determine the output pulse width per data bit that the SPI Flash SO output offers to the FPGA. So what this means in terms of the SO-to-DIN SPI interface: The SPI Flash SO output pulse width must be greater than or equal to the minimum DIN pulse width requirement to satisfy the DIN data valid requirement, as seen in Figure 4.5. As the CCLK frequency increases, shown in the figure as the t_{CCLK} period decreasing, the t_V and t_{HO} times approach the setup and hold times of the FPGA. When these times touch the FPGA data valid bandwidth boundaries, then t_{CCLK} will have achieved maximum frequency bandwidth.

Figure 4.5 SPI Flash Serial Output (SO) to Xilinx Data Input (DIN) Timing



Reviewing this interface in greater detail:

FPGA

Assume t_{DIN_SU} = Minimum DIN setup time requirement (master mode)

t_{DIN_HO} = Minimum DIN hold time requirement (master mode)

Therefore,

Minimum DIN input data pulse width for FPGA = $t_{DIN\_SU} + t_{DIN\_HO}$

SPI Flash

t_V = Maximum SO time delay from the CCLK falling edge

t_{HO} = Minimum SO hold time after the next CCLK falling edge

t_{SCK} = CCLK frequency setting via the system clock or programmed into the FPGA via its ConfigRate options

Thus,

Minimum SO output data pulse width = t_{CCLK} - (t_V - t_{HO})

4.3.1 Potential Bandwidth

Potential bandwidth corresponds to the fastest frequency the SPI interface can be run, limited only by the setup, hold and delay times associated with the SO output and DIN input. Mathematically, the equation for maximum frequency (f_{MAX}) is the inverse of the setup, hold and delay times associated with the interface, namely:

$$f_{MAX} = 1 / [(t_V + t_{HO})_{SPI\ Flash} + (t_{DIN\_SU} + t_{DIN\_HO})_{FPGA}]$$

4.3.1.1 Virtex-5 Example

Maximum potential frequencies for Virtex-5 FPGAs is:

t_{DIN_SU} = 4 ns, t_{DIN_HO} = 0 ns

For S25FL-P SPI Flash, the SO delay and hold times are:

t_V = 8 ns, t_{HO} = 0 ns

Therefore, f_{MAX} = 1/ (8ns + 0 + 4ns + 0) = 83.33 MHz

4.3.1.2 Spartan-3E Example

Another example would be to look at the fastest configuration time for a Spartan-3E FPGA example.

The setup time (7 ns min.) and hold time (0 ns) for a Spartan-3E device when connected to a Spansion S25FL-A SPI Flash (t_V = 9 ns and t_{HO} = 0) produces a maximum operating frequency of:

f_{MAX} = 1/ (9ns + 0 + 7ns + 0) = 62.5 MHz

4.3.2 Available Bandwidth

The Available Frequency Bandwidth equals the Potential Bandwidth minus the Configuration Clock Frequency (tCCLK). Mathematically,

Available Frequency Bandwidth = [1 / (t_V + t_{HO} + t_{DIN_SU} + t_{DIN_HO})] - [1/t_{CCLK}]

The meaning of this equation is that there is available bandwidth as long as the current FPGA configuration clock frequency (1/t_{CCLK}) is less that the Potential Bandwidth, which means that a higher SPI clock frequency can be connected to the SPI Flash up to the point that there is no more available bandwidth.

4.3.3 FPGA Configuration Speeds

Now let's look at the configuration clock speed from the Xilinx FPGAs to determine how fast they can be programmed, and then with that information the fastest configuration time can be calculated. The Configuration Clock (CCLK) has two sources, the external System Clock or the internal CCLK derived by setting the BitGen ConfigRate option. The ConfigRate derived clock is not normally used due to the large variance (up to 50%) that can occur due to internal voltage and thermal variations. So let's assume that the CCLK is derived from the System Clock operating at the maximum Potential Bandwidth in a Virtex-5 application.

Suppose the Virtex-5 application has an embedded XC5VLX330 FPGA with 79,704,832 configuration bits. Each tCCLK clock pulse from the FPGA inputs one configuration bit into the FPGA. So the fastest configuration time for our example assumes that the Spansion S25FL128P SPI Flash device is run at the maximum clock rate of 83.33 MHz previously described. The fastest configuration time for this example is:

Fastest Configuration Time = (Maximum t_{CCLK}) x (# of Configuration Bits)

(Virtex-5) = (1/83.33 MHz) x (79,704,832) = 0.96 seconds

Picking an XC3X1600E Spartan-3E device for our example application, the fastest configuration time that can be expected is:

Fastest Configuration Time = (1/62.5 MHz) x (5,969,696 configuration bits)

(Spartan-3E) = 0.0955 seconds

5. References

1. *Xilinx Spartan-3 Generation Configuration User Guide UG332*.
2. *Xilinx Virtex-5 FPGA Configuration User Guide UG191*.
3. *Xilinx Application Note XAPP951: Configuring Xilinx FPGAs with SPI Serial Flash*.

6. Revision History

Section	Description
Revision 01 (October 25, 2007)	
	Initial release
Revision 02 (March 3, 2008)	
	Updated Figures 3.1 and 3.2. Added section 4.3.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spansion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spansion product under development by Spansion. Spansion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spansion assumes no liability for any damages of any kind arising out of the use of the information in this document.