MachXO3 Programming and Configuration User Guide

Technical Note

FPGA-TN-02055-2.7

December 2021

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CRC	Cyclic Redundancy Check
EBR	Embedded Block RAM
GOE	Global Output Enable
GSR	Global Set Reset
GWDIS	Global Write Disable
I^2C	Inter-Integrated Circuit
LUT	Look Up Table
MSPI	Master Serial Peripheral Interface
NVCM	Non Volatile Configuration Memory
POR	Power On Reset
SDM	Self-Download Mode
SSPI	Slave Serial Peripheral Interface

1. Introduction

The MachXO3™ is an SRAM-based Programmable Logic Device that includes an internal Non-Volatile Configuration Memory (NVCM) in the MachXO3L version and On-Chip Flash in the MachXO3LF version. This allows the MachXO3L family to support the best of both worlds – infinitely configurable (SRAM) and non-volatile (NVCM/Flash) capabilities. The MachXO3L/LF provides a rich set of features for programming and configuration of the FPGA. You have many options available to you for building the programming solution that fits your needs. Each of the options available is described in detail so that you can put together the programming and configuration solution that meets your needs.

MachXO3L/LF devices contain two types of memory, SRAM and NVCM (MachXO3L) or Flash (MachXO3LF). SRAM memory contains the active configuration, essentially the *fuses* that define the behavior of the FPGA. The active configuration is, in most cases, retrieved from a non-volatile memory. The non-volatile memory holds the configuration data that is loaded into the FPGAs SRAM. The MachXO3L/LF provides an internal NVCM or Flash memory that stores the configuration data loaded into the MachXO3L/LF SRAM.

2. MachXO3L/LF Features

The key programming and configuration features of MachXO3L devices are:

- Instant-on configuration from internal NVCM (MachXO3L) or Flash (MachXO3LF) – powers up in milliseconds
- Infinite number of configuration cycle through volatile SRAM technology
- MachXO3L multi-time (up to nine times) programmability through non-volatile technology (NVCM)
- Single-chip, secure solution
- Multiple programming and configuration interfaces:
  - 1149.1 JTAG
  - Self-download
  - Slave SPI
  - Master SPI
  - Dual Boot
  - I^2C
- Programming and configuration ports:
  - Slave SPI
  - Master SPI
  - I^2C
  - JTAG
- (MachXO3LF) User Flash Memory (UFM) for non-volatile data storage:
  - Configuration Flash memory overflow
  - EBR Initialization data
  - Application specific data
- (MachXO3LF) Transparent programming of non-volatile memory
- Optional dual boot with external SPI memory
- Optional security bits for design protection
- Optional Bitstream Compression support
- TransFR Capability
  - Leave alone I/O using non-JTAG implementation
- SED support

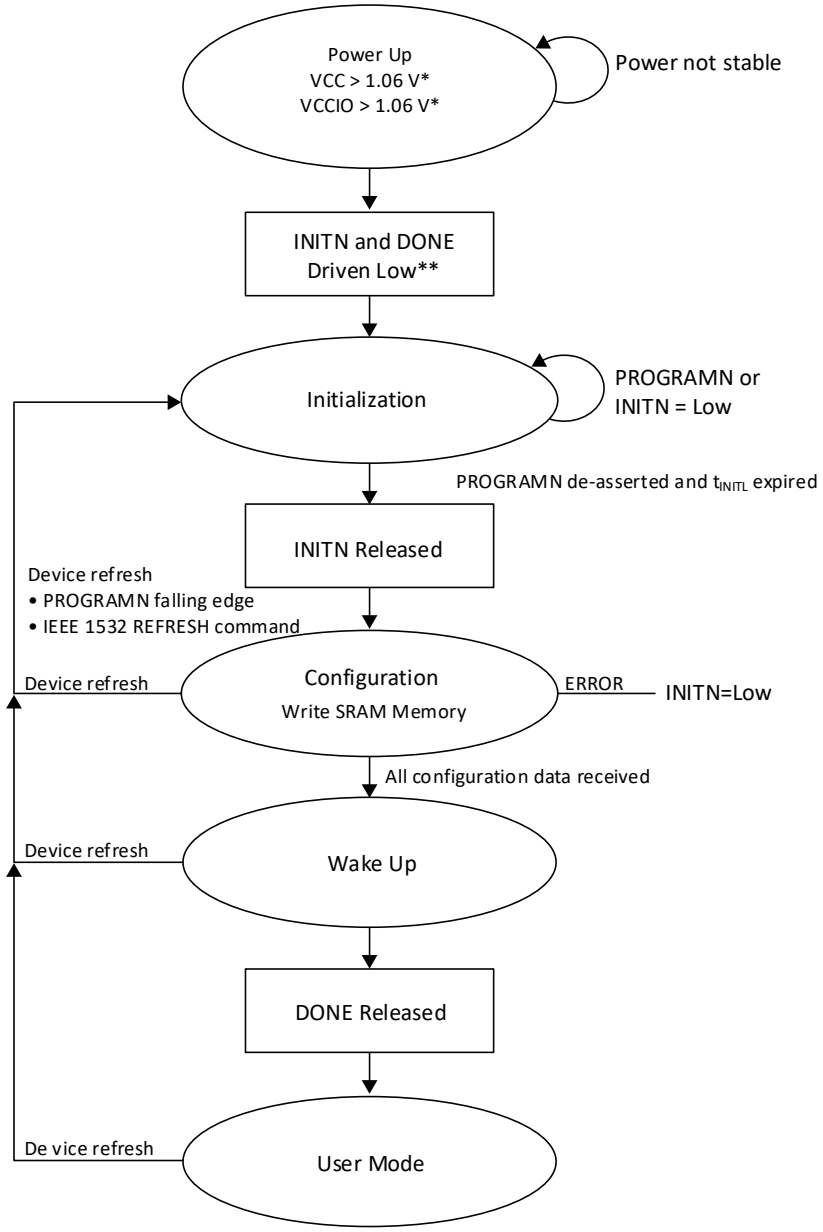
3. Definition of Terms

This document uses the following terms to describe common functions:

- BIT – The BIT file is the configuration data for the MachXO3L/LF that is stored in an external SPI Flash. It is a binary file and is programmed unmodified into the SPI Flash.
- Configuration – Configuration refers to a change in the state of the MachXO3L/LF SRAM memory cells.
- Configuration Data – This is the data read from the non-volatile memory and loaded into the FPGA's SRAM configuration memory. This is also referred to as a bitstream, or device bitstream.
- Configuration Mode – The configuration mode defines the method the MachXO3L/LF uses to acquire the configuration data from the non-volatile memory.
- Internal Flash Memory (MachXO3LF) – JED file or bit file can be programmed directly into the internal flash sector. User does not need to know where an actual page of the configuration data starts. The MachXO3LF configuration engine handles the parsing in the flash to SRAM transfer.
- Internal NVCM (MachXO3L) – JED file or bit file can be programmed directly into the internal non-volatile sector. You do not need to know where an actual page of the configuration data starts. The MachXO3L configuration engine handles the parsing in the NVCM to SRAM transfer.
- JEDEC – The JEDEC file contains the configuration data programmed into the MachXO3L/LF Configuration NVCM or Flash. Format information is provided later in this technical note.
- Number Formats – The following nomenclature is used to denote the radix of numbers
  - 0x – Numbers preceded by *0x* are hexadecimal
  - b (suffix) – Numbers suffixed with *b* are binary
  - All other numbers are decimal
- Offline Mode – Offline mode is a term that is applied to both non-volatile memory programming and SRAM configuration. When using offline mode programming/configuration the FPGA no longer operates in user mode. The contents of the non-volatile or SRAM configuration memory are updated, but the MachXO3L/LF does not perform your logic operations until offline mode programming/configuration is complete.
- Port – A port refers to the physical connection used to perform programming and some configuration operations. Ports on the MachXO3L include JTAG, SPI, and I^2C physical connections.
- Programming – Programming refers to the process used to alter the contents of the internal or external non-volatile configuration memory.
- Transparent Mode (MachXO3LF) – Transparent mode is used to update the Configuration Flash, and User Flash Memory while leaving the MachXO3LF in User Mode.
- User Mode – The MachXO3L is in user mode when configuration is complete, and the FPGA is performing the logic functions you have programmed it to perform.

4. Configuration Process and Flow

Prior to becoming operational, the FPGA goes through a sequence of states including initialization, configuration, and wake-up.



* The voltage level is for the MachXO3 E device. Voltage level may vary for other devices.
** The external INITN and DONE are bidirectional, open-drain I/O only when enabled.

Figure 4.1. Configuration Flow

The MachXO3L/LF sysCONFIG™ ports provide industry standard communication protocols for programming and configuring the FPGA. Each of the protocols shown in Table 4.1 provides a way to access the MachXO3L/LF device's internal NVCM/Flash, or to load its configuration SRAM. The Memory Space Accessibility section provides information about the capabilities of each sysCONFIG port.

The sysCONFIG ports capable of accessing the SRAM have a priority order. The operation of the Configuration Logic is not defined when a low priority sysCONFIG port is interrupted by a higher priority sysCONFIG port. Do not permit simultaneous access to the Configuration Logic using a sysCONFIG port.

4.1. Power-up Sequence

In order for the MachXO3L/LF to operate, power must be applied to the device. During a short period of time, as the voltages applied to the system rise, the FPGA has an indeterminate state.

As power continues to ramp, a Power On Reset (POR) circuit inside the FPGA becomes active. The POR circuit, once active, makes sure the external I/O pins are in a high-impedance state. It also monitors the VCC and VCCIO0 input rails. The POR circuit waits for the following conditions:

- V_{CC} > 1.06 V*
- V_{CCIO0} > 1.06 V*

***Note:** The voltage level is for the MachXO3L/LF E device. Voltage level may vary for other devices.

When these conditions are met, the POR circuit releases an internal reset strobe, allowing the device to begin its initialization process. The MachXO3L/LF asserts INITN active low, and drives DONE low. When INITN and DONE are asserted low the device moves to the initialization state, as shown in Figure 4.2.

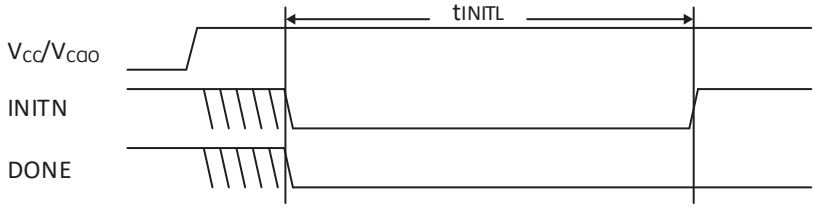


Figure 4.2. Configuration from Power-On-Reset Timing

Note: The external INITN and DONE are bidirectional, open-drain I/O only when enabled.

4.2. Initialization

The MachXO3L/LF enters the memory initialization phase immediately after the Power On Reset circuit drives the INITN and DONE status pins low. The purpose of the initialization state is to clear all of the SRAM memory inside the FPGA.

The FPGA remains in the initialization state until all of the following conditions are met:

- The t_{INITL} time period has elapsed
- The PROGRAMN pin is deasserted
- The INITN pin is no longer asserted low by an external master

The INITN pin provides two functions during the initialization phase. The first is to indicate the FPGA is currently clearing its configuration SRAM. The second is to act as an input preventing the transition from the initialization state to the configuration state.

During the t_{INITL} time period, the FPGA is clearing the configuration SRAM. When the MachXO3L/LF is part of a chain of devices each device has different t_{INITL} initialization times. The FPGA with the slowest t_{INITL} parameter can prevent other devices in the chain from starting to configure. Premature release of the INITN in a multidevice chain may cause configuration of one or more chained devices to fail to configure intermittently.

4.3. Configuration

The rising edge of the INITN pin causes the FPGA to enter the configuration state. The FPGA is able to accept the configuration bitstream created by the Diamond development tools.

The MachXO3L/LF begins fetching configuration data from non-volatile memory. The memory used to configure the MachXO3L/LF is either the internal memory, or an external SPI Flash. The MachXO3L/LF does not leave the Configuration state if there are no memories with valid configuration data. It is necessary to program the non-volatile memory internal or attached to the FPGA, or to program it using the JTAG port.

During the time the FPGA receives its configuration data, the INITN control pin takes on its final function. INITN is used to indicate an error exists in the configuration data. When INITN is high, configuration proceeds without issue. If INITN is asserted low, an error has occurred and the FPGA does not operate.

4.4. Wake-up

Wake-up is the transition from configuration mode to user mode. The MachXO3L/LF's fixed four-phase wake-up sequence starts when the device has correctly received all of its configuration data. When all configuration data is received, the FPGA asserts an internal DONE status bit. The assertion of the internal DONE causes a Wake Up state machine to run that sequences four controls. The four control strobes are:

- Global Output Enable (GOE)
- Global Set/Reset (GSR)
- Global Write Disable (GWDISn)
- External DONE

The first phase of the Wake-Up process is for the MachXO3L/LF to release the Global Output Enable. When it is asserted, permits the FPGA's I/O to exit a high-impedance state and take on their programmed output function. The FPGA inputs are always active. The input signals are prevented from performing any action on the FPGA flip-flops by the assertion of the Global Set/Reset (GSR).

The second phase of the Wake-Up process releases the Global Set/Reset and the Global Write Disable controls.

The Global Set/Reset is an internal strobe that, when asserted, causes all I/O flip-flops, Look Up Table (LUT) flip-flops, distributed RAM output flip-flops, and Embedded Block RAM output flip-flops that have the GSR enabled attribute to be set/cleared per their hardware description language definition.

The Global Write Disable is a control that overrides the write enable strobe for all RAM logic inside the FPGA. The inputs on the FPGA are always active, as mentioned in the Global Output Enable section. Keeping GWDIS asserted prevents accidental corruption of the instantiated RAM resources inside the FPGA.

The last phase of the Wake-Up process is to assert the external DONE pin. The external DONE is a bidirectional, open-drain I/O only when it is enabled. An external agent that holds the external DONE pin low prevents the wake-up process of the MachXO3L/LF from proceeding. Only after the external DONE, if enabled, is active high does the final wake-up phase complete. Wake-Up completes uninterrupted when the external DONE pin is not enabled.

Once the final wake-up phase is complete, the FPGA enters user mode.

4.5. User Mode

The MachXO3L/LF enters User Mode immediately following the Wake-Up sequence has completed. User Mode is the point in time when the MachXO3L/LF begins performing the logic operations you designed. The MachXO3L/LF remains in this state until one of three events occurs:

- The PROGRAMN input pin is asserted
- A REFRESH command is received via one of the configuration ports
- Power is cycled

4.6. Clearing the Configuration Memory and Re-initialization

The current user mode configuration of the MachXO3L remains in operation until it is actively cleared, or power is lost. Several methods are available to clear the internal configuration memory of the MachXO3L/LF. The first is to remove power and reapply power. Another method is to toggle the PROGRAMN pin. Lastly you can reinitialize the memory through a Refresh command. Any active configuration port can be used to send a Refresh command.

- Assertion of the PROGRAMn input
- Cycling power to the MachXO3L/LF
- Sending the Refresh command using a configuration port

Invoking one of these methods causes the MachXO3L/LF to drive INITN and DONE low. The MachXO3L/LF enters the initialization state as described earlier.

4.7. Memory Space Accessibility

The two internal memories, NVCM (MachXO3L)/Flash (MachXO3LF) and SRAM, of the MachXO3L/LF have the ability to be read and written. Each port on the MachXO3L/LF has a different level of access to each memory space. Table 4.1 provides a cross-reference of the MachXO3L/LF ports and the memory space they can access.

As shown in Table 4.1, the JTAG port has the ability to read and write both of the internal memory spaces. No other port has ability to read the SRAM configuration memory. The JTAG port has the ability to access the two memory spaces in Offline or Transparent mode. Every other port has some limitation on the functions that can be performed.

Table 4.1. Memory Space Accessibility of Different Ports

Port	On-Chip NVCM/Flash		SRAM	
	Read	Write	Read	Write
JTAG	Yes	Yes	Yes	Yes
SPI Port	Yes	Yes	No	Yes
I^2C Port	Yes	Yes	No	Yes
Internal WISHBONE (MachXO3LF)	Yes*	Yes*	No	No

*__Note:__ In Transparent mode only.

4.8. On-chip Flash Programming (MachXO3LF Only)

As shown in Figure 4.1, on-chip Flash is programmed with different programming modes. These programming modes are discussed in the next sections. Within the different programming modes, there are two methods of programming the on-chip Flash: Offline and Background programming.

Offline Programming – This method requires the device to enter into programming mode. When in programming mode, the device stops working, until the programming is completed. When using Diamond Programmer, the Offline Mode is selected using operations starting with FLASH. Unless noted by the operation, the Flash sectors accessed are Feature, Configuration and UFM.

Background Programming – This method allows the device to continue operating in User Mode, while the configuration logic programs the on-chip Flash memory. When the on-chip Flash memory programming is completed, the device can download into the SRAM with REFRESH instruction. When using Diamond Programmer, the Background Mode is selected using operations starting with XFLASH. Unless noted by the operation, the Flash sectors accessed are Configuration and UFM.

Note that if background programming is used on the MachXO3LF devices with two PLLs, the system must put the right side PLL in reset state during background programming. The required duration, erase portion, of the background Flash programming time is specified in Table 98 of Using Hardened Control Functions in MachXO3 Devices Reference Guide (FPGA-TN-02064). The left side PLL can stay active during background programming.

4.9. Bitstream/PROM Sizes)

The MachXO3L/LF is an SRAM based FPGA. The SRAM configuration memory must be loaded from a non-volatile memory that can store all of the configuration data. The size of the configuration data is variable. It is based on the amount of logic available in the FPGA, and the number of pre-initialized Embedded Block RAM (EBR) components. A MachXO3L/LF design using the largest device, with every EBR pre-initialized with unique data values, and generated without compression turned on requires the largest amount of storage.

The Configuration Memory is, for most designs, large enough to store the compressed configuration data that is loaded into the SRAM configuration memory. However, as the amount of logic in the design increases, and the amount of pre-initialized EBR increases, the size of the configuration data also increases. It is possible, but unlikely, that the configuration data can get too large for the internal memory.

MachXO3LF Only; Storing configuration data in the MachXO3LF's internal Flash memory has special considerations. The Flash memory in the MachXO3LF provides three independent sectors. The first sector is dedicated for use in holding compressed configuration data, and is called Configuration Flash. The second sector, called the User Flash Memory, provides three different functions. It provides additional Configuration Flash storage for large configuration data images, it can store EBR contents, or it is available for use as general purpose Flash memory. The third sector is the Feature Row.

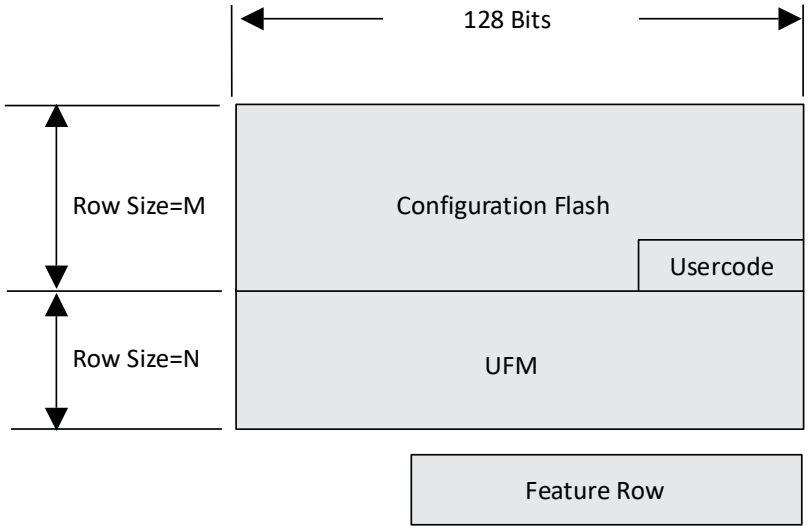


Figure 4.3. Flash Memory Space of a MachXO3LF Device

The Configuration Flash is, for most designs, large enough to store the compressed configuration data that is loaded into the SRAM configuration memory. However, as the amount of logic in the design increases, and the amount of pre-initialized EBR increases, the size of the configuration data also increases. The increase in size can cause the configuration data to overflow into the UFM sector. It is also possible, but unlikely, that the configuration data can get too large for the internal Flash memory altogether. In the event configuration data grows too large to fit in the combined Configuration Flash/UFM memory space the design needs to be modified so that it is smaller, or an external configuration memory must be used. You can provide input to the software generating the configuration data to prevent the overflow into the UFM.

In the event the configuration data is too large for the internal memory you can store the device bitstream in an external SPI Flash. Table 4.2 shows the maximum uncompressed bitstream sizes allowing you to select an SPI Flash.

Table 4.2. Maximum Configuration Bits

Device	Uncompressed Bitstream Size Without EBR	Uncompressed Bitstream Size With EBR	Maximum Internal NVCM/Flash	Units
MachXO3L/LF-640E 121 Ball Package	0.35	0.41	0.33	Mb
MachXO3L/LF-1300C 256 Ball Package	0.51	0.58	0.47	Mb
MachXO3L/LF-1300E	0.35	0.41	0.33	Mb
MachXO3L/LF-1300E 256 Ball Package	0.51	0.58	0.47	Mb
MachXO3L/LF-2100C	0.51	0.58	0.47	Mb
MachXO3L/LF-2100C 324 Ball Package	0.93	1.02	0.80	Mb
MachXO3L/LF-2100E	0.51	0.58	0.47	Mb
MachXO3L/LF-2100E 324 Ball Package	0.93	1.02	0.80	Mb
MachXO3L/LF-4300C	0.93	1.02	0.80	Mb
MachXO3L/LF-4300C 400 Ball Package	1.47	1.70	1.38	Mb
MachXO3L/LF-4300E	0.93	1.02	0.80	Mb
MachXO3L/LF-6900C	1.47	1.70	1.38	Mb
MachXO3L/LF-6900E	1.47	1.70	1.38	Mb
MachXO3L/LF-9400C	2.11	2.56	2.0	Mb
MachXO3L/LF-9400E	2.11	2.56	2.0	Mb

4.10. Feature Row

The MachXO3L/LF includes a Feature Row that is used to control FPGA resources. The Feature Row permits more flexibility in selecting the functions available for configuration, increases the number of available I/O on the device, and eliminates the need to make changes to your hardware. Feature Row can be erased or programmed independently.

MachXO3L/LF Feature Row is used to determine how the MachXO3L/LF SRAM configuration memory is loaded. In the MachXO3L, it can be programmed nine times. When Feature Row is erased, Feature Row sets its value back to HW default Mode state. Feature Row can be modified using Programming File Utility under Tools > Feature Row Editor.

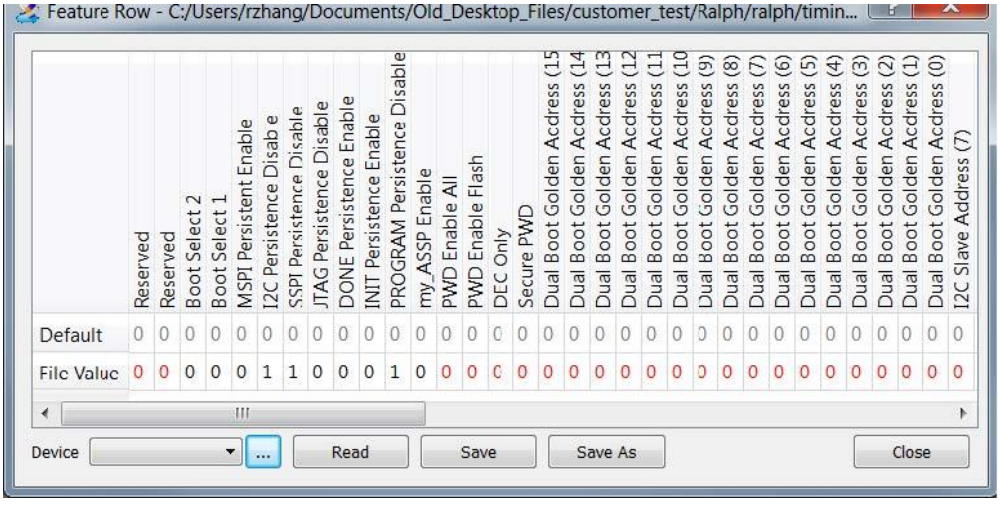


Figure 4.4. Feature Row Example

A relationship of Feature Row option and Diamond Spreadsheet View is shown in Table 4.3 and Table 4.4.

Table 4.3. Feature Row Options in Diamond Spreadsheet View

Master SPI Port	Configuration	BOOT_SEL[2:1], MSPI_Persistent_Enable
ENABLE	CFG*	101
ENABLE	EXTERNAL	011
EFB_USER	CFG*	000
DISABLE	CFG*	000

***Note:** CFG includes CFG or CFG_EBR.

A full list of the functions controlled by the Feature Row and their default values for MachXO3L are shown in Table 4.4.

Table 4.4. MachXO3L Feature Row Elements

Feature	SW Default Mode State (Programmed)	HW Default Mode State (Erased)
PROGRAMN Persistence	Disabled	Enabled
INITn Persistence	Disabled	Disabled
DONE Persistence	Disabled	Disabled
Custom IDCODE	0x00000000	0x00000000
TraceID™	00000000	00000000
Security[1]	OFF	OFF
JTAG Port Persistence	Enabled	Enabled
SSPI Port Persistence	Enabled	Enabled
I^2C Port Persistence	Disabled	Enabled
MSPI Port Persistence	Disabled	Disabled
I^2C Programmable Primary Configuration Address[2, 3]	yyyxxxxx00	1111000000
SRAM OTP	OFF	OFF
Config NVCM/Flash OTP	OFF	OFF
my_ASSP Enable	OFF	OFF
Password Enable Flash	OFF	OFF
Password Enable All	OFF	OFF

Notes:

1. Enabled/disabled using the CONFIG_SECURE preference.
2. *y* and *x* are user-programmable from IPexpress™.
3. 1111000001 is a reserved address when the device is erased.

It is strongly recommended that the Feature Row only be modified during development, and rarely, if ever, upgraded in the field. The reason for this recommendation is the Feature Row is responsible for controlling the availability of the Configuration Ports. It is possible to cause active Configuration Ports to become unavailable, preventing future updates.

Changing the Feature Row can also prevent the MachXO3L/LF from configuring. The PROGRAMN, INITN, and DONE control and status pins are enabled and disabled using the Feature Row. The PROGRAMN input pin may be recovered for use as a general purpose I/O. Erasing Feature Row state causes the PROGRAMN input to act as PROGRAMN, not as a general purpose I/O. If the general purpose I/O is driven active low, the MachXO3L is never allowed to complete its configuration process.

Feature Row can be erased or altered by Diamond Programmer under Advanced Security Keys Programming. Feature Row settings can be altered using the Diamond Spreadsheet View. Spreadsheet View allows you to edit the configuration settings for the MachXO3L/LF, and then saves your settings in the Lattice Preference File (LPF). These settings are applied to the MachXO3L/LF configuration data during the Map, Place, and Route build phases.

4.11. Key Features

- Not intended to be modified in the field; only for development.
- Change in Feature Row settings may cause active configuration ports to become unavailable.
- Can be altered using Diamond Programmer or Diamond Spreadsheet View.
- Can be altered independently under Advanced Security Keys Programming in Programmer.
- Will be erased and re-programmed during On-Chip Memory updates, so keep Feature Row contents consistent.
- MachXO3L can be programmed up to nine times.

4.12. sysCONFIG™ Ports

Table 4.5. MachXO3L/LF Programming and Configuration Ports

Interface	Port	Description
JTAG	JTAG (IEEE 1149.1 and IEEE 1532 compliant)	4-wire or 5-wire JTAG Interface
sysCONFIG	SSPI	Slave Serial Peripheral Interface (SPI)
	MSPI	Master Serial Peripheral Interface (SPI)
	I^2C	Inter-integrated Circuit (I^2C) Interface
Internal	WISHBONE Internal	WISHBONE bus interface

4.13. sysCONFIG Pins

The MachXO3L/LF provides a set of sysCONFIG I/O pins that you can use to program and configure the FPGA. The sysCONFIG pins are grouped together to create ports (such as JTAG, SSPI, I^2C, MSPI) that are used to interact with the FPGA for programming, configuration, and access of resources inside the FPGA. The sysCONFIG pins in a configuration port group may be active, and used for programming the FPGA, or they can be reconfigured to act as general purpose I/O.

Recovering the configuration port pins for use as general purpose I/O requires you to adhere to the following guidelines:

- You must DISABLE the unused port. You can accomplish this by using the Diamond Spreadsheet View's Global Preferences tab. Each configuration port is listed in the sysCONFIG options tree.
- You must prevent external logic from interfering with device programming. Make sure that recovered sysCONFIG pins are not asserted when the MachXO3L/LF is in Feature Row HW Default Mode state. One example is driving PROGRAMN with an active low signal after the MachXO3L/LF is in Feature Row HW Default Mode state. Failure to reprogram the Feature Row with PROGRAMN disabled prevents the FPGA from configuring and entering user mode.
- Ensure to selectively enable and disable the JTAG port when using JTAGENB. Any external logic connected to the JTAG I/O must not contend with the JTAG programming port.

Table 4.6 lists the default state of the shared sysCONFIG pins. As you can see, an HW Default Mode Feature Row device has the JTAG, SPI Slave and I^2C ports enabled. Upon entry to User Mode the MachXO3L, the default state of the SSPI, and I^2C sysCONFIG pins become general purpose I/O. This means you lose the ability to program the MachXO3L using I^2C when using the default sysCONFIG port settings. To retain the I^2C sysCONFIG pins in user mode, be sure to ENABLE them using the Diamond Spreadsheet View editor.

Unless specified otherwise, the sysCONFIG pins are powered by the VCCIO0 voltage. It is crucial you take this into consideration when provisioning other logic attached to Bank 0.

The function of each sysCONFIG pin is described in detail.

Table 4.6. Default State of the sysCONFIG Pins

Pin Name	Associated sysCONFIG Port	Pin Function in Feature Row Erased Mode (Configuration/HW Mode)	Pin Direction (Configuration Mode)	Default Function in User Mode (SW default Mode)
PROGRAMN	SDM	PROGRAMN	Input with weak pull up	User-defined I/O
INITN	SDM	I/O	I/O with weak pull up	User-defined I/O
DONE	SDM	I/O	I/O with weak pull up	User-defined I/O
MCLK/CCLK	SSPI/MSPI	SSPI	Input with weak pull up	SSPI
SN	SSPI/MSPI	SSPI	Input with weak pull up	SSPI
SI/SISPI	SSPI/MSPI	SSPI	Input	SSPI
SO/SPISO	SSPI/MSPI	SSPI	Output	SSPI
CSSPIN	MSPI	I/O	I/O with weak pull up	User-defined I/O
SCL	I^2C	I^2C	Bi-Directional	User-defined I/O
SDA	I^2C	I^2C	Bi-Directional	User-defined I/O

Table 4.7. Default State in Diamond for Each Port

sysCONFIG Port	Diamond Default*
SDM_PORT	Disable
SLAVE_SPI_PORT	Enable
I2C_PORT	Disable
MASTER_SPI_PORT	Disable
JTAG_PORT	Enable

*__Note__: This Default setting can be modified in the Diamond Spreadsheet View, Global Preferences tab.

4.13.1. Self-Download Port Pins

PROGRAMN

The PROGRAMN is an input used to configure the FPGA. The PROGRAMN pin, when enabled, is sensitive to a high-to-low transition, and has an internal weak pull-up. When PROGRAMN is asserted low, the FPGA exits user mode and starts a device configuration sequence at the Initialization phase, as described earlier. Holding the PROGRAMN pin low prevents the MachXO3L from leaving the Initialization phase. The PROGRAMN has a minimum pulse width assertion period in order for it to be recognized by the FPGA. You can find this minimum time in MachXO3 Family Data Sheet (FPGA-DS-02032) in the AC timing section.

Be aware of the following special cases when the PROGRAMN pin is active:

- If the device is currently being programmed via JTAG, then PROGRAMN is ignored until the JTAG mode programming sequence is complete.
- Toggling the PROGRAMN pin during device configuration interrupts the process and restart the configuration cycle.
- Asserting PROGRAMN on a device in Feature Row HW Default Mode state disables the SSPI and I^2C ports. Start SSPI or I^2C programming operations after PROGRAMN is deasserted.
- PROGRAMN is active during power-up, even when PROGRAMN has been reserved as a general purpose I/O. Do not allow any input signal attached to PROGRAMN to transition from high to low at a frequency greater than the VCC (min) to INITN rising edge time period. High to low PROGRAMN assertions more frequently prevent the MachXO3L from configuring, causing the FPGA to remain in a continuous RESET condition. See Figure 4.5.

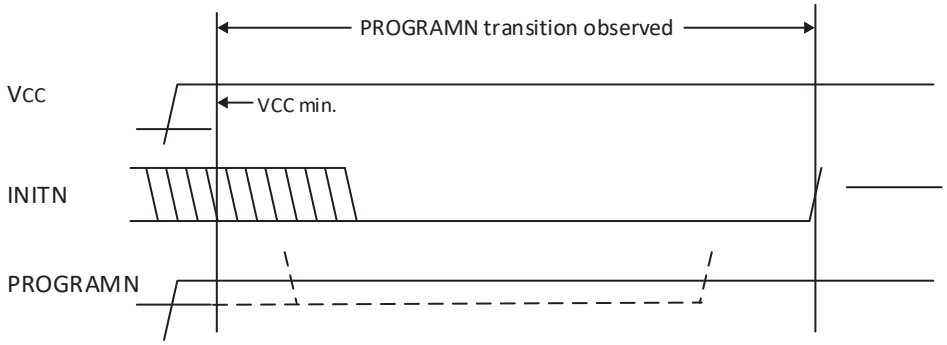


Figure 4.5. Period PROGRAMN is Always Observed

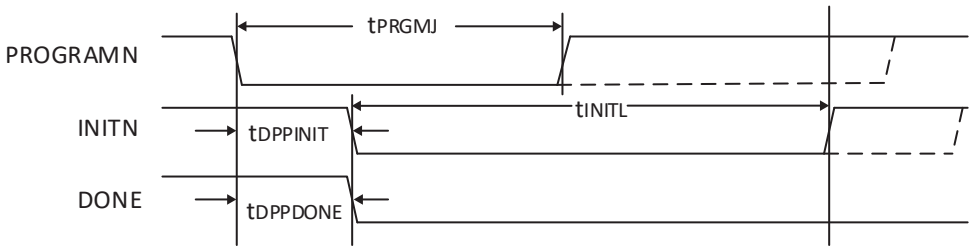


Figure 4.6. Configuration from PROGRAMN Timing

INITN

The INITn pin is a bidirectional, open-drain I/O pin only when it is enabled. It has the following functions:

- After power is applied, after a PROGRAMN assertion, or a REFRESH command it goes low to indicate the SRAM configuration memory is being erased. The low time assertion is specified with the t_{INITL} parameter.
- After the t_{INITL} time period has elapsed the INITn pin is deasserted (that is active high) to indicate the MachXO3L is ready for its configuration bits. The MachXO3L/LF begins loading configuration data from either the internal NVCM/Flash or an external SPI Flash.
- INITn can be asserted low by an external agent before the t_{INITL} time period has elapsed in order to prevent the FPGA from reading configuration bits. This is useful when there are multiple programmable devices chained together. The programmable device with the longest t_{INITL} time can hold all other devices in the chain from starting to get data until it is ready itself.
- The last function provided by INITn is to signal an error during the time configuration data is being read. Once t_{INITL} has elapsed and the INITn pin has gone high, any subsequent INITn assertion signals the MachXO3L/LF has detected an error during configuration.

The following conditions causes INITN to become active, indicating the Initialization state is active:

- Power has just been applied
- PROGRAMN falling edge occurred
- The IEEE 1532 REFRESH command has been sent using a slave configuration port (JTAG, SSPI, or I^2C).

If the INITN pin is asserted due to an error condition, the error can be cleared by correcting the configuration bitstream and forcing the FPGA into the Initialization state.

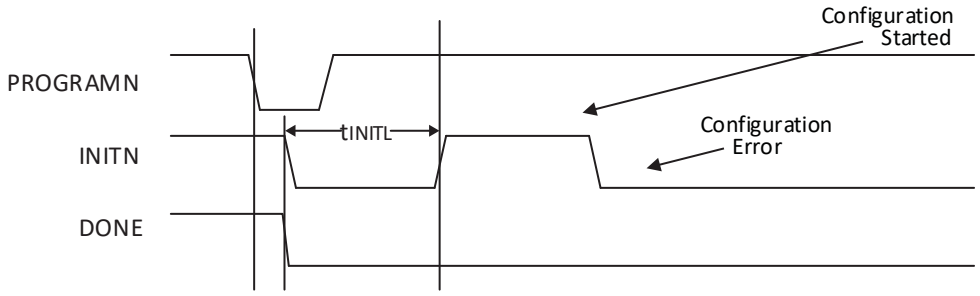


Figure 4.7. Configuration Error Notification

The INITN pin of a MachXO3L device is not visible external to the device when in the Feature Row HW Default Mode state. The INITN pin, when in this mode, is pulled high by default. The INITN behavior described in Figure 4.7 is only visible outside the MachXO3L when the INITN pin is enabled.

The INITN can be recovered as a general purpose I/O. By default, the INITN pin is disabled. You can use the Diamond Spreadsheet View to enable it.

If an error is detected when reading the bitstream, INITN goes low, the internal DONE bit is not set, the DONE pin stays low, and the device does not wake up. The device fails configuration when the following happens:

- The bitstream CRC error is detected
- The invalid command error detected
- A timeout error is encountered when loading from the on-chip NVCM/Flash
- The program done command is not received when the end of on-chip SRAM configuration or on-chip NVCM/Flash memory is reached

DONE

The DONE pin is a bidirectional open drain with a weak pull-up that signals the FPGA is in User mode.

The DONE pin drives low in tandem with the INITN pin when the FPGA enters Initialization mode. As described earlier, this condition happens when power is applied, PROGRAMN is asserted, or an IEEE 1532 Refresh command is received via an active configuration port. During POR and configuration, the DONE pin is driven low.

By default, the DONE pin is a general purpose I/O when the MachXO3L is in the Feature Row HW Default Mode state. The default mode causes the MachXO3L to automatically sequence through the Wake-Up sequence after the internal DONE bit is asserted. You have the ability to enable the DONE output pin by setting the SDM_port configuration parameter using the Diamond Spreadsheet view. The FPGA does not stall waking up waiting for the DONE pin to be asserted high.

The DONE pin has the following functions:

- Externally driving DONE pin - Easy for Daisy Chaining
  The FPGA can be held from entering User mode indefinitely by having an external agent keep the DONE pin asserted low. In order to use DONE to stall entering User mode the SDM_PORT must enable the DONE I/O, and the FPGA Feature Row must be programmed. (This feature is supported in Diamond 3.5 and later. Earlier versions of Diamond do not enable the stall feature when SDM_PORT enables DONE I/O). A common reason for keeping DONE driven low is to allow multiple FPGAs to be completely configured. As each FPGA reaches the DONE state, it is ready to begin operation. The last FPGA to configure can cause all FPGAs to start in unison.
- Monitoring DONE pin for status of FPGA configuration.
  Sampling the DONE pin is a way for an external device to tell if the FPGA has finished configuration. However, when using IEEE 1532 JTAG to configure SRAM the DONE pin is driven by a boundary scan cell, so the state of the DONE pin has no meaning during IEEE 1532 JTAG configuration (once configuration is complete, DONE takes on the behavior defined by the SDM_PORT setting in the Feature Row). The DONE pin is pulled high when the configuration is successful and you can observe this by enabling this pin in the SDM port setting in Diamond software. If the DONE pin is enabled using the SDM port setting the DONE pin is pulled high as soon as configuration is complete and this behavior can make a part appear to be successfully configured to other logic monitoring the DONE pin.

4.13.2. Master and Slave SPI Configuration Port Pins

Table 4.8. Master SPI Configuration Port Pins

Pin Name	Function	Direction	Description
MCLK/CCLK	MCLK	Output with weak pull-up	Master clock used to time data transmission/reception from the MachXO3L Configuration Logic to a slave SPI PROM. A 1k pull-up resistor is required on MCLK for External and Dual Boot configuration modes.
CSSPIN	CSSPIN	Output	Chip select used to enable an external SPI PROM containing configuration data
SI/SISPI	SISPI	Output	SISPI carries output data from the MachXO3L Configuration Logic to the slave SPI PROM
SO/SPISO	SPISO	Input	SPISO carries output data from the slave SPI PROM to the MachXO3L Configuration Logic
SN	SN/IO	Input	MachXO3L Configuration Logic slave SPI chip select input. Pull high externally whenever the MSPI port is active.

Table 4.9. Slave SPI Configuration Port Pins

Pin Name	Function	Direction	Description
MCLK/CCLK	CCLK	Input with weak pull-up	Clock used to time data transmission/reception from an external SPI master device to the MachXO3L Configuration Logic.
SI/SISPI	SI	Input	SI carries output data from the external SPI master to the MachXO3L Configuration Logic
SO/SPISO	SO	Output	SO carries output data from the MachXO3L Configuration Logic to the external SPI master
SN	SN	Input with weak pull-up	MachXO3L Configuration Logic slave SPI chip select input. SN is an active low input.

MCLK/CCLK

The MCLK/CCLK, when active, are clocks used to sequentially load the configuration data for the FPGA. The pin functions are described below.

The MCLK/CCLK pin's default state for a MachXO3L in the Feature Row HW Default Mode state is to act as the configuration clock (such as CCLK). This allows an external SPI master controller to program the MachXO3L/LF. The maximum CCLK frequency and the data setup/hold parameters can be found in the AC timing section of MachXO3 Family Data Sheet (FPGA-DS-02032). The Feature Row must be configured to ENABLE the Slave SPI Port if you want to use the port to reprogram the MachXO3L/LF after it enters user mode.

The MCLK/CCLK pin functions as a Master Clock (MCLK) when the MachXO3L/LF is configured in Dual Boot or External Boot modes. A 1k pull-up resistor is required when using these modes. The MCLK becomes an output and provides a reference clock for an SPI Flash attached to the MachXO3L/LF's Master SPI Configuration port. MCLK actively drives until all of the configuration data has been received. When the MachXO3L enters user mode the MCLK output tri-states. This allows the MCLK to become a general purpose I/O. The MCLK is reserved for use, in most post-configuration applications, as the reference clock for performing memory transactions with the external SPI PROM.

The MachXO3L/LF generates MCLK from an internal oscillator. The initial frequency of the MCLK is nominally 2.08 MHz. The MCLK frequency can be altered using the MCCLK_FREQ parameter. You can select the MCCLK_FREQ using the Diamond Spreadsheet View. For a complete list of the supported MCLK frequencies, see Table 4.10.

Table 4.10. MachXO3L/LF MCLK Valid Frequencies (MHz)

2.08	9.17	33.25
2.46	10.23	38.00
3.17	13.30	44.33
4.29	14.78	53.20
5.54	20.46	66.50
7.00	26.60	88.67
8.31	29.56	133.00

During the initial stages of device configuration, the frequency value specified using MCCLK_FREQ is loaded into the FPGA. Once the MachXO3L/LF accepts the new MCLK_FREQ value the MCLK output begins driving the selected frequency. Make sure when selecting the MCLK_FREQ that you do not exceed the frequency specification of your configuration memory, or of your PCB. Review the MachXO3L AC specifications in MachXO3 Family Data Sheet (FPGA-DS-02032) when making MCLK_FREQ decisions.

SN

The SN pin is the Slave SPI ports chip select. An external SPI bus master asserts the SN pin active low in order to perform actions using the MachXO3L's programming and configuration logic. The SN pin is available when the MachXO3L is in the Feature Row HW Default Mode state, and in user mode when the Slave SPI port is set to the ENABLE setting. The SN pin is a general purpose I/O in user mode when the Slave SPI port is set to the DISABLE setting.

Proper operation of the MachXO3L/LF depends upon maintaining the SN pin in the correct state:

- SN must be deasserted (i.e. held high) when configuring using Master SPI mode
- SN must be deasserted when the MachXO3L/LF is in user mode
- SN must be deasserted when accessing the Configuration Logic in the MachXO3L/LF using I^2C
- When SN is asserted, CSSPIN must be deasserted. Deasserting CSSPIN places the shared SPI pins into a high impedance state.
    - The Master SPI port and the Slave SPI port share three common pins, SI/SISPI, SO/SPISO, and MCLK/CCLK. The MachXO3L/LF permits both ports to be available at the same time. They are not permitted to be accessed at the same time. The Slave SPI and the Master SPI port must be time multiplexed when both ports are enabled.

Lattice recommends the SN pin be pulled high externally to augment the weak internal pull-up.

CSSPIN

The CSSPIN pin is an active low chip select used by the Master SPI configuration mode to enable an external SPI Flash. When the MachXO3L/LF is programmed to configure in either External or Dual Boot mode the CSSPIN pin is asserted to the attached SPI Flash. The MachXO3L/LF asserts CSSPIN until all configuration data bytes have been loaded, at which time the CSSPIN enters a high impedance state.

When the MachXO3L/LF is in the Feature Row HW Default Mode state the CSSPIN is a general purpose I/O with a weak pulldown. It must have an external pull-up resistor when the External and Dual Boot configuration modes are used. CSSPIN must ramp in tandem with the SPI PROM VCC input. It remains a general purpose I/O when the FPGA enters user mode. You must ENABLE the Master SPI port to reserve CSSPIN for use by the internal SPI Master logic.

When configuring from an external SPI Flash, ensure that the SPI Flash VCC and the MachXO3L/LF VCCIO2 are at the same level. Ensure that the SPI Flash VCC is at the recommended operating level.

Some SPI PROM manufacturers require the chip select input of the PROM ramp in unison to the PROMs VCC rail. The CSSPIN pin, by default, has a weak pull-down resistor internally. Adding a 4.7 kΩ to 10 kΩ pull-up resistor to the CSSPIN pin on the MachXO3L is recommended.

SI/SISPI

The SI/SISPI is a dual function bidirectional pin. The direction depends upon whether a Master or Slave mode is active. The SI/SISPI is an input data pin when using the Slave SPI mode and is an output data pin when using the Master SPI mode. In Master SPI mode, the MachXO3L/LF drives SI/SISPI until all configuration data bytes have been loaded, at which time the SI/SISPI enters a high impedance state.

At least one of the sysCONFIG preferences, SLAVE_SPI_PORT or MASTER_SPI_PORT, must be set to ENABLE in order to preserve this pin as SI/SISPI and allow access to the SPI interface.

SO/SPISO

The SO/SPISO pin is a dual function bidirectional pin. The direction depends upon whether a Master or Slave mode is active. The SO/SPISO is an input data pin when using the Master SPI mode and is an output data pin when using the Slave SPI mode.

At least one of the sysCONFIG preferences, SLAVE_SPI_PORT or MASTER_SPI_PORT, must be set to ENABLE in order to preserve this pin as SO/SPISO and allow access to the SPI interface.

4.13.3. I²C Configuration Port Pins

SCL

The MachXO3L provides an I²C configuration port. The SCL is the I²C Serial Clock pin, and is used to initiate and time transactions on the I²C bus. It is a bidirectional, open-drain signal that is an output when the MachXO3L I²C controller is mastering transactions on the bus, and is an input when an external I²C master is accessing resources inside the MachXO3L/LF. SCL requires an external pull-up resistor in order to operate.

The SCL pin is available when the MachXO3L/LF is in the Feature Row HW Default Mode state. You must ENABLE the I2C_PORT and instantiate the EFB for the I²C port to continue to be available in user mode (see the I2C Configuration Mode section for details). The SCL pin becomes a general purpose I/O if you do not ENABLE the I2C_PORT.

SDA

The SDA pin is the I²C serial data input/output pin. It is bidirectional, open-drain, and requires an external pull-up resistor in order to operate. The pin changes direction dynamically during data transactions on the I²C bus. The current state depends on the current bus master and the operation being performed by that master.

The SDA pin is available when the MachXO3L/LF is in the Feature Row HW Default Mode state. You must ENABLE the I2C_PORT and instantiate the EFB if you want the I²C port to continue to be available in user mode (see the I2C Configuration Mode section for details). The SDA pin becomes a general purpose I/O if you do not ENABLE the I2C_PORT.

4.13.4. JTAG Configuration Port Pins

The JTAG pins provide a standard IEEE 1149.1 Test Access Port (TAP). The JTAG port is the only configuration port on the MachXO3L that is capable of performing configuration, programming, and multi-device configuration functions. Programming and configuration over the JTAG port uses IEEE 1532 compliant commands. In addition to the IEEE 1532 capabilities, the MachXO3L/LF provides all of the mandatory IEEE 1149.1 Test Access Port commands allowing printed circuit board assembly verification.

The JTAG port is enabled by default when the MachXO3L/LF is in the Feature Row HW Default Mode state. Like all of the other configuration port pins, the JTAG pins can become general purpose I/O. Unlike the other ports, the default state for the JTAG port is to remain active in user mode (i.e. ENABLE state). The JTAG pins can be recovered to be general purpose I/O by setting the JTAG_PORT preference to the DISABLE state. It is recommended the JTAG port remain dedicated programming pins.

The JTAG port, when set in the DISABLE state, enables the JTAGENB input. JTAGENB permits the JTAG pins to be multiplexed. Asserting JTAGENB high causes the JTAG pins to take on the IEEE 1149.1 personality. Deasserting JTAGENB (that is driven low) causes the JTAG port pins to become general purpose I/O. Design the JTAG port circuitry carefully when taking advantage of JTAG port pin multiplexing. Avoid bus contention between logic attached to the JTAG port.

When the device is programmed through IEEE 1149.1 control, the sysCONFIG programming pins, such as DONE, cannot be used to determine programming progress. This is because the state of the boundary scan cell drives the pin, per the IEEE JTAG standard, rather than normal internal logic.

Table 4.11. JTAG Port Pins

Pin Name	Pin Function (Configuration Mode)	Pin Direction (Configuration Mode)	Default Function (User Mode)
TDI	TDI	Input with weak pull-up	TDI
TDO	TDO	Output with weak pull-up	TDO
TCK	TCK	Input	TCK
TMS	TMS	Input with weak pull-up	TMS
JTAGENB	I/O	Input/output with weak pull-down	I/O

TDO

The Test Data Output (TDO) pin is used to shift out serial test instructions and data. When TDO is not being driven by the internal circuitry, the pin is in a high impedance state. The only time TDO is not in a high impedance state is when the JTAG state machine is in the Shift IR or Shift DR state. This pin should be wired to TDO of the JTAG connector, or to TDI of a downstream device in a JTAG chain. An internal pull-up resistor on the TDO pin is provided. The internal resistor is pulled up to VCCIO Bank 0.

TDI

The Test Data Input (TDI) pin is used to shift in serial test instructions and data. This pin should be wired to TDI of the JTAG connector, or to TDO of an upstream device in a JTAG chain. An internal pull-up resistor on the TDI pin is provided. The internal resistor is pulled up to VCCIO of Bank 0.

TMS

The Test Mode Select (TMS) pin is an input pin that controls the progression through the 1149.1 compliant state machine states. The TMS pin is sampled on the rising edge of TCK. The JTAG state machine remains in or transitions to a new TAP state depending on the current state of the TAP, and the present state of the TMS input. An internal pull-up resistor is present on TMS per the JTAG specification. The internal resistor is pulled to the VCCIO of Bank 0.

TCK

The test clock pin (TCK) provides the clock used to time the other JTAG port pins. Data is shifted into the instruction or data registers on the rising edge of TCK and shifted out on the falling edge of TCK. The TAP is a static design permitting TCK to be stopped in either the high or low state. The maximum input frequency for TCK is specified in the DC and Switching Characteristics section of MachXO3 Family Data Sheet (FPGA-DS-02032). The TCK pin does not have a pull-up. An external pull-down resistor of 4.7 kΩ is recommended to avoid inadvertently clocking the TAP controller as power is applied to the MachXO3L/LF.

JTAGENB

The JTAG ENABLE pin, also known as the IEEE 1149.1 conformance pin, is an input pin that can be used to multiplex the JTAG port. The JTAGENB pin is only active in user mode. The JTAGENB pin is a user I/O while the JTAG port is in the ENABLE state. Figure 4.8 shows the default behavior of the JTAG port of a MachXO3L device.

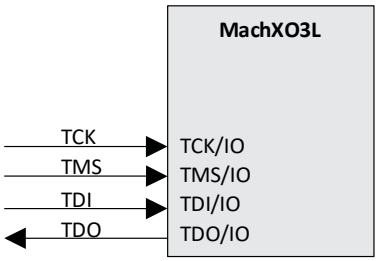


Figure 4.8. Default JTAG Port with JTAG_PORT = ENABLE

The JTAG port can become general purpose I/O. By setting the JTAG_PORT preference in the Diamond Spreadsheet View to the DISABLE state. When the JTAG port is in the DISABLE state the JTAGENB pin becomes a dedicated input. Driving the JTAGENB low disables the JTAG port and the four JTAG pins become general purpose I/O. Driving the JTAGENB input high enables the JTAG port. Figure 4.9 shows JTAG port behavior under the control of the JTAGENB.

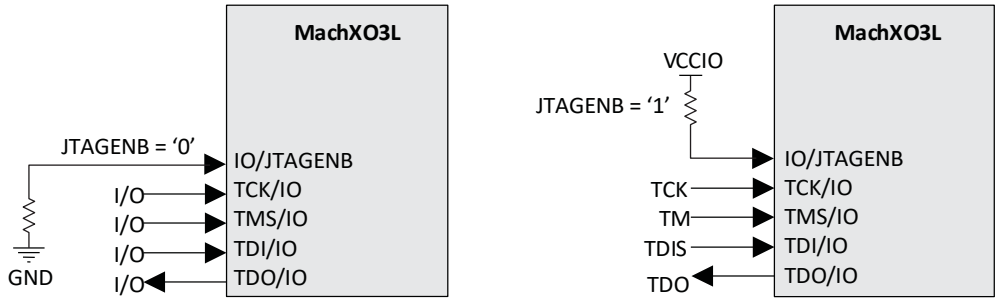


Figure 4.9. JTAG Port Behavior with JTAG_PORT = DISABLE

It is critical when using the JTAGENB feature that logic attached to the JTAG I/O pins not contend with a JTAG programming system. The external logic must ignore any JTAG transactions performed by an external programming system.

Lattice parallel port or USB download cables provide an output called ispEN. The ispEN signal can be attached to the JTAGENB input to control the availability of the JTAG port. An alternate mechanism to control the JTAGENB input is to use a shunt that can be installed or removed as required.

5. Configuration Modes

The MachXO3L provides multiple options for loading the configuration SRAM from a non-volatile memory. The previous section described the physical interface necessary to interact with the MachXO3L configuration logic. This section focuses on describing the functionality of each of the different configuration modes. Descriptions of important settings required in the Diamond Spreadsheet View are also discussed.

5.1. SDM Mode

The advantages of Self Download Configuration Mode include:

- Speed – The MachXO3L is ready to run in a few milliseconds depending on the density of the device.
- Security – The configuration data is never seen outside the device during the load to SRAM. You can prevent the internal memory from being read.
- Reduced cost – There is no need to purchase a PROM specifically reserved for programming the MachXO3L.
- Reduced board space – Elimination of an external PROM allows your board to be smaller.

The MachXO3L/LF retrieves the configuration data from the internal NVCM/Flash when it is using Self-Download Mode. SDM is triggered when power is applied, a REFRESH command is received, or by asserting the PROGRAMN pin. Self-Download Mode cannot be used when the Configuration Memory overflow occurs. Master SPI Configuration Mode must be used in the event of the Memory overflow.

5.2. Master SPI Configuration Mode (MSPI)

Master SPI Configuration Mode is the only other self-controlled configuration mode available to the MachXO3L/LF. When the MachXO3L/LF has the Master SPI Configuration mode (MSPI) enabled it is able to automatically retrieve the configuration data from an externally attached SPI Flash. The MSPI configuration port is not available when the MachXO3L/LF is in the Feature Row HW Default Mode state. Lattice recommends having a secondary configuration port available, one that is active when the MachXO3L/LF is in Feature Row HW Default Mode state, that allows you to recover the MachXO3L/LF in the event of a programming error.

To ensure that the MachXO3L/LF operates correctly using the MSPI configuration mode, make sure that:

- The POR of the SPI Flash device is lower than the POR of the MachXO3L/LF or the SPI Flash must be powered first.
- SPI Flash F_{max} is greater than the MachXO3L/LF MCLK F_{max}
- Board routing requirements to ensure the MachXO3L/LF setup and hold time parameters are met. Refer to MachXO3 Family Data Sheet (FPGA-DS-02032) for detailed setup and hold time information.

If the SPI Flash POR is higher than the MachXO3L/LF POR and has a slow ramp, here is what happens:

1. MachXO3L/LF powers up.
2. MachXO3L/LF begins toggling MCLK.
3. The preamble from the SPI Flash does not return because its POR level is not met.
4. MachXO3L/LF times out because it fails to get the preamble in time and the boot up likewise fails.

It is highly recommended that an SPI Flash be chosen which POR level is lower than the MachXO3L/LF POR. If one is not available here are some workaround solutions:

- Processor to hold INITN
- Processor to hold PROGRAMN
- RC delay to INITN

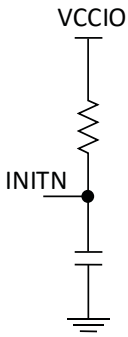


Figure 5.1. RC Delay

Table 5.1. Master SPI Port Pins

Pin Name	Function
MCLK	Clock output from the MachXO3L Configuration Logic and Master SPI controller. Connect MCLK to the SCLK input of the Slave SPI device.
SISPI	Serial Data output from the MachXO3L to the slave SPI SI input.
SPISO	Serial Data input to the MachXO3L configuration logic from the slave SPI SO output.
CSSPIN	Chip select output from the MachXO3L configuration logic to the slave SPI Flash holding configuration data for the MachXO3L.

Table 4.2 provides information about the amount of memory needed for MachXO3L/LF configuration data by device density. Select an SPI Flash that accepts 03 hex Read Opcodes. The MachXO3L/LF is only able to use the 03 hex Read Opcode.

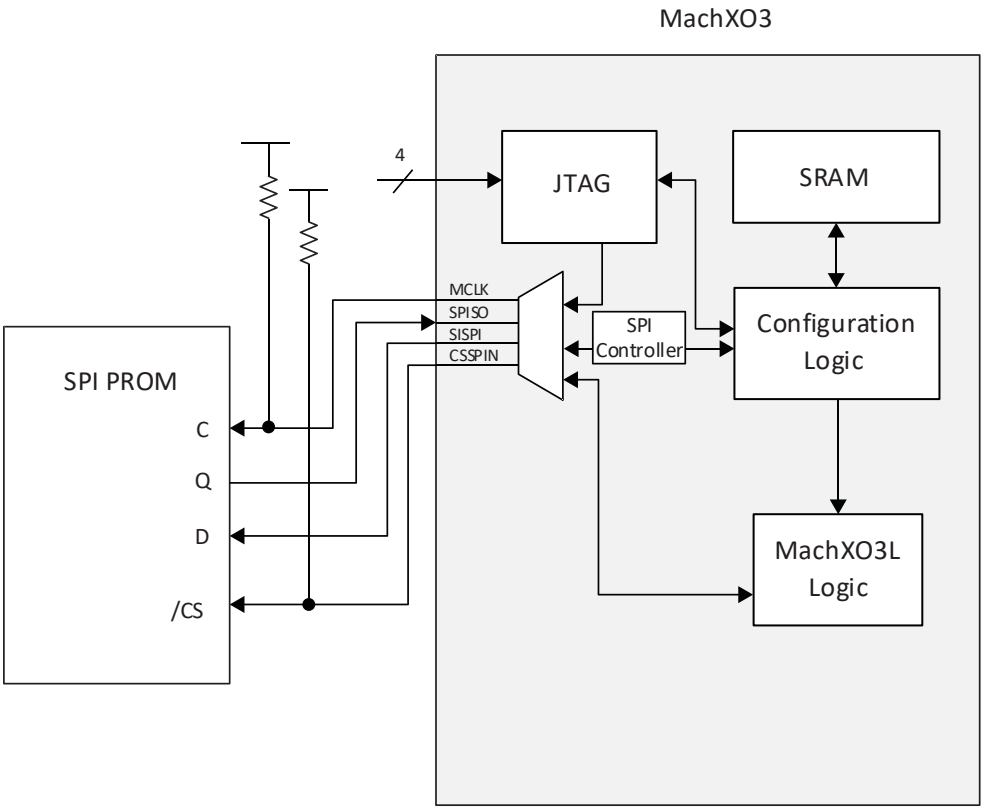


Figure 5.2. Master SPI Configuration Mode

The MachXO3L/LF begins retrieving configuration data from the SPI Flash when power is applied, a REFRESH command is received, or the PROGRAMN pin is asserted and released. The MCLK/CCLK I/O takes on the Master Clock (MCLK) function, and begins driving a nominal 2.08 MHz clock to the SPI Flash's SCLK input. CSSPIN is asserted low, commands are transmitted to the PROM over the SI/SISPI output, and data is read from the PROM on the SO/SPISO input pin. When all of the configuration data is retrieved from the PROM the CSSPIN pin is deasserted, and the MSPI output pins are tri-stated.

The MCLK frequency always starts downloading the configuration data at the nominal 2.08 MHz frequency. The MCCLK_FREQ parameter, accessed using Spreadsheet View, can be used to increase the configuration frequency. The configuration data in the PROM has some padding bits, and then the data altering the MCLK base frequency is read. The MachXO3L reads the remaining configuration data bytes using the new MCLK frequency.

After the MachXO3L/LF enters user mode the Master SPI configuration port pins tri-state. This allows data transfers across the SPI. There are two primary methods available for transferring data across the SPI bus. The first method available to you is to enable the Embedded Function Block (EFB) in the MachXO3L/LF. Using IPexpress™, you instantiate the EFB and you choose the features you want active. One of the features available in the EFB is an SPI Master Controller. The SPI Master Controller in the EFB attaches directly Master SPI configuration port pins. The controller provides a set of status, control, and data registers for initiating SPI bus transactions.

The second way to perform Master SPI configuration port transactions is to master them from the JTAG port. The MachXO3L/LF includes a JTAG to MSPI passthru circuit that allows the slave SPI Flash to be erased, programmed, and read. The primary method for programming the attached SPI Flash is to use Diamond Programmer to transfer a configuration data file from your personal computer. This is useful during board development and debug.

Note: To support JTAG to MSPI passthru programming mode, a 1kΩ pull-up resister is required on MCLK.

Another way to program an SPI Flash using the JTAG port is to use the Lattice ispVME solution. ispVME is C code written for an embedded microprocessor. The microprocessor reads a data file crafted by the Diamond Deployment Tool, and runs the ispVME code. The firmware uses port I/O to drive the JTAG port of the MachXO3L/LF, which in turn passes the data to the Master SPI port. Refer to the ispVME tool suite for information about updating an attached SPI Flash using a microprocessor.

To set the MachXO3L/LF for operation using the MSPI configuration mode you must:

- Store the entire configuration data in an external SPI Flash.
- The data must start at offset 0x000000 within the PROM.
- Set the preferences as shown in Table 5.2.

Table 5.2. Master SPI Configuration Software Settings

Preference	Setting
MASTER_SPI_PORT	ENABLE
CONFIGURATION	EXTERNAL

The BIT file must be programmed into the external SPI Flash. There are several ways to get the data into the SPI Flash:

- Diamond Programmer can transmit the SPI Flash data using a JTAG download cable
- A microprocessor running ispVME
- Automatic Test Equipment can program the SPI Flash using JTAG
- Pre-programmed SPI Flash memories can be pre-assembled onto your printed-circuit board

Once the SPI Flash contains your configuration data, you can test the configuration. Assert the PROGRAMN, transmit a REFRESH command, or cycle power to the board, and the MachXO3L/LF configures from the external SPI Flash.

5.3. Dual Boot Configuration Mode

Dual Boot Configuration Mode is a combination of Self Download Mode and Master SPI Configuration Mode. The MachXO3L, when set up in Dual Boot Mode, tries by default to configure first from external flash PROM using MSPI mode. If the configuration fails, the MachXO3L attempts to configure itself from the internal NVCM using SDM. The preset load order may be reversed if desired. The MachXO3LF, when set up in Dual Boot Mode, tries by default to configure first from the internal Flash memory using SDM. If the SDM configuration fails, the MachXO3LF attempts to configure itself using MSPI mode. The preset load order may be reversed if desired using the DUALBOOTGOLDEN configuration option in the Diamond software spreadsheet view.

Dual Boot Configuration Mode can be utilized in conjunction with the MachXO3LF Soft Error Detection (SED) feature without restriction. However, Soft Error Correction (SEC) use is limited to the primary image only. Refer to MachXO3 Soft Error Detection/Correction Usage Guide (FPGA-TN-02062) for more information on the use of the SED and SEC features.

The first boot attempt is from the primary configuration image. If the primary configuration fails, the second boot attempt is from the golden/failsafe configuration image. The primary image can fail in one of two ways:

- A bitstream CRC error is detected
- A time-out error is encountered when loading

A CRC error is caused by incorrect or corrupt data. Data is read from the primary image in rows. As each row enters the Configuration Engine the data is checked for CRC consistency. Before the data enters the Configuration SRAM the CRC must be correct. Any incorrect CRC causes the device to erase the Configuration SRAM and retrieve configuration data from the golden/failsafe image location.

It is possible for the data to be correct from a CRC calculation perspective, but not be functionally correct. In this instance, the internal DONE bit is never active. The MachXO3L/LF counts the number of master clock pulses it has provided after the Power On Reset signal was released. When the count expires without DONE becoming active the FPGA attempts to get its configuration data from the golden/failsafe image location.

Dual boot configuration mode typically requires two configuration data files. One of the two configuration data files is a fail-safe image that is rarely, if ever, updated. The second configuration data file is a working image that is routinely updated. For MachXO3L, the failsafe image is stored in internal NVCM memory while the primary working image is stored in the external SPI memory. It is the opposite for MachXO3LF: The failsafe image is stored in the external SPI memory while the primary working image is stored in internal flash memory. One Diamond project can be used to create both the working and the fail-safe configuration data files. Configure the Diamond project with an implementation named *working*, and an implementation named *failsafe*. Read the Diamond Online Help for more information about using Diamond implementations.

Use the following preferences to build a dual-boot design:

Table 5.3. Dual Boot Configuration Software Settings

Preference	Setting
CONFIGURATION	CFG
MASTER_SPI_PORT	ENABLE
DUALBOOTGOLDEN	INTERNAL/EXTERNAL
COMPRESS_CONFIG	ON | OFF

In the Diamond flow, JEDEC file option should be selected when generating configuration data that is stored in the internal NVCM/Flash. The bitstream file option should be selected when generating configuration data that is stored in the external SPI flash. For dual boot applications, the configuration data must be located in the external SPI Flash starting at address 0x010000. This differs from a single image Master SPI Configuration Mode, which requires the configuration data to be stored at offset 0x000000.

To prevent the MachXO3L/LF from using dual boot mode when using the User Master SPI controller, set the MASTER_SPI_PORT preference to EFB_USER. This reserves the Master SPI configuration port pins and prevents dual-boot.

5.4. Slave SPI Mode (SSPI)

The MachXO3L/LF provides a Slave SPI configuration port that allows you to access features provided by the Configuration Logic. You can reprogram the SRAM, NVCM/Flash and Feature Row, and access status/control registers within the Configuration Logic block. Reprogramming the NVCM/Flash can be done using offline or transparent operations.

Table 5.4. Slave SPI Port Pins

Pin Name	Description
CCLK	Configuration clock input that is driven by an SPI master controller.
SI	Serial Data Input to the MachXO3L/LF Configuration Logic for command and data.
SO	Serial Data Output from the MachXO3L/LF configuration logic.
SN	Chip select to enable the MachXO3L/LF configuration logic.

In the Slave SPI mode, the MCLK/CCLK pin becomes CCLK (that is Configuration clock). Input data is read into the MachXO3L/LF device on the SI pin at the rising edge of CCLK. Output data is valid on the SO pin at the falling edge of CCLK. The SN acts as the chip select signal. When SN is high, the SSPI interface is deselected and the SO/SPISO pin is tri-stated. Commands can be written into and data read from the MachXO3L/LF when SN is asserted. The MachXO3L/LF SSPI port only accepts Mode 0 bus transactions to the Configuration Logic.

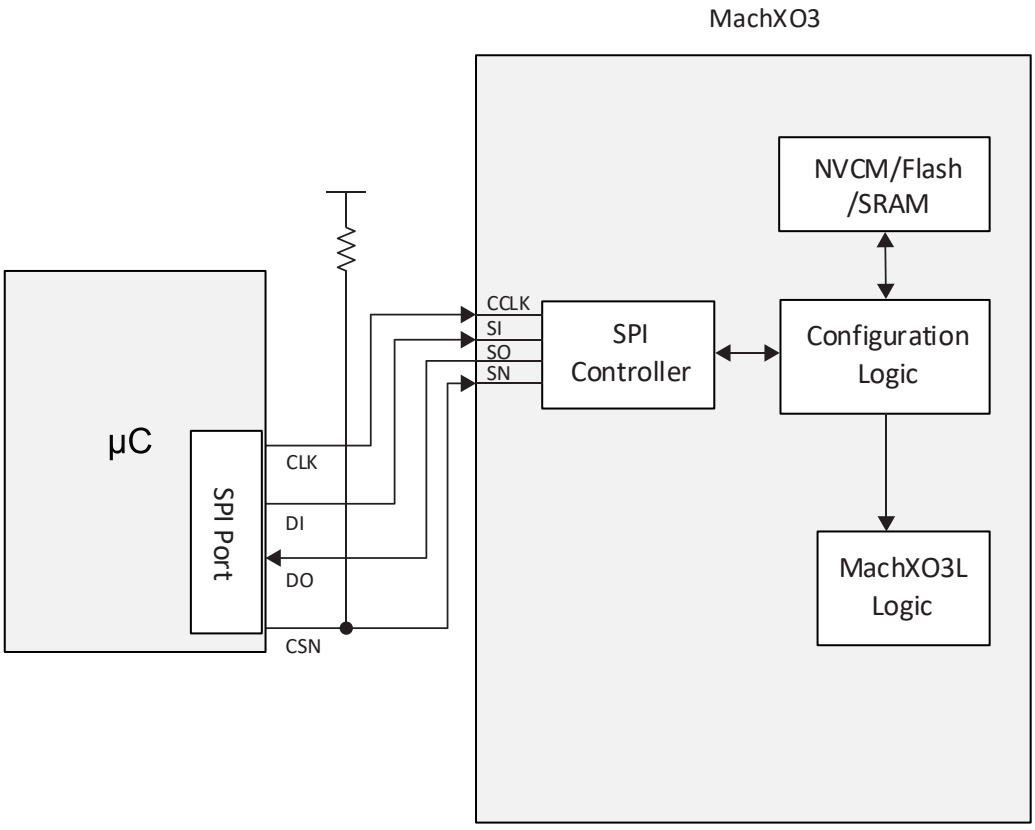


Figure 5.3. Slave SPI Configuration Mode

The SSPI port is active when the MachXO3L/LF is in Feature Row HW Default Mode state. Diamond's default preference for the SLAVE_SPI_PORT is to ENABLE the port. Use the Spreadsheet View to DISABLE the SLAVE_SPI_PORT preference in your design to keep the SSPI port to be used as general purpose I/O in user mode. Lattice recommends you keep a secondary programming port active in the event the SSPI port is accidentally disabled.

The SSPI port is used to erase, program, and verify the Configuration Flash, User Flash Memory, and the Feature Row. It is capable of directly accessing the configuration SRAM. To prevent unintentional erasure of the Feature Row, it is recommended the SSPI port be used to perform transparent updates of the Flash memory. The SSPI port can issue a REFRESH command to make a newly programmed image active. The REFRESH command can be safely used when the MachXO3 is using External or Dual Boot configuration mode because the REFRESH operation does not begin until SN is deasserted.

Programming the MachXO3L/LF using the SSPI port is complex. Lattice provides *C* source code called SSPIEMbedded to insulate you from the complexity of programming the MachXO3L/LF. It is recommended that SSPIEmbedded be used when you want to reprogram the MachXO3L/LF NVCM (MachXO3L)/Flash (MachXO3LF) or SRAM.

Accessing the status registers is less complex and does not require the use of the SSPIEmbedded code.

5.5. I²C Configuration Mode

The MachXO3L/LF has an I²C Configuration port for use in accessing the configuration logic. An I²C master can communicate to the configuration logic using 10-bit or 7-bit addressing modes. The I²C SCL input can accept a clock frequency up to 400 kHz. You can reprogram the SRAM, NVCM/Flash and Feature Row, and access status/control registers within the configuration logic block. Reprogramming the NVCM/Flash can be done in offline or in transparent operations.

Table 5.5. I²C Port Pins

Pin Name	Description
SCL	I²C bus clock
SDA	I²C bus data line

The I²C Configuration port is available when the MachXO3L/LF is in Feature Row erased state. The default state set for the I2C_PORT in the Diamond design software is to place the I2C_PORT in the DISABLE state. You must make sure the I2C_PORT is set to the ENABLE state to leave the I²C interface active in user mode. Lattice recommends making a second configuration port available (for example, JTAG) in order to recover from erroneously disabling the I²C port.

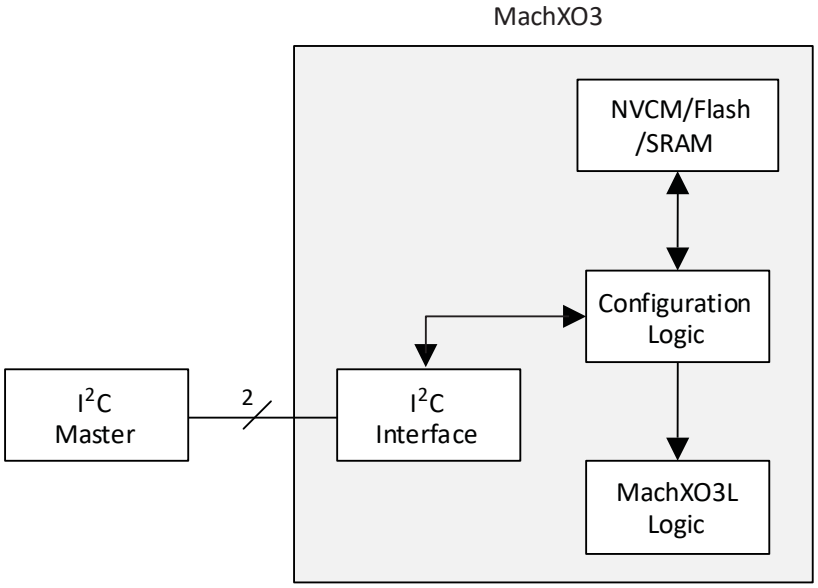


Figure 5.4. I²C Configuration Logic

There are two hardened I²C controllers in a MachXO3L/LF device, a primary and a secondary. The primary controller provides an interface to the MachXO3L/LF Configuration Logic. The primary I²C controller is the only one that permits access to the Configuration Logic OR can be a User Mode I²C controller. The Secondary I²C controller is always a User Mode I²C controller.

When the MachXO3L/LF is in Feature Row HW Default Mode state the I²C port is enabled, and you may interact with the primary I²C controller. Whenever the I²C port is enabled access to the Configuration Logic is possible. It is necessary to instantiate the Embedded Function Block (EFB) to preserve access to the Configuration Logic in User Mode. Moreover, when instantiated, the EFB *wb_clk_i* input must be connected to a valid clock source of at least 7.5x the I²C bus rate (for example, >3.0 MHz when I²C rate = 400 kHz).

An external I²C master accesses the Configuration Logic using address 1000000 (7-bit mode) or 1111000000 (10-bit mode) unless the EFB I²C base address has been modified. Use IPexpress, not Spreadsheet View, to modify the address to which the Primary and Secondary I²C controllers respond. It is necessary to instantiate the EFB in order to change the address. The address is shared by the Primary and Secondary I²C controllers.

Table 5.6 shows the address decoding used to access the I²C resources in the MachXO3L/LF.

Table 5.6. Slave Addresses for I²C Ports

Slave Address	I²C Function
yyyxxxxx00	Primary I²C Controller Configuration Logic address. Always responds to 7-bit or 10-bit addresses.
yyyxxxxx01	User Mode Primary I²C Controller address.
yyyxxxxx10	User Mode Secondary I²C Controller address.
yyyxxxxx11	Primary I²C Configuration Logic Reset. Always responds to 7-bit or 10-bit addresses.

Note: The Slave I²C addresses will ACK under all circumstances even if not all ports are configured to be active by the user.

The fourth I²C resource in the MachXO3L/LF is located at offset 3. In some instances, an I²C memory transaction to the configuration logic may be interrupted or abandoned. It is possible for a command to be accepted by the configuration logic that causes the configuration logic to respond with data. In the event that the I²C memory transaction is interrupted or abandoned, the configuration logic continues to return the queued data. New incoming I²C commands may be considered padding bytes or may be misinterpreted. Clear this condition by writing any value to offset 3. The configuration logic command interpreter resets, any queued data is flushed, and subsequent I²C memory transactions to the Configuration Logic operates correctly.

6. WISHBONE Configuration Mode (MachXO3LF Only)

The MachXO3LF can access the Configuration Flash, User Flash Memory, and the Feature Row from an internal WISHBONE bus. To use the WISHBONE, bus the Embedded Function Block must be inserted into your design. You design logic to interface to the EFB and then perform WISHBONE bus transactions to access resources attached to the configuration logic.

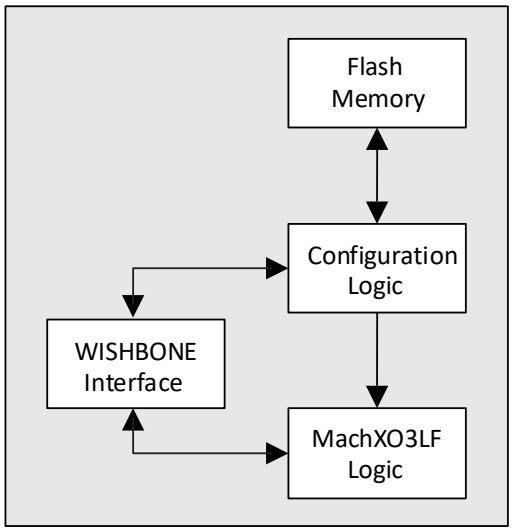


Figure 6.1. WISHBONE Configuration Mode

The MachXO3LF must be in user mode in order to access the WISHBONE interface. Accessing and updating the resources made available by the configuration logic must be done in Transparent mode. Attempting accesses to the configuration logic in offline mode causes a deadlock because the MachXO3LF leaves user mode.

You can get more detailed information about the MachXO3LF WISHBONE interface by reading Using Hardened Control Functions in MachXO3 Devices (FPGA-TN-02063).

6.1. JTAG Mode

The JTAG port is the most flexible configuration and programming port available on the MachXO3L/LF. The JTAG provides:

- Offline NVCM/Flash programming
- Transparent Flash memory programming (MachXO3LF)
- Offline SRAM configuration
- Full access to the MachXO3L/LF Configuration Logic
- Device chaining
- IEEE 1149.1 testability
- IEEE 1532 Compliant programming

The JTAG port is available when the MachXO3L/LF is in Feature Row HW Default Mode state. The MachXO3L/LF JTAG port pins are not dedicated to performing the IEEE 1149.1 TAP function. The JTAG port may be recovered for use as general purpose I/O or vice versa. See the sysCONFIG Pins section for details.

The MachXO3L/LF JTAG port is a valuable asset due to its flexibility. It provides the best capabilities for system and device debug. Lattice recommends the JTAG port remain accessible in every MachXO3L/LF design. Advantages for keeping the JTAG port active include:

Multi-Chain Architectures

The JTAG port is the only configuration and programming port that permits the MachXO3L/LF to be combined in a chain of other programmable logic.

Reveal Debug

The Lattice Reveal debug tool is an embed-able logic analyzer tool. It allows you to analyze the logic inside the MachXO3L/LF in the same fashion as an external logic analyzer permits analysis of board level logic. Reveal access is only available via the MachXO3L/LF JTAG port.

SRAM Readback

The JTAG port is the only sysCONFIG port able to directly access the MachXO3L/LF's configuration SRAM.

Boundary Scan Testability

Board level connectivity testing performed using IEEE 1149.1 JTAG is a key capability for assuring the quality of assembled printed-circuit-boards. Preserving the MachXO3L/LF JTAG port is vital for boundary scan testability. Lattice provides Boundary Scan Description Language files for the MachXO3L/LF on the Lattice website.

6.2. TransFR Operation

The MachXO3L/LF, like other Lattice FPGAs, provides for the TransFR™ capability. TransFR is described in Minimizing System Interruption During Configuration Using TransFR Technology (TN1087). The following is an example of how you can update bitstream in MachXO3L by using the TransFR feature.

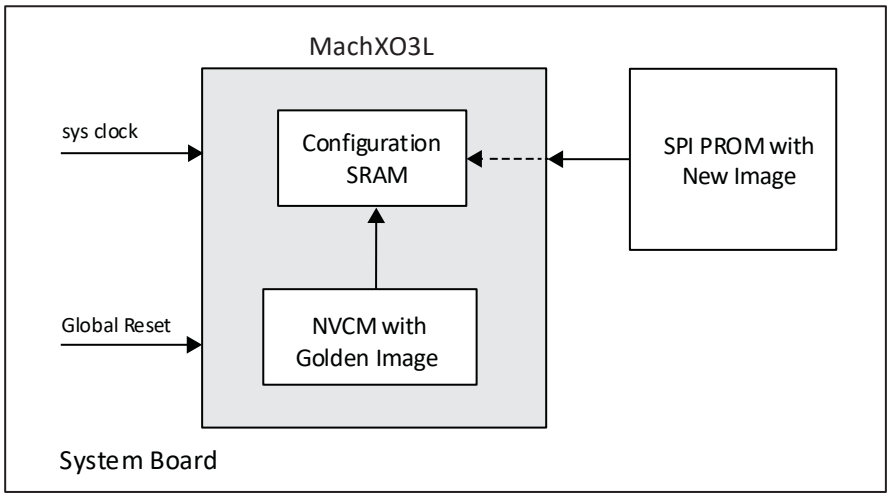
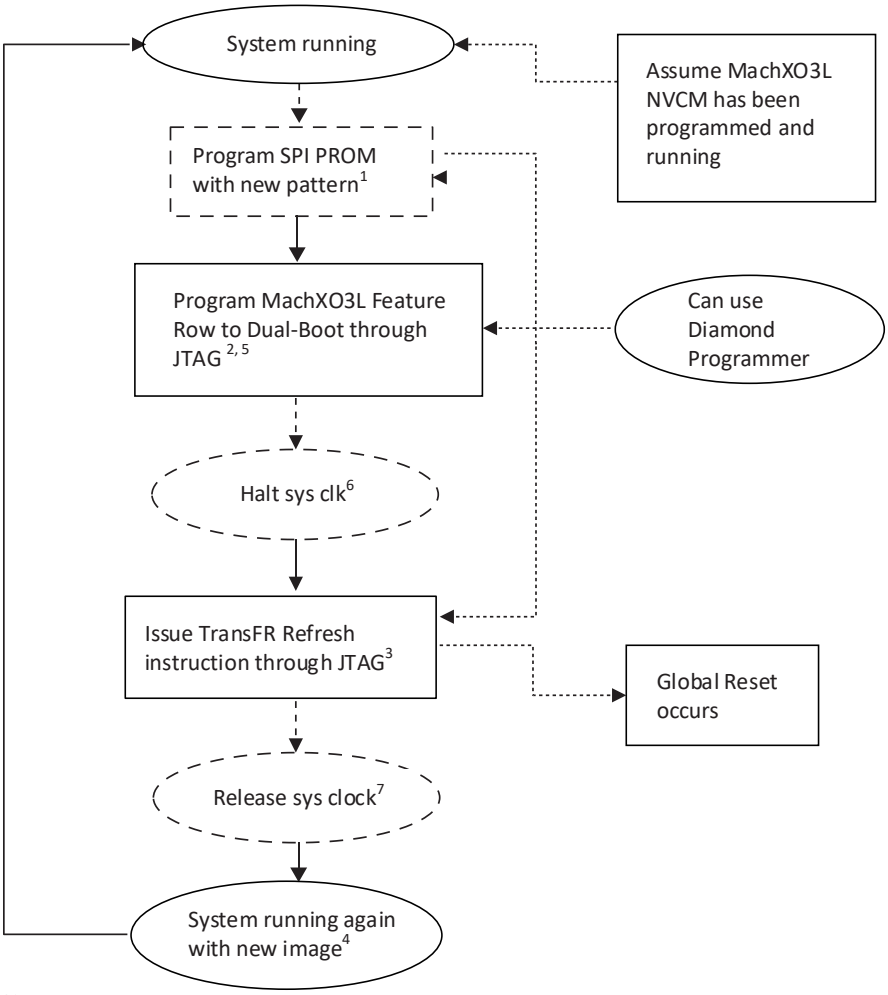


Figure 6.2. Bitstream Update Using TransFR

The example assumes that you have the golden image stored in NVCM in order to initiate the system, and is then use SPI PROM as a resource for image updates without disturbing the system. Figure 6.3 shows the process flow for performing this task.



Notes:
1. User can use operations such as “SPI Flash Background Erase, Program, Verify” for this.
2. User can use operations such as “XNVCM Program Feature Rows” for this.
3. User can use operations like “XNVCM TransFR” for this.
4. If new image failed to config MachXO3L, the golden image in NVCM will still config MachXO3L, so system will still be running with original image.
5. Feature Row only needs to be programmed if changes need to be made, for instance, disable or enable JTAG, Slave Port. If no changes need to be made, please skip this step.
6. This step is optional.

Figure 6.3. Example Process Flow

Caution when using the above process flow:

Since a Global Reset is triggered during device wake-up after Refresh instruction is issued, attention needs to be given in designing I/O with following conditions:

- Register output pins
- Impact on the system board level when value changes (may shut off the board, for instance)
- Register is set/reset by global reset

For the I/O in the example above, the state of the I/O is not changed during the TransFR refresh, but may change once the device gets into user mode right after the TransFR refresh. Following are design tips to avoid this:

- For critical I/O, try not to use global reset.
- For critical I/O, if you have to use global reset, try to use the set/reset option so that when GSR occurs, the state of the I/O pin does not trigger a system crash.

6.3. Password

The MachXO3 supports a password-based security access feature also known as Flash Protect Key. The Flash Protect Key feature provides a method of controlling access to the Configuration and Programming modes of the device. When enabled, the Configuration and Programming edit mode operations (including Write, Verify, and Erase operations) are allowed only when coupled with a Flash Protect Key which matches that expected by the device.

The Flash Protect Key feature requires that a device accessing a MachXO3 device through a sysCONFIG port (JTAG, SSPI, I^2C or WISHBONE) provide a valid digital Password, also known as the Flash Protect Key, to unlock the device and allow configuration or programming operations to proceed. Without a valid Flash Protect Key, you can perform only rudimentary non-configuration operations such as Read Device ID.

The 64-bit Flash Protect Key is stored in the Feature Row. Two additional feature row fuses are specified for enabling the feature: PWD_Enable and PWD_Enable_all.

You can read more about the Password feature in Using Password Security with MachXO3 Devices (FPGA-TN-02072).

7. Software Selectable Options

The operation of the MachXO3L/LF configuration logic is managed by options selected in the Diamond design software. Other FPGAs provide dedicated I/O pins to select the configuration mode. The MachXO3L/LF uses the non-volatile Feature Row to select how it configures. The Feature Row's default state needs to be modified in almost every design. You use the Diamond Spreadsheet View to make the changes to the operation of the MachXO3L/LF Feature Row which alters the operation of the configuration logic.

The configuration logic preferences are accessed using Spreadsheet View. Click on the Global Preferences tab, and look for the sysCONFIG tree. The sysCONFIG section is shown in Figure 7.1. The sysCONFIG preferences are divided into three categories:

- Configuration mode and port related
- Bitstream generation related
- Security related

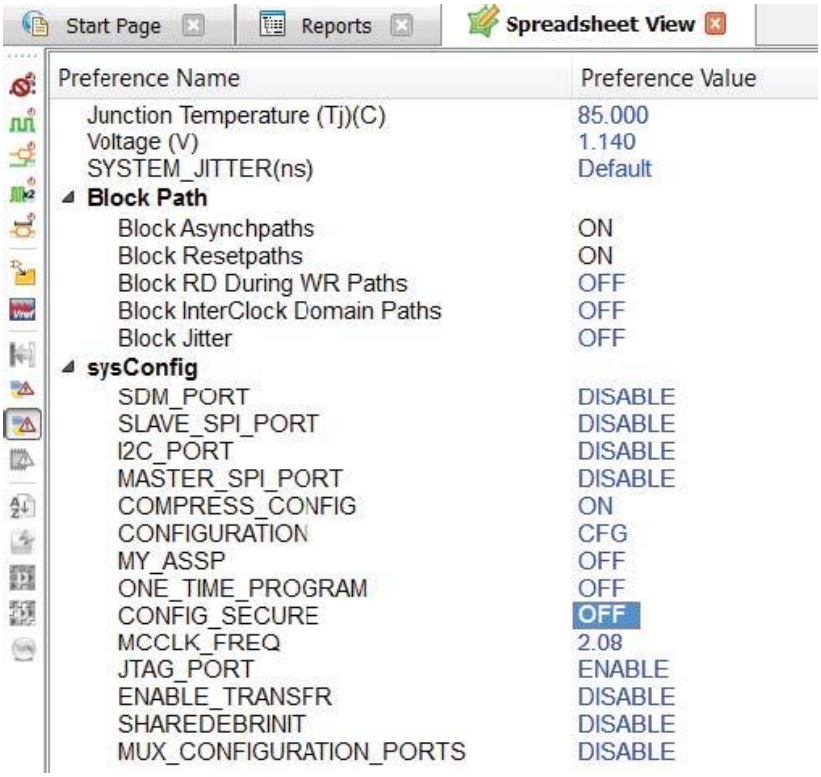


Figure 7.1. sysCONFIG Preferences in Global Preferences Tab, Diamond Spreadsheet View

7.1. Configuration Mode and Port Options

The configuration and port options allow you to decide which configuration ports continue to operate after the MachXO3L/LF device is in user mode. You can also control the availability of status pins, as well as the speed at which configuration data is read from an external PROM. The selections made here are saved in the Feature Row and remain in effect until the Feature Row is erased. The only exception is the MCCLK_FREQ parameter, which is stored in the configuration data.

The configuration and port options can be used in any combination.

Table 7.1. Configuration Mode/Port Options

Option Name	Default Setting	All Settings
JTAG_PORT	ENABLE	ENABLE, DISABLE
SLAVE_SPI_PORT	DISABLE	ENABLE, DISABLE
MASTER_SPI_PORT	DISABLE	ENABLE, DISABLE, EFB_USR
I2C_PORT	DISABLE	ENABLE, DISABLE
SDM_PORT*	DISABLE	DISABLE, PROGRAMN, DONE, INITN, PROGRAMN_DONE, PROGRAMN_DONE_INITN
MCCLK_FREQ	2.08	See description in the MCCLK Frequency section
ENABLE_TRANSFR	DISABLE	DISABLE, ENABLE

*__Note:__ The default for SDM_PORT was PROGRAMN in ispLEVER 8.1 SP1 and Diamond 1.1.

7.1.1. JTAG Port

The JTAG_PORT preference allows you to decide how the JTAG configuration port pins operate when theMachXO3L device is in user mode. There are two states to which the JTAG_PORT can be set:

- ENABLE – In this mode, the JTAG I/O pins are dedicated and provide an IEEE 1149.1 JTAG interface.
- DISABLE – In this mode, the JTAG I/O pins are controlled dynamically using the JTAGENB pin.

The JTAGENB pin is only available when the JTAG_PORT is in the DISABLE state. JTAGENB, when asserted high, makes the four JTAG I/O act as an IEEE 1149.1 JTAG port. JTAGENB driven low causes the four I/O to be available for use as general purpose I/O.

Lattice recommends designing so that the JTAG port can be accessed in the event reprogramming the MachXO3L disables your primary configuration port.

7.1.2. Slave SPI Port

The SLAVE_SPI_PORT allows you to preserve the Slave SPI configuration port after the MachXO3L/LF device enters user mode. There are two states to which the SLAVE_SPI_PORT preference can be set:

- ENABLE – This setting preserves the SPI port I/O when the MachXO3L/LF device is in user mode. When the pins are preserved, an external SPI master controller can interact with the configuration logic. The preference also prevents you from over-assigning I/O to the port pins.
- DISABLE – This setting disconnects the SPI port pins from the configuration logic. By itself, it does not make the port pins general purpose I/O. Both the SLAVE_SPI_PORT and MASTER_SPI_PORT must be in the DISABLE state for the SPI port pins to be general purpose I/O.

The SLAVE_SPI_PORT can be enabled at the same time as the MASTER_SPI_PORT. It is necessary to guarantee that the internal SPI master controller not perform SPI transactions at the same time as an external SPI master. It is your responsibility to prevent two SPI masters from operating simultaneously.

7.1.3. Master SPI Port

The MASTER_SPI_PORT allows you to preserve the SPI configuration port after the MachXO3L/LF device enters user mode. There are three states to which the MASTER_SPI_PORT preference can be set:

- ENABLE – This setting preserves the SPI port I/O when the MachXO3L/LF is in user mode. This preference makes External or Dual Boot configuration modes active. Using this preference in combination with CONFIGURATION = EXTERNAL enables external boot mode. This preference in combination with CFG enables Dual Boot mode. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. The preference also prevents you from over-assigning I/O to the port pins.
- EFB_USER – This setting preserves the SPI port I/O when the MachXO3L/LF is in user mode. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. The preference also prevents you from over-assigning I/O to the port pins.
- DISABLE – This setting disconnects the SPI port pins from the configuration logic. By itself it does not make the port pins general purpose I/O. Both the SLAVE_SPI_PORT and MASTER_SPI_PORT must be in the DISABLE state for the SPI port pins to be general purpose I/O.

The MASTER_SPI_PORT can be enabled at the same time as the SLAVE_SPI_PORT. It is necessary to guarantee that the internal SPI Master controller not perform SPI transactions at the same time as an external SPI Master. It is your responsibility to prevent two SPI masters from operating simultaneously.

7.1.4. I²C Port

The I2C_PORT allows you to preserve the I²C configuration port after the MachXO3L/LF device enters user mode. There are two states to which the I2C_PORT preference can be set:

- ENABLE – This setting preserves the I²C port I/O when the MachXO3L/LF is in user mode. When the pins are preserved, and the EFB is instantiated with *wb_clk_i* input connected to a valid clock source of at least 7.5x the I²C bus rate, an external I²C Master controller can interact with the configuration logic. The preference also prevents you from over-assigning I/O to the port pins.
- DISABLE – This setting disconnects the I²C port pins from the configuration logic. The port pins become general purpose I/O.

In order to use the primary and secondary I²C controllers in the EFB, the I2C_PORT must be in the ENABLE state.

7.1.5. SDM Port

The SDM_PORT allows you to select the programming status pins after the MachXO3L device enters user mode. There are six states to which the SDM_PORT preference can be set:

- DISABLE – This setting causes the PROGRAMN, DONE, and INITN status pins to become general purpose I/O.
- PROGRAMN – This setting preserves the PROGRAMN pin when the MachXO3L/LF device is in user mode. Asserting this pin active low causes the MachXO3L device to reconfigure. The DONE and INITN pins are general purpose I/O.
- DONE – This setting preserves the DONE pin when the MachXO3 device is in user mode. The PROGRAMN and INITN pins are general purpose I/O.
- INITN – This setting preserves the DONE pin when the MachXO3 device is in user mode. The PROGRAMN and DONE pins are general purpose I/O.
- PROGRAMN_DONE – This setting preserves the PROGRAMN and DONE pins when the MachXO3L/LF device enters user mode. INITN is a general purpose I/O.
- PROGRAMN_DONE_INITN – This setting preservers PROGRAMN, DONE, and INITN in user mode.

Lattice recommends setting the SDM_PORT to PROGRAMN when using Master SPI or Dual Boot Configuration Modes. The PROGRAMN pin is the only way to perform a *warm* reconfiguration of the MachXO3L/LF device, unless another configuration port is available to transmit a REFRESH command.

7.1.6. MCCLK Frequency

The MCLK_FREQ preference allows you to alter the MCLK frequency used to retrieve data from an external SPI Flash when using EXTERNAL or Dual Boot configuration modes. The MachXO3L/LF uses a nominal 2.08MHz (+/-5.5%) clock frequency to begin retrieving data from the external SPI Flash. The MCLK_FREQ value is stored in the incoming configuration data. It is not stored in the Feature Row. The MachXO3L device reads a series of padding bits, a *start of data* word (0xBDB3) and a control register value. The control register contains the new MCLK_FREQ value. The MachXO3L/LF switches to the new clock frequency shortly after receiving the MCLK_FREQ value. The MCLK_FREQ has a range of possible frequencies available from 2.08 MHz up to 133 MHz (see Table 4.10). Make sure not to exceed the maximum clock rate of your SPI Flash, or of your printed circuit board.

Lattice recommends having a back-up configuration port available in the event you specify a clock frequency that is out of specification.

7.1.7. ENABLE_TRANSFR

The TransFR function used by the MachXO3L/LF requires the configuration data loaded into the configuration SRAM, and any future configuration data file loaded into the internal NVCM/Flash memory have the ENABLE_TRANSFR set to the ENABLE state. See the TransFR Operation section, and Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02025) for more information about using TransFR with the MachXO3L/LF.

7.2. Bitstream Generation Options

The Bitstream Generation options allow you to decide how the Diamond development tools create the configuration data for the MachXO3L/LF device. The CONFIGURATION, USERCODE, CUSTOM_IDCODE, and SHAREDEBRINIT settings are saved in the Feature Row and remain in effect until the Feature Row is erased. The other options allow you to control the JEDEC and BIT files that are generated by Diamond.

7.2.1. COMPRESS_CONFIG

The COMPRESS_CONFIG preference alters the way JEDEC and BIT files are generated. The COMPRESS_CONFIG default setting is to be ON.

JEDEC files, when they are built, are always compressed. The configuration time is slightly reduced when reading configuration data from the external PROM and the Diamond tool creates a JEDEC file you can program into the internal memory.

7.2.2. CONFIGURATION

The CONFIGURATION preference allows you to control the Configuration memory sectors. The CONFIGURATION preference has two possible settings:

- CFG – The CFG preference is the default mode for building configuration data. The configuration bitstream is stored in the NVCM/Flash. The configuration data includes EBR initialization data.
- CFG_EBRUFM (MachXO3LF) – This preference creates configuration data that is stored in the Configuration Flash. EBR initialization data is stored in the lowest page addresses of the UFM sector. The UFM sector is available in user mode. You must restore the EBR initialization data when making changes to the UFM to guarantee correct operation.
- CFGUFM (MachXO3LF) – This preference creates configuration data that is stored in the Configuration Flash. This mode differs from CFG by allowing the configuration data to overflow into the UFM. The configuration data increases in size as EBR initialization data is added to the design.
- EXTERNAL – This preference generates configuration data that is stored in an external memory.

The CONFIGURATION preference defaults to the CFG state in the current release of the Diamond software. The Diamond design software only generates JEDEC files when your entire design fits within the NVCM/Flash memory.

Use the EXTERNAL preference to build configuration data for use with Master SPI Configuration Mode. When the configuration data exceeds the space available in the internal memory, it is necessary to switch to EXTERNAL mode. EXTERNAL mode does not use any internal resources.

7.2.3. USERCODE

The MachXO3L/LF Configuration NVCM (MachXO3L)/Flash (MachXO3LF) sector contains a 32-bit register for storing a user-defined value. The default value stored in the register is 0x00000000. Using the USERCODE preference you can assign any value to the register you desire. Suggested uses include the configuration data version number, a manufacturing ID code, date of assembly, or the JEDEC file checksum.

The format of the USERCODE field is controlled using the USERCODE_FORMAT preference. Data entry can be performed in either Binary, Hex, or ASCII formats.

7.2.4. USERCODE_FORMAT

The USERCODE_FORMAT preference selects the format for the data field used to assign a value in the USERCODE preference. The USERCODE_FORMAT has three options:

- Binary – USERCODE is set using 32 *1* or *0* characters.
- Hex – USERCODE is set using eight hexadecimal digits (that is 0-9A-F)
- ASCII – USERCODE is set using up to four ASCII characters

7.2.5. CUSTOM_IDCODE

The CUSTOM_IDCODE preference is used to assign a 32-bit register that resides in the Feature Row. The CUSTOM_IDCODE field is only active when the MY_ASSP preference is in the ON state. The value assigned can be entered in binary or hexadecimal, according to the CUSTOM_IDCODE_FORMAT preference. See the MY_ASSP section for more information about how to assign a value to the CUSTOM_IDCODE preference.

7.2.6. CUSTOM_IDCODE_FORMAT

The CUSTOM_IDCODE_FORMAT preference selects the format for the data field used to assign a value in the CUSTOM_IDCODE preference. The CUSTOM_IDCODE_FORMAT has two options:

- Binary – CUSTOM_IDCODE is set using 32 *1* or *0* characters
- Hex – CUSTOM_IDCODE is set using eight hexadecimal digits (that is 0-9A-F)

7.2.7. SHAREDEBRINIT

When set to ENABLE, this preference allows one copy of a unique memory initialization file to be stored in the internal memory. This copy of the initialization values can be shared among multiple EBRs. Doing so reduces the bit-stream size of the design and saves internal memory space for other applications.

7.2.8. MUX_CONFIGURATION_PORTS

The MUX_CONFIGURATION_PORTS is used in the event that all configuration ports are disabled. Disabling all of the available configuration ports turns the MachXO3 into a *write one time* device. MUX_CONFIGURATION_PORTS confirms the removal of all configuration ports. The control is only active when all of the configuration ports are set to the ENABLE state. MUX_CONFIGURATION_PORTS set to the ENABLE state enables the JTAGENB input pin, permitting the JTAG port pins to be multiplexed. Setting MUX_CONFIGURATION_PORTS to the ENABLE state causes the Diamond build tools to honor the removal of all configuration ports. If the JTAGENB input pin is hard connected to GND on the PCB, this allows the MachXO3 to become a *write one time* device. In other application scenarios, you can control the JTAGENB to provide dynamic selection between the JTAG port and GPIO.

7.2.9. DUALBOOTGOLDEN

The DUALBOOTGOLDEN preference is used to specify the location of the fail-safe boot image in Dual Boot applications. When set to INTERNAL, the MachXO3 device attempts to configure first from external flash PROM using MSPI mode. If the configuration fails, the MachXO3 then attempts to configure itself from the *Golden* fail-safe image in internal NVCM/Flash using SDM. When set to EXTERNAL, the MachXO3 device attempts to configure first from the internal NVCM/Flash memory using SDM. If the SDM configuration fails, the MachXO3 then attempts to configure itself from the *Golden* fail-safe image using MSPI mode.

7.3. Security Options

The Security Options allow you to select from a range of options for tracking or securing the MachXO3L/LF device. Table 7.2 provides a summary of these options.

Table 7.2. Security Options

Option Name	Default Setting	All Settings
TRACEID	<all zero>	8-bit arbitrary
MY_ASSP	OFF	OFF, ON
CONFIG_SECURE	OFF	OFF, ON
ONE_TIME_PROGRAM	OFF	OFF, NVMEM, NVMEM_SRAM

7.3.1. TRACEID

The MachXO3L/LF introduces a new feature called TraceID. TraceID stamps each MachXO3L/LF with a unique 64-bit ID. No two MachXO3L/LF devices has the same TraceID value even when they are loaded with the same configuration data. This differs from a USERCODE which is present in the configuration data. Every device that receives the configuration data using a USERCODE receives the same USERCODE value.

The TraceID is 64 bits long with the least significant 56 bits being immutable data. The 56 bits are a combination of the wafer lot, the wafer number and the X/Y coordinates locating the die on the wafer. The most significant eight bits are provided by you and are stored in the Feature Row. The TraceID is changed using the Diamond Spreadsheet View. You enter a unique 8-bit binary value in the TraceID field and generate configuration data.

You can read more about the TraceID feature in Using TraceID (FPGA-TN-02027).

7.3.2. MY_ASSP

Every Lattice device has its own identification code identifying the device family, device density, and other parameters (for example voltage, device stepping, and others). The code is accessible from any MachXO3L/LF configuration port. The value stored in the IDCODE register allows you to uniquely identify a Lattice device.

The MY_ASSP preference permits you to change the value returned when the IDCODE is read from the FPGA. Set the MY_ASSP preference to the ON state. Turning the MY_ASSP ON enables the CUSTOM_IDCODE preference.

7.3.3. CUSTOM_IDCODE

The CUSTOM_IDCODE is the value you assign to override the default IDCODE in the MachXO3L/LF device. You are only allowed to enter a 32-bit hexadecimal or binary value when the MY_ASSP preference is ON.

Overriding the IDCODE prevents the Lattice programming software from being able to identify the MachXO3L/LF device, and as a result, prevents Programmer from being able to directly program the MachXO3L device. It is necessary to migrate to generating Serial Vector Format (SVF) files in order to program MY_ASSP enabled MachXO3L/LF devices.

7.3.4. CONFIG_SECURE

When this preference set to ON, the read-back of the SRAM memory and the NVCM/Flash memory are blocked using any of the sysCONFIG ports (JTAG, SPI, I^2C, WISHBONE). Note that all other functions using sysCONFIG ports (JTAG, SPI, I^2C, WISHBONE) are operational except read-back as mentioned earlier.

In MachXO3LF devices, the read-back of the UFM will also be blocked if the bitstream overflows into the UFM block. The MachXO3L/LF device cannot be read back, nor can it be programmed without erasing. The device must be erased in order to reset the security setting. The CONFIG_SECURE fuse and the NVCM/Flash are erased in tandem. Once the security fuses are reset, the device can be programmed again.

7.3.5. ONE_TIME_PROGRAM

The MachXO3L/LF has One Time Programmable (OTP) fuses that can be used to prevent the on-chip memory from being erased or programmed. The MachXO3L/LF device has OTP security fuses for the SRAM and NVCM/Flash memory sectors. This preference provides options to set the OTP security for each memory sector.

- NVMEM – The NVCM/Flash cannot be erased or programmed
- NVMEM_SRAM – The NVCM/Flash and SRAM cannot be erased or programmed

Once the ONE_TIME_PROGRAM preference is set for the NVMEM memory, the on-chip NVCM/Flash memory cannot be erased or programmed. The configuration data is prevented from further modification, but the SDM mode can still be used to configure the device.

When the ONE_TIME_PROGRAM preference is set for the NVMEM_SRAM memory, the device acts like an ASIC. You are no longer able to reprogram the internal NVCM/Flash, and the SRAM cannot be changed from the JTAG port. Configuration of SRAM from on-chip NVCM/Flash or external SPI Flash is still enabled.

7.3.6. BACKGROUND_RECONFIG

The BACKGROUND_RECONFIG preference specifies the behavior of the PROGRAMN pin (when enabled) and the sysCONFIG REFRESH command.

When set to OFF (default), toggling PROGRAMN or transmitting the REFRESH command initiates a standard *warm-boot* SRAM reconfiguration sequence from the specified configuration image (internal or external) or images (dual boot). The *warm-boot* sequence executes initialization, configuration and wake-up steps before reentering user mode (see Power-up Sequence section).

When set to ON, toggling PROGRAMN or transmitting the REFRESH command initiates a *background* SRAM reconfiguration sequence from the specified configuration image (internal or external) or images (dual boot). The *background* sequence executes the configuration step alone. The initialization and wake-up steps are bypassed while the device remains in user mode throughout the sequence. This is typically used in conjunction with soft-error-detection (SED) to support soft-error-correction (SEC). The device operates without disruption. Only erroneous SRAM cells are corrected.

8. Device Wake-up Sequence

When configuration is completed (the SRAM is loaded), the device wakes up in a predictable fashion. If the MachXO3L/LF device is the only device in the chain, or the last device in a chain, the wake-up process should be initiated by the completion of configuration. Once configuration is completed, the internal DONE bit is set and the wake-up process begins. Figure 8.1 shows the wake-up sequence using the internal clock.

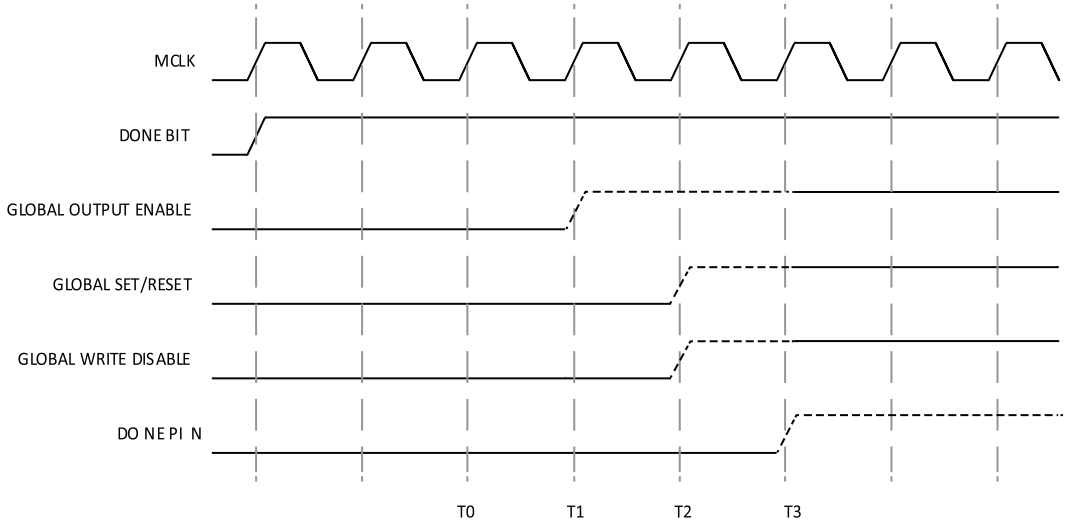


Figure 8.1. Wake-up Sequence Using Internal Clock

8.1. Wake-up Signals

Three internal signals, GOE, GSR, and GWDIS, determine the wake-up sequence.

- When low, GOE prevents the device's I/O buffers from driving the pins. The GOE only controls output pins. Once the internal DONE is asserted the MachXO3L/LF responds to input data.
- GSR is used to set and reset the core of the device. GSR is asserted (low) during configuration and deasserted (high) in the wake-up sequence.
- When the GWDIS signal is low, it safeguards the integrity of the RAM Blocks and LUTs in the device. This signal is low before the device wakes up. This control signal does not control the primary input pin to the device but controls specific control ports of EBR and LUTs.
- When high, the DONE pin indicates that configuration is completed and that no errors are detected.

8.2. Wake-up Clock Selection

The clock source used to complete the four state transitions in the wake-up sequence is user-selectable. Once the MachXO3L/LF is configured, it enters the wake-up state, which is the transition between the configuration mode and user mode. This sequence is synchronized to a clock source, which defaults to MCLK/CCLK when sysCONFIG is used, or TCK when JTAG is used.

You can change the clock used by instantiating the START macro in your Verilog or VHDL. The clock must be supplied on an external input pin, because the MachXO3L/LF does not begin internal operations until the Wake-up sequence is complete. There is no external indication the device is ready to perform the last four state transitions. You must either provide a free running clock frequency, or you must wait until the device is guaranteed to be ready to wake up. Using the START macro provides another mechanism for holding off configuring one or more programmable devices and then starting them synchronously.

Verilog

```
module START (STARTCLK);
input STARTCLK;
endmodule

START u1 (.STARTCLK(<clock_name>)) /* synthesis syn_noprune=1 */;
```

VHDL

```
COMPONENT START
     PORT(
           STARTCLK      :     IN STD_ULOGIC
         );
END COMPONENT;
attribute syn_noprune: boolean ;
attribute syn_noprune of START: component is true;

begin
u1: START port map (STARTCLK =><clock name>);
```

9. Advanced Configuration Information

9.1. NVCM/Flash Programming

The MachXO3's internal NVCM/Flash is the heart of the FPGA's configuration system. It is flexible, allowing you to store the FPGA's configuration data, as well as storing design specific data in the internal memory. It is also a resource that uses a precise erase and programming sequence. Lattice provides several methods for programming the MachXO3 NVCM/Flash:

- JTAG or Slave SPI programming
- VMEmbedded – *C* source for use with an embedded microprocessor controlling the JTAG port
- SSPIEmbedded – *C* source for use with an embedded microprocessor controlling the SSPI port
- Custom – The information in this section, and information from Using Hardened Control Functions in MachXO3 Devices Reference Guide (FPGA-TN-02064), permits creation of a custom solution.

The NVCM (MachXO3L)/Flash (MachXO3LF) space can be accessed by the JTAG, I^2C, and SPI ports. These configuration ports may use offline or transparent modes to erase, program, and verify the MachXO3 NVCM (MachXO3L)/Flash (MachXO3LF) resources. The WISHBONE interface is only permitted to use transparent programming operations (MachXO3LF). The sequence and timing of the commands presented to the Configuration Logic are identical across all of the configuration ports. There are slight differences due to communication protocol standards when transmitting commands and data. The command and timing flow common to all configuration ports is described first. Protocol variances are described afterward.

Each MachXO3 contains a certain quantity of internal memory. The amount of memory depends on the device density of the MachXO3. Table 9.1 shows the number of internal memory pages available for each MachXO3 device density. Each page represents 128 bits of data.

Table 9.1. Number of Pages of NVCM/Flash Memory for the MachXO3 Family

MachXO3 Device Density	NVCM0 /Configuration Flash (Pages)	NVCM1/UFM (Pages)	Total NVCM/(CFG + UFM) Bridged (Usable Pages)*
9400	12,539	3,582	16,121
6900	9,211	2,046	11,257
4300	5,758	767	6,525
2100	3,198	639	3,837
1300	2,175	511	2,686
640	1,151	191	1,342

*__Note:__ Usable (NVCM0 + NVCM1)/(CFG+UFM) (CONFIGURATION = CFGUFM) page count may be less than the actual bitstream size due to device limitations.

9.2. MachXO3 JEDEC File Format

All Lattice non-volatile devices support JEDEC files. Utilities are available in the Deployment Tool software for converting the JEDEC file into other programming file formats, such as STAPL, SVF, or bitstream (hex or binary). The relevant detail about the JEDEC file is provided in Table 9.1 for completeness.

Table 9.2. MachXO3 JEDEC File Format

JEDEC Field	Syntax	Description
Don't Care	My design	Characters appearing before the ^B character are don't care. All character sets or internal language can be used here except ^B.
Start-of-text	^B	^B (Control-B 0x02) marks the beginning of the JEDEC file. Only ASCII characters are legal after ^B. The character * is the delimiter to mark the ending of a JEDEC field. The CR and LF are treated as regular white spaces and have no delimiter function in a JEDEC file.

JEDEC Field	Syntax	Description
Header	My design	The first field is the header, which does not have an identifier to indicate its start. Only ASCII characters are legal after ^B. The header is terminated by an asterisk character *.
Field Terminator	*	Each field in the JEDEC file is terminated with an asterisk.
Note (Comment)	NOTE my design	The key word N marks the beginning of the comment. It can appear anywhere in the JEDEC file. Lattice's JEDEC files add *OTE* to the N key word to make it a more meaningful word NOTE.
Fuse Count	QF3627736	The key word QF identifies the total real fuse count of the device.
Default Fuse State	F0 or F1	The key word F identifies the fuse state of those fuses not included in the link field. F0 = fill them with zeros (0), F1 = fill them with ones (1). It is defined for the purpose of reducing JEDEC file size. It has no meaning in Lattice's JEDEC file. Lattice recommends using compression to reduce file size instead.
Security Setting	G0 or G1	JEDEC standard defines G<0,1> to program security <0=no, 1=yes>
OTP and Security Setting	G0, G1, G2, or G3	Lattice enhances the G field to cover OTP fuse programming as well. G<0=both no, 1=only security yes, 2=only OTP yes, 3=both yes>.
Link Field	L0000000 101011…100011 …….. 111111…101100 110 101011…100011 …….. 111111…101100 110 …….. …….. 101011…100011 …….. 111111…101100 110* NOTE SED_CRC* L3627704 111111….111111* CC1B9	The keyword L identifies the first fuse address of the fuse pattern that follows after the white space. The number of digit shown following the L keyword must be the same as that on the QF field. In this example, QF3627736 has seven digits, thus L0000000 should have seven zeros. The fuse address traditionally starts counting from 0. The link field is the most critical portion of the JEDEC file where the programming pattern is stored. The programming data is written into this field in the manner mirroring exactly the fuse array layout of the silicon physically. Row address is written from top to bottom in ascending order: Top = Row 0, Bottom = Last Row. The column address is written from left to right in ascending order: Left most = bit 0, Right most = last bit. Row 0 is selected first by the INIT_ADDRESS command. The first bit to shift into the device is bit 0 for programming. The first to shift out from the device is also bit 0 when verify. The end of the Configuration data is marked by *NOTE END CONFIG DATA**. It is not necessary to program any page data containing all *0* values. NVCM1/UFM pages, if present in the JEDEC, are preceded by a *NOTE TAG DATA** line. If the JEDEC file is encrypted, all the data in the link field are encrypted. The column size increases accordingly to include filler bits to make the column size packet (128-bit, or 16 bytes, per packet) bounded.
Fuse Checksum	CC1B9	The checksum of all the fuses = Fuse count. The fuse state of all the fuses can be found from the Link field. If it is not specified in the link field, then use the Default Fuse State in their places. If the JEDEC file is encrypted, the fuse checksum is calculated after encryption. The fuse checksum prior to encryption can be found on one of the comments.
U Field	UA Home	This is the place to store the 32-bit USERCODE. The 32-bit USERCODE can be expressed in UA = ASCII, UH = ASCII Hex, U = Binary. Lattice enhanced this field for storing the CRC value of encrypted JEDEC.
E Field	EH 012..ABCDEF	JEDEC standard defines this field to hold the architecture fuses. Lattice uses this field to store the Feature Row and FEABITS. The Feature Row data is on the first line. The FEABITS values are on line 2.

JEDEC Field	Syntax	Description
End-of-text	^C	^C (CTLC) marks the ending of the JEDEC file.
Transmission Checksum	ABCD	This is the checksum of the whole file starting from ^B to ^C. All characters and white space, including the ^B and ^C, are included in the checksum calculation.

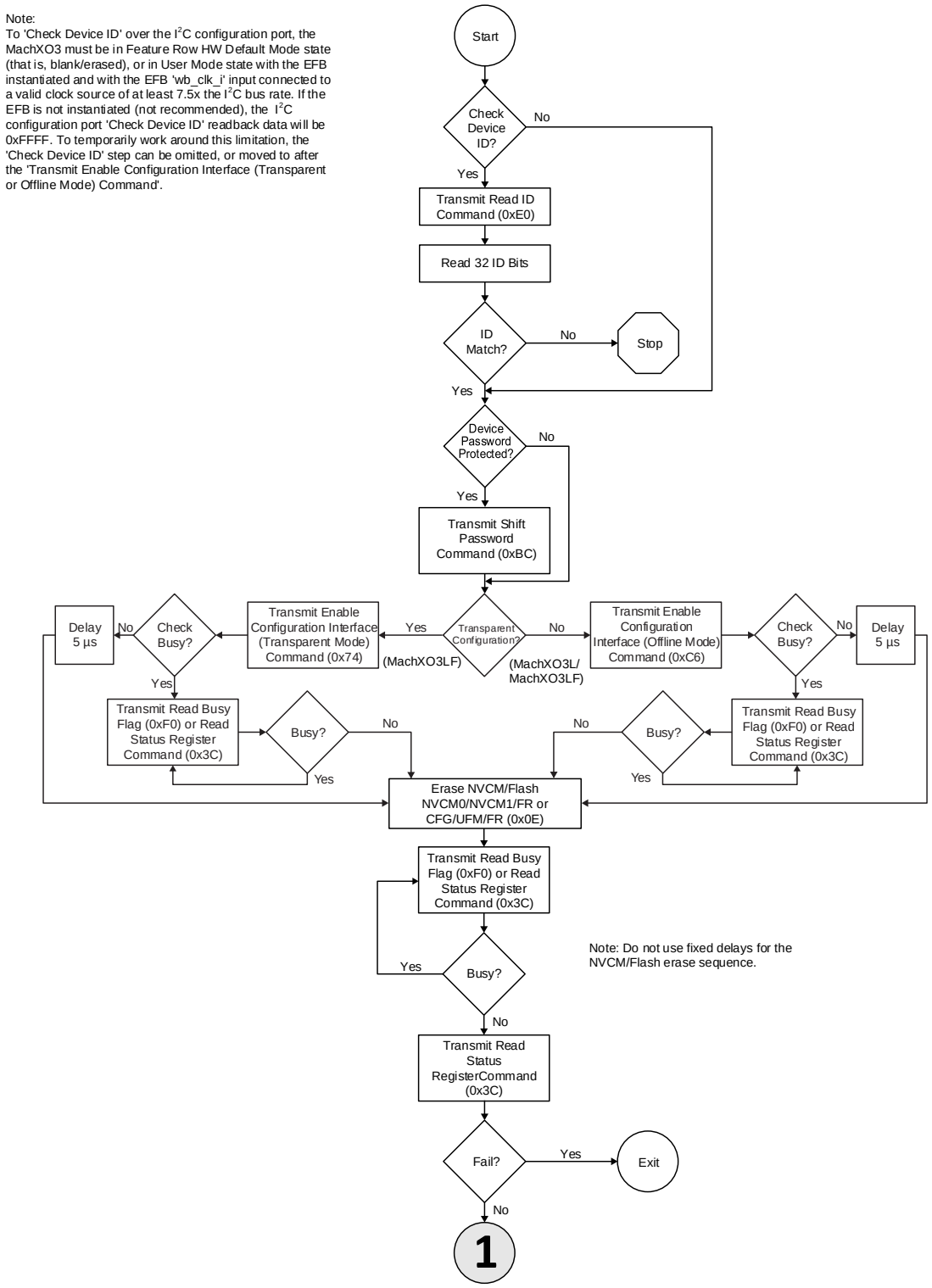
An example of a MachXO3 JEDEC file is shown in Figure 9.1.

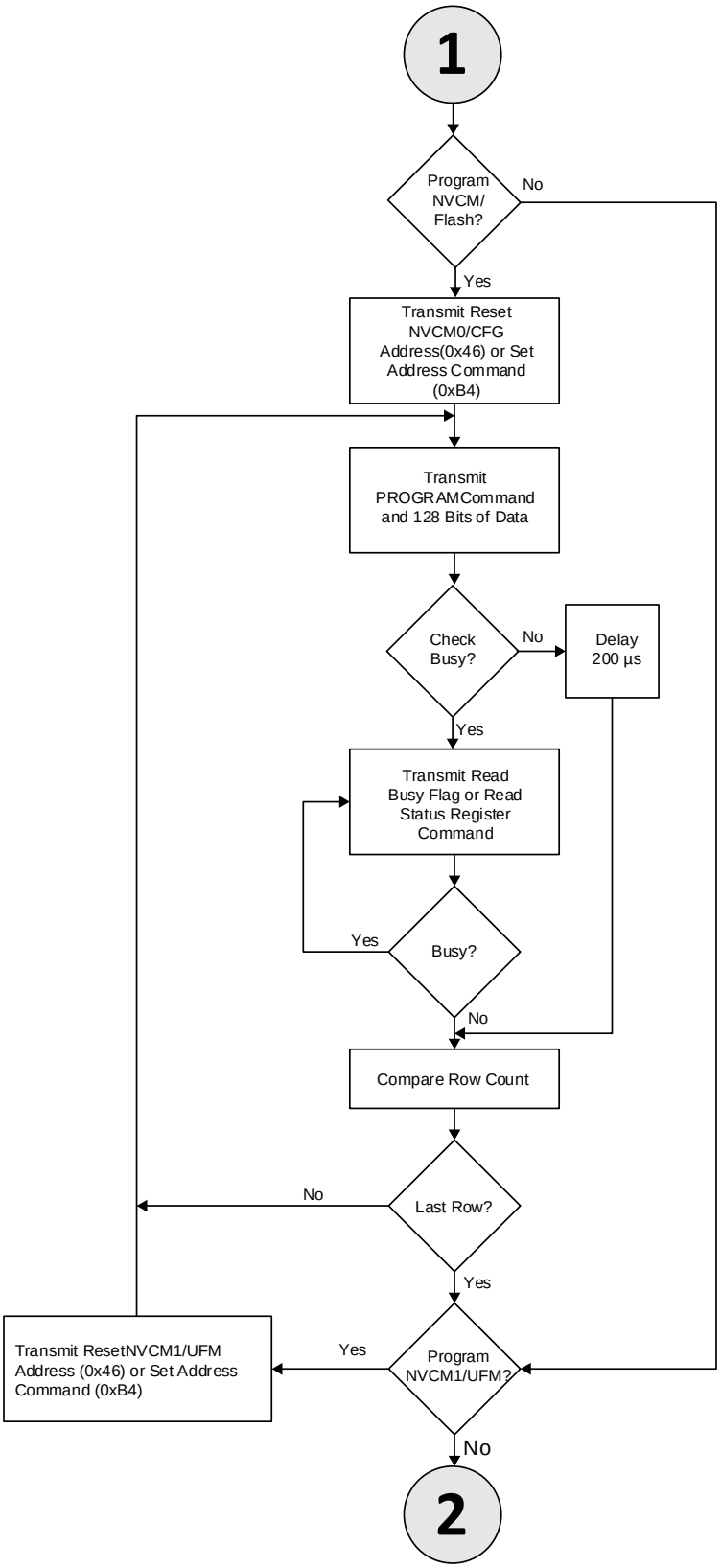
```
*
NOTE Diamond_1.2_Production (92) JEDEC Compatible Fuse File.*
NOTE Copyright (C), 1992-2010, Lattice Semiconductor Corporation.*
NOTE All Rights Reserved.*
NOTE DATE CREATED:Fri Dec 02 14:50:08 2011*
NOTE DESIGN NAME:control_SoC_demo_impl1.ncd*
NOTE DEVICE NAME:LCMXO3L-1300E-6CSFBGA256*
NOTE TRACEID 0x84*
NOTE LVDS_72P M9*
NOTE LVDS_72N N10*
NOTE LVDS_68P P8*
NOTE LVDS_68N M8*
NOTE PIN ASSIGNMENTS*
NOTE PINS cap_btn1 : M10 : inout*
NOTE PINS LCD_COM0 : B14 : out*
NOTE PINS xout : N7 : out*
...
NOTE PINS xin : M5 : in*
NOTE PINS Icc_analog_cmp_p : P8 : in*
QP132*
QF343936*
G0*
F0*
L000000
111111111111111110111101101100111111111111111111001110110000000000000000000000000010000000000000000000000000110000000 110
0100001001010000100100000100100000000100111111111010001100000000000000000000000000010111000111000000000000010100110100000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000001000100010001000000000000100010000000000001000100000000000100010000000001000100100010000001000100000010001000000 000
010001000100010000001000100000000000000000010001000000000000000000000000000000000000000000100010001001000000000000100100000 000
000001001000000000000001001000000000001000100010010000000100100000001001000000000010001000100100000001001000000000000000000010 010
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
...
000000000000000000000000000000000000000000010111101011010101111111111111111111111111111111111111111111111111111111111101110 010
0001000001000000000000101110000000000000000001100000000000000000000011100100001000001010100000010100000000100000000000000100001 001
1111111000000000000000000000000000000000000000000011000101000000000000000000000000110010101111111010111010101111101111000101000 100
001000100000000000000000000000000110000000000000000000000000000000001111111111111111111111111111111111111111111111111111111111 111
*
NOTE EBR_INIT DATA*
L137984
111111111111111111111111111111111111111111110110000000000000000000000000000000000000000000000110000000000101100100001000000000000010000 000
00011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110 100
011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110100011110 001
...
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
0010001000000000000000000000000000000000000000000000000000000000000000000000101111000000000000000000000000000011111111111111111111111111111111 111
*
NOTE END CONFIG DATA*
L184832
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
...
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
*
NOTE TAG DATA*
L343808
010001000100010000001000100000000000000000010001000000000000000000000000000000000000000000100010001001000000000000100100000 000
000001001000000000000001001000000000001000100010010000000100100000001001000000000010001000100100000001001000000000000000000010 010
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 000
*
C5CC8*
NOTE FEATURE_ROW*
E0000000000000000000000000000000000000001000010000100000000000000000000
0000010000000000*
NOTE User Electronic Signature Data*
UH CAFEBABE*
 6243
```

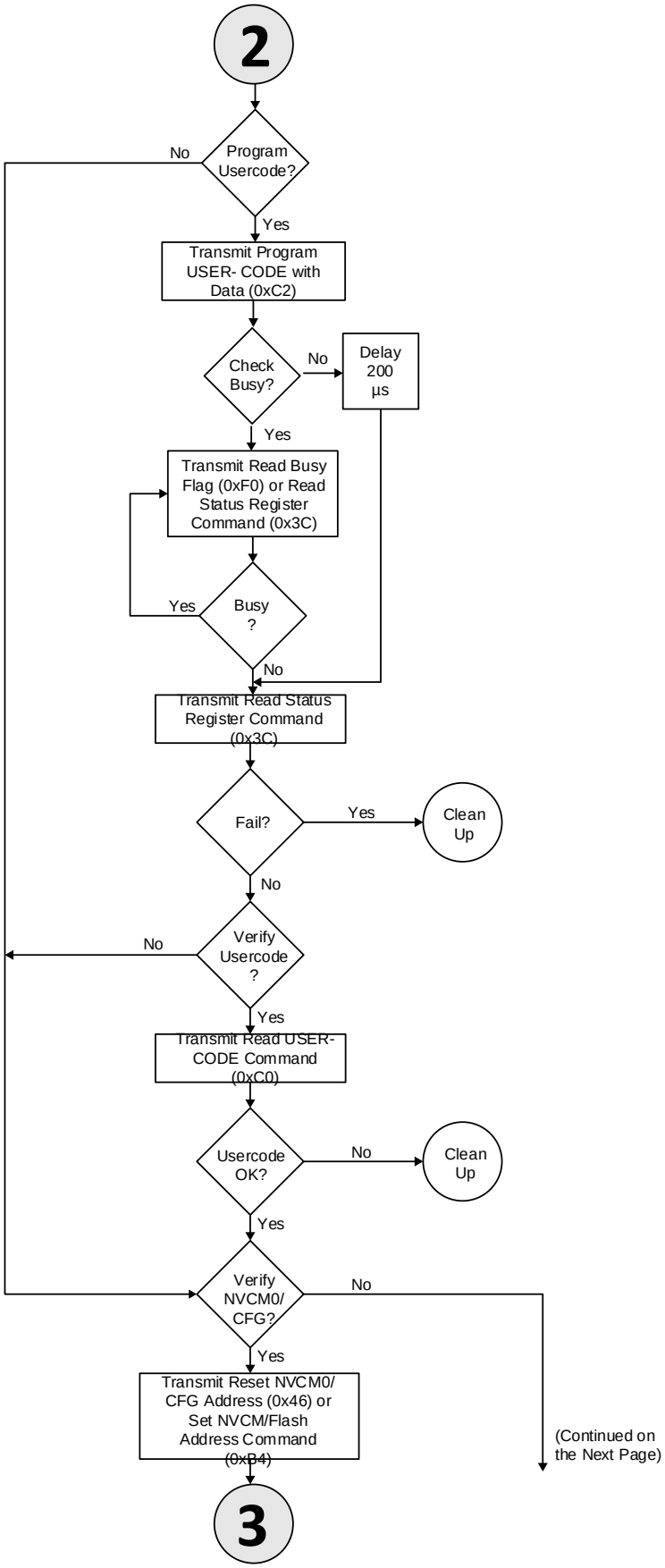
Figure 9.1. JEDEC File Example

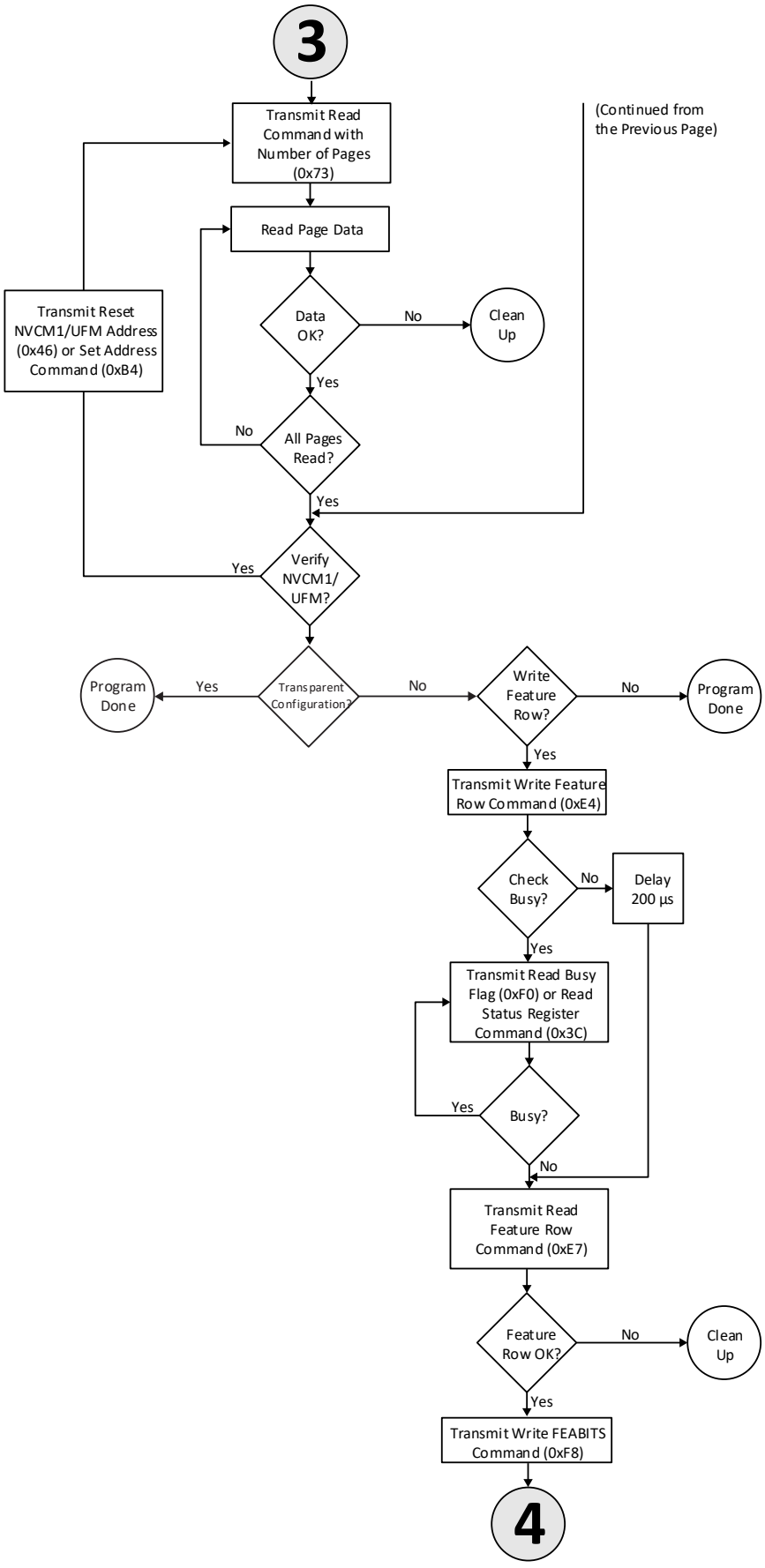
9.3. MachXO3 NVCM/Flash Programming Flow

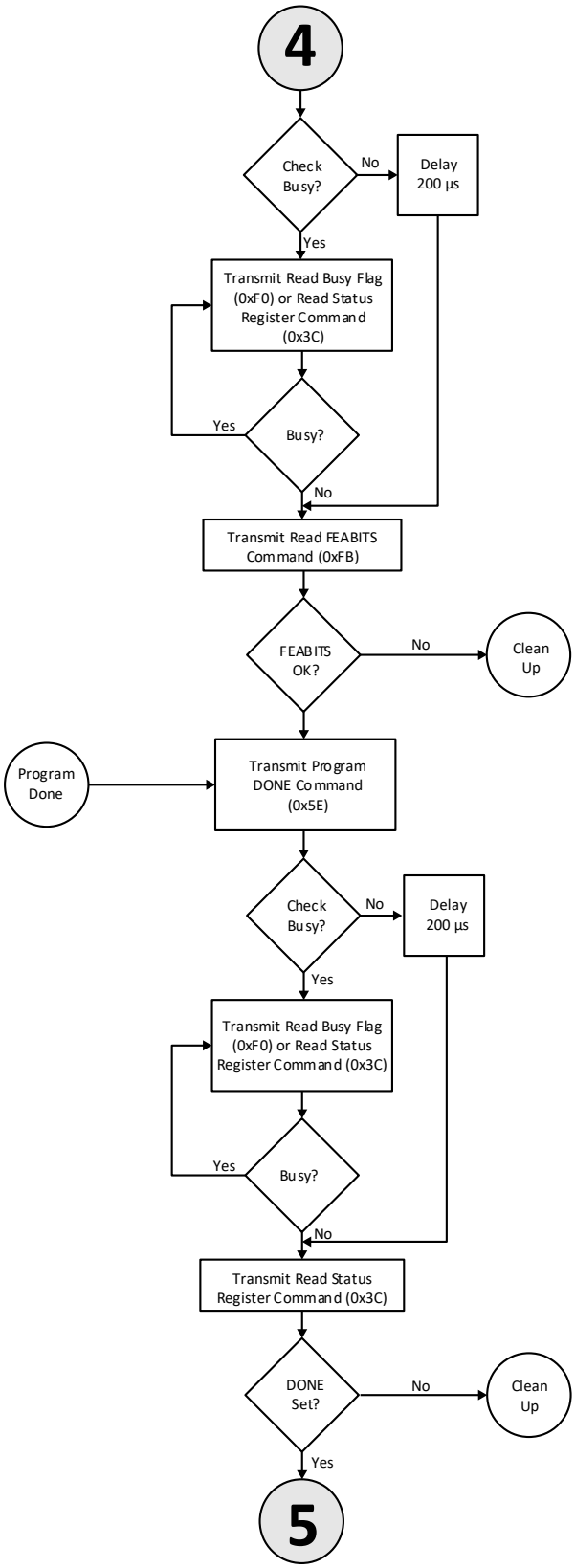
The MachXO3 NVCM/Flash memory erasure, and programming requires a specific set of steps and timing. The flow chart in this section describes the command sequences and the timing required for successful NVCM/Flash programming. The commands and timing are common between all of the configuration ports. There are some minor variations in the protocol, but not the timing, based on the configuration port used. Exceptions are described in the configuration port specific sections.

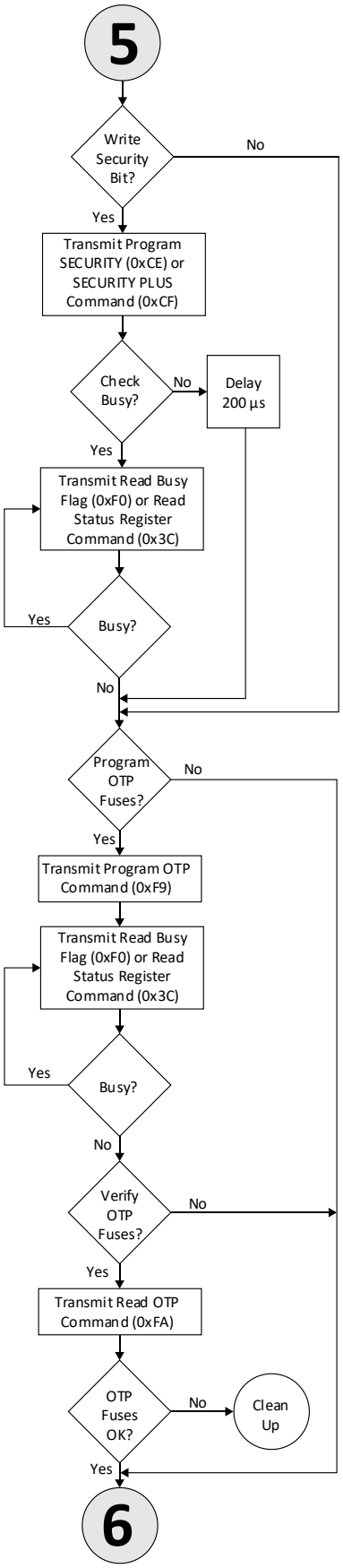


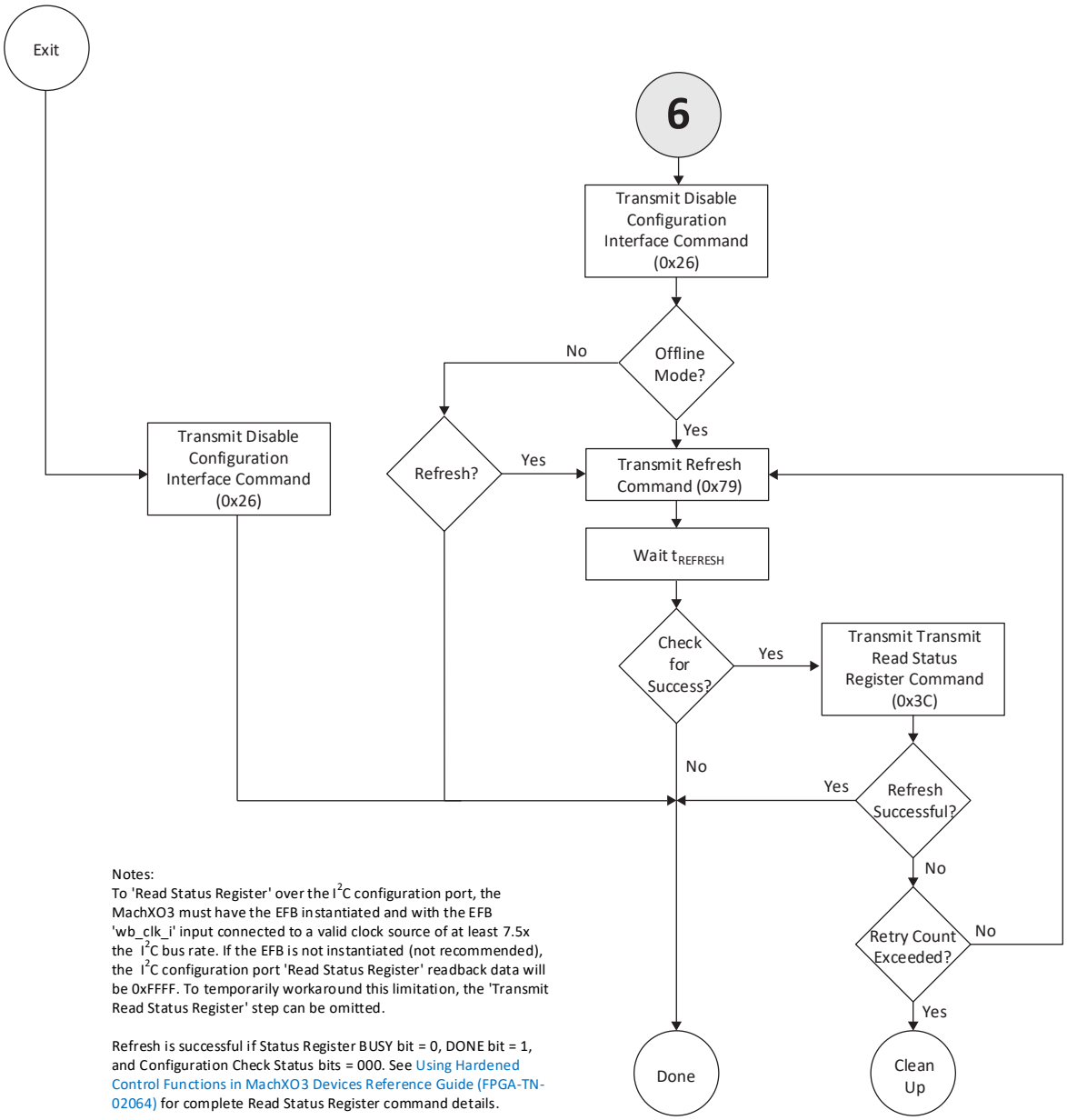












Notes:
To 'Read Status Register' over the I^2C configuration port, the MachXO3 must have the EFB instantiated and with the EFB 'wb_clk_i' input connected to a valid clock source of at least 7.5x the I^2C bus rate. If the EFB is not instantiated (not recommended), the I^2C configuration port 'Read Status Register' readback data will be 0xFFFF. To temporarily workaround this limitation, the 'Transmit Read Status Register' step can be omitted.

Refresh is successful if Status Register BUSY bit = 0, DONE bit = 1, and Configuration Check Status bits = 000. See Using Hardened Control Functions in MachXO3 Devices Reference Guide (FPGA-TN-02064) for complete Read Status Register command details.

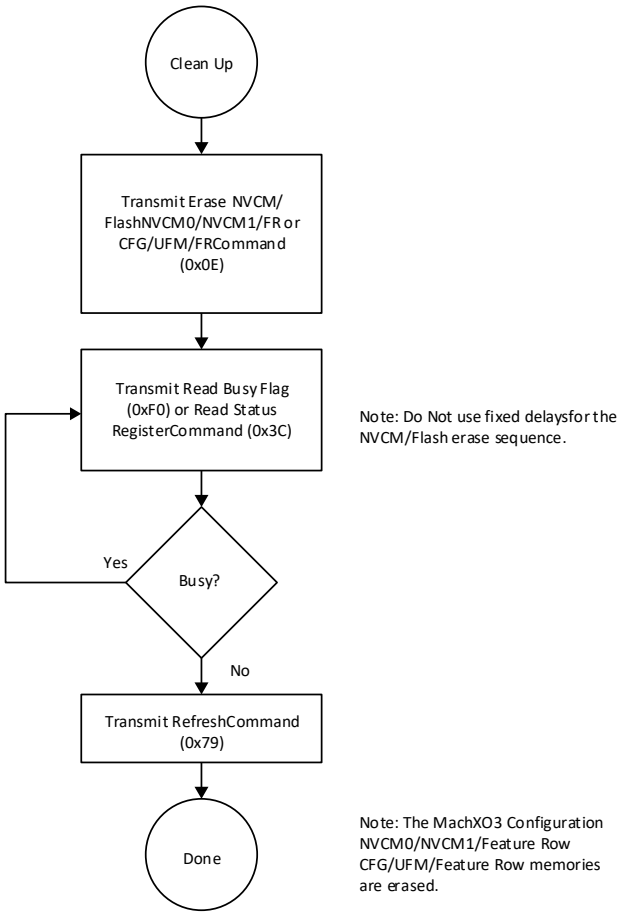
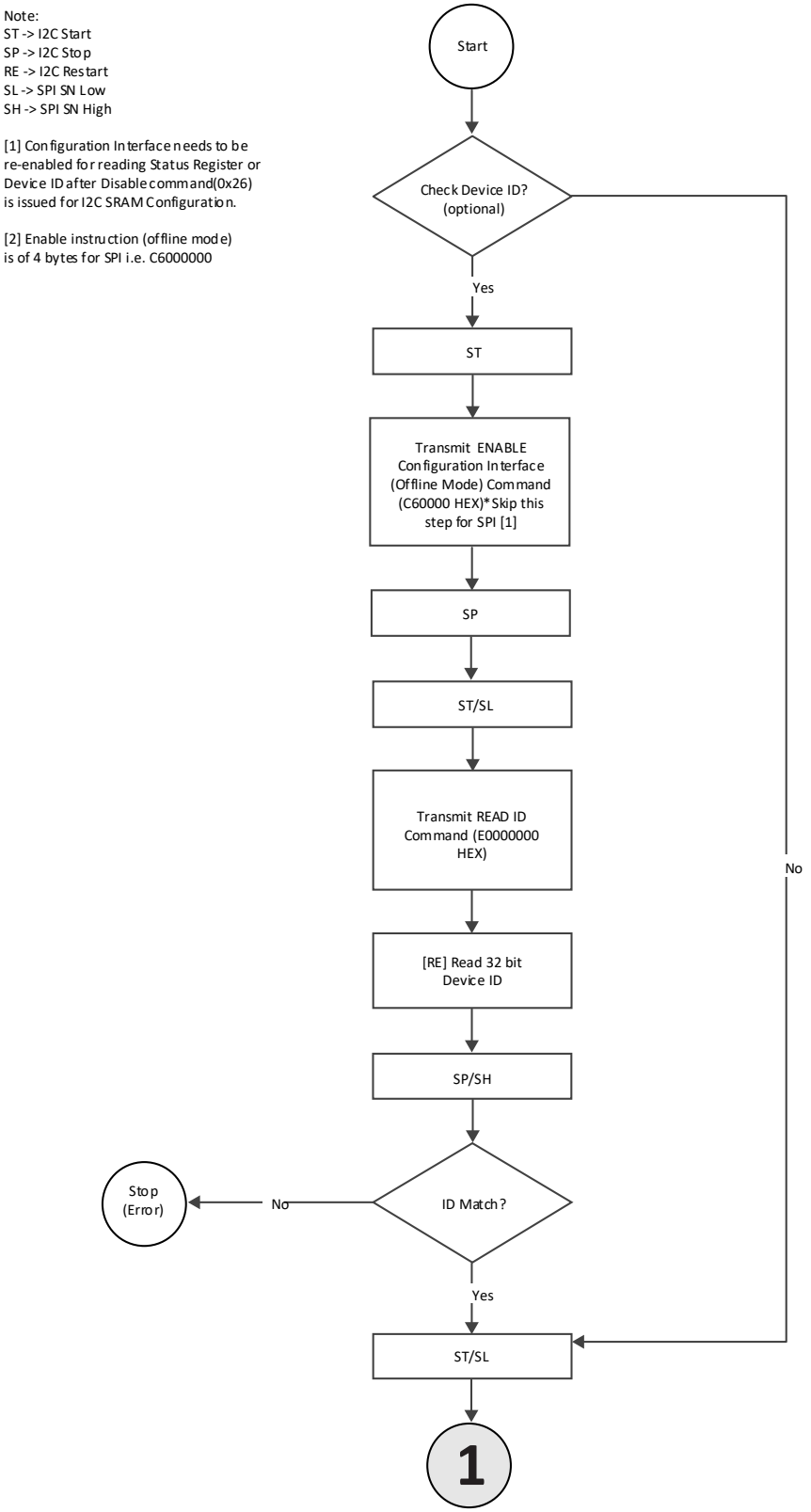
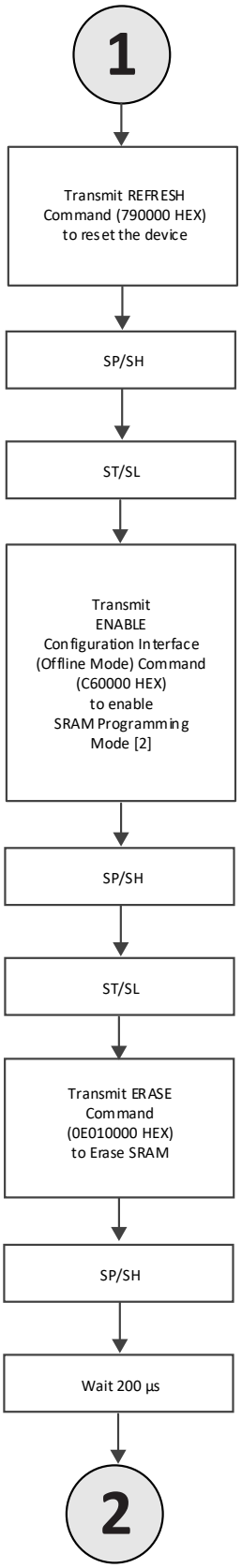


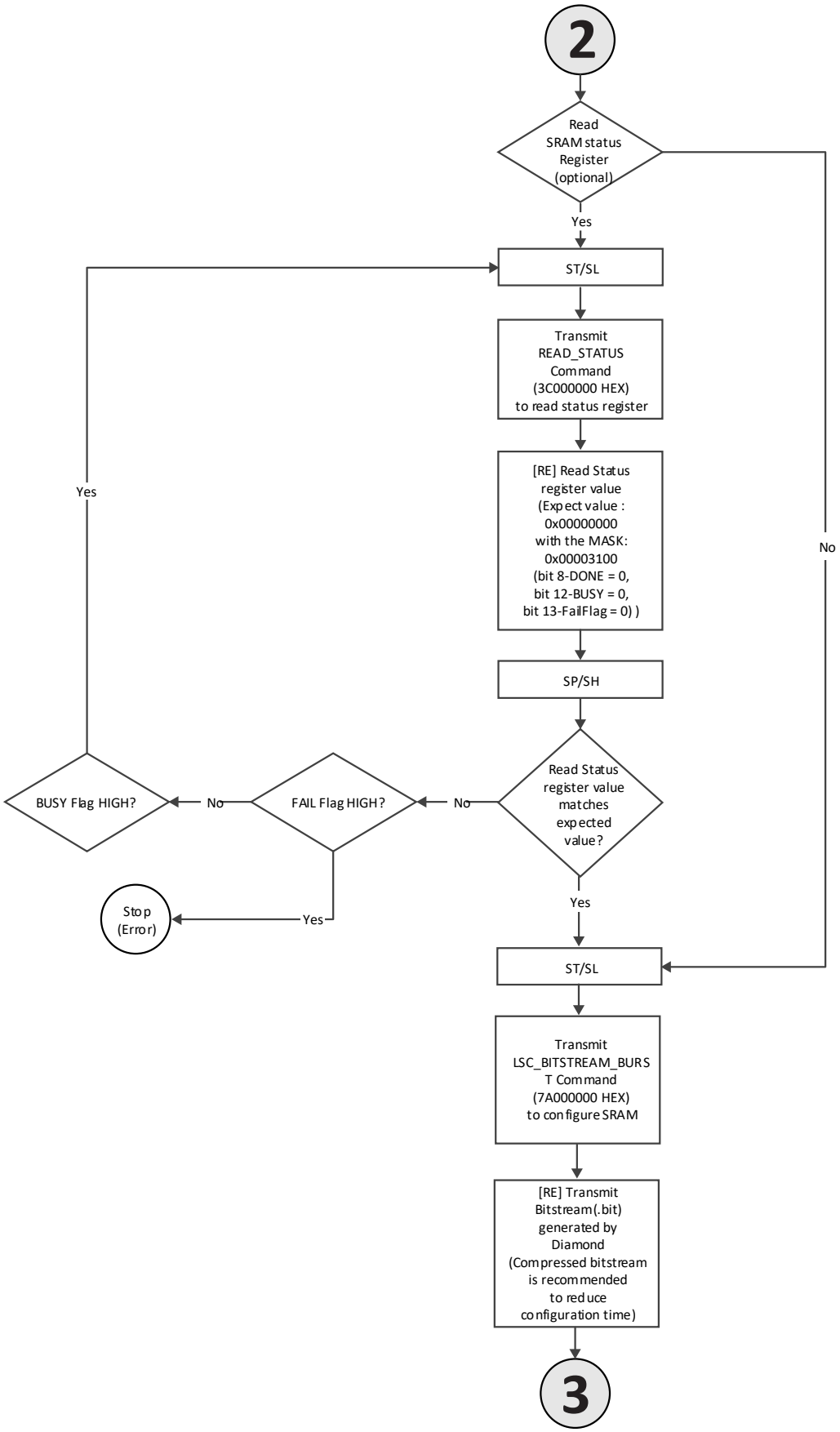
Figure 9.2. MachXO3 NVCM/Flash Memory Programming Flow

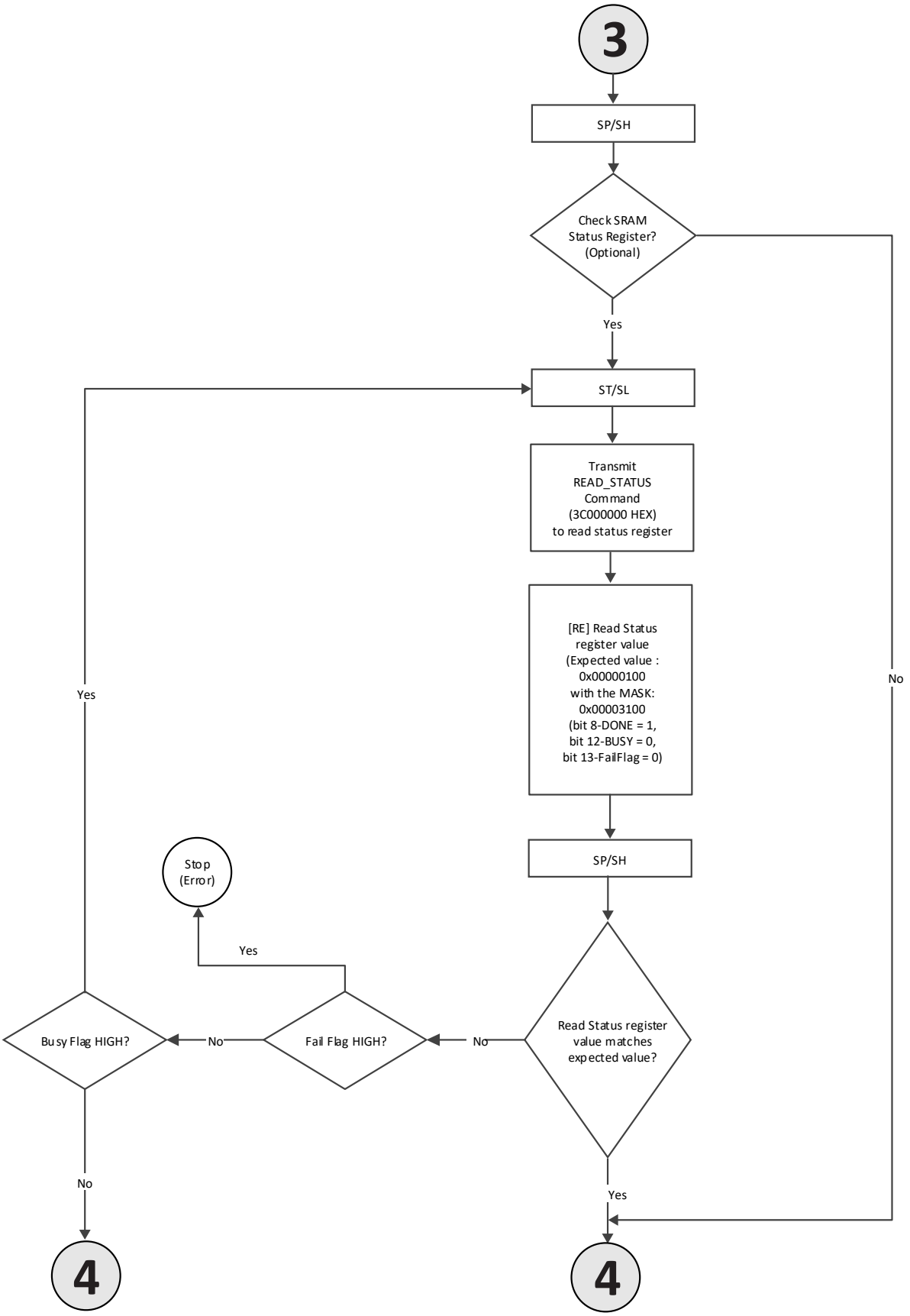
9.4. MachXO3 Slave SPI/I²C SRAM Configuration Flow

MachXO3 Slave SPI/I²C SRAM configuration requires a specific set of steps and timing. The flow chart in this section describes the command sequences and the timing required for successful SSPI/I²C SRAM configuration.









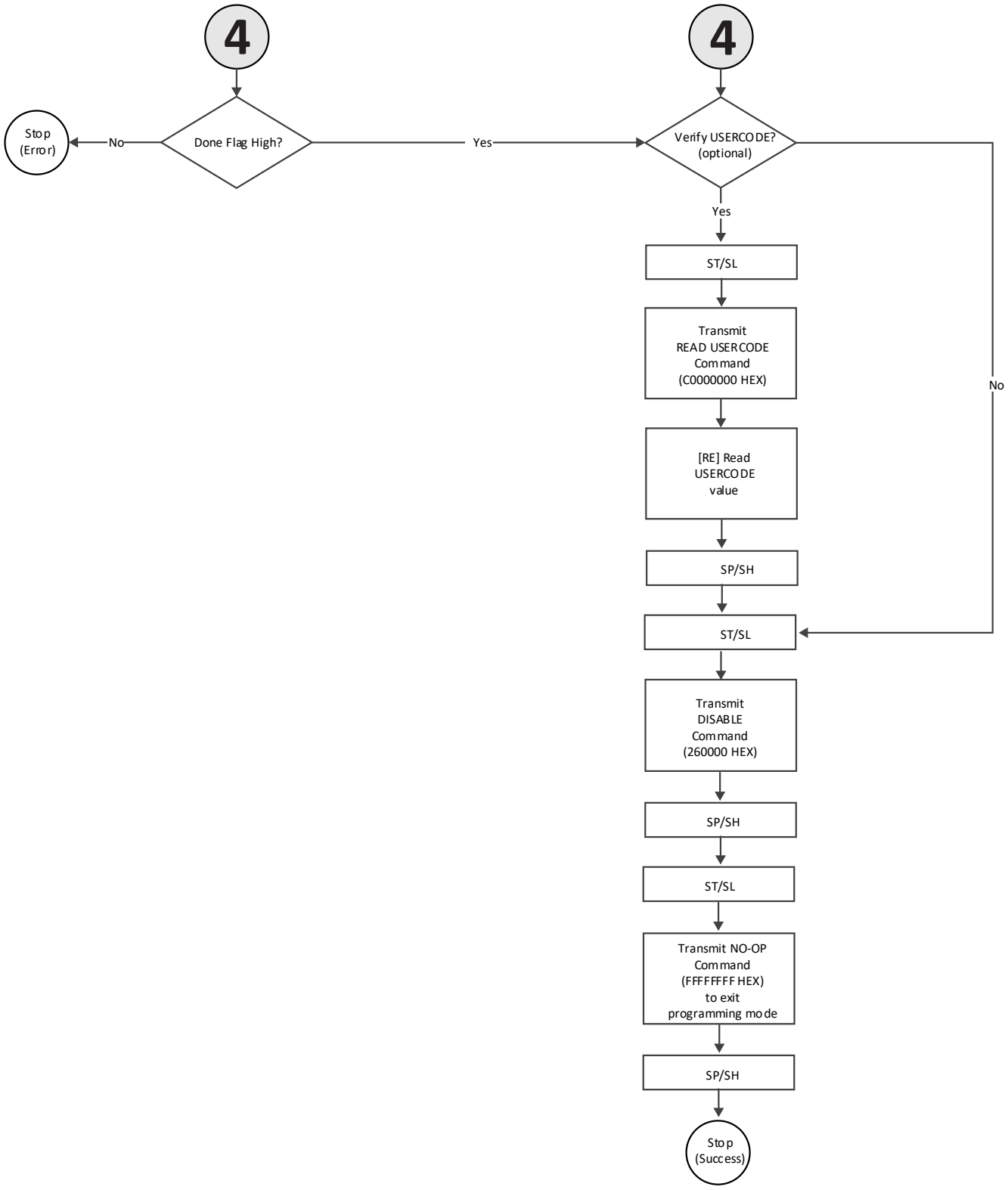


Figure 9.3. MachXO3 Slave SPI/I²C SRAM Configuration Flow

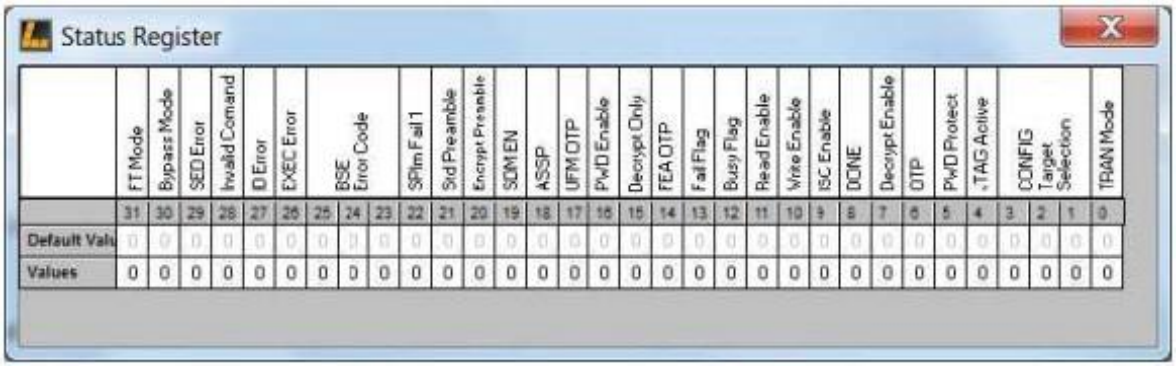


Figure 9.4. Status Register Value After Erase

Expected value from SO: 0x00000000 with the MASK: 0x00003100

(bit 8-DONE = 0, bit 12-BUSY = 0, bit 13-FailFlag = 0)

Mask = 0 means don't care

Mask = 1 means care

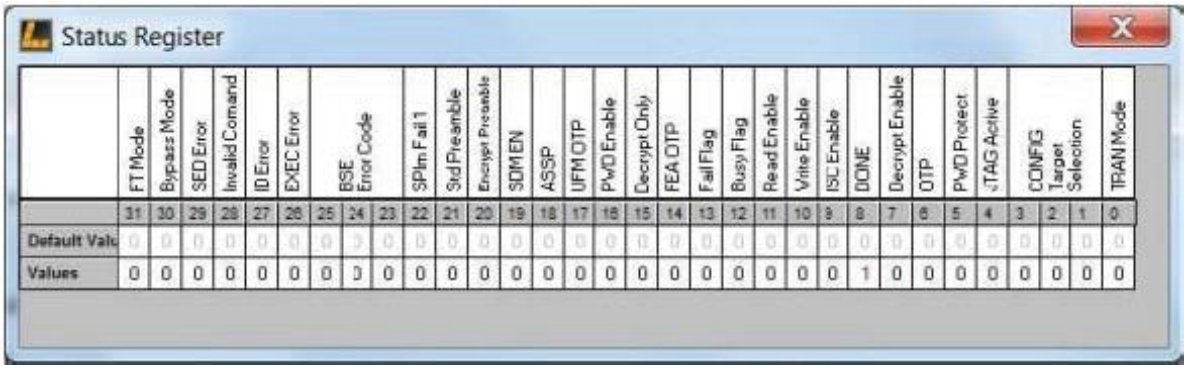


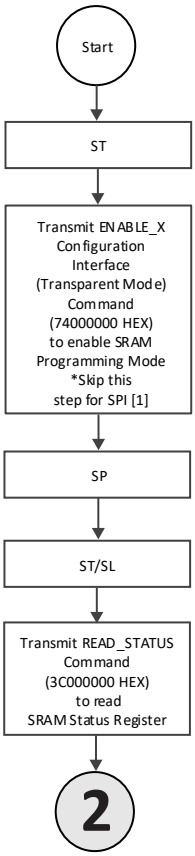
Figure 9.5. Status Register Value After Program

Expected value from SO: 0x00000100 with the MASK: 0x00003100

(bit 8-DONE = 1, bit 12-BUSY = 0, bit 13-FailFlag = 0)

Mask = 0 means don't care

Mask = 1 means care



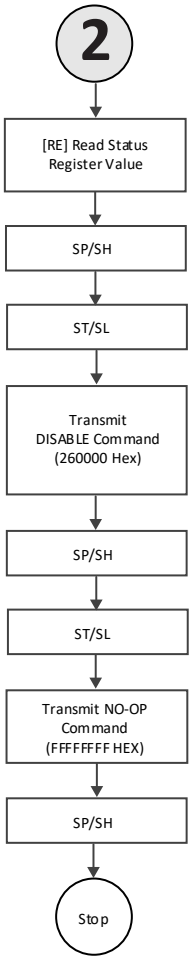


Figure 9.6. Slave SPI/I²C SRAM Read Status Register Flow

9.5. MachXO3 Programming Commands

Table 9.3. MachXO3 sysCONFIG Programming Commands

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Read Device ID [IDCODE_PUB]	0xE0	00 00 00	N/A	YY YY YY YY	YY characters represent the device-specific ID code.
Enable Configuration Interface (Transparent Mode) [ISC_ENABLE_X]	0x74	08 00 00[1]	N/A	N/A	Enable the Configuration Logic for device programming in transparent mode.[1]
Enable Configuration Interface (Offline Mode) [ISC_ENABLE]	0xC6	08 00 00[1]	N/A	N/A	Enable the Configuration Logic for device programming in Offline mode.[1]
Read Busy Flag [LSC_CHECK_BUSY]	0xF0	00 00 00	N/A	YY	Bit 1 0 7 Busy Ready
Read Status Register [LSC_READ_STATUS]	0x3C	00 00 00	N/A	YY YY YY YY	Bit 1 0 12 Busy Ready 13 Fail OK

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Erase [ISC_ERASE]	0x0E	0Y 00 00	N/A	N/A	Y = Memory space to erase Y is a bitwise OR Bit 1=Enable 16 Erase SRAM 17 Erase Feature Row 18 Erase NVCM0/CFG 19 Erase NVCM1/UFM
Erase NVCM1/UFM [LSC_ERASE_TAG]	0xCB	00 00 00	N/A	N/A	Erase the NVCM1/UFM sector only.
Reset Configuration NVCM/CFG Address [LSC_INIT_ADDRESS]	0x46	00 00 00	N/A	N/A	Set Page Address pointer to the beginning of the NVCM0/Configuration Flash sector.
Set Address [LSC_WRITE_ADDRESS]	0xB4	00 00 00	M0 00 PP PP	N/A	Set the Page Address pointer to the NVCM/Flash page specified by the least significant 14 bits of the PP PP field. The *M* field defines the NVCM/Flash space to access. Field 0x0 0x4 M NVCM0/CFG NVCM1/UFM
Program Page [LSC_PROG_INCR_NV]	0x70	00 00 01	YY * 16	N/A	Program one NVCM/Flash page. Can be used to program the NVCM0/CFG or NVCM1/UFM.
Reset NVCM1/UFM Address [LSC_INIT_ADDR_NVCM1/UFM]	0x47	00 00 00	N/A	N/A	Set the Page Address Pointer to the beginning of the NVCM1/UFM sector.
Program NVCM1/UFM Page [LSC_PROG_TAG]	0xC9	00 00 01	YY * 16	N/A	Program one NVCM1/UFM page.
Program USERCODE [ISC_PROGRAM_USERCODE]	0xC2	00 00 00	YY * 4	N/A	Program the USERCODE.
Read USERCODE [USERCODE]	0xC0	00 00 00	N/A	YY * 4	Retrieves the 32-bit USERCODE value.
Write Feature Row [LSC_PROG_FEATURE]	0xE4	00 00 00	YY * 8	N/A	Program the Feature Row bits.
Read Feature Row [LSC_READ_FEATURE]	0xE7	00 00 00	N/A	YY * 8	Retrieves the Feature Row bits.
Write FEABITS [LSC_PROG_FEABITS]	0xF8	00 00 00	YY * 2	N/A	Program the FEABITS.
Read FEABITS [LSC_READ_FEABITS]	0xFB	00 00 00	N/A	YY * 2	Retrieves the FEABITS.
Read NVCM/Flash [LSC_READ_INCR_NV]	0x73	M0 PP PP	N/A	See the Reading NVCM/Flash Pages section.	Retrieves PPPP count pages. Only the least significant 14 bits of PP PP are used. The *M* field must be set based on the configuration port being used to read the NVCM/Flash. 0x0 I^2C 0x0 or 0x1 JTAG/SSPI

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Read NVCM1/UFM [LSC_READ_NVCM1/UFM]	0xCA	M0 PP PP	N/A	See the Reading NVCM/Flash Pages section.	Retrieves PPPP count NVCM1/UFM pages. Only the least significant 14 bits of PP PP are used for the page count. The *M* field must be set based on the configuration port being used to read the NVCM1/UFM. 0x0 I^2C 0x0 or 0x1 JTAG/SSPI
Program DONE [ISC_PROGRAM_DONE]	0x5E	00 00 00	N/A	N/A	Program the DONE status bit enabling SDM.
Program OTP Fuses [LSC_PROG_OTP]	0xF9	00 00 00	UCFSUCFS	N/A	Makes the selected memory space One Time Programmable. Matching bits must be set in unison to activate the OTP feature. Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 NVCM0/CFG OTP NVCM0/CFG Writable 3, 7 NVCM1/UFM OTP NVCM1/UFM Writable
Read OTP Fuses [LSC_READ_OTP]	0xFA	00 00 00	N/A	UCFSUCFS	Read the state of the One Time Programmable fuses. Bit 1 0 0, 4 SRAM OTP SRAM Writable 1, 5 Feature Row OTP Feature Row Writable 2, 6 NVCM0/CFG OTP NVCM0/CFG Writable 3, 7 NVCM1/UFM OTP NVCM1/UFM Writable
Disable Configuration Interface [ISC_DISABLE]	0x26	00 00	N/A	N/A	Exit Offline or Transparent programming mode. ISC_DISABLE causes the MachXO3 to automatically reconfigure when leaving Offline programming mode. Thus, when leaving Offline programming mode, the Configuration SRAM must be explicitly cleared using ISC_ERASE (0x0E) prior to transmitting ISC_DISABLE. The recommended exit command from Offline programming mode is LSC_REFRESH (0x79), wherein ISC_ERASE and ISC_DISABLE are not necessary. See Figure 9.2.
Bypass [ISC_NOOP]	0xFF	FF FF FF	N/A	N/A	No Operation and Device Wakeup.
Refresh [LSC_REFRESH]	0x79	00 00	N/A	N/A	Force the MachXO3 to reconfigure. Transmitting a REFRESH command reconfigures the MachXO3 in the same fashion as asserting PROGRAMN.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Program SECURITY [ISC_PROGRAM_SECURITY]	0xCE	00 00 00	N/A	N/A	Program the Security bit (Secures NVCM0/CFG sector).[2]
Program SECURITY PLUS [ISC_PROGRAM_SECPLUS]	0xCF	00 00 00	N/A	N/A	Program the Security Plus bit (Secures NVCM/Flash Sectors).[2]
Read TraceID code [UIDCODE_PUB]	0x19	00 00 00	N/A	YY*8	Read 64-bit TraceID.
Configure SRAM [LSC_BITSTREAM_BURST]	0x7A	00 00 00	Compressed bitstream	N/A	Shift in bitstream (.bit) generated by Diamond. Recommend using compressed bit- stream to reduce configuration time. Number of bits varies depending on compression ratio.
Program Flash Protect Key [LSC_PROG_PASSWORD]	0xF1	00 00 00	YY*8	N/A	Program the 64-bit Password into the device.
Read Flash Protect Key [LSC_READ_PASSWORD]	0xF2	00 00 00	N/A	YY*8	Read the 64-bit Password from the device.
Shift Flash Protect Key [LSC_SHIFT_PASSWORD]	0xBC	00 00 00	YY*8	N/A	Present the 64-bit Password. When enabled (PWD_enable = 1), the write data is compared to the Password contained into the Feature Row. If the values match, the device is unlocked for programming and configuration operations. The device remains unlocked until a Disable Configuration command is received, a Refresh command is issued, or a power cycle event occurs.

Notes:

1. Transmits the command opcode and first two operand bytes when using the I2C port and SSPI port. The final operand byte must not be transmitted.
2. SECURITY and SECURITY PLUS commands are mutually exclusive.

9.6. Reading NVCM/Flash Pages

Reading the NVCM0/CFG and NVCM1/UFM pages requires a specific procedure. The NVCM0/CFG and NVCM1/UFM pages are accessible from any of the MachXO3's configuration ports. The JTAG and Slave SPI configuration ports all behave identically when performing read operations. The I^2C port requires a modified access protocol. A high-level representation of the data flow, by port, is shown in Figure 9.8.

All ports start the read process in the same way, by sending a Read NVCM/Flash/Read NVCM1/UFM command. The MachXO3 begins the read process once the command byte has been accepted by the configuration logic. The Page Address Pointer determines the first page returned from the MachXO3. For the first returned page to be valid (for example, for single-page read operations), a retrieval delay of 240 ns for MachXO3-2100 and bigger devices or 360 ns for MachXO3-1300 and smaller devices must be observed. The Retrieval delay time is from the end of the Command byte transmission to the end of the first Operand byte transmission See Figure 9.7. Note that for slower interface clock rates, 240 ns or 360 ns may be consumed entirely by the normal transmission of the first Operand and no additional delay may be necessary.

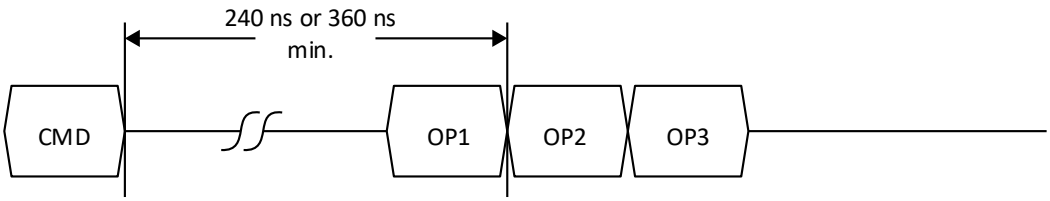
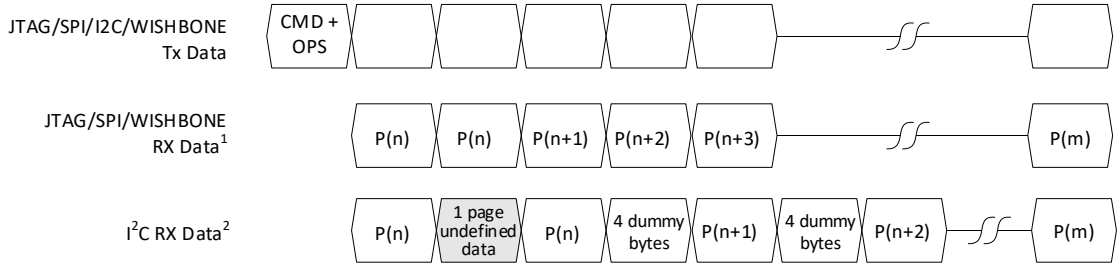


Figure 9.7. Retrieval Delay Timing Requirement for Single-Page Reads



Notes:
1. JTAG/SSPI must transmit data in order to read data back.
   The data sent by the JTAG/SSPI master is not specified (or don't care).
2. The I^2C must use RESTART between sending the CMD and reading the data.
   (Issuing a STOP terminates a CMD and resets the I^2C state machine.)

CMD + OPS = Read NVCM/CFG or Read NVCM1/UFM command byte + 3 operand bytes.

Figure 9.8. NVCM/Flash Page Command and Data Sequence

Figure 9.8 shows a multiple page read sequence. The Read NVCM/CFG/Read NVCM1/UFM command is transmitted to the MachXO3.

As shown in Figure 9.8, all interfaces return the page at the Page Address Pointer immediately. For single-page read operations, all configuration ports are allowed to terminate the read immediately following the transfer of the final byte of the first page. The I^2C interface differs only in the Read NVCM/CFG/Read NVCM1/UFM operand bytes.

Reading more than one page requires special handling. The multiple page read duplicates the page selected by the Page Address Pointer. The result of this behavior is that the page count must be one greater than the desired number of pages. For example, reading two pages requires the page count supplied in the Read NVCM/CFG/Read NVCM1/UFM command to be assigned a value of 3. If the Page Address Pointer is 0000, the MachXO3 returns three pages, Page 0, Page 0, and Page 1. A restriction must be observed when using the WISHBONE interface to read the NVCM or Flash. When reading 13 or more pages, the page count must be set to the maximum (16383 decimal or 0x3FFF). The user logic is not required to read this number of pages and may safely truncate the read operation after the desired number of pages have been read.

The I^2C interface has additional overhead when reading NVCM/Flash pages. Reviewing Figure 9.8 shows how the data is presented during a multiple page read request. When the page count is three, and the Page Address Pointer is 0000, the I^2C interface returns Page 0, 16 undefined bytes, Page 0, 4 dummy bytes, and Page 1. Reading the final four dummy bytes is optional.

References

- MachXO3 Family Data Sheet (FPGA-DS-02032)
- Using Hardened Control Functions in MachXO3 Devices Reference Guide (FPGA-TN-02064)
- MachXO3 Soft Error Detection/Correction Usage Guide (FPGA-TN-02062)
- Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02025)
- Using TraceID (FPGA-TN-02027)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 2.7, December 2021

Section	Change Summary
All	• Changed document title from MachXO3 Programming and Configuration Usage Guide to MachXO3 Programming and Configuration User Guide. • Minor adjustments in formatting, including changing 1K to 1k across the document.
Advanced Configuration Information	Updated note 1 in Table 9.3.

Revision 2.6, September 2021

Section	Change Summary
Configuration Modes	Corrected that the SSPI port is capable of directly accessing the configuration SRAM in section 5.4. Slave SPI Mode (SSPI).
Software Selectable Options	Updated the content of section 7.3.4 CONFIG_SECURE to clarify that the read-back of the UFM will also be blocked if the bitstream overflows into the UFM block.
Advanced Configuration Information	Added yes/no into Figure 9.2. MachXO3 NVCM/Flash Memory Programming Flow after the 'Program OTP Fuses?' step.

Revision 2.5, August 2021

Section	Change Summary
All	Corrected multiple occurrences of 'MachXO2' to 'MachXO3'.
Advanced Configuration Information	Fixed "1" to superscript in Table 9.3.

Revision 2.4, February 2021

Section	Change Summary
Acronyms in This Document	Updated content.
Configuration Process and Flow	Updated content, including Table 4.8, in Master and Slave SPI Configuration Port Pins section to change 1K pull-up resistor from Recommended to Required.
Software Selectable Options	• Updated Table 7.1 to add DONE and INITN settings. • Updated SDM Port to add DONE and INITN states, and change PROGRAM to PROGRAMN. • Updated MUX_CONFIGURATION_PORTS section to change Disable state to Enable and add the statement If the JTAGENB input pin is hard connected to GND on the PCB, this allows the MachXO3 to become a write one time device. In other application scenarios, you can control the JTAGENB to provide dynamic selection between the JTAG port and GPIO.

Revision 2.3, March 2020

Section	Change Summary
All	• Changed document number from TN1279 to FPGA-TN-02055. • Updated document template.
Disclaimers	Added this section.
Configuration Process and Flow	• Added footnote for Figure 4.1 and Figure 4.2. • Re-arranged bullet list in Wake-up section. • Updated Table 4.1. • Updated bullet list in Key Features section. • Updated content in INITN and DONE section.
WISHBONE Configuration Mode (MachXO3LF Only)	Added footnote on Table 5.6.
Software Selectable Options	Updated content of CONFIG_SECURE section.
Device Wake-up Sequence	Re-arranged bullet list in Wake-up Signals section.
Advanced Configuration Information	• Updated Figure 9.2 and Figure 9.8. • Updated Table 9.3.

Revision 2.2, August 2017

Section	Change Summary
Feature Row	Removed statement regarding programmable non-volatile memory space for Feature Row.
NVCM/Flash Programming	Revised 9400 and 6900 device densities data and footnote in Table 19, Number of Pages of NVCM/Flash Memory for the MachXO3 Family.

Revision 2.1, March 2017

Section	Change Summary
MachXO3L/LF Features	In the MachXO3L multi- time programmability feature, changed “two times” to “nine times”.
Bitstream/PROM Sizes	Revised Table 2, Maximum Configuration Bits. • Added 121 Ball Package to MachXO3L/LF-640E device. • Updated values.
Feature Row	• Changed “it can be reprogrammed up to eight times.” to “it can be programmed nine times.” • Changed “MachXO3L reserves eight times” to “MachXO3L reserves nine times”.

Revision 2.0, December 2016

Section	Change Summary
WISHBONE Configuration Mode (MachXO3LF Only)	Removed reference to Appendix C.
Reading NVCM/Flash Pages	Added content on restriction to be observed when using the WISHBONE interface to read the configuration flash or UFM.

Revision 1.9, September 2016

Section	Change Summary
Dual Boot Configuration Mode	Added paragraph on the use of Dual Boot Configuration Mode with MachXO3LF SED and SEC features.

Revision 1.8, April 2016

Section	Change Summary
Bitstream/PROM Sizes	Added bitstream sizes for MachXO3L/LF-9400 to Table 2, Maximum Configuration Bits.
Feature Row	Updated section. Added password security features to Table 4, MachXO3L Feature Row Elements.
Master SPI Configuration Mode (MSPI)	Removed "RC delay to PROGRAMN" workaround.
Dual Boot Configuration Mode	Mentioned DUAL- BOOTGOLDEN configuration in introductory paragraph.
Password	General update.
Bitstream Generation Options	Added DUALBOOT- GOLDEN feature.
Security Options	Added BACKGROUND_RECONFIG feature.
NVCM/Flash Programming	Updated Table 19, Number of Pages of NVCM/Flash Memory for the MachXO3 Family.
MachXO3 NVCM/Flash Programming Flow	Added Password shift decision to Figure 20, MachXO3 NVCM/Flash Memory Programming Flow.
MachXO3 Programming Commands	Added Password commands to Table 21, MachXO3 sysCONFIG Programming Commands.

Revision 1.7, February 2016

Section	Change Summary
Initialization	Changed tINTIL to tINITL.
Memory Space Accessibility	Changed sentence to "The JTAG port has the ability to access the two memory spaces in Offline and Transparent mode."
Self-Download Port Pins	Changed tINTIL to tINITL.
Feature Row	Added footnote 3 in Table 4, MachXO3 Feature Row Elements.
Dual Boot Configuration Mode	General content update.
Slave SPI Mode (SSPI)	• Changed sentence to "Reprogramming the NVCM/Flash can be done using offline or transparent operations." • Added paragraph.
I^2C Configuration Mode	Changed sentence to "Reprogramming the NVCM/Flash can be done in offline or in transparent operations."
JTAG Mode	Added Transparent Flash bullet.
NVCM/Flash Programming	Added transparent mode information.
MachXO3 NVCM/Flash Programming Flow	Revised Figure 20, MachXO3 NVCM/Flash Memory Programming Flow (Steps Start-1 and 3-4).
MachXO3 Programming Commands	Revised Table 21, MachXO3 sysCONFIG Programming Commands. • Added ISC_ENABLE_X command. • Added Transparent mode in ISC_DISABLE command notes.

Revision 1.6, April 2015

Section	Change Summary
Self-Download Port Pins	Added information on the DONE pin.
Master SPI Configuration Mode (MSPI)	Added information on delaying MachXO3L/LF POR. • Updated workaround solutions when SPI Flash POR is higher than the MachXO3L/LF POR. • Updated Figure 10, RC Delay.
MachXO3 Slave SPI/I2C SRAM Configuration Flow	• Revised Figure 21, MachXO3 Slave SPI/I2C SRAM Configuration Flow. • Added Figure 24, Slave SPI/I2C SRAM Read Status Register Flow.
Technical Support Assistance	Updated contact information.

Revision 1.5, March 2015

Section	Change Summary
All	• Product name/trademark adjustment. • Added MachXO3LF support. • Added UFM support.
MachXO3L/LF Features	Added MachXO3LF features.
Definition of Terms	Added Internal Flash Memory and Transparent Mode (MachXO3LF) definitions.
Memory Space Accessibility	In Table 1, Memory Space Accessibility of Different Ports, Internal WISHBONE (MachXO3LF) port information is added.
On-chip Flash Programming (MachXO3LF Only)	Added this section.
Bitstream/PROM Sizes	• Added information on storing configuration data in MachXO3LF Flash memory. • Added Figure 3, Flash Memory Space of a MachXO3LF Device. • Updated Table 2, Maximum Configuration Bits. Revised device names.
Feature Row	In Table 5, MachXO3L/LF Programming and Configuration Ports, Internal interface information is added.
Dual Boot Configuration Mode	Added MachXO3LF information.
WISHBONE Configuration Mode (MachXO3LF Only)	Added this section.
Bitstream Generation Options	Added CFG_EBRUFM (MachXO3LF) and CFGUFM (MachXO3LF) descriptions.
MachXO3 Slave SPI SRAM Configuration Flow	Added this section.
I^2C Configuration Mode	Added new EFB instantiation requirement for I2C configuration port access per Product Bulletin PB1412.
MachXO3 NVCM/Flash Programming Flow	Revised Figure 21, MachXO3 Slave SPI SRAM Configuration Flow: Added notes regarding EFB instantiation requirement for I^2C configuration port access.
MachXO3 Programming Commands	Revised Table 21, MachXO3 sysCONFIG Programming Commands. Added Configure SRAM [LSC_BITSTREAM_BURST] command.

Revision 1.4, October 2014

Section	Change Summary
Reading NVCM Pages	Added information on retrieval delay.

Revision 1.3, September 2014

Section	Change Summary
Feature Row	• Added Figure 3, Feature Row Example. • Added Table 3, Feature Row Option and Diamond Spreadsheet View. • Updated Table 4, MachXO3 Feature Row Elements. Changed I2C Slave Address feature to I^2C Programmable Primary Configuration Address and updated default mode state information.

Revision 1.2, July 2014

Section	Change Summary
Feature Row	Product name/trademark adjustment.
MachXO3L Features	Removed background programming information.
Definition of Terms	Removed background programming information.
Memory Space Accessibility	Removed reference to Transparent mode.
Bitstream/PROM Sizes	Updated Table 2, Maximum Configuration Bits. Added devices.
Feature Row	Updated Table 3, MachXO3L Feature Row Elements. Revised feature to Config NVCM OTP
sysCONFIG Ports	Updated Table 5, Default State of the sysCONFIG Pins. Revised PROGRAMN pin information.
sysCONFIG Pins	Updated Table 6, Default State in Diamond for Each Port. Revised SLAVE_SPI_PORT default state.
sysCONFIG Ports	Added information on SPI Flash POR and MachXO3L POR conditions. Added Figure 8, RC Delay.
Slave SPI Mode (SSPI)	Removed reference to transparent operations.
I^2C Configuration Mode	Revised NVCM and Feature Row programming information.
JTAG Mode	Removed reference to transparent NVCM programming.
Software Selectable Options	Added this section.
Advanced Configuration Information	Added this section.
TransFR Operation	Updated Figure 16, JEDEC File Example. Revised NOTE DEVICE NAME to LCMXO3L-1300E-6CSFBGA256.

Revision 1.1, April 2014

Section	Change Summary
Reading NVCM Pages	Corrected typos.

Revision 1.0, February 2014

Section	Change Summary
All	Initital release.

www.latticesemi.com