MachXO3D Programming and Configuration User Guide

Technical Note

FPGA-TN-02069-2.0

March 2026

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 all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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
AES	Advanced Encryption Standard
CCLK	Configuration Clock
CFG	Configuration Flash Memory
CRC	Cyclic Redundancy Check
DSA	Digital Signature Algorithm
EBR	Embedded Block RAM
ECDSA	Elliptic Curve Digital Signature Algorithm
EFB	Embedded Function Block
ESB	Embedded Security Block
FIFO	First In First Out
FSM	Finite State Machine
GOE	Global Output Enable
GPIO	General Purpose Input Output
GSR	Global Set Reset
GWDIS	Global Write Disable
HSE	Hardware Security Engine
I^2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
LUT	Look Up Table
MCLK	Master Clock
MSPI	Master Serial Peripheral Interface
POR	Power On Reset
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
Rx	Receiver
SDM	Self-Download Mode
SEC	Soft Error Correction
SED	Soft Error Detection
SFDP	Serial Flash Discoverable Parameters
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSPI	Slave Serial Peripheral Interface
SVF	Serial Vector Format
TAP	Test Access Port
TCK	Test Clock
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select
UDS	Unique Device Secret
UFM	User Flash Memory

1. Introduction

The MachXO3D™ device is an SRAM-based Programmable Logic Device (PLD) that includes an internal Flash memory, which allows it to operate similarly to a non-volatile device. The MachXO3D device provides a rich set of features for the programming and configuration of the FPGA. One key feature of the MachXO3D device is its embedded security capability. An embedded security block provides the authentication and encryption of the bitstream to prevent malicious attacks. MachXO3D also has flexible and robust access control for the configuration ports to enable the various programming and update needs. MachXO3D provides many user options to program and configure the device to address every customer's needs. Each of the available options is described in detail so that you can put together the programming and configuration solution that meets your needs.

The MachXO3D device contains two types of memory, SRAM and Flash. 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 FPGA's SRAM. The MachXO3D device provides an internal Flash memory that can be used to store the configuration data loaded into the MachXO3D SRAM.

2. MachXO3D Devices Features

Key programming and configuration features of MachXO3D devices are:

- Instant-on configuration from internal Flash – powers up in milliseconds.
- Up to 10,000 programming cycles for the internal configuration flash memory
- Single-chip, secure solution
- Optional bitstream authentication using ECDSA256
- Optional bitstream encryption using AES256
- Multiple hard and soft lock controls for the Flash/SRAM access from the configuration port or internal logic
- Multiple programming and configuration interfaces:
  - 1149.1 JTAG
  - Self-download
  - Slave SPI
  - Master SPI
  - Dual Boot
  - I^2C
  - WISHBONE bus
- Programming and configuration ports:
  - Slave SPI
  - Master SPI
  - I^2C
  - JTAG
  - Internal WISHBONE
- User Flash Memory (UFM) for non-volatile data storage:
  - Configuration Flash memory overflow
  - EBR initialization data
  - Application specific data
- Transparent programming of non-volatile memory
- On-chip dual boot
- Optional multi-boot with external SPI memory
- Optional Bitstream compression
- TransFR capability
  - Leave-alone I/O (non-JTAG mode)
- Soft Error Detect (SED) support

3. Definition of Terms

This document uses the following terms to describe common functions:

- AES – Advanced Encryption Standard is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.
- BIT – The BIT file is the configuration data for the MachXO3D 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 MachXO3D SRAM memory cells.
- Configuration data – This configuration data is the data read from the non-volatile memory and loaded into the FPGA SRAM configuration memory. This is also referred to as a bitstream, or device bitstream.
- Configuration mode – The configuration mode defines the method the MachXO3D device uses to acquire the configuration data from the non-volatile memory.
- Digest – A message digest is a cryptographic hash function containing a string of digits created by a one-way hashing formula.
- ECDSA – Elliptic Curve Digital Signature Algorithm is the elliptic curve analog of the DSA.
- Internal flash memory – On-die, non-volatile flash-type memory. The MachXO3D device contains multiple flash sectors for FPGA configuration image storage (up to 2), general-purpose user flash (up to 4 sectors), and device feature definitions.
- JEDEC – The JEDEC file contains the configuration data programmed into the MachXO3D Configuration 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 MachXO3D device 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 MachXO3D devices 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.
- Signature – A digital signature guarantees the authenticity of an electronic document or message in digital communication and uses encryption techniques to provide proof of original and unmodified documentation.
- Transparent mode – Transparent mode is used to update the Configuration Flash and the User Flash Memory while leaving the MachXO3D devices in user mode.
- User mode – The MachXO3D device is in user mode when configuration is complete, and the FPGA is performing the logic functions it is programmed 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.

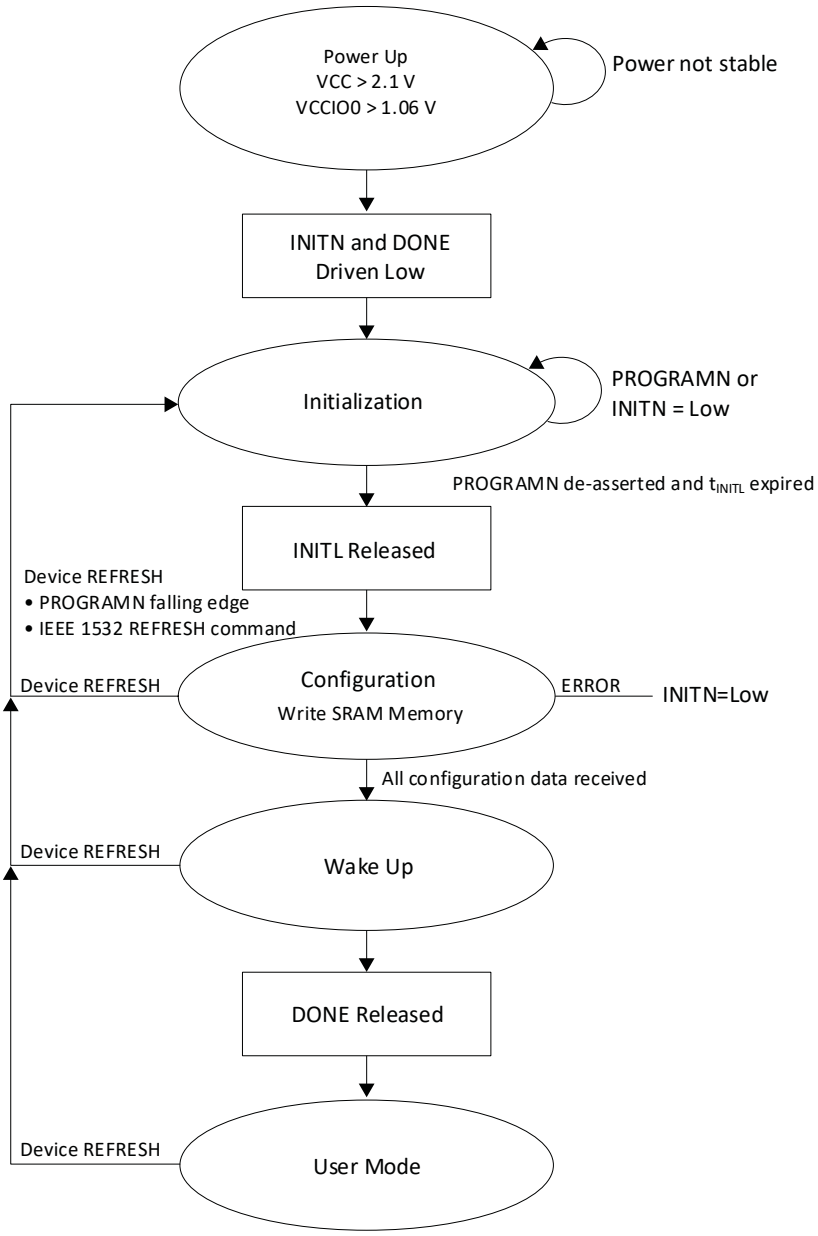


Figure 4.1. Configuration Flow

The MachXO3D 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 MachXO3D device internal 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

For the MachXO3D device 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 state becomes indeterminate.

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:

- VCC > 2.1 V
- VCCIO0 > 1.06 V

When these conditions are met, the POR circuit releases an internal reset strobe, allowing the device to begin its initialization process. The MachXO3D device 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.

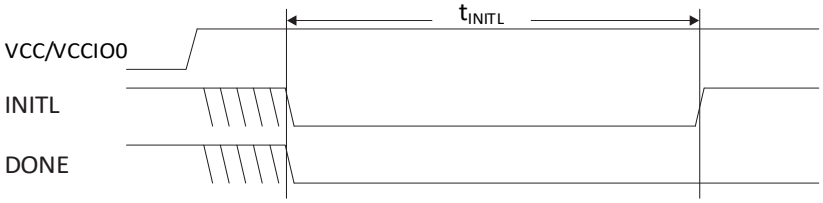


Figure 4.2. Configuration from Power-On-Reset Timing

4.2. Initialization

The MachXO3D device enters the memory initialization phase immediately after the POR circuit drives the INITN and DONE status pins low. The purpose of the initialization state is to clear all the SRAM memory inside the FPGA.

The FPGA remains in the initialization state until all 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 MachXO3D device is part of a chain of devices, each device carries 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 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 can accept the configuration bitstream created by the Lattice Diamond® development tools.

The MachXO3D device begins fetching configuration data from non-volatile memory. Based on the user option, the data can be decrypted and/or authenticated. The memory used to configure the MachXO3D device is either internal memory or an external SPI flash. The MachXO3D device does not leave the configuration state if there are no memories with valid configuration data or the data fails to pass authentication. It is necessary to program the non-volatile memory internal or attached to the FPGA, or to program it using the JTAG port with correct data.

During the time the FPGA receives its configuration data, the INITN control pin takes on its final function. INITN is used to indicate that an error exists in the configuration data. When INITN is high, configuration proceeds without issue. If INITN is asserted low, an error occurrs and the FPGA fails to operate.

4.4. Wake-up

Wake-up is the transition from configuration mode to user mode. The MachXO3D device's fixed four-phase wake-up sequence starts when the device correctly receives all 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 those sequences four controls. The four control strobes are:

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

The first phase of the wake-up process is for the MachXO3D device to release the Global Output Enable. When it is asserted, the FPGA I/O is permitted 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 with 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. 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 MachXO3D device from proceeding. Only after the external DONE, if enabled, is active high does the final wake-up phase complete. Wake-up completes uninterruptedly 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 MachXO3D device enters user mode immediately after the wake-up sequence has been completed. User mode is the point in time when the MachXO3D device begins performing the logic operations you designed. The MachXO3D device 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 MachXO3D remains in operation until it is actively cleared, or power is lost. Several methods are available to clear the internal configuration memory of the MachXO3D device:

- Cycle power to the MachXO3D device.
- Assert the PROGRAMN input.
- Issue the REFRESH command using a configuration port.

Any active configuration port can be used to send a REFRESH command. Invoking one of these methods causes the MachXO3D device to drive INITN and DONE low. The MachXO3D device enters the initialization state as described earlier.

4.7. Memory Space Accessibility

The internal Flash memories and SRAM of the MachXO3D device can be read and written. Each port on the MachXO3D device has a different level of access to each memory space. Table 4.1 provides a cross-reference of the MachXO3D device ports and the memory space they can access.

As shown in Table 4.1, the JTAG, SPI, and I2C can read and write both internal memory spaces while wishbone can only access the internal flash and not the SRAM.

Table 4.1. Memory Space Accessibility of Different Ports

Port	On-Chip Flash		SRAM	
	Read	Write	Read	Write
JTAG	Yes	Yes	Yes	Yes
SPI Port	Yes	Yes	Yes	Yes
I^2C Port	Yes	Yes	Yes	Yes
Internal WISHBONE	Yes[1]	Yes[1]	No	No

Note:

1. In Transparent mode only.

When the lock policy is implemented, it further controls accessibility. Refer to the Lock Bits and Lock Control Policy section for more details.

4.8. On-chip Flash Programming

On-chip Flash is programmed with different programming modes. These programming modes are discussed in the following sections. Within the different programming modes, there are two methods of programming the on-chip Flash: Offline and Background programming.

4.8.1. Offline Programming

Offline programming requires the device to enter programming mode. When in programming mode, the device stops working until the programming is completed. When using Lattice 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.

When bitstream authentication and encryption are enabled, the programming needs to get through the decryption with a pre-programmed AES key and pass the bitstream signature checking with a pre-programmed ECDSA public key. See MachXO3D Embedded Security Block (FPGA-TN-02091) for details.

4.8.2. Background Programming

Background programming 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 instructions. 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.

4.9. Bitstream/PROM Sizes

The MachXO3D device is an SRAM-based FPGA. The SRAM configuration memory must be loaded from a non-volatile memory that can store all 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 MachXO3D 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.

Storing configuration data in the MachXO3D device's internal Flash memory has special considerations. The flash memory in the MachXO3D device provides three types of independent sectors to store the configuration data.

The first type of sector is dedicated to holding compressed configuration data, and is called Configuration Flash, or CFG. There are two CFG sectors, CFG0 and CFG1, in the MachXO3D device. Each CFG0 and CFG1 sector can store a whole bitstream.

The second type of sector is called User Flash Memory (UFM). There are four UFM sectors, UFM0, UFM1, UFM2, and UFM3, in the MachXO3D device. The UFM0 and UFM1 provide three different functions, such as additional configuration flash storage for large configuration data images, storage for EBR contents, and use as general-purpose flash memory. For UFM2 and UFM3, they can only be used as general-purpose flash memory.

The third type of sector is the Feature Row.

Figure 4.3 shows the flash memory space of a MachXO3D device. CFG0 and CFG1 have the same memory size. UFM0 and UFM1, likewise, have the same size. In this figure, M, N, O, and P represent the total page numbers for CFG0/1, UFM0/1, UFM2, and UFM3, respectively. Refer to *Table 9.1 UFM Resources in MachXO3D Devices* and *Table 10.1 Configuration Flash Resources in MachXO3D Devices* of the Using Hardened Control Functions in MachXO3D (FPGA-TN-02117) for their specific values in each MachXO3D device density.

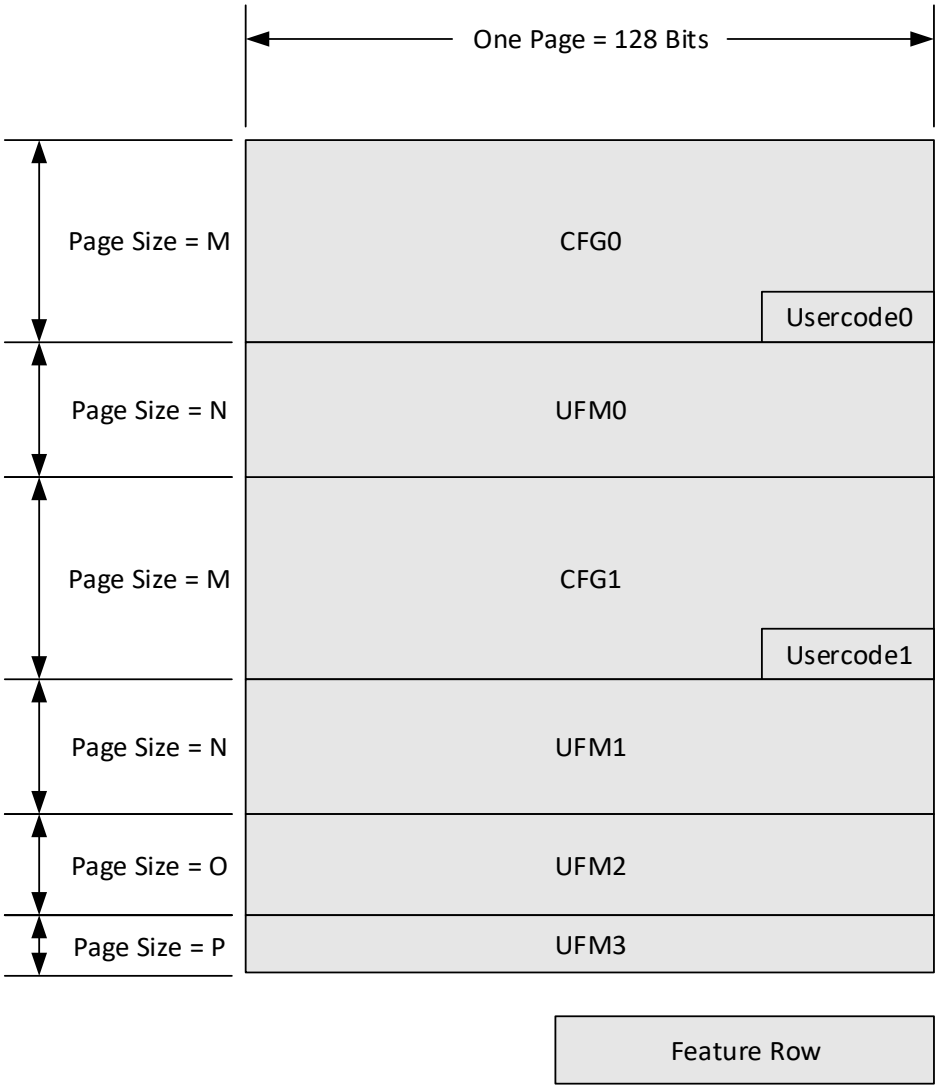


Figure 4.3. Flash Memory Space of a MachXO3D Device

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. In the MachXO3D device, the configuration data in the CFG0 can only cross into the UFM0, while the configuration data in the CFG1 can only cross into the UFM1. It is also possible, but unlikely, that the configuration data can get too large for the internal Flash memory altogether. If 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 for generating the configuration data to allow or prevent the overflow into the UFM.

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 Flash	Units
MachXO3D-4300	0.93	1.02	0.80	Mb
MachXO3D-9400	2.11	2.54	2.0	Mb

4.10. Configuration/Boot-up Time

Configuration time for MachXO3D can be estimated by calculating the bitstream loading time from either external memory or internal flash, together with the time needed for bitstream decryption and authentication by Configuration Engine if you have turned on the bitstream security. The configuration time depends on the size of the bitstream, and the type of security features selected. The estimated configuration time can be calculate based on the equation below.

Configuration Time = (Bitstream Size / Configuration Clock Frequency) + (Bitstream Authentication Clock Cycle / Security Block Clock Frequency) + (Bitstream Decryption Clock Cycle / Security Block Clock Frequency)

Where

Bitstream Size	Bitstream size can be obtained in section 4.9 Bitstream/PROM Sizes. For worst case scenario, select the Uncompressed Bitstream Size With EBR.
Configuration Clock Frequency	Configuration Clock Frequency can be found in FPGA-DS-02026 MachXO3D Family Datasheet under Section 3.22, 3.23, and 3.24 depending on the selected configuration mode (MSPI, SSPI, JTAG or I^2C). If the configuration is from internal flash, refer to the Flash Download Time in FPGA-DS-02026 MachXO3D Family Datasheet under section 3.21. For worst case scenario, MSPI clock should minus 10% from the F_{max} to cater the process variation on the internal oscillator. If you are using more than 1 data lane for MSPI/SSPI, configuration clock frequency needs to be multiplied with the number of data lanes.
Bitstream Authentication Clock Cycle	Bitstream Authentication Clock Cycle is needed for the Security Block to carry out ECDSA verification, which will be 6.7 million cycles, and 9.2 million cycles depends on whether turning the Clock Randomization Feature OFF or ON respectively. Turning on the Clock Randomization will increase the ECDSA computation time.
Security Block Clock Frequency	Security Block Clock Frequency is constant across device density, which is 90 MHz. For worst case scenario, the clock should minus 10% from 90 MHz to cater the process variation on the internal oscillator
Bitstream Decryption Clock Cycle	Bitstream Decryption Clock Cycle is needed for Security Block to decrypt the incoming encrypted bitstream. Bitstream Decryption Clock Cycle = (Bitstream Size in Bytes divided 16 Bytes) × 15 clock cycles.

4.11. Feature Row

The MachXO3D device includes a feature row to control FPGA resources. The Feature Row permits more flexibility in selecting the functions available for configuration, increasing the number of available I/Os on the device, and eliminating the need to make changes to your hardware. Feature Row can be erased or programmed independently.

MachXO3D Feature Row is used to determine how the MachXO3D SRAM configuration memory is loaded. When Feature Row is erased, Feature Row sets its value back to the Hardware (HW) Default Mode state. The contents of Feature Row are typically specified using the Diamond Spreadsheet View. They can also be manually modified using the Programming File Utility under Tools > Feature Row Editor.

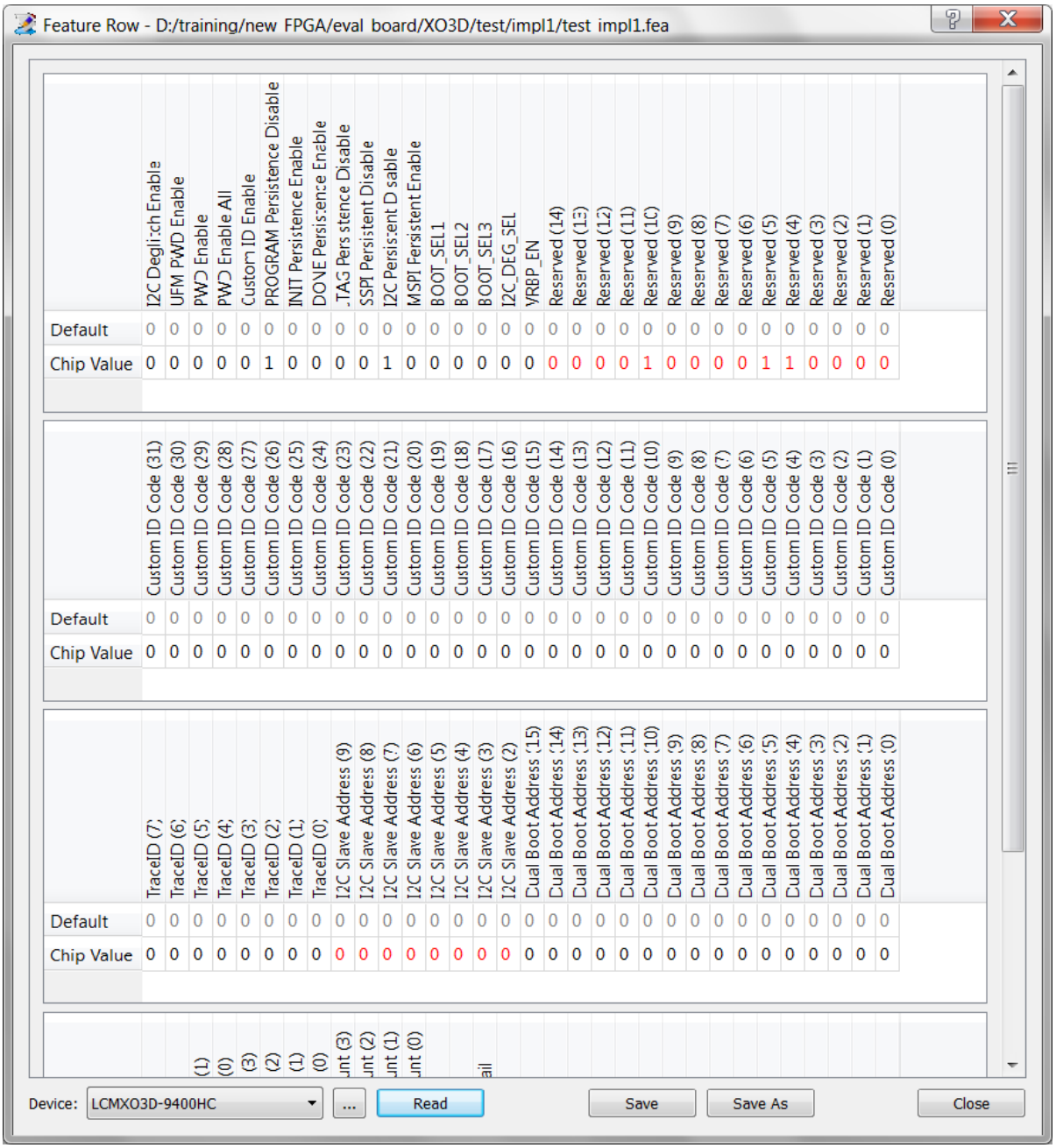


Figure 4.4. Feature Row Example

Functions controlled by the Feature Row and their default values for MachXO3D devices are shown in Table 4.3.

Table 4.3. MachXO3D Feature Row Elements

Feature	Software Default Mode State (Programmed)	Hardware Default Mode State (Erased)
BOOT_SEL[3:1]	011	000
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
my_ASSP Enable	OFF	OFF
Password Enable	OFF	OFF
Password Enable All	OFF	OFF
UFM Password Enable	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 that 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 MachXO3D device 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 MachXO3D device is never allowed to complete its configuration process.

Feature Row settings are specified using the Diamond Spreadsheet View, which allows you to edit the configuration settings for the MachXO3D device and then save your settings in the Lattice Preference File (LPF). These settings are applied to the MachXO3D device configuration data during the Map, Place, and Route build phases. Alternately, the feature row of a device can be modified using the Program Feature Row utility in Diamond Programmer.

Key Features:

- Not intended to be modified in the field, only for development.
- A change in feature row settings may cause active configuration ports to become unavailable.
- Can be altered using Diamond Spreadsheet View.
- Can be programmed under Program Feature Row in Diamond Programmer.

4.12. Lock Bits and Lock Control Policy

MachXO3D devices contain security bits that, when set, can control the access to the SRAM configuration and flash spaces. The MachXO3D device provides read, program, and erase permission control for each Flash sector, as well as the SRAM.

To support this, three security bits are deployed in each sector: SEC_PROG, SEC_READ, and SEC_ERASE. Once the bit is set, the corresponding operation, which is read, program, or erase, is prohibited. For the SRAM, once the SEC_PROG is set, beside blocking the configuration of the SRAM with the external configuration ports (JTAG, SSPI, and I^2C), it also stops the Master SPI interface from booting the device from external SPI FLASH PROM. This means that the device can only be booted up from internal Flash Configuration Memory.

The MachXO3D device also provides hard lock and soft lock modes for flexibility in permission control. In Soft Lock mode, access from user logic through the internal WISHBONE bus is not prohibited. This applies even if lock control bits are set for the external configuration ports (JTAG, SSPI, and I^2C). Also, user logic can alter the shadow register of access control bits to allow external configuration ports (JTAG, SSPI and I^2C) to access Flash memory. In Hard Lock mode, access from both the external configuration ports and the internal WISHBONE bus is prohibited. Also, user logic cannot alter the status of access control bits. The SEC_HLOCK bit is used to choose between soft lock and hard lock modes.

With the different combinations of the four security bits mentioned above, different policies can be created for each flash sector and the SRAM. Table 4.4 shows seven policies. Each line stands for one policy. *1* means locked, and *0* means unlocked. The polices shown in Table 4.4 are for reference purposes only. They are not limited to those listed. Desired policies must be set into the device using these four security bits. Take one example as Policy 6: SEC_HLOCK is set, so neither the external configuration ports nor the internal Wishbone bus can access and change the policy security bits. And since SEC_PROG, SEC_READ, and SEC_ERASE are all set, neither external configuration ports nor the internal WISHBONE bus can do anything to the Flash.

Table 4.4. MachXO3D Device Security Lock Control Bits

Mode	Policy #	Security Bits In Lock Policy Row for Each Sector				External CFG Port			Internal WISHBONE Bus		
		SEC_HLOCK	SEC_READ	SEC_PROG	SEC_ERASE	READ	PROG	ERASE	READ	PROG	ERASE
No Security	0	0	0	0	0	Y[1]	Y[1]	Y[1]	Y[2]	Y[2]	Y[2]
Soft Lock	1	0	1	0	0	N[3]	Y[1]	Y[1]	Y[2]	Y[2]	Y[2]
	2	0	1	1	0	N[3]	N[3]	Y[1]	Y[2]	Y[2]	Y[2]
	3	0	1	1	1	N[3]	N[3]	N[3]	Y[2]	Y[2]	Y[2]
Hard Lock	4	1	1	0	0	N[4]	Y[2]	Y[2]	N[4]	Y[2]	Y[2]
	5	1	1	1	0	N[4]	N[4]	Y[2]	N[4]	N[4]	Y[2]
	6	1	1	1	1	N[4]	N[4]	N[4]	N[4]	N[4]	N[4]

Notes:

1. Accessible, but may be altered through the WISHBONE bus.
2. Accessible.
3. Not accessible but may be altered through the WISHBONE bus.
4. Not accessible.

A typical usage in the Golden/Update image policy is:

1. Golden Image 0: It is hard locked as using policy 6.
2. User Working Image 1: It is soft locked as using policy 3.
3. To update Image 1, the command from the external configuration port is passed to the soft IP in the fabric. The fabric security IP validates and authenticates the external port lock/unlock request. If authenticated, the fabric uses the internal WISHBONE bus to unlock Image 1.
4. After a successful update and Image 1 passes the audit, the external configuration port can request the fabric IP to relock Image 1 to prevent it from being updated by any external activity.

The MachXO3D device also provides a set of permission control settings to disable access from the separate external configuration ports, such as JTAG, Slave SPI and Slave I^2C. These configuration ports support both hard lock and soft

lock modes. Each port has its own control bits. Any command or part of commands sent by these ports can be blocked with the settings of the control bits.

All MachXO3D security lock control bits can be set through the external configuration port using the Diamond Programmer tool or through the internal WISHBONE bus using user logic.

4.13. sysCONFIG Ports

Table 4.5. MachXO3D Device 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

4.14. sysCONFIG Pins

The MachXO3D device 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 JTAG, SSPI, I^2C, and 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 MachXO3D device is in the Feature Row HW Default Mode state. One example is driving PROGRAMN with an active low signal after the MachXO3D device is in the Feature Row HW Default Mode state. Failure to reprogram the feature row with PROGRAMN disabled prevents the FPGA from configuring and entering user mode.
- Use care when using JTAGENB to selectively enable and disable the JTAG port. 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. An HW default Mode Feature Row device has the JTAG, SPI Slave, and I^2C ports enabled. Upon entry to user mode, the MachXO3D device, the default state of the SSPI, and the I^2C sysCONFIG pins become general-purpose I/O. This means you lose the ability to program the MachXO3D device 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 that you take this into consideration when provisioning other logic attached to Bank 0.

The function of each sysCONFIG pin is described in detail in Table 4.6 and Table 4.7.

Table 4.6. Default State of the sysCONFIG Pins

Pin Name	Associated sysCONFIG Port	Pin Function in Feature Row Erased Mode (Configuration/Hardware Default Mode)	Pin Direction (Configuration Mode)	Default Function in User Mode (Software 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/D0	SSPI/MSPI[1]	SSPI	Input	SSPI
SO/SPISO/D1	SSPI/MSPI[1]	SSPI	Output	SSPI
CSSPIN	MSPI	I/O	I/O with weak pull up	User-defined I/O
SCL/D2	I^2C[1]	I^2C	Bidirectional	User-defined I/O
SDA/D3	I^2C[1]	I^2C	Bidirectional	User-defined I/O

Note:

1. SI/SISPI/D0 and SO/SPISO/D1 are used for Dual MSPI read mode. SI/SISPI/D0, SO/SPISO/D1, SCL/D2, and SDA/D3 are used for Quad MSPI read mode.

Table 4.7. sysCONFIG Port Default Settings in Diamond

sysCONFIG Port	Diamond Default[1]
SDM_PORT	DISABLE
SLAVE_SPI_PORT	ENABLE
I2C_PORT	DISABLE
MASTER_SPI_PORT	DISABLE
JTAG_PORT	ENABLE

Note:

1. This default setting can be modified in the Diamond Spreadsheet View, Global Preferences tab.

4.14.1. 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 MachXO3D device from leaving the initialization phase. The PROGRAMN has a minimum pulse width assertion period, t_{PRGMJ} (Figure 4.6), for it to be recognized by the FPGA. You can find this minimum time in the sysCONFIG Timing Specification section of the MachXO3D Family Data Sheet (FPGA-DS-02026).

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 restarts the configuration cycle.
- Asserting PROGRAMN on a device in the 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 it is 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 period. High to low PROGRAMN assertions more frequently prevent the MachXO3D device from configuring, causing the FPGA to remain in a continuous RESET condition. See Figure 4.5.

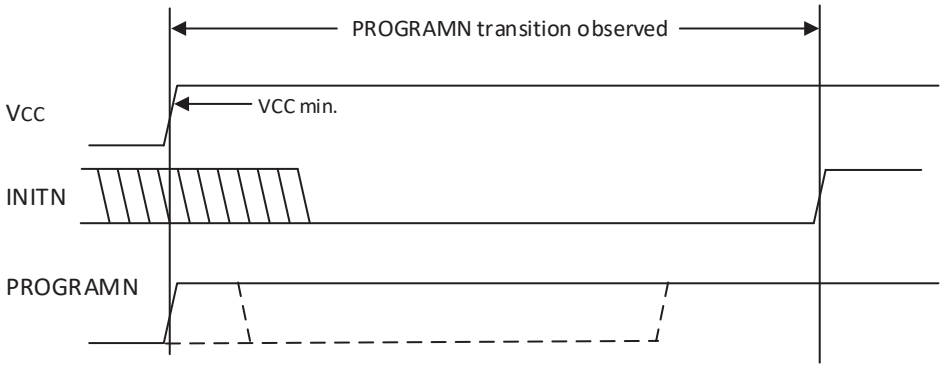


Figure 4.5. Period PROGRAMN is Always Observed

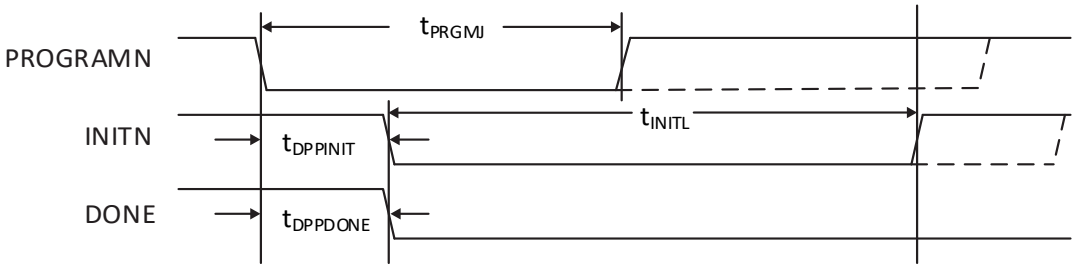


Figure 4.6. Configuration from PROGRAMN Timing

INITN

The INITN pin is a bidirectional open-drain control pin. It has the following functions:

- After power is applied, after a PROGRAMN is asserted, or after a REFRESH command is transmitted, INITN 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 elapses, the INITN pin is deasserted, which is active high, to indicate that the MachXO3D device is ready for its configuration bits. The MachXO3D device begins loading configuration data from either the internal flash or an external SPI flash.
- INITN can be asserted low by an external agent before the t_{INITL} time elapses 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 the configuration data is being read. Once t_{INITL} elapses and the INITN pin goes high, any subsequent INITN assertion signals that an error is detected by the MachXO3D device during configuration.

The following conditions cause INITN to become active, indicating the initialization state is active:

- Power is applied.
- PROGRAMN falling edge occurs.
- The IEEE 1532 REFRESH command is 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.

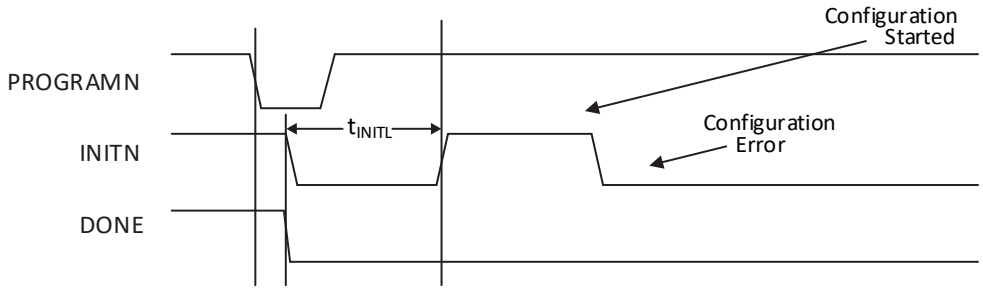


Figure 4.7. Configuration Error Notification

The INITN pin of a MachXO3D device is not visible externally 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 MachXO3D device 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 the configuration when the following happens:

- The bitstream CRC error is detected.
- The invalid command error detected.
- A time-out error is encountered when loading from the on-chip Flash.
- The program done command is not received when the end of on-chip SRAM configuration or on-chip 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. DONE is first able to indicate entry into user mode only after an internal DONE bit is asserted. The internal DONE bit defines the beginning of the FPGA wake-up state.

The DONE output pin is controlled by the SDM_PORT configuration parameter that is modified in the Diamond Spreadsheet View. By default, the DONE pin is a general-purpose I/O when the MachXO3D device is in the Feature Row HW Default Mode state. The default mode causes the MachXO3D device to automatically sequence through the wake-up sequence after the internal DONE bit is asserted. The FPGA does not stall waking up, waiting for the DONE pin to be asserted high.

The FPGA can be held from entering user mode indefinitely by having an external agent keep the DONE pin asserted low. 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.

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.

Sampling the DONE pin is a way for an external device to tell if the FPGA configuration is complete. 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, the DONE pin takes on the behavior defined by the SDM_PORT setting in the Feature Row. The DONE pin is also pulled high when the FPGA is in the Feature Row HW Default Mode state. This behavior can make a part appear to be successfully configured for other logic monitoring the DONE pin.

4.14.2. 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 MachXO3D configuration logic to a slave SPI PROM. A 1 kΩ pull-up resistor is required on MCLK when master SPI configuration port is enabled.
CSSPIN	CSSPIN	Output	Chip select used to enable an external SPI PROM containing configuration data. A 10-kΩ pull-up resistor is recommended on CSSPIN.
SI/SISPI/D0	SISPI, D0	Output/Input	SISPI carries output data from the MachXO3D configuration logic to the slave SPI PROM.
SO/SPISO/D1	SPISO, D1	Input	SPISO carries output data from the slave SPI PROM to the MachXO3D configuration logic.
D2	D2	Input	Data input from SPI Flash in QUAD read mode.
D3	D3	Input	Data input from SPI Flash in QUAD read mode.
SN	SN, I/O	Input	MachXO3D 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 MachXO3D Configuration Logic.
SI/SISPI/D0	SI	Input	SI carries output data from the external SPI master to the MachXO3D Configuration Logic
SO/SPISO/D1	SO	Output	SO carries output data from the MachXO3D configuration logic to the external SPI master
SN	SN	Input with weak pull-up	MachXO3D 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 as follows:

- The MachXO3D MCLK/CCLK pin's default state in the Feature Row HW Default Mode state acts as the configuration clock, CCLK. This allows an external SPI master controller to program the MachXO3D device. The maximum CCLK frequency and the data setup/hold parameters are found in the sysCONFIG Port Timing Specifications of the MachXO3D Family Data Sheet (FPGA-DS-02026). The feature row must be configured to enable the slave SPI port if you want to use the port to reprogram the MachXO3D device after it enters the user mode.
- The MCLK/CCLK pin functions as a master clock (MCLK) when the MachXO3D device is configured in dual boot or external boot mode. A 1 kΩ 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 MachXO3D device's Master SPI Configuration port. MCLK actively drives until all the configuration data is received. When the MachXO3D device 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 MachXO3D device 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. MachXO3D 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 MachXO3D device accepts the new MCLK_FREQ value, the MCLK output begins driving the selected frequency. When selecting MCLK_FREQ, make sure that you do not exceed the frequency specification of your configuration memory, or of your PCB. Review the MachXO3D device AC specifications in the MachXO3D Family Data Sheet (FPGA-DS-02026) when making MCLK_FREQ decisions.

SN

The SN pin is the slave SPI port chip select. An external SPI bus master asserts the SN pin is active low to perform actions using the MachXO3D device programming and configuration logic. The SN pin is available when the MachXO3D device 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 MachXO3D device depends upon maintaining the SN pin in the correct state:

- SN must be deasserted (that is, held high) when configuring using Master SPI mode
- SN must be de-asserted when configuring using the SDM mode
- SN must be deasserted when the MachXO3D device is in user mode
- SN must be deasserted when accessing the configuration logic in the MachXO3D device 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 MachXO3D device 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 ports 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 MachXO3D device is programmed to configure in either external or dual boot mode, the CSSPIN pin is asserted to the attached SPI Flash. The MachXO3D device asserts the CSSPIN until all configuration data bytes are loaded, at which time the CSSPIN enters a high impedance state.

When the MachXO3D device is in the Feature Row HW Default Mode state, the CSSPIN is a general-purpose I/O with a weak pull-up. 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 an external SPI Flash, ensure that the SPI Flash VCC and the MachXO3D 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 with the PROM's VCC rail. The CSSPIN pin, by default, has a weak pull-up resistor internally. Adding a 10 kΩ pull-up resistor to the CSSPIN pin on the MachXO3D device is recommended.

SI/SISPI

The SI/SISPI is a dual-function bidirectional pin. The direction depends on whether master or slave mode is active. The SI/SISPI is an input data pin when using the slave SPI mode and an output data pin when using the master SPI mode. In Master SPI mode, the MachXO3D device drives SI/SISPI until all configuration data bytes are 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 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 on whether master or slave mode is active. The SO/SPISO is an input data pin when using the Master SPI mode and 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 to preserve this pin as SO/SPISO and allow access to the SPI interface.

4.14.3. I^2C Configuration Port Pins

SCL

The MachXO3D device provides an I^2C configuration port. The SCL is the I^2C Serial Clock pin and is used to initiate and time transactions on the I^2C bus. It is a bidirectional, open-drain signal that is an output when the MachXO3D I^2C controller is mastering transactions on the bus and an input when an external I^2C master is accessing resources inside the MachXO3D device. SCL requires an external pull-up resistor to operate.

The SCL pin is available when the MachXO3D device is in the Feature Row HW Default Mode state. You must enable the I2C_PORT and instantiate the EFB for the I^2C 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 set the I2C_PORT to ENABLE.

SDA

The SDA pin is the I^2C serial data input/output pin. It is bidirectional, open-drain, and requires an external pull-up resistor to operate. The pin changes direction dynamically during data transactions on the I^2C 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 MachXO3D device is in the Feature Row HW Default Mode state. You must enable the I2C_PORT and instantiate the EFB if you want the I^2C 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 set the I2C_PORT to ENABLE.

4.14.4. 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 MachXO3D device that can perform configuration, programming, and multi-device configuration functions. Programming and configuration over the JTAG port use IEEE 1532-compliant commands. In addition to the IEEE 1532 capabilities, the MachXO3D device 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 MachXO3D device 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, that is, in the ENABLE state. The JTAG pins can be recovered to be general-purpose I/O by setting the JTAG_PORT preference to the disbale state. It is recommended that the JTAG port remain dedicated to programming pins.

The JTAG port, when set to 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. De-asserting 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 MachXO3D Family Data Sheet (FPGA-DS-02026). 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 MachXO3D device.

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 MachXO3D device.

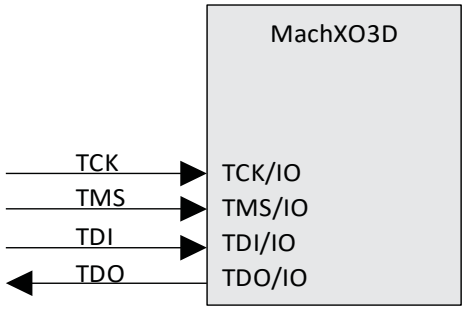


Figure 4.8. Default JTAG Port with JTAG_PORT = ENABLE

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.

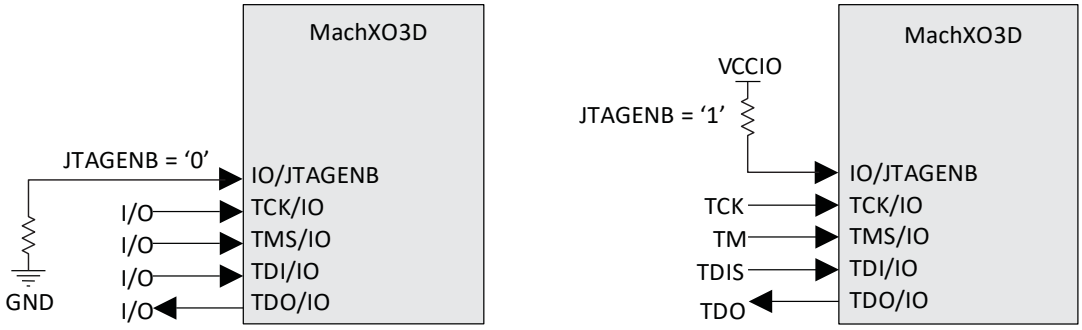


Figure 4.9. JTAG Port Behavior with JTAG_PORT = DISABLE

It is critical that the external logic attached to the JTAG I/O pins does not contend with a JTAG programming system when using the JTAGENB feature. 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 MachXO3D device provides multiple options for loading the configuration SRAM from a non-volatile memory. The previous section describes the physical interface necessary to interact with the MachXO3D device 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 (SDM) include:

- Speed: The MachXO3D device 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 MachXO3D device.
- Reduced board space: Elimination of an external PROM allows your board to be smaller.

The MachXO3D device retrieves the configuration data from the internal Flash, CFG0 or CFG1, when it is using Self-Download Mode. To set the MachXO3D device operation using the SDM Configuration Mode, you must:

- Store the entire configuration data in CFG0 or CFG1.
- Set the preference as shown in Table 5.1.

Note: It is recommended that the SLAVE_SPI_PORT option be disabled if slave SPI mode is not required when configuring in SDM. This prevents interference from the slave SPI port SN pin.

Table 5.1. SDM Configuration Software Settings

Preference	Setting
CONFIGURATION	CFG | CFG_EBRUFM | CFGUFM
PRIMARY_BOOT	IMAGE_0 | IMAGE_1
SECONDARY_BOOT	NONE

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. MSPI configuration Mode must be used in the event of the Memory overflow.

5.2. Master SPI (MSPI) Configuration Mode

Master SPI (MSPI) Configuration Mode is the only other self-controlled configuration mode available to the MachXO3D device. When the MachXO3D device has the MSPI Configuration Mode 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 MachXO3D device is in the Feature Row HW Default Mode state. Lattice recommends having a secondary configuration port available, one that is active when the MachXO3D device is in Feature Row HW Default Mode state, which allows you to recover the MachXO3D device in the event of a programming error.

To ensure that the MachXO3D device operates correctly using the MSPI Configuration Mode, make sure that:

- The POR of the SPI Flash device is lower than the POR of the MachXO3D device, or the SPI Flash must be powered first.
- SPI Flash F_{max} is greater than the MachXO3D MCLK F_{max}.
- Board routing requirements to ensure the MachXO3D device setup and hold time parameters are met.

Refer to the MachXO3D Family Data Sheet (FPGA-DS-02026) for detailed setup and hold time information.

If the SPI Flash POR is higher than the MachXO3D POR and has a slow ramp, what happens by default is listed below:

1. MachXO3D device powers up.
2. MachXO3D device begins toggling MCLK.
3. The preamble from the SPI Flash does not return because its POR level is not met.
4. MachXO3D device times out because it fails to get the preamble in time and the boot up likewise fails.

   In this case, you can increase preamble timer up to about 126 ms to avoid this issue. Further, if the maximum timer is not enough to delay the preamble detection, we can enable the preamble to retry count up to 3. Usually with these methods, it can read the preamble after the SPI Flash device is ready. If the preamble retry does not work in the worst case, following are some workaround solutions:

   - Processor to hold INITN
   - Processor to hold PROGRAMN
   - RC delay to INITN as shown in Figure 5.1.

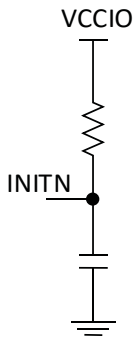


Figure 5.1. RC Delay

Table 5.2. Master SPI Port

Pin Name	Function
MCLK[1]	Clock output from the MachXO3D device configuration logic and SPI master controller. Connect MCLK to the SCLK input of the Slave SPI device.
CSSPIN[2]	Chip select output from the MachXO3D device configuration logic to the slave SPI Flash holding configuration data for the MachXO3D device.
SISPI/D0	Serial Data output from the MachXO3D device to the slave SPI SI input. Or data input in DUAL and QUAD read mode.
SPISO/D1	Serial Data input to the MachXO3D device configuration logic from the slave SPI SO output. Or data input in DUAL and QUAD read mode.
D2	Data input from SPI Flash in QUAD read mode.
D3	Data input from SPI Flash in QUAD read mode.

Notes:

1. Use 1 kΩ pull-down resistor.
2. Use 4.7 kΩ pull-up resistor.

Table 4.2 provides information about the amount of memory needed for MachXO3D device configuration data by device density. The MachXO3D device supports both the standard read with the 03 hex Read Opcode and the fast read with the 0B hex Read Opcode.

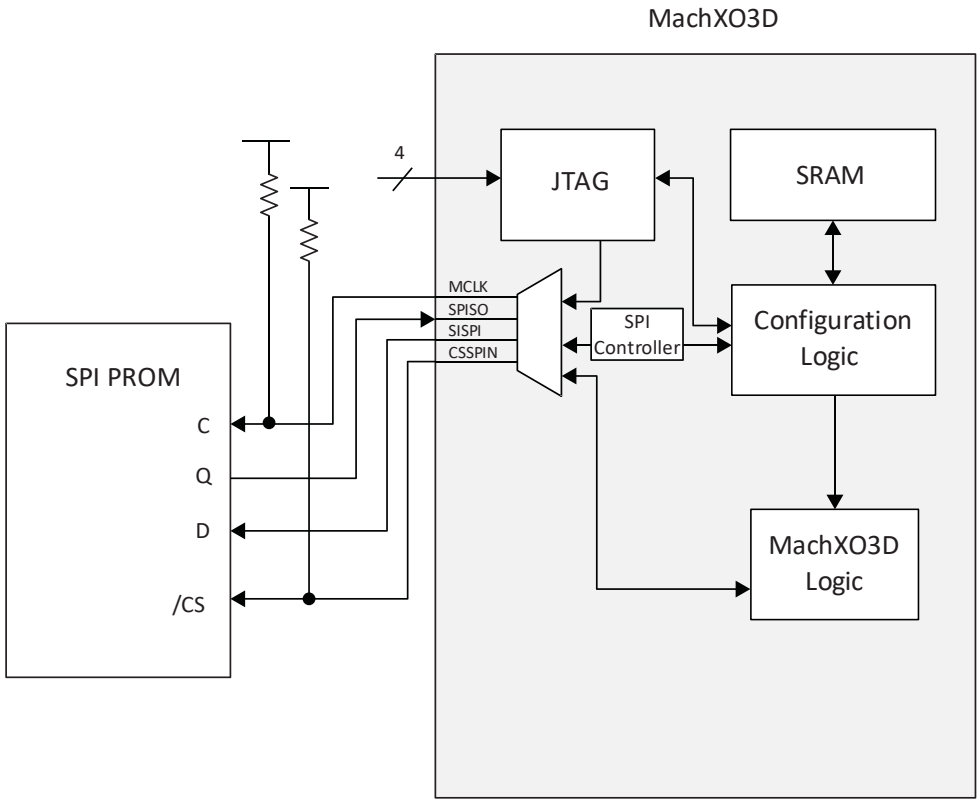


Figure 5.2. Master SPI Configuration Mode

The MachXO3D device 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/D0 output, and data is read from the PROM on the SO/SPISO/D1 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 MachXO3D device reads the remaining configuration data bytes using the new MCLK frequency.

After the MachXO3D device enters user mode the MSPI 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 MachXO3D device. 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 MSPI configuration port transactions is to master them from the JTAG port. The MachXO3D device 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 1 kΩ pull-down 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 MachXO3D device, 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.

The MSPI Configuration Mode in the MachXO3D devices are expanded to support new industry standard Dual/Quad I/O SPI Flash memory. The support of Serial Multi I/O Flash memory enables fast parallel read.

A typical SPI Flash interface uses either four or six interface signals to the FPGA. The standard SPI Flash uses CLK, CS, SI and SO while Quad SPI Flash uses CLK, CS, I/O0, I/O1, I/O2, and I/O3, maintaining function and pin-out compatibility with the single SPI Flash devices, while adding Dual-I/O and Quad-I/O SPI capabilities.

In Dual mode, the Fast-Read Dual Output (BBh) instruction is issued and is similar to the standard Fast Read (0Bh) instruction except that data is output on two pins, SO and SIO0, instead of just SO. This allows data to be transferred from the dual output at twice the rate of standard SPI devices. In QUAD mode, the Fast-Read Quad Output (EBh) instruction is issued and is similar to the standard Fast Read (0Bh) instruction except that data is output on four data pins, instead of just SO. This allows data to be transferred from the quad output at four times the rate of standard SPI devices.

To change the SPI read mode to fast read, dual read or quad read, the Deployment Tool must be used to generate the hex file used for programming the SPI Flash device. Diamond flow only generate bitstream with default SPI read mode which is slow serial (03h) read mode.

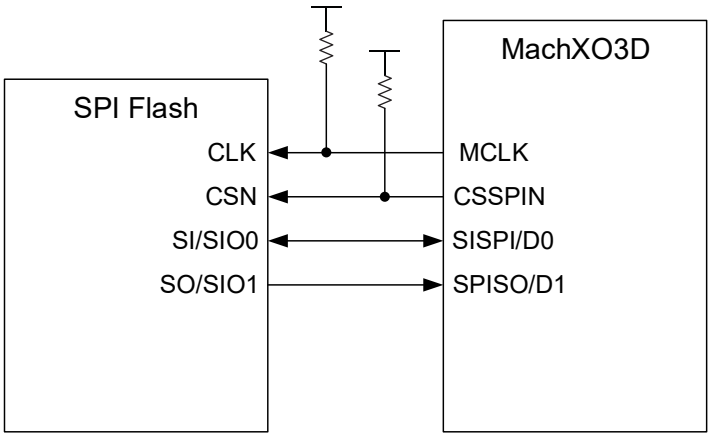


Figure 5.3. One Dual SPI Flash Interface

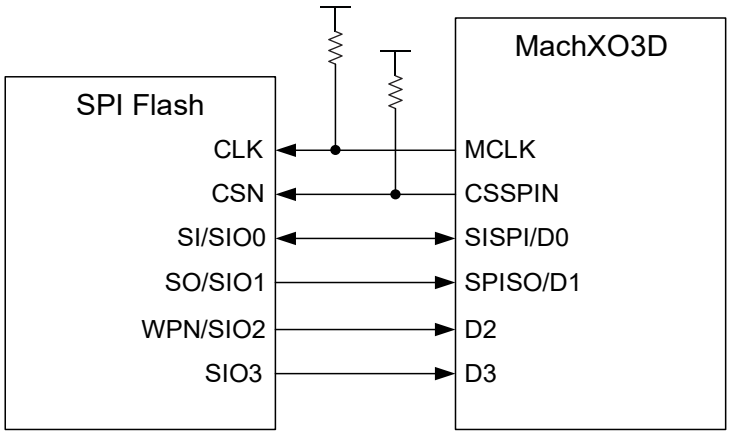


Figure 5.4. One Quad SPI Flash Interface

To set the MachXO3D device for operation using the MSPI Configuration Mode you must:

- Store the entire configuration data in an external SPI Flash.
- Make sure that the data starts at offset 0x000000 within the PROM.
- Set the preference as shown in Table 5.3.

Table 5.3. MSPI Configuration Software Settings

Preference	Setting
CONFIGURATION	EXTERNAL
PRIMARY_BOOT	EXTERNAL
SECONDARY_BOOT	NONE

The BIT file must be programmed into the external SPI Flash starting at address 0x000000. There are several ways to get the data into the SPI Flash:

- Diamond Programmer transmits the SPI Flash data using a JTAG download cable
- A microprocessor running ispVME
- Automatic Test Equipment programs the SPI Flash using JTAG
- Pre-programmed SPI Flash memories is 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 MachXO3D device configures from the external SPI Flash.

5.3. Dual Boot Configuration Mode

The MachXO3D device, when set up in Dual Boot Configuration Mode, has two types of configurations: golden image dual configuration and version-based dual configuration.

5.3.1. Golden Image Dual Configuration

In Golden Image Dual Configuration, the MachXO3D device tries to configure first from the primary image stored in an internal Flash memory sector or the external SPI Flash memory. If the first configuration fails, the MachXO3D device attempts to configure itself from the golden image stored in another internal Flash memory sector or external SPI Flash memory. The dual boot sequence may be changed if you want to use the CONFIGURATION/PRIMARY_BOOT/SECONDARY_BOOT options in the Diamond software spreadsheet view. The MachXO3D device supports seven golden image dual boot sequences. The primary image and/or the golden image can be flexibly stored in the internal Flash memory and/or the external SPI Flash memory. These dual boot sequences can be divided into three categories. The first is the dual boot only from the internal Flash memory, such as CFG0_CFG1 and CFG1_CFG0. The second is the dual boot only from the external SPI Flash memory, like EXTERNAL_EXTERNAL. In this case, the primary SPI Flash start address is 0x000000, the secondary or golden SPI Flash start address can be next to the primary or anywhere on the remainder of the memory, and the JUMP command is located at the last sector of the SPI Flash (for example, 0x3FF0000 for 512 Mb size and 0x007F000 for 64 Mb size). The third is the dual boot from both the internal Flash memory and the external SPI Flash memory, such as CFG0_EXTERNAL, EXTERNAL_CFG0, CFG1_EXTERNAL, and EXTERNAL_CFG1.

Dual Boot Configuration Mode can be utilized in conjunction with the MachXO3D device Soft Error Detection (SED) feature without restriction. Refer to the MachXO3D Soft Error Detection (SED)/Correction (SEC) Usage Guide (FPGA-TN-02124) 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. If both fail, the device stays in the un-programmed state. 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. The 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 functionally correct. In this instance, the internal DONE bit never becomes active. The MachXO3D device counts the number of master clock pulses provided

after the Power On Reset signal is 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 golden or fail-safe image that is rarely, if ever, updated. The second configuration data file is a primary or working image that is routinely updated. 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 shown in Table 5.3 to build a dual-boot design.

Table 5.3. Dual Boot Configuration Software Settings

Preference	Setting
CONFIGURATION	CFG|CFG_EBRUFM|CFGUFM|EXTERNAL
PRIMARY_BOOT	IMAGE_0|IMAGE_1|EXTERNAL|LATEST|FORMER
SECONDARY_BOOT	NONE|IMAGE_0|IMAGE_1|EXTERNAL|LATEST|FORMER

The legal combinations of CONFIGURATION, PRIMARY_BOOT and SECONDARY_BOOT settings are listed in Table 5.4.

Table 5.4. Legal combination of CONFIGURATION, PRIMARY_BOOT and SECONDARY_BOOT Settings

CONFIGURATION	PRIMARY_BOOT	SECONDARY_BOOT	Boot	Boot Sequence
CFG|CFG_EBRUFM|CFGUFM	IMAGE_0	NONE	Single	CFG0 (SW default)
		IMAGE_1	Dual	CFG0_CFG1
		EXTERNAL	Dual	CFG0_EXT
	IMAGE_1	NONE	Single	CFG1
		IMAGE_0	Dual	CFG1_CFG0
		EXTERNAL	Dual	CFG1_EXT
	LATEST	FORMER	Dual	Latter_Former
	FORMER	LATEST	Dual	Former_Latter
EXTERNAL	EXTERNAL	NONE	Single	EXT
		IMAGE_0	Dual	EXT_CFG0
		IMAGE_1	Dual	EXT_CFG1
		EXTERNAL	Dual	EXT_EXT

In the Diamond flow, the JEDEC file option should be selected when generating configuration data that is stored in the internal Flash. The bitstream file option should be selected when generating configuration data that is stored in the external SPI flash. If both dual boot images are stored in the external SPI Flash, the primary SPI Flash start address is 0x000000, the golden SPI Flash start address can be next to the primary or anywhere on the remainder of the memory, and the JUMP command is located at the last sector of the SPI Flash (for example, 0x3FF0000 for 512 Mb size and 0x007F000 for 64 Mb size). But for the external SPI Flash to store only one of the dual boot images, the image, regardless of the primary or golden configuration image, must be located in the external SPI Flash, starting at address 0x000000.

5.3.2. Version Based Dual Configuration

For each of the configuration images stored in the internal CFG0 and CFG1, there is one 4-bit version tag for it. After programming an image into one of CFG0 and CFG1, a new version number is generated by increasing the version number for the other of CFG0 and CFG1 by one. And it is automatically programmed and used for the new image. Version tag *0000* is assumed to be larger than *1111*. It means the later programmed bitstream has a larger version number. In this type of configuration, the dual boot configurations are from CFG0 and CFG1. And the boot sequence depends on the version number. The first configuration can be from the latter image of the two images stored in CFG0 and CFG1 with PRIMARY_BOOT = LATEST and SECONDARY_BOOT = FORMER, or from the former bitstream with

PRIMARY_BOOT = FORMER and SECONDARY_BOOT = LATEST. If this fails, configure the other image. If both fail, the device stays in its unprogrammed state.

5.3.3. Status Register Readout during Boot Failure

Boot1Fail can be used to check if the device boots in the primary or secondary image. Boot1Fail is set to 1 when the following conditions are met:

- Internal DONE bit of primary image=0
- The primary image is fully erased, include the internal DONE bit of the target sector.
- Preamble is intact but Invalid command or invalid bitstream data is accidentally programmed with DONE bit=1

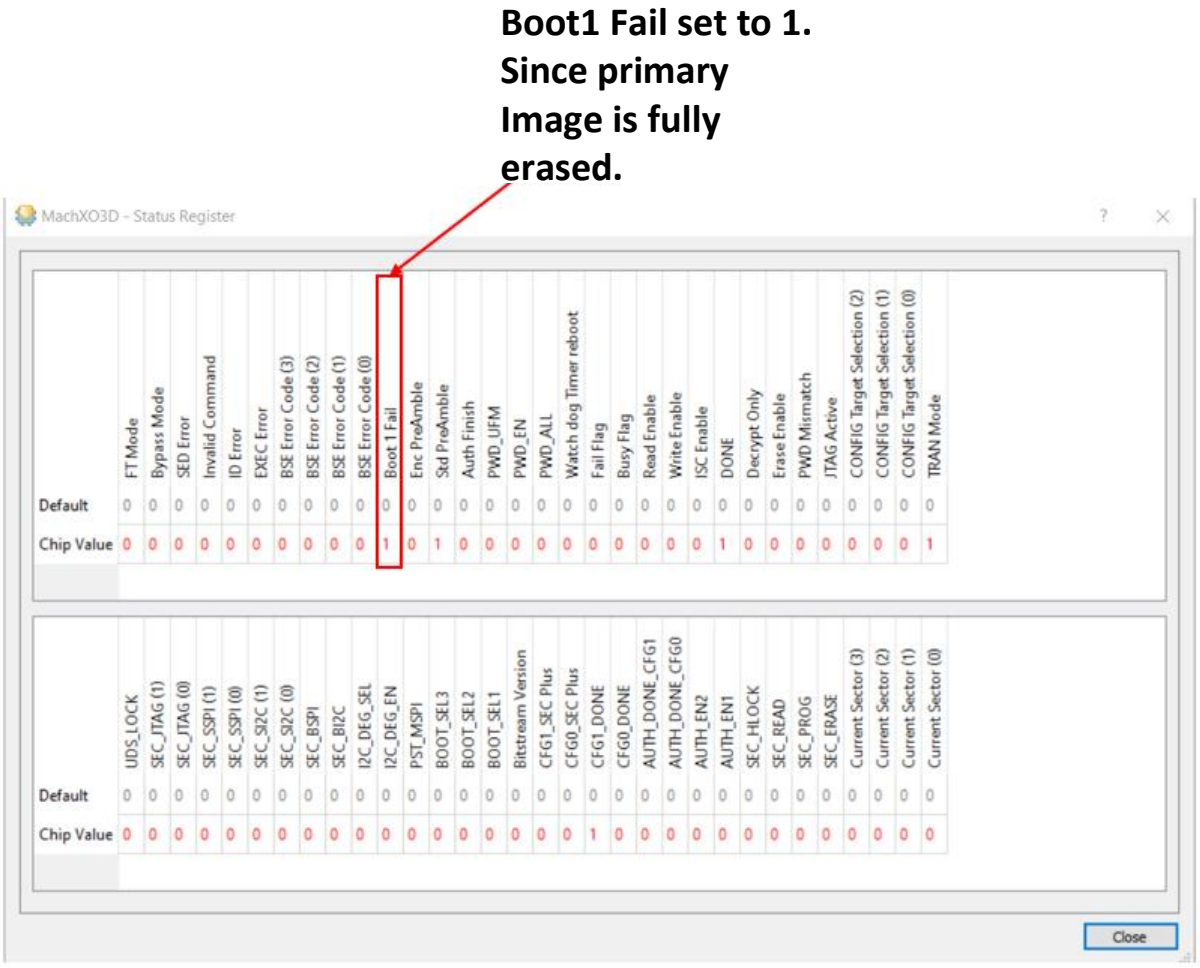


Figure 5.5. MachXO3D – Status Register

However, on conditions when DONE bit=1 and preamble is not found in the primary image, the Boot1Fail will not trigger but the device will still boot on the golden image.

Given these conditions, it is best to check SRAM Usercode to determine which image the device boots. It is advisable that primary and golden image have unique usercode to distinguish which image the device boots.

5.4. Slave SPI Mode (SSPI)

The MachXO3D device provides a slave SPI configuration port that allows you to access features provided by the configuration logic. You can reprogram the SRAM, Flash, and Feature Row and access status/control registers within the configuration logic block. Reprogramming the Flash can be done using offline or transparent operations.

Table 5.5. 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 MachXO3D device configuration logic for command and data.
SO	Serial Data Output from the MachXO3D device configuration logic.
SN	Chip select to enable the MachXO3D device configuration logic.

In the slave SPI mode, the MCLK/CCLK pin becomes the configuration clock (CCLK). Input data is read into the MachXO3D 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 selects 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 MachXO3D device when SN is asserted. The MachXO3D device SSPI port only accepts Mode 0 bus transactions in the configuration logic.

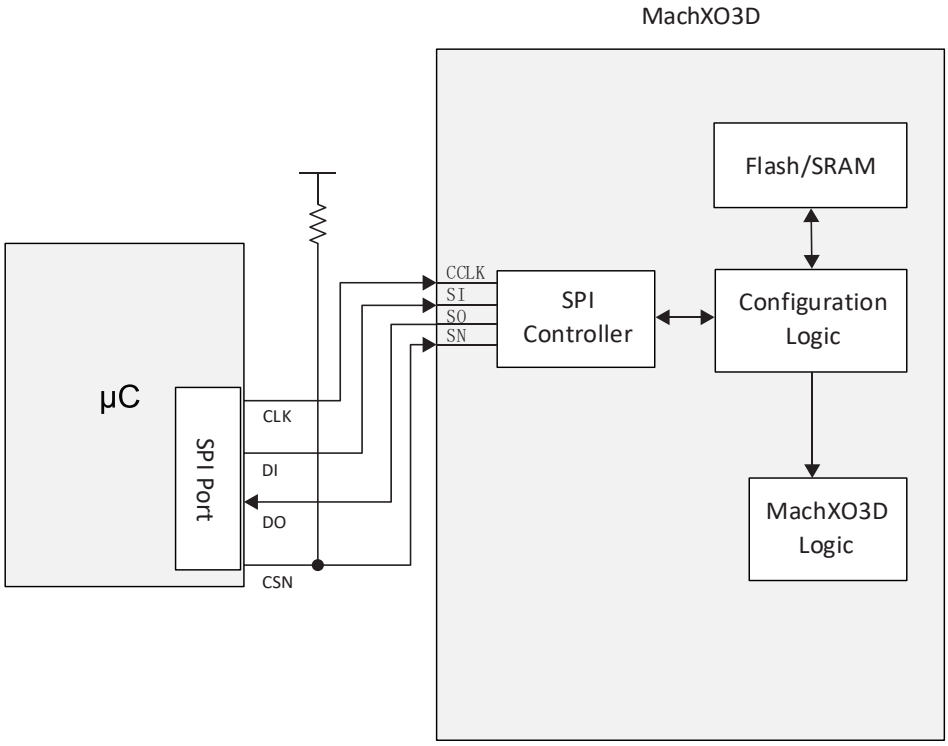


Figure 5.6. Slave SPI Configuration Mode

The SSPI port is active when the MachXO3D device is in the Feature Row HW Default Mode state. Lattice 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 not capable of directly accessing the configuration SRAM. To prevent unintentional erasure of the Feature Row, it is recommended that 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

MachXO3D device is using External or Dual Boot Configuration Mode because the REFRESH operation does not begin until SN is deasserted.

Programming the MachXO3D device using the SSPI port is complex. As such, Lattice provides C source code called SSPIEMbedded to insulate you from the complexity of programming the MachXO3D device. Use SSPIEmbedded to reprogram the flash 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 MachXO3D device has an I²C configuration port for use in accessing the configuration logic. An I²C master can communicate with 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, Flash, and Feature Row and access status/control registers within the configuration logic block. Reprogramming the Flash can be done offline or in transparent operations.

Note: When programming the device flash through the I2C port, ensure that no other I2C target device on the same bus interrupts the transaction. Interruptions can cause programming failure.

Table 5.6. I²C Port Pins

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

The I²C configuration port is available when the MachXO3D device is in the 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, to recover from erroneously disabling the I²C port.

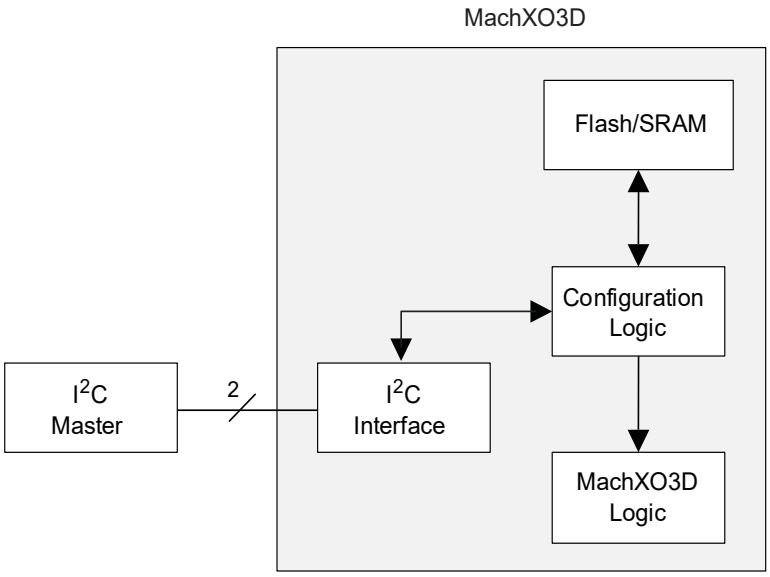


Figure 5.7. I²C Configuration Logic

There are two hardened I²C controllers in a MachXO3D device, a primary and a secondary. The primary controller provides an interface to the MachXO3D 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 MachXO3D device is in the Feature Row HW Default Mode state, the I²C port is enabled. 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 7.5 times the I²C bus rate, for example, >3.0 MHz when the I²C rate = 400 kHz.

An external I²C master accesses the configuration logic using address 1000000 in 7-bit mode or 1111000000 in 10-bit mode, unless the EFB I²C base address is 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 to change the address. The address is shared by the primary and secondary I²C controllers.

Table 5.7 shows the address decoding used to access the I²C resources in the MachXO3D device.

Table 5.7. 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.

The fourth I²C resource in the MachXO3D device 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 operate correctly.

When the device first boots up, the output of the receiver (Rx) first in first out (FIFO) buffer holds random data. A blind read operation returns this random data. If the random data resembles a valid command such as the refresh command (0x79), the device interprets this as an instruction and attempts to execute it. To prevent this, ensure that you send a valid read command before performing a read operation. Alternatively, you can clear the Rx FIFO by reading the device ID on the WISHBONE bus before sending any read command through the I²C bus. This sets the Rx FIFO output data to a benign value.

5.6. WISHBONE Configuration Mode

The MachXO3D device can access the Configuration Flash, User Flash Memory, and Feature Row from an internal WISHBONE bus. To use the WISHBONE bus, the Embedded Function Block (EFB) must be inserted into your design. You can design the logic to interface to the EFB, then perform WISHBONE bus transactions to access resources attached to the configuration logic.

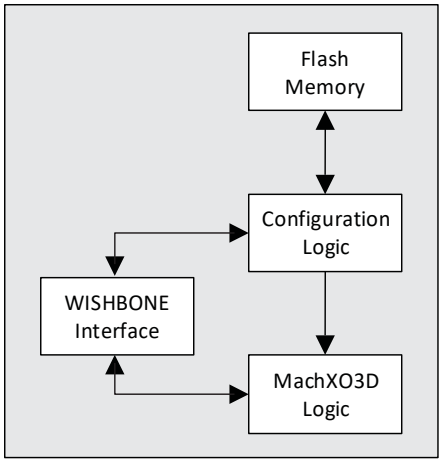


Figure 5.8. WISHBONE Configuration Mode

In order to access the WISHBONE interface, the MachXO3D device must be in user mode. Accessing and updating the resources made available by the configuration logic must be completed in transparent mode. Attempting access to the configuration logic in offline mode causes a deadlock because the MachXO3D device leaves user mode.

You can get more detailed information about the MachXO3D WISHBONE interface from Using Hardened Control Functions in MachXO3D (FPGA-TN-02117).

5.7. JTAG Mode

The JTAG port is the most flexible configuration and programming port available on the MachXO3D device. The JTAG provides:

- Offline Flash programming
- Transparent Flash memory programming
- Offline SRAM configuration
- Full access to the MachXO3D configuration logic
- Device chaining
- IEEE 1149.1 testability
- IEEE 1532 compliant programming

The JTAG port is available when the MachXO3D device is in Feature Row HW Default Mode state. The MachXO3D 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 MachXO3D JTAG port is a valuable asset due to its flexibility. It provides the best capabilities for system and device debugging. Lattice recommends the JTAG port remain accessible in every MachXO3D 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 MachXO3D device to be combined in a chain of other programmable logic.

- Reveal Debug

  The Lattice Reveal debug tool is an embeddable logic analyzer tool. It allows you to analyze the logic inside the MachXO3D device in the same fashion as an external logic analyzer permits analysis of board level logic. Reveal access is only available via the MachXO3D JTAG port.

- SRAM Readback

  The JTAG port is the only sysCONFIG port able to directly access the MachXO3D device configuration SRAM.

- Boundary Scan Testability

  Board level connectivity testing performed using IEEE 1149.1 JTAG is a key capability for ensuring the quality of assembled printed circuit boards. Preserving the MachXO3D JTAG port is vital for boundary scan testability. Lattice provides Boundary Scan Description Language files for the MachXO3D device on the Lattice website.

5.7.1. JTAG Daisy Chain

A JTAG daisy chain is a configuration that allows multiple devices to be connected in series and accessed through a single JTAG interface. Key features of the JTAG daisy chain include:

- Single JTAG interface – The host programmer connects to the first device in the chain and communicates with all devices sequentially.
- Signal flow
  - TDI, TMS, and TCK signals are provided by the host.
  - TDO from each device is connected to the TDI of the next device in the chain.
- Power supply – All devices are powered by a common voltage source, typically 1.8 V, 2.5 V, or 3.3 V.
- Pull-up/down resistors – Used on clock, control, and data lines such as TDI, TMS, TDO, and TCK to ensure stable logic levels and prevent floating inputs or outputs.

The advantages of the JTAG daisy chain include:

- Efficiency – Enables programming or testing of multiple devices without needing separate JTAG interfaces.
- Scalability – Easily extendable by adding more devices to the chain.
- Simplified design – Reduces the number of required pins and simplifies PCB layout.

Figure 5.9 shows a typical implementation of a JTAG daisy chain with three devices. Data passes from one device to the next through the TDO and TDI pins.

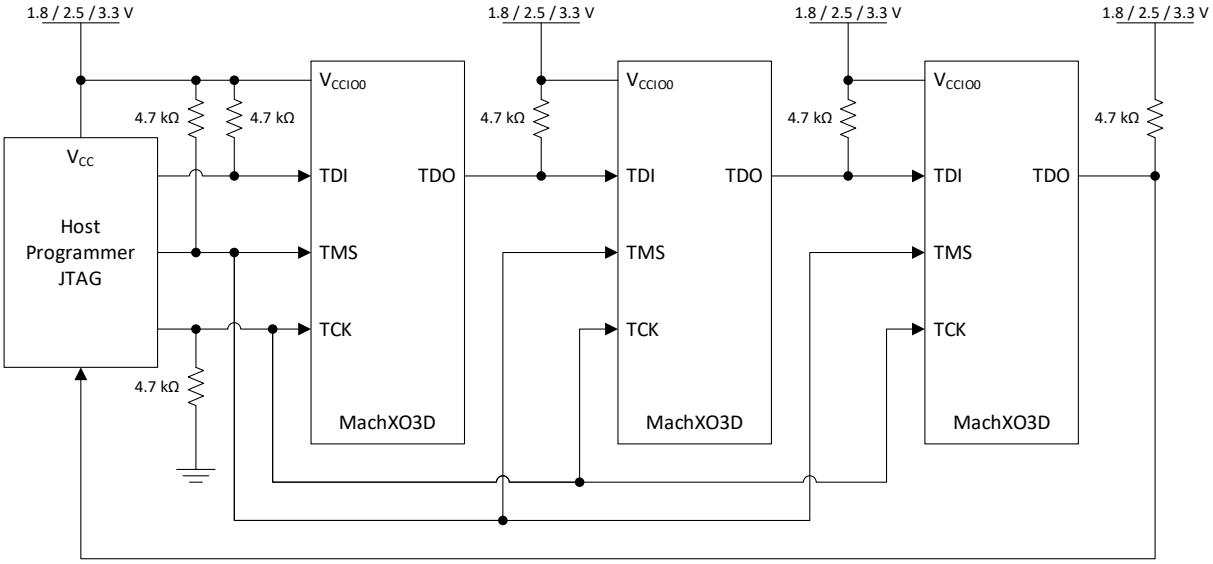


Figure 5.9. JTAG Daisy Chain Example

5.8. TransFR Operation

The MachXO3D device, 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 the bitstream on a MachXO3D device by using the TransFR feature.

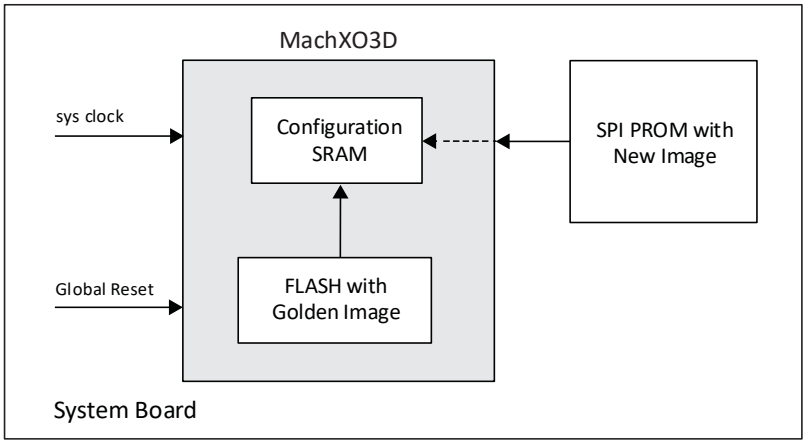
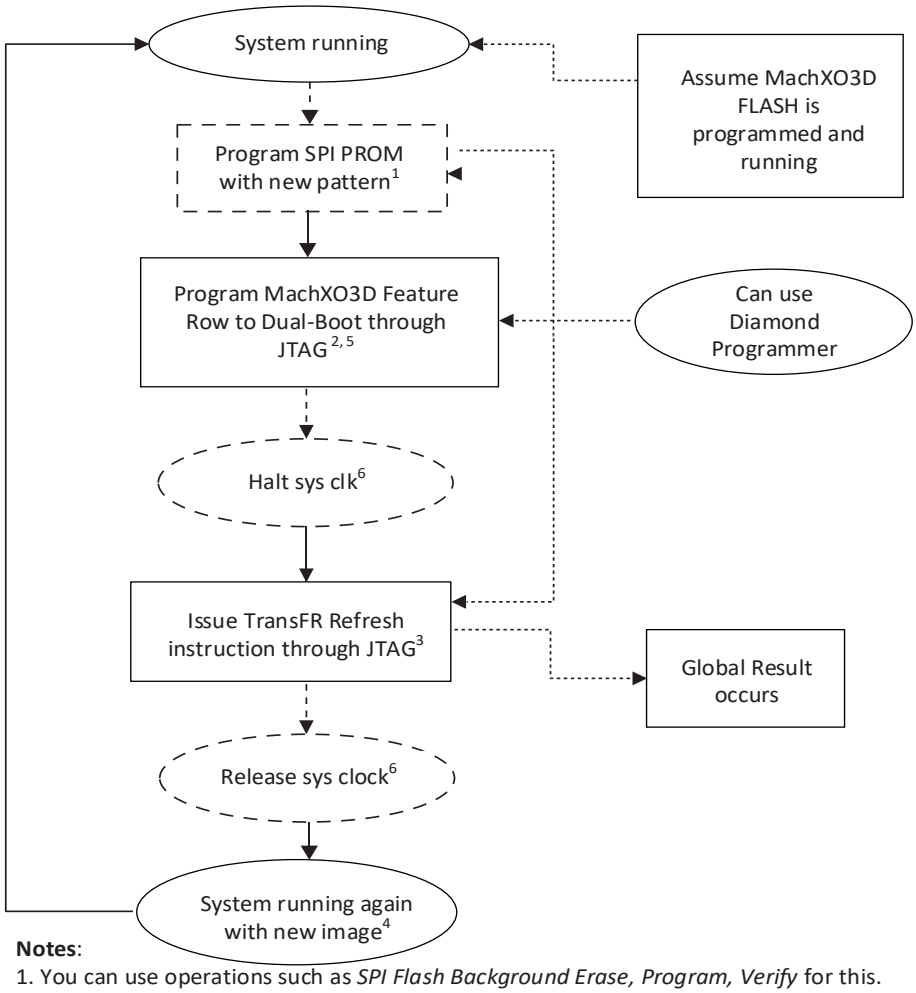


Figure 5.10. Bitstream Update Using TransFR

The example assumes that you have the golden image stored in Flash to initiate the system, and then use SPI PROM as a resource for image updates without disturbing the system. Figure 5.11 shows the process flow for performing this task.



Notes:
1. You can use operations such as *SPI Flash Background Erase, Program, Verify* for this.
2. You can use operations such as *Program Feature Row* for this.
3. You can use operations like *XFLASH TransFR* for this.
4. If new image fails to configure MachXO3D device, the golden image in FLASH still configures MachXO3D device, so system still runs with original image. If the bitstream needs to be authenticated and decrypted, the BACKGROUND_RECONFIG_SECURITY option must be set to ON.
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, skip this step.
6. This step is optional.

Figure 5.11. Example Process Flow

Caution when using the above process flow:

Since a global reset is triggered during device wake-up after a REFRESH instruction is issued, attention needs to be given to designing I/O with the 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.

5.9. Password

The MachXO3D device 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 MachXO3D 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 128-bit Flash Protect Key is stored in the feature row. Three additional feature row fuses are specified for enabling the feature: PWD_EN, PWD_ALL, and PWD_UFM.

For more information on password security, refer to the MachXO3 Using Password Security technical note (FPGA-TN-02072).
Note: Please note that section 4.3 in FPGA-TN-02072 is not applicable.

6. Software Selectable Options

The operation of the MachXO3D device configuration logic is managed by the options selected in the Diamond design software. Other FPGAs provide dedicated I/O pins to select the configuration mode. The MachXO3D device uses the non-volatile Feature Row to select how it configures. The Feature Row default state needs to be modified in almost every design. You use the Diamond Spreadsheet View to make changes to the operation of the MachXO3D 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 6.1. The sysCONFIG preferences are divided into three categories:

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

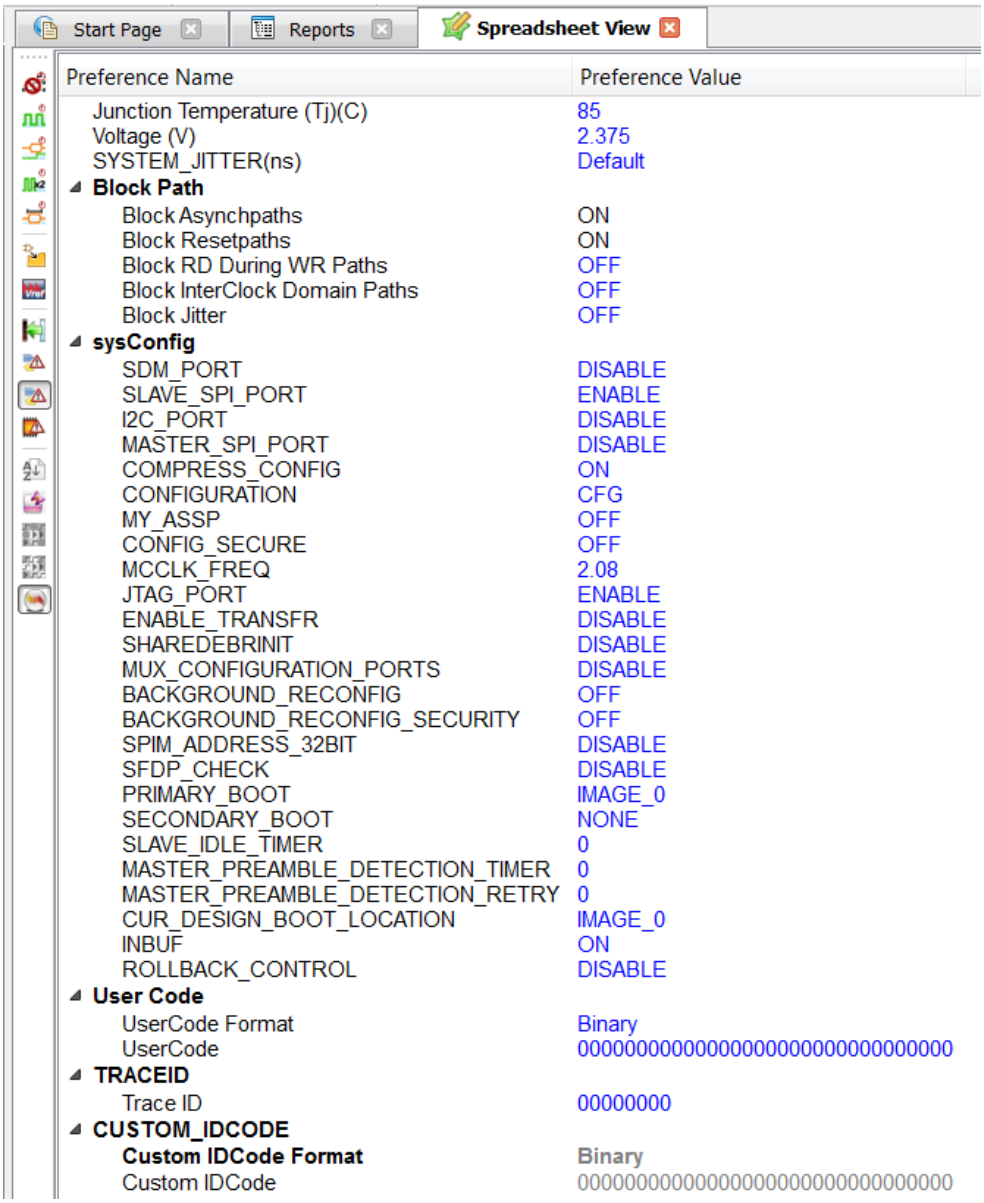


Figure 6.1. sysCONFIG Preferences in Global Preferences Tab, Diamond Spreadsheet View

6.1. Configuration Mode and Port Options

The configuration and port options allow you to decide which configuration ports continue to operate after the MachXO3D 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 6.1. Configuration Mode/Port Options

Option Name	Default Setting	All Settings
JTAG_PORT	ENABLE	ENABLE, DISABLE
SLAVE_SPI_PORT	DISABLE	ENABLE, DISABLE
MASTER_SPI_PORT[1]	DISABLE	ENABLE, DISABLE
I2C_PORT	DISABLE	ENABLE, DISABLE
I2C_GLITCH_FILTER[2]	DISABLE	ENABLE, DISABLE
I2C_GLITCH_FILTER_RANGE[2]	R_16_50NS[3]	R_8_25NS[3], R_16_50NS[3]
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

Notes:

1. MASTER_SPI_PORT setting is automatically disabled or enabled by Diamond based on the CONFIGURATION, PRIMARY_BOOT, and/or SECONDARY_BOOT settings.
2. I2C_GLITCH_FILTER and I2C_GLITCH_FILTER_RANGE are automatically disabled when I2C_PORT is disabled.
3. R_8_25NS means the filter performance can range from 8 to 25 ns. R_16_50_ns means the filter performance can range from 16 to 50 ns. Filter performance varies within these ranges. Use of an external filter is recommended if high accuracy is required.

6.1.1. JTAG Port

The JTAG_PORT preference allows you to decide how the JTAG configuration port pins operate when the MachXO3D device is in user mode. There are two states in which the JTAG_PORT preference can be set:

- ENABLE – In this mode the JTAG I/O 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 DISABLED state. JTAGENB, when asserted high, makes the four JTAG I/Os act as an IEEE 1149.1 JTAG port. JTAGENB driven low causes the four I/Os to be available for use as general-purpose I/Os.

Lattice recommends that the JTAG port must be accessible when reprogramming the MachXO3D device to disable your primary configuration port.

6.1.2. Slave SPI Port

The SLAVE_SPI_PORT allows you to decide how the Slave SPI sysCONFIG pins operate when the MachXO3D device is in 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 MachXO3D 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 does 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.

6.1.3. Master SPI Port

The MASTER_SPI_PORT allows you to preserve the SPI configuration port after the MachXO3D device enters user mode. For the MachXO3D device, this option is set depending on the CONFIGURATION, PRIMARY_BOOT, and/or SECONDARY_BOOT settings. There are two states in which the MASTER_SPI_PORT preference can be set:

- ENABLE – This setting preserves the SPI port I/O when the MachXO3D device is in user mode. This preference makes external or dual boot configuration modes with external SPI booting active. After entering user mode, the SPI master controller in the EFB has access to the SPI port for performing SPI bus transactions. Preserving this port allows background programming of the connected external SPI Flash via JTAG using Diamond Programmer. 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 does 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.

6.1.4. I²C Port

The I2C_PORT allows you to preserve the I²C configuration port after the MachXO3D device enters user mode. There are two states in which the I2C_PORT preference can be set:

- ENABLE – This setting preserves the I²C port I/O when the MachXO3D device 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 7.5 times 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.

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

6.1.5. SDM Port

The SDM_PORT allows you to select the programming status pins after the MachXO3D device enters user mode. There are six states in 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.
- PROGRAM – This setting preserves the PROGRAMN pin when the MachXO3D device is in user mode. Asserting this pin as active low causes the MachXO3D device to reconfigure. The DONE and INITN pins are general-purpose I/O.
- DONE – This setting preserves the DONE pin when the MachXO3D device is in user mode. The PROGRAMN and INITN pins are general-purpose I/O.
- INITN – This setting preserves the INITN pin when the MachXO3D device is in user mode. The PROGRAMN and DONE pins are general-purpose I/O.
- PROGRAM_DONE – This setting preserves the PROGRAMN and DONE pins when the MachXO3D device enters user mode. INITN is a general-purpose I/O.
- PROGRAM_DONE_INITN – This setting preserves PROGRAM, DONE, and INITN in user mode.

Lattice recommends setting the SDM_PORT to PROGRAMN when using Master SPI or Dual Boot Configuration Mode. The PROGRAMN pin is the only way to perform a warm reconfiguration of the MachXO3D device, unless another configuration port is available to transmit a REFRESH command.

6.1.6. 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 MachXO3D device uses a nominal 2.08 MHz (+/- 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 MachXO3D 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 MachXO3D device 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). Take care not to exceed the maximum clock rate of your SPI flash or of your printed circuit board.

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

6.1.7. ENABLE_TRANSFR

The TransFR function used by the MachXO3D device requires the configuration data loaded into the configuration SRAM, and any future configuration data file loaded into the internal flash memory has the ENABLE_TRANSFR set to the ENABLE state. See the TransFR Operation section and Minimizing System Interruption During Configuration Using TransFR Technology (TN1087) for more information about using TransFR with the MachXO3D device.

6.1.8. SLAVE_IDLE_TIMER

When downloading the bitstream from the slave port, SSPI, or I^2C, the MachXO3D device configuration module starts running a timer on the configuration clock domain. This timer is enabled to recover from any hang-up caused by the external SPI master or I^2C master. When the timer expires, the configuration engine goes back to the reset state. It allows the configuration engine to wait and receive the new coming bitstream from any slave port. There are 16 options for the setting of this timer. The default option is for the infinite timer, or the disabled timer. The other 15 options define the timer from 10 ms to 128 ms.

Table 6.2. SLAVE_IDLE_TIMER Values

Option	Timeout Period	Option	Timeout Period
0	Infinite (disabled)	8	1000 ms
1	128000 ms	9	640 ms
2	64000 ms	10	320 ms
3	32000 ms	11	160 ms
4	16000 ms	12	80 ms
5	8000 ms	13	40 ms
6	4000 ms	14	20 ms
7	2000 ms	15	10 ms

6.1.9. MASTER_PREAMBLE_DETECTION_TIMER

When configuring from external SPI FLASH in master mode, the configuration module starts running a timer on the MSPI clock. At the beginning of the configuration, the MSPI expects a valid preamble code from the SPI Flash. If this timer expires before detecting the valid preamble, it performs reconfiguration if the MASTER_PREAMBLE_DETECTION_RETRY option is not 3. Otherwise, it boots from the secondary image in external SPI FLASH or internal Flash memory if dual boot mode is enabled. If no boot is successful, the device is in an unprogrammed state. There are 16 options for the setting of this timer. The corresponding timer values are in the range from 3.8 µs to 126 ms. The default is 126 ms.

Table 6.3. MASTER_PREAMBLE_DETECTION_TIMER Values

Option	Preamble Timer Value (µs)	Option	Preamble Timer Value (µs)
0	126156.8	8	492.8
1	63078.4	9	246.4
2	31539.2	10	123.2
3	15769.6	11	61.6
4	7884.8	12	30.8
5	3942.4	13	15.4
6	1971.2	14	7.70
7	985.6	15	3.85

6.1.10. MASTER_PREAMBLE_DETECTION_RETRY

Once the timer of MASTER_PREAMBLE_DETECTION_TIMER expires, the configuration module can retry the preamble detection flow by restarting the configuration. If all the retry attempts fail, except for Option 2 (Infinite), it boots from the secondary image in the external SPI flash or from the internal Flash memory if dual boot mode is enabled. If no boot is successful, the device stays in the unprogrammed state. There are four options for setting the number of retry attempts, as defined in Table 6.4. The default value is 0, which is Retry once.

Table 6.4. MASTER_PREAMBLE_DETECTION_RETRY Values

Option	Retry Time
0	Retry once
1	Retry twice
2	Infinite
3	No retry

6.1.11. SPIM_ADDRESS_32BIT

The SPIM_ADDRESS_32BIT preference can be used to support 32-bit addressing, typically found on larger SPI Flash devices. When set to OFF by default, the MachXO3D device uses a 24-bit address to access the SPI Flash device via the MSPI configuration port. When set to ON, the address to access the SPI Flash device is extended to 32-bit.

6.1.12. SFDP_CHECK

The SFDP_CHECK preference controls the MachXO3D device to check the SFDP signature in the external SPI Flash in the MSPI Configuration Mode. The SFDP_CHECK preference has three options:

- DISABLE – The default DISABLE option disables the checking of the SFDP signature when booting from the external SPI Flash in the MSPI Configuration Mode.
- ENABLE_SFDP – This preference enables the checking of the SFDP signature when booting from the external SPI Flash in the MSPI Configuration Mode. When the SFDP check fails, booting stops or retries based on the MASTER_PREAMBLE_DETECTION_TIMER preference.
- ENABLE_SFDP_PREAMBLE – This preference enables the checking of the SFDP signature when booting from the external SPI Flash in the MSPI Configuration Mode. When the SFDP check fails, booting continues to detect the preamble in the external SPI Flash.

6.2. Bitstream Generation Options

The Bitstream Generation options allow you to decide how the Diamond development tools create the configuration data for the MachXO3D 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.

6.2.1. COMPRESS_CONFIG

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

6.2.2. CONFIGURATION

The CONFIGURATION preference allows you to control the configuration memory sectors to store the configuration image. 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 CFG0 or CFG1 Flash sector. The configuration data includes EBR initialization data.
- CFG_EBRUFM – This preference creates configuration data that is stored in the CFG0 or CFG1 Flash sectors. EBR initialization data is stored in the lowest page addresses of the UFM0 or UFM1 sectors. The UFM0 or UFM1 sector is available in user mode. You must restore the EBR initialization data when making changes to the UFM0 or UFM1 to guarantee correct operation.
- CFGUFM – This preference creates configuration data that is stored in the CFG0 or CFG1 Flash sectors. This mode differs from CFG by allowing the configuration data to overflow into the UFM0 or UFM1. 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. This setting is used for both the single image MSPI Configuration Mode and the Dual Boot Configuration Mode, with the primary configuration image stored in the external SPI Flash.

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 Flash memory.

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.

6.2.3. PRIMARY_BOOT

The PRIMARY_BOOT preference allows you to control which configuration memory sector is used for the single image configuration mode or the first boot in the dual boot configuration mode. The PRIMARY_BOOT preference has five possible settings:

- IMAGE_0 – This preference is the default setting. It uses CFG0 or CFG0 plus UFM0 sectors for the primary boot. The CONFIGURATION preference determines whether UFM0 is used to store the configuration image.
- IMAGE_1 – This preference uses CFG1 or CFG1 plus UFM1 sectors for the primary boot. The CONFIGURATION preference determines whether UFM1 is used to store the configuration image.
- EXTERNAL – This preference uses the external memory for the primary boot.
- LATEST – This preference is one of the settings for the version-based Dual Boot Configuration Mode. The configuration starts with one of the CFG0 and CFG1 that has the later configuration image or the larger version number first. If the process fails, the former configuration image in the other internal Flash sector is loaded.
- FORMER – This preference is one of the settings for the version-based Dual Boot Configuration Mode. The configuration starts with one of the CFG0 and CFG1 that has the former configuration image or the smaller version number first. If the process fails, the later configuration image in the other internal Flash sector is loaded.

6.2.4. SECONDARY_BOOT

The SECONDRY_BOOT preference allows you to control which configuration memory sector is used for the secondary boot in the Dual Boot Configuration Mode. The SECONDARY_BOOT preference has six possible settings:

- NONE – This preference is the default setting. It is used to select the single image configuration mode.
- IMAGE_0 – This preference is one of the settings for the golden image in Dual Boot Configuration Mode. It uses CFG0 or CFG0 plus UFM0 sectors to store the golden configuration image for the secondary boot. The CONFIGURATION preference determines whether UFM0 is used to store the configuration image.
- IMAGE_1 – This preference is one of the settings for the golden image in Dual Boot Configuration Mode. It uses CFG1 or CFG1 plus UFM1 sectors to store the golden configuration image for the secondary boot. The CONFIGURATION preference determines whether UFM1 is used to store the configuration image.
- EXTERNAL – This preference is one of the settings for the Golden Image Dual Boot Configuration Mode. It uses the external memory to store the golden configuration image for the secondary boot.
- LATEST – This preference is one of the settings for the version-based Dual Boot Configuration Mode. The configuration starts with one of the CFG0 and CFG1 that has the former configuration image or the smaller version number first. If the process fails, the later configuration image in the other internal Flash sector is loaded.
- FORMER – This preference is one of the settings for the version-based Dual Boot Configuration Mode. The configuration starts with one of the CFG0 and CFG1 that has the later configuration image or the larger version number first. If the process fails, the former configuration image in the other internal Flash sector is loaded.

6.2.5. USERCODE

The MachXO3D device configuration flash sector contains a 32-bit register for storing a user-defined value. The USERCODE is also stored in the generated bitstream for redundancy. 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, the 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.

6.2.6. 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.

6.2.7. 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.

6.2.8. 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.

6.2.9. 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 bitstream size of the design and saves internal memory space for other applications.

6.2.10. MUX_CONFIGURATION_PORTS

The MUX_CONFIGURATION_PORTS is used to allow the disabling of all device configuration ports and allow for the configuration ports to be recovered as GPIO. Disabling all the available configuration ports allow the MachXO2 to become a *write one time* device. MUX_CONFIGURATION_PORTS confirms the removal of all configuration ports. The control is only active when all the configuration ports are set to DISABLE. Setting MUX_CONFIGURATION_PORTS to ENABLE activates the JTAGENB input pin, permitting the JTAG port pins to be multiplexed. Setting MUX_CONFIGURATION_PORTS to ENABLE 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 MachXO2 to effectively 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.

6.2.11. CUR_DESIGN_BOOT_LOCATION

The CUR_DESIGN_BOOT_LOCATION preference is used to specify the location of the configuration image generated by the current Diamond project in Dual Boot applications. When set to IMAGE_0, it is targeted to the CFG0 space of the internal Flash. When set to IMAGE_1, it is targeted to the CFG1 space of the internal Flash.

6.2.12. ROLLBACK_CONTROL

When the ROLLBACK_CONTROL preference is set to ENABLE, the MachXO3D device provides a mechanism to prevent a bitstream from updating to an earlier version. The version number is stored in USERCODE and is treated as a 32-bit unsigned integer in this mechanism. In addition, this mechanism is applicable only to the dual configuration mode, except for the boot sequence = EXT_EXT. Authentication is required to be enabled. The flow of the mechanism is different for the various boot sequences.

For the dual boot sequences, using both CFG0 and CFG1, after programming the bitstream to CFG0 or CFG1, you need to send the LSC_PROG_AUTH_DONE command to try to program the pre-authentication Done bit or AUTH_DONE bit. The configuration engine then authenticates this bitstream and checks to see whether its version number is newer than the one in the other internal memory that contains a bitstream. If both authentication and version checking pass, the pre-authentication Done bit is programmed. Otherwise, it is not programmed. You can run Flash Programming Mode > FLASH Version Rollback Protect in Diamond Programmer to program both the bitstream and the pre-authentication Done bit. When booting, the configuration engine checks the pre-authentication Done bit when the preference is enabled. It continues to boot only when the bit is programmed.

For the dual boot sequences with one boot from the external SPI Flash memory and the other from the internal memory (CFG0 or CFG1), only booting from the SPI Flash checks its version embedded in its bitstream against the version in the internal bitstream. The booting continues if the version is newer than the internal one or if the internal bitstream does not exist. When booting from the bitstream in the internal memory, the pre-authentication Done bit is checked. Only the programmed bit allows for continuous booting. Hence, you need to program it by running Advanced Security Keys Programming > Security Program Auth Done FlashA or Security Program Auth Done FlashB in Diamond Programmer after programming its bitstream.

Rollback and user information are found in the USERCODE space when ROLLBACK_CONTROL is enabled. The version rollback prevention feature utilizes USERCODE to track the revision. When this feature is enabled, make sure that the value of the USERCODE is *incremental* with the version, that is, a later version has a larger number USERCODE.

6.3. Security Options

The Security options allow you to select from a range of options for tracking or securing the MachXO3D device. Table 6.5 provides a summary of these options.

Table 6.5. Security Options

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

6.3.1. TRACEID

TraceID stamps each MachXO3D device with a unique 64-bit ID. No two MachXO3D devices have 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 (TN1207).

6.3.2. MY_ASSP

Every Lattice device has its own identification code identifying the device family, device density, and other parameters, for example, voltage and device stepping. The code is accessible from any MachXO3D device 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 MY_ASSP ON enables the CUSTOM_IDCODE preference.

6.3.3. CUSTOM_IDCODE

The CUSTOM_IDCODE is the value you assign to override the default IDCODE in the MachXO3D 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 MachXO3D device and, as a result, prevents Diamond Programmer from directly programming the MachXO3D device. It is necessary to migrate to generating Serial Vector Format (SVF) files to program MY_ASSP enabled MachXO3D devices.

6.3.4. CONFIG_SECURE

When this preference is set to ON, the read-back of the SRAM memory and the Flash memory are blocked. The MachXO3D device cannot be read back, nor can it be programmed without erasing. The device must be erased to reset the security setting. The CONFIG_SECURE fuse and the Flash are erased in tandem. Once the security fuses are reset, the device can be programmed again.

6.3.5. BACKGROUND_RECONFIG

The BACKGROUND_RECONFIG preference specifies the behavior of the PROGRAMN pin and the sysCONFIG REFRESH command. It has four possible settings:

- OFF (default) – Toggling PROGRAMN or transmitting the REFRESH command initiates a standard 'warm-boot' SRAM reconfiguration sequence from the specified internal or external configuration image or dual-boot images. The 'warm-boot' sequence executes initialization, configuration, and wake-up steps before reentering user mode (see the Configuration Process and Flow section).
- SRAM_ONLY – Toggling PROGRAMN or transmitting the REFRESH command initiates a 'background' SRAM reconfiguration sequence from the specified internal or external configuration image or dual-boot images. 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. With this setting, the SRAM cells except EBRs and all hardened blocks in the MachXO3D device are updated during the background reconfiguration.
- SRAM_EBR – The function is the same as that of the SRAM_ONLY, except that EBRs are also updated during the background reconfiguration.
- ON – The function is the same as that of SRAM_EBR, except that all hardened blocks in the MachXO3D device are updated during the background reconfiguration.

Note: When the ENABLE_TRANSFR option is set to ENABLE, BACKGROUND_RECONFIG must be set to ON.

6.3.6. BACKGROUND_RECONFIG_SECURITY

The BACKGROUND_RECONFIG_SECURITY preference controls the user logic to access the MachXO3D device embedded security block, or ESB in transparent access mode. When set to default OFF, the user logic is allowed access to the ESB in transparent access mode. When set to ON, the configuration logic takes over control of the security engine module in transparent access/reconfiguration mode and user access to the ESB in transparent access mode is suspended. This option must be turned on to allow the configuration logic to utilize the security engine to perform bitstream authentication and decryption, for example during signed bitstream reconfiguration in background/transparent mode. Note that bitstream reconfiguration in background/transparent mode requires the ENABLE_TRANSFR option to be set to ENABLE.

7. Device Wake-up Sequence

When configuration is completed, that is, the SRAM is loaded, the device wakes up in a predictable fashion. If the MachXO3D 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 then the wake-up process begins. Figure 7.1 shows the wake-up sequence using the internal clock.

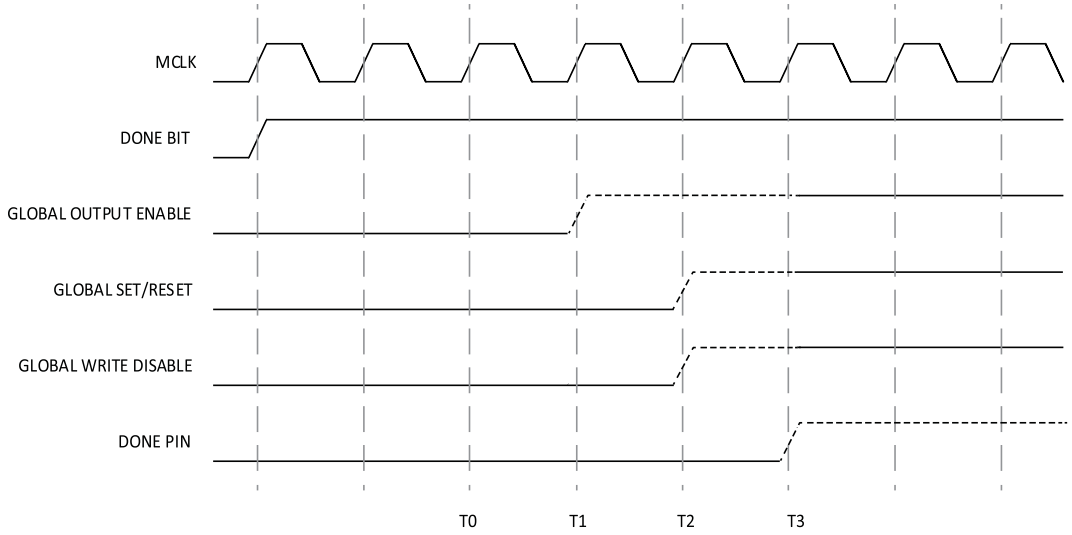


Figure 7.1. Wake-up Sequence Using Internal Clock

7.1. Wake-up Signals

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

- GSR – GSR is used to set and reset the core of the device. GSR is asserted low during configuration and de-asserted high in the wake-up sequence.
- GWDIS – 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 thew EBR and LUTs.
- GOE – When low, GOE prevents the device I/O buffers from driving the pins. The GOE only controls output pins. Once the internal DONE is asserted, the MachXO3D device responds to input data.

  When high, the DONE pin indicates that the configuration is complete and that no errors are detected.

7.2. Wake-up Clock Selection

The clock source used to complete the four state transitions in the wake-up sequence is user-selectable. Once the MachXO3D device 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 file. The clock must be supplied on an external input pin because the MachXO3D device does not begin internal operations until the wake-up sequence is completed. There is no external indication that the device is ready to perform the last four state transitions. You must either provide a free running clock frequency or 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>);
```

8. Advanced Configuration Information

8.1. Flash Programming

The MachXO3D device internal flash is the heart of the FPGA configuration system. It is flexible, allowing you to store the FPGA configuration data, as well as storing design specific data in the internal memory. It is also a resource that uses a precise erasing and programming sequence. Lattice provides several methods for programming the MachXO3D 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 the Using Hardened Control Functions in MachXO3D (FPGA-TN-02117), permits creation of a custom solution.

The Flash 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 MachXO3D Flash resources. The WISHBONE interface is only permitted to use transparent programming operations. The sequence and timing of the commands presented to the configuration logic are identical across all 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 MachXO3D device contains a certain quantity of internal memory. The amount of memory depends on the MachXO3D device density. Refer to Table 9.1 and Table 10.1 in Using Hardened Control Functions in MachXO3D (FPGA-TN-02117) for the number of internal memory pages available for each MachXO3D device density.

8.2. MachXO3D Device 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, no matter whether in hex or binary format. Relevant details about the JEDEC file are provided in Table 8.1 for completeness.

Table 8.1. MachXO3D Device 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.
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 JEDEC files add OTE to the N keyword to make it a more meaningful word NOTE.
Fuse Count	QF3627736	The keyword QF identifies the total real fuse count of the device.
Default Fuse State	F0 or F1	The keyword F identifies the fuse state of those fuses not included in the link field. F0 = fill them with zeroes (0), F1 = fill them with ones (1). It is defined for the purpose of reducing JEDEC file size. It has no meaning in Lattice 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>
Link Field	L0000000	The keyword L identifies the first fuse address of the fuse pattern that

JEDEC Field	Syntax	Description
	101011…100011 …….. 111111…101100 110 101011…100011 …….. 111111…101100 110 …….. …….. 101011…100011 …….. 111111…101100 110* NOTE SED_CRC* L3627704 111111….111111* CC1B9	follows after the white space. The number of digits 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 zeroes. 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. UFM pages, if present in the JEDEC, are preceded by a *NOTE USER MEMORY DATA UFM0** line or a *NOTE USER MEMORY DATA UFM1** 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 bits or 16 bytes per packet, bounded.
Fuse Checksum	CC1B9	The checksum of all the fuses = Fuse count. The fuse state of all the fuses are 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 are found in 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.
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.

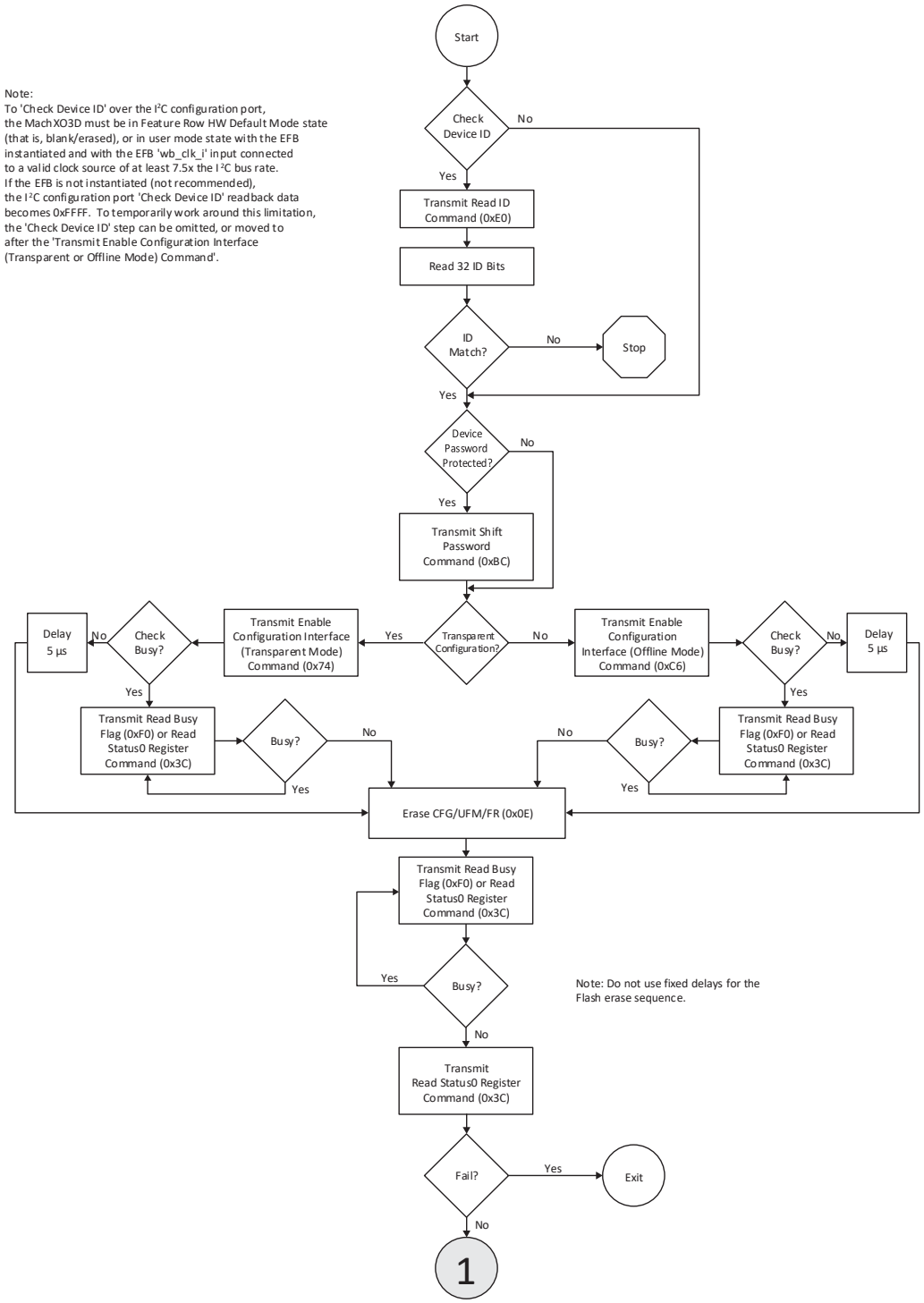
An example of a MachXO3D device JEDEC file is shown in Figure 8.1.

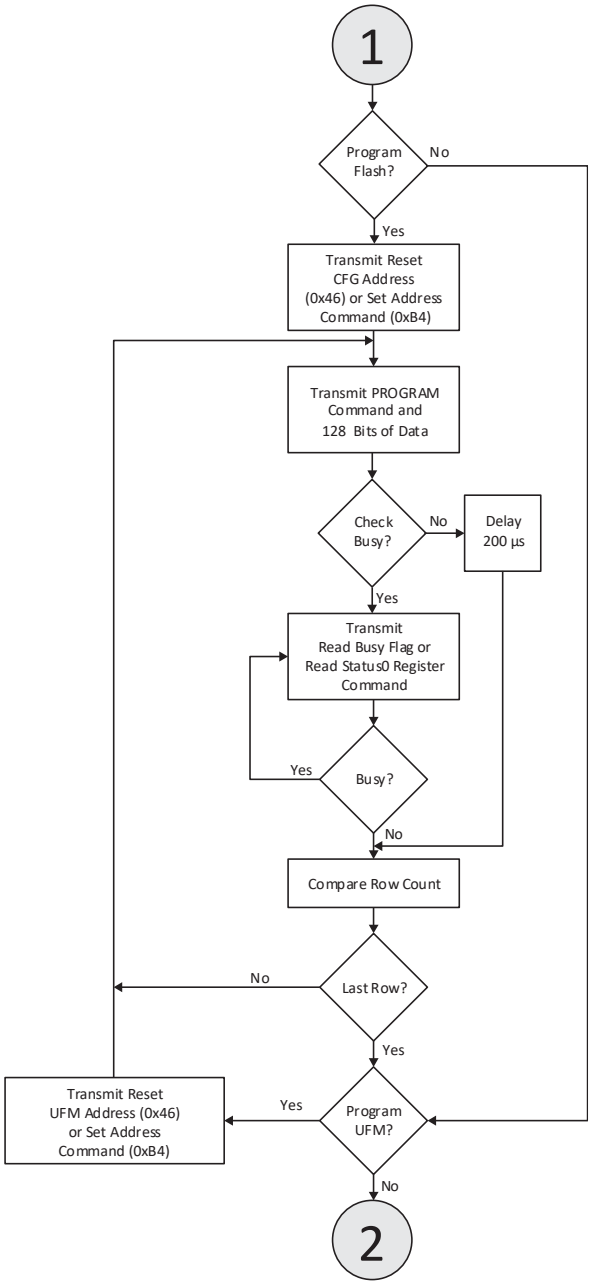
```
*
NOTE Diamond (64-bit) 3.11.0.187.1 JEDEC Compatible Fuse File.*
NOTE Copyright (C), 1992-2010, Lattice Semiconductor Corporation.*
NOTE All Rights Reserved.*
NOTE DATE CREATED:      Tue Jun 26 09:24:31 2018*
NOTE DESIGN NAME:       SimpleDesign_impl1.ncd*
NOTE DEVICE NAME:       LCMXO3D-6900HC-5TQFP144*
NOTE JEDEC FILE STATUS: Advanced   Version 1.6*
NOTE PIN ASSIGNMENTS*
NOTE PINS DOUT[3] : 82 : out*
NOTE PINS DOUT[4] : 83 : out*
NOTE PINS DOUT[5] : 91 : out*
NOTE PINS DOUT[6] : 77 : out*
NOTE PINS DOUT[7] : 95 : out*
NOTE PINS DOUT[2] : 84 : out*
NOTE PINS DOUT[1] : 94 : out*
...
NOTE PINS wb_dat_i[1] : 13 : in*
NOTE PINS wb_dat_i[0] : 1 : in*
QP144*
QF2063872*
G0*
F0*
L0000000
11111111111111111111111111111111111111111111111111101111011011001111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111001110110000000000000000000000000011100010000000000
00000000000000000010001001011100011000001000011000000100000000000000000000000000101000000101001001000001010000000001010100100 0
00110000000000110000100001000000000001000001000000000010001000000000000110000000001000100000000000000000000000000000000000000000
00000000000000000100011000000000000000000000000000001000101000000000000000000000000011111111111111111111111111111111111111101000110
00000000000000000000000101110001110000000000011011110000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
...
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000010101110000011011111111111111111111111111111111111111111111111111111111111111111111000010100000000000000000000000011001010
11111110101110101011111000110001000100110010001100000000000000000000000000110001010000000000000000000000000111111111111111111111111
*
NOTE EBR_INIT DATA*
L275072
11111111111111111111111111111111111111011000000000000000000000000000000000000000000010000000000000101100100001000000000000010000000
00000000000000000100000001000000001100000010000000001010000001100000001110000001000000000100100000101000000010110000011000000001101 00
000111000000111100001000000000100010000100100000100110000101000000101010000101100000101110000110000000110010000110100000110110000
...
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00100011000000000000000000000000011000101000000000000000000000000001011110000000000000000000000000011111111111111111111111111111111
*
NOTE END CONFIG DATA*
L284544
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
...
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
*
NOTE USER MEMORY DATA UFM0*
L2062592
00000000000000010000000100000001100000010000000010100000110000001110000100000001001000010100000101100001100000011010000111000001111
0001000000010001000100100001001100010100000010101000101100001011100011000000110010001101000011011000111000000111010001111000011111
00000000000000010000000100000001100000010000000010100000110000001110000100000001001000010100000101100001100000011010000111000001111
0001000000010001000100100001001100010100000010101000101100001011100011000000110010001101000011011000111000000111010001111000011111
00000000000000010000000100000001100000010000000010100000110000001110000100000001001000010100000101100001100000011010000111000001111
0001000000010001000100100001001100010100000010101000101100001011100011000000110010001101000011011000111000000111010001111000011111
00000000000000010000000100000001100000010000000010100000110000001110000100000001001000010100000101100001100000011010000111000001111
0001000000010001000100100001001100010100000010101000101100001011100011000000110010001101000011011000111000000111010001111000011111
00000000000000010000000100000001100000010000000010100000110000001110000100000001001000010100000101100001100000011010000111000001111
0001000000010001000100100001001100010100000010101000101100001011100011000000110010001101000011011000111000000111010001111000000000
*
C83B5*
NOTE User Electronic Signature Data*
UHCAFEBABE*
A6FE
```

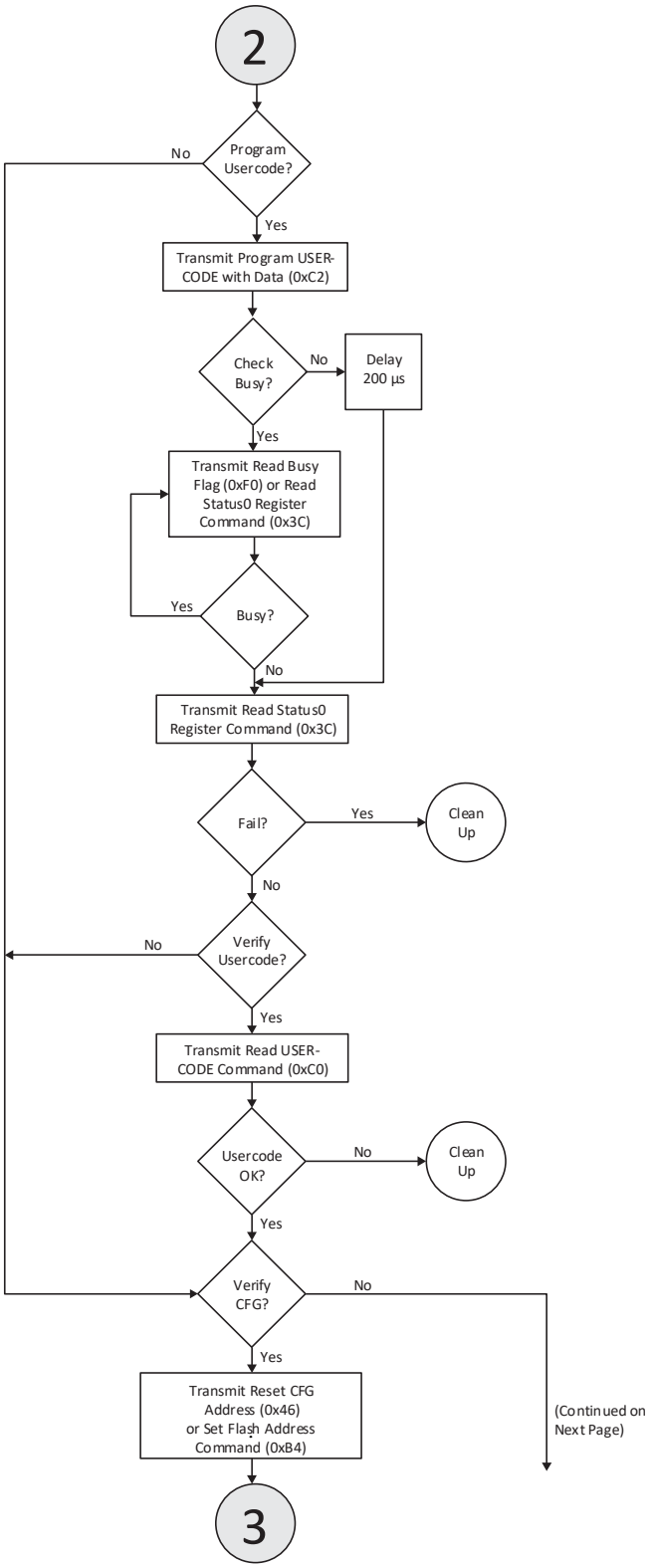
Figure 8.1. JEDEC File Example

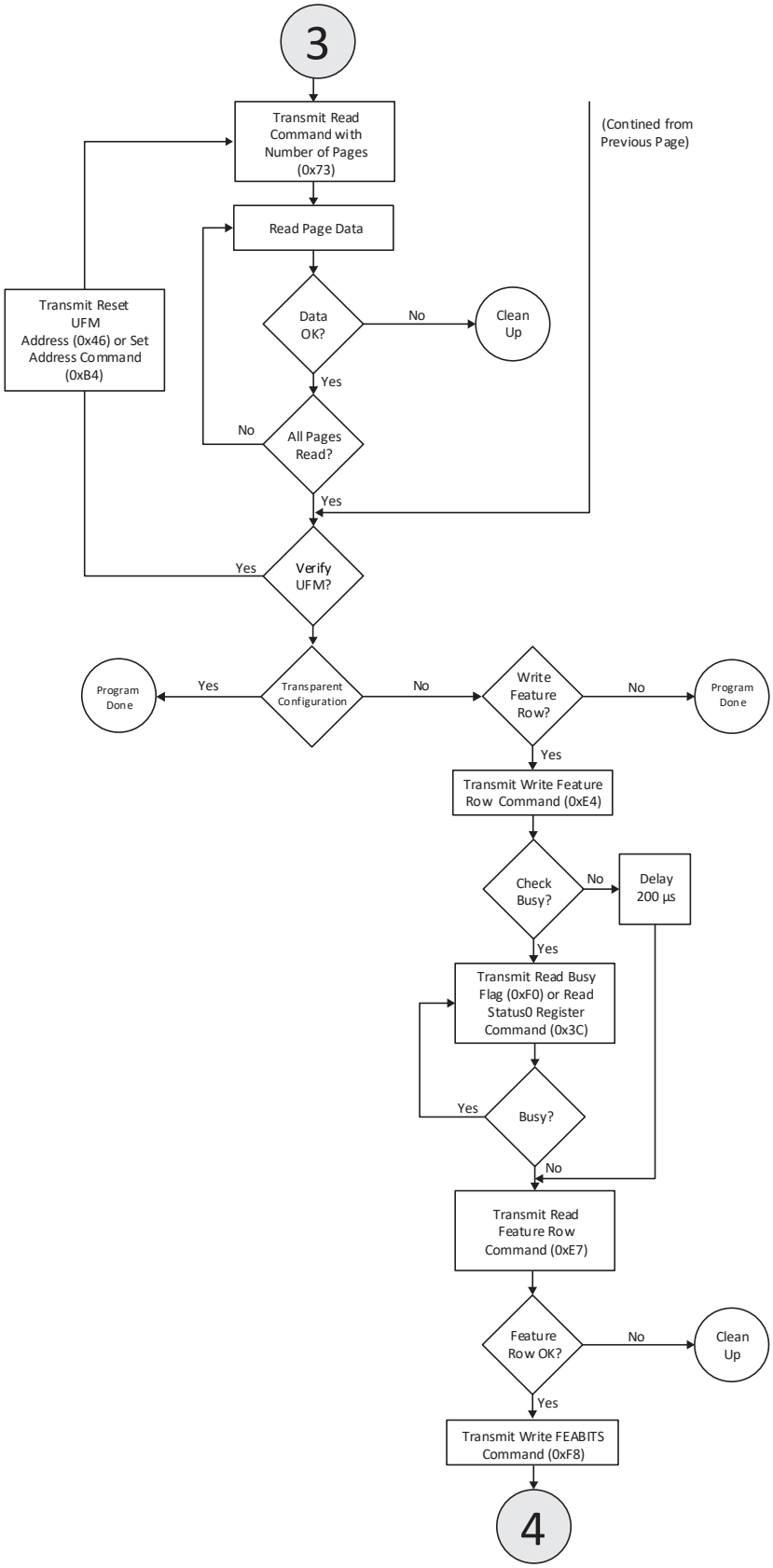
8.3. MachXO3D Flash Programming Flow

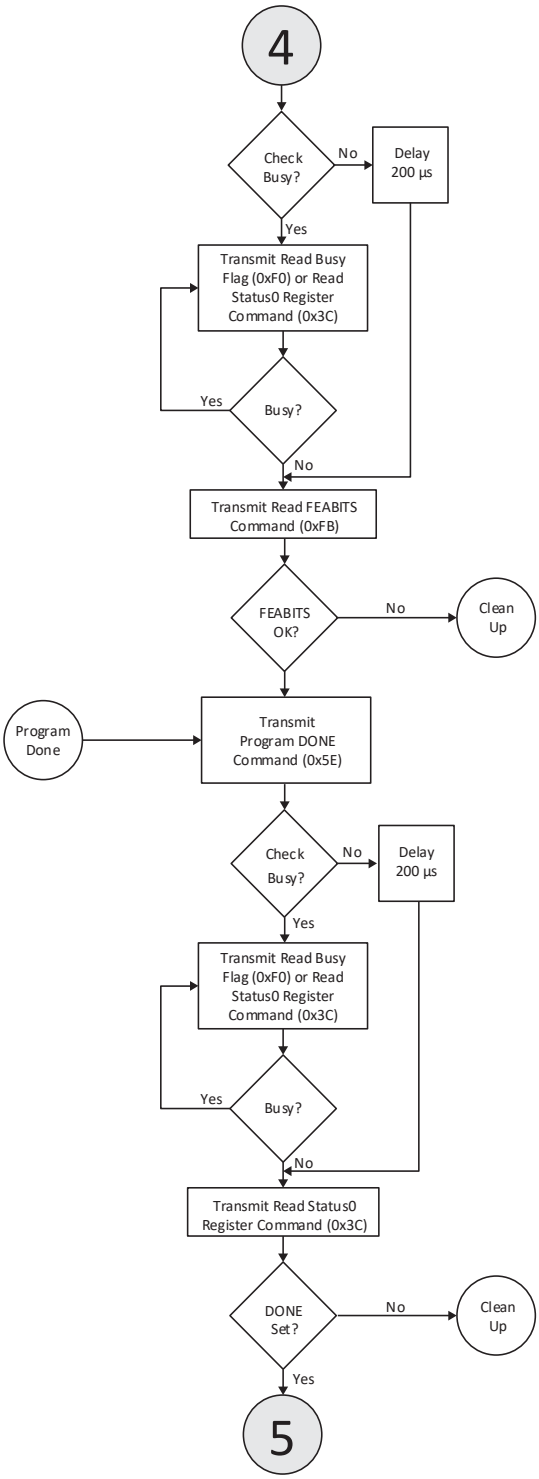
The MachXO3D Flash memory erasing and programming require a specific set of steps and timing. The flowchart in this section describes the command sequences and the timing required for successful Flash programming. The commands and timing are common among all of the configuration ports. There are some minor variations in the protocol, but not the timing, based on the configuration port used.

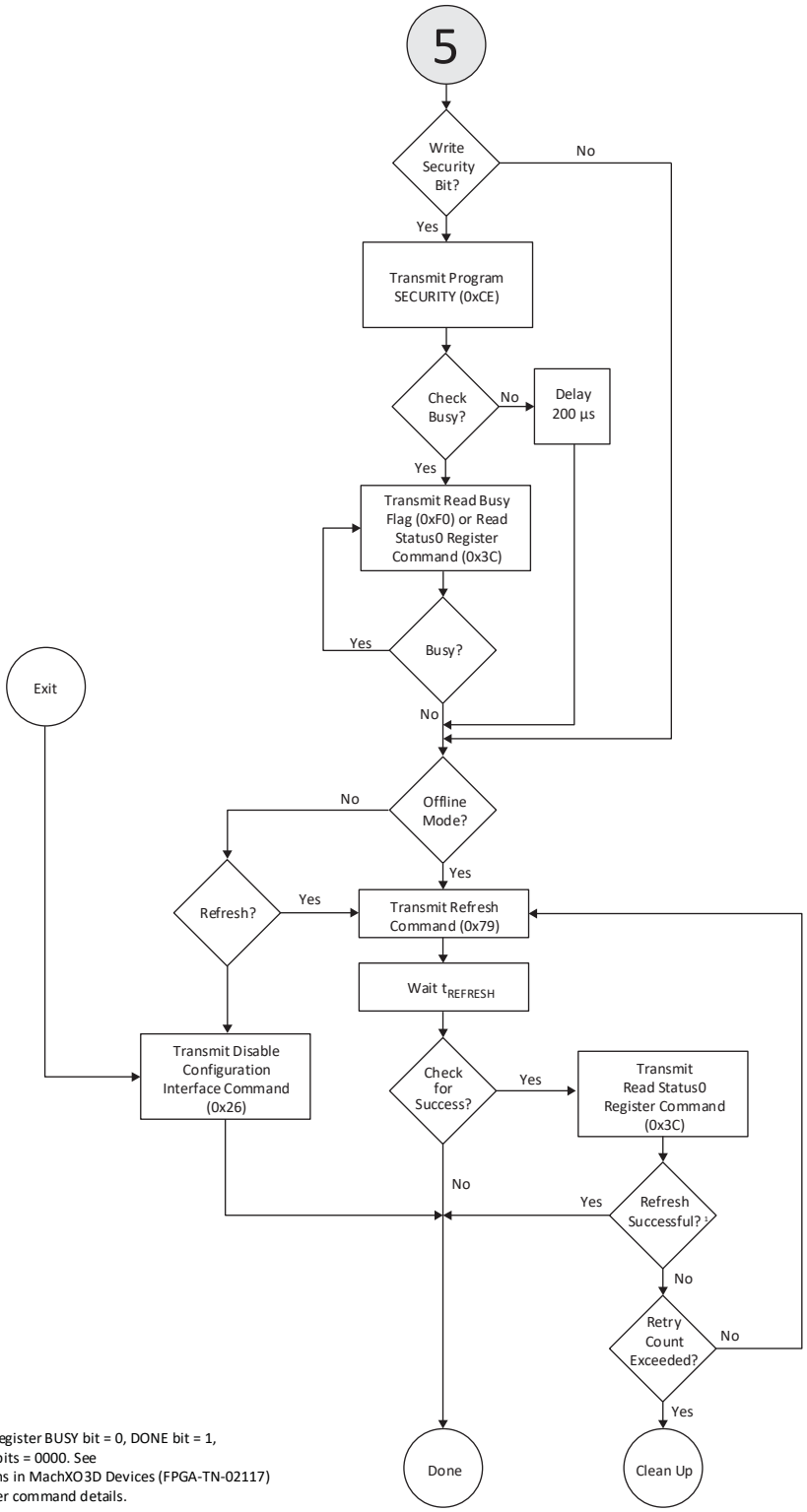












Notes:
Refresh is successful if Status0 Register BUSY bit = 0, DONE bit = 1, and Configuration Check Status bits = 0000. See Using Hardened Control Functions in MachXO3D Devices (FPGA-TN-02117) for complete Read Status Register command details.

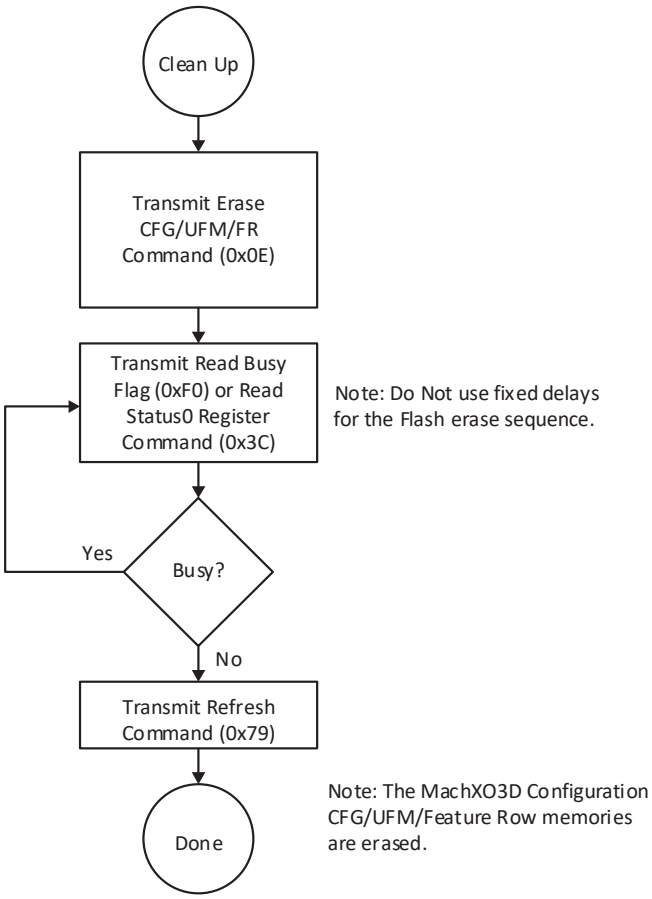
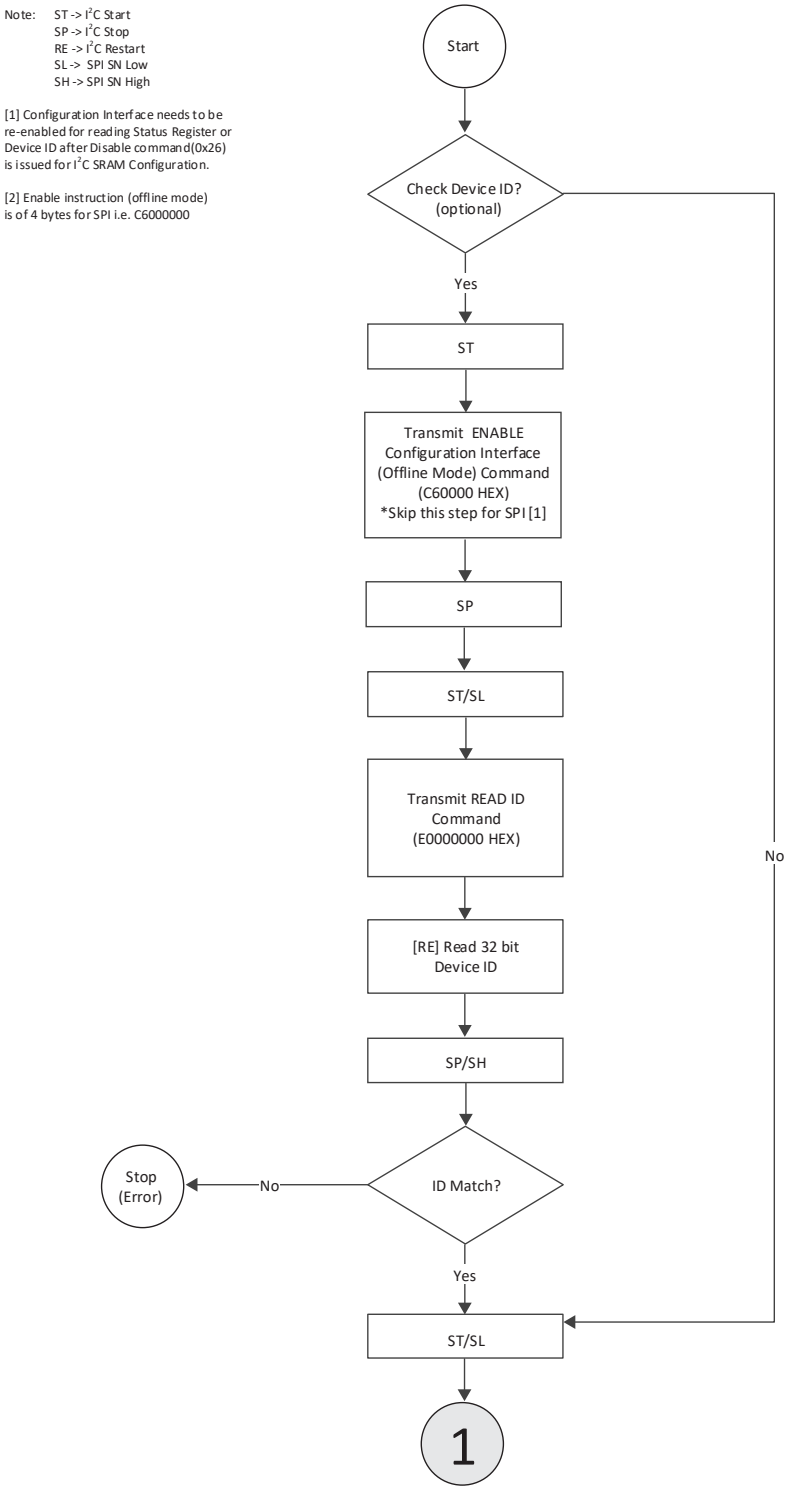
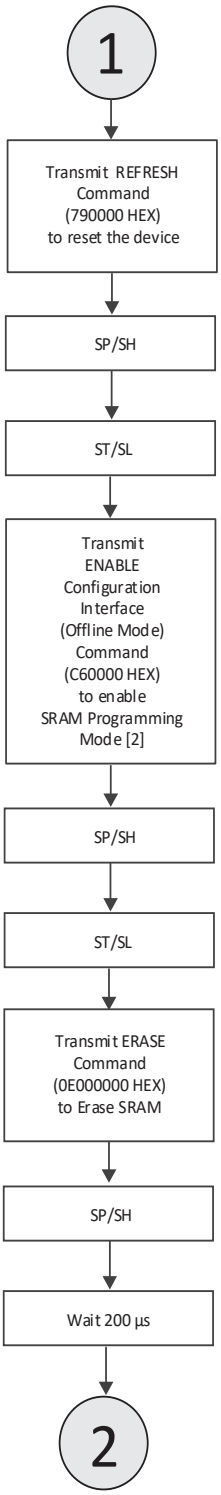


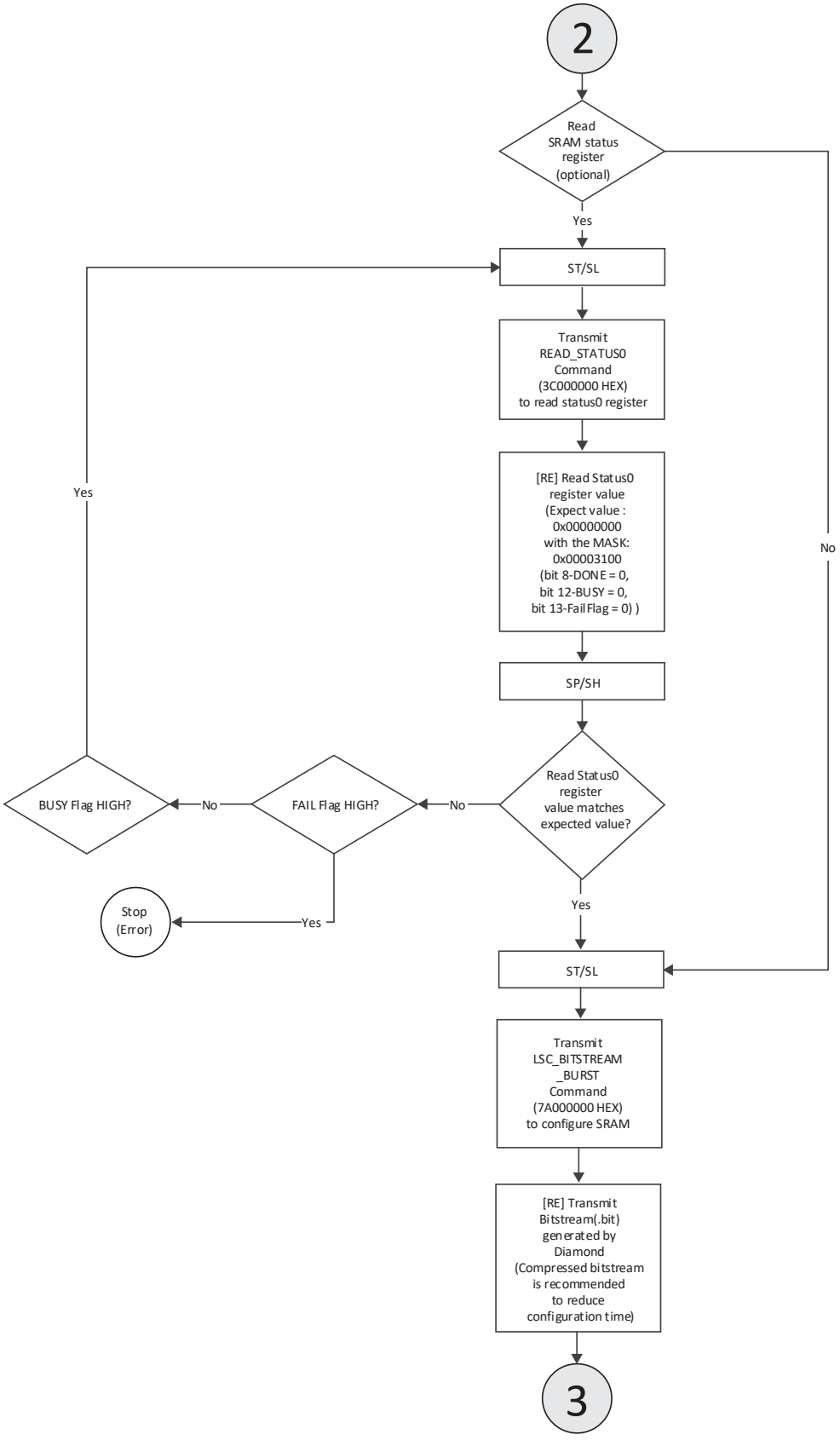
Figure 8.2. MachXO3D Flash Memory Programming Flow

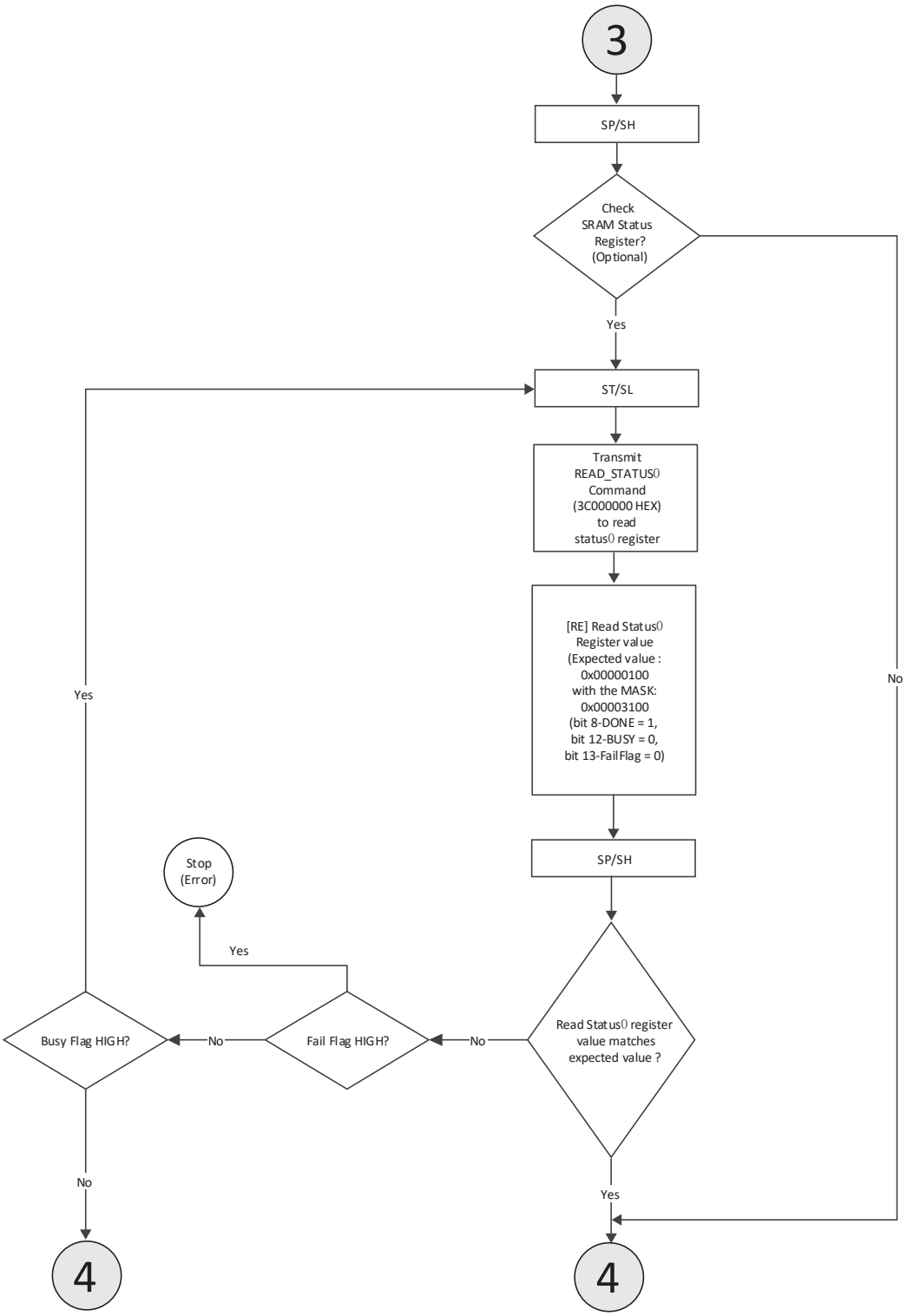
8.4. MachXO3D Slave SPI/I²C SRAM Configuration Flow

MachXO3D 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.









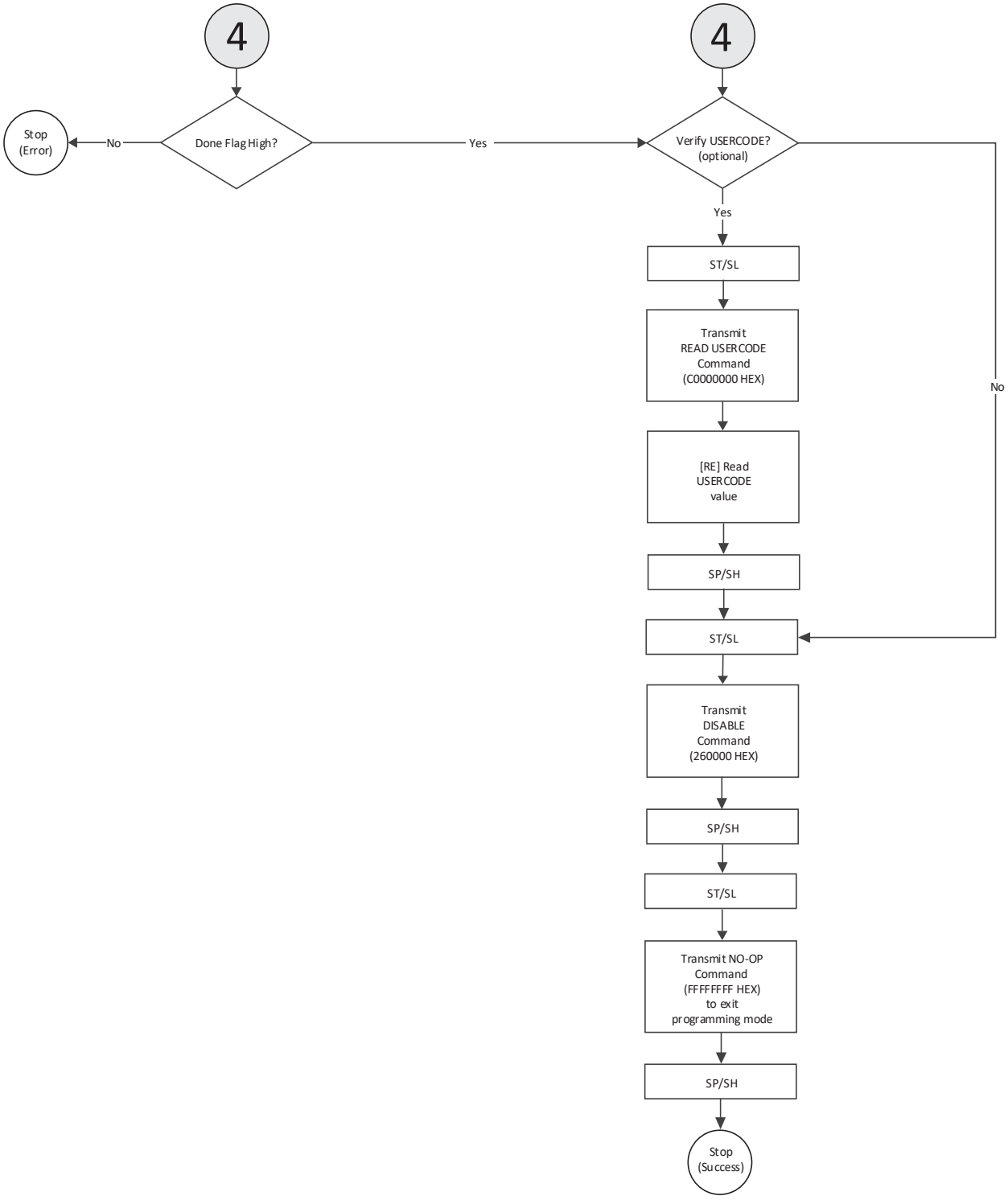
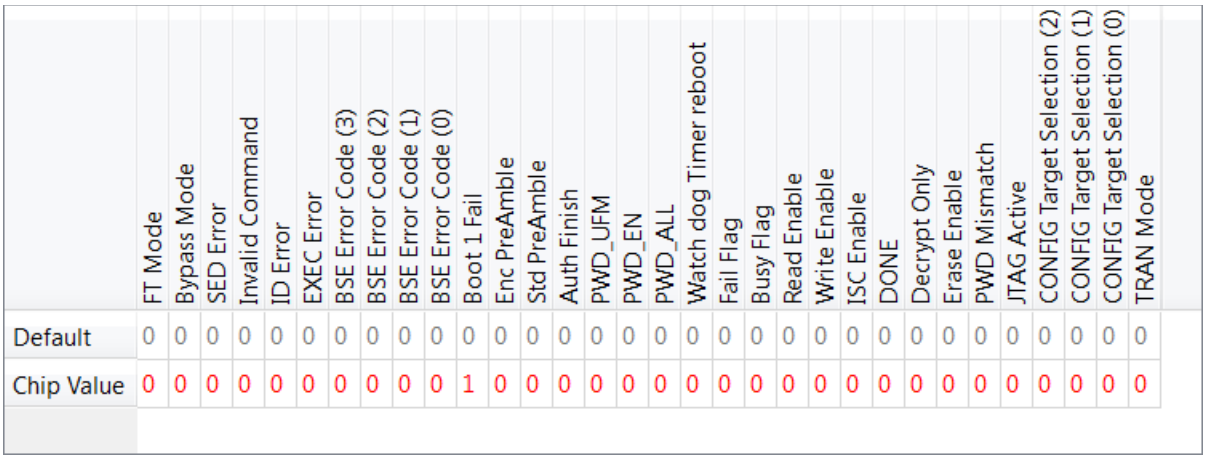


Figure 8.3. MachXO3D Slave SPI/I^{2}C SRAM Configuration Flow



	FT Mode	Bypass Mode	SED Error	Invalid Command	ID Error	EXEC Error	BSE Error Code (3)	BSE Error Code (2)	BSE Error Code (1)	BSE Error Code (0)	Boot 1 Fail	Enc PreAmble	Std PreAmble	Auth Finish	PWD_UFM	PWD_EN	PWD_ALL	Watch dog Timer reboot	Fail Flag	Busy Flag	Read Enable	Write Enable	ISC Enable	DONE	Decrypt Only	Erase Enable	PWD Mismatch	JTAG Active	CONFIG Target Selection (2)	CONFIG Target Selection (1)	CONFIG Target Selection (0)	TRAN Mode
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

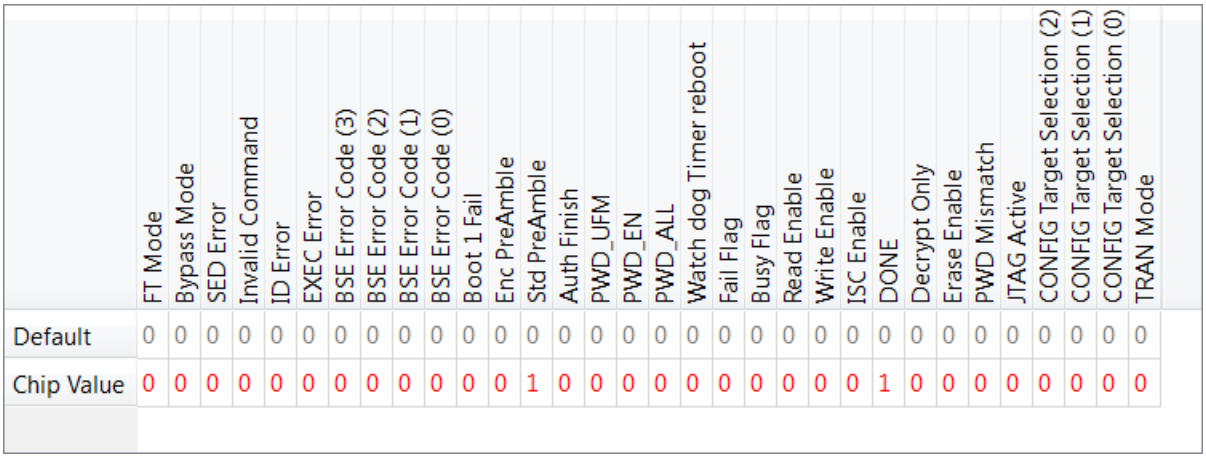
Figure 8.4. Status0 Register Value after Erasing

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



	FT Mode	Bypass Mode	SED Error	Invalid Command	ID Error	EXEC Error	BSE Error Code (3)	BSE Error Code (2)	BSE Error Code (1)	BSE Error Code (0)	Boot 1 Fail	Enc PreAmble	Std PreAmble	Auth Finish	PWD_UFM	PWD_EN	PWD_ALL	Watch dog Timer reboot	Fail Flag	Busy Flag	Read Enable	Write Enable	ISC Enable	DONE	Decrypt Only	Erase Enable	PWD Mismatch	JTAG Active	CONFIG Target Selection (2)	CONFIG Target Selection (1)	CONFIG Target Selection (0)	TRAN Mode
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

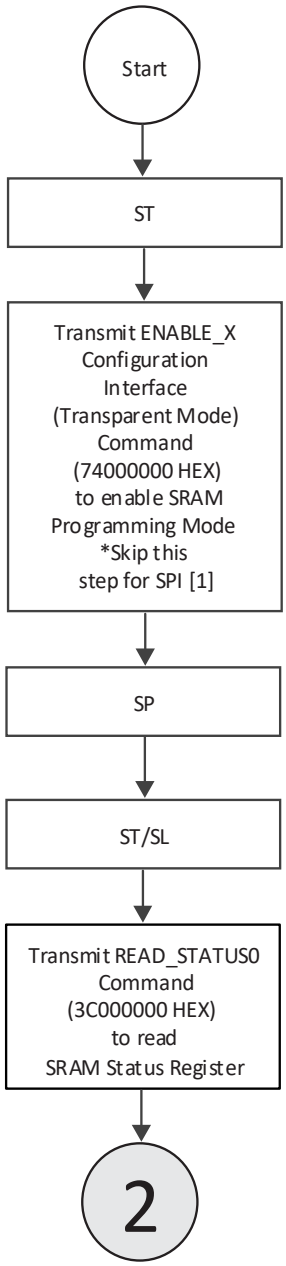
Figure 8.5. Status0 Register Value after Programming

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



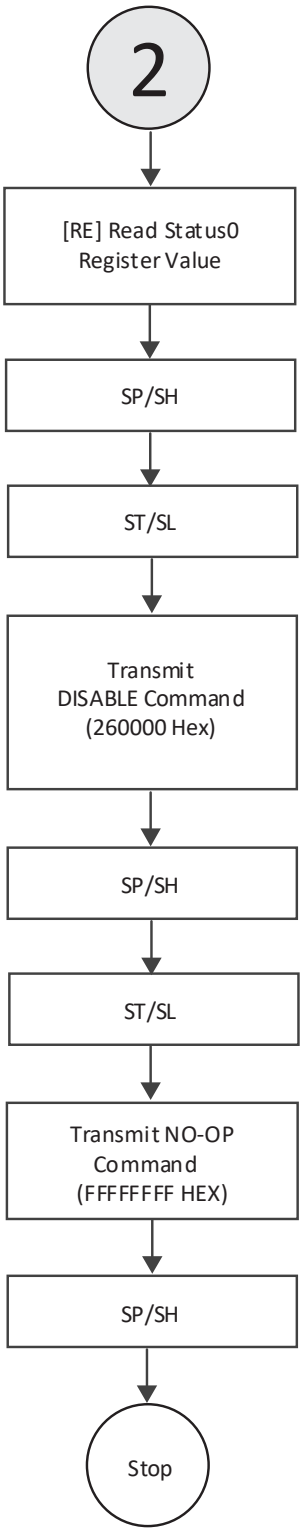


Figure 8.6. Slave SPI/I²C SRAM Read Status0 Register Flow

8.5. MachXO3D Programming Commands

Table 8.2. MachXO3D 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	0Y 00 00[1]	N/A	N/A	Enables the configuration logic for transparent SRAM read access or transparent flash programming access. Y identifies the memory type to access.[1] Bit 1 0 19 Flash SRAM
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 Status0 Register [LSC_READ_STATUS0]	0x3C	00 00 00	N/A	YY YY YY YY	Bit 1 0 12 Busy Ready 13 Fail OK Refer to the Status Registers section for more information.
Read Status1 Register [LSC_READ_STATUS1]	0x3D	00 00 00	N/A	YY YY YY YY	Read Status Register 1 Refer to the Status Registers section for more information.
Erase [ISC_ERASE]	0x0E	0Y YY 00	N/A	N/A	Erase the different internal memories. The bit in YYY defines which memory is erased in Flash access mode. Bit 1=Enable 8 Erase CFG0 9 Erase CFG1 10 Erase UFM0 11 Erase UFM1 12 Erase UFM2 13 Erase UFM3 14 Erase CSEC 15 Erase USEC 16 Erase PUBKEY 17 Erase AESKEY 18 Erase FEA 19 Reserved In SRAM access mode this command is used to erase SRAM with all bits of operands reserved.
Erase UFM [LSC_ERASE_TAG]	0xCB	00 YY 00	N/A	N/A	Erase the UFM sectors. The bit in YY defines which sector is erased in Flash access mode. Bit 1=Enable 8 Reserved 9 Reserved 10 Erase UFM0 11 Erase UFM1

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
					12 Erase UFM2 13 Erase UFM3 14 Reserved 15 Erase USEC
Reset Memory Address [LSC_INIT_ADDRESS]	0x46	YY YY 00	N/A	N/A	Set Page Address pointer to the beginning of the different internal Flash sectors. The bit in YYYY defines which sector is selected. Bit Flash sector selected 8 CFG0 9 CFG1 10 UFM0 11 UFM1 12 UFM2 13 UFM3 14 CSEC 15 USEC 16 PUBKEY 17 AESKEY 18 FEA 19 TRIM 20 Reserved 21 Reserved 22 Reserved Only one of the above bits is allowed to be set at a time. Otherwise, no sector is selected. Bit 23 is CRC check bit, 0 for no CRC check and 1 for CRC check.
Set Address [LSC_WRITE_ADDRESS]	0xB4	00 00 00	00 0P PP PP	N/A	Set the Page Address pointer to the Flash page specified by the least significant 18 bits of the P PP PP field. Bit17 ~ Bit14 selects the Flash space to access. Bit[17:14] Flash sector selected 4'h0 CFG0 4'h1 UFM0 4'h2 Reserved 4'h3 FEA 4'h4 CFG1 4'h5 UFM1 4'h6 PUBKEY 4'h7 CSEC 4'h8 UFM2 4'h9 UFM3 4'hA ASEKEY 4'hB USEC Bit13 ~ Bit0 defines the page address.
Program Page [LSC_PROG_INCR_NV]	0x70	00 00 01	YY * 16	N/A	Program one Flash page. Can be used to program the CFG or UFM.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Reset UFM Address [LSC_INIT_ADDR_UFM]	0x47	00 YY 00	N/A	N/A	Set the Page Address Pointer to the beginning of the UFM sectors. The bit in YY defines which sector is selected. Bit UFM sector selected 8 Reserved 9 Reserved 10 UFM0 11 UFM1 12 UFM2 13 UFM3 14 Reserved 15 USEC Only one of the above bits is allowed to be set at a time. Otherwise, no sector is selected.
Program UFM Page [LSC_PROG_TAG]	0xC9	00 00 01	YY * 16	N/A	Program one 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 * 16	N/A	Program the Feature Row bits.
Read Feature Row [LSC_READ_FEATURE]	0xE7	00 00 00	N/A	YY * 16	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 Flash [LSC_READ_INCR_NV]	0x73	M0 PP PP	N/A	See the Reading 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 Flash. 0x0 I^2C 0x0 or 0x1 JTAG/SSPI
Read UFM [LSC_READ_UFM]	0xCA	M0 PP PP	N/A	See the Reading Flash Pages section.	Retrieves PPPP count 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 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.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
Program AES feature bits [LSC_PROG_AES_FEA]	0xF9	00 00 00	PP	N/A	Program AES feature decryption bits Bit 1=Enable 0 DEC_ONLY 1 RAND_AES 2 RAND_NOISE
Read AES feature bits [LSC_READ_AES_FEA]	0xFA	00 00 00	N/A	PP	Read AES feature decryption bits Bit 1=Enable 0 DEC_ONLY 1 RAND_AES 2 RAND_NOISE
Disable Configuration Interface [ISC_DISABLE]	0x26	0000	N/A	N/A	Exit Offline or Transparent programming mode. ISC_DISABLE causes the MachXO3D 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 8.2.
Bypass [ISC_NOOP]	0xFF	FF FF FF	N/A	N/A	No Operation and Device Wakeup.
Refresh [LSC_REFRESH]	0x79	0000	N/A	N/A	Force the MachXO3D to reconfigure. Transmitting a REFRESH command reconfigures the MachXO3D in the same fashion as asserting PROGRAMN.
Program SECURITY [ISC_PROGRAM_SECURITY]	0xCE	00 00 00	YY	N/A	Programs security or lock bits to current sector of Flash memory. Bit 1=Enable 0 SEC_ERASE 1 SEC_PROG 2 SEC_READ 3 SEC_HLOCK See the Lock Bits and Lock Control Policy section.
Program SECURITY PLUS [ISC_PROGRAM_SECPLUS]	0xCF	00 00 00	N/A	N/A	Program SECPLUS bit to set the same security setting of CFG0 or CFG1 sector to the UFM0 or UFM1 sector in case the bitstream is across into the UFM sector.
Program AUTH_DONE [LSC_PROG_AUTH_DONE]	0xCC	00 00 00	N/A	N/A	Program AUTH_DONE bit in CFG0 or CFG1 if pre-authentication feature is required.
Read TraceID code [UIDCODE_PUB]	0x19	00 00 00	N/A	YY*8	Read 64-bit TraceID.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
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*16	N/A	Program the 128-bit Password into the device.
Read Flash Protect Key [LSC_READ_PASSWORD]	0xF2	00 00 00	N/A	YY*16	Read the 128-bit Password from the device.
Shift Flash Protect Key [LSC_SHIFT_PASSWORD]	0xBC	00 00 00	YY*16	N/A	Present the 128-bit Password. When enabled (PWD_EN = 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.
Program Control Register 0 [LSC_PROG_CTRL0]	0x22	00 00 00	YY * 4	N/A	Program the 32-bit Control Register 0. Refer to the Control Registers section for more information.
Program Control Register 1 [LSC_PROG_CTRL1]	0x23	00 00 00	YY * 4	N/A	Program the 32-bit Control Register 1. Refer to the Control Registers section for more information.
Read Control Register 0 [LSC_READ_CTRL0]	0x20	00 00 00	N/A	YY * 4	Read 32 bits of Control Register 0. Refer to the Control Registers section for more information.
Read Control Register 1 [LSC_READ_CTRL1]	0x21	00 00 00	N/A	YY * 4	Read 32 bits of Control Register 1. Refer to the Control Registers section for more information.
Device Control [LSC_DEVICE_CTRL]	0x7D	YY 00 00	N/A	N/A	The 8 bits in YY define the real-time control actions. Bit 1=Enable 23 Reserved 22 Reset CFG logic [21:19] CHECK operations (can be used to control SED and launch dry run for internal flash sections): 000: No action 001: Launch one-time check for SED 010: Launch a bitstream dry run for CFG0 sector. It reads the on-die bitstream and checks the CRC. This command will not

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Notes
					actually write the configuration SRAM. This is done in the background during normal device operation. 011: Launch bitstream dry run for CFG1 sector 18 WAKEUP 17 STANDBY 16 Reserved

Note: Transmit the command opcode and first two operand bytes when using the I^2C port. The final operand byte must not be transmitted.

8.6. Reading Flash Pages

Reading the CFG and UFM pages requires a specific procedure. The CFG and UFM pages are accessible from any of the MachXO3D device 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 8.8.

All ports start the read process in the same way, by sending a Read Flash/Read UFM command. The MachXO3D device begins the read process once the command byte is accepted by the configuration logic. The Page Address Pointer determines the first page returned from the MachXO3D device. For the first returned page to be valid (for example, for single-page read operations), a retrieval delay of 240 ns 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, refer Figure 8.7. Note that for slower interface clock rates, 240 ns may be consumed entirely by the normal transmission of the first operand, and no additional delay may be necessary.

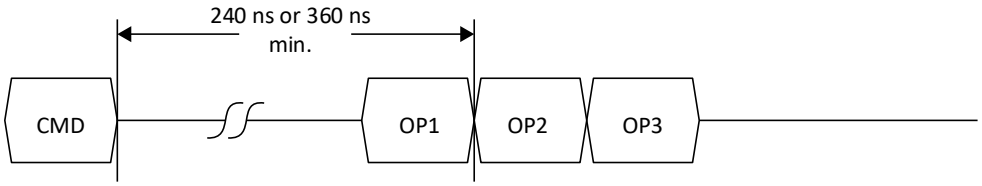
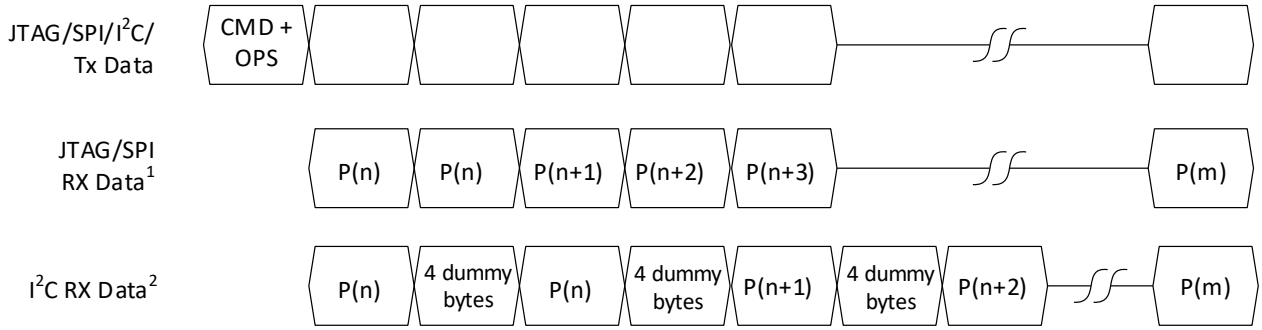


Figure 8.7. Retrieval Delay Timing Requirement for Single-Page Reads



Notes:
1. JTAG/SSPI must transmit data to read data back. The data sent by the JTAG/SSPI master is not specified (that is, 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 CFG or Read UFM command byte + 3 operand bytes.

Figure 8.8. Flash Page Command and Data Sequence

Figure 8.8 shows a multiple page read sequence. The Read CFG/Read UFM command is transmitted to the MachXO3D device. As shown in Figure 8.8, all interfaces return the page to 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 CFG/Read UFM operand bytes.

Reading more than one page requires special handling. The multiple pages 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 CFG/Read UFM command to be assigned a value of 3. If the Page Address Pointer is 0000, the MachXO3D device returns three pages, Page 0, Page 0, and Page 1. A restriction must be observed when using the WISHBONE interface to read the 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 are read.

The I^2C interface has additional overhead when reading Flash pages. Reviewing Figure 8.8 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, 4 dummy bytes, Page 0, 4 dummy bytes, and Page 1. Reading the final four dummy bytes is optional.

8.7. Status Registers

Table 8.3. Status Register 0

Field	Name	Description
[31]	FlowThrough Mode (FT Mode)	Device is in flow-through mode to feed the bitstream data to the downstream daisy chain devices.
[30]	Bypass Mode	Device is in bypass mode to feed the bitstream data to the downstream daisy chain devices.
[29]	SED Error	Soft error detection logic detected an error.
[28]	Invalid Command	Instruction found in the Command field or OPCODE of the frame is invalid.
[27]	ID Error	ID code mismatch detected for the verify_id instruction/command. Once set, this bit can only be cleared by LSC_DEVICE_CTRL (with configuration reset option) or LSC_REFRESH command execution, or PROGRAMN pin pulsing.
[26]	Execution Error (EXEC Error)	Status bit indicates any of the following: • Previous command execution encountered an error. • Bitstream error has occurred.

Field	Name	Description
		• SED error is detected. Once set, this bit can only be cleared by LSC_DEVICE_CTRL (with configuration reset option) command execution or re-entering the ISC ACCESS state.
[25:22]	BSE Error Code	0000 – No Error 0001 – ID Error; a mismatch in the device ID code is detected when executing the VERIFY ID command. 0010 – CMD Error; an illegal command is detected. 0011 – CRC Error; a CRC checksum error is detected. 0100 – PRMB Error; preamble detection error in the following cases: • Decrypt only is set but PREAMBLE_PLN is detected. • Master mode preamble detection fails (time out after defined times). • In SDM mode, no preamble after 16'h8000 clock cycles. 0101 – ABRT Error; bitstream engine execution aborted by user activity. 0110 – OVFL Error; bitstream engine detected last address of SRAM array is programmed but BSE has not terminated. 0111 – SDM EOF Error; bitstream read passed the size of the internal non-volatile memory array. 1000 – AUTH Fail; authentication fails. 1001 – AUTH Setup Error; hardware security engine (HSE) is not available or other setup error. 1010 – AUTH Bitstream Error; bitstream does not match the required format for example, there is no signature in the bitstream. 1011 – TO in Slave Mode; slave mode booting fails due to time out. 1100 – VRBP Version Check Fail; version check for version rollback protection fails (check is only for external bitstream).
[21]	Boot 1 Fail	Primary pattern failed when dual-boot is enabled. Once set, this bit can only be cleared by LSC_DEVICE_CTRL (with configuration reset option) or LSC_REFRESH command execution, or PROGRAMN pin pulsing.
[20]	Encrypt Preamble (Enc PreAmble)	An encrypt preamble (for encrypted bitstream) is detected from the last bitstream execution. This status bit is cleared at the beginning of the next refresh event or the next bitstream engine activation.
[19]	Std Preamble (Std PreAmble)	A standard preamble (for non-encrypted bitstream) is detected from the last bitstream execution. This status bit is cleared at the beginning of the next refresh event or the next bitstream engine activation.
[18]	Auth Finish	Authentication has completed.
[17]	PWD_UFM	Password protection covers UFM sectors.
[16]	PWD_EN	Password protection is enabled.
[15]	PWD_ALL	Password protection is enabled and covers CFG sectors as well.
[14]	Watchdog Timer Reboot	Time out of user mode watch dog timer; reboot triggered.
[13]	Fail Flag	Previous command execution failed.
[12]	Busy Flag	Configuration logic is busy executing a previous command (an execution FSM is actively running).
[11]	Read Enable	0 – Selected configuration target is read-protected if at least one of the following conditions is met: • Selected configuration target security bit (read lock) is set. • Password protection is enabled, and password mismatch occurs. 1 – Selected configuration target is read-enabled.
[10]	Write Enable	0 – Selected configuration target is write-protected if at least one of the following conditions is met: • Selected configuration target security bit (write lock) is set. • Password protection is enabled, and password mismatch occurs.

Field	Name	Description
		• AUTH_DONE bit is set. 1 – Selected configuration target is write-enabled.
[9]	ISC Enable	Device is in ISC ACCESS state and ISC activities are enabled.
[8]	DONE	Device has received and executed the ISC_DONE command.
[7]	Decrypt Only	Decrypt Only bit within the flash memory is set; only encrypted bitstream is accepted.
[6]	Erase Enable	0 – Selected configuration target is erase-protected. Read activity is disabled if at least one of the following conditions is met: • Selected configuration target security bit (erase lock) is set. • Password protection is enabled, and password mismatch occurs. 1 – Selected configuration target is erase-enabled.
[5]	PWD Mismatch	Password protection is enabled, and password mismatch has occurred.
[4]	JTAG Active	A JTAG ISC engine is active.
[3:1]	CONFIG Target Selection	Internal memory array targeted for configuration 000 – SRAM Array (default); write to or read from the SRAM array. 001 – E_Fuse_Normal; write to or read from the e-fuse array (feature row). A single command is used for writes (no verification on-chip before writing to the e-fuse array). 100 – Flash Normal; write to or read from the flash array. A single command is used for writes (no verification on-chip before writing to the flash). 111 – Flash UFM; write to or read from the UFM sector of the flash array. A single command is used for writes (no verification on-chip before writing to the UFM flash).
[0]	TRAN Mode	Device is in transparent mode.

Table 8.4. Status Register 1

Field	Name	Description
[31]	UDS_LOCK	Unique device secret (UDS) page is locked.
[30:29]	SEC_JTAG	JTAG port security setting SEC_JTAG[1] SEC_JTAG[0] Description 0 0 JTAG port is unlocked. All instructions are allowed. 0 1 Partial lock mode. A limited set of IEEE 1149.1 instructions are allowed. 1 0 Partial lock mode. All IEEE 1532 and some IEEE 1149.1 instructions are allowed to support incremental programming in transparent mode. 1 1 JTAG port is locked. No Instructions are allowed.
[28:27]	SEC_SSPI	Slave SPI port security setting SEC_SSPI[1] SEC_SSPI[0] Description 0 0 Target SPI port is unlocked. All commands are allowed. 0 1 Target SPI port is locked (identical to SEC_SSPI[1:0] == 11). 1 0 Partial lock mode. All IEEE 1532 and some IEEE 1149.1 instructions are allowed to support incremental programming in transparent mode. 1 1 Target SPI port is disabled.

Field	Name	Description
[26:25]	SEC_SI2C	Slave I2C port security setting SEC_SI2C[1] SEC_SI2C[0] Description 0 0 Slave I2C port is unlocked. All commands are allowed. 0 1 Slave I2C port is locked (identical to SEC_SI2C[1:0] == 11). 1 0 Partial lock mode. All IEEE 1532 and some IEEE 1149.1 instructions are allowed to support incremental programming in transparent mode. 1 1 Slave I2C port is disabled.
[24]	SEC_BSPI	JTAG-to-SPI bridge security setting 0 – JTAG-to-SPI bridge is enabled. 1 – JTAG-to-SPI bridge is disabled.
[23]	SEC_BI2C	I2C bridge security setting 0 – I2C bridge is enabled. 1 – I2C bridge is disabled.
[22]	I2C_DEG_SEL	Primary I2C port de-glitch range selection.
[21]	I2C_DEG_EN	Primary I2C port de-glitch is enabled.
[20]	PST_MSPI	Persist master SPI sysCONFIG port.
[19:17]	BOOT_SEL	Boot selection mode. Refer to the Program FEABITs (0xF8) section in Using Hardened Control Functions in MachXO3D Devices Reference Guide (FPGA-TN-02119) for more information.
[16]	Bitstream Version	0 – Bitstream in CFG1 is later than CFG0. 1 – Bitstream in CFG0 is later than CFG1.
[15]	CFG1_SEC Plus	SEC_PLUS bit in CFG1 sector.
[14]	CFG0_SEC Plus	SEC_PLUS bit in CFG0 sector.
[13]	CFG1_DONE	DONE bit in CFG1 sector.
[12]	CFG0_DONE	DONE bit in CFG0 sector.
[11]	AUTH_DONE_CFG1	Authentication is done and has passed for the bitstream residing in CFG1.
[10]	AUTH_DONE_CFG0	Authentication is done and has passed for the bitstream residing in CFG0.
[9]	AUTH_EN2	Authentication mode 0 – No authentication is required. 1 – Authentication or verification is required.
[8]	AUTH_EN1	Authentication type AUTH_EN2 AUTH_EN1 Description 0 X No authentication is required. 1 0 HMAC authentication is required. 1 1 ECDSA signature verification is required.
[7]	SEC_HLOCK[1]	Hard lock / Soft lock selection bit 0 – Soft Lock 1 – Hard Lock
[6]	SEC_READ[1]	Read operation is prohibited.
[5]	SEC_PROG[1]	Program operation is prohibited.
[4]	SEC_ERASE[1]	Erase operation is prohibited.
[3:0]	Current Sector	Current flash sector being accessed 0001 – CFG0 0010 – CFG1 0011 – TRIM 0100 – FEA 0101 – PUBKEY

Field	Name	Description
		0110 – AESKEY 0111 – CSEC 1001 – UFM1 1011 – UFM2 1100 – UFM3 1101 - USEC

Note:

1. Bit related to security policy definition. In flash access mode, these bits show the security setting of the current sector after combining local and central security setting bits. In SRAM access mode, these bits show the SRAM security setting.

8.8. Control Registers

Table 8.5. Control Register 0

Field	Name	Default	Description
[31:30]	Reserved	0	Reserved
[29]	Wake Up	0	Control bit for transparent reconfiguration for the event of PROGRAMN pin toggle or REFRESH command execution.
[28]	NDR	0	Enable TransFR to latch and freeze I/O during reconfiguration.
[27]	Reserved	0	Reserved
[26]	No CDM	0	Control bits prevent download of the critical device control information from non-volatile memory to shadow registers in the event of refresh or PROGRAMN pin toggle.
[25]	TRANEdit	0	Enable transparent CRAM programming, if this bit is disabled, both EBR and IP Transparent Programming will be disabled.
[24]	TranEBR	0	Enable transparent EBR programming.
[23]	TranIP	0	Enable transparent Hard IP programming.
[22]	TranHSE	0	Enable security engine access in user function mode.
[21]	NO BOOT	0	Do not boot on REFRESH command execution or PROGRAMN pin pulsing.
[20]	SRME	0	This control bit is the Slow Response Mode Enable bit, which enables the automatic insertion of the Lattice specific protocol to handle the issue caused by the slow response of the non-volatile operation for high speed SSPI read. If this bit is set to *1*, an all zero byte (8'H00) will be transmitted right before the actual data as an indicator for data validation. If the data is not yet available, a dummy value (which is not equal to 8'H00) will be transmitted repeatedly until the data is available for transmitting. Upon becoming available, then the flag value of 8'H00 will be transmitted, followed by the actual data. It is up to the SPI master device to continue to toggle the SPI clock (SCK) and maintain the Slave Select (SS) low until that valid data is transmitted. In cases where the read data is immediately available, if this control bit is set, then no dummy bytes will be transmitted, but valid data will still be preceded by the flag byte (8'H00). If this control bit is clear (1'B0), then no dummy byte nor any flag byte will be transmitted. However, the data produced by reads which conflict with certain internal resources is also not guaranteed. Data produced by reading of device ID is always guaranteed, as it never conflicts with those slower internal resources.
[19]	SPIM	0	Control bit for Master SPI boot address selection.
[18:17]	P_DONE	0	PROGRAM_DONE overload control option for Bitstream. If bit 18 is set to *1*, the PROGRAM_DONE command is set to overload with either BYPASS or FLOW_THROUGH function, depending on bit 17. 10 - Overload with BYPASS

Field	Name	Default	Description
			11 - Overload with FLOW_THROUGH 0X - No Overload (Default)
[16:15]	INTN OPT	0	Over-ride INITN. If CR0[16] is set to 1, the INITN pin is overridden by CR0[15].
[14:13]	DONE OPT	0	Over-ride DONE. If CR0[14] is set to 1, the DONE pin is overridden by CR0[13].
[12]	Erase all	0	Enable Erase all operation
[11:10]	STX DUM	0	These two control bits determine the number of dummy bytes padding to add before the first frame is read out during incremental Bitstream read back (slave mode). 00 - 0 byte 01 - 4 bytes 10 - 8 bytes 11 - 16 bytes
[9:6]	Reserved	0	Reserved
[5:0]	Master Clock Frequency	0	Control bit that divides the clock coming from the on-chip oscillator. Value - 0x00 to 0x3F.

Table 8.6. Control Register 1

Field	Name	Default	Description
[31:30]	Reserved	2	Reserved
[29]	Reserved	0	Reserved
[28]	Reserved	0	Reserved
[27]	Reserved	0	Reserved
[26]	EBR Initialization enable	0	0 – Disables EBR initialization. 1 – Enables EBR initialization for power cycling, refresh, or PROGRAMN pin pulsing. Hardware automatically writes 0s to every EBR address.
[25:24]	HSE clock selection	1	Sets the HSE clock frequency with respect to the internal oscillator frequency. 0 – Divide by 5 1 – Divide by 5 2 – Divide by 5 3 – Divide by 6
[23]	CPHA	0	Selects the clock format. 0 – Sampling of data occurs at the rising edge of the clock. 1 – Sampling of data occurs at the falling edge of the clock.
[22]	CPOL	0	Selects an inverted or non-inverted clock. 0 – Active-high clock is selected. In idle state, clock is low. 1 – Active-low clock is selected. In idle state, clock is high.
[21]	TX Edge	0	Adjusts the master SPI Tx clock edge.
[20]	RX Edge	0	Adjusts the master SPI Rx clock edge.
[19]	Reserved	0	Reserved
[18]	Reserved	0	Reserved
[17]	32-bit SPIM address	0	Enables master SPI 32-bit addressing mode.
[16]	No Bulk Erase	0	0 – Automatic SRAM bulk erase for power cycling, refresh, or PROGRAMN pin pulsing. 1 – Disable automatic bulk erase.
[15]	SFDP_EN	0	Enables the reading of SFDP signature during post power-up test.
[14]	SFDP_COF (Continue on Failure)	0	Enables SFDP continue on failure operation.

Field	Name	Default	Description
[13:12]	Reserved	0	Reserved
[11:8]	Slave mode configuration timer count value	0	Slave SPI mode idle timer count value.
[7:4]	Master mode preamble/ signature detection timer count value	0	Master SPI mode bitstream preamble or signature detection timer count value.
[3:2]	Master mode preamble/ signature detection retry count value	0	Master SPI mode bitstream preamble detection retry count value. Number of times the preamble or signature detection is retried in master SPI mode (booting from external flash). 0 – Retry 1 time 1 – Retry 2 times 2 – Infinite retires 3 – No retry
[1:0]	Reserved	0	Reserved

8.9. Modifying the MachXO3D Feature Row

The feature row has a total of 128 bits divided as follows:

- 32-bit NV CR1
- 64-bit FEATURE
- 32-bit FEABITS

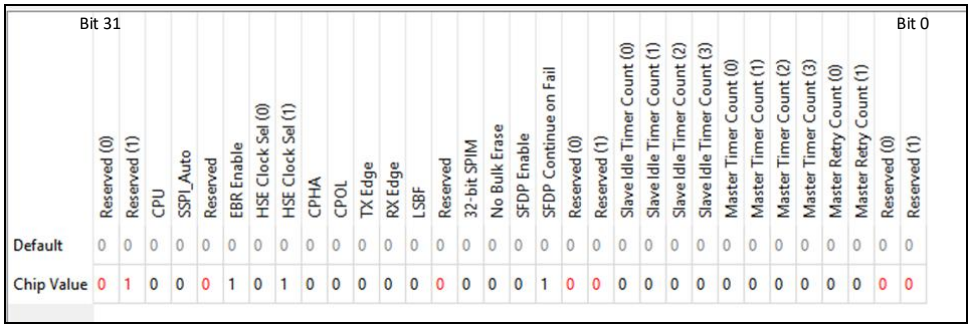


Figure 8.9. Feature Row – Control Register 1 (NV CR1) Bits

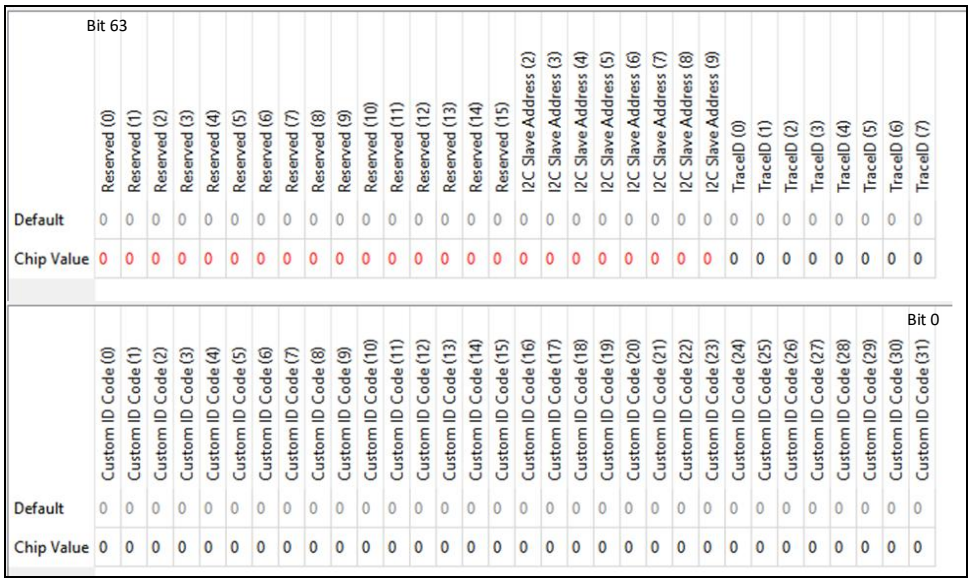


Figure 8.10. Feature Row – FEATURE Bits (Custom ID/TraceID/I2C Slave Address)

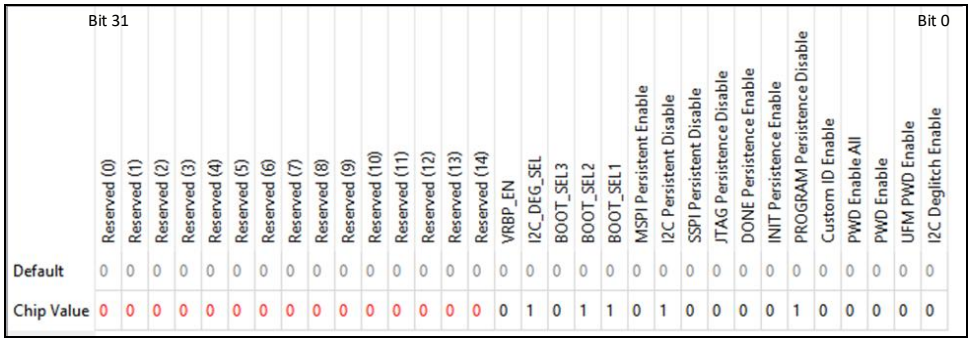


Figure 8.11. Feature Row – FEABITS

8.9.1. Modifying the Feature Row Programming File

You can access the Feature Row Editor through the Programming File Utility to enable or disable silicon features in Lattice FPGA devices by modifying feature row fuse settings in the data file (.fea). Before modifying the feature row, you must first create a project in the Lattice Diamond software and run the *Export Files* process. The .fea file is generated inside the *Implementation* folder.

To modify the feature row, perform the following steps:

1. In the Diamond software, select **Tools > Programmer > Design > Utilities > Programming File Utility**. The Programming File Utility window appears.
2. In the Programming File Utility, select **Tools > Feature Row Editor > Browse .fea file > Read .fea file**.
3. Double click the chip value associated with the feature row bit to modify to toggle the value from 1 to 0, or from 0 to 1.
4. Click **Save** to overwrite the existing data file (.fea). Otherwise, click **Save As** to save the data file (.fea) with a different name.

8.9.2. Programming the Feature Row Using the .fea Programming File

To program the feature row using the .fea file, perform the following steps:

1. Run the Diamond Programmer.
2. Select **Design > JTAG Scan**.

3. Once scanning of the device is done, select **Edit > Device Properties**. The Device Properties window appears.
4. Change the device properties to the following settings:
   - Access Mode: Feature Row Programming
   - Port Interface: JTAG Interface (or any available interface such as Slave SPI or I2C)
   - Operation: Program Feature Row

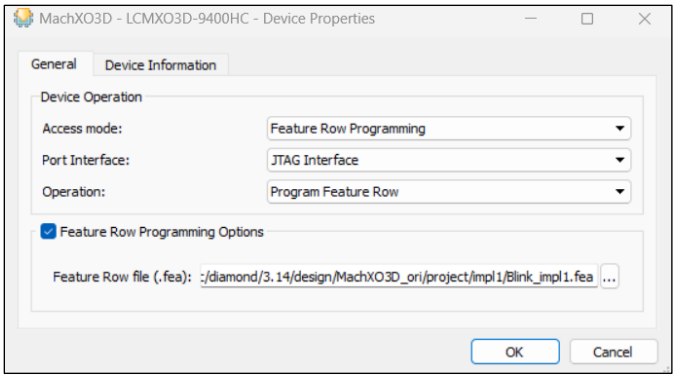


Figure 8.12. Device Properties – Program Feature Row

5. Under **Feature Row Programming Options**, select the .fea file from the *Implementation* folder.
6. Click **OK** to close the Device Properties window.
7. Select **Design > Program**.

8.9.3. Modifying the Feature Row without Using the .fea Programming File

Feature row bits can be modified without having to open the .fea file through the Programming File Utility.

To modify the feature row, perform the following steps:

1. In the Diamond Programmer, select the device whose feature row you want to modify.
2. Select **Edit > Device Properties**. The Device Properties window appears.
3. Change the device properties to the following settings:
   - Access Mode: Feature Rows Programming
   - Port Interface: JTAG Interface (or any available interface such as Slave SPI or I2C)
   - Operation: Update Feature Row

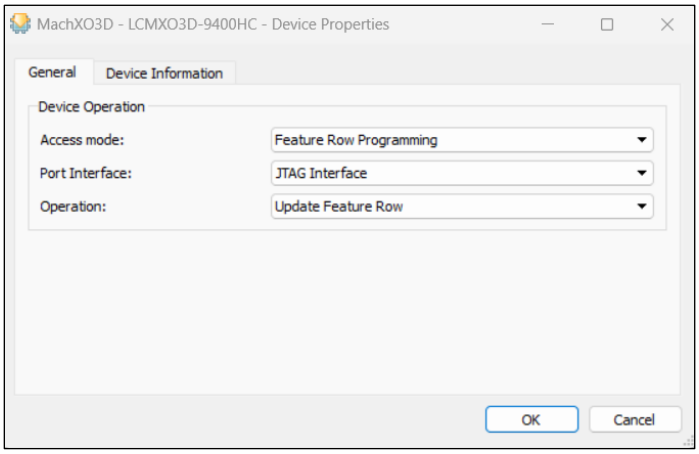


Figure 8.13. Device Properties – Update Feature Row

4. Click **OK** to close the Device Properties window.
5. Select **Design > Program**. The Feature Row window appears.
6. Double click the chip value associated with the feature row bit to modify to toggle the value from 0 to 1.

Note: Update Feature Row only allows individual bits to be modified from 0 to 1. In addition, only modifications to values displayed in black are applied.

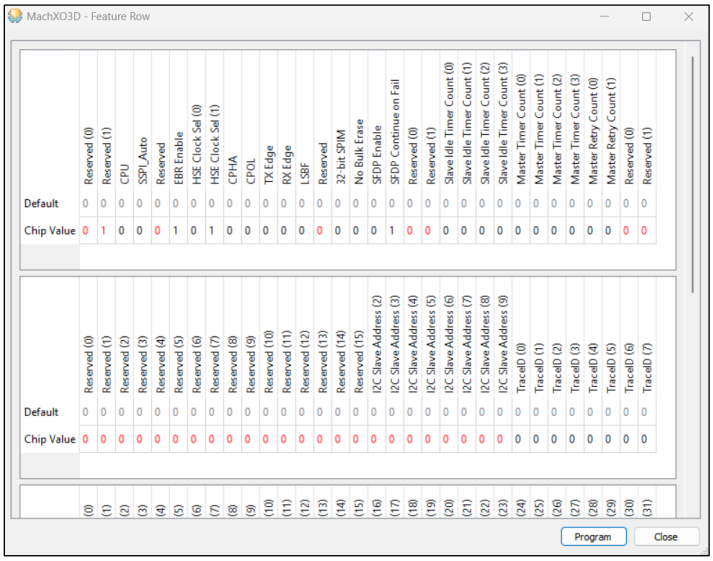


Figure 8.14. Modifying Feature Row Bits

7. Click **Program**. A dialog box appears notifying you that feature row bit or bits will be overwritten.
8. Click **Yes**. The device feature row is programmed.

8.9.4. Programming the No CDM Bit in CR0

Shadow register programming is volatile. Settings are cleared after a power cycle or REFRESH event. Refer to the Control Registers section for information on the No CDM bit. To avoid clearing the shadow register and retain the shadow register settings after a REFRESH event, perform the following to set the No CDM bit in CR0 to 1 before triggering the REFRESH event:

1. In the Device Properties window, change the device properties to the following settings:
   - Access Mode: Static Random Cell Mode
   - Port Interface: JTAG Interface (or any available interface such as Slave SPI or I2C)
   - Operation: SRAM Program Control Register 0

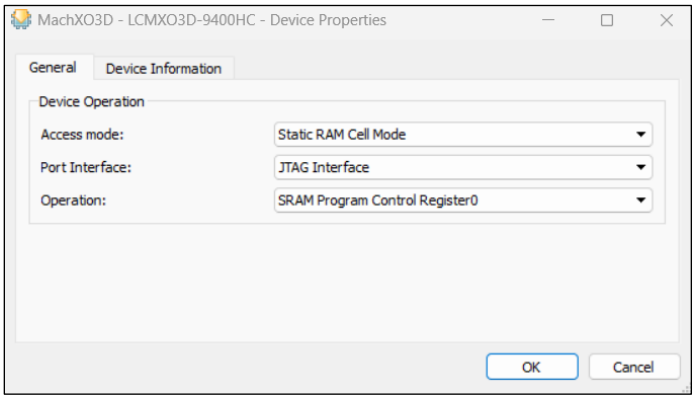


Figure 8.15. Device Properties – SRAM Program Control Register0

2. Double click the chip value associated with *No CDM* to toggle the value from 0 to 1.

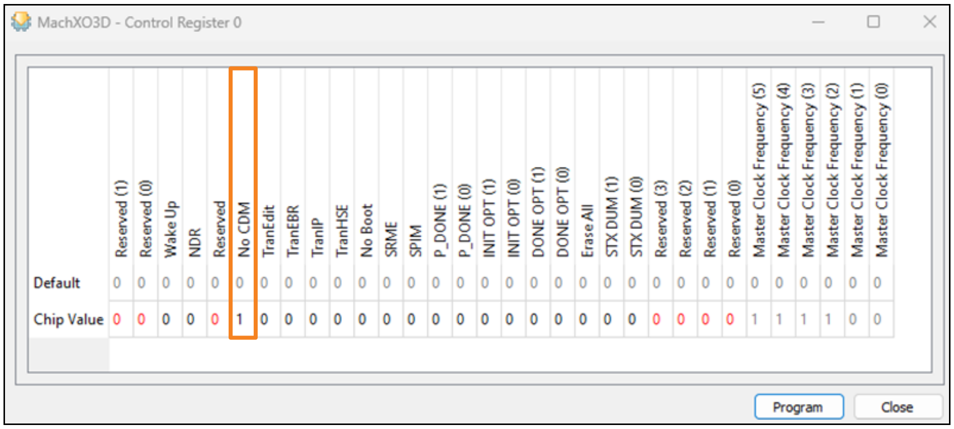


Figure 8.16. Modifying No CDM Bit

3. Click **Program**.

8.9.5. Feature Row Programming File (.fea)

Figure 8.17 maps bits in the .fea file (generated by the Lattice Diamond software) to the feature row bits read out through the Diamond Programmer.

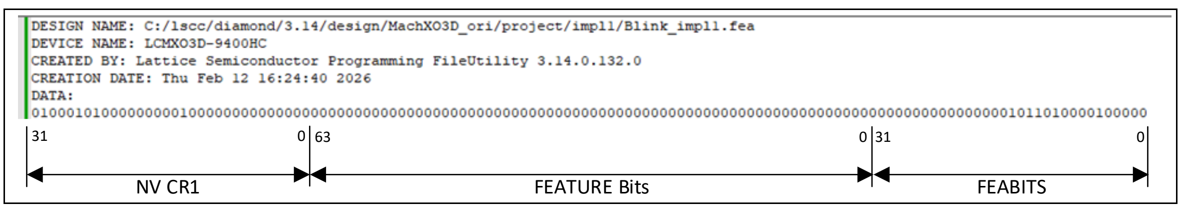


Figure 8.17. Sample .fea File

For information on the individual bits:

- 32-bit NV CR1 – refer to the Control Registers section.
- 64-bit FEATURE – refer to the following table.

Table 8.7. 64-bit FEATURE Bits

Field	Name	Default	Description
[63:48]	Reserved	16’b0	Reserved
[47:40]	I2C Slave Address[7:0]	8’b0	I2C slave address
[39:32]	Trace ID[7:0]	8’b0	User programmable MSB of the trace ID
[31:0]	Custom ID code[31:0]	32’b0	Customer ID code

- 32-bit FEABITS – refer to Using Hardened Control Functions in MachXO3D Devices Reference Guide (FPGA-TN-02119).

8.10. JTAG Daisy Chain Programming Modes

In the Lattice Diamond Programmer software, you can change the JTAG daisy chain programming mode by selecting **Edit > Settings**, then clicking the **Programming** tab in the **Settings** window. Figure 8.18 shows the two available programming modes: sequential mode and turbo mode.

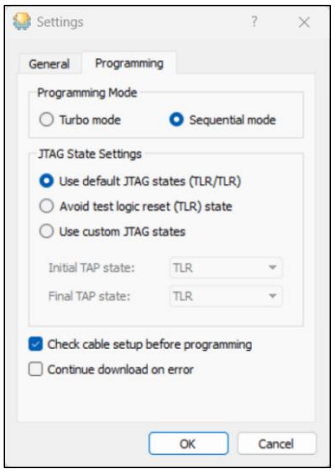


Figure 8.18. Selecting JTAG Daisy Chain Programming Mode

8.10.1. Sequential Mode

Sequential mode programming refers to the process of programming devices in a JTAG daisy-chain configuration. Each device is programmed sequentially according to its order in the chain. In sequential mode programming, the complete bitstream, including the internal DONE bit, is loaded into each device. If programming is interrupted or fails, device behavior depends on the failure condition. Some devices might boot from their primary image while others might revert to the golden image.

During direct programming in sequential mode, each device exits programming and immediately begins its boot process after the bitstream is successfully loaded, regardless of the programming status of other devices in the chain.

Note: Direct programming refers to programming the device while the device is not in User Mode.

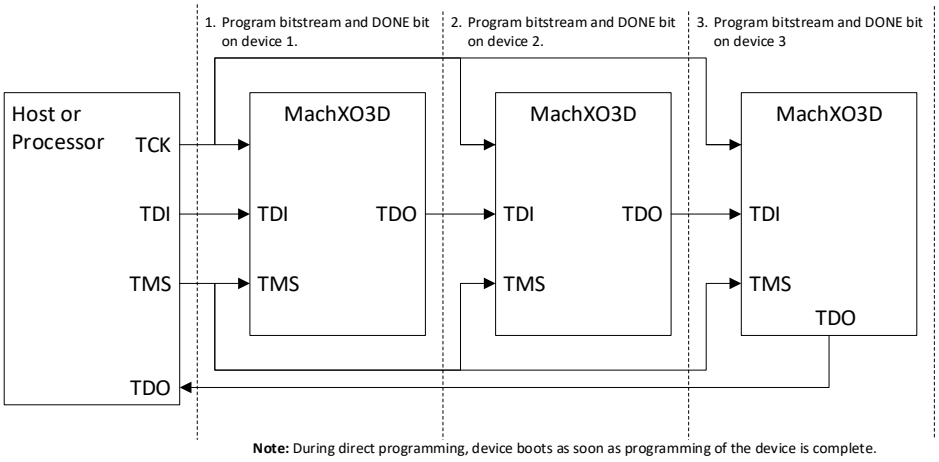


Figure 8.19. Sequential Mode Programming

8.10.2. Turbo Mode Programming

Turbo mode programming also operates within a JTAG daisy-chain configuration but with one key distinction. While the bitstream for each device is programmed sequentially, the internal DONE bits of the devices are programmed in parallel after the final device in the chain has been programmed. This ensures that booting is synchronized across all

devices. If programming is interrupted or fails, all devices will boot from their respective golden images, thereby providing a predictable fallback behavior.

During direct programming in turbo mode, all devices receive their DONE bits in parallel and exit programming mode simultaneously. As a result, all devices initiate boot simultaneously.

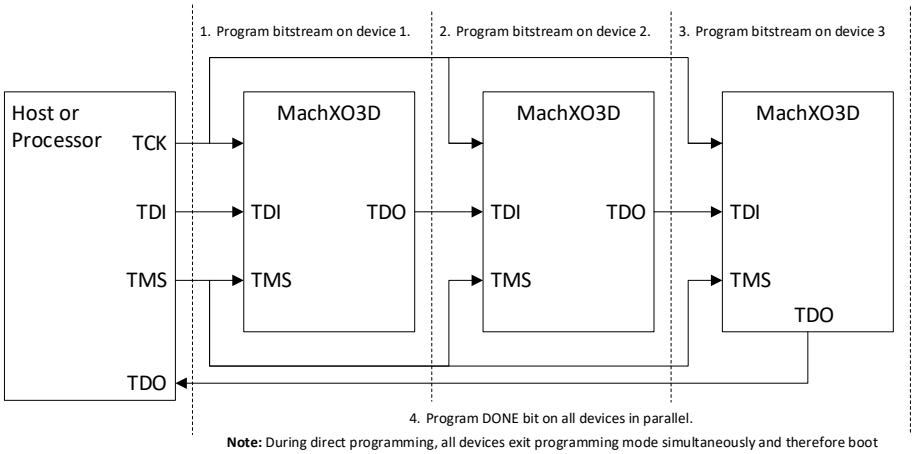


Figure 8.20. Turbo Mode Programming

Reference

- MachXO3D Family Data Sheet (FPGA-DS-02026)
- MachXO3D Embedded Security Block (FPGA-TN-02091)
- Using Hardened Control Functions in MachXO3D (FPGA-TN-02117)
- Using Hardened Control Functions in MachXO3D Devices Reference Guide (FPGA-TN-02119)
- MachXO3D Soft Error Detection (SED)/Correction (SEC) Usage Guide (FPGA-TN-02124)
- MachXO3 Using Password Security technical note (FPGA-TN-02072)
- Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02198)
- Using TraceID (FPGA-TN-02084)
- Lattice Diamond FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 2.0, March 2026

Section	Change Summary
All	Made minor editorial changes.
Acronyms in This Document	Added *FSE*, *GPIO*, *HSE*, *SFDP*, and *UDS*.

Configuration Process and Flow	In the Master and Slave SPI Configuration Port Pins section:
	• In Table 4.8. Master SPI Configuration Port Pins:
	  • Updated description for MCLK to indicate requirement for a pull-up resistor instead of a pull-down resistor.
	  • Added statement on recommended pull-up resistor value to description for CSSPIN.
	• Under the MCLK/CCLK section, updated description to indicate requirement for a pull-up resistor instead of a pull-down resistor.
	• Under the SN section, added SN condition for SDM mode.
	• Under the CSSPIN section:
	  • Updated CSSPIN description to indicate weak pull up
	  • Updated recommended pull-up resistor value for CSSPIN to only 10 kΩ.

Configuration Modes	• In the SDM Mode section:
	  • Added note on recommended setting for SLAVE_SPI_PORT option in SDM.
	• In the I2C Configuration Mode section:
	  • Added note on ensuring that I2C target devices sharing the same bus do not interrupt device flash programming over the I2C port.
	• Added the JTAG Daisy Chain section.

Software Selectable Options	In the Slave SPI Port section: • Updated description of the SLAVE_SPI_PORT option.

Advanced Configuration Information	• In Table 8.2. MachXO3D sysCONFIG Programming Commands:
	  • Updated operand and notes for ISC_ENABLE_X command.
	  • Updated notes for LSC_READ_STATUS0 command.
	  • Added LSC_READ_STATUS1 command.
	  • Updated the LSC_PROG_FEATURE command write data to *YY*16*.
	  • Updated the LSC_READ_FEATURE command read data to *YY*16*.
	  • Updated cross reference to section in notes for ISC_PROGRAM_SECURITY command.
	  • For the LSC_PROG_CTRL0 command:
	  • Fixed typographical error in the command name *Program Control Register 0*.
	  • Updated write data to *YY*4*.
	  • Updated notes.
	  • Added LSC_PROG_CTRL1 command.
	  • For the LSC_READ_CTRL0 command:
	  • Updated command opcode to 0x20.
	  • Updated read data to *YY*4*.
	  • Updated notes.
	  • Added LSC_READ_CTRL1 command.
	  • Removed note on Control Register 0 bits.
	• Added the Status Registers, Control Registers, Modifying the MachXO3D Feature Row, and JTAG Daisy Chain Programming Modes sections.
	• In the Control Registers section:
	  • Moved the Control Register 0 table into section.
	  • In Table 8.5. Control Register 0:
	  • Corrected description for INTN OPT.

Section	Change Summary
References	Added Using Hardened Control Functions in MachXO3D Devices Reference Guide.

Revision 1.9, April 2025

Section	Change Summary
Acronyms in This Document	Added *Rx* and *FIFO*.
Configuration Process and Flow	Updated link to MachXO3D Embedded Security Block (FPGA-TN-02091) in the Offline Programming section.
Configuration Modes	• Added description of Rx FIFO when device first boots up in the I2C Configuration Mode section. • Updated Figure 5.10. Example Process Flow.
Security Options	• Added note on ENABLE_TRANSFR in the BACKGROUND_RECONFIG section. • Updated description and added note on bitstream reconfiguration in the BACKGROUND_RECONFIG_SECURITY section.
Advanced Configuration Information	Added LSC_DEVICE_CTRL command in Table 8.2. MachXO3D sysCONFIG Programming Commands.
References	• Updated link to MachXO3D Embedded Security Block (FPGA-TN-02091). • Updated document numbers and added links for Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02198) and Using TraceID (FPGA-TN-02084).

Revision 1.8, November 2024

Section	Change Summary
All	Made editorial fixes.
Configuration Process and Flow	• Updated the Table 4.1. Memory Space Accessibility of Different Ports and table description. • Updated the Table 4.3. MachXO3D Feature Row Elements.
Configuration Modes	• Updated the section Password. • Added section 5.3.3 Status Register Readout during Boot Failure.
Advanced Configuration Information	• Updated the Table 8.2. MachXO3D sysCONFIG Programming Commands. • Added Table 8.3. Control Register 0.
References	Updated the reference list.

Revision 1.7, July 2024

Section	Change Summary
All	• Made editorial fixes
Configuration Process Overflow	• Added section Configuration/Boot-up Time • Updated Figure 4.1. Configuration Flow and Figure 4.2. Configuration from Power-On-Reset Timing.

Revision 1.6, April 2024

Section	Change Summary
Configuration Process and Flow	Minor editorial fixes.
Software Selectable Options	Added paragraph in the ROLLBACK_CONTROL section regarding USERCODE value when the version rollback prevention feature is enabled.

Revision 1.5, March 2024

Section	Change Summary
Software Selectable Options	Table 6.1. Configuration Mode/Port Options in the Configuration Mode and Port Options section: • Added I2C_GLITCH_FILTER and I2C_GLITCH_FILTER_RANGE options. • Numbered existing note to table. • Added Note 2 and Note 3 to table.

Revision 1.4, February 2024

Section	Change Summary
All	• Minor editorial fixes. • Changed the title to *MachXO3D Programming and Configuration User Guide*.
Disclaimers	Updated this section.
Advanced Configuration Information	• Updated the I2C data format in Figure 8.8. Flash Page Command and Data Sequence. • Updated the last paragraph of Section 8.6 Reading Flash Pages. • Changed from *1 page undefined data* to *4 dummy bytes*.
References	Updated this section.
Technical Support Assistance	Added the Lattice Answer Database link.

Revision 1.3, August 2021

Section	Change Summary
Configuration Process and Flow	• Updated the description for the MCLK/CCLK pin in Table 4.7. Master SPI Configuration Port Pins. • Updated the second bullet point of the MCLK/CCLK section.
Configuration Modes	Updated content, including the notes for Table 5.2. Master SPI Port, in the Master SPI (MSPI) Configuration Mode section to change the description for the MCLK/CCLK pin from pull-up resistor to pull-down-resistor.

Revision 1.2, April 2021

Section	Change Summary
Configuration Modes	Updated the description of dual boot configuration and generating configuration data in the Golden Image Dual Configuration section.
Software Selectable Options	Updated the Read Flash [LSC_READ_INCR_NV] and the Read UFM [LSC_READ_UFM] notes in Table 8.2. MachXO3D sysCONFIG Programming Commands.

Revision 1.1, February 2021

Section	Change Summary
Configuration Process and Flow	Updated content, including Table 4.7, in Master and Slave SPI Configuration Port Pins section to change 1K pull-up resistor from Recommended to *Required*.
Software Selectable Options	Updated Table 6.1 to add DONE and INITN settings.

Revision 1.0, September 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com