PmodSF™ Demo Project

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Revision: February 16, 2012

Overview

This document describes a VHDL demo project that interfaces with a PmodSF. The project demonstrates data transfer between an FPGA board and an SPI flash ROM. The project was developed in Xilinx ISE Design Suite 13.2 based on a Nexys3 board, but it can be retargeted to any other FPGA/board easily.

Digilent Adept Suite is used to program the demo project onto the FPGA, and to transfer SPI ROM read/write data between the Nexys3 board and the PC. Please refer to the Reference Manuals listed below for more detailed information.

References

Digilent Nexys3 Reference Manual and Schematic
Digilent PmodSF Reference Manual and Schematic
Digilent Adept Reference Manual
Digilent Parallel Interface Model Reference Manual
ST Micro M25P16/ M25P128 Data Sheet

Description

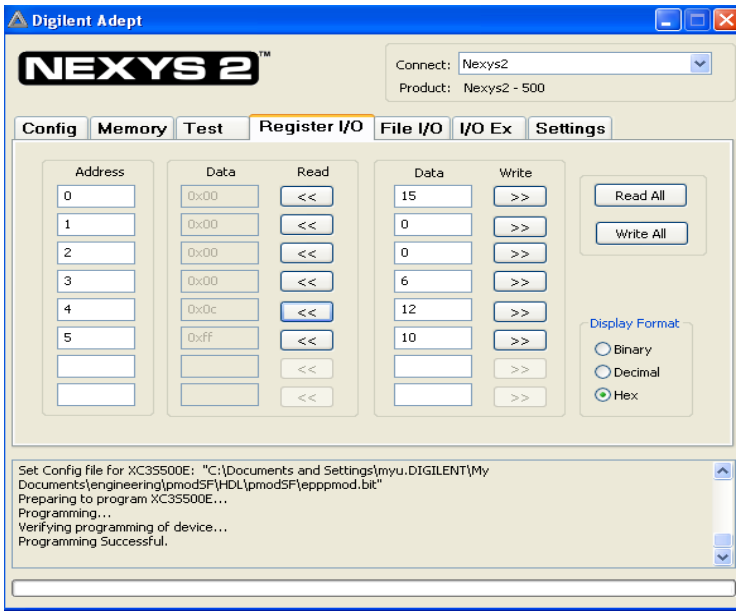
This demo project is composed of two major blocks: a USB interface that implements registers in the FPGA for Adept to read and write; and an SPI controller that can read and write data from the flash ROM on the PmodSF. A simple 8-bit control bus, the Digilent parallel interface bus (based on the Enhanced Parallel Port, or EPP, specification), is used to transfer data internally between these two blocks.

The USB interface block communicates with firmware in the USB controller to implement six 8-bit registers in the FPGA. Registers can be read from the Register I/O part of the Digilent's freely available Adept software. Three 8-bit registers are used to hold the 24-bit ROM address, one register holds the read/write data, and two are used for control. Register data is communicated between the USB interface block and the SPI controller using Digilent's parallel interface bus. Bus timings and signal definitions closely follow the EPP specification; please see Digilent Parallel Interface Model Reference Manual at www.digilentinc.com for a detailed description of bus timing and control.

The SPI ROM must be placed in a write enable mode before write or erase data. Register 3 is used to pass mode control information directly to the SPI ROM from the Transport interface (please see the ST Micro M25P16/ M25P128 data sheet for information on operating modes). Registers 0, 1, and 2 are used to hold the LSB, middle byte, and MSB (respectively) of the ROM address and register 4 is used to buffer the read or write data. Register 5 is used to send an erase command to the Flash ROM, the data sent to register 5 is ignored; writing anything to register 5 will start an erase operation.

When the ROM is in write mode, writing a data value to register 4 will cause that byte to be sent to the ROM for programming at the address specified in registers 0, 1, and 2. Writing any values to register 5 will start a ROM erase cycle using the address specified in the three address registers. Reading register 4 will cause a byte to be read from the ROM at the address specified in registers 0, 1, and 2, and returned.

As an example, the screen shot shows a ROM address of 0x00000F, and a mode control register value of 6 (which sets the ROM to write/erase mode). After clicking the write button for register 3 to set the mode, clicking the write button for register 4 will write the value 12 to address 0x00000F.



To use the reference design “as is”, simply attach the PmodSF to the bottom row of JA port on the Nexys3 board, program the FPGA on the Nexys3 board with the bit file created from the downloaded source files, and then run Digilent’s Adept software. Select the Register I/O tab, and enter register values as desired.

The following table summarizes the address definitions used in the reference design.

Address	Data	Comments
0	Addr 7:0	Least significant byte of flash ROM address
1	Addr 15:8	Middle byte of flash ROM address
2	Addr 23:16	Most significant byte of flash ROM address
3	Mode	Sets operating mode of Flash ROM (refer to ROM data sheet)
4	Data	Write: send data to Flash ROM; Read: acquire data from Flash ROM
5	Erase	Write any value to start erase at addressed specified in registers 0, 1, & 2