MachXO2 Programming via WISHBONE Demo

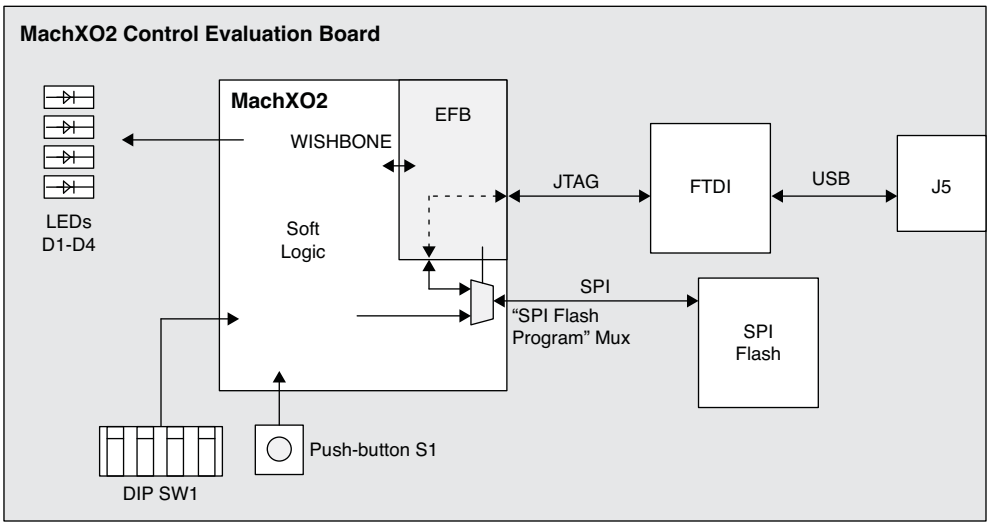
User's Guide

April 2012
UG57_01.0

Introduction

This document provides technical information and instructions for using the MachXO2™ Programming via WISHBONE demonstration design. This demo demonstrates the feasibility of self-reconfiguration: A user design operating in the programmable fabric altering and enabling the contents of Configuration Flash Memory via the EFB WISHBONE interface. This document provides a circuit description of the demo logic as well as instructions for running the demo on a MachXO2 Control Evaluation Board. A high-level block diagram of the Programming via WISHBONE Demonstration Design is shown in Figure 1.

Figure 1. Board-Level Block Diagram



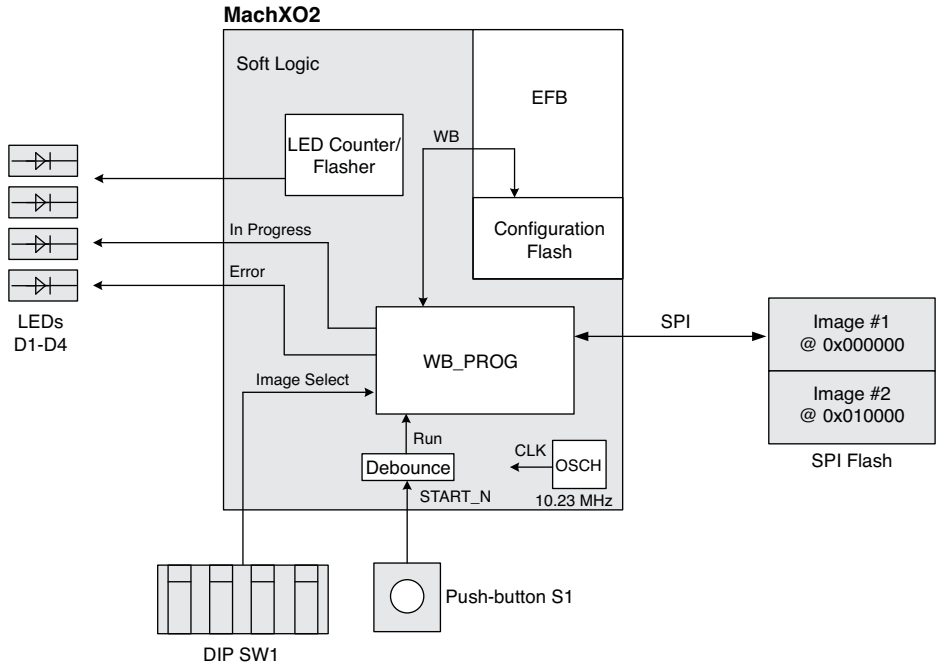
This design demonstrates the ability for the MachXO2 WISHBONE interface to access its internal Flash memory and configuration logic. Data is read from an on-board SPI Flash and processed into commands to program the internal Flash transparently. A state machine controls the generation of commands for each of the phases. After successful programming, the refresh command is issued, and the new pattern is loaded and begins operation.

The demonstration makes use of two external images. When loaded into the FPGA, each separate image displays a unique flashing sequence on the MachXO2 Control Evaluation Board LED field. In addition to indicating to the user that the selected image successfully configured the device, it also demonstrates that the source of the FPGA image or images can be arbitrarily located in the external SPI memory.

The demonstration also shows a user's ability to recover the Hardened SPI port pins for GPIO. In programming initial contents of the SPI Flash, the Lattice Diamond® Programmer tool (or ispVM™ System software) takes advantage of the fact that the external SPI Flash device is connected to the Hardened SPI port. However, the demonstration design itself accesses the external SPI Flash through a soft SPI core. The demonstration design disables the hardened Slave SPI port and the soft core uses the recovered GPIO for its SPI port pins. Note: The external SPI Flash is used simply as a storage device from which data can be read. The design does not demonstrate the dual-boot or external boot capabilities of the MachXO2 device. A block diagram of the FPGA design is shown in Figure 2.

The principles of the design can be extended to make use of virtually any external memory source. The design could also be adapted to become a bridge between an external master controller using a standard or proprietary data link and the configuration Flash.

Figure 2. MachXO2 Block Diagram



The demo package includes the following:

- Verilog source code for two demo logic designs
- Embedded Function Block (EFB) configuration files (.lpc, .ipx)
- Lattice Diamond implementation project files (.ldf) along with the preference files (.lpf) for each demo project
- Aldec Active-HDL simulation scripts (.do) and Verilog test bench.
- JEDEC (.jed) and bitstream (.bit) files for each design
- A readme.txt file with complete file listing and design compilation instructions.

Demo design hardware requirements:

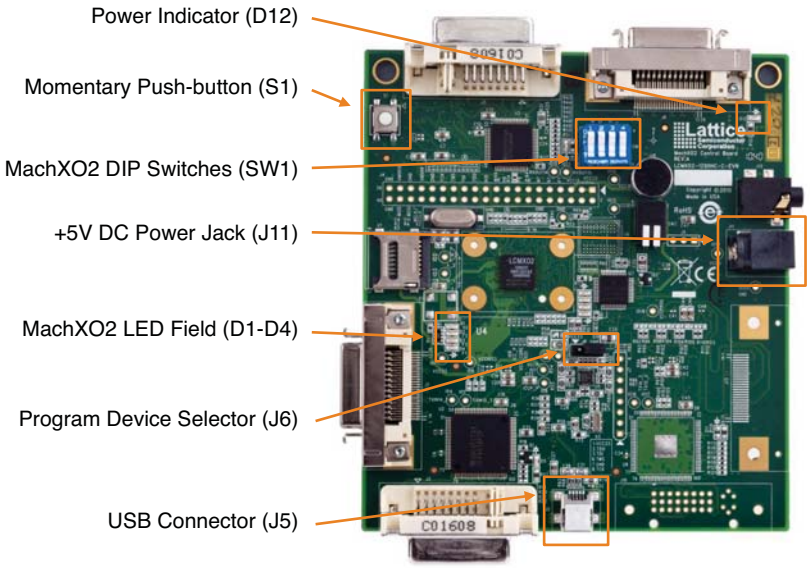
- MachXO2 Control Evaluation Board Revision E with a LCMXO2-1200HC CPLD, 132-ball csBGA package
- 5V DC power supply for the MachXO2 Control Evaluation Board
- Windows PC or Linux machine for implementing the demo project and downloading the bitstream
- JTAG download cable

Demo design software requirements:

- Lattice Diamond design software version 2.0 (or later). Version 2.0 is available June, 2012.
- (Optional) ispVM System programming software version 18.0.1 (or later)

A diagram of the relevant features of the MachXO2 Control Evaluation Board is shown in Figure 3.

Figure 3. MachXO2 Control Evaluation Board, Top Side



Setting up the Demo

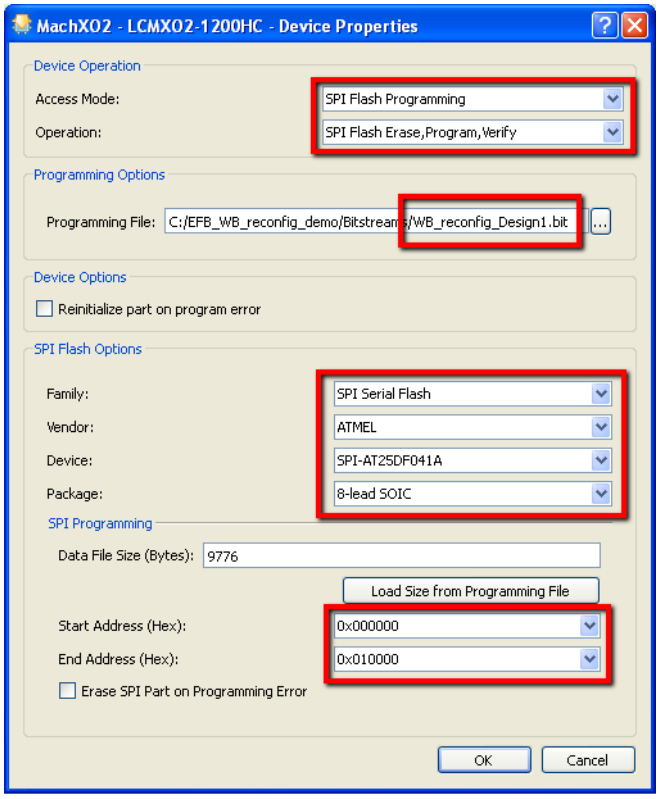
Prior to running the demo, perform the following steps to pre-load the external SPI device with two FPGA images, and then to pre-program one of the images into the MachXO2 device.

- Apply +5V DC power to the board using J11. An AC-DC convertor is supplied with the MachXO2 Control Evaluation Board for this purpose. The Power Indicator (D12) will illuminate.
- Verify the Program Device Selector (J6) shunt is placed across pins 2-3.
- Connect the MachXO2 Control Evaluation Board to a PC using a standard mini-USB cable.
- Using ispVM 18.0.1 or later, or Diamond Programmer 2.0 or later, program the two FPGA .bit images into the external SPI Flash device. Follow the steps shown in Figures 4-9 for Diamond Programmer, or in Figures 10-16 if using ispVM System software.

Program Image #1 into SPI Flash Using Diamond Programmer

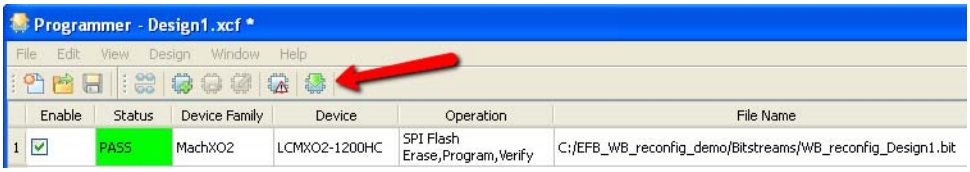
After launching Diamond Programmer, chose **Edit > Device Properties**. Enter the information as shown in Figure 4, pointing to the actual location of WB_reconfig_Design1.bit on your file system.

Figure 4. Setting Device Properties for Image #1



Click **OK** to close the Device Properties dialog box. Then select **Design > Program** or click the icon shown in Figure 5 to program Image #1 into the external SPI Flash at location 0x000000.

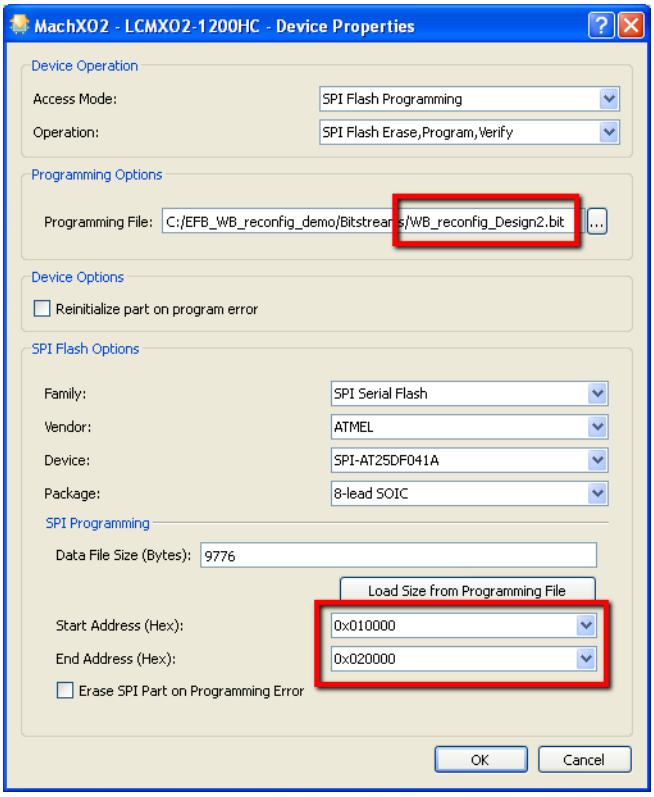
Figure 5. Program Image #1 into External SPI Flash



Program Image #2 into SPI Flash Using Diamond Programmer

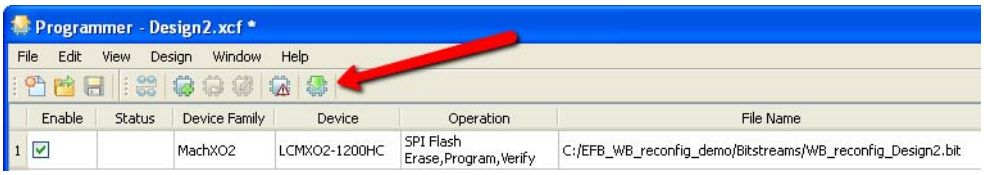
To program the second image into the external SPI Flash, chose **Edit > Device Properties** again. As shown in Figure 6, modify the file path name to point to the actual location of WB_reconfig_Design2.bit on your file system, and change the Start Address to 0x010000.

Figure 6. Setting Device Properties for Image #2



Click **OK** to close the Device Properties dialog box. Then select **Design > Program** or click the icon shown in Figure 7 to program Image #2 into the external SPI Flash at location 0x010000.

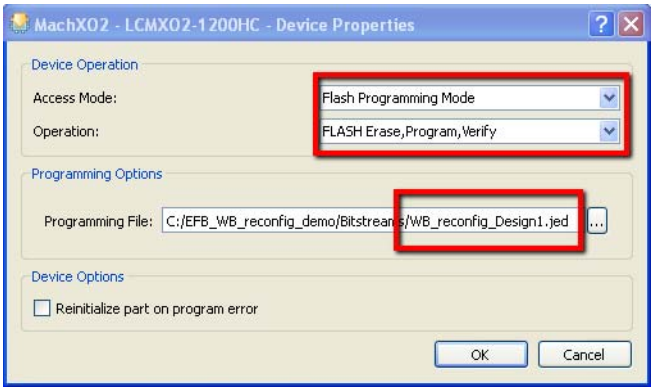
Figure 7. Program Image #2 into External SPI Flash



Program Image #1 into MachXO2 Using Diamond Programmer

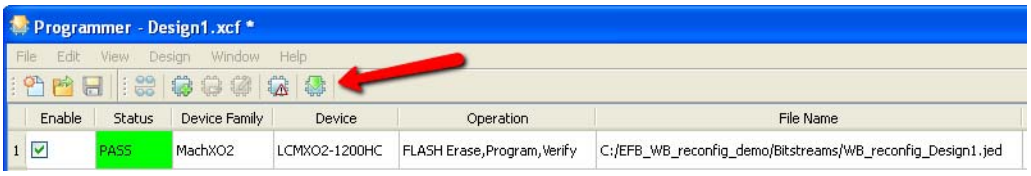
To pre-load Image #1 directly into the MachXO2 device, chose **Edit > Device Properties** one last time. Change the Access Mode and Operation as shown in Figure 8, and point to the actual location of WB_reconfig_Design1.jed on your file system.

Figure 8. Set Flash Programming Mode, Select Source .jed file and Address Location for Image #1



Click **OK** to close the Device Properties dialog box. Then select **Design > Program** from the menus or click the icon shown in Figure 9 to program Image #1 into the MachXO2 internal configuration Flash.

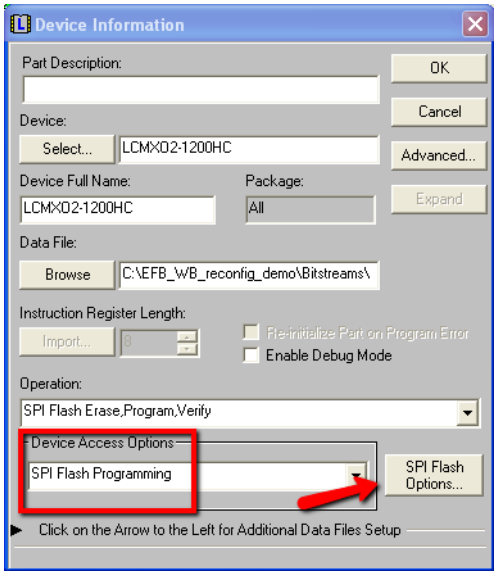
Figure 9. Program Image #1 into MachXO2 Configuration Flash



Program Image #1 into SPI Flash Using ispVM System Software

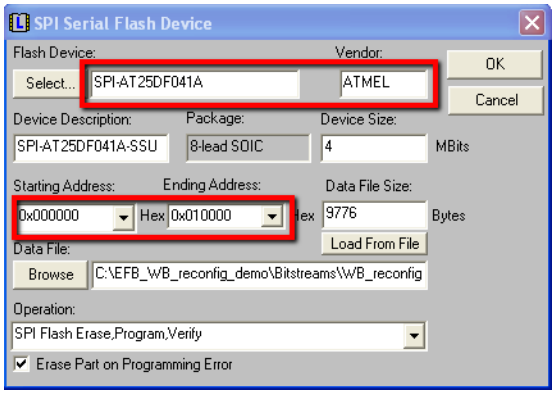
After launching ispVM, chose **ispTools > Scan Chain** to auto-detect the LCMXO2-1200HC device. Select **Edit > Edit Device** to bring up the Device Information dialog box. Select **SPI Flash Programming** in the Device Access Options pull-down, as shown in Figure 10, and click the **SPI Flash Options** button.

Figure 10. Device Options Dialog Box



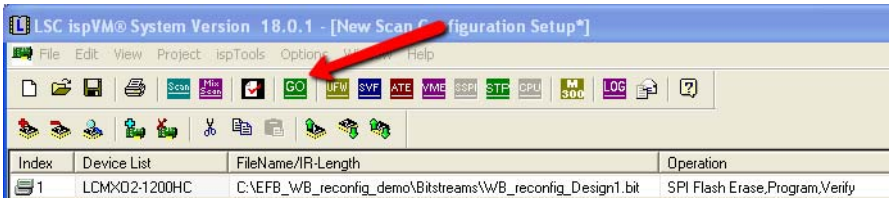
Set the Flash Device, Starting Address, and Data File ("WB_reconfig_Design1.bit") in the SPI Serial Flash Device dialog box, as shown in Figure 11.

Figure 11. Set SPI Serial Flash Device Properties for Image #1



Click **OK** twice to close the dialog boxes. Then select **Project > Download** or click the **GO** icon as shown in Figure 12 to program Image #1 into the external SPI Flash at location 0x000000.

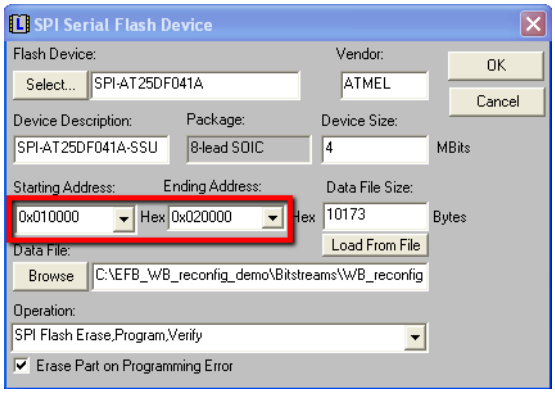
Figure 12. Program Image #1 into External SPI Flash



Program Image #2 into SPI Flash Using ispVM System Software

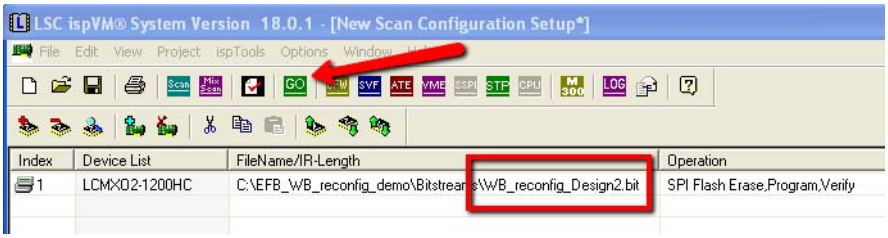
Select **Edit > Edit Device** to bring up the Device Information dialog box, then click the **SPI Flash Options** button to bring up the SPI Serial Flash Device dialog box again. As shown in Figure 13, modify the file path name to point to the actual location of WB_reconfig_Design2.bit on your file system, and change the Start Address to **0x010000**.

Figure 13. Select Target Address and .bit File for Image #2



Click **OK** twice to close the dialog boxes. Then select **Project > Download** or click the **GO** icon as shown in Figure 14 to program Image #2 into the external SPI Flash at location 0x010000.

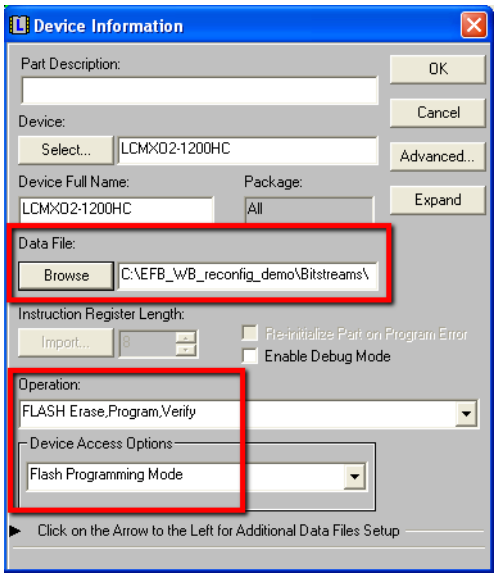
Figure 14. Program Image #2 into External SPI Flash



Program Image #1 into MachXO2 Using ispVM System Software

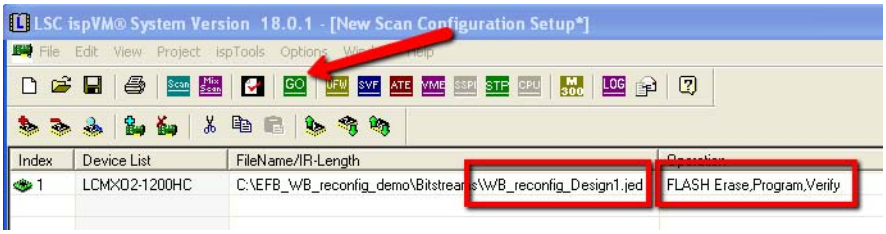
To pre-load Image #1 directly into the MachXO2 device, select **Edit > Edit Device** to again bring up the Device Information dialog box. Select **Flash Program Mode** in the Device Access Options pull-down. Verify the Operation is set to **FLASH Erase, Program, Verify**. Click **Browse** to point to the location of WB_reconfig_Design1.jed on your file system, as shown in Figure 15.

Figure 15. Device Information Dialog for MachXO2 Programming



Click **OK**, then select **Project > Download** from the menus or click the **GO** icon as shown in Figure 16 to program Image #1 into the MachXO2 internal configuration Flash.

Figure 16. Select Flash Erase, Verify, Program Mode, .jed File for Image #1 and Execute



Running the Demo

Once prepared, the demo may be run stand-alone, without the need for an external computer.

Controls and Indicators:

- MachXO2 DIP switches:
  - **1**: PROGRAMN: Toggle on-off to assert the MachXO2 device PROGRAMN pin. When released, the MachXO2 device will boot from its internal Configuration Flash memory.
  - **2**: Controls the MachXO2 asynchronous reset. Toggle on-off to reset the device's internal logic.
  - **3**: (unused)
  - **4**: Image select
    - 'Off' selects image #1 (external SPI address 0x000000)
    - 'On' selects image #2 (external SPI address 0x010000)

- MachXO2 LED Field:
  - **D1**: Configuration Error. Illuminated if the MachXO2 image copy does not verify with the external image. The error remains until device reset.
  - **D2, 3**: Image ID:
    - Image #1: Three short flashes followed by an off period
    - Image #2: One long flash followed by an off period
  - **D4**: Status: Illuminated while Program/Verify is in progress
- Momentary Pushbutton – Executes the Demonstration Design State Machine
  - Retrieves configuration data from external SPI Flash
  - Writes configuration data into internal Configuration Flash array
  - Verifies contents of Configuration Flash array
  - Issues REFRESH to transfer contents of the Configuration Flash to the device fabric SRAM

The procedure for executing the MachXO2 Programming via WISHBONE demo is as follows:

1. Apply +5V DC power to the board using J11. An AC-DC convertor is supplied with the MachXO2 Control Evaluation Board for this purpose. The power indicator (D12) will illuminate.
2. Once power is stable, note the flashing pattern from the board's LED field.
3. Select the desired source image using DIP switch 4.
4. Press and release the Momentary push-button. D4 will light for a few seconds indicating "Programming in Progress", then D2 and D3 will assume new behavior defined by the image.
5. PROGRAMN can be asserted (DIP switch 1) to verify that the new image is programmed into internal Configuration Flash.
6. If desired, cycle the power to show the newly programmed image is non-volatile.

Theory of Operation

The WB_prog module implements two master circuits:

1. A SPI master circuit (spi_flash_intf.v) to retrieve data from the external SPI Flash device
2. A WISHBONE master circuit (wb_prog_ssm.v) to pass data to/from the internal Configuration Flash memory. The Embedded Function Block is instantiated within WB_prog. A FIFO is used for buffering data and for serial-to-parallel conversion (data_fifo.ipx).

The controlling state machine is contained in wb_prog_ssm.v. A diagram of the state machine operation is shown in Figure 17. The state machine executes these major steps once the Start input is recognized:

1. Configuration Access Enabled
2. Erase Configuration Flash
3. Program Configuration Flash with data from SPI Flash
4. Verify Configuration Flash contents
5. REFRESH if verify successful, else assert error and stop

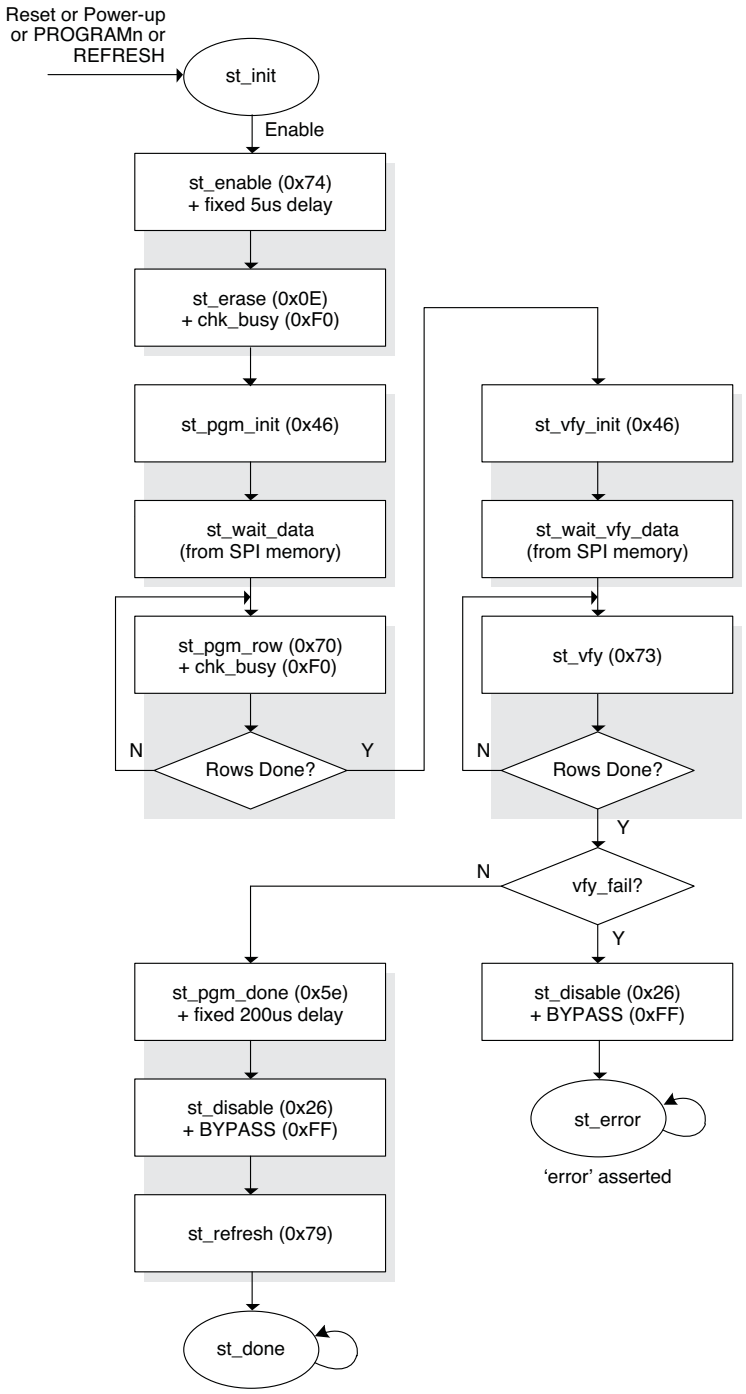
All clocking is provided by the internal user oscillator (currently set to the 10.23MHz setting). All logic uses the same clock, with clock gating for the SPI Flash clock output when pauses are required. Clock speed can be increased, but the following should be noted:

- Delay values in wb_cfg.v need to be adjusted to maintain the same time references. The most critical are DELAY_ISCEN and DELAY_DONE, which are used as fixed delays (no polling).

- The SPI Flash interface logic uses slow SPI reads, which have maximum values that are vendor and device specific. Utilizing a fast read mode would require logic changes to that module.

To allow reading ahead and controlling synchronization, an EBR-based FIFO is used. The FIFO writes data from the SPI Flash one bit per cycle and supplies data eight bits at a time to the WISHBONE interface. The SPI Flash writes to the FIFO, pausing when the FULL flag is asserted. On the read side, an almost empty flag is used to indicate if sufficient data is available for one frame of data prior to programming or verifying.

Figure 17. Flow Diagram



Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2012	01.0	Initial release.

Appendix A. Internal Configuration Commands

Table 1 lists the internal configuration commands involved for the various operations of this demo.

Table 1. Internal Configuration Commands

Operation	Command (Hex)	Operand (Hex)	Data	Return Value
Enable Configuration Access (Transparent)	74	08 00 00	—	—
Erase UFM	0E	08 00 00	—	—
Status check	F0	00 00 00	—	1 byte status data
Init Address	46	00 00 00	—	—
Program Config Data	70	00 00 00	16 bytes configuration data	—
Read Config Data	73	10 00 00	—	16 bytes configuration data
Program DONE	5E	00 00 00	—	—
Disable Configuration Access	26	00 00	—	—
Bypass	FF	—	—	—
REFRESH	79	00 00 00	—	—