Contents
1. Introduction ................................................................................................................................................. 4
   1.1. Demo Design Overview .......................................................................................................................... 4
   1.2. MachXO3-9400 Development Board and Resources .............................................................................. 5
2. Functional Description ............................................................................................................................... 6
   2.1. General .................................................................................................................................................. 6
   2.2. User Design Block .................................................................................................................................. 8
   2.3. Multiplexer Latch to Hold Outputs ........................................................................................................ 8
   2.4. Message Control Block ......................................................................................................................... 8
3. Demo Package ............................................................................................................................................... 10
   3.1. Hardware Requirements ........................................................................................................................ 10
   3.2. Software Requirements ........................................................................................................................ 10
4. Port Assignments and Descriptions ......................................................................................................... 11
5. Demo Package Directory Structure ........................................................................................................ 11
6. Running the Demo ....................................................................................................................................... 12
   6.1. Programming the Device ....................................................................................................................... 12
   6.2. Updating the Flash Image ..................................................................................................................... 15
   6.3. Preparing for Update ............................................................................................................................. 17
   6.4. Updating the Design ............................................................................................................................ 18
   6.5. Resuming Normal Operation .............................................................................................................. 21
7. Rebuilding the Design ............................................................................................................................. 22
References ......................................................................................................................................................... 25
Technical Support .......................................................................................................................................... 25
Revision History ............................................................................................................................................. 26
Figures
Figure 1.1 Demo Resources ................................................................. 5
Figure 2.1. Hitless I/O Demo Design Block Diagram ................................................................. 6
Figure 2.2. Hitless I/O Demo Timing Diagram ................................................................. 7
Figure 2.3. TransFR I/O Cell Simplified Block Diagram ...................................................... 8
Figure 2.4. Hitless I/O New Top Level Block Diagram ...................................................... 9
Figure 5.1. Demo Package Directory Structure ............................................................. 11
Figure 6.1. Diamond Programmer Getting Started Page .................................................. 12
Figure 6.2. Diamond Programmer Main Interface .................................................. 12
Figure 6.3. Device Properties Option ............................................................. 13
Figure 6.4. Browse Programming File ............................................................. 13
Figure 6.5. Programming the Device ............................................................. 14
Figure 6.6. Device Properties Option ............................................................. 15
Figure 6.7. Access Mode Options ............................................................. 15
Figure 6.8. Operation Options ............................................................. 16
Figure 6.9. Browse Programming File ............................................................. 16
Figure 6.10. Programming the Device .......................................................... 17
Figure 6.11. Preparations for the Demonstration .................................................. 18
Figure 6.12. Edit I/O State Option ............................................................. 18
Figure 6.13. Edit I/O State Dialog Box ............................................................. 19
Figure 6.14. Device Properties Option ............................................................. 19
Figure 6.15. Access Mode Options ............................................................. 20
Figure 6.16. Operation Options ............................................................. 20
Figure 6.17. Programming the Device .......................................................... 21
Figure 7.1. Opening Design File ............................................................. 22
Figure 7.2. Setting original_up_count_impl1 Active .................................................. 22
Figure 7.3. Enable Transfer ................................................................. 23
Figure 7.4. Generating the Bitstream .......................................................... 24

Tables
Table 2.1. Truth Table for Normal_Operation, Hitless_Enable .................................................. 7
Table 4.1. FPGA Demo Design Ports ........................................................................ 11
1. Introduction

In mission critical systems such as data centers, storage and networking, feature improvements and bug fixes are performed through background updates. Designers, however, avoid updating the control programmable logic device (PLD) because it forces a power cycle or reset of the entire system to enable the new algorithm to take effect. The PLD typically performs power management, reset management, glue logic and other housekeeping functions on the board that should not be interrupted.

The Lattice Semiconductor MachXO3™ control PLD offers designers the option of performing updates in the background and enabling new algorithms to take effect without interrupting board-level operations by eliminating the need to power-cycle or reset the device. The LCMXO3LF-9400C Hitless I/O Demo showcases this feature known as hitless or zero-downtime system update. The Lattice Diamond® software is used in this demo, enabling key features such as TransFR and Leave Alone.

This demo user guide describes the MachXO3 hitless I/O technology and provides the details of the design.

1.1. Demo Design Overview

System-critical functions are controlled and supervised by CPLD or FPGA logic. It can be time consuming, inconvenient and costly to remove these systems from service to perform updates to the reprogrammable logic. It is advantageous to update system control logic and state machines in the background without impacting dependent downstream circuits such as power supplies or alarms. The MachXO3 hitless I/O feature is specifically designed to hold critical signal output states stable while the underlying logic is updated. The output states that are put on hold do not glitch during the update process. These can optionally be used to preset the logic state machines to resume control at any desired operating point upon resumption of normal operation.

The hitless I/O design uses silicon features built into the MachXO3 FPGA in concert with minimal additions to the user design logic. MachXO3 supports background reprogramming and TransFR mode reconfiguration. A simple multiplexer latch circuit is added to the critical user outputs and a simple common Message Control block (MCB) is used for interfacing with the System Update Controller. The System Update Controller communicates with the Message Control block over a suitable channel, for example signal wire or I2C, to coordinate the update event.

The hitless I/O demo consists of five parts:

- Normal operation — Loading the ‘original’ design. The output LEDs are under the ‘original’ design control.
- Updating the Flash image — Programming the ‘new’ design into Configuration Flash in the background.
- Preparing for update — Communicating to the device regarding a design update using an external switch. The binary counter value is frozen.
- Updating the design — Reconfiguring the configuration SRAM with the new bitstream image contained in the configuration Flash array using the TransFR and Leave Alone features. LED outputs are held stable without glitching.
- Resuming normal operation — The updated ‘new’ design is active with the outputs driven by the updated design.
1.2. MachXO3-9400 Development Board and Resources

The onboard buttons, DIP Switches and LEDs of the MachXO3LF-9400 Development Board are used to demonstrate the hitless I/O feature of the MachXO3 device. Figure 1.1 shows the top side of the MachXO3-9400 Development Board and resources used for the demo.

![Figure 1.1 Demo Resources](image-url)
2. Functional Description

2.1. General

The following is a general description of the hitless I/O feature operation. Figure 2.1 shows the demo design block diagram and Figure 2.2 shows the timing diagram.

![Hitless Feature Diagram](image)

**Figure 2.1. Hitless I/O Demo Design Block Diagram**

A Normal_operation signal is used to support the hitless I/O circuit. Normal_operation must have a default global reset signal GSR value of False. Following the release of the GSR, Normal_operation is typically asserted by the user logic to indicate that the User Design circuit is free to operate. Normal_operation typically remains asserted until the start of the hitless I/O operation. A message from an external controller is used to deassert Normal_operation, typically after the new bitstream is programmed into Flash memory and just prior to a sysConfig REFRESH command or PROGRAMN pin toggle.

The User Design outputs are captured and held with the soft 2:1 multiplexer latch. Normal_operation then remains deasserted until the MachXO3 device begins reconfiguration. As reconfiguration begins, the TransFR circuit in the Boundary Scan logic in the I/O cell latches the output value (Phase 2 of the Non-JTAG mode TransFR Sequence. Refer to TN1087 – Minimizing System Interruption During Configuration Using TransFR Technology). The TransFR Boundary Scan logic holds the output states through Phases 3 and 4 until the device wake-up is complete and the device re-enters User Mode. The re-established soft 2:1 multiplexer re-acquires the TransFR cell value during the device wake-up sequence (Phase 4) and holds it until the assertion of Normal_operation.
The hitless I/O logic must cover two primary scenarios when the FPGA enters User Mode: Power-up and Hitless Reconfiguration. The control input Hitless_en is driven by an external controller to determine which scenario is active. If Hitless_en is false, the user logic starts from a power-up reset state. If true, the user logic can utilize the values held in the TransFR Boundary Scan I/O cells to initialize the control logic to resume from the pre-update state. See Table 2.1 for the truth table showing both the general behavior of the feature and the specific behavior of the demonstration design.

Table 2.1. Truth Table for Normal_Operation, Hitless_Enable

<table>
<thead>
<tr>
<th>Hitless_enable</th>
<th>Normal_operation</th>
<th>User Design Output</th>
<th>Demo Design Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>Normal Operation</td>
<td>Normal Counter Operation</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>User Defined Power-up State</td>
<td>‘0’ (LED’s ON)*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>User Defined Hold/Preload State</td>
<td>Feedback Value</td>
</tr>
</tbody>
</table>

*Note: All LED cathodes are driven by the design output. As such, LED ON represents ‘0’ and LED OFF represents ‘1’.

As shown in Figure 2.3, the hitless top level design is divided into three major blocks:

- User design block
- Multiplexer latch to hold outputs
- Message control block
For detailed pin/port descriptions, see the Port Assignments and Descriptions section.

2.2. User Design Block

The User Design block contains the original user logic, in this case a simple 8-bit up-counter design. The original user logic is updated to include two accommodations which support the hitless I/O process:

- Enable input
- Addition of a multiplexer latch to select the asynchronous reset Preset/Restore value.

In the demonstration, this block is updated to become a down-counter. The reset_n logic is placed on the GSR net by the software.

2.3. Multiplexer Latch to Hold Outputs

The hitless I/O design also features multiplexer latch(es) and a few basic logic cells added to the top level to hold the outputs stable. A 2:1 multiplexer latch is instantiated for each original output signal which is to become hitless, and the output is changed into a bidirectional I/O. A common combinatorial logic path is instantiated for the Hitless_en control signal. The path must be 100% combinatorial so that the 2:1 muxes are controlled by the state of Hitless_en prior to the release of GSR.

2.4. Message Control Block

This hitless I/O design addition is used to communicate messages to the hitless top level design that the transFR operation is about to occur. SW4 represents the 'external controller' as described in the General section. The message is conveyed by closing SW4, a momentary switch on the MachXO3-9400 Development Board. The closing edge (falling edge) of Hold_output is detected and used to negate Normal_operation until the device is refreshed.

Following the FPGA image update, Normal_operation is re-asserted after a small delay, providing time for the circuit to synchronize to the previous counter state as preserved in the LED_count I/O cells.
A second, nearly identical design project is included as part of the hitless I/O demonstration. As shown in Figure 2.4, only the user design block is changed to include a Down Counter block in place of the prior Up Counter. All the other blocks remain untouched.

The user will be able to switch between the designs without causing glitches or state changes to the LED outputs.

Figure 2.4. Hitless I/O New Top Level Block Diagram
3. Demo Package
The demo package includes the following:
- Lattice Diamond project and preference files
- JED file for programming the MachXO3 internal configuration flash

3.1. Hardware Requirements
To run the demo, the following hardware are required:
- PC running Windows 7 Operating System
- MachXO3-9400 Development Board
- Mini USB cable for programming the MachXO3 device

3.2. Software Requirements
To run the demo, the following software are required:
- Lattice Diamond version 3.9 or later
- Lattice Diamond Programmer software for bitstream downloading

Note: These software programs are available at www.latticesemi.com/en/Products/DesignSoftwareAndIP.
4. Port Assignments and Descriptions

Table 4.1. FPGA Demo Design Ports

<table>
<thead>
<tr>
<th>Port Name</th>
<th>I/O Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset_n</td>
<td>Input</td>
<td>1</td>
<td>Asynchronous reset, active low (Button SW2)</td>
</tr>
<tr>
<td>Clk</td>
<td>Input</td>
<td>1</td>
<td>Input clock from crystal</td>
</tr>
<tr>
<td>Hitless_en</td>
<td>Input</td>
<td>1</td>
<td>Toggle switch (SW1 BIT1) to differentiate between power on and hitless I/O operation. Down (0) – Power on, Up (1) – Hitless I/O</td>
</tr>
<tr>
<td>hold_output</td>
<td>Input</td>
<td>1</td>
<td>Push-button switch (SW4) to hold the output state</td>
</tr>
<tr>
<td>LED_count</td>
<td>Output</td>
<td>8</td>
<td>LEDs indicating the output values</td>
</tr>
</tbody>
</table>

5. Demo Package Directory Structure

Figure 5.1 shows the demo package directory structure.
6. Running the Demo

6.1. Programming the Device

To program the device:

Loading the initial design. The onboard output LEDs start counting upwards.

1. Launch the Diamond Programmer Software (version 3.9 or above). In the Getting Started dialog box as shown in Figure 6.1, select **Create a new project file from JTAG scan** and click **OK**.

![Figure 6.1. Diamond Programmer Getting Started Page](image)

2. The Diamond Programmer starts scanning the board attached to the USB cable.

![Figure 6.2. Diamond Programmer Main Interface](image)
3. Click the device row to highlight it and select **Edit > Device Properties** from the menu bar. This allows you to edit the access mode, operation mode and select the programming file.

![Device Properties Option](image)

**Figure 6.3. Device Properties Option**

4. Set device properties as shown in **Figure 6.4**.

   - In the Device Properties dialog box, select **Flash Programming Mode** in Access mode and **Flash Erase, Program, Verify** in Operation. Select the bitstream file `\bitstream\MachXO3LF_Hitless_IO_demo_original_up_count_impl1.jed` in Programming file. Click **OK**.

![Browse Programming File](image)

**Figure 6.4. Browse Programming File**
The next step is to program the device.

5. From the menu bar, select **Design > Program**.

![Programming the Device](image.png)

**Figure 6.5. Programming the Device**

When the programming of the device is completed, the LEDs on the MachXO3LF 9400 Development Board start counting up from zero (LED OFF represents 1; LED ON represents 0).
6.2. Updating the Flash Image

To update the flash image:

1. Click the device row to highlight it and from the menu bar, select Edit > Device Properties. This allows you to edit the access mode as well as operation.

2. From the Access Mode drop-down list select Flash Background Mode as shown in Figure 6.7.

![Device Properties Option](image1)

![Access Mode Options](image2)
3. From the Operation drop-down list select **XFLASH Erase, Program, Verify**.

![Figure 6.8. Operation Options](image)

4. Select the bitstream file \bitstream\MachXO3LF_Hitless_IO_demo_new_down_cnt_impl2.jed in Programming file. Click **OK**.

![Figure 6.9. Browse Programming File](image)
The next step is to program this bitstream into the device.

5. From the menu bar, select **Design > Program**. During the Program operation, the LEDs do not change counting state nor glitch. This can be confirmed using an oscilloscope.

![Figure 6.10. Programming the Device](image)

**6.3. Preparing for Update**

Before the new design is transferred to active SRAM, it is necessary to communicate with the device regarding the imminent update.

In this demo, an external switch is used to communicate with the device regarding the design update. Check that all bits of SW1 are ‘Up’, then press the button SW4 momentarily to freeze the LED binary count output on LED7~0. Refer to **Figure 6.11**.

Note that if SW1 BIT1 is down, the counter resets to the power up state when button SW4 is pressed (LED OFF represents 1; LED ON represents 0). When SW1 BIT1 is up if SW4 is not pressed to freeze the outputs, the outputs may glitch during the update step.
6.4. Updating the Design

To update the design:

1. Highlight the device row by clicking anywhere on the row. From the menu bar select **Edit > Edit I/O State**.

2. In the Edit I/O State dialog box, from the I/O state drop-down list select **Leave Alone** as shown in Figure 6.13. Click **OK** to confirm this selection.
3. Click the device row to highlight it and from the menu bar, select Edit > Device Properties. This allows you to edit the access mode as well as operation.

4. From the Access Mode drop-down list select Flash Background Mode as shown in Figure 6.15.
5. Under Operation, select XFLASH TransFR to transfer the ‘new’ bitstream from Configuration Flash to SRAM using the TransFR feature.

6. From the menu bar, select Design > Program to execute the Flash-to-SRAM transfer.
6.5. Resuming Normal Operation

When TransFR is complete, the new bitstream begins operation seamlessly, starts counting down from the LED count value counted up and preserved in the Preparing for Update section (LED OFF represents 1; LED ON represents 0). Not only has the design been successfully updated without changing or glitching the LED output count value, the new circuit is able to resume counting from the previous operation point by referring to the held output states.
7. Rebuilding the Design

To rebuild the design:

1. Open the Lattice Design file (*.ldf file) by clicking **Open** under Project.

![Figure 7.1. Opening Design File](image)

2. There are two implementations in this project: original_up_count_impl1 and new_down_cnt_impl2. Make sure to set the appropriate implementation to active.

![Figure 7.2. Setting original_up_count_impl1 Active](image)
3. Run the Synthesize phase on the design to make sure there are no errors. When you see the green checkbox beside Synthesize Design, click Spreadsheet View.

4. In the Global Preferences tab, make sure that the ENABLE_TRANSFR feature is enabled. Save any changes to Spreadsheet View using File > Save.

5. After these preferences are set up, run the project to generate the bitstream. Check the checkbox before Bitstream File and JEDEC File in Process View and click Export Files.

6. Select Process > Rerun All from the menu bar to run the entire process.
Figure 7.4. Generating the Bitstream

The rebuilt design can be programmed in the FPGA as described in the Running the Demo section.
References

- DS1047 – MachXO3 Family Data Sheet
- TN1087 – Minimizing System Interruption During Configuration Using TransFR Technology
- TN1279 – MachXO3 Programming and Configuration Usage Guide

Technical Support

For assistance, submit a technical support case at www.latticesemi.com/techsupport.
# Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2017</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>