APPLICATION NOTE

Cypress® Radiation Tolerant SPI Flash configuration options for Xilinx® FPGA's

Author: Rahul Patil

Associated Part Family: CYRS16B256/CYRS16B512

Xilinx FPGA's for Aerospace and Defense 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. Cypress SPI (Serial Peripheral Interface) Flash is radiation tolerant to operate in the areas of Aerospace and Defense and can be easily connected to Xilinx FPGAs in order to configure the FPGA at power-up. The Single/Quad/Octal SPI configuration mode is supported for Xilinx® Kintex Ultrascale$_{TM}$ FPGA's.

Content

1 Introduction

SPI Flash programming solutions using the CYRS16BXXX family of devices are now available for use with Xilinx Aerospace and Defense rated FPGA's. The Cypress radiation tolerant SPI Flash devices can be used in single SPI, QUAD SPI or DUAL QUAD SPI configuration options. These devices are available in 256Mb and 512Mb configuration memory densities. The Cypress SPI Radiation tolerant NOR FLASH supports the same commands and frequencies of the commercial grade Cypress SPI Flash devices. Two Cypress hardware solutions to connect and program the SPI Flash currently exist.

- Cypress SPI Flash configuration solution devkit (CYDK-NOR) is a mezzanine card which is compatible with the Alpha Data Development platform for Space grade FPGA's (ADA-SDEV-KIT2). This mezzanine card is setup to use the Xilinx ToolChain based programming approach for programming the SPI Flash.
- Standalone SPI Flash programming daughter card solution that uses a standalone Flash programmer from FlashCatUSB (FlashCatUSB mach)

2 SPI Connection 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#) s

Quad Serial peripheral interface (QSPI) expands the data bus to 4 bits and adds QPI commands. 2 QPI devices can be used in parallel to provide 8-bit databus for the SPI dual quad SPI (DQSPI) configuration interface, an additional Slave select is also used for this connection.

Following SPI/QSPI/DQSPI protocol, the master is assigned to the FPGA device and the slave to the SPI Flash device, as shown in Figure 1. Per these connections, the SPI Flash is available to configure the FPGA at power-up.

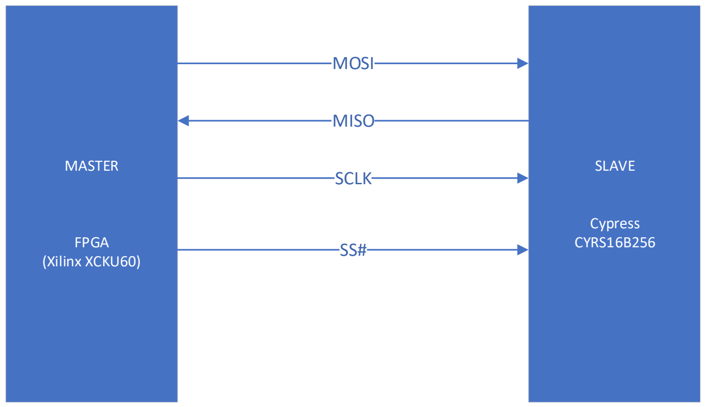


Figure 1. Direct configuring FPGA to interface with SPI Flash

Figure 2 shows the SPI Flash device connection when used in Quad SPI connection mode.

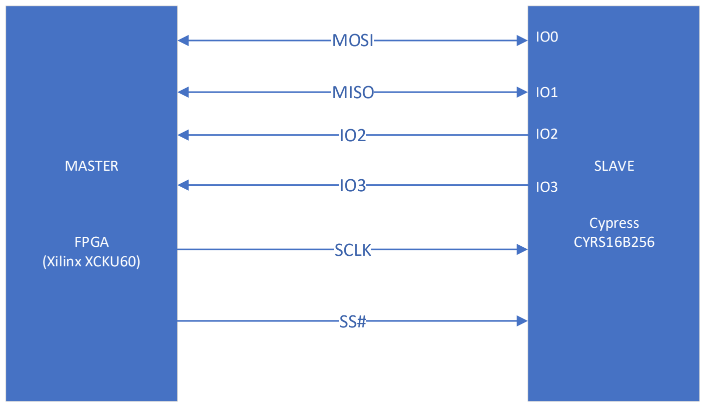


Figure 2. QSPI configuration connection with SPI Flash

Figure 3 shows the Dual Quad SPI device connection using 2 Quad SPI Flash devices in parallel. In the case of the 512Mb devices two separate Quad SPI busses are available on the same device as shown in Figure 4

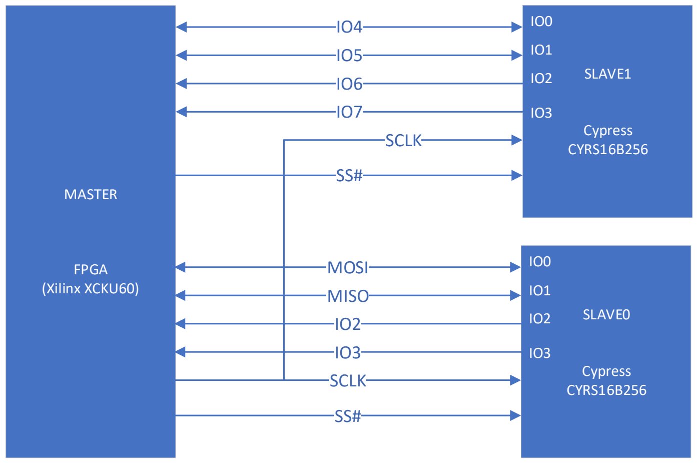


Figure 3. DQSPI configuration connection with 2 x 256Mb SPI Flash

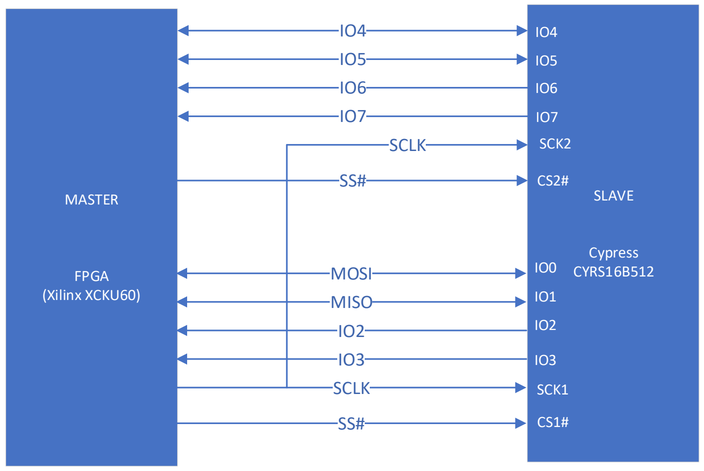


Figure 4. DQSPI configuration connection with single 512Mb SPI Flash

3 SPI Flash Connections

The SPI pins to FPGA configuration pins connection table is shown below. Some FPGA pins will not have a corresponding SPI Flash pin but are essential in configuring the FPGA to the correct mode and I/O voltage level. The CYRS16B256 and CYRS16B512 only support 3.3V I/O CMOS voltage levels.

The pin connection table is based on the Cypress CYRS16B256/CYRS16B512 parts.

SPI x1 pin name	SPI x4 pin name	SPI x8 pin name	FPGA pin name	Description
SI/IO0	SI/IO0	SI/IO0	D00_MOSI	Master out slave in, when in 1-bit SPI mode. D[0] of D[3:0] when in Quad SPI mode

SO/IO1	SO/IO1	SO/IO1	D01_DIN	Slave out master in when in 1-bit SPI mode. D[1] of D[3:0] when in Quad SPI mode. D[1] of D[7:0] when in Dual Quad SPI mode
Not applicable	IO2	IO3	D02	D[2] of D[3:0] when in Quad SPI mode. D[2] of D[7:0] when in Dual Quad SPI mode
Not applicable	IO3	IO4	D03	D[3] of D[3:0] when in Quad SPI mode. D[3] of D[7:0] when in Dual Quad SPI mode
Not applicable	Not applicable	IO4	D04	D[4] of D[7:0] when in Dual Quad SPI mode
Not applicable	Not applicable	IO5	D05	D[5] of D[7:0] when in Dual Quad SPI mode
Not applicable	Not applicable	IO6	D06	D[6] of D[7:0] when in Dual Quad SPI mode
Not applicable	Not applicable	IO7	D07	D[7] of D[7:0] when in Dual Quad SPI mode
CS#	CS#	CS1#	FCS_B	Active low chip select for D[3:0], connect with a pullup to VCCO_0 ≤ 4.7KΩ
Not applicable	Not applicable	CS2#	FCS2_B	Active low chip select for D[7:4], connect with a pullup to VCCO_0 ≤ 4.7KΩ
SCK	SCK	SCK1#	CCLK	FPGA sourced configuration clock
Not applicable	Not applicable	SCK2#		In DQSPI mode connect this clock pin on the SPI Flash to the CCLK
			CFGBVS	This pins value determines the I/O operating voltage range of Bank0. Tie this pin to high or VCCO_0. The VCCO_0 voltage will be 3.3v since Cypress SPI Flash only supports 3.3V I/O
			M[2:0]	Mode configuration bits set to M[2:0] =001, Master SPI
			PROGRAM_B	Active low input, when pulsed low FPGA configuration is cleared, new configuration is initiated.
			INIT_B	Active low FPGA initialization pin, after initialization is completed FPGA will tristate this pin. And will begin FPGA configuration. External masters can force this pin low to delay the configuration by driving it low actively. This pin is connected with a pullup to VCCO_0 ≤ 4.7KΩ
			DONE	When DONE is high it indicates the FPGA programming was completed. This pin is connected with a pullup to VCCO_0 ≤ 4.7KΩ

4 Programming Options

There are 3 ways to program the SPI flash, they are discussed in more detail in the following sections:

- Xilinx Toolchain based programming, this is achieved by communicating to the FPGA through the JTAG interface and programming the flash indirectly through the FPGA.
- Direct programming the SPI Flash, additional master is connected to the SPI bus
- Standalone SPI flash programmer, SPI flash is programmed using a third-party standalone NOR Flash programmer.

4.1 Xilinx ToolChain based programming

Xilinx Vivado tools can be used to generate a bit file to program through the JTAG interface of the FPGA using the Xilinx programming cable. The Vivado (2019.2 and later versions) toolchain supports generation of the bit file and programming in this mode for the Kintex XCKU060 FPGA. In this mode the tools will program the bit file through the Xilinx configuration cable connected to the JTAG interface of the FPGA, the FPGA SPI configuration interface is then used to program the incoming bit file on the JTAG interface to the SPI Flash.

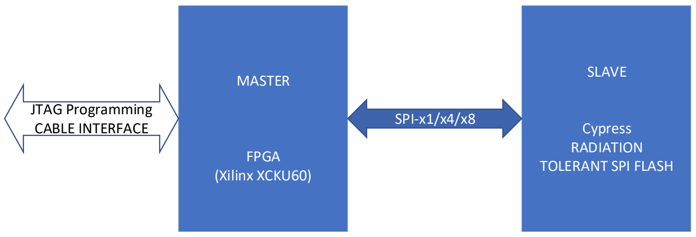
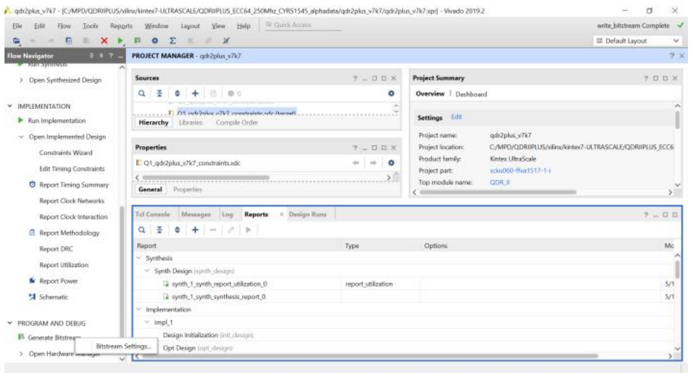


Figure 5. Xilinx tool-based programming

4.1.1 Bit file generation

In the Vivado IDE after implementation has been completed, before generating the bit file, choose the SPI configuration method (i.e. SPI, QSP, DQSPI, master serial mode etc.). The steps to do this are as follows:

1. Open the Implemented design in Vivado IDE.
2. Right click on the Generate bitstream tab option and the bitstream settings will pop-up, click on this option



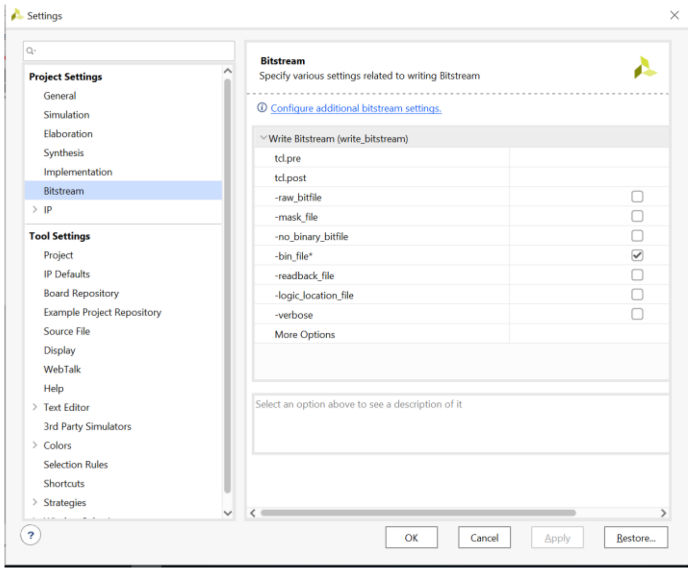
qdr2plus_v7k7 - [C:/MPD/QDRIIPLUS/xilinx/kintex7-ULTRASCALE/QDRIIPLUS_ECC64_250Mhz_CYRS1545_alphadata/qdr2plus_v7k7/qdr2plus_v7k7.xpr] - Vivado 2019.2

Q1_qdr2plus_x7k7_constraints.xdc

Project name:	qdr2plus_v7k7
Project location:	C:/MPD/QDRIIPLUS/xilinx/kintex7-ULTRASCALE/QDRIIPLUS_ECC6
Product family:	Kintex UltraScale
Project part:	xcku060-ffva1517-1-i
Top module name:	QDR_II

Report	Type	Options	Mo
Synthesis			
Synth Design (synth_design)			
synth_1_synth_report_utilization_0	report_utilization		5/1
synth_1_synth_synthesis_report_0			5/1
Implementation			
impl_1			
Design Initialization (init_design)			
Opt Design (opt_design)			

3. In the pop-up window select the .bin file options and click on the “configure additional bitstream settings”


Settings

Project Settings: General, Simulation, Elaboration, Synthesis, Implementation, Bitstream, IP

Tool Settings: Project, IP Defaults, Board Repository, Example Project Repository, Source File, Display, WebTalk, Help, Text Editor, 3rd Party Simulators, Colors, Selection Rules, Shortcuts, Strategies

Bitstream

Specify various settings related to writing Bitstream

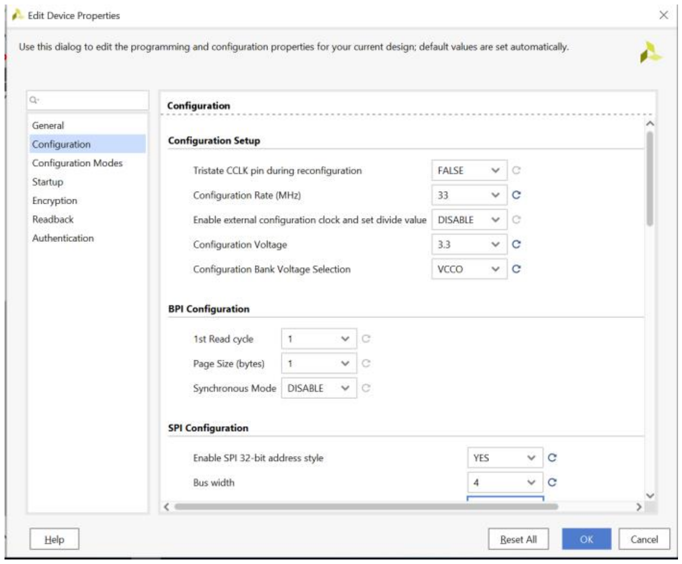
Configure additional bitstream settings.

Write Bitstream (write_bitstream)

tcl.pre	
tcl.post	
-raw_bitfile	☐
-mask_file	☐
-no_binary_bitfile	☐
-bin_file*	☑
-readback_file	☐
-logic_location_file	☐
-verbose	☐
More Options	

Select an option above to see a description of it

4. In the next widow select the "configuration" option on the left-hand side of the window panel. All the configuration options should now be seen on the right-hand side of the window panel.


Edit Device Properties

Use this dialog to edit the programming and configuration properties for your current design; default values are set automatically.

General, Configuration, Configuration Modes, Startup, Encryption, Readback, Authentication

Configuration

Configuration Setup

Tristate CCLK pin during reconfiguration	FALSE
Configuration Rate (MHz)	33
Enable external configuration clock and set divide value	DISABLE
Configuration Voltage	3.3
Configuration Bank Voltage Selection	VCCO

BPI Configuration

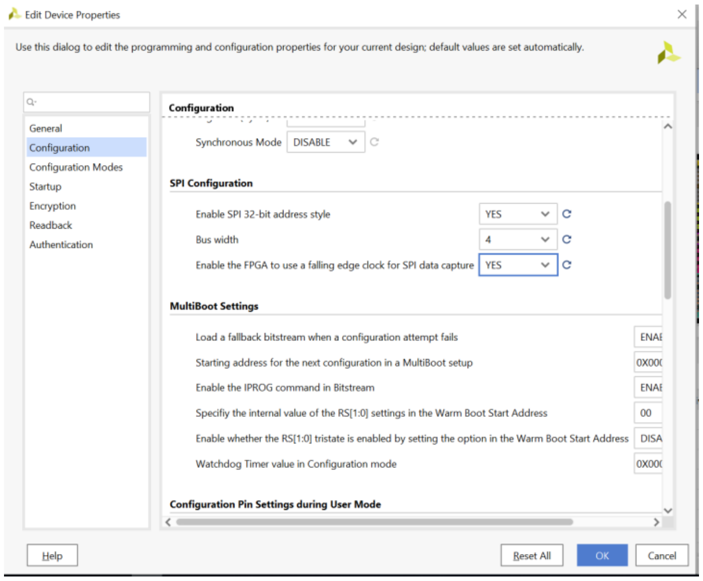
1st Read cycle	1
Page Size (bytes)	1
Synchronous Mode	DISABLE

SPI Configuration

Enable SPI 32-bit address style	YES
Bus width	4

In the configuration setup the following is changed for default values

- Configuration Rate (MHz): 33Mz (The clock frequency will vary depending on the board design and other factors that may limit the max frequency of operation for the configuration interface)
- Configuration Voltage: 3.3V
- Configuration bank voltage Selection: VCCO



Use this dialog to edit the programming and configuration properties for your current design; default values are set automatically.

Configuration

Synchronous Mode: DISABLE

SPI Configuration

Enable SPI 32-bit address style: YES

Bus width: 4

Enable the FPGA to use a falling edge clock for SPI data capture: YES

MultiBoot Settings

Load a fallback bitstream when a configuration attempt fails: ENAB

Starting address for the next configuration in a MultiBoot setup: 0X000

Enable the IPROG command in Bitstream: ENAB

Specifiy the internal value of the RS[1:0] settings in the Warm Boot Start Address: 00

Enable whether the RS[1:0] tristate is enabled by setting the option in the Warm Boot Start Address: DISA

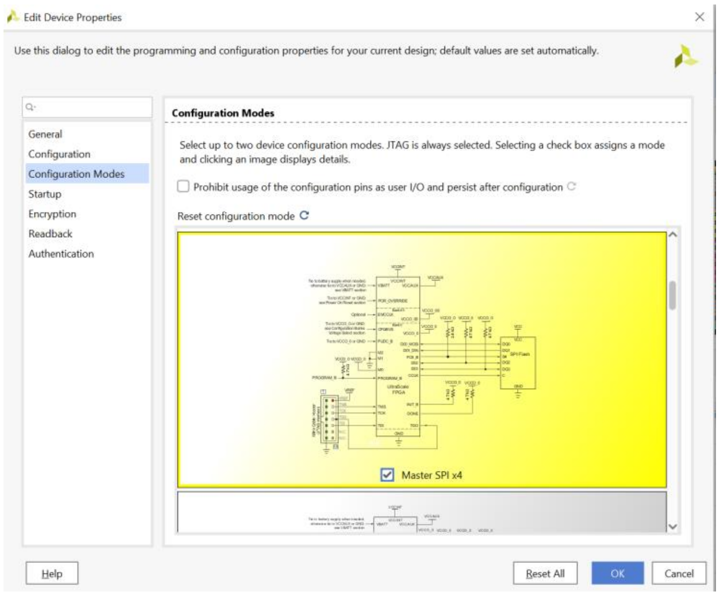
Watchdog Timer value in Configuration mode: 0X000

Configuration Pin Settings during User Mode

In the SPI configuration section, the following was changed.

- Enable SPI 32-bit address style: YES
- Bus width: 4 (set for QSPI, if DQSPI set to 8)
- Enable the FPGA to use a falling edge clock for the SPI data capture: YES

5. Select "configuration modes" in the left-hand side panel of the window and select the corresponding SPI mode in this case Master SPI x4. Click ok to accept the settings and click ok to close the setting window.



Use this dialog to edit the programming and configuration properties for your current design; default values are set automatically.

Configuration Modes

Select up to two device configuration modes. JTAG is always selected. Selecting a check box assigns a mode and clicking an image displays details.

☐ Prohibit usage of the configuration pins as user I/O and persist after configuration

Reset configuration mode

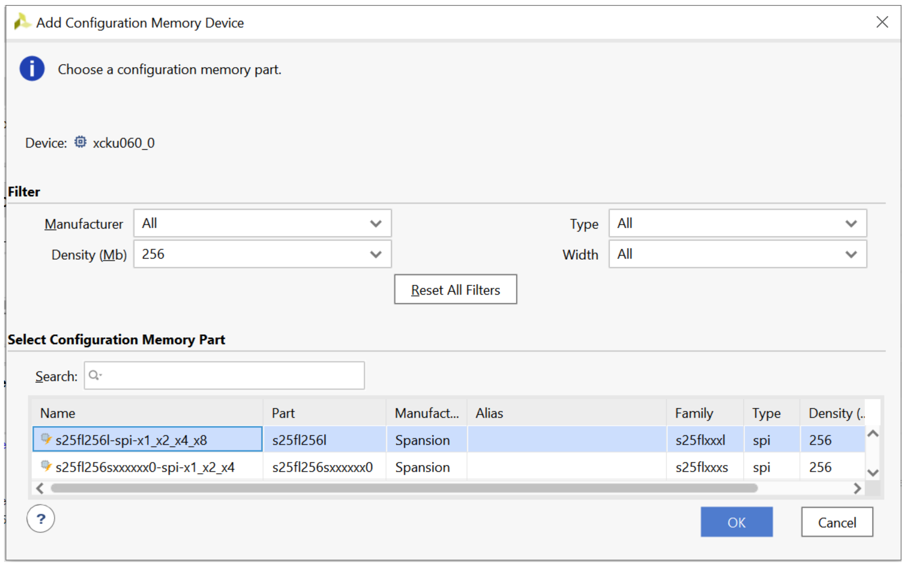
☑ Master SPI x4

6. Generate bitstream and. bit and .bin file will be generated with the QSPI programming options. Similar steps can be followed for the DQSPI programming file also.

4.1.2 Programming the flash device

1) Generate the bit file using the Xilinx Vivado flow (version 2019.2 and later support the Rad-Hard SPI flash natively).

2) Open the Hardware Manager in the Program and Debug section of the IDE.
3) Connect to the hardware and setup the Xilinx device (select Kintex Radiation tolerant device: XCKU060).
4) Add the correct configuration memory by selecting **Spansion S25fl256l-spi-x1_x4_x8**



Choose a configuration memory part.

Device: xcku060_0

Filter

Manufacturer: All
Type: All
Density (Mb): 256
Width: All

Select Configuration Memory Part

Search:

Name	Part	Manufact...	Alias	Family	Type	Density (...
s25fl256l-spi-x1_x2_x4_x8	s25fl256l	Spansion		s25flxxxl	spi	256
s25fl256sxxxxxx0-spi-x1_x2_x4	s25fl256sxxxxxx0	Spansion		s25flxxxs	spi	256

Figure 6. Cypress Radiation tolerant SPI Flash selection in Vivado IDE

5) Follow Vivado IDE instructions on selecting the bin file and programming the SPI Flash device.

4.2 Direct Programming the SPI Flash

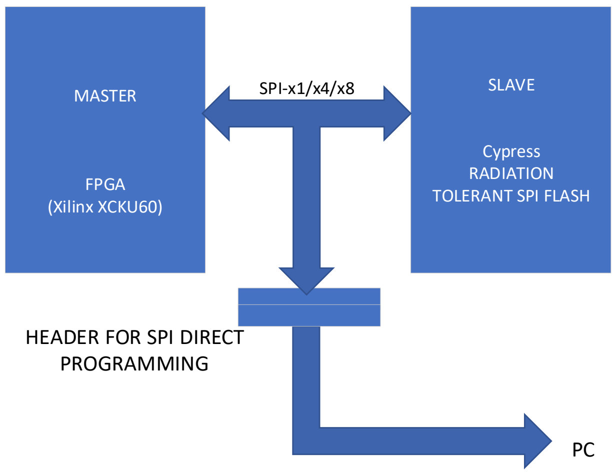


Figure 7. Direct Programming of SPI FLASH

In this mode of programming the SPI Flash can be programmed directly by another SPI master such as a microcontroller or PC.

The FPGA supplies the CCLK output from its internal oscillator to drive the clock input of SPI Flash. ISP (In-System Programming control) signal 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 setting. The following figure shows the detailed connection for this type of programming option.

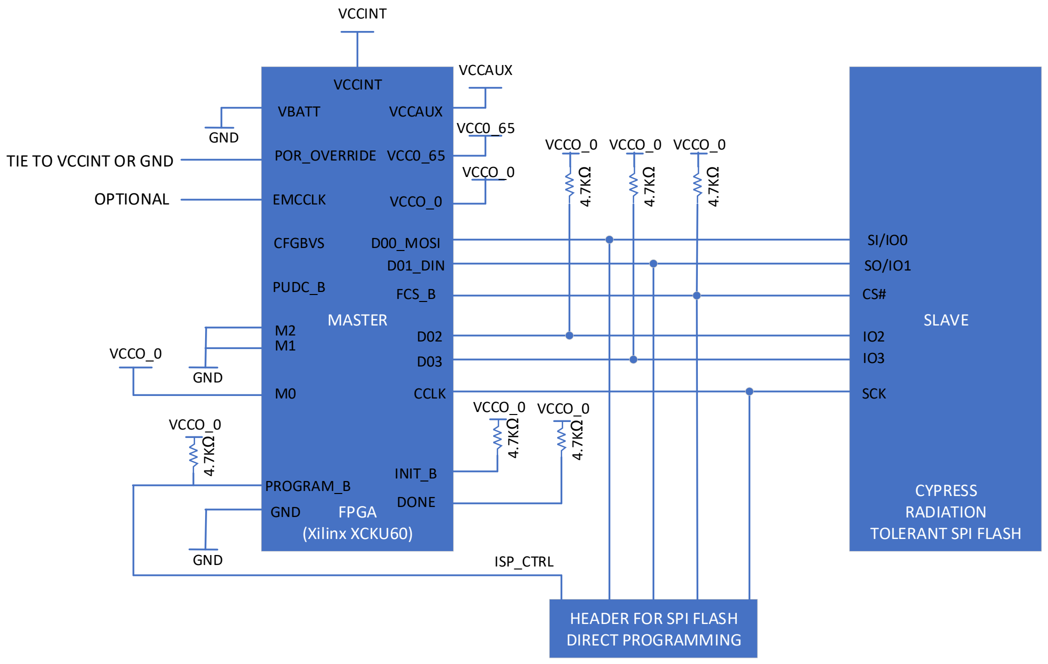


Figure 8. Detailed connections for SPI direct programming

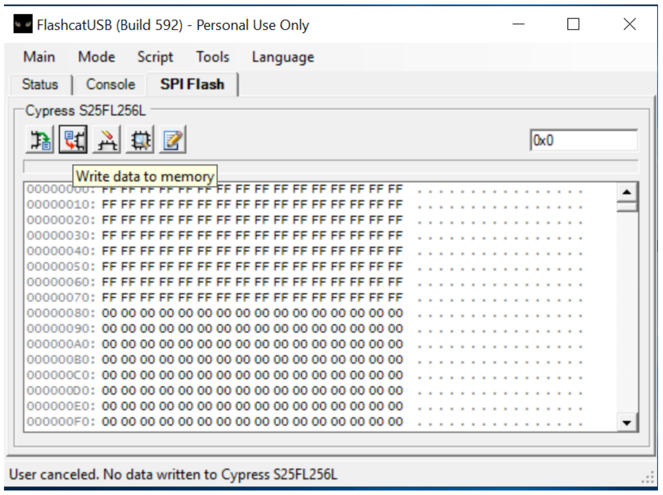
4.3 Standalone SPI Flash Programmer

A standalone SPI Flash Programmer can also be used to program the SPI Flash device. Generic off the shelf programmers are widely available. EmbeddedComputers® have developed a RAD-HARD SPI Flash device socket board that will plug into their FlashCatUSB-Mach programmer (https://www.embeddedcomputers.net/products/FlashcatUSB_Mach1/).

Using the FlashCatUSB software (https://www.embeddedcomputers.net/software/) the SPI Flash device can be programmed from a computer with a USB port. Install the USB driver and run the software as instructed in the user manual. When the FlashCatUSB-Mach is connected to the USB port, it will detect the Flash device in the Socket. With the flash device successfully detected the software will create a Flash tab. All Flash read and write operations will be done through this tab.

In Vivado generate a .bin file that can be used by the FlashCatUSB software. Follow steps below to program the SPI Flash

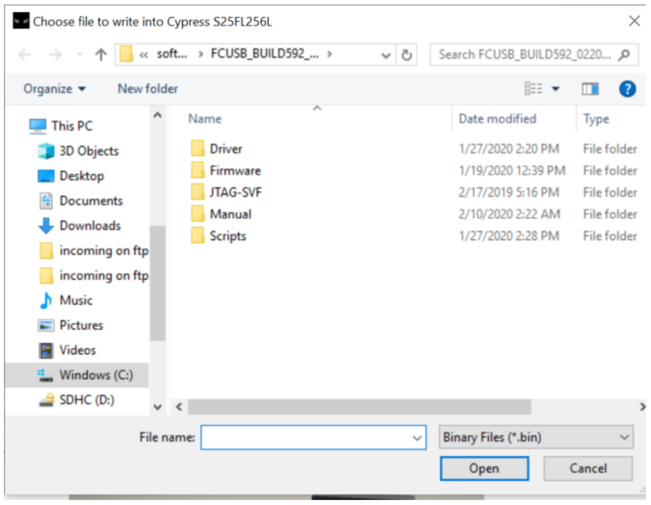
1. Run the FlashCatUSB.exe program. The GUI should now be visible. Connect the FlashCatUSB programmer with the SPI Flash plugged into the programmer board. On the Status tab it should show the software is now connected to the FlashCatUSB programmer.
2. Click on the SPI Flash tab in the GUI and it should switch to the following display



```
00000000: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000010: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000030: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000050: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000060: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000070: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

User canceled. No data written to Cypress S25FL256L

3. Click on the “Write Data to memory” button. A new window will pop up asking for the location of the bin file to program to the SPI Flash.

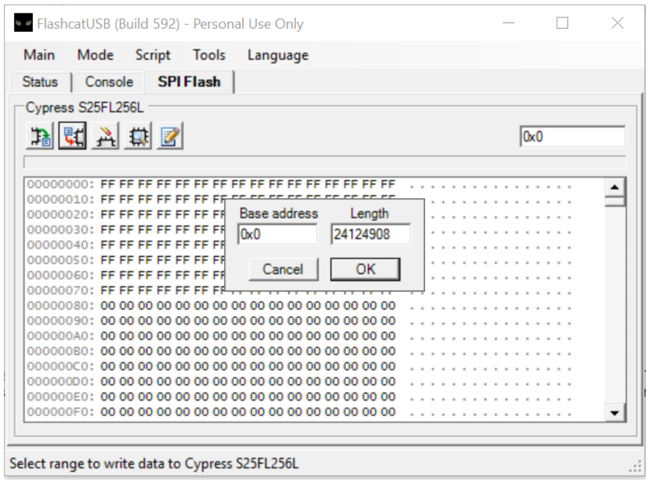


Name	Date modified	Type
Driver	1/27/2020 2:20 PM	File folder
Firmware	1/19/2020 12:39 PM	File folder
JTAG-SVF	2/17/2019 5:16 PM	File folder
Manual	2/10/2020 2:22 AM	File folder
Scripts	1/27/2020 2:28 PM	File folder

File name:
Binary Files (*.bin)
Open
Cancel

4. Select the .bin file in your directory and click the open button.

5. Click OK on the pop-up window if it shows the correct size to proceed

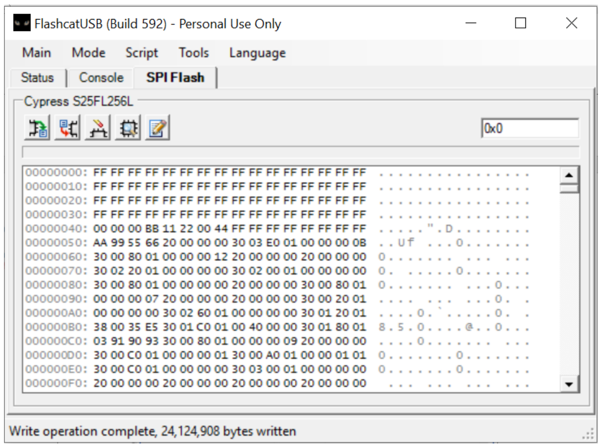


Base address: 0x0
Length: 24124908

```
00000000: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000010: FF FF FF FF FF FF FF                             ................
00000020: FF FF FF FF FF FF FF                             ................
00000030: FF FF FF FF FF FF FF                             ................
00000040: FF FF FF FF FF FF FF                             ................
00000050: FF FF FF FF FF FF FF                             ................
00000060: FF FF FF FF FF FF FF                             ................
00000070: FF FF FF FF FF FF FF                             ................
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Select range to write data to Cypress S25FL256L

6. The software will now begin programming the SPI Flash
7. The Window will update to show if the write operation was successful



```
00000000: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000010: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000030: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
00000040: 00 00 00 BB 11 22 00 44 FF FF FF FF FF FF FF FF  .....".D........
00000050: AA 99 55 66 20 00 00 00 30 03 E0 01 00 00 00 0B  ..Uf ...0.......
00000060: 30 00 80 01 00 00 00 12 20 00 00 00 20 00 00 00  0....... ... ...
00000070: 30 02 20 01 00 00 00 00 30 02 00 01 00 00 00 00  0. .....0.......
00000080: 30 00 80 01 00 00 00 00 20 00 00 00 30 00 80 01  0....... ...0...
00000090: 00 00 00 07 20 00 00 00 20 00 00 00 30 00 20 01  .... ... ...0. .
000000A0: 00 00 00 00 30 02 60 01 00 00 00 00 30 01 20 01  ....0.`.....0. .
000000B0: 38 00 35 E5 30 01 C0 01 00 40 00 00 30 01 80 01  8.5.0....@..0...
000000C0: 03 91 90 93 30 00 80 01 00 00 00 09 20 00 00 00  ....0....... ...
000000D0: 30 00 C0 01 00 00 00 01 30 00 A0 01 00 00 01 01  0.......0.......
000000E0: 30 00 C0 01 00 00 00 00 30 03 00 01 00 00 00 00  0.......0.......
000000F0: 20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00   ... ... ... ...
```

Write operation complete, 24,124,908 bytes written

An image of the FlashCatUSB-MACH programmer board and socket board are shown below.

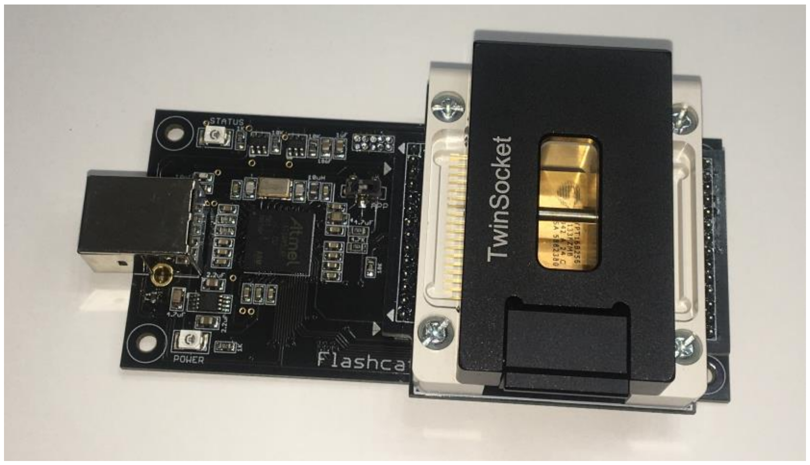


Figure 9. FlashCatUSB-Mach programmer+ socket board

5 Other Considerations

Since the power-up timing and voltage thresholds are different for both FPGAs and SPI Flash devices, it is important to carefully review the differences to insure compatibility between devices on power-up.

5.1 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_0 - to reach their individual power-on thresholds before starting the configuration process. In many applications, VCCO_0, 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_0 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.

After the three FPGA voltages reach their Power On Reset thresholds, the FPGA starts its configuration process:

- Clears its internal configuration
- De-asserts INIT_B, and selects SPI Flash
- Sends the appropriate read command to start configuration bitstream from SPI Flash

After Cypress 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 the CYRS16B256 and CRYS16B512 t_{pu} is ~300us.

The power-on time difference is considerable between the FPGA and SPI Flash devices and although the FPGA power up time is in the magnitude of milliseconds compared to microseconds for the SPI Flash, this should still be considered since these devices could be connected to different power supplies, which requires a circuit solution to guarantee power-on compatibility between FPGA and SPI Flash. The following solution can be used to protect against accessing the SPI Flash before it has completed t_{pu}:

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 10, there are many available devices on the market from companies such as TI and Analog Devices.

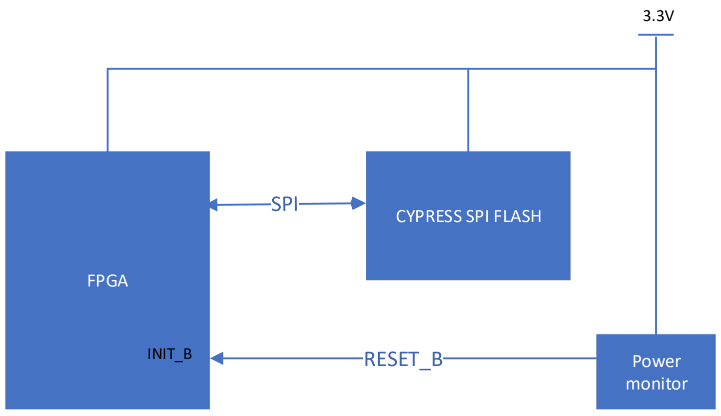


Figure 10. Power On Reset using power monitor supervisor

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 10.

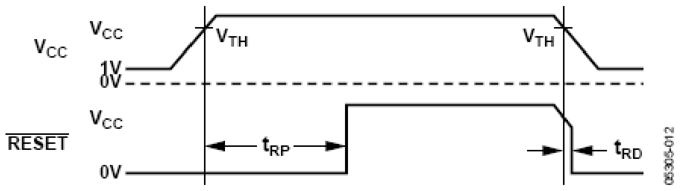


Figure 11. Power On Reset using power monitor supervisor

6 Summary

This application note provided the various options available for connecting and configuring the Cypress Radiation tolerant SPI Flash. More resources and FPGA specific application notes related to programming and configuration can be found on the Xilinx website. Other Cypress Radiation tolerant SPI Flash resources and information can be found on the cypress website.

6.1 References

1. Xilinx UltraScale Architecture Configuration User Guide-UG570
2. Embedded computers-FlashCatUSB-Mach User Guide
3. SPI Configuration and Flash Programming in UltraScale FPGAs-XAPP1233

Document History

Document Title: APPLICATION NOTE - Cypress® Radiation Tolerant SPI Flash configuration options for Xilinx® FPGA's
Document Number: 002-30400

Revision	ECN	Submission Date	Description of Change
**			New Application Note.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Code Examples | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709