MachXO3D™ Dual Boot Demo

User Guide

FPGA-UG-02068-1.0

October 2018

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
GUI	Graphic User Interface
NVM	Non-volatile Memory
SPI	Serial Peripheral Interface

1. Introduction

In computing, booting is starting up a computer or computer appliance until it can be used. The process involves a chain of stages. At each stage, a smaller, simpler program loads and then executes the larger and more complicated program of the next stage. When a computer is turned off, its software—including operating systems, application code, and data—remains stored in non-volatile memory (NVM) from which the operating system programs and data can be loaded into the RAM in subsequent booting. Those applications stored in NVM need to be read or updated from time to time.

One of the biggest risks in the field upgrade applications is disruption during the field upgrade process, for example:

- Power disruption
- Data file corruption
- Communications disruption

Even if the system does not require a field upgrade, the pattern corruption can still occur due to the following problems caused by data retention issue of flash devices:

- Read fatigue
- Charge loss

To eliminate the risk completely, Lattice provides a Dual Boot feature that switches to load from the second known good pattern, Golden pattern, when the first applied pattern, Primary pattern, is corrupted. This feature enhances the reliability of a field upgradeable system, as the Golden pattern is less affected by these problems since most of the time it is in a dormant state.

The Lattice Semiconductor MachXO3D™ devices provide a unique Dual Boot feature to optionally boot from two sectors of internal flash, Flash A (that is CFG0) and Flash B (that is CFG1). Combining the legacy feature of Lattice Mach family with Master SPI Configuration Mode from external SPI flash (that is EXT), MachXO3D devices provide more Dual Boot options with fixed orders such as:

CFG0 – CFG1	(Primary is internal Flash A, Golden is internal Flash B)
CFG1 – CFG0	(Primary is internal Flash B, Golden is internal Flash A)
CFG0 – EXT	(Primary is internal Flash A, Golden is external SPI Flash)
CFG1 – EXT	(Primary is internal Flash B, Golden is external SPI Flash)
EXT – CFG0	(Primary is internal external SPI Flash, Golden is internal Flash A)
EXT – CFG1	(Primary is internal external SPI Flash, Golden is internal Flash B)
EXT – EXT	(Primary is start space of external SPI Flash, Golden is other space of external SPI Flash)

The MachXO3D devices feature invisible version tag for both CFG0 and CFG1. After programming an image into CFG0 or CFG1, a new version number larger than the current one in the other of CFG0 or CFG1 is automatically programmed. It means the latter programmed image has the larger version number. This enhanced MachXO3D Dual Boot feature is based on the version number of the image with additional two Ping-pong Dual Boot options:

- Check version and boot from the former version within CFG0 and CFG1
- Check version and boot from the latter version within CFG0 and CFG1

This document shows the Dual Boot features of the MachXO3D devices and how they are enabled.

1.1. Demo Overview

The MachXO3D device is preloaded with two internal images CFG0 and CFG1, and two additional images with different start addresses in the external SPI Flash. Updating feature row enables the different boot up mode by using the Primary image from target physical location, and appoints the back up or Golden image, in case Primary image is disrupted. Figure 1.1 shows the physical locations for the images used in this demo.

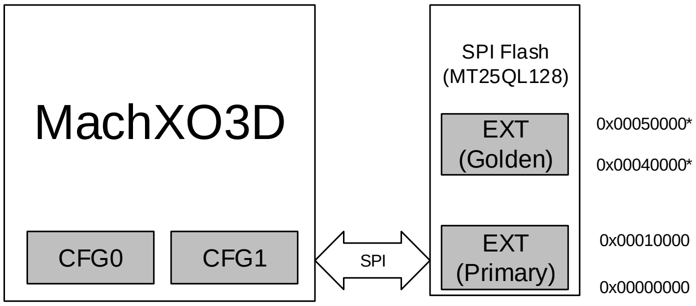


Figure 1.1. Boot Image Locations for the MachXO3D Development Board

Note: The image in EXT (Golden) is only used in the EXT-EXT mode, and its address range is flexible with Deployment Tool.

To demo different modes, the boot images are designed to drive different numbers on Segment LEDs as below:

- Boot from internal Flash A -Segment LED Blinking with "1"
- Boot from internal Flash B -Segment LED Blinking with "2"
- Boot from start space of external SPI Flash -Segment LED Blinking with "3"
- Boot from other space of external SPI Flash -Segment LED Blinking with "4"

This demo covers the following operations:

- Programming the image to CFG0 and CFG1 on the MachXO3D development board
- Preparing external boot image for EXT-EXT Dual Boot mode
- Enabling fixed order Dual Boot, proving the boot from Primary image is under control
- Proving fallback boot from Golden image after failing to boot from Primary image
- Demonstrating Ping-pong update method after checking version and its fall back boot

1.2. MachXO3D Development Board and Resources

Figure 1.2 shows the top side of the MachXO3D Development Board and resources used for this demo.

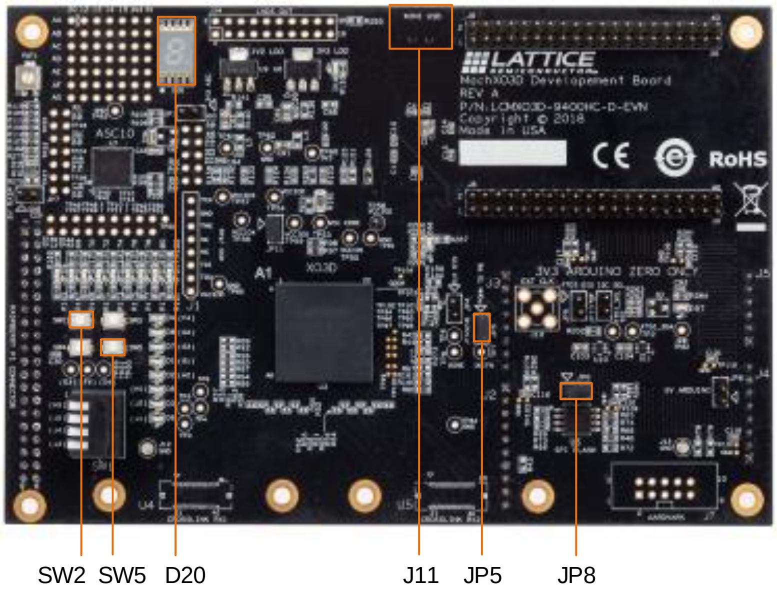


Figure 1.2. MachXO3D Development Board

- **J11** must be connected with Mini-USB cable to provide board power and JTAG access path through USB.
- **JP5** must be populated to enable the SW5 to PROGRAMN path.
- **SW2** is system reset that stops the internal clock.
- **SW5** toggles PROGRAMN to trigger re-boot sequence. Pushing this button pulls down the PROGRAMN.
- **JP8** must be populated to enable external SPI Flash access. It is used to force failure when booting from external SPI Flash.
- **D20** is a blue segment LED indicating different image with different number.

For more detailed hardware information, refer to MachXO3D Development Board User Guide (FPGA-DS-02026).

1.3. MachXO3D Devices Boot Mode Options

The boot mode are controlled by dedicated feature row bits that can be erased or programmed independently. Table 1.1 provides all boot configurations for MachXO3D devices. This document only covers Dual Boot features. You can try the remaining boot features with other configurations.

Table 1.1. Booting Mode

Boot Mode	Boot Order	Boot Control Bit in Feature Row			
		MSPI_Persistent Enable	**BOOT_SEL1**	**BOOT_SEL2**	**BOOT_SEL3**
Dual	CFG0-CFG1	0	0	0	0
Dual	CFG1-CFG0	0	1	0	0
Dual	CFG0-EXT	1	0	0	0
Dual	CFG1-EXT	1	0	0	1
Dual	EXT-CFG0	1	0	1	0
Dual	EXT-CFG1	1	0	1	1
Dual	check version and boot from the former	0	1	0	1
Dual	check version and boot from the latter	0	1	1	1
Dual	EXT-EXT	1	1	1	x
No Boot*		0	0	1	x
Single	EXT	1	1	0	x
Single	CFG0	0	1	1	0
Single	CFG1	0	0	0	1

***Note**: No Boot option keeps MachXO3D devices in unconfigured status right after power up, without using any image loaded to the SRAM.

2. Demo Package

2.1. Hardware Requirements

To run the demo, the following hardware are required:

- PC running Windows 7 Operating System
- MachXO3D Development Kit including Mini-USB cable

2.2. Software Requirements

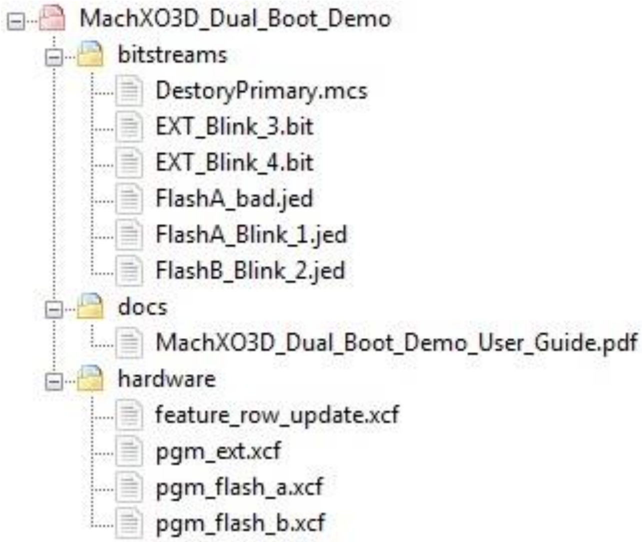
To run the demo, the following software are required:

- MachXO3D Devices Dual Boot Demo package
- Lattice Diamond Programmer 3.11 or later

Note: The software programs are available at www.latticesemi.com/en/Products/DesignSoftwareAndIP.

3. Demo Package Directory Structure

Figure 3.1 shows the demo package directory structure.



```
MachXO3D_Dual_Boot_Demo
    bitstreams
        DestoryPrimary.mcs
        EXT_Blink_3.bit
        EXT_Blink_4.bit
        FlashA_bad.jed
        FlashA_Blink_1.jed
        FlashB_Blink_2.jed
    docs
        MachXO3D_Dual_Boot_Demo_User_Guide.pdf
    hardware
        feature_row_update.xcf
        pgm_ext.xcf
        pgm_flash_a.xcf
        pgm_flash_b.xcf
```

Figure 3.1. Demo Package Directory Structure

4. Prepare the Demo

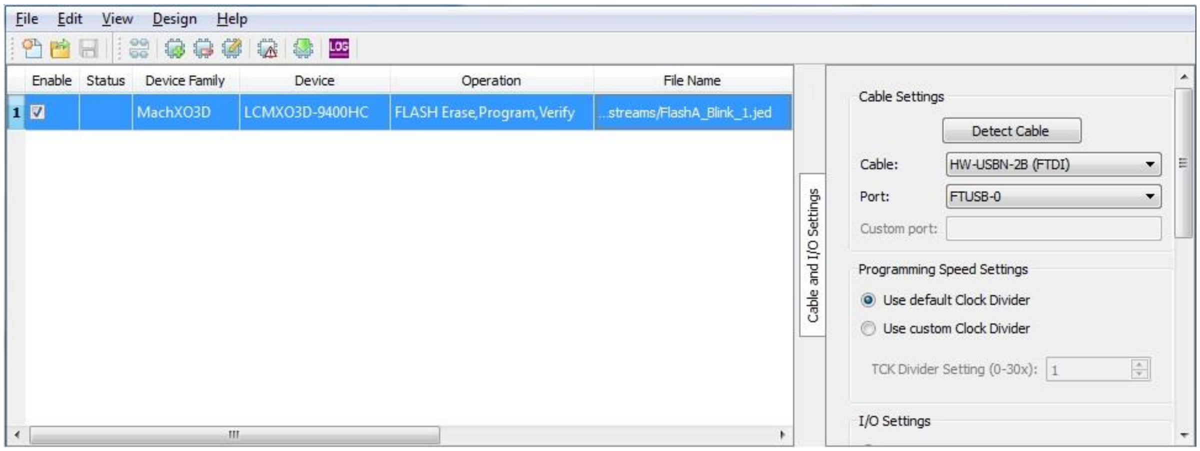
4.1. Prepare the Hardware and Software for the Demo

Before running the Dual Boot demo:

1. Make sure **JP5** and **JP8** are populated when the MachXO3 development board is not powered ON.
2. Power ON the board by connecting **J11** to the PC through the Mini-USB cable.
3. Open the Diamond Programmer software.

4.2. Program the Image to Internal Flash A

1. In the Diamond Programmer graphic user interface (GUI), select **File > Open File…**. Browse to find and open **pgm_flash_a.xcf** from the demo package downloaded (as shown in Figure 3.1). The Diamond Programmer is opened with configuration settings as shown in Figure 4.1.



	Enable	Status	Device Family	Device	Operation	File Name
1	☑		MachXO3D	LCMXO3D-9400HC	FLASH Erase,Program,Verify	...streams/FlashA_Blink_1.jed

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

● Use default Clock Divider

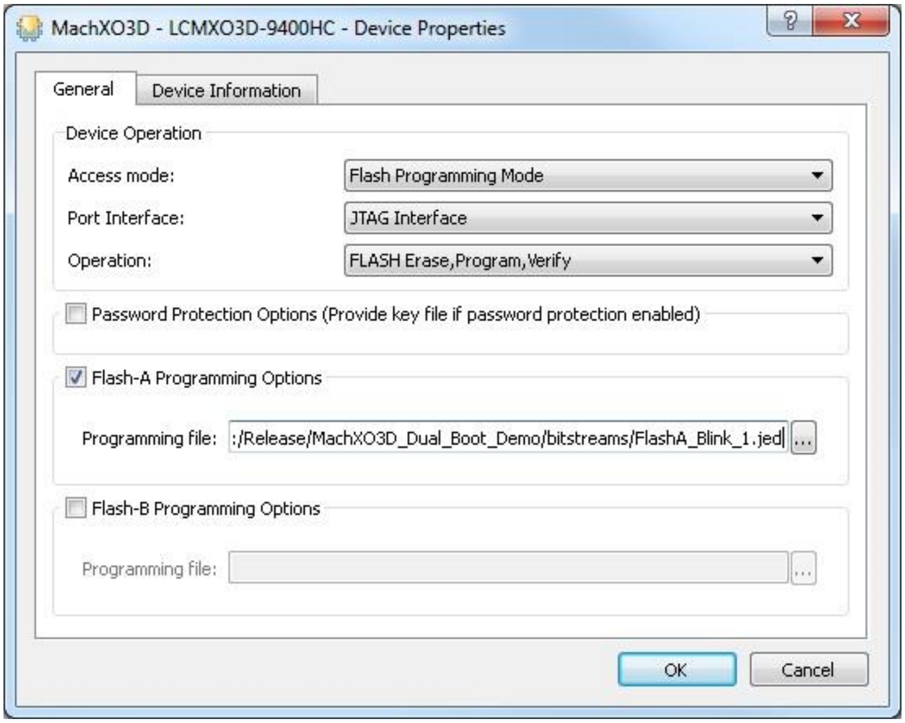
○ Use custom Clock Divider

TCK Divider Setting (0-30x): 1

I/O Settings

Figure 4.1. Load Configuration File for Programming Internal Flash A

2. Select to highlight the only file information row. You can now see file information clearly. Double-click **FlashA_Blink_1.jed** from the **File Name** column (as shown in Figure 4.1). The **Device Properties** dialog box is opened as shown in Figure 4.2.



MachXO3D - LCMXO3D-9400HC - Device Properties

General | Device Information

Device Operation

Access mode: Flash Programming Mode

Port Interface: JTAG Interface

Operation: FLASH Erase,Program,Verify

□ Password Protection Options (Provide key file if password protection enabled)

☑ Flash-A Programming Options

Programming file: :/Release/MachXO3D_Dual_Boot_Demo/bitstreams/FlashA_Blink_1.jed

□ Flash-B Programming Options

Programming file:

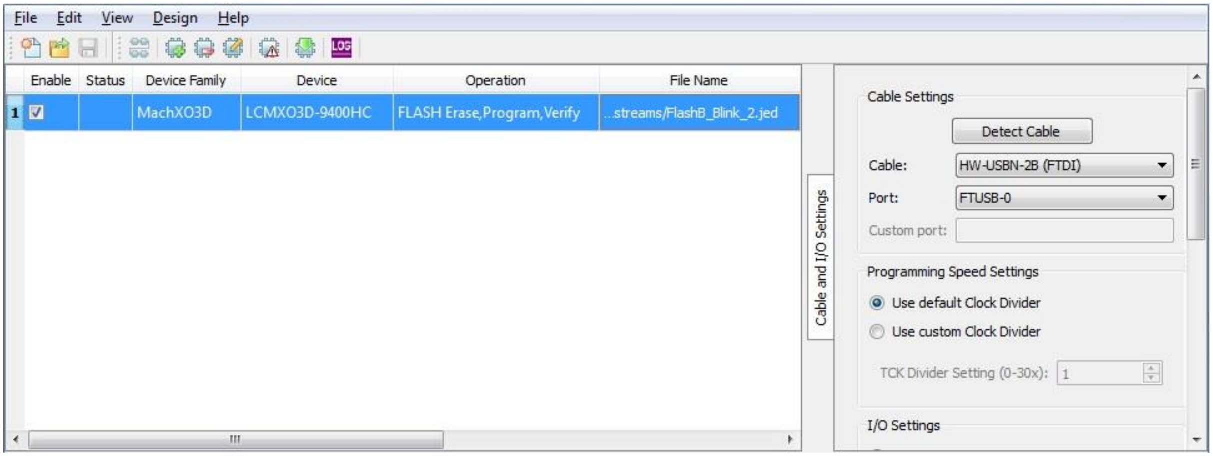
OK | Cancel

Figure 4.2. Set Device Properties for Programming Internal Flash A

3. Note that at this phase you may not be able to operate the *.jed file because of the different demo package installation directory. To make the *.jed file work, in the **Flash-A Programming Options** area, click the **…** button to re-appoint to **FlashA_Blink_1.jed**.
4. Click **OK** as shown in Figure 4.2, and return to Programmer **pgm_flash_a.xcf** interface as shown in Figure 4.1.
5. Click the **Program** button ( ) from the toolbar (Figure 4.1) to download image to Internal Flash A.

4.3. Program the Image to Internal Flash B

1. In the Diamond Programmer GUI, select **File > Open File…**. Browse to find and open **pgm_flash_b.xcf** from the demo package downloaded (as shown in Figure 3.1). The Diamond Programmer is opened with configuration settings as shown in Figure 4.3.



	Enable	Status	Device Family	Device	Operation	File Name
1	☑		MachXO3D	LCMXO3D-9400HC	FLASH Erase,Program,Verify	...streams/FlashB_Blink_2.jed

Cable Settings

Detect Cable

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

● Use default Clock Divider

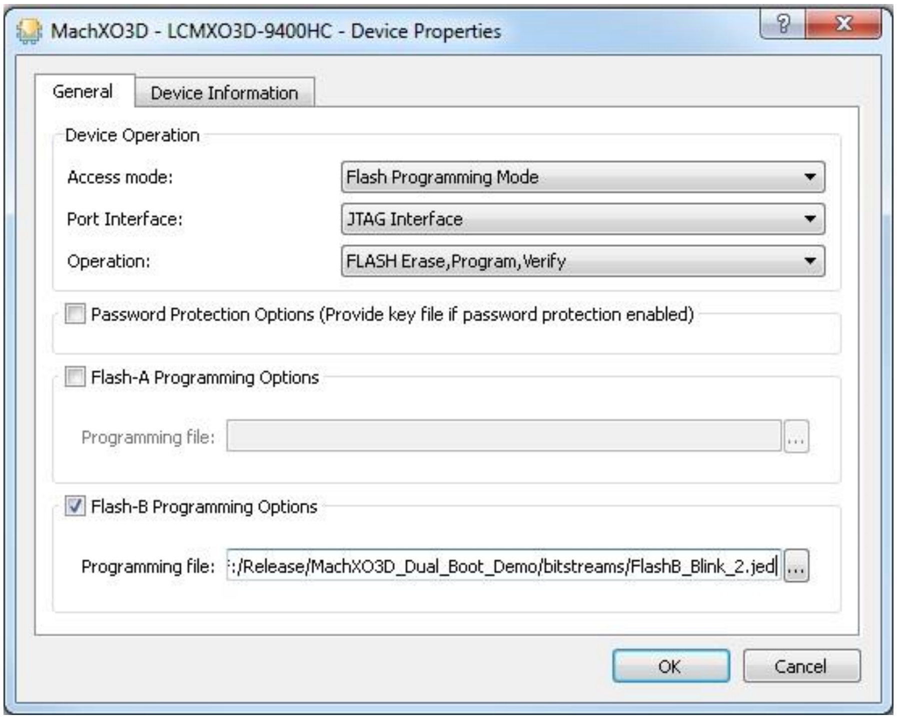
○ Use custom Clock Divider

TCK Divider Setting (0-30x): 1

I/O Settings

Figure 4.3. Load Configuration File for Programming Internal Flash B

2. Select to highlight the only file information row. You can now see file information clearly. Double click **FlashA_Blink_1.jed** from the **File Name** column (as shown in Figure 4.3). The **Device Properties** dialog box is opened as shown in Figure 4.4.



MachXO3D - LCMXO3D-9400HC - Device Properties

Device Operation

Access mode: Flash Programming Mode

Port Interface: JTAG Interface

Operation: FLASH Erase,Program,Verify

□ Password Protection Options (Provide key file if password protection enabled)

□ Flash-A Programming Options

Programming file:

☑ Flash-B Programming Options

Programming file: :/Release/MachXO3D_Dual_Boot_Demo/bitstreams/FlashB_Blink_2.jed

Figure 4.4. Set Device Properties for Programming Internal Flash B

3. Note that at this phase you may not be able to operate the *.jed file because of the different demo package installation directory. To make the *.jed file work, in the **Flash-B Programming Options** area, click the **…** button to re-appoint to **FlashB_Blink_2.jed**.

4. Click **OK** as shown in Figure 4.4, and return to Programmer **pgm_flash_b.xcf** interface as Figure 4.3.

5. Click the **Program** button ( ) from the toolbar to download image to internal Flash B.

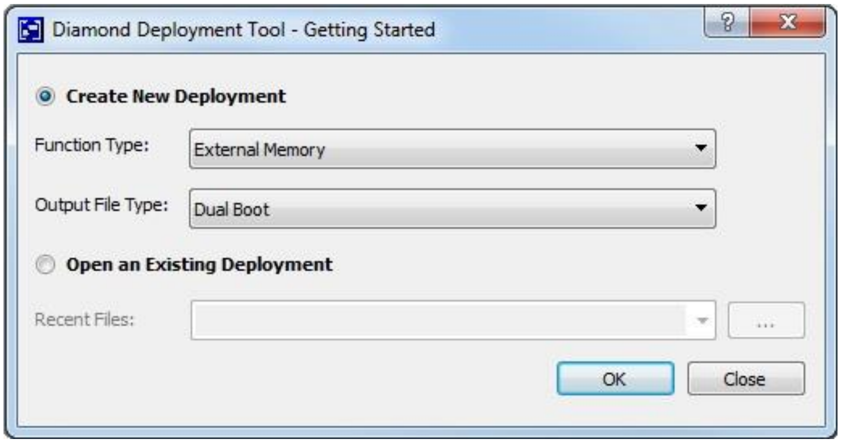
4.4. Create Dual Boot File for External SPI Flash

To create Dual Boot file for external SPI flash:

1. From the Windows **Start** menu, choose **Lattice Diamond Programmer > Deployment Tool** to invoke the Deployment Tool.

2. In the **Getting Started** dialog of the Deployment Tool (Figure 4.5), choose the **Create New Deployment** option, **External Memory** as the **Function Type**, **Dual Boot** as the **Output File Type**, to generate Dual Boot file for external SPI Flash.

3. Click **OK** to start generating the Dual Boot file for external SPI Flash.


Diamond Deployment Tool - Getting Started

● Create New Deployment

Function Type: External Memory

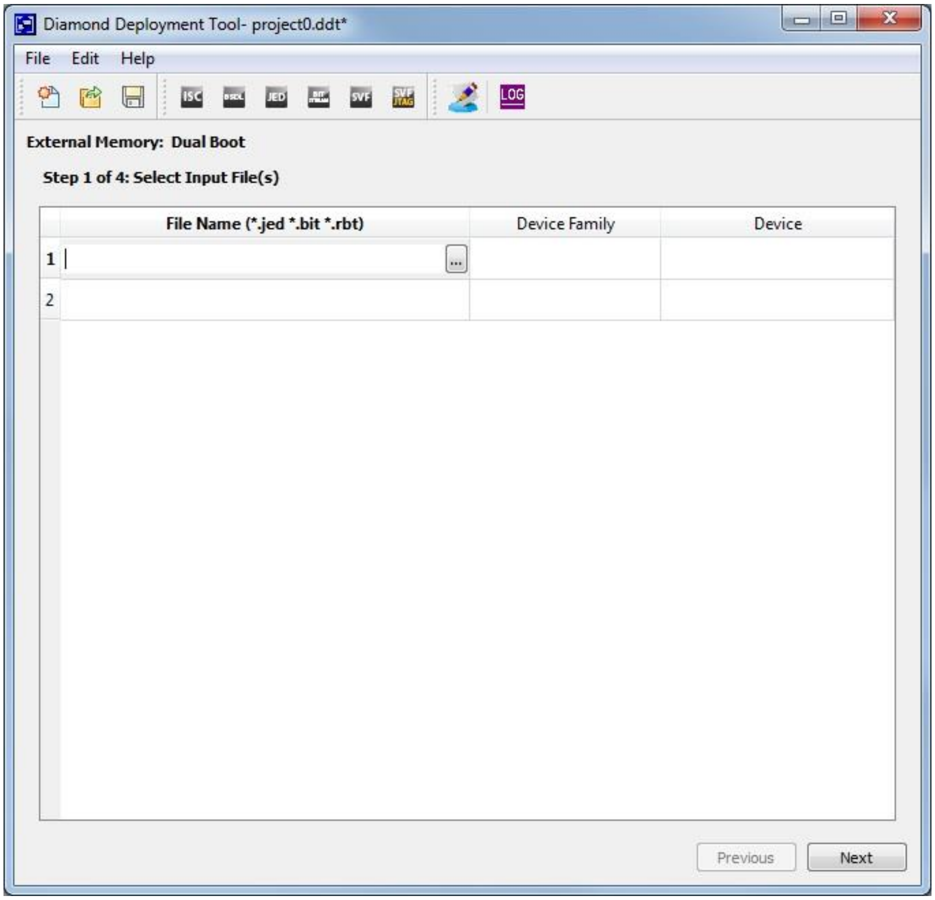
Output File Type: Dual Boot

○ Open an Existing Deployment

Recent Files:

Figure 4.5. Generate Dual Boot File for External Memory

4. To generate the Dual Boot file, click the blank area under the **File Name** field. The **…** button appears (Figure 4.6).
5. Click the **…** button to find and add bit files from the demo package directory, one is **EXT_Blink_3.bit** and the other is **EXT_Blink_4.bit** (as shown in Figure 3.1). After adding the two bit files, you can get the device family and device information accordingly (as shown in Figure 4.7).
6. Click **Next**.

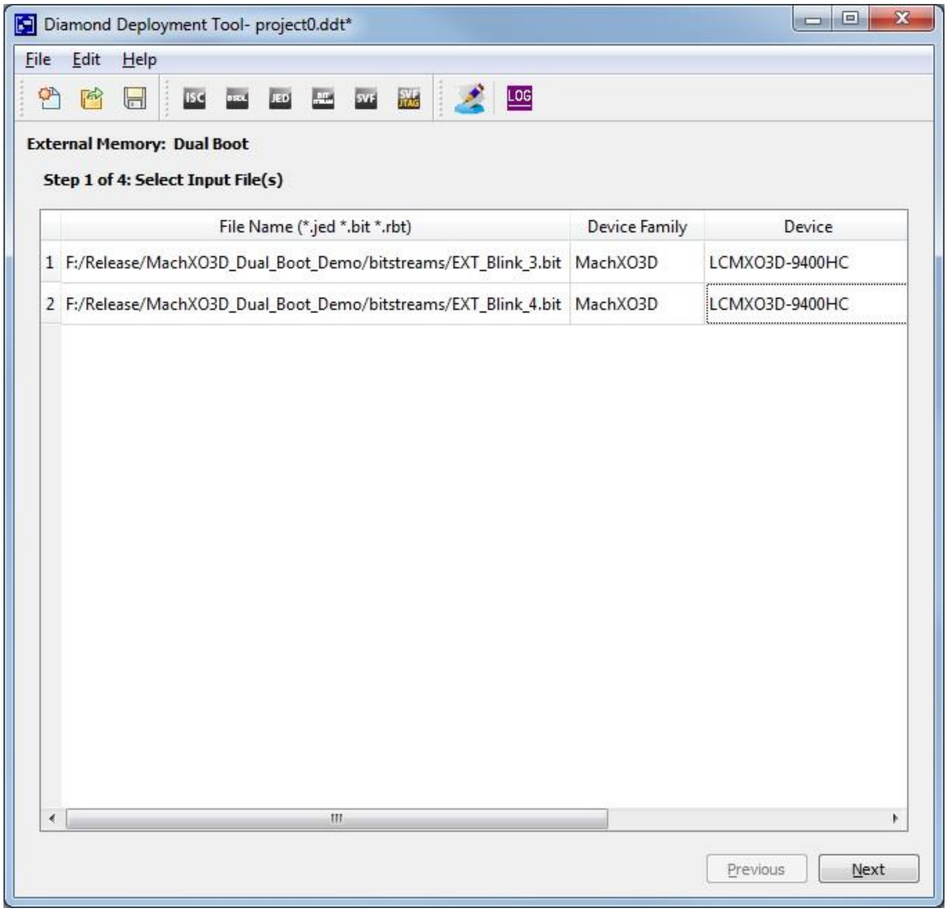

Diamond Deployment Tool- project0.ddt*

External Memory: Dual Boot

Step 1 of 4: Select Input File(s)

	File Name (*.jed *.bit *.rbt)	Device Family	Device
1			
2			

Figure 4.6. Select Input Files (1)



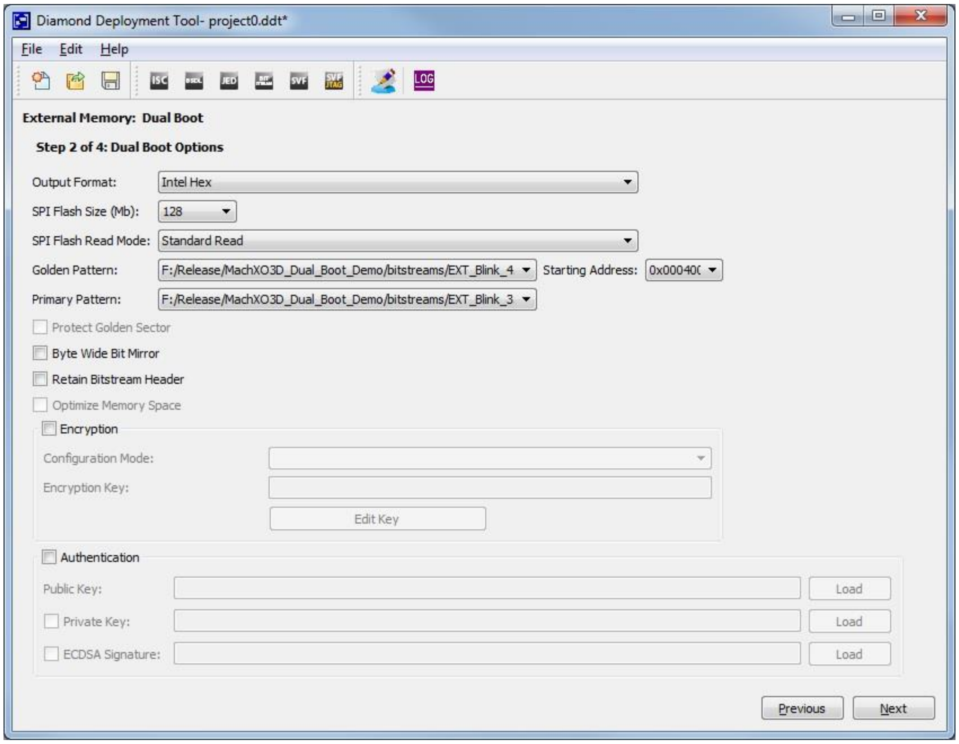
External Memory: Dual Boot

Step 1 of 4: Select Input File(s)

	File Name (*.jed *.bit *.rbt)	Device Family	Device
1	F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3.bit	MachXO3D	LCMXO3D-9400HC
2	F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_4.bit	MachXO3D	LCMXO3D-9400HC

Figure 4.7. Select Input Files (2)

7. As shown in Figure 4.8, choose **Intel Hex** as the **Output Format**, **128** for **SPI Flash Size (Mb)**, **EXT_Blink_4.bit** as the **Golden Pattern**, **EXT_Blink_3.bit** as the **Primary Pattern**, and with **Starting Address** from **0x00040000** to **0x00FA0000** per system requirements. The default is **0x00040000**.
8. Click **Next**.



External Memory: Dual Boot

Step 2 of 4: Dual Boot Options

Output Format: Intel Hex

SPI Flash Size (Mb): 128

SPI Flash Read Mode: Standard Read

Golden Pattern: F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_4 Starting Address: 0x00040(

Primary Pattern: F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3

□ Protect Golden Sector

□ Byte Wide Bit Mirror

□ Retain Bitstream Header

□ Optimize Memory Space

□ Encryption

Configuration Mode:

Encryption Key:

□ Authentication

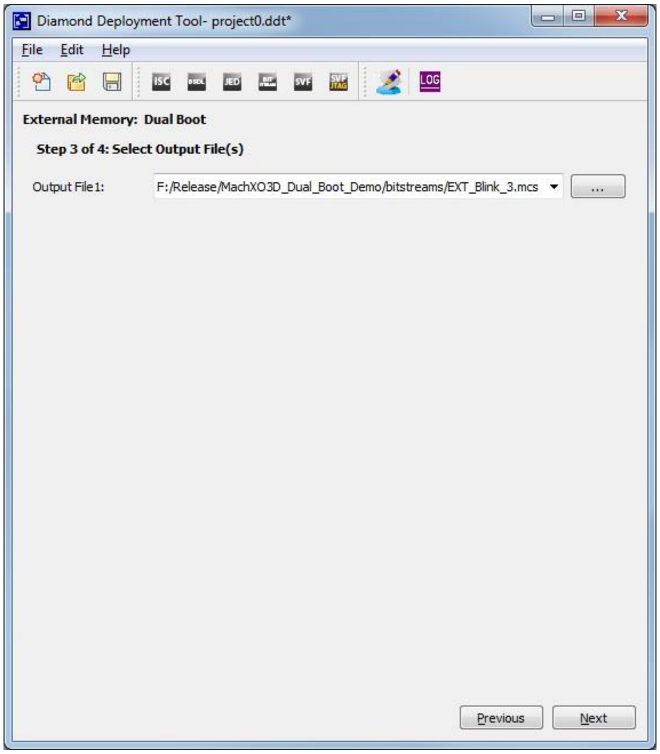
Public Key:

□ Private Key:

□ ECDSA Signature:

Figure 4.8. Dual Boot Options

9. Enter the desired output file name (*.mcs), for example, Dual_EXT_3_4.mcs, into the demo package **bitstreams** folder, as shown in Figure 4.9.
10. Click **Next**.



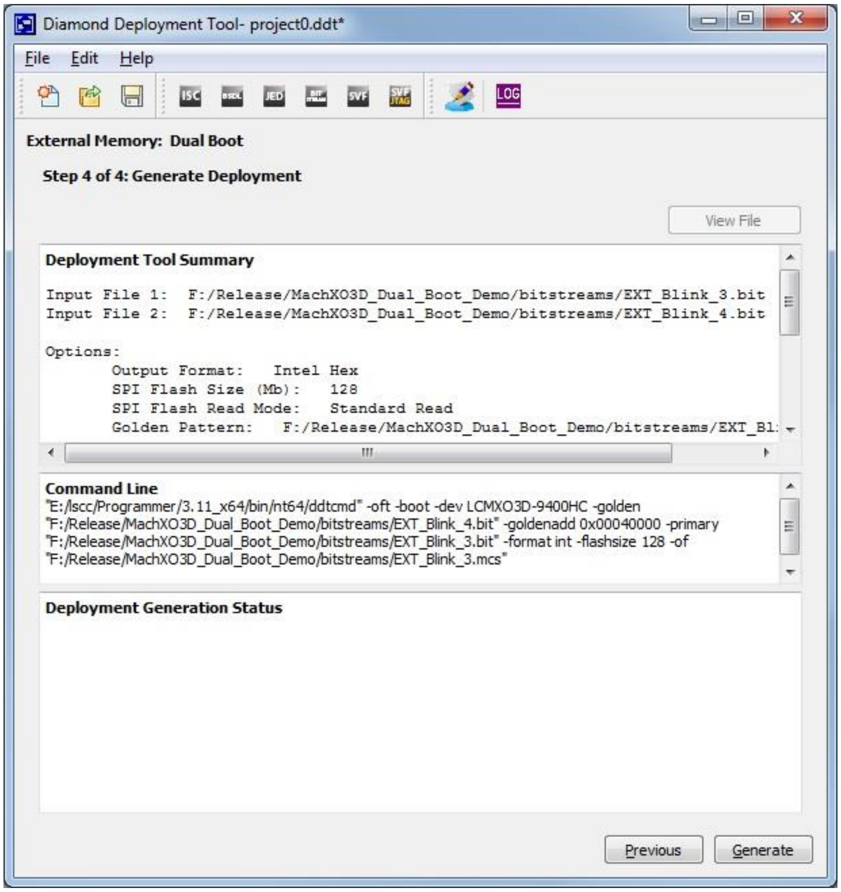
External Memory: Dual Boot

Step 3 of 4: Select Output File(s)

Output File 1: F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3.mcs

Figure 4.9. Define Output File

11. Now you can review the information displayed in the **Deployment Tool Summary** area (as shown in Figure 4.10). Click the **Generate** button.



External Memory: Dual Boot

Step 4 of 4: Generate Deployment

Deployment Tool Summary

```
Input File 1:  F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3.bit
Input File 2:  F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_4.bit

Options:
       Output Format:   Intel Hex
       SPI Flash Size (Mb):   128
       SPI Flash Read Mode:   Standard Read
       Golden Pattern:   F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Bl:
```

Command Line

```
"E:/lscc/Programmer/3.11_x64/bin/nt64/ddtcmd" -oft -boot -dev LCMXO3D-9400HC -golden
"F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_4.bit" -goldenadd 0x00040000 -primary
"F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3.bit" -format int -flashsize 128 -of
"F:/Release/MachXO3D_Dual_Boot_Demo/bitstreams/EXT_Blink_3.mcs"
```

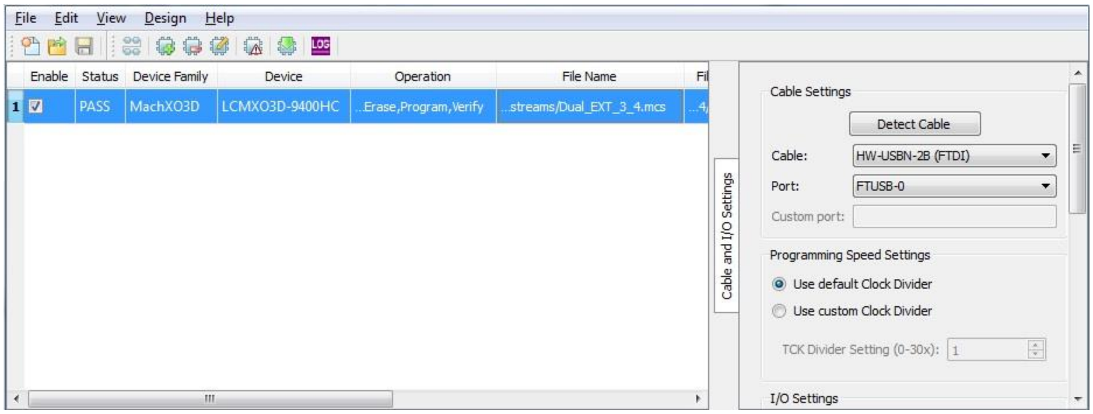
Deployment Generation Status

Figure 4.10. Generate Deployment

4.5. Program the Bitstream to External SPI Flash

With the bitstream file, for example, Dual_EXT_3_4.mcs, created in the previous Create Dual Boot File for External SPI Flash section, you can program this bitstream file to the external SPI Flash.

1. In the Diamond Programmer GUI, select **File > Open File…**. Browse to find and open **pgm_ext.xcf** from the demo package downloaded (as shown in Figure 3.1). The Diamond Programmer is opened with configuration settings as shown in Figure 4.11.



	Enable	Status	Device Family	Device	Operation	File Name	Fi
1	☑	PASS	MachXO3D	LCMXO3D-9400HC	...Erase,Program,Verify	...streams/Dual_EXT_3_4.mcs	...4,

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

● Use default Clock Divider

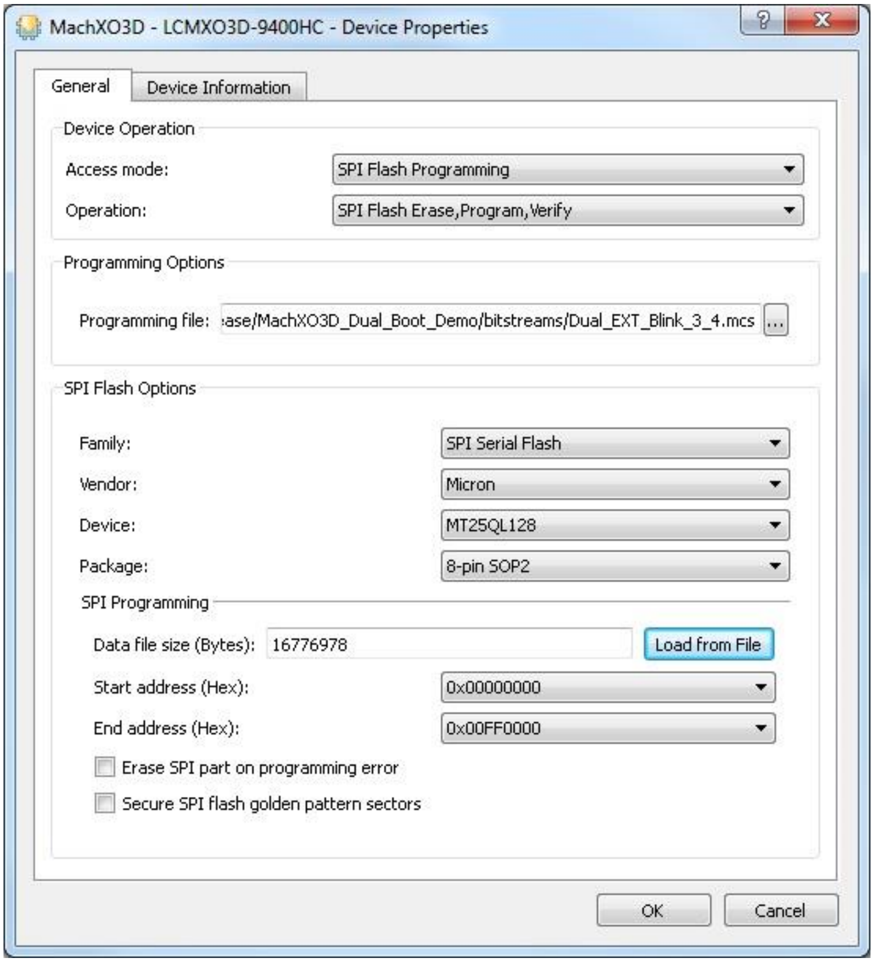
○ Use custom Clock Divider

TCK Divider Setting (0-30x): 1

I/O Settings

Figure 4.11. Loading Configuration File for Programming External SPI Flash

2. Select to highlight the only file information row. You can now see file information clearly. Double-click **Dual_EXT_3_4.mcs** from the **File Name** column (as shown in Figure 4.11). The **Device Properties** dialog box is opened as shown in Figure 4.12.



MachXO3D - LCMXO3D-9400HC - Device Properties

Device Operation
Access mode: SPI Flash Programming
Operation: SPI Flash Erase,Program,Verify

Programming Options
Programming file: :ase/MachXO3D_Dual_Boot_Demo/bitstreams/Dual_EXT_Blink_3_4.mcs ...

SPI Flash Options
Family: SPI Serial Flash
Vendor: Micron
Device: MT25QL128
Package: 8-pin SOP2

SPI Programming
Data file size (Bytes): 16776978
Start address (Hex): 0x00000000
End address (Hex): 0x00FF0000
☐ Erase SPI part on programming error
☐ Secure SPI flash golden pattern sectors

Figure 4.12. Setting Device Properties for External SPI Flash

3. In the **Device Properties** dialog box, click **…** in the **Programming Options** area to change the directory of the bitstream file to a customer-specific one, as shown in the Create Dual Boot File for External SPI Flash section.

4. Click **OK**. Return to Programmer **pgm_ext.xcf** interface as shown in Figure 4.11.

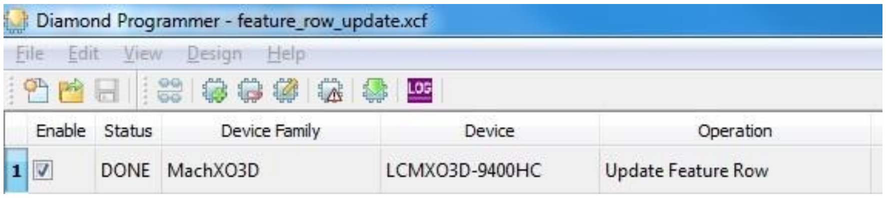
5. Click the **Program** button ( ) from the toolbar to download image to external SPI Flash.

5. Run the Demo

5.1. EXT-CFG0 Dual Boot Demo

To run the EXT-CFG0 Dual Boot demo:

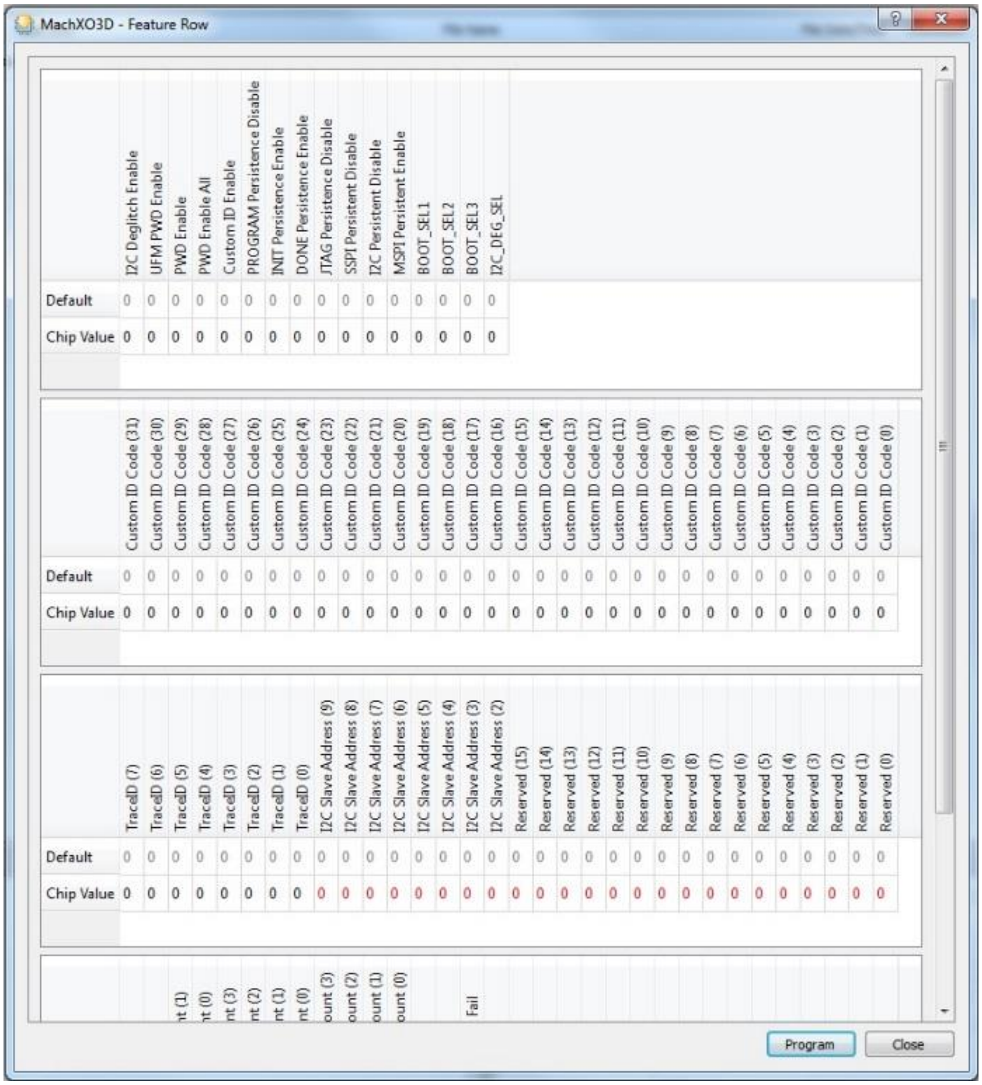
1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. From the Diamond Programmer GUI, choose **File > Open file…**. Select **feature_row_update.xcf** from the demo package downloaded (as shown in Figure 3.1). The feature row update configuration information is shown (Figure 5.1).



	Enable	Status	Device Family	Device	Operation
1	☑	DONE	MachXO3D	LCMXO3D-9400HC	Update Feature Row

Figure 5.1. Load Configuration File for Feature Row Update

3. Click the **Program** button ( ) to open the feature row update interface. The default settings are shown in Figure 5.2.



MachXO3D - Feature Row

	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

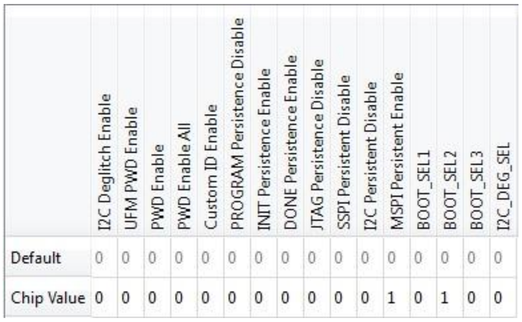
	Custom ID Code (31)	Custom ID Code (30)	Custom ID Code (29)	Custom ID Code (28)	Custom ID Code (27)	Custom ID Code (26)	Custom ID Code (25)	Custom ID Code (24)	Custom ID Code (23)	Custom ID Code (22)	Custom ID Code (21)	Custom ID Code (20)	Custom ID Code (19)	Custom ID Code (18)	Custom ID Code (17)	Custom ID Code (16)	Custom ID Code (15)	Custom ID Code (14)	Custom ID Code (13)	Custom ID Code (12)	Custom ID Code (11)	Custom ID Code (10)	Custom ID Code (9)	Custom ID Code (8)	Custom ID Code (7)	Custom ID Code (6)	Custom ID Code (5)	Custom ID Code (4)	Custom ID Code (3)	Custom ID Code (2)	Custom ID Code (1)	Custom ID Code (0)
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	TraceID (7)	TraceID (6)	TraceID (5)	TraceID (4)	TraceID (3)	TraceID (2)	TraceID (1)	TraceID (0)	I2C Slave Address (9)	I2C Slave Address (8)	I2C Slave Address (7)	I2C Slave Address (6)	I2C Slave Address (5)	I2C Slave Address (4)	I2C Slave Address (3)	I2C Slave Address (2)	Reserved (15)	Reserved (14)	Reserved (13)	Reserved (12)	Reserved (11)	Reserved (10)	Reserved (9)	Reserved (8)	Reserved (7)	Reserved (6)	Reserved (5)	Reserved (4)	Reserved (3)	Reserved (2)	Reserved (1)	Reserved (0)
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

nt (1) nt (0) nt (3) nt (2) nt (1) nt (0) ount (3) ount (2) ount (1) ount (0) Fail

Figure 5.2. Feature Row Update

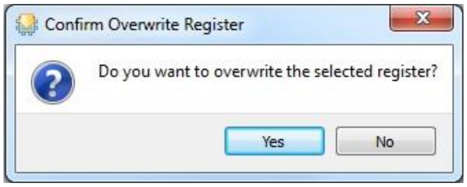
4. Click the **Chip Value** under the selected bit name to toggle the value between 1 and 0. Change feature row setting for EXT-CFG0, according to Table 1.1, as shown in Figure 5.3.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Figure 5.3. Feature Row Settings for EXT-CFG0 Dual Boot Mode

5. Click the **Program** button in the **Feature Row** dialog box (Figure 5.2). The **Confirm Overwrite Register** dialog box opens, as shown in Figure 5.4.



Do you want to overwrite the selected register?

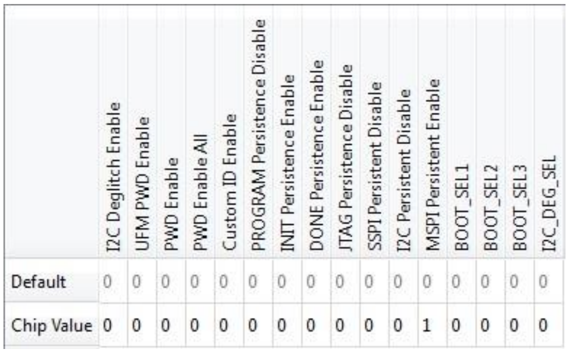
Figure 5.4. Confirm Overwrite Register

6. Click **Yes** to confirm overwriting the selected register. The new feature row configuration is updated to the MachXO3D device.
7. The Segment LED **D20** (Figure 1.2) blinks with 3, which indicates the MachXO3D development board is booted from the external SPI Flash.
8. Power OFF the board. Remove jumper **JP8** to physically disable the external SPI Flash.
9. Try rebooting the board with push button **SW5.** Or, repower the board. **D20** blinks with 1, which indicates the MachXO3D development board fails to boot from the external SPI Flash. The MachXO3D development board switches to boot from the internal Flash A.
10. Recover the initial hardware configuration by adding jumper **JP8** for running the follow-up Demo steps.

5.2. CFG0-EXT Dual Boot Demo

To run the CFG0-EXT Dual Boot demo:

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change the feature row setting for CFG0-EXT according to Table 1.1, as shown in Figure 5.5.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

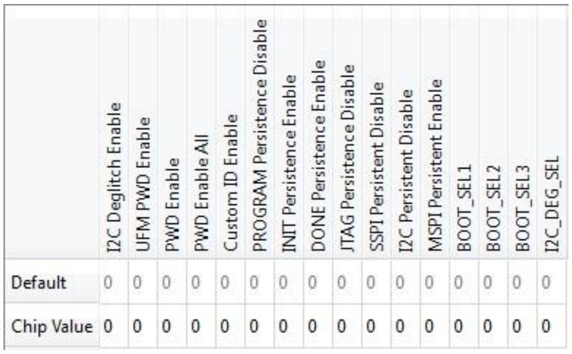
Figure 5.5. Feature Row Settings for CFG0-EXT Dual Boot Mode

3. After the new feature row configuration is updated to the MachXO3D device, the segment LED **D20** blinks with 1, which indicates the MachXO3D development board is booted from the internal Flash A.
4. From the Diamond Programmer GUI, choose **File > Open file…**. Open the **pgm_flash_a.xcf** from the demo package downloaded (as shown in Figure 3.1). The Diamond Programmer is opened with configuration settings as shown in Figure 4.1. Open the Device Properties dialog (Figure 4.2), click **…** to change the **Programming file** to **FlashA_bad.jed** for the demo package in the **Flash-A Programming Options** area. **FlashA_bad.jed** is an engineering bit file simulates internal Flash A with one bit error.
5. Click **OK** as shown in Figure 4.2, and return to Programmer **pgm_flash_a.xcf** interface as shown in Figure 4.1.
6. Click the **Program** button ( ) from the toolbar (Figure 4.1) to download image to Internal Flash A.
7. After programming the internal Flash A, try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 3, which indicates the MachXO3D development board fails to boot from the internal Flash A. The MachXO3D development board switches to boot from the external SPI Flash.

5.3. CFG0-CFG1 Dual Boot Demo

To run the CFG0-CFG1 Dual Boot demo:

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change the feature row setting for CFG0-CFG1 according to Table 1.1, as shown in Figure 5.6.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

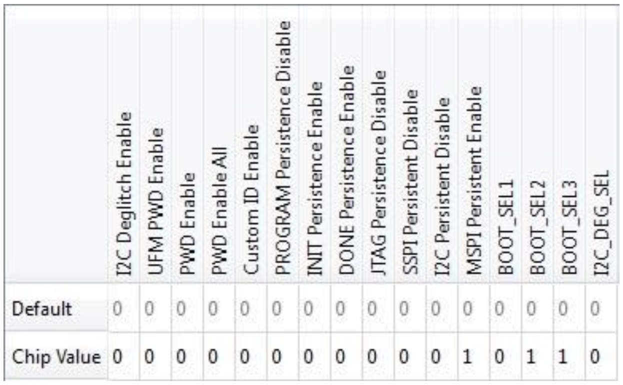
Figure 5.6. Feature Row Settings for CFG0-CFG1 Dual Boot Mode

3. As the continued demo after the CFG0-EXT Dual Boot Demo section with a bad pattern in internal Flash A, after the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 2, which indicates the MachXO3D development board is booted from the internal Flash B as the backup.
4. Repeat all the steps in the Program the Image to Internal Flash A section to recover the good pattern to internal Flash A.
5. Try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 1, which indicates the MachXO3D development board is successfully booted from the internal Flash A.

5.4. EXT-CFG1 Dual Boot Demo

To run the EXT-CFG1 Dual Boot demo:

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change the feature row setting for EXT-CFG1 according to Table 1.1, as shown in Figure 5.7.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

Figure 5.7. Feature Row Settings for EXT-CFG1 Dual Boot Mode

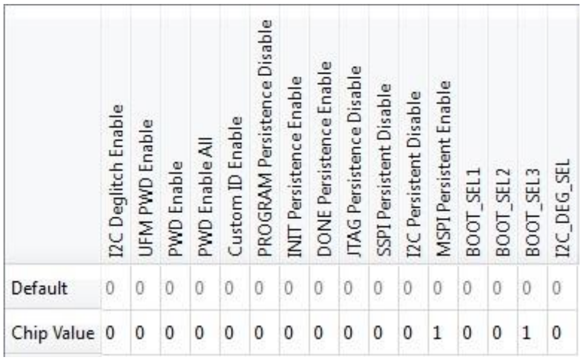
3. After the new feature row configuration is updated to the MachXO3D device, the segment LED **D20** blinks with 3, which indicates the MachXO3D development board is booted from the external SPI Flash.
4. Power OFF the board. Remove jumper **JP8** to physically disable the external SPI Flash.

5. Try rebooting the board by pushing button **SW5**. Or, repower the board. **D20** blinks with 2, which indicates MachXO3D fails to boot from the external SPI Flash. The MachXO3D development board switches to boot from the internal Flash B.
6. Recover the initial hardware configuration by adding jumper **JP8** for running the follow-up Demo steps.

5.5. CFG1-EXT Dual Boot Demo

To run the CFG1-EXT Dual Boot demo:

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change the feature row setting for CFG1-EXT according to Table 1.1, as shown in Figure 5.8.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

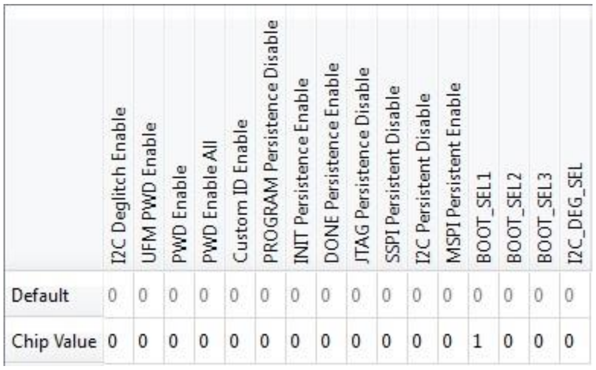
Figure 5.8. Feature Row Settings for CFG1-EXT Dual Boot Mode

3. After the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 2, which indicates the MachXO3D development board is booted from internal Flash B.
4. From the Diamond Programmer GUI, choose **File > Open file…**. Open the **pgm_flash_b.xcf** from the demo package downloaded (as shown in Figure 3.1). The Diamond Programmer is opened with configuration settings as shown in Figure 4.3. Open the Device Properties dialog (Figure 4.4), click **…** to change the **Programming file** to **FlashA_bad.jed** for the demo package in the **Flash-B Programming Options** area. **FlashA_bad.jed** is an engineering bit file simulates internal Flash B with one bit error.
5. Click **OK** as shown in Figure 4.4, and return to Programmer **pgm_flash_b.xcf** interface as Figure 4.3.
6. Click the **Program** button ( ) from the toolbar to download image to internal Flash B.
7. After programming the internal Flash B, try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 3, which indicates the MachXO3D development board fails to boot from the internal Flash B. The MachXO3D development board switches to boot from the external SPI Flash.

5.6. CFG1-CFG0 Dual Boot Demo

To run the CFG1-CFG0 Dual Boot demo:

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change the feature row setting for CFG1-CFG0 according to Table 1.1, as shown in Figure 5.9.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

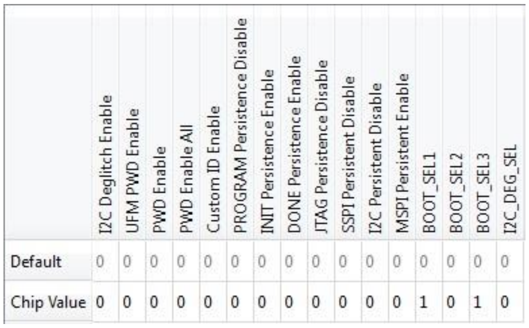
Figure 5.9. Feature Row Settings for CFG1-CFG0 Dual Boot Mode

3. As the continue demo after the CFG1-EXT Dual Boot Demo section with a bad pattern in internal Flash B, after the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 1, which indicates the MachXO3D development board is booted from internal Flash A as the backup.
4. Repeat steps described in the Program the Image to Internal Flash B section to recover the good pattern to internal Flash B.
5. Try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 2, which indicates the MachXO3D development board is successfully booted from the internal Flash B.

5.7. Ping-pong Dual Boot Demo - Check Version and Boot from the Former Version

This demo is to show you how to boot from the former version within CFG0 and CFG1 by checking version automatically.

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change feature row settings for checking version and primarily booting from the former version according to Table 1.1, as shown in Figure 5.10.



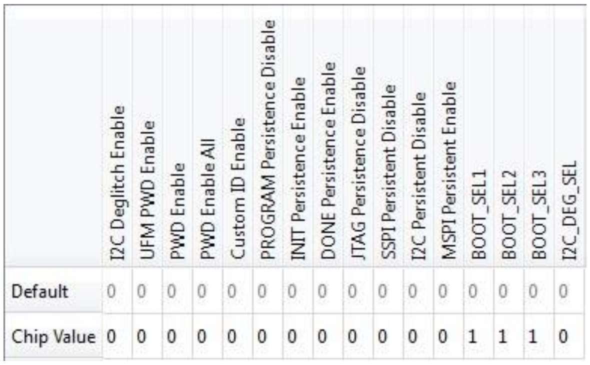
	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Figure 5.10. Feature Row Settings for Former Version as Primary Dual Boot Mode

3. After the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 1, which indicates the MachXO3D development board is booted from the internal Flash A with former version tag. Internal Flash B is the newly updated pattern with larger version number, as it was updated in the previous CFG1-CFG0 Dual Boot Demo section.
4. Repeat steps described in the Program the Image to Internal Flash A section to reprogram the good pattern to internal Flash A.
5. Try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 2, which indicates the MachXO3D development board is booted from internal Flash B with former version tag.

5.8. Ping-pong Dual Boot Demo - Check Version and Boot from the Latter Version

1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change feature row setting for checking version and primarily booting from the latter version according to Table 1.1, as shown in Figure 5.11.



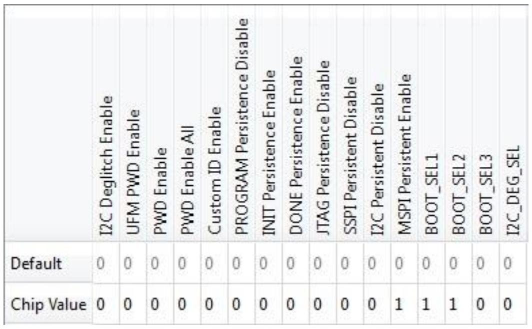
	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Figure 5.11. Feature Row Settings for Latter Version as Primary Dual Boot Mode

3. After the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 1, which indicates MachXO3D development board is booted from internal Flash A with latter version tag. Internal Flash A is the newly updated pattern with larger version number, as it was updated in the previous Ping-pong Dual Boot Demo - Check Version and Boot from the Former Version section.
4. Repeat steps described in the Program the Image to Internal Flash B section to reprogram the good pattern to internal Flash B.
5. Try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 2, which indicates the MachXO3D development board is booted from internal Flash B with latter version tag.
6. Repeat steps described in the CFG0-EXT Dual Boot Demo section to program **FlashA_bad.jed** to internal Flash A so as to verify the backup boot function under Ping-pong Dual Boot mode. **D20** blinks with 2, which indicates the MachXO3D development board is booted from internal Flash B when the board fails to boot from internal Flash A with latter version tag.

5.9. EXT-EXT Dual Boot Demo

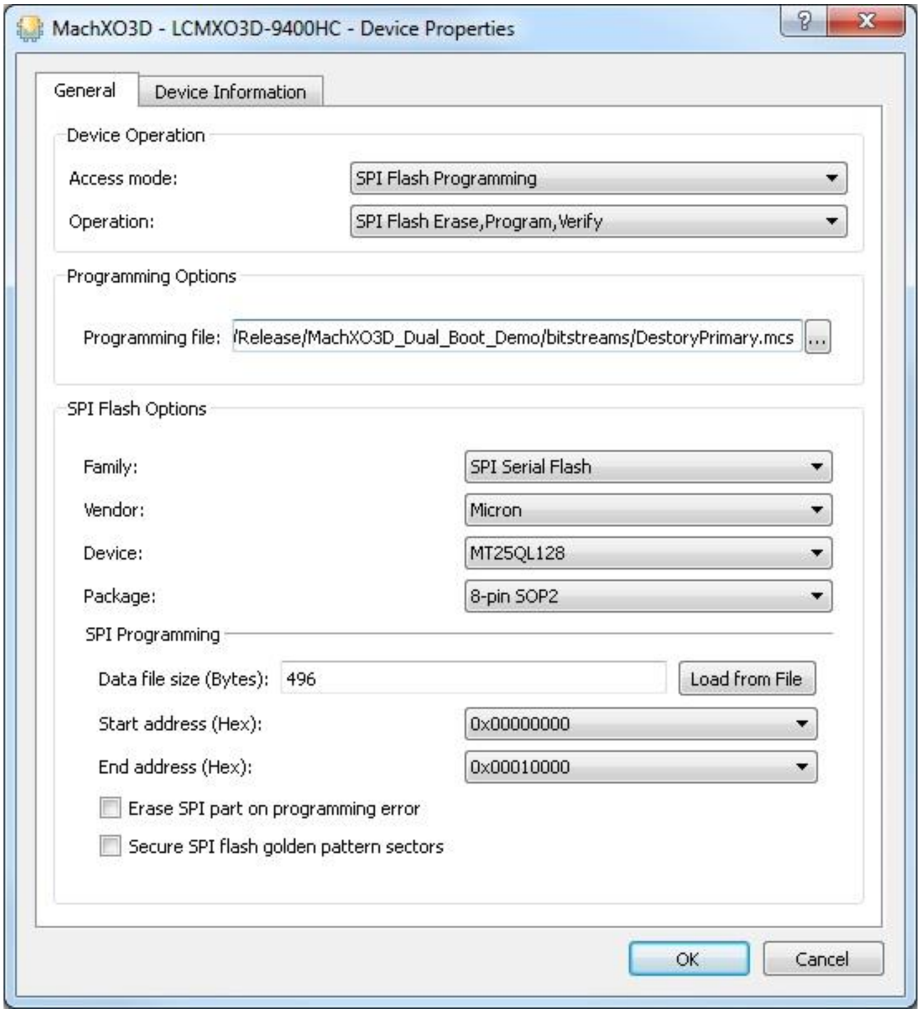
1. Open the Diamond Programmer. Make sure the board is powered ON with Mini-USB cable connecting to PC.
2. Repeat the feature row updating steps as described in the previous EXT-CFG0 Dual Boot Demo section to change feature row setting for EXT-EXT according Table 1.1, as shown in Figure 5.12.



	I2C Deglitch Enable	UFM PWD Enable	PWD Enable	PWD Enable All	Custom ID Enable	PROGRAM Persistence Disable	INIT Persistence Enable	DONE Persistence Enable	JTAG Persistence Disable	SSPI Persistent Disable	I2C Persistent Disable	MSPI Persistent Enable	BOOT_SEL1	BOOT_SEL2	BOOT_SEL3	I2C_DEG_SEL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Figure 5.12. Feature Row Settings for EXT-EXT Dual Boot Mode

3. After the new feature row configuration is updated to the MachXO3D device, the Segment LED **D20** blinks with 3, which indicates the MachXO3D development board is booted from Primary of the external SPI Flash.
4. From the Diamond Programmer GUI, choose **File > Open file…**. Browse to find and open **pgm_ext.xcf** from the demo package downloaded (as shown in Figure 3.1). Click **…** to change the **Programming file** to **DestoryPrimary.mcs** in the **Programming Options** area, as shown in Figure 5.13, which destroys the Primary pattern of the external SPI Flash.



MachXO3D - LCMXO3D-9400HC - Device Properties

Device Operation

Access mode: SPI Flash Programming

Operation: SPI Flash Erase,Program,Verify

Programming Options

Programming file: /Release/MachXO3D_Dual_Boot_Demo/bitstreams/DestoryPrimary.mcs

SPI Flash Options

Family: SPI Serial Flash

Vendor: Micron

Device: MT25QL128

Package: 8-pin SOP2

SPI Programming

Data file size (Bytes): 496

Start address (Hex): 0x00000000

End address (Hex): 0x00010000

☐ Erase SPI part on programming error

☐ Secure SPI flash golden pattern sectors

Figure 5.13. Set Device Properties for External SPI Flash to Disrupt Primary Pattern

5. Click **OK**. Return to Programmer **pgm_ext.xcf** interface as shown in Figure 4.11.
6. Click the **Program** button ( ) from the toolbar to download image to external SPI Flash.
7. After programming the external SPI Flash, try rebooting the board by pushing **SW5**. Or, repower the board. **D20** blinks with 4, which indicates the MachXO3D development board is booted from the Golden pattern of the external SPI Flash after the board fails to boot from the Primary pattern of the external SPI Flash.

References

Lattice Semiconductor Documents

This is a list of related documents that are available from your Lattice Semiconductor sales representative.

Document	Title
FPGA-DS-02026	*MachXO3D Family Data Sheet*
FPGA-EB-02020	*MachXO3D Development Board User Guide*
FPGA-TN-02069	*MachXO3D Programming and Configuration Usage Guide*

Technical Support

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

Revision History

Revision 1.0, October 2018

Section	Change Summary
All	Initial release.

7th Floor, 111 SW 5th Avenue
Portland, OR 97204, USA
T 503.268.8000
www.latticesemi.com