XAPP1313 (v1.0) June 14, 2017

Spartan-7 FPGA Configuration with SPI Flash and Bank 14 at 1.35V

Author: Randal Kuramoto

Summary

This application note describes a method for configuring a Spartan®-7 FPGA from a 1.8V serial peripheral interface (SPI) NOR flash memory connected to the FPGA dedicated I/O bank 0 at 1.8V and multi-use FPGA I/O bank 14 at 1.35V. This method allows the Spartan-7 FPGA bank 14 to interface to a low-power 1.35V DDR3L memory and low-pin-count SPI configuration storage flash. This mixed, low-voltage configuration is not standard. Thus, data sheet configuration specifications do not apply, and Vivado® design tools do not directly support this configuration. This application note provides the implementation requirements, tool work-arounds that include a SPI flash programming method, and considerations for this non-standard configuration.

For Spartan-7 FPGA configuration using standard voltages see the *7 Series FPGAs Configuration User Guide* (UG470) [Ref 1] and the *Using SPI Flash with 7 Series FPGAs Application Note* (XAPP586) [Ref 2].

Download the reference design files for this application note from the Xilinx website. For detailed information about the design files, see Reference Design.

Application

A Spartan-7 FPGA, with a 1.35V DDR3L memory interface on the multi-use FPGA bank 14, can configure from a 1.8V SPI flash memory with the assistance of a level translator as shown in Figure 1.

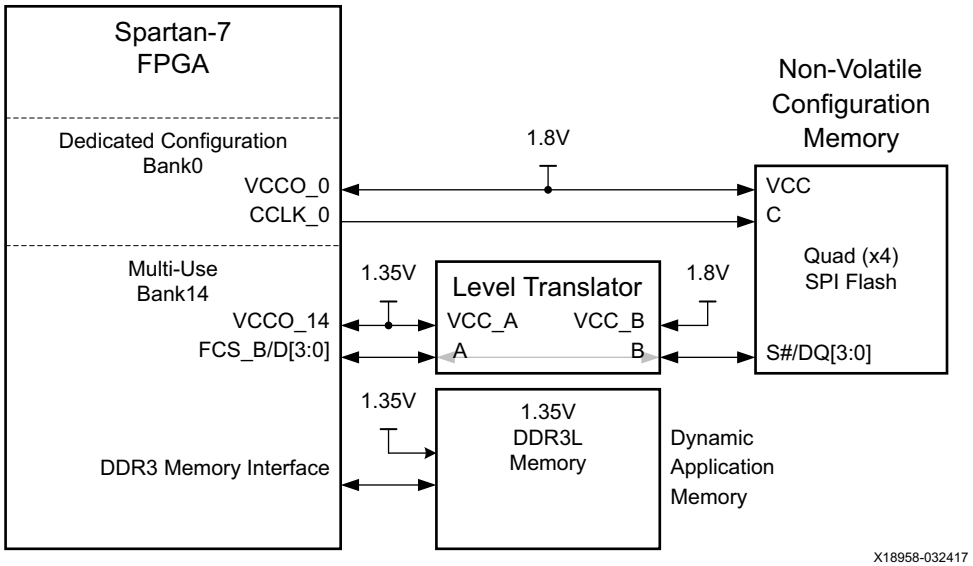

Figure 1: **Spartan-7 FPGA with 1.8V SPI Configuration Flash and 1.35V DDR3L**

Before resorting to this level-translated SPI flash configuration solution, use the Memory Interface Generator from the Vivado design tools IP Catalog. See the *Zynq-7000 All Programmable SoC and 7 Series Devices Memory Interface Solutions User Guide* (UG586) [Ref 3] to check for a DDR3 memory interface with a data width of 8-bits or 16-bits (without a controller chip select pin) that fits into an alternate FPGA I/O bank other than bank 14.

Implementation of the level-translated SPI flash configuration solution shown in Figure 1 requires the following procedures:

- Selecting a SPI Flash
- Designing the Board
- Determining the Configuration Clock Rate
- Specifying 1.35V-compatible I/O Standards for Bank 14 Pins
- Building the FPGA Bitstream
- Generating the Configuration Memory File
- Programming the SPI Flash Memory

See the *Zynq-7000 All Programmable SoC and 7 Series Devices Memory Interface Solutions User Guide* (UG586) [Ref 3] for implementation of the 1.35V DDR3L memory interface solution.

Selecting a SPI Flash

Select a SPI flash for the Spartan-7 FPGA configuration as follows:

1. Determine the minimum memory image size in terms of megabits (Mb) using Table 1.
2. Find a supported SPI x4 flash configuration memory in the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 7] that can fit the minimum flash memory image determined in Table 1 step 6.

Table 1: **Determining the Minimum Flash Memory Size**

Step	Instruction and Description	Value	Units
1	Find the FPGA bitstream length in the *7 Series FPGAs Configuration User Guide* (UG470) [Ref 1].		Mb(1)
2	Specify optional user application data size.		Mb
3	Sum bitstream length (from step 1) and user data size (from step 2).		Mb
4	Round sum from step 3 up to the next whole number of megabits. This is the minimum size of each configuration (plus optional user data) image.		Mb
5	Set the number of configuration image spaces in the flash memory: • Specify 1 for a simple, single FPGA configuration. • Specify 2 for a MultiBoot-Fallback FPGA configuration solution that requires separate configuration image spaces for a MultiBoot update image and for a fallback, known-good (golden) image. See the *MultiBoot with 7 Series FPGAs and SPI Application Note* (XAPP1247) [Ref 4] for MultiBoot-Fallback images and data space required for additional timer images.		(Integer)
6	Multiply rounded sum from step 4 by the number of configuration image spaces in the flash memory from step 5. This is the minimum size of flash memory.		Mb

Notes:

1. One megabit (Mb) = 2^20 bits.

This application note's reference design includes a SPI x4 flash (also known as a *quad* or *multi-I/O SPI* flash) that supports data transfers over a 4-bit-wide data bus. These SPI x4 flash devices extend the SPI bus standard that normally transfers 1-bit-wide serial (x1) data in each direction over two data lines, with two additional data lines. This enables 4-bit wide data transfers in either direction using the two existing data lines and two additional data lines as a 4-bit-wide (x4) data bus.

Designing the Board

The relevant FPGA configuration pin connections and descriptions are illustrated in Figure 2 and listed in Table 4.

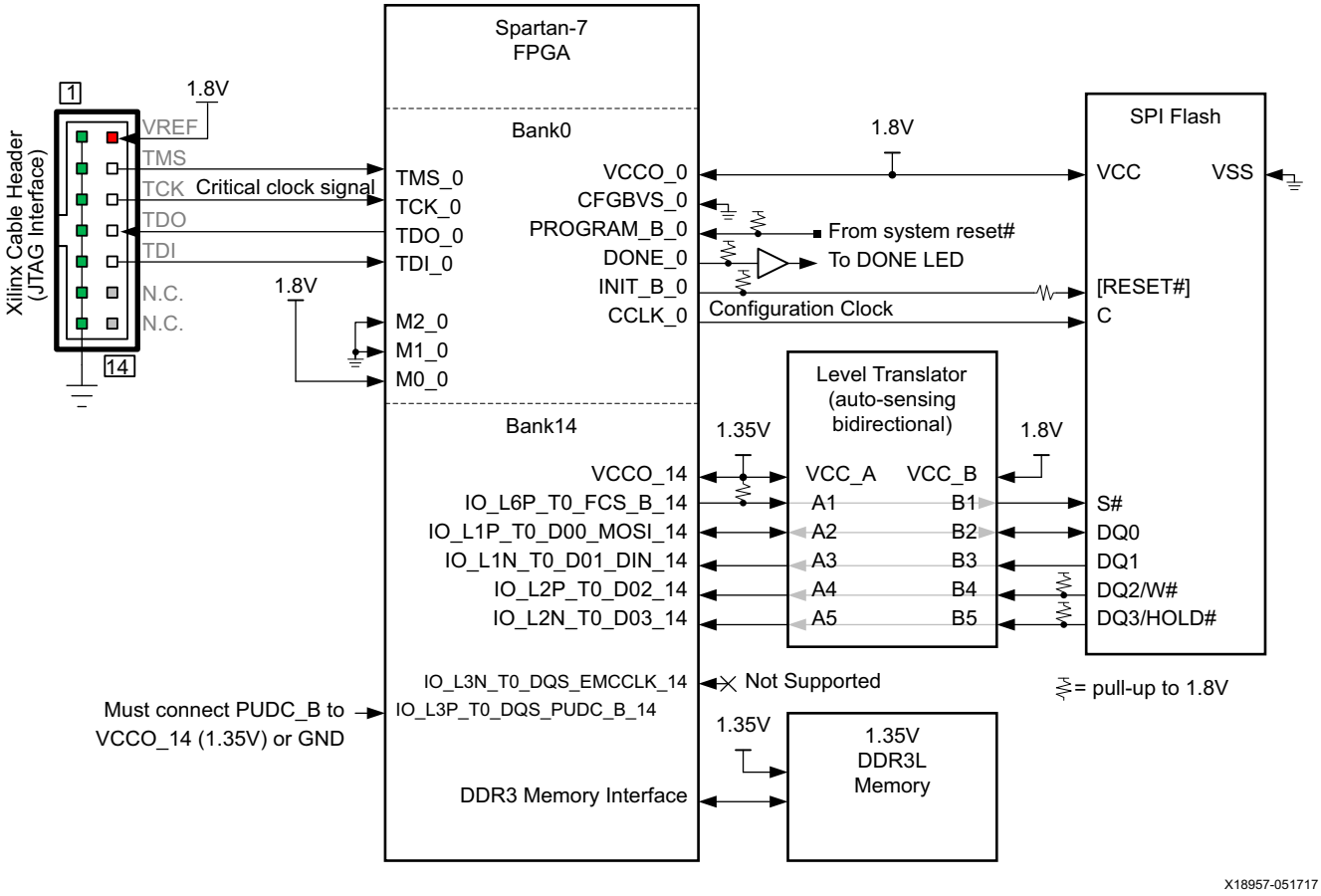

Figure 2: **FPGA and SPI Flash Configuration Schematic Diagram**

Table 2: **FPGA Configuration Pin Descriptions and Connections**

FPGA Pin Name_Bank#	FPGA Pin Direction	Pin Description and Board Connection
VCCO_0	(Power)	FPGA dedicated configuration bank 0 I/O power pin. **IMPORTANT:** Must connect to 1.8V.
VCCO_14	(Power)	FPGA multi-use bank 14 I/O power pin. Connect to 1.35V.
CFGBVS_0	Input	Configuration bank voltage select input for bank 0. Must connect to GND for 1.8V bank 0 I/O configuration.
M2_0, M1_0, M0_0	Input	Configuration mode. Set M[2:0] = 001 for SPI configuration mode.
TCK_0	Input	JTAG test clock input. Connect to JTAG cable connector TCK pin. Critical JTAG clock signal. Apply appropriate clock signal board design and layout rules.

Table 2: **FPGA Configuration Pin Descriptions and Connections *(Cont'd)***

FPGA Pin Name_Bank#	FPGA Pin Direction	Pin Description and Board Connection
TMS_0	Input	JTAG test mode select input. Connect to JTAG cable connector TMS pin.
TDI_0	Input	JTAG test data input. Connect to JTAG cable connector TDI pin, or if Spartan-7 FPGA is in a multiple-device JTAG daisy chain, connect to TDO pin of prior device in the chain.
TDO_0	Output	JTAG test data output. Connect to JTAG cable connector TDO pin, or if Spartan-7 FPGA is in a multiple-device JTAG daisy chain, connect to TDI pin of next device in the chain.
PROGRAM_B_0	Input	Active-Low configuration reset input pin. Connect to an external 4.7kΩ resistor to V_{CCO_0} (1.8V).
INIT_B_0	Bidirectional: input and open-drain output	Active-Low configuration initialization or error status output pin. If the SPI flash has a dedicated, active-Low RESET# pin, recommend connecting INIT_B via a 0Ω resistor to the SPI flash RESET# pin, which assures the SPI flash is reset from any incidental activity whenever a new FPGA configuration is initiated. Since many substitute SPI flash devices do not have this dedicated RESET# pin, the 0Ω resistor allows the INIT_B pin to be disconnected from the SPI flash pin. Also recommend connecting to a buffer to drive an LED to indicate FPGA configuration initialization or error when INIT_B is Low.
DONE_0	Bidirectional: input and open-drain output	Configuration done status output pin. The DONE pin has a weak internal pull-up. If DONE is connected to an external circuit that specifies a maximum Low-to-High rise time, then add an external resistor, e.g. 330Ω, to V_{CCO_0} (1.8V) to ensure a sufficiently fast Low-to-High rise time. Also recommend connecting to a buffer to drive an LED to indicate FPGA configuration success when DONE is High.
CCLK_0	Output	Configuration clock output pin. Connect to the SPI flash clock (C) input pin. Apply appropriate clock signal board design and layout rules to ensure clean rising and falling edges.
IO_L6P_T0_FCS_B_14	Output	Active-Low SPI flash chip select (FCS_B) output. Connect to the SPI flash chip select (S#) pin via a level translator and connect to an external 2kΩ resistor pull-up to 1.35V to deselect the SPI flash by default.
IO_L1P_T0_D00_MOSI_14	Bidirectional	Standard SPI bus, master-output, slave input (MOSI) signal; and quad (x4) data bus bit 0 (DQ0) input during the quad-SPI flash read extension to the SPI bus. Connect to the SPI flash DQ0 pin via an auto-sensing, bidirectional level translator.
IO_L1N_T0_D01_DIN_14	Input	Standard SPI bus, master-data-input (DIN), slave output signal; and quad (x4) data bus bit 1 (DQ1) input during the quad SPI flash read extension to the SPI bus. Connect to the SPI flash DQ1 pin via a level translator.

Table 2: **FPGA Configuration Pin Descriptions and Connections *(Cont'd)***

FPGA Pin Name_Bank#	FPGA Pin Direction	Pin Description and Board Connection
IO_L2P_T0_D02_14	Input	Quad (x4) data bus bit 2 (DQ2) input during the quad SPI flash read extension to the SPI bus. Connect to the SPI flash DQ2 pin via a level translator, and connect SPI flash DQ2 pin to an external 2kΩ resistor pull-up to 1.8V to disable the SPI flash active-Low write-protect function.
IO_L2N_T0_D03_14	Input	Quad (x4) data bus bit 3 input during the quad SPI flash read extension to the SPI bus. Connect to the SPI flash DQ3 pin via a level translator, and connect SPI flash DQ3 pin to an external 2kΩ resistor pull-up to 1.8V to disable the SPI flash active-Low HOLD function.
IO_L3P_T0_DQS_PUDC_B_14	Input	Active-Low enable for internal SelectIO pull-ups during configuration. Must connect to either GND or VCCO_0 (1.8V), depending on your system preference for enabling the FPGA SelectIO internal pull-up, or not, before and during FPGA configuration.
IO_L3N_T0_DQS_EMCCLK_14	Input	The external master configuration clock (EMCCLK) source feature is not supported with a V_{CCO} of 1.35V.

Note: Refer to the *Zynq-7000 All Programmable SoC and 7 Series Devices Memory Interface Solutions User Guide* (UG586) [Ref 3] for the DDR3 memory interface pinout and the *7 Series FPGAs PCB Design Guide* (UG483) [Ref 5] for board design guidelines.

Determining the Configuration Clock Rate

The starting configuration clock rate is specified as F_{MCCK_START} in the *Spartan-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS189) [Ref 6], and is approximately 3 MHz. The FPGA sends read commands to the SPI flash at this starting clock rate for reading the configuration bitstream from the SPI flash. By default, the FPGA continues to clock the SPI read data at the same clock rate as the starting clock rate, but a command in the beginning of the bitstream can instruct the FPGA to change the clock rate on-the-fly to a different rate specified via the Vivado design tool BITSTREAM.CONFIG.CONFIGRATE property for transferring the remainder of the bitstream. The allowed configuration rates are: 3 (default), 6, 9, 12, 16, 22, 26, 33, 40, 50, or 66, where each setting is the nominal frequency in MHz. The actual CCLK frequency from the FPGA can vary from the nominal setting by the percent tolerance ($F_{MCCKTOL}$) specified in the *Spartan-7 FPGAs Data Sheet: DC and AC Switching Characteristics* [Ref 6]. Use Table 3 to calculate a maximum configuration rate and typical configuration time.

Table 3: **Calculating the Maximum Configuration Rate**

Step	Step Instruction and Description	Value	Unit	Example(1)
	Calculating Maximum BITSTREAM.CONFIG.CONFIGRATE Setting			
1	Specify the SPI flash clock Low to output valid time (t_{CLQV}).		ns	6.0
2	Specify the level translator push-pull signal propagation delay time.		ns	9.9
3	Use this FPGA D[03:00] setup time (3.5 ns).	3.5	ns	3.5

Table 3: **Calculating the Maximum Configuration Rate *(Cont'd)***

Step	Step Instruction and Description	Value	Unit	Example[1]
4	Estimate the sum of the following two board trace propagation delay times: • FPGA CCLK output pin to SPI flash C input pin. • Longest path of the data signals from the SPI flash DQ output pin to level translator and from the level translator to the FPGA D[n] input pin. Assuming a trace delay of 165 ps per inch, the sum of the above for a SPI flash that is 3 inches or closer to the FPGA is approximately 1.0 ns.		ns	1.0
5	Sum of lines 1, 2, 3, and 4. This is the minimum CCLK period, assuming the Vivado tool BITSTREAM.CONFIG.SPI_FALL_EDGE property is set to Yes.		ns	20.4
6	The maximum CCLK frequency = 1/(sum from step 5).		MHz, max	49
7	If applicable, specify the level translator maximum push-pull signal data rate.		Mb/s, max	30
8	Select the lower of the applicable values from step 6 and step 7.		MHz, max	30
9	Specify the frequency tolerance, master mode with respect to nominal CCLK ($F_{MCCKTOL}$).		%	50
10	The maximum configuration rate setting ≤ [(step 8) * 100%] / [100% + (step 9)].		MHz, typ	20
11	From the allowable CONFIGRATE settings (3, 6, 9, 12, 16, 22, 26, 33, 40, 50, or 66), choose a value that is ≤ (line 10).		MHz, typ	16
	Calculating Configuration Time			
12	Specify the bitstream length from the *7 Series FPGAs Configuration User Guide* (UG470) [Ref 1].		bits	4,310,752
13	Typical configuration time ~= (line 12)/(line 11)/1,000,000/4 bits per cycle.[2]		s, typ	0.067

Notes:

1. Example values are for a Spartan-7 XC7S6 FPGA, example SPI flash, and TXS0108E level translator where the timing for 1.35V translation is interpolated between 1.2V and 1.5V specifications.
2. Set the Vivado design tool BITSTREAM.GENERAL.COMPRESS property to TRUE to potentially reduce the bitstream length and thus reduce the configuration time versus the typical configuration time.

Note: The external master configuration clock feature that uses a clock source from the EMCCLK pin on bank 14 is not supported in this application with bank 14 = 1.35V. The corresponding bitstream property, BITSTREAM.CONFIG.EXTMASTERCCLK_EN, must remain in the default DISABLE setting. The Vivado design tool flags all other settings for this property as an unsupported I/O voltage error and will not generate a bitstream.

Specifying 1.35V-compatible I/O Standards for Bank 14 Pins

The memory interface generator should generate appropriate pin properties for the memory interface pins. For other user-defined pins in bank 14 of the FPGA design, including pins through which the design accesses the SPI flash memory, a 1.35V-compatible I/O standard must be specified, such as the SSTL135_R I/O standard, since there is no LVCMOS135 I/O standard. For example,

```
set_property IOSTANDARD SSTL135_R [get_ports FCS_B]
```

Building the FPGA Bitstream

The following Vivado design tool properties are required for a bitstream that works with this configuration solution:

```
set_property CONFIG_VOLTAGE 1.8 [current_design]
set_property CFGBVS GND [current_design]
```

To reduce the configuration time, the following Vivado design tool properties can be set:

```
set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE rate [current_design]
```

(where rate = the CONFIGRATE from line 11 in Table 3)

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
```

If the following property is set for a MultiBoot-Fallback configuration (see the *MultiBoot with 7 Series FPGAs and SPI Application Note* (XAPP1247) [Ref 4]) and has a byte address value ≥ 2^24, which is almost never the case for a Spartan-7 FPGA design:

```
set_property BITSTREAM.CONFIG.NEXT_CONFIG_ADDR address [current_design]
```

Then the following Vivado design tool property must also be set to support the 32-bit address value:

```
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
```

As noted in the Determining the Configuration Clock Rate, page 6, the external master configuration clock (EMCCLK) feature is not supported, and thus, the following Vivado design tool property must remain in its default setting:

```
set_property BITSTREAM.CONFIG.EXTMASTERCCLK_EN DISABLE [current_design]
```

Generating the Configuration Memory File

For Vivado design tool instructions on generating the configuration memory file from a bitstream, refer to the *Creating a Configuration Memory File* section in the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 7]. For MultiBoot-Fallback configuration, also see write_cfgmem examples in the *MultiBoot with 7 Series FPGAs and SPI Application Note* (XAPP1247) [Ref 4].

An example Vivado design tool Tcl command for generating a simple configuration memory file containing just one FPGA bitstream is as follows:

```
write_cfgmem -force -format mcs -interface spix4 -size spi_flash_size
-loadbit "up 0 filename.bit" -file filename.mcs
```

where

`-interface spix4` = Correlates with the BITSTREAM.CONFIG.SPI_BUSWIDTH (see above)

`spi_flash_size` = Integer SPI flash size in megabits

`filename.bit` = Input bitstream file

`filename.mcs` = Output configuration memory file in MCS format

Programming the SPI Flash Memory

The Vivado design tool can indirectly program the contents of a configuration memory file into the SPI configuration flash via a JTAG cable connection to the Spartan-7 FPGA. To setup the FPGA for indirect SPI flash programming, the Vivado design tool normally clears the FPGA configuration and then downloads its own pre-built bitstream to the FPGA to connect from the FPGA JTAG port to the FPGA SPI flash interface. However, the Vivado design tool pre-built bitstream for indirect SPI flash programming supports only the standard configuration interface voltages (3.3V, 2.5V, 1.8V, or 1.5V) as specified in the *7 Series FPGAs Configuration User Guide* (UG470) [Ref 1]. Thus, a special procedure that downloads a special version of the pre-built bitstream is required for indirect programming of the SPI flash connected to the 1.35V bank 14. This special procedure follows:

Pre-requisites:

- Vivado design tools or lab tools is installed on the host PC.
- The reference design files for this application note are unzipped to a directory on the host PC.
- JTAG cable is connected between the host PC and target board, containing the Spartan-7 FPGA.
- The target board power is ON.

Procedure:

1. Start the Vivado design tool in the directory where the reference design files are located.
2. See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 7] for *Opening the Hardware Manager* and *Opening Hardware Target Connections*.
3. In the Vivado Hardware Manager, Hardware view, identify the Spartan-7 FPGA in the JTAG chain. The Spartan-7 device is identified as "xc7s50_0" in Figure 3.

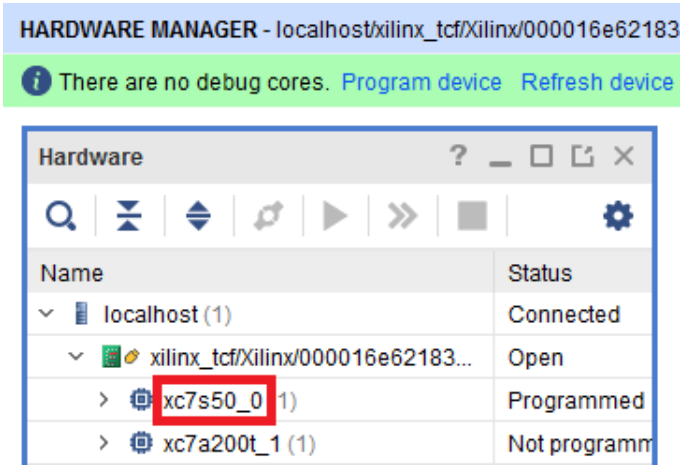

Figure 3: **Vivado Hardware Manager View of Devices in JTAG Chain**

4. Close only the Hardware Manager, but keep the Vivado design tool open.
5. In the Vivado Tcl Console command line, source the `vivado_spartan7_135Vbank14_program_spi_flash.tcl` script:

   ```
   source vivado_spartan7_135Vbank14_program_spi_flash.tcl
   ```

6. Execute the following Vivado design tool Tcl command:

   ```
   vivado_spartan7_135Vbank14_program_spi_flash device filename.mcs
   ```

   where

   device = the Spartan-7 FPGA device identified in step 3, e.g. "xc7s50_0"

   filename.mcs = the configuration memory file to be programmed into the SPI flash

IMPORTANT: This procedure with its special pre-built bitstreams is *ONLY* for use with Spartan-7 FPGAs where bank 14 is powered at 1.8V or lower.

New Board Bring-Up and Debug

The following is a list for checking new boards and the SPI flash configuration solution after initial board checks for proper power supply voltages and basic connectivity:

1. Power-on the board. Assuming the SPI flash is blank, check for the following:
   a. DONE pin = Low
   b. INIT_B pin = High, following a brief Low period (2-50 ms) from initial power on

   If either result is not as expected, re-check the power supplies and re-check the Spartan-7 FPGA schematic connections.

2. Connect the JTAG cable and use Vivado Hardware Manager to detect devices on the JTAG chain. See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 7].

a. One of the device(s) in the JTAG chain should be the Spartan-7 FPGA.

b. Repeat the Hardware Manager refresh device operation multiple times and check for consistent results, e.g., same device(s) appear in the JTAG chain and the device IDCODE values remain constant for each refresh.

If either result is not as expected, re-check the JTAG cable connections, re-check the JTAG connector and signal schematic connections, check for opens or shorts in the JTAG signals, and check for signal integrity issues on the JTAG TCK signal.

3. Use the Vivado Hardware Manager to program the Spartan-7 FPGA with a simple test bitstream. Check that DONE is High after bitstream programming.
4. Use the special procedure in Programming the SPI Flash Memory, page 9 to program the SPI flash memory with a bitstream from a configuration memory file.

   a. Check that DONE is High during the SPI flash programming procedure

   b. Pulse the FPGA PROGRAM_B pin or cycle power, and check that DONE goes High after the expected configuration time, calculated in Table 3.

   If either result is not as expected, check that the FPGA M[2:0] pins are set to `001` for SPI configuration mode, use the Vivado Hardware Manager to check the Spartan-7 FPGA configuration status register values, probe the SPI flash pins for expected CCLK and FCS_B signal behavior, probe the SPI flash data pins for expected command and data, probe CCLK signal for signal interity issues, or repeat SPI flash programming but with a simple SPI x1 mode bitstream set for the default configuration rate.

Reference Design

Download the reference design files for this application note from the Xilinx website. Table 4 shows the reference design matrix.

Table 4: **Reference Design Matrix**

Parameter	Description
General	
Developer names	Randal Kuramoto
Target devices	Spartan-7 FPGAs
Source code provided	No
Source code format	N/A
Design uses code and IP from existing Xilinx application note and reference designs or third party	No
Simulation	
Functional simulation performed	N/A
Timing simulation performed	N/A

Table 4: **Reference Design Matrix *(Cont'd)***

Parameter	Description
Test bench used for functional and timing simulations	N/A
Test bench format	N/A
Simulator software/version used	N/A
SPICE/IBIS simulations	No
Implementation	
Synthesis software tools/versions used	N/A
Implementation software tools/versions used	Vivado design tools 2017.1
Static timing analysis performed	Yes. The timing analysis for the configuration rate on sample hardware was performed as shown in the example in Table 3.
Hardware Verification	
Hardware verified	Yes. Basic configuration and indirect SPI flash programming functionality was verified on sample hardware implemented as shown in Figure 2 with a TXS0108E level translator at room temperature.
Hardware platform used for verification	Internal Xilinx test board

Conclusion

This application note provides a method for configuring a Spartan®-7 FPGA from a 1.8V serial peripheral interface (SPI) NOR flash memory connected to the FPGA dedicated I/O bank 0 at 1.8V and multi-use FPGA I/O bank 14 at 1.35V.

For applications requiring connection of a 1.5V DDR3 memory interface to the FPGA bank 14, the same strategies can be applied from this application note, including:

- Selecting a SPI Flash
- Designing the Board
  - 1.8V SPI flash
  - FPGA bank 0, V_{CCO_0} = 1.8V
  - FPGA bank 14, V_{CCO_14} = 1.5V
  - Auto-sensing, bidirectional level translator for signals between the FPGA bank 14 and 1.8V SPI flash

  Note: The FPGA EMCCLK feature is supported for 1.5V configuration.
- Determining the Configuration Clock Rate
  - Apply the same timing guidelines from this application note.

- Specifying 1.35V-compatible I/O Standards for Bank 14 Pins
  - CONFIG_VOLTAGE property = 1.5
  - CFGBVS property = GND
  - FPGA bank 14 I/O standard for user-defined pins = LVCMOS15
- Building the FPGA Bitstream
- Generating the Configuration Memory File
- Programming the SPI Flash Memory
  - Use the same instructions and files as described in this application note.

References

This application note uses the following references:

1. *7 Series FPGAs Configuration User Guide* (UG470)
2. *Using SPI Flash with 7 Series FPGAs Application Note* (XAPP586)
3. Zynq-7000 All Programmable SoC and 7 Series Devices Memory Interface Solutions User Guide (UG586)
4. *MultiBoot with 7 Series FPGAs and SPI Application Note* (XAPP1247)
5. *7 Series FPGAs PCB Design Guide* (UG483)
6. *Spartan-7 FPGAs Data Sheet: DC and AC Switching Characteristics:* (DS189)
7. Vivado Design Suite User Guide: Programming and Debugging (UG908)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/14/2017	1.0	Initial Xilinx release.

Please Read: Important Legal Notices