MachXO4 Programming and Configuration User Guide

Technical Note

FPGA-TN-02393-1.2

April 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.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ACK	Acknowledge
ASSP	Application-Specific Standard Part
BSE	Bitstream Engine
CDM	Control Information Download Mode
CFG	Configuration
CRAM	Configuration Random-Access Memory (also known as configuration SRAM or SRAM configuration memory in this document)
CRC	Cyclic Redundancy Check
DR	Data Register (Shift-DR)
EBR	Embedded Block RAM
EFB	Embedded Function Block
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GOE	Global Output Enable
GPIO	General Purpose I/O
GSR	Global Set Reset
GWDIS	Global Write Disable
I2C	Inter-Integrated Circuit
ID	Identification
IR	Instruction Register (Shift-IR)
ISC	In System Configuration
JTAG	Joint Test Action Group
LSB	Least Significant Bit
LUT	Look Up Table
MSB	Most Significant Bit
MSPI or SPIm	Controller Serial Peripheral Interface
OTP	One-Time Programmable
PCB	Printed Circuit Board
PLD	Programmable Logic Device
PLL	Phase-Locked Loop
POR	Power On Reset
RAM	Random-Access Memory
Rx	Receiver
SDM	Self-Download Mode
SEC	Soft Error Correction
SED	Soft Error Detection
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSPI	Target Serial Peripheral Interface
STAPL	Standard Test and Programming Language
SVF	Serial Vector Format
TAP	Test Access Port
TCK	Test Clock
TDI	Test Data Input
TDO	Test Data Output

Abbreviation	Definition
TMS	Test Mode Select
UFM	User Flash Memory

1. Introduction

MachXO4™ devices are SRAM-based programmable logic devices (PLDs) that include internal flash memory. This grants MachXO4 devices infinite configurability and non-volatile memory capabilities. MachXO4 devices also have flexible and robust access control for the configuration ports to enable various programming and update needs. MachXO4 devices provide a rich set of features for configuring and programming the field programmable gate array (FPGA). The many options available provide the flexibility to build a programming or configuration solution that suits a particular set of needs. This document describes the available options.

MachXO4 devices contain two types of memory: static random-access memory (SRAM) and internal flash memory. SRAM contains the configuration data that defines the behavior of the FPGA. The configuration data, in most cases, is retrieved from a non-volatile memory. The non-volatile memory holds the configuration data that is loaded into the FPGA device's SRAM. MachXO4 devices provide the non-volatile memory (internal flash memory) that can store the configuration data.

2. Features

The following are main programming and configuration features of MachXO4 devices:

- Instant-on configuration from internal flash memory – powers up in milliseconds
- Infinite number of configuration cycles through volatile SRAM technology
- Up to 10000 programming cycles with internal flash memory
- Single-chip, secure solution
- Self-download mode (SDM)
- Multiple programming and configuration interfaces:
  - JTAG (IEEE 1149.1)
  - Controller serial peripheral interface (SPI)
  - Target SPI
  - Inter-integrated circuit (I2C)
  - WISHBONE
- User flash memory (UFM) for non-volatile data storage:
  - Configuration flash memory overflow
  - Embedded block RAM (EBR) initialization data
  - Application-specific data
- Transparent programming of non-volatile memory
- Dual-boot with internal memory and external SPI memory
- Bitstream compression
- TransFR capability – leave-alone I/O (non-JTAG mode)
- Soft error detection (SED) support

3. Definition of Terms

Table 3.1 lists the terms used in this document to describe common functions, features, or concepts.

Table 3.1. Definition of Terms

Term	Definition
Background Reconfiguration	The process in which selected on-chip memory resources are reconfigured while leaving the FPGA device in user mode. It is typically associated with SRAM reconfiguration to support soft error detection and correction.
BIT File	The BIT file is the configuration data for the FPGA device. It is a binary file and is programmed unmodified into the SPI flash by the Lattice Radiant™ Programmer.
Configuration	A change in the state of the SRAM memory cells.
Configuration Data	The data read from the non-volatile memory and loaded into the FPGA device's SRAM configuration memory. This is also referred to as a bitstream or device bitstream.
Configuration Mode	The method the FPGA uses to acquire the configuration data from the non-volatile memory.
HEX File	The HEX file is the configuration data for the FPGA device in the hexadecimal format.
Internal flash memory	On-die, non-volatile flash-type memory. JEDEC file or BIT file can be programmed directly into the internal flash sector. User does not need to know where an actual page of the configuration data starts. The MachXO4 configuration engine handles the parsing in the flash to SRAM transfer. There are multiple flash sectors for FPGA configuration image storage, general-purpose user flash, and device feature definitions.
JEDEC	The JEDEC file contains the configuration data programmed into the internal flash.
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. **Note:** When specifying binary numbers in relation to register bit settings, the suffix b may be excluded. Binary numbers may also be represented in the Verilog format.
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 FPGA device does not perform the logic operations until offline mode programming/configuration is complete.
Port	The physical connection used to perform programming and some configuration operations. Ports on MachXO4 devices include JTAG, controller SPI, target SPI, and I2C.
Programming	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 Programming Mode	This mode is used to access the configuration flash and user flash memory while leaving the FPGA device in user mode.
User Mode	The FPGA device is in user mode when configuration is complete and the device is performing the logic functions it has been programmed to perform.

4. Configuration Details

The MachXO4 device SRAM configuration memory contains the configuration data that defines the functional behavior of the FPGA in user mode. Configuration data that is loaded into the SRAM configuration memory is either retrieved from an external or internal non-volatile memory or transmitted to the device through a configuration port.

4.1. Bitstream and Internal Flash Sizes

MachXO4 devices are SRAM-based FPGAs. The SRAM configuration memory must be loaded from an external or internal non-volatile memory that can store all the configuration data. The size of the configuration data varies. It is dependent on the amount of logic resources available in the FPGA and the number of pre-initialized embedded block RAM (EBR) components. A design using the largest MachXO4 device, with every EBR pre-initialized with unique data values and generated without compression enabled, requires the largest amount of storage.

Table 4.1. Bitstream and Internal Flash Sizes

Device[1]	Maximum Internal Flash (Mb)	Scenario	Uncompressed Bitstream Size (Mb)[2]
LFMXO4-010	0.33	No EBR	0.35
		Maximum EBR	0.41
LFMXO4-015	0.33	No EBR	0.35
		Maximum EBR	0.41
LFMXO4-015 (256 Ball Package)	0.47	No EBR	0.51
		Maximum EBR	0.58
LFMXO4-025	0.47	No EBR	0.51
		Maximum EBR	0.58
LFMXO4-050	0.80	No EBR	0.93
		Maximum EBR	1.02
LFMXO4-050 (400 Ball Package)	1.38	No EBR	1.47
		Maximum EBR	1.70
LFMXO4-080	1.38	No EBR	1.47
		Maximum EBR	1.70
LFMX04-110	2.0	No EBR	2.11
		Maximum EBR	2.56

Notes:

1. MachXO4 devices have on-chip (internal) non-volatile memory. If the on-chip non-volatile memory is insufficient, use an external non-volatile memory.
2. Both unencrypted and encrypted bitstreams are the same size. Bitstream compression ratio varies depending on the bitstream so only uncompressed bitstream sizes are shown.

There are special considerations when storing configuration data in the internal flash memory. The internal flash memory provides three types of independent sectors. The first sector type is the configuration flash dedicated to storing compressed configuration data. The MachXO4 (LFMXO4) device has one sector of this type. Each sector can store a complete bitstream. The second sector type is the user flash memory (UFM) which has three possible functions: additional configuration flash storage for large configuration data images, storage for EBR contents, or general-purpose flash memory. The MachXO4 (LFMXO4) device has one sector of this type. The third sector type is the feature row.

Figure 4.1 shows the flash memory space for the MachXO4 (LFMXO4) device.

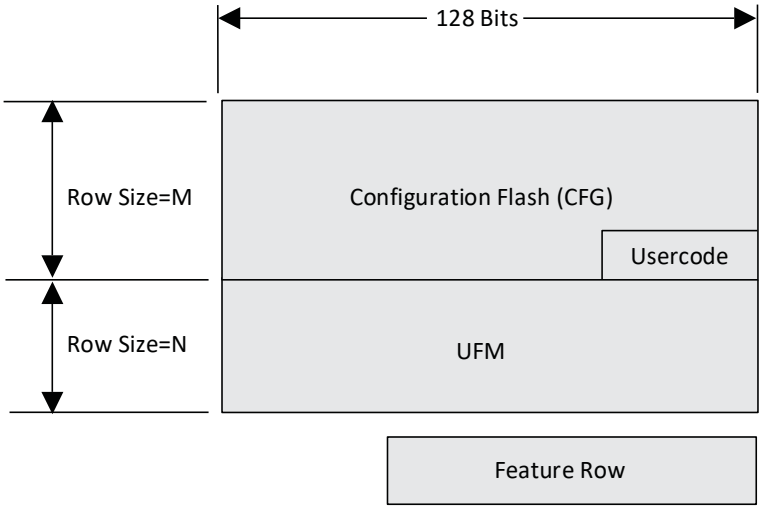


Figure 4.1. Flash Memory Space for MachXO4 (LFMXO4) Device

The configuration flash can typically store the compressed configuration data of most designs. However, increased logic utilization and EBR pre-initialization might cause configuration data overflow into the UFM sector. Overflow into the UFM sector can be allowed or prevented through software when generating the configuration data. If the configuration data is too large for the internal flash memory, either modify the design to reduce the configuration data size or use an external SPI storage device.

4.2. Programming and Configuration Ports

Table 4.2 lists the ports supported by MachXO4 devices for programming and configuration, which include the industry standard JTAG interface. Each port provides a method to access the MachXO4 device memory resources.

Table 4.2. MachXO4 Device Programming and Configuration Ports

Interface	Port	Description
JTAG	JTAG (IEEE 1149.1 and IEEE 1532)	4-wire or 5-wire JTAG interface
sysCONFIG™	Controller SPI	Controller serial peripheral interface
	Target SPI	Target serial peripheral interface
	I2C	Inter-integrated circuit (I2C) interface
Internal	WISHBONE	WISHBONE bus interface

4.3. Memory Space Accessibility

MachXO4 devices contain two types of memory: SRAM and internal flash memory. The MachXO4 device ports have different access levels to each memory space. Table 4.3 lists the different ports and their memory space accessibility.

Table 4.3. Memory Space Accessibility of Different Ports

Port	Internal Flash		SRAM	
	Read	Write	Read	Write
JTAG	Yes	Yes	Yes	Yes
SPI	Yes	Yes	Yes	Yes
I2C	Yes	Yes	Yes	Yes
WISHBONE	Yes[1]	Yes[1]	No	No

Note:

1. Only applicable in transparent programming mode.

4.4. sysCONFIG Pins

MachXO4 devices provide a set of I/O pins that are used to program and configure the FPGA. These pins are grouped together to create ports (such as JTAG, target SPI, controller SPI, and I2C). These ports are used to interact with the FPGA for programming and configuration, and to access resources within the FPGA. sysCONFIG pins in a configuration port group can be active and used for programming the FPGA or they can be reconfigured to act as general purpose I/O pins.

Notes:

- In this document, I/O pins used for programming and configuration are generally referred to as sysCONFIG pins.
- Unless otherwise specified, sysCONFIG pins are powered by the V_{CCIO0} and V_{CCIO2} voltages. This is an important consideration when provisioning other logic attached to I/O banks 0 and 2.

The following are guidelines when repurposing configuration port pins for use as general purpose I/O pins:

- Disable the unused configuration port through the Lattice Radiant Device Constraint Editor in the Global tab. Each configuration port is listed in the sysCONFIG options tree.
- External logic must be prevented from interfering with device programming. Make sure repurposed sysCONFIG pins are not asserted when the MachXO4 device is in the feature row hardware default mode state. For example, driving PROGRAMN with an active low signal after the MachXO4 device is in the feature row hardware default mode state. Failure to reprogram the feature row with PROGRAMN disabled prevents the FPGA from configuring and entering user mode.
- Use JTAGENB with care when selectively enabling or disabling the JTAG port. Any external logic connected to the JTAG pins must not contend with the JTAG programming port.

4.5. sysCONFIG Pin List

Table 4.4 lists the sysCONFIG pins of the device and the default states of these pins. In the hardware default mode state, the target SPI, I2C, and JTAG ports are enabled. In user mode, the MachXO4 (LFMXO4) device retains only the JTAG port as dictated by the port default setting. To retain use of a port in user mode, you must enable the port through the Lattice Radiant Device Constraint Editor in the Global tab.

Table 4.4. Default State of sysCONFIG Pins

Port	Port Default Setting in Radiant[1]	sysCONFIG Pins			
	LFMXO4	Name	Hardware Default Mode State[2]		User Mode Default
			Pin Function	Pin Direction	LFMXO4
SDM	Disable	PROGRAMN	PROGRAMN	Input with weak pull-up	GPIO
		INITN	GPIO	Input/Output with weak pull-up	GPIO
		DONE	GPIO	Input/Output with weak pull-up	GPIO
Controller SPI	Disable	MCLK/CCLK[3]	CCLK	Input with weak pull-up	GPIO
		CSSPIN	GPIO	Input/Output with weak pull-up	GPIO
		SI/SISPI/D0[3, 4, 5]	SI	Input	GPIO
		SO/SPISO/D1[3, 4, 5]	SO	Output	GPIO
		SCL/D2[3, 5]	SCL	Bi-directional	GPIO
		SDA/D3[3, 5]	SDA	Bi-directional	GPIO
Target SPI	Disable	MCLK/CCLK[3]	CCLK	Input with weak pull-up	GPIO
		SN	SN	Input with weak pull-up	GPIO
		SI/SISPI/D0[3]	SI	Input	GPIO
		SO/SPISO/D1[3]	SO	Output	GPIO
I2C	Disable	SCL/D2[3]	SCL	Bi-directional	GPIO
		SDA/D3[3]	SDA	Bi-directional	GPIO
JTAG	Enable	TCK	TCK	Input	TCK
		TMS	TMS	Input with weak pull-up	TMS
		TDI	TDI	Input with weak pull-up	TDI
		TDO	TDO	Output with weak pull-up	TDO
		JTAGENB	GPIO	Input/Output with weak pull-down	GPIO

Notes:

1. The port default setting can be modified through the Lattice Radiant Device Constraint Editor in the Global tab.
2. When the feature row is erased, feature row settings default to the hardware default mode state settings. The hardware default mode state is also referred to as the feature row erased mode state or feature row hardware default mode state.
3. Shared I/O pin between multiple ports.
4. SI/SISPI/DO and SO/SPISO/D1 are used for controller SPI dual read mode.
5. SI/SISPI/DO, SO/SPISO/D1, SCL/D2, and SDA/D3 are used for controller SPI quad read mode.

4.6. Self-Download Mode Pins

Self-download mode pins are dual-purpose I/O pins. In user mode, these pins are general purpose I/O pins by default. These pins are powered by the V_{CCIO0} voltage.

4.6.1. PROGRAMN

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 the initialization phase of the device configuration process. Holding the PROGRAMN pin low prevents the MachXO4 device from booting from internal flash or external SPI flash. PROGRAMN must be asserted low for a minimum period of t_{PRGRMJ} for it to be recognized by the FPGA. This minimum time is defined in the sysCONFIG Port Timing Specifications of the MachXO4 Family Data Sheet (FPGA-DS-02125).

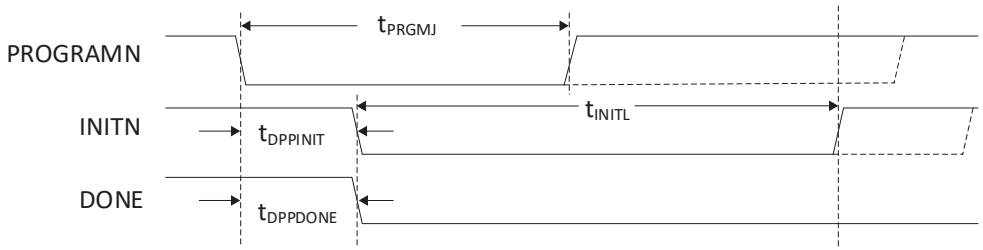


Figure 4.2. Configuration from PROGRAMN Timing

The following are conditions to be aware of in relation to the PROGRAMN pin:

- If the device is currently being programmed using JTAG, 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 process.
- Driving PROGRAMN high on a device in feature row hardware default mode state disables the target SPI and I2C ports. Start target SPI or I2C programming operations after PROGRAMN is driven low.
- PROGRAMN is active during power-up, even when PROGRAMN has been reserved as a general purpose I/O. Do not allow any input signal attached to PROGRAMN to transition from high to low at a frequency greater than the V_{CC} (min) to INITN rising edge period. Such high to low PROGRAMN assertions might prevent the MachXO4 device from configuring causing the FPGA to remain in a continuous reset condition.

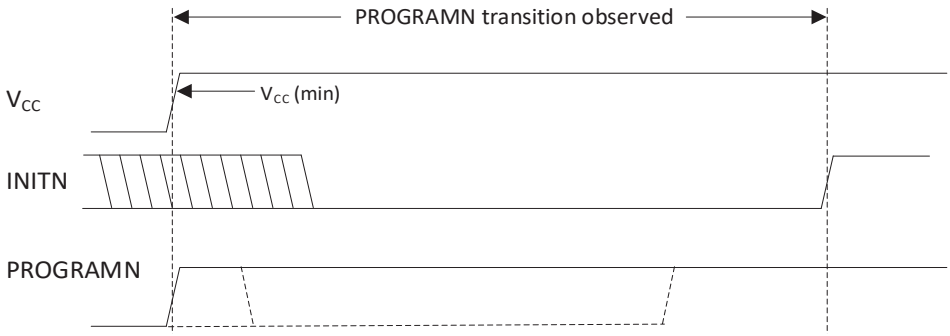


Figure 4.3. Period to Avoid PROGRAMN Transitions

4.6.2. INITN

The INITN pin is a bidirectional open-drain control pin. The following conditions cause INITN to toggle low:

- Power is applied (power up).
- PROGRAMN pin is pulsed (falling-edge transition has occurred).
- REFRESH command is received through a configuration port (JTAG, target SPI, or I2C).

INITN toggles low to indicate that the initialization phase is in progress. After the t_{INITL} period has elapsed, the INITN pin is de-asserted (toggles high) to indicate that the device is ready to accept configuration data. The device begins loading configuration data from either the internal flash or external SPI storage device.

INITN can be asserted low by an external agent before the t_{INITL} period has elapsed (in effect holding INITN low) to prevent the FPGA from reading configuration data. This is useful when there are multiple programmable devices chained together. The programmable device with the longest t_{INITL} period can hold all other devices in the chain from accepting configuration data until it is ready itself.

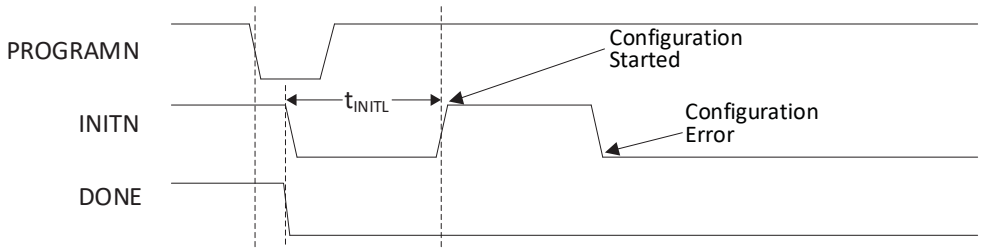


Figure 4.4. Configuration Error Notification

Once the t_{INITL} period has elapsed and INITN is de-asserted, INITN functions as an error signal. Any subsequent assertion of INITN indicates that the device has detected an error during configuration. The following are conditions that can cause device configuration to fail:

- Bitstream cyclic redundancy check (CRC) error is detected.
- Invalid command error is detected.
- A time out error is encountered when loading from the flash.
- The program done command is not received at the end of on-chip SRAM configuration or when the end of the internal flash is reached.

When an error is detected during device configuration as indicated by INITN, the internal DONE bit is not set and the DONE pin remains low so the device does not wake up. The error can be cleared by correcting the configuration bitstream and forcing the FPGA back into the initialization phase.

By default, the INITN pin is a general purpose I/O pin with weak pull-up (INITN is disabled). You can use the SDM_PORT option in the Lattice Radiant Device Constraint Editor to enable the INITN pin. The INITN behavior shown in Figure 4.4 is observable only when the INITN pin is enabled.

In feature row hardware default mode state, the INITN pin is disabled by default (functions as a general purpose I/O pin).

4.6.3. DONE

The DONE pin is a bi-directional open-drain pin with weak pull-up. The DONE pin is de-asserted low in tandem with the INITN pin when the FPGA enters the initialization phase. After an internal DONE bit is set, the DONE signal is used to indicate whether the FPGA is in user mode. Setting the internal DONE bit marks the beginning of the FPGA wake-up phase. The DONE pin is released high when the FPGA enters user mode.

The FPGA can be prevented from entering user mode indefinitely by having an external agent keep the DONE pin de-asserted low. An FPGA is ready to start operation only after DONE toggles high. A common reason for keeping DONE held low is to allow multiple FPGAs to finish configuration so that operation can start in unison only after configuration of the last FPGA. To use DONE to stall entering user mode, the DONE pin must be enabled through the SDM_PORT option in the Lattice Radiant Device Constraint Editor and the feature row must be programmed.

By default, the DONE pin is a general purpose I/O pin with weak pull-up (DONE pin is disabled). Consequently, after the internal DONE bit is set, the device immediately enters the FPGA wake-up phase. You can use the SDM_PORT option in the Lattice Radiant Device Constraint Editor to enable the DONE pin.

In feature row hardware default mode state, the DONE pin is disabled by default (functions as a general purpose I/O pin).

When using JTAG to configure the SRAM, the DONE pin is driven by a boundary scan cell so the state of the DONE pin is not meaningful. Once configuration is complete, the DONE pin assumes the behavior as defined by the SDM_PORT setting in the feature row. If the DONE pin is enabled, it will be pulled high as soon as configuration is complete. This behavior can be used to monitor the configuration status of the device.

4.7. Controller SPI sysCONFIG Pins

Controller SPI sysCONFIG pins are dual-purpose I/O pins. Follow the guidelines presented in the sysCONFIG Pins section to use these pins as either sysCONFIG pins or general purpose I/O pins in user mode. These pins are powered by the V_{CCIO0} and V_{CCIO2} voltages. Selected pins are shared between the target SPI, controller SPI, and I2C ports.

4.7.1. MCLK

The MCLK/CCLK pin is an input/output pin with weak pull-up shared between the controller SPI port and target SPI port. On the controller SPI port, MCLK is an output clock signal used to drive an external SPI memory device to sequentially load configuration data for the FPGA. The MCLK/CCLK pin functions as the MCLK pin (output pin with weak pull-up) when the device is configured for dual-boot or external boot. A 1-kΩ pull-up resistor is required when using these boot modes. MCLK provides the reference clock for the SCLK input of the SPI storage device attached to the MachXO4 device controller SPI port. Several different output clock frequencies are supported. The maximum MCLK frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing Specifications of the MachXO4 Family Data Sheet (FPGA-DS-02125). MCLK actively drives the memory device until all the configuration data is received. When the device enters user mode, the MCLK output tri-states. This allows the MCLK to become a general purpose I/O. In most post-configuration applications, MCLK is used as the reference clock for performing memory transactions with the external SPI storage device. Refer to the Controller SPI Mode section for details. You must enable the MASTER_SPI_PORT option to use the MCLK/CCLK pin for the controller SPI port.

The MachXO4 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 option selected using the Lattice Radiant Device Constraint Editor. For a complete list of the supported MCLK frequencies, refer to Table 7.7.

At startup, the lowest frequency MCLK is used by the FPGA. During the initial stages of device configuration, the frequency value specified using MCCLK_FREQ contained in the bitstream is loaded into the FPGA. Once the device accepts this new MCCLK_FREQ value, the MCLK output begins driving the selected frequency. When selecting the MCLK frequency, do not exceed the frequency specification of the configuration memory or the PCB. When making decisions on MCCLK_FREQ value, first review the sysCONFIG Port Timing Specifications of the MachXO4 Family Data Sheet (FPGA-DS-02125).

In the feature row hardware default mode state, the MCLK/CCLK pin functions as the CCLK pin by default.

4.7.2. CSSPIN

The CSSPIN pin is an input/output pin with weak pull-up. On the controller SPI port, CSSPIN is an active-low chip-select output that drives the external SPI storage device chip select. In dual-boot or external boot mode, the MachXO4 device asserts CSSPIN until all configuration data is loaded, after which the CSSPIN pin enters a high-impedance state. A pull-up resistor is required when using these boot modes. Adding a 4.7 kΩ to 10 kΩ pull-up resistor to the CSSPIN pin on the MachXO4 device is recommended. You must enable the controller SPI port to use the CSSPIN for controller SPI operations.

Some SPI storage device manufacturers require the chip select input of the storage device to ramp in unison with the storage device V_{CC} rail. If necessary, ensure CSSPIN ramps in tandem with the SPI storage device V_{CC} input. When configuring from an external SPI flash, ensure that the SPI flash V_{CC} input is at a recommended operating level that matches the MachXO4 device V_{CCIO2} voltage.

The SN pin must be de-asserted (high) when the CSSPIN pin is asserted (low).

In the feature row hardware default mode state, the CSSPIN pin functions as a general purpose I/O pin with weak pull-up.

4.7.3. SISPI/D0

The SI/SISPI/D0 pin is a bi-directional pin shared between the controller SPI port and target SPI port. On the controller SPI port, SISPI is the serial data output for SPI command and data. The pin becomes D0 of the data bus in dual or quad read mode. The MachXO4 device drives SISPI until all configuration data is loaded, after which the SISPI pin enters a high-impedance state. You must enable the MASTER_SPI_PORT option to use the SI/SISPI/D0 pin for the controller SPI port.

In the feature row hardware default mode state, the SI/SISPI/D0 pin functions as the SI pin by default.

4.7.4. SPISO/D1

The SO/SPISO/D1 pin is a bi-directional pin shared between the controller SPI port and the target SPI port. On the controller SPI port, SPISO is the serial data input. The pin becomes D1 of the data bus in dual or quad read mode. You must enable the MASTER_SPI_PORT option to use the SO/SPISO/D1 pin for the controller SPI port.

In the feature row hardware default mode state, the SO/SPISO/D1 pin functions as the SO pin by default.

4.7.5. D2

The SCL/D2 pin is a bi-directional pin shared between the controller SPI port and the I2C port. On the controller SPI port, the pin becomes D2 of the data bus in quad read mode. You must enable the MASTER_SPI_PORT option to use SCL/D2 for the controller SPI port.

In the feature row hardware default mode state, the SCL/D2 pin functions as the SCL pin by default.

4.7.6. D3

The SDA/D3 pin is a bi-directional pin shared between the controller SPI port and the I2C port. On the controller SPI port, the pin becomes D3 of the data bus in quad read mode. You must enable the MASTER_SPI_PORT option to use SDA/D3 for the controller SPI port.

In the feature row hardware default mode state, the SDA/D3 pin functions as the SDA pin by default.

4.8. Target SPI sysCONFIG Pins

Target SPI sysCONFIG pins are dual-purpose I/O pins. Follow the guidelines presented in the sysCONFIG Pins section to use these pins as either sysCONFIG pins or general purpose I/O pins in user mode. These pins are powered by the V_{CCIO2} voltage. Selected pins are shared between the target SPI and controller SPI ports.

4.8.1. CCLK

The MCLK/CCLK pin is an input/output pin with weak pull-up shared between the target SPI port and controller SPI port. On the target SPI port, CCLK is the input clock signal for the target SPI configuration interface driven by the external SPI controller to sequentially load configuration data for the FPGA. The maximum CCLK frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing Specifications of the MachXO4 Family Data Sheet (FPGA-DS-02125).

In the feature row hardware default mode state, the MCLK/CCLK pin functions as the CCLK pin by default. The feature row must be configured to enable the target SPI port if you want to use the port to reprogram the MachXO4 device after it enters user mode.

4.8.2. SN

On the target SPI port, SN is the active-low chip-select input for the target SPI configuration interface. An external SPI bus controller asserts the SN pin low to perform actions using the MachXO4 device programming and configuration logic. Adding an external pull-up resistor to the SN pin is recommended to augment the weak internal pull-up. In user mode, you must enable the SLAVE_SPI_PORT option to use the SN pin for the target SPI port. Otherwise, the SN pin is a general purpose I/O.

In the feature row hardware default mode state, the SN pin is available by default.

Proper operation of the MachXO4 device depends upon maintaining the SN pin in the correct state:

- SN must be de-asserted (high) when configuring using the controller SPI mode.
- SN must be de-asserted when configuring using the SDM mode.
- SN must be de-asserted when the MachXO4 device is in user mode.
- SN must be de-asserted when accessing the configuration logic in the MachXO4 device using the I2C port.
- When SN is asserted (low), CSSPIN must be de-asserted (high). De-asserting CSSPIN places the shared SPI pins into a high-impedance state. The controller SPI port and target SPI port share three common pins: MCLK/CCLK, SI/SISPI/D0, and SO/SPISO/D1. The MachXO4 device permits both ports to be available at the same time. However, these ports are not permitted to be accessed at the same time. The controller SPI port and target SPI port must be time-multiplexed when both ports are enabled.

4.8.3. SI/SISPI

The SI/SISPI/D0 is a bi-directional pin shared between the target SPI port and controller SPI port. On the target SPI port, SI is an input data pin. The SI signal carries data from the external SPI controller to the MachXO4 device configuration logic. You must enable the SLAVE_SPI_PORT option to use the SI/SISPI/D0 pin for the target SPI port.

In the feature row hardware default mode state, the SI/SISPI/D0 pin functions as the SI pin by default.

4.8.4. SO/SPISO

The SO/SPISO/D1 pin is a bi-directional pin shared between the target SPI port and controller SPI port. On the target SPI port, SO is an output data pin. The SO signal carries data from the MachXO4 device configuration logic to the external SPI controller. You must enable the SLAVE_SPI_PORT option to use the SO/SPISO/D1 pin for the target SPI port.

In the feature row hardware default mode state, the SO/SPISO/D1 pin functions as the SO pin by default.

4.9. I2C sysCONFIG Pins

I2C sysCONFIG pins are dual-purpose I/O pins. Follow the guidelines presented in the sysCONFIG Pins section to use these pins as either sysCONFIG pins or general purpose I/O pins in user mode. These pins are powered by the V_{CCIO0} voltage. Selected pins are shared between the I2C and controller SPI ports.

4.9.1. SCL

The SCL/D2 pin is a bi-directional, open-drain pin. On the I2C port, SCL is the I2C serial clock pin. It is an output when the MachXO4 device I2C controller is controlling transactions on the bus and an input when an external I2C controller is accessing the resources inside the MachXO4 device. SCL requires an external pull-up resistor to operate. In user mode, you must enable the I2C_PORT option and instantiate the embedded function block (EFB) to use the SCL pin for the I2C port (refer to the I2C Configuration Mode section for details). Otherwise, the SCL pin is a general purpose I/O.

In feature row hardware default mode state, the SCL pin is available by default.

4.9.2. SDA

The SDA/D3 pin is a bi-directional, open-drain pin. On the I2C port, SDA is the I2C serial data input/output pin. The pin direction changes dynamically during data transactions on the I2C bus. The state of the pin depends on the current bus controller and the operation being performed by the controller. SDA requires an external pull-up resistor to operate. In user mode, you must enable the I2C_PORT option and instantiate the EFB to use the SDA pin for the I2C port (refer to the I2C Configuration Mode section for details). Otherwise, the SDA pin is a general purpose I/O.

In feature row hardware default mode state, the SDA pin is available by default.

4.10. JTAG Pins

JTAG pins provide a standard IEEE 1149.1 test access port (TAP). The JTAG port is the only configuration port on the MachXO4 device that can perform configuration, programming, and multi-device configuration functions. Programming and configuration over the JTAG port uses IEEE 1532-compliant commands. In addition to IEEE 1532 capabilities, the MachXO4 device provides all the mandatory IEEE 1149.1 TAP commands allowing printed circuit board assembly verification. In user mode, the JTAG port is enabled by default. You can use JTAG pins as general purpose I/O pins by disabling the JTAG_PORT option. However, although JTAG pins can be used as general purpose I/O pins, it is recommended that the JTAG pins be used as dedicated configuration and programming pins. These pins are powered by the V_{CCIO0} voltage.

In feature row hardware default mode state, the JTAG port is enabled by default.

When the JTAG port is disabled, the JTAGENB input is enabled. JTAGENB permits the JTAG pins to be multiplexed. Asserting JTAGENB (high) causes the JTAG pins to function according to the IEEE 1149.1 standard. De-asserting JTAGENB (low) causes the JTAG pins to become general purpose I/O pins. Design the JTAG port circuitry carefully when taking advantage of the JTAG pin multiplexing capability. Avoid bus contention between the logic attached to the JTAG port.

When using JTAG to configure or program the device, sysCONFIG pins cannot be used to determine the progress of configuration or programming. For example, the DONE pin is driven by the boundary scan cell according to the JTAG standard rather than normal internal logic, so the state of the DONE pin is not meaningful.

4.10.1. TCK

The test clock (TCK) pin 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. 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 MachXO4 Family Data Sheet (FPGA-DS-02125). The TCK pin does not have an internal pull-up resistor. An external pull-down resistor of 4.7 kΩ is recommended to avoid inadvertently clocking the TAP controller as power is applied to the MachXO4 device.

4.10.2. TMS

The test mode select (TMS) pin is an input pin that controls the progression through the IEEE 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 on the TMS pin is provided according to the JTAG specification. The internal resistor is pulled up to the V_{CCIO0} voltage.

4.10.3. TDI

The test data input (TDI) pin is used to shift in serial test instructions and data. Wire this pin 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 the V_{CCIO0} voltage.

4.10.4. 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. Wire this pin 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 the V_{CCIO0} voltage.

4.10.5. JTAGENB

The JTAG enable (JTAGENB) 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 functionality is only applicable in user mode when the JTAG_PORT option is disabled through the Lattice Radiant Device Constraint Editor in the Global tab. Otherwise, the pin is a general purpose I/O when the JTAG port is enabled. Figure 4.5 shows the default behavior of the MachXO4 device JTAG port.

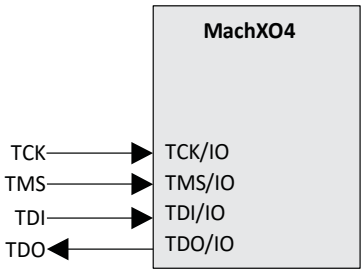


Figure 4.5. JTAG Port Behavior with JTAG_PORT = ENABLE

When the JTAG port is disabled through the JTAG_PORT option in the Lattice Radiant Device Constraint Editor in the Global tab, the JTAGENB pin becomes a dedicated input pin. Driving JTAGENB low disables the JTAG port thereby making the four JTAG pins general purpose I/O pins. Driving JTAGENB high enables the JTAG port. Figure 4.6 shows the JTAG port behavior as controlled by JTAGENB.

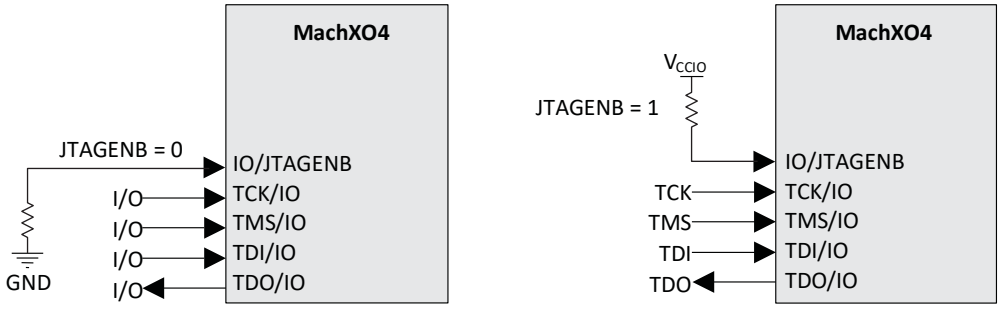


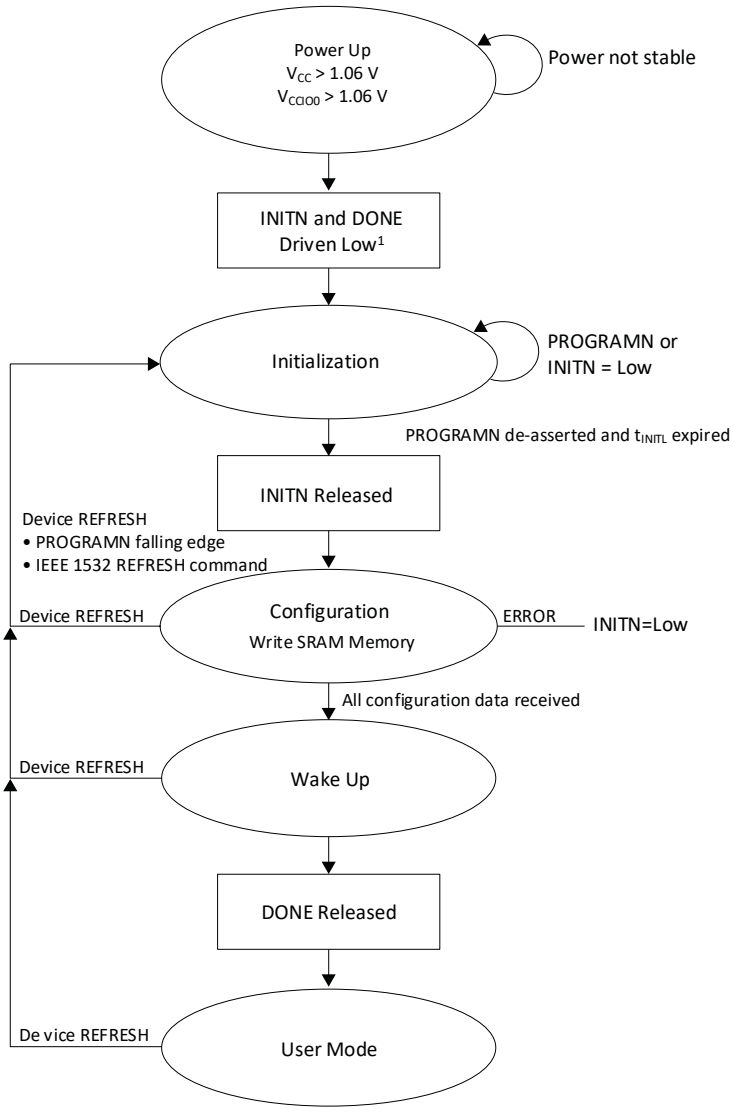
Figure 4.6. JTAG Port Behavior with JTAG_PORT = DISABLE

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

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

5. Configuration Process and Flow

Before entering user mode, the MachXO4 device goes through a sequence of phases, including initialization, configuration, and wake-up.



Note:
1. External INITN and DONE are bidirectional, open-drain I/O pins only when enabled.

Figure 5.1. Configuration Flow

Table 4.3 shows the ports supported by MachXO4 devices for programming and configuration, which includes the industry standard JTAG interface. Each port provides a method to access the MachXO4 device SRAM or internal non-volatile memory. The Memory Space Accessibility section provides information about the capabilities of each port.

The ports capable of accessing the SRAM have a priority order. However, do not permit simultaneous access to the configuration logic from sysCONFIG ports. The operation of the configuration logic is not defined when a low priority port is interrupted by a higher priority port. Any attempt to interrupt an on-going configuration is not defined and not supported.

5.1. Power-Up Sequence

For the MachXO4 device to operate, power must be applied to the device. During a short period of time, as power supplies ramp, the FPGA stays in an indeterminate state. As power ramp-up continues, a power-on reset (POR) circuit inside the FPGA becomes active. Once active, the POR circuit ensures the external I/O pins are in a high-impedance state with weak pull-down. It also monitors the V_{CC} and V_{CCIO0} input rails to ensure the following conditions are met:

- V_{CC} > 1.06 V
- V_{CCIO0} > 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 MachXO4 device drives INITN and DONE low. When INITN is asserted low and DONE is de-asserted low, the device enters the initialization state.

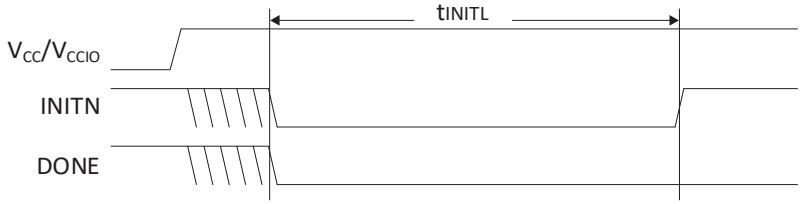


Figure 5.2. Configuration from POR Timing

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

5.2. Initialization

The MachXO4 device enters the initialization phase immediately after the POR circuit drives INITN and DONE low, the PROGRAMN pin is pulsed (falling-edge transition has occurred), or the REFRESH command is issued. The purpose of the initialization phase is to clear the SRAM configuration memory of the FPGA. If control information download mode (CDM) is enabled (default), critical device control information is downloaded from non-volatile memory (feature row settings) into the shadow registers during the initialization phase.

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

- The t_{INITL} period has elapsed.
- The PROGRAMN pin is de-asserted (high).
- The INITN pin is no longer asserted low by an external controller if applicable.

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

During the t_{INITL} period, the FPGA clears the configuration SRAM. When the MachXO4 device is part of a chain of devices, each device has a different t_{INITL} initialization time. The FPGA with the slowest t_{INITL} parameter can prevent other devices in the chain from starting to configure. Prematurely releasing INITN high in a multi-device chain might cause configuration of one or more chained devices to intermittently fail.

5.3. Configuration

Releasing the INITN pin so that the signal goes high causes the MachXO4 device to enter the configuration phase. The FPGA can then accept a configuration bitstream created by the Lattice Radiant software.

The MachXO4 device begins fetching configuration data from non-volatile memory. The non-volatile memory is either internal memory or an external SPI storage device. The MachXO4 device does not exit the configuration phase if there is no valid configuration data in the non-volatile memory. Programming of the non-volatile memory (internal or external) with the correct data through the ports provided is essential for device configuration to succeed.

During configuration, INITN functions as an error signal. Any subsequent assertion of INITN indicates that an error exists in the configuration data and the FPGA device will not operate.

5.3.1. Configuration Time

The configuration time for the MachXO4 device can be estimated by calculating the bitstream loading time from either external memory or internal flash. The configuration time depends on the size of the bitstream. The estimated configuration time can be calculated based on the following equation.

Configuration Time = (Bitstream Size / Configuration Clock Frequency)

5.3.2. Configuration Time Parameters

Table 5.1. Configuration Time Parameters

Parameter	Description
Bitstream Size	Bitstream size can be obtained in the Bitstream and Internal Flash Sizes section. For the worst-case scenario, select the bitstream size with maximum EBR.
Configuration Clock Frequency	Configuration clock frequency can be found in MachXO4 Family Data Sheet (FPGA-DS-02125) and MachXO4 Root-of-Trust Family Data Sheet (FPGA-DS-02126) under the timing specifications for the selected configuration mode (controller SPI, target SPI, I2C, or JTAG). If configuration uses internal flash, refer to the flash download time in the datasheet. For the worst-case scenario, use a controller SPI clock that is minus 10% from the f_{MAX} to cater for process variation on the internal oscillator. If you are using more than one data lane for the controller SPI or target SPI, multiply the configuration clock frequency by the number of data lanes.

5.4. Wake-up

The wake-up phase covers the transition from the configuration phase to user mode. When configuration is complete (after configuration memory has been loaded), the MachXO4 device goes through a wake-up sequence involving a set of internal and external signals. The FPGA asserts an internal DONE status bit which starts the wake-up state machine to sequentially release four control strobe signals. The FPGA enters user mode when the wake-up phase completes.

5.4.1. Wake-Up Signals

Table 5.2 lists the internal and external (control strobe) signals.

Table 5.2. Wake-Up Signals

Signal Name	Description
DONE	Internal DONE status bit asserted (high) after all configuration data has been loaded.
Global Output Enable (GOE)	GOE controls the output drivers. When GOE is de-asserted (low), the device I/O buffers are prevented from driving the pins. When GOE is asserted (high), the device I/O pins exit the high-impedance state with weak pull-down and take on their programmed output function. **Note:** The FPGA inputs are always active. However, the input signals are prevented from performing any action on the FPGA flip-flops by the assertion of GSR.
Global Set/Reset (GSR)	GSR is a control signal that, when asserted (low), causes all I/O flip-flops, look-up table (LUT) flip-flops, distributed RAM output flip-flops, and embedded block RAM output flip-flops that have the *GSR enabled* attribute to be set or cleared per their hardware description language definition. GSR is used to set or reset the core of the device when asserted or de-asserted, respectively. GSR is asserted (low) during configuration and de-asserted (high) in the wake-up sequence. In user mode, the user design controls the GSR signal.
Global Write Disable (GWDISn)	GWDISn is a control signal that overrides the write enable strobe for all RAM logic inside the FPGA and is asserted (low) before device wake-up. Since the inputs of the FPGA are always active, keeping GWDISn asserted prevents accidental corruption of the instantiated RAM resources inside the FPGA thereby safeguarding the integrity of the RAM blocks and LUTs in the device. This signal does not control the primary input pin but controls specific control ports for the RAM blocks and LUTs.
External DONE	The external DONE pin is a bi-directional, open-drain I/O pin when enabled. An external agent can hold the external DONE pin low to prevent the wake-up process of the MachXO4 device from proceeding. The wake-up process completes when the external DONE signal toggles high. Wake-up completes uninterrupted when the external DONE pin is not enabled.

5.4.2. Wake-Up Sequence

When the MachXO4 device is the only device in a chain or the last device in a chain, the wake-up sequence is initiated by the completion of configuration. The following is the wake-up sequence:

1. DONE bit set to 1.
2. Global Output Enable (GOE) signal asserts (high).
3. Global Set/Reset (GSR) signal de-asserts (high) and Global Write Disable (GWDISn) signal de-asserts (high).
4. DONE pin is released (high-impedance state). This causes the DONE signal to go high due to an external pull-up resistor.

Figure 5.3 shows the wake-up sequence using the internal clock.

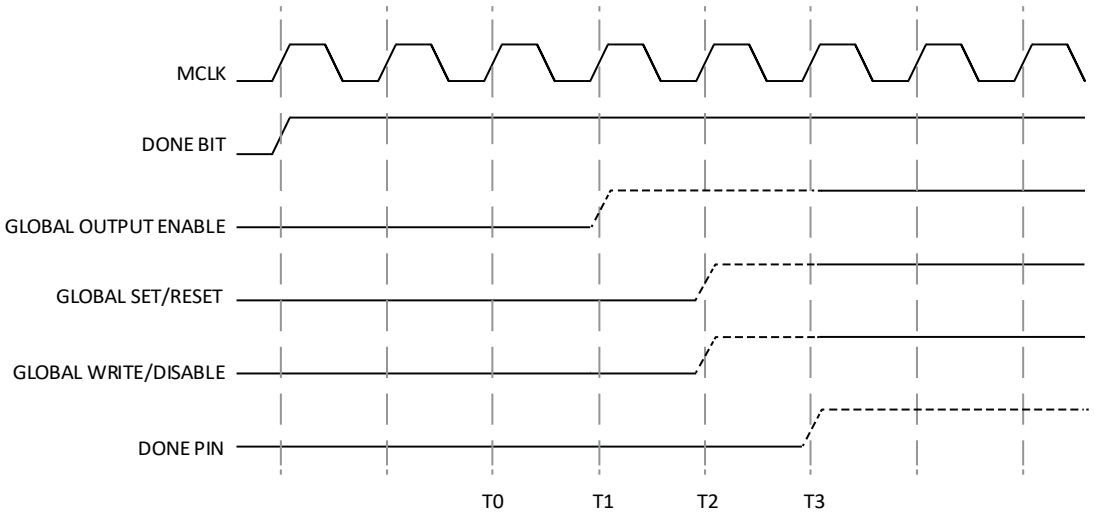


Figure 5.3. Wake-up Sequence Using Internal Clock

5.4.3. Wake-Up Clock Selection

The clock source used to complete the four state transitions in the wake-up sequence is user-selectable. The wake-up sequence is synchronized to the clock source, which defaults to MCLK/CCLK when a sysCONFIG port is used, or TCK when the JTAG port 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 MachXO4 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>);
```

5.5. User Mode

The MachXO4 device enters user mode immediately after the wake-up phase completes. In user mode, the device begins performing its programmed logic operations as defined by the user design.

The device remains in user mode until one of the following events occurs:

- PROGRAMN pin is pulsed.
- REFRESH command is received through a configuration port. Any active configuration port can be used to send a RERESH command.
- Power is cycled or power supply levels drop below their specified trigger levels.

When one of these events occurs, the device exits user mode and enters the initialization phase.

5.6. Internal Flash Programming

There are two methods of programming the internal flash: offline programming mode and transparent programming mode.

5.6.1. Offline Programming Mode

In offline programming mode, the device stops working until the programming is completed. When using the Lattice Radiant Programmer, the offline programming mode is selected using operations starting with FLASH. Unless noted by the operation, the flash sectors accessed are the configuration flash, user flash memory (UFM), and feature row.

5.6.2. Transparent Programming Mode

Transparent programming mode allows the device to continue operating in user mode while the configuration logic programs the internal flash memory. When the internal flash memory programming is completed, the device can download configuration data into the SRAM with the REFRESH instruction. When using the Radiant Programmer, transparent programming mode is selected using operations starting with XFLASH. Unless noted by the operation, the flash sectors accessed are the configuration flash and UFM.

Note: If transparent programming is used on a MachXO4 (LFMXO4) device with two PLLs, the system must put the right side PLL in the reset state during transparent programming. The required duration (erase portion) of background flash programming is specified in the Read One UFM Page (All Devices, WISHBONE/SPI) table of the MachXO4 Hardened Control Functions Software Reference Guide (FPGA-TN-02404). The left side PLL can stay active during transparent programming.

5.7. Feature Row

The MachXO4 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/O pins on the device, and eliminating the need to make changes to your hardware. The feature row can be erased or programmed independently.

The MachXO4 device feature row is used to determine how the MachXO4 device SRAM configuration memory is loaded. When the feature row is erased, feature row values are set back to the hardware default mode state. The contents of the feature row are typically specified using the Lattice Radiant Device Constraint Editor. They can also be manually modified using the Programming File Utility under **Tools** > **Feature Row Editor**.

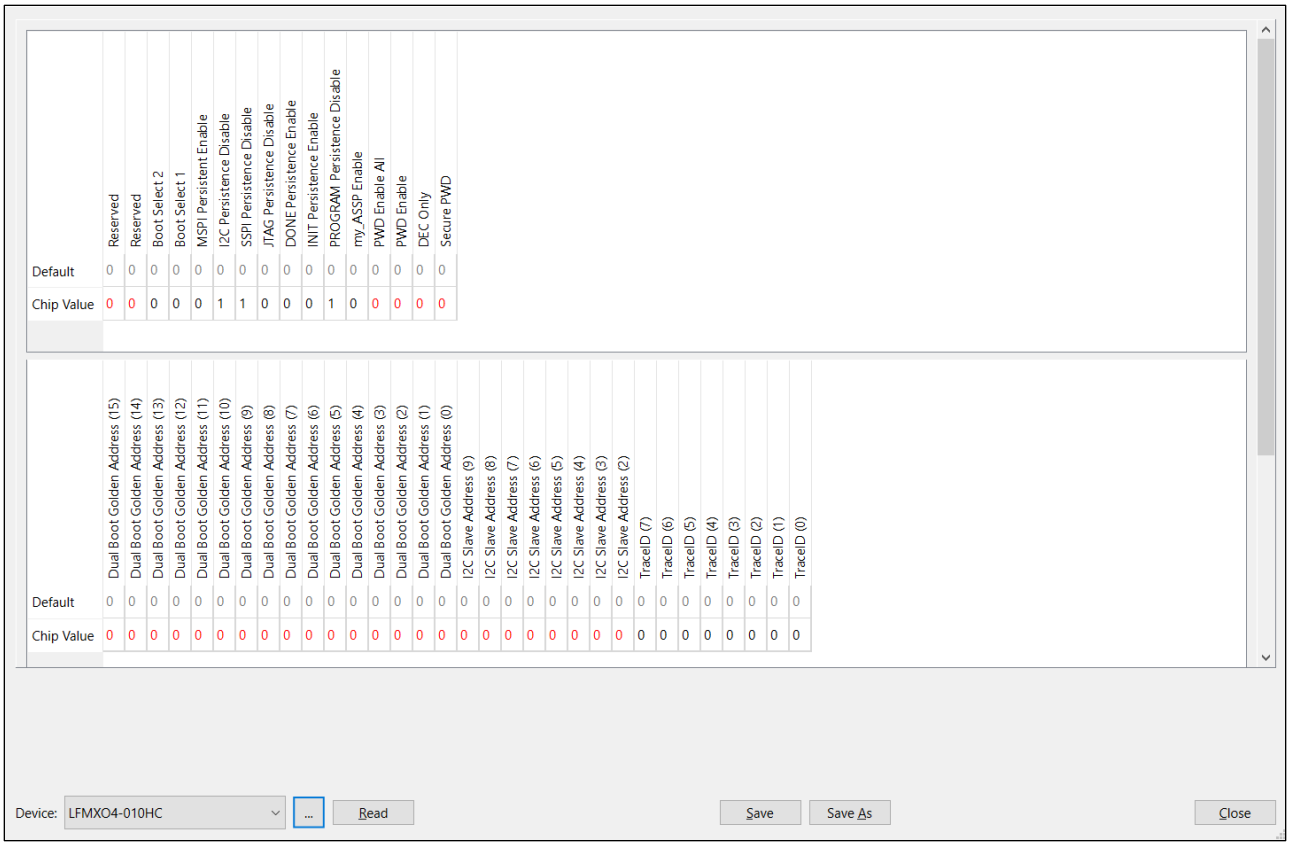


Figure 5.4. Feature Row Example – MachXO4 (LFMXO4) Device

Functions controlled by the feature row and their default values for the MachXO4 device are shown in Table 5.3.

Table 5.3. MachXO4 Device Feature Row Elements

Feature	MachXO4 (LFMXO4)	
	User Mode Default	Hardware Default Mode
BOOT_SEL[2:1], MSPI Persistent Enable[1]	000	000
PROGRAMN Persistence	Disabled	Enabled
INITN Persistence	Disabled	Disabled
DONE Persistence	Disabled	Disabled
Custom IDCODE	0x00000000	0x00000000
TraceID™	00000000	00000000
Security[2]	OFF	OFF
JTAG Port Persistence	Enabled	Enabled
SSPI Port Persistence	Enabled	Enabled
I2C Port Persistence	Disabled	Enabled
MSPI Port Persistence	Disabled	Disabled
I2C Programmable Primary Configuration Address[3, 4]	yyyxxxxx00	1111000000
SRAM OTP	OFF	OFF
Config Flash OTP	OFF	OFF
my_ASSP Enable	OFF	OFF
Password Enable Flash	OFF	OFF
Password Enable All	OFF	OFF
DUALBOOTGOLDEN	Internal	Internal

Notes:

1. Refer to Table 5.4 for more information.
2. Enabled/Disabled using the CONFIG_SECURE option.
3. *y* and *x* are user-programmable from IP Catalog.
4. 1111000001 is a reserved address when the device is erased.

It is strongly recommended that the feature row be modified only during development. This is because the feature row controls the availability of configuration ports, which might inadvertently be made unavailable, thus preventing future updates. In addition, the feature row can make changes such as enabling and disabling the PROGRAMN, INITN, and DONE control and status pins, which might potentially prevent the MachXO4 device from configuring. For example, the PROGRAMN pin can be alternatively used as a general purpose I/O pin. Erasing the feature row causes this pin to switch from a general purpose I/O to the PROGRAMN input. If the pin continues is be driven low as a general purpose I/O, the MachXO4 device will not be able to complete its configuration process.

The feature row can be erased or modified using the Lattice Radiant Device Constraint Editor. The Device Constraint Editor allows you to edit the configuration settings for the MachXO4 device, and then saves the settings in the .pdc file. These settings are applied to the MachXO4 device configuration data during the map, place, and route build phases. Alternatively, the feature row can also be modified using the program feature row utility in the Lattice Radiant Programmer (Programming File Utility under **Tools** > **Feature Row Editor**).

Table 5.4 shows the correlation between the Device Constraint Editor and feature row (boot select) settings.

Table 5.4. Correlation of Device Constraint Editor and Feature Row Settings for MachXO4 (LFMXO4) Device

Feature Row			Device Constraint Editor			Boot Mode[1]
BOOT_SEL[2]	BOOT_SEL[1]	MSPI Persistent Enable	CONFIGURATION	DUALBOOTGOLDEN	MASTER_SPI_PORT	
0	0	0	CFG, CFG_EBRUFM, or CFGUFM	INTERNAL	DISABLE	Single boot (CFG) – Boots from internal flash.
0	1	1	EXTERNAL	INTERNAL	ENABLE	Single boot (EXT) – Boots from external SPI flash.
0	0	1	CFG, CFG_EBRUFM, or CFGUFM	EXTERNAL	ENABLE	Dual boot (CFG_EXT) – Boots from internal flash first; if configuration fails, attempts boot from golden image in external SPI flash.
1	0	1	CFG, CFG_EBRUFM, or CFGUFM	INTERNAL	ENABLE	Dual boot (EXT_CFG) – Boots from external SPI flash first; if configuration fails, attempts boot from golden image in internal flash.

Note:

1. When referencing boot modes, CFG, EXT, and CFG_EXT denote the different boot modes. The configuration data used in each boot mode depends on the selected CONFIGURATION option. For example, for the CFG and CFG_EXT boot modes, the CONFIGURATION setting can be CFG, CFG_EBRUFM, or CFGUFM.

6. Device Configuration

The MachXO4 device provides multiple options for loading configuration data into the SRAM configuration memory from a non-volatile memory. This section describes the functionality of each of the different configuration modes and important settings required in the Lattice Radiant Device Constraint Editor where applicable.

6.1. Self-Download Mode

In self-download mode (SDM), the MachXO4 device automatically retrieves configuration data from the internal flash.

To operate the MachXO4 device in SDM, perform the following:

- Store the entire configuration data in the configuration flash.
- Set the options as shown in Table 6.1.

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

Table 6.1. SDM Configuration Software Settings

Option	Settings
	MachXO4 (LFMXO4)
CONFIGURATION	CFG, CFG_EBRUFM, or CFGUFM

SDM is triggered when power is applied or cycled, a REFRESH command is received, or by pulsing the PROGRAMN pin. SDM cannot be used when configuration memory overflow occurs. An overflow occurs when the configuration data becomes large enough (for example, because of EBR initialization) that it cannot fit into CFG or CFG/UFM. When overflow still happens with CFG/UFM, the controller SPI mode must be used.

The following are advantages of SDM:

- Speed – The MachXO4 device is ready to run in a few milliseconds depending on the density of the device.
- Security – The configuration data is never exposed externally during SRAM loading. You can prevent the internal memory from being read.
- Reduced cost – There is no need to purchase an external SPI storage device specifically reserved for programming.
- Reduced board space – Eliminating use of an external SPI storage device allows your board to be smaller.

6.2. Controller SPI Mode

Controller SPI mode is the only other automatic configuration mode available in the MachXO4 device. When controller SPI mode is enabled, the MachXO4 device automatically retrieves configuration data from an external SPI storage device. The controller SPI port is not available when the MachXO4 device is in the feature row hardware default mode state. Lattice recommends having a secondary configuration port available when the MachXO4 device is in the feature row hardware default mode state to allow you to recover the device in the event of a programming failure.

For the MachXO4 device to operate correctly in the controller SPI mode, ensure the following:

- The POR of the external SPI storage device is lower than the POR of the MachXO4 device. Otherwise, the SPI storage device must be powered first.
- The SPI storage device f_{MAX} is greater than the MachXO4 device MCLK f_{MAX}.
- Board routing requirements are followed to ensure the MachXO4 device setup and hold time parameters are met. Refer to MachXO4 Family Data Sheet (FPGA-DS-02125) for detailed setup and hold time information.

If the SPI storage device POR is higher than the MachXO4 device POR and has a slow ramp, the following sequence of events will occur:

1. MachXO4 device powers up.
2. MachXO4 device begins toggling MCLK.
3. Preamble from the SPI storage device does not return because its POR level is not met.
4. MachXO4 device times out because it fails to get the preamble in time and boot up fails.

As a workaround, you can increase the preamble timer to around 126 ms. If the maximum timer value is still insufficient to delay preamble detection, you can enable the preamble retry count of up to 3 times. With this workaround, the device will typically read the preamble after the SPI storage device is ready. However, if this does not work, the following are other workarounds:

- External host controller to hold INITN.
- External host controller to hold PROGRAMN.
- Add RC delay to INITN as shown in Figure 6.1.

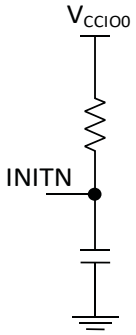


Figure 6.1. RC Delay

To operate the MachXO4 device in controller SPI mode, perform the following:

- Store the entire configuration data in an external SPI storage device.
- Ensure data starts at offset 0x000000 within the SPI storage device.
- Set the options as shown in Table 6.2.

Table 6.2. Controller SPI Configuration Software Settings

Option	Settings
	MachXO4 (LFMXO4)
CONFIGURATION	EXTERNAL
MASTER_SPI_PORT	ENABLE

The BIT file must be programmed into the external SPI storage device. You can achieve this by:

- Transmitting the data using the Lattice Radiant Programmer with JTAG.

- Running ispVME on a microprocessor with JTAG.
- Transmitting the data using an automatic test equipment with JTAG.
- Pre-programming the SPI storage device and pre-assembling it onto your printed circuit board.

Once the SPI storage device contains your configuration data, you can test the configuration.

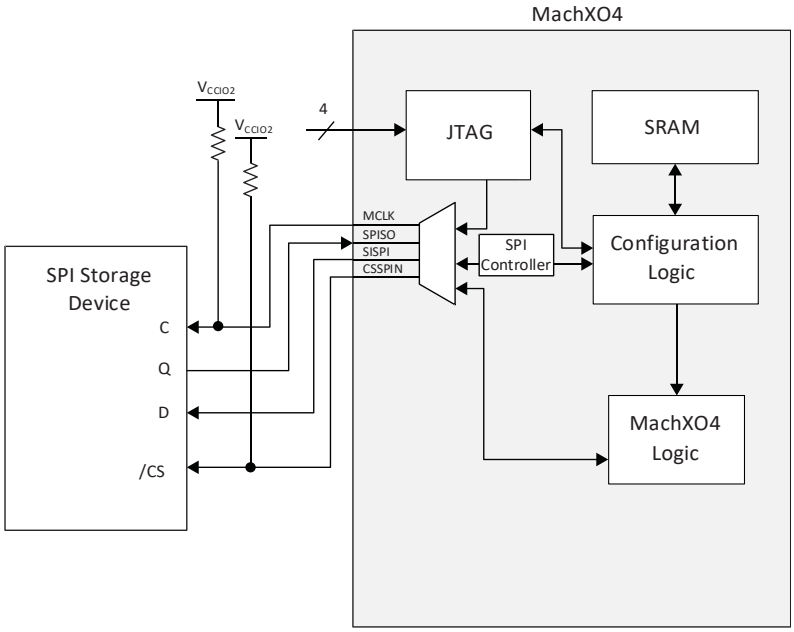


Figure 6.2. Controller SPI Mode

The MachXO4 device begins retrieving configuration data from the SPI storage device when power is applied or cycled, a REFRESH command is received, or the PROGRAMN pin is pulsed. The MCLK/CCLK pin functions as the MCLK and begins driving a nominal 2.08 MHz clock to the SPI storage device SCLK input. Refer to the MCLK section for additional information on using other MCLK frequencies. CSSPIN is asserted low, commands are transmitted to the storage device through the SISPI pin output, and data is read from the storage device through the SPISO pin input. When all configuration data is retrieved from the SPI storage device, the CSSPIN pin is de-asserted.

After the MachXO4 device enters user mode, the controller SPI port pins tri-state. This allows data transfers across the SPI. There are two methods available for transferring data across the SPI bus. The first method is to enable the EFB in the MachXO4 device. Using IP Catalog, you instantiate the EFB and choose the features you want active. A feature available in the EFB is the SPI controller. The SPI controller in the EFB attaches directly to the controller SPI port pins. The SPI controller provides a set of status, control, and data registers for initiating SPI bus transactions.

The second method is to perform controller SPI port transactions from a controller through the JTAG port. The MachXO4 device includes a JTAG-to-MSPI pass-through circuit that allows the SPI storage device to be erased, programmed, and read. The primary method for programming the attached SPI storage device using the JTAG port is to use the Lattice Radiant Programmer to transfer a configuration data file from your PC. This is useful during board development and debug. Another method for programming the SPI storage device 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 Radiant deployment tool and runs the ispVME code. The firmware uses port I/O to drive the JTAG port of the MachXO4 device, which in turn passes the data to the controller SPI port. Refer to the ispVME tool suite for information about updating an attached SPI storage device using a microprocessor.

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

The MachXO4 (LFMXO4) device only supports standard serial read (03h read opcode). An SPI interface has either four or six interface signals. Standard SPI flash uses four signals (CLK, CS, SI, and SO) while quad SPI flash uses six signals (CLK, CS, I/O0, I/O1, I/O2, and I/O3). Quad SPI flash maintains functionality and pin-out compatibility with standard SPI flash while adding dual and quad I/O SPI capabilities.

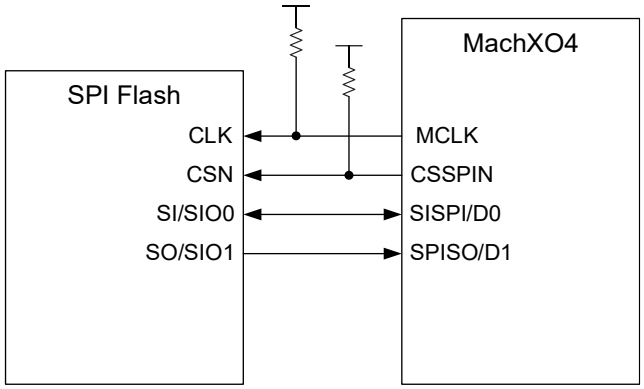


Figure 6.3. Standard or Dual SPI Flash Interface

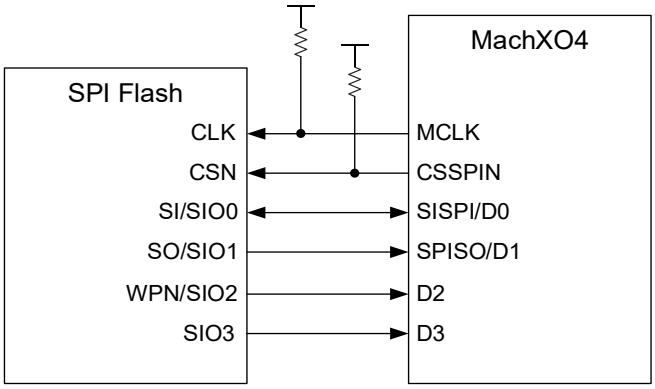


Figure 6.4. Quad SPI Flash Interface

The Radiant flow only generates the bitstream with the default SPI read mode (slow serial read).

6.3. Dual-Boot Configuration Mode

The dual-boot configuration mode is a combination of self-download mode and controller SPI mode. The MachXO4 (LFMXO4) device supports golden image dual-boot configuration.

6.3.1. Golden Image Dual-Boot Configuration

For the MachXO4 (LFMXO4) device in dual-boot mode, the device by default attempts to configure first from the primary image stored in the internal flash memory through SDM. If SDM configuration fails, the device attempts to configure from the golden image stored in an external SPI storage device through the controller SPI mode. The boot order can be reversed if desired using the DUALBOOTGOLDEN option in the Lattice Radiant Device Constraint Editor.

The dual-boot configuration mode can be utilized in conjunction with the MachXO4 device soft error detection (SED) feature without restriction. However, soft error correction (SEC) use is limited to the primary image only for the MachXO4 (LFMXO4) device. Refer to MachXO4 Soft Error Detection (SED) and Correction (SEC) User Guide (FPGA-TN-02406) for more information on the use of the SED and SEC features.

The primary image can fail for one of the following reasons:

- A bitstream CRC error is detected.
- A time-out error is encountered while waiting for a valid bitstream.

A CRC error is caused by incorrect or corrupt data. Data is read from the primary image in rows. As each row enters the configuration engine, the data is checked for CRC consistency. Before the data enters the configuration SRAM, the CRC must be correct. Any incorrect CRC causes the device to erase the configuration SRAM and retrieve configuration data from the golden image. CRC calculation does not include the EBR initialization data.

It is possible for the data to be correct from a CRC calculation perspective, but not be functionally correct. In this instance, the internal DONE bit is never active. The MachXO4 device counts the number of controller clock pulses it has provided after the power-on reset signal is released. When the count expires without DONE becoming active, the device attempts to get its configuration data from the golden image.

The dual-boot configuration mode requires two configuration images. One of the two configuration images is a fail-safe image that is rarely, if ever, updated. The other configuration image is a working image that is routinely updated. For the MachXO4 (LFMXO4) device, the golden fail-safe image is stored in the external SPI storage device while the primary working image is stored in the internal flash memory. One Radiant project can be used to create both the fail-safe and working images. Configure the Radiant project with an implementation named *working*, and an implementation named *failsafe*. Read the Radiant Online Help for more information about using Radiant implementations.

To operate the MachXO4 device in golden image dual-boot configuration mode, perform the following:

- For the MachXO4 (LFMXO4) device, store the entire configuration data for the primary image in the configuration flash (internal flash memory) and the entire configuration data for the golden image in the external SPI storage device.
- For the MachXO4 (LFMXO4) device, ensure data for the golden image starts at offset 0x010000 within the external SPI storage device. Note that this differs from the single image controller SPI mode, which requires the configuration data to be stored at offset 0x000000.
- Set the options as shown in Table 6.3.

Table 6.3. Golden Image Dual-Boot Configuration Software Settings

Option	Settings
	MachXO4 (LFMXO4)
CONFIGURATION	CFG, CFG_EBRUFM, CFGUFM, or EXTERNAL
MASTER_SPI_PORT	ENABLE
DUALBOOTGOLDEN	INTERNAL or EXTERNAL
COMPRESS_CONFIG	ON or OFF

In the Radiant flow, the JEDEC file option must be selected when generating configuration data that is stored in the internal flash while the bitstream file option must be selected when generating configuration data that is stored in the external SPI storage device.

Note: To prevent the MachXO4 (LFMXO4) device from using dual-boot configuration mode when using the controller SPI in the EFB, set the MASTER_SPI_PORT option to EFB_USER. This reserves the controller SPI port pins and prevents dual-boot configuration.

6.3.2. Status Register Readout during Boot Failure

The SPIm Fail 1 (LFMXO4 device) status register bit can be used to check if the device successfully boots from either the primary image or golden image. SPIm Fail 1 is set to 1 when the following conditions are met:

- Primary image internal DONE bit = 0.
- Primary image is fully erased including the internal DONE bit.

However, if the internal DONE bit = 1 and the preamble is not found in the primary image, SPIm Fail 1 is not set to 1 as the device boots from the golden image. Checking the SRAM user code is recommended to determine the image used for device boot up. The primary and golden images must have distinct user codes to facilitate differentiation.

6.4. Target SPI Mode

The MachXO4 device provides a target SPI port that allows you to access features provided by the configuration logic. You can reprogram the SRAM, flash, and feature row, and access status and control registers within the configuration logic block. Reprogramming the flash can be done using offline programming mode or transparent programming mode.

In the target SPI mode, input data is read into the MachXO4 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. SN acts as the chip select signal. When SN is de-asserted (high), the target SPI interface is deselected and the SO/SPISO pin is tri-stated. When SN is asserted, commands can be written into and data can be read from the MachXO4 device. The MachXO4 device target SPI port only accepts mode 0 bus transactions to the configuration logic.

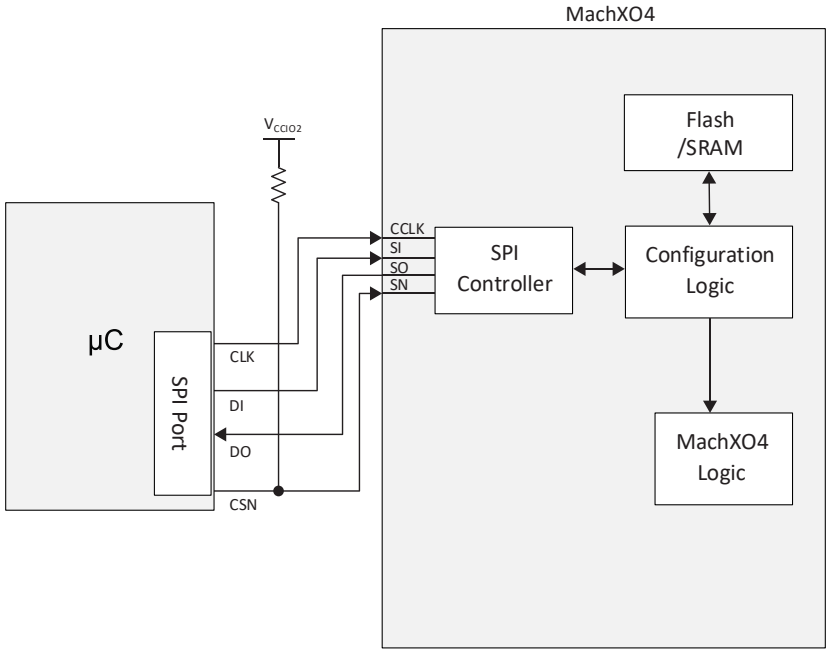


Figure 6.5. Target SPI Mode

The target SPI port is active when the MachXO4 device is in the feature row hardware default mode state. The user mode default for the SLAVE_SPI_PORT option is DISABLE for the MachXO4 (LFMXO4) device. Use the Device Constraint Editor to set the SLAVE_SPI_PORT option in your design to keep the target SPI port or use the port pins as general purpose I/O pins in user mode. Lattice recommends keeping a secondary programming port active (for example, JTAG) in the event the target SPI port is accidentally disabled.

The target SPI port is used to erase, program, and verify the configuration flash, user flash memory, and feature row. It is capable of directly accessing the configuration SRAM. To prevent unintentional erasure of the feature row, it is recommended that the target SPI port be used to perform transparent updates of the flash memory. The target SPI port can issue a REFRESH command to make a newly programmed image active. The REFRESH command can be safely used when the MachXO4 device is using external or dual-boot configuration mode because the REFRESH operation does not begin until SN is de-asserted.

Programming the MachXO4 device using the target SPI port is complex. Lattice provides C source code called SSPIEMbedded to insulate you from the complexity of programming the MachXO4 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.

6.5. I2C Configuration Mode

The MachXO4 device has an I2C port for use in accessing the configuration logic. An I2C controller can communicate with the configuration logic using 10-bit or 7-bit addressing mode. The SCL input can accept a clock frequency up to 400 kHz. You can reprogram the SRAM, flash, and feature row, and access status and control registers within the configuration logic block. Reprogramming the flash can be done in offline programming mode or transparent programming mode.

The I2C port is available when the MachXO4 device is in the feature row hardware default mode state. The user mode default for the I2C_PORT option is DISABLE. Use the Device Constraint Editor to set the I2C_PORT option to ENABLE to keep the I2C port active in user mode. Lattice recommends keeping a secondary programming port active (for example, JTAG) in the event the I2C port is accidentally disabled.

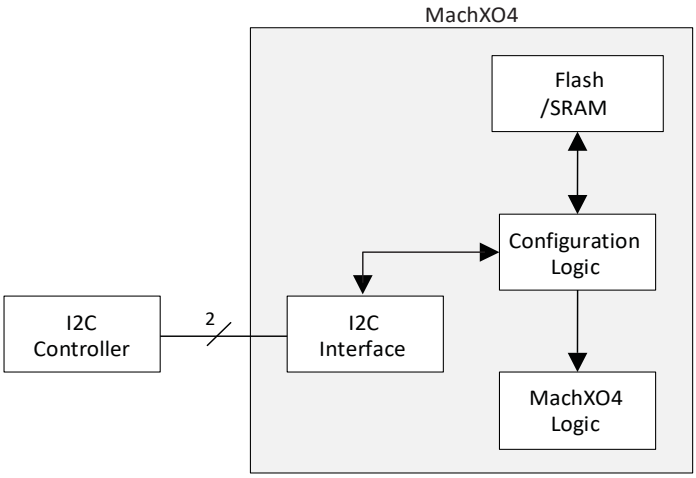


Figure 6.6. I2C Configuration Mode

There are two hardened I2C controllers in the MachXO4 device. The primary I2C controller provides an interface to the MachXO4 device configuration logic. Apart from being a user mode I2C controller, the primary I2C controller also permits access to the configuration logic. The secondary I2C controller is always a user mode I2C controller.

When the MachXO4 device is in the feature row hardware default mode state, the I2C port is enabled allowing you to interact with the primary I2C controller. Whenever the I2C port is enabled, access to the configuration logic is possible. In user mode, the EFB must be instantiated to preserve access to the configuration logic. Moreover, when instantiated, the EFB *wb_clk_i* input must be connected to a valid clock source of at least 7.5X the I2C bus rate (for example, ≥ 3.0 MHz when I2C rate = 400 kHz).

An external I2C controller accesses the configuration logic using address 1000000 (7-bit mode) or 1111000000 (10-bit mode) unless the EFB I2C base address has been modified. Use IP Catalog, not the Device Constraint Editor, 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 6.4 shows the address decoding used to access the I2C resources in the MachXO4 device.

Table 6.4. Target Addresses for I2C Ports

Target Address	I2C Function
yyyxxxxx00	Primary I2C controller configuration logic address. Always responds to 7-bit or 10-bit address.
yyyxxxxx01	User mode primary I2C controller address.
yyyxxxxx10	User mode secondary I2C controller address.
yyyxxxxx11	Primary I2C controller configuration logic reset. Always responds to 7-bit or 10-bit address.

Note: Target I2C addresses can acknowledge (ACK) under all circumstances even if not all ports are configured to be active by the user.

Another I2C resource in the MachXO4 device is located at offset 3 (target address yyyxxxxx11). 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 continue responding with queued data even with the I2C memory transaction interrupted or abandoned. Any new incoming I2C commands might be considered padding bytes or misinterpreted. Clear this condition by writing any value to offset 3. The configuration logic command interpreter resets and any queued data is flushed thereby allowing subsequent I2C memory transactions to the configuration logic to resume correct operation.

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.

6.6. WISHBONE Configuration Mode

The MachXO4 device can access the configuration flash, user flash memory, and the feature row from an internal WISHBONE bus. To use the WISHBONE bus, the EFB must be inserted into your design. You can design logic to interface with the EFB, then perform WISHBONE bus transactions to access resources attached to the configuration logic.

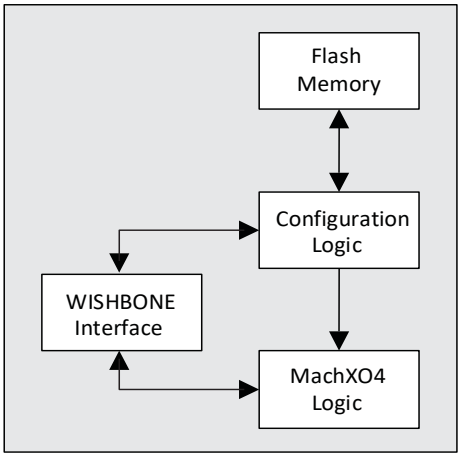


Figure 6.7. WISHBONE Configuration Mode

To access the WISHBONE interface, the MachXO4 device must be in user mode. Accessing and updating resources made available by the configuration logic must be performed in transparent programming mode. Attempting access to the configuration logic in offline programming mode causes a deadlock because the MachXO4 device exits user mode.

For more information about the MachXO4 device WISHBONE interface, refer to MachXO4 Hardened Control Functions User Guide (FPGA-TN-02403).

6.7. JTAG Mode

The JTAG port is the most flexible configuration and programming port available on the MachXO4 device. The JTAG port provides the following functionality:

- Offline flash programming
- Transparent flash memory programming
- Offline SRAM configuration
- Full access to the MachXO4 configuration logic
- Device chaining
- IEEE 1149.1 testability
- IEEE 1532-compliant programming

The JTAG port is enabled when the MachXO4 device is in the feature row hardware default mode state. The MachXO4 device JTAG port pins are not dedicated pins for the IEEE 1149.1 TAP function. JTAG pins can be used as general purpose I/O pins. However, the flexibility and capabilities of the JTAG port makes it most suitable for system and device debug. Therefore, Lattice recommends that the JTAG port remain enabled in every MachXO4 device design and JTAG pins be used as dedicated configuration and programming pins.

The advantages of keeping the JTAG port active include:

- Multi-chain architectures – The JTAG port is the only configuration and programming port that permits the MachXO4 device to be combined in a chain of other programmable logic devices.
- Reveal debug – The Lattice Reveal debug tool is an embeddable logic analyzer tool. It allows you to analyze the logic inside the MachXO4 device similar to how an external logic analyzer permits analysis of board level logic. Reveal access is only available through the MachXO4 device JTAG port.
- SRAM read-back – The JTAG port is the only configuration port that can directly access the MachXO4 device configuration SRAM.
- Boundary scan testability – Board level connectivity testing performed using IEEE 1149.1 JTAG is a key capability for assuring the quality of assembled printed circuit boards. Preserving the MachXO4 device JTAG port is vital for boundary scan testability. Lattice provides boundary scan description language files for the MachXO4 device family through the Lattice website.

6.7.1. 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 6.8 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.

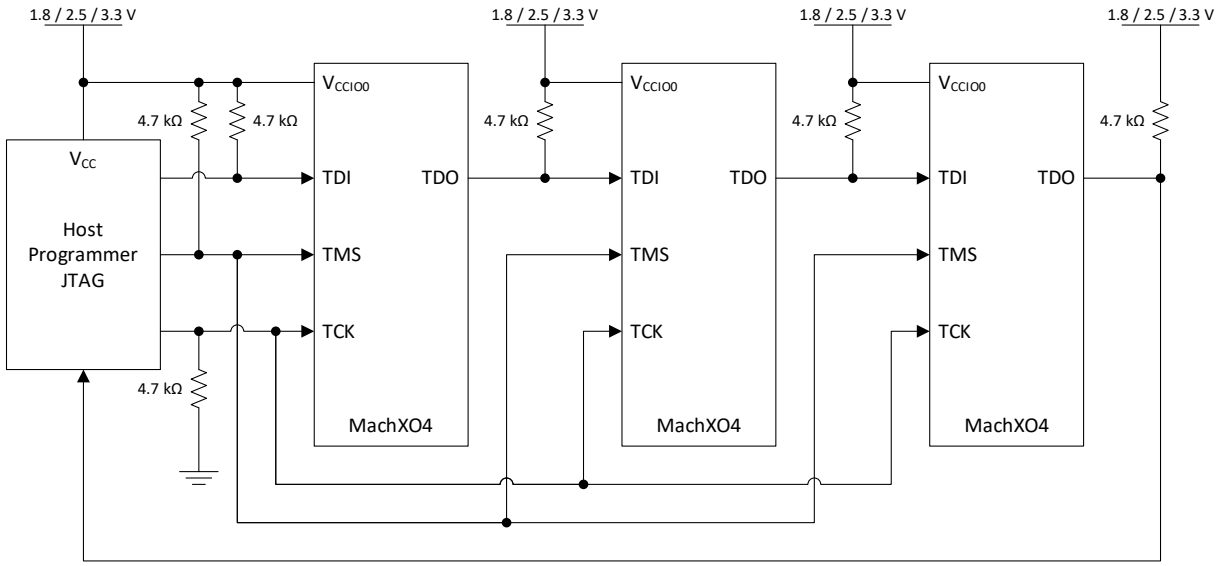


Figure 6.8. JTAG Daisy Chain Example

6.8. TransFR Operation

The MachXO4 device supports the TransFR™ capability. TransFR is described in Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02198). The following is an example of how you can update the bitstream in the MachXO4 device using the TransFR feature.

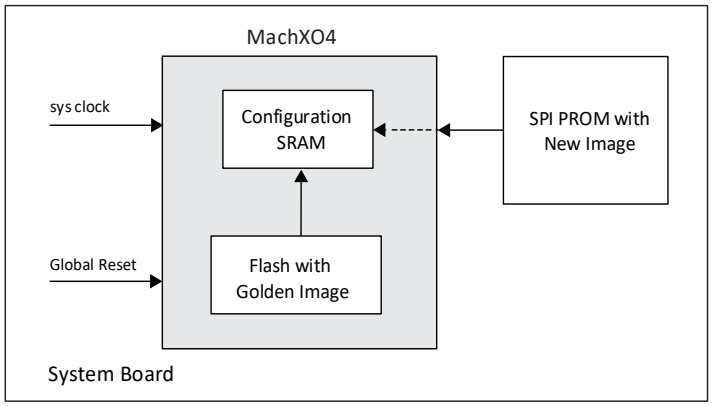


Figure 6.9. Bitstream Update Using TransFR

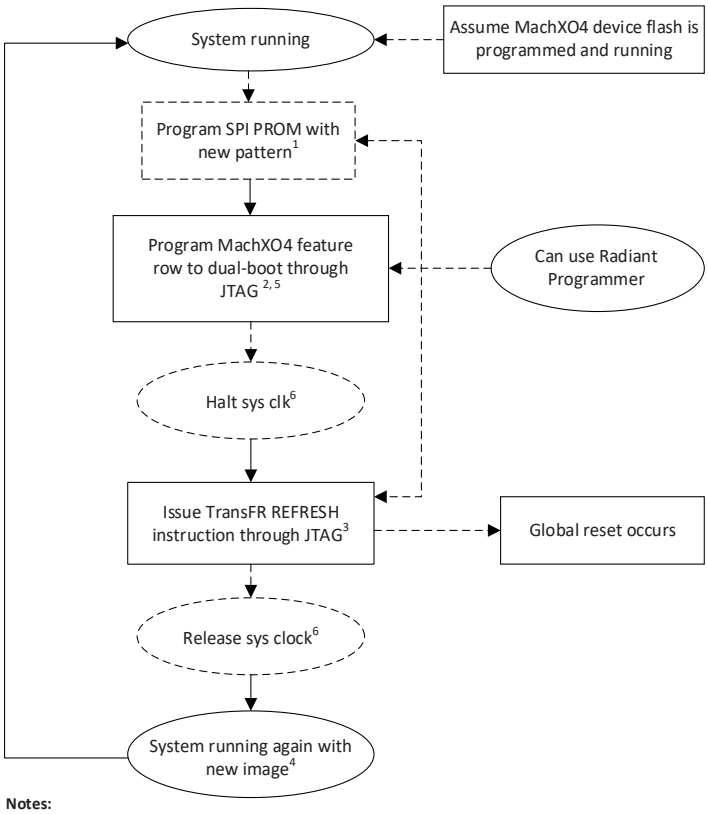
This example assumes you have the golden image stored in flash to initiate the system, then use the external SPI storage device as a resource for image updates without disturbing the system.

Figure 6.10 shows the process flow for performing this task. Use this process flow with caution. Because a global reset is triggered during device wake-up after the REFRESH instruction is issued, attention needs to be given to designing I/O pins with the following characteristics:

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

For I/O pins that meet these characteristics, the state of the I/O pin is not changed during the TransFR refresh but might change once the device enters user mode after the refresh. The following are design tips to avoid this:

- Do not use global reset for critical I/O pins.
- If you must use global reset for critical I/O pins, use the set/reset option so that when GSR occurs, the state of the I/O pin does not trigger a system crash.



Notes:
1. Use operations such as *SPI Flash Background Erase, Program, Verify* for this.
2. Use operations such as *Program Feature Row* for this.
3. Use operations such as *FLASH TransFR* or *XFLASH TransFR* for this.
4. If a new image fails to configure the MachXO4 device, the golden image in flash still configures the MachXO4 device, so the system still runs with the original image.
5. The feature row only needs to be programmed if changes need to be made, for instance disable or enable the JTAG or target SPI port. If no changes need to be made, skip this step.
6. This step is optional.

Figure 6.10. Example Process Flow for Updating Bitstream Using TransFR Feature

6.9. Password-Based Flash Protection

The MachXO4 device supports a password-based security access feature 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 that matches the key expected by the device.

The flash protect key feature requires that a device accessing the MachXO4 device through a configuration port (JTAG, target SPI, I2C, or WISHBONE) provides 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.

For the MachXO4 (LFMXO4) device, the 64-bit flash protect key is stored in the feature row. Two additional feature row settings are specified for enabling the feature: PWD_Enable and PWD_Enable_all.

For more information about the password feature, refer to Using Password Security with MachXO4 Devices (FPGA-TN-02408).

7. Software Selectable Options

The operation of the MachXO4 device configuration logic is managed by options selected in the Lattice Radiant design software. The MachXO4 device uses the non-volatile feature row to determine how it configures. The feature row default settings can be modified using the Lattice Radiant Device Constraint Editor. Modifying the feature row settings changes the operation of the configuration logic.

7.1. Accessing Software Selectable sysCONFIG Options

To access the software selectable sysCONFIG options, perform the following:

1. In the Lattice Radiant design software, open your design project file.
2. Open Device Constraint Editor.
3. Click on the **Global** tab.
4. Locate the sysCONFIG tree. Figure 7.1 shows sysCONFIG options in the Global page in Device Constraint Editor. sysCONFIG options can be divided into three categories:
   - Configuration mode and port options
   - Bitstream generation options
   - Security option

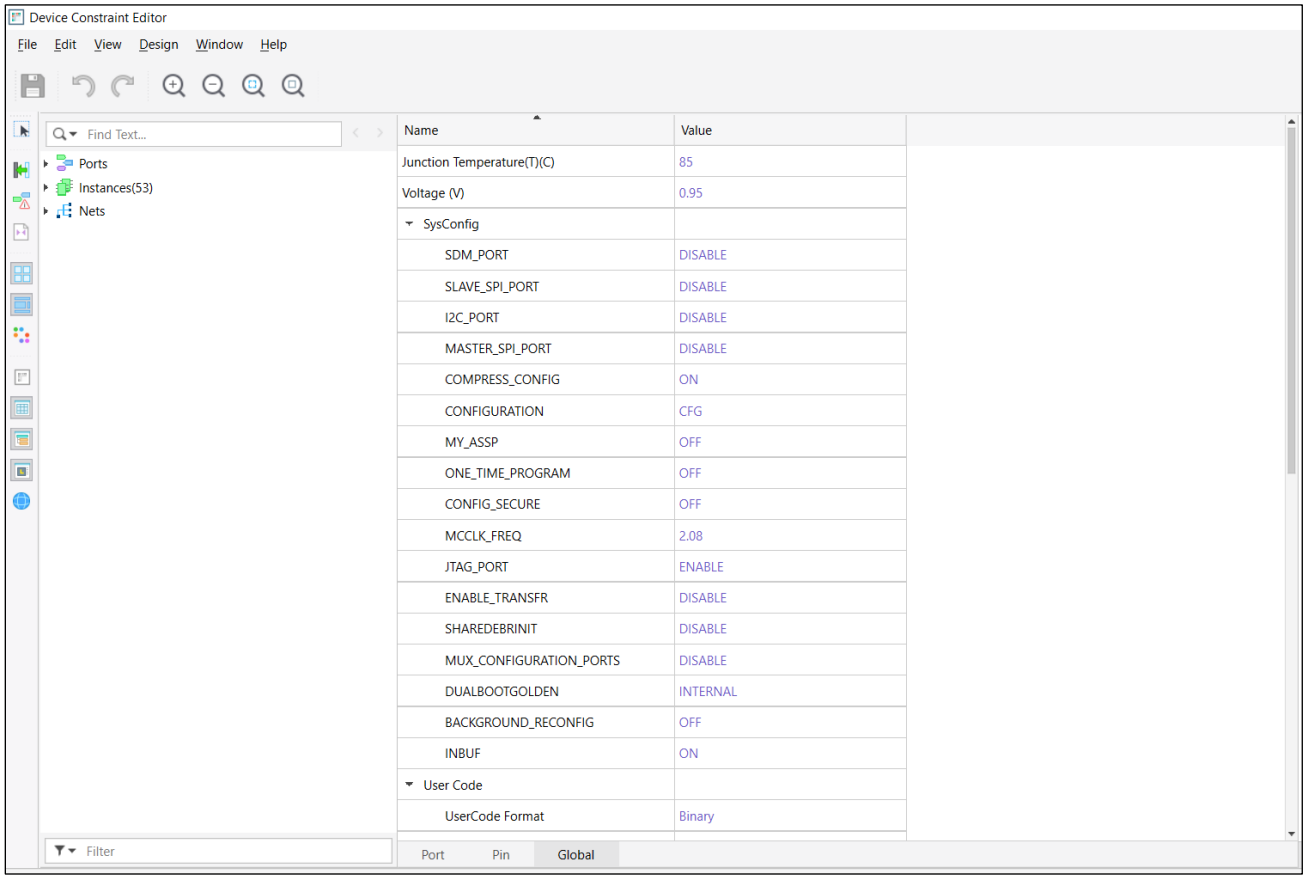


Figure 7.1. sysCONFIG Options on Global Tab of Device Constraint Editor

7.2. Configuration Mode and Port Options

The configuration mode and port options allow you to decide which configuration ports continue to operate after the MachXO4 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 SPI storage device. 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.

Table 7.1 shows the configuration mode and port options. These options can be used in any combination.

Table 7.1. Configuration Mode and Port Options

Option Name	Default Setting	All Settings
JTAG_PORT	ENABLE	DISABLE, ENABLE
SLAVE_SPI_PORT	DISABLE	DISABLE, ENABLE
MASTER_SPI_PORT	DISABLE	DISABLE, ENABLE, EFB_USR
I2C_PORT	DISABLE	DISABLE, ENABLE
SDM_PORT	DISABLE	DISABLE, PROGRAMN, DONE, INITN, PROGRAMN_DONE, PROGRAMN_DONE_INITN
MCCLK_FREQ	2.08	2.08, 2.46, 3.17, 4.29, 5.54, 7.00, 8.31, 9.17, 10.23, 13.30, 14.78, 20.46, 26.60, 29.56, 33.25, 38.00, 44.33, 53.20, 66.50, 88.67, 133.00
ENABLE_TRANSFR	DISABLE	DISABLE, ENABLE

7.2.1. JTAG_PORT

The JTAG_PORT option allows you to decide how the JTAG configuration port pins operate when the device is in user mode. Lattice recommends keeping a secondary programming port active (for example, JTAG) in the event the primary port is accidentally disabled.

Table 7.2. JTAG_PORT Option

Option Name	Setting(s)	Description
JTAG_PORT	DISABLE	JTAG I/O pins are controlled dynamically using the JTAGENB pin. The JTAGENB pin is only available when JTAG_PORT is disabled. When JTAGENB is asserted high, the four JTAG I/O pins provide the IEEE 1149.1 JTAG interface. When JTAGENB is asserted low, the four JTAG I/O pins can be used as general purpose I/O pins.
	ENABLE (default)	JTAG I/O pins are dedicated I/O pins providing the IEEE 1149.1 JTAG interface.

7.2.2. SLAVE_SPI_PORT

The SLAVE_SPI_PORT option allows you to decide how the target SPI sysCONFIG pins operate when the device is in user mode. SLAVE_SPI_PORT can be enabled at the same time as MASTER_SPI_PORT. Therefore, you must ensure that the internal SPI controller does not perform SPI transactions at the same time as the external SPI controller.

Table 7.3. SLAVE_SPI_PORT Option

Option Name	Setting(s)	Description
SLAVE_SPI_PORT	DISABLE (default)	Disconnects the SPI port I/O pins from the configuration logic. Setting only this option alone to disabled will not make the pins become general purpose I/O pins. Both the SLAVE_SPI_PORT and MASTER_SPI_PORT options must be disabled to allow the pins to become general purpose I/O pins.
	ENABLE	Preserves the target SPI port I/O pins when the device is in user mode. When the pins are preserved, an external SPI controller can interact with the configuration logic. This preference also prevents you from over-assigning I/O functions to the port pins.

7.2.3. MASTER_SPI_PORT

The MASTER_SPI_PORT option allows you to preserve the controller SPI port after the device enters user mode. MASTER_SPI_PORT can be enabled at the same time as SLAVE_SPI_PORT. Therefore, you must ensure that the internal SPI controller does not perform SPI transactions at the same time as the external SPI controller.

Table 7.4. MASTER_SPI_PORT Option

Option Name	Setting(s)	Description
MASTER_SPI_PORT	DISABLE (default)	Disconnects the SPI port I/O pins from the configuration logic. Setting only this option alone to disabled will not make the pins become general purpose I/O pins. Both the MASTER_SPI_PORT and SLAVE_SPI_PORT options must be disabled to allow the pins to become general purpose I/O pins.
	ENABLE[1]	Preserves the controller SPI port I/O pins when the device is in user mode. Enabling this option makes the controller SPI or dual-boot configuration (with external SPI storage device) mode active. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. Preserving this port allows transparent programming of the connected external SPI storage device through JTAG using the Radiant Programmer. This preference also prevents you from over-assigning I/O functions to the port pins.
	EFB_USER	Preserves the controller SPI port I/O pins when the device is in user mode. After entering user mode, the SPI controller in the EFB has access to the SPI port for performing SPI bus transactions. This preference also prevents you from over-assigning I/O functions to the port pins.

Note:

1. For the MachXO4 (LFMXO4) device, enabling MASTER_SPI_PORT in combination with CONFIGURATION = EXTERNAL enables controller SPI mode. Enabling MASTER_SPI_PORT in combination with CONFIGURATION = CFG enables dual-boot configuration mode (CFG_EXT or EXT_CFG depending on DUALBOOTGOLDEN).

7.2.4. I2C_PORT

The I2C_PORT option allows you to preserve the I2C port after the MachXO4 device enters user mode. To use the primary and secondary I2C controllers in the EFB, I2C_PORT must be enabled.

Table 7.5. I2C_PORT Option

Option Name	Setting(s)	Description
I2C_PORT	DISABLE (default)	Disconnects the I2C port I/O pins from the configuration logic. The port pins become general purpose I/O pins.
	ENABLE	Preserves the I2C port I/O pins when the device is in user mode. When the pins are preserved and the EFB is instantiated with the wb_clk_i input connected to a valid clock source of at least 7.5X the I2C bus rate, an external I2C controller can interact with the configuration logic. This preference also prevents you from over-assigning I/O functions to the port pins.

7.2.5. SDM_PORT

The SDM_PORT option allows you to select the SDM port control and status pins. The SDM_PORT setting takes effect during the device initialization phase and remains in effect after the device enters user mode. Lattice recommends setting SDM_PORT to PROGRAMN when using the controller SPI or dual-boot configuration mode. The PROGRAMN pin can be used to initiate device reconfiguration if a configuration port is not available to transmit the REFRESH command.

Table 7.6. SDM_PORT Option

Option Name	Setting(s)	Description
SDM_PORT	DISABLE (default)	PROGRAM, INITN, and DONE pins become general purpose I/O pins in user mode.
	PROGRAMN	Preserves the PROGRAMN pin when the device is in user mode. The INITN and DONE pins are general purpose I/O pins. Pulsing PROGRAMN causes the device to exit user mode and enter the initialization phase.
	INITN	Preserves the INITN pin when the device is in user mode. The PROGRAMN and DONE pins become general purpose I/O pins.
	DONE	Preserves the DONE pin when the device is in user mode. The PROGRAMN and INITN pins become general purpose I/O pins.
	PROGRAM_DONE	Preserves the PROGRAMN and DONE pins when the device is in user mode. The INITN pin becomes a general purpose I/O pin.
	PROGRAM_DONE _INITN	Preserves the PROGRAMN, INITN, and DONE pins when the device is in user mode.

7.2.6. MCCLK_FREQ

The MCCLK_FREQ option allows you to alter the MCLK frequency used to retrieve data from an external SPI storage device when using the external SPI or dual-boot configuration mode. Lattice recommends having a back-up configuration port available in the event a clock frequency that is out of specification is specified.

Table 7.7. MCCLK_FREQ Option

Option Name	Setting(s)	Description
MCCLK_FREQ	2.08 (default)	Default clock frequency used by the device (nominal 2.08 MHz ±5.5%) to begin retrieving data from the external SPI storage device. The MCCLK_FREQ value is stored in the bitstream (incoming configuration data) and not the feature row.
	2.46, 3.17, 4.29, 5.54, 7.00, 8.31, 9.17, 10.23, 13.30, 14.78, 20.46, 26.60, 29.56, 33.25, 38.00, 44.33, 53.20, 66.50, 88.67, 133.00	Sets the MCCLK_FREQ value to the selected clock frequency (in MHz). The MCCLK_FREQ value is stored in the bitstream (incoming configuration data) and not the feature row. The 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 device switches to the new clock frequency shortly after receiving the MCLK_FREQ value. Possible frequencies range from 2.08 MHz up to 133 MHz. The maximum frequency is limited by the SPI storage device or system design.

7.2.7. ENABLE_TRANSFR

The ENABLE_TRANSFR option allows you to enable or disable the TransFR operation. The TransFR operation supported by the MachXO4 device requires the configuration data loaded into the configuration SRAM and any future configuration data file loaded into the internal flash memory to have ENABLE_TRANSFR set to ENABLE. Refer to the TransFR Operation section and Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02198) for more information about using TransFR with the MachXO4 device.

Table 7.8. ENABLE_TRANSFR Option

Option Name	Setting(s)	Description
ENABLE_TRANSFR	DISABLE (default)	Disables TransFR operation.
	ENABLE	Enables TransFR operation.

7.3. Bitstream Generation Options

The bitstream generation options allow you to decide how the Lattice Radiant development tools create the configuration data for the MachXO4 device. The CONFIGURATION CUSTOM_IDCODE, and DUALBOOTGOLDEN settings are saved in the feature row as applicable and remain in effect until the feature row is erased. There are options to control the JEDEC and BIT files generated by Radiant.

Table 7.10 shows the bitstream generation options.

Table 7.10. Bitstream Generation Options

Option Name	Default Setting	All Settings
COMPRESS_CONFIG	ON	OFF, ON
CONFIGURATION	CFG	CFG, CFG_EBRUFM, CFGUFM, EXTERNAL
USERCODE	0x00000000	32-bit user code (user electronic signature)
USERCODE_FORMAT	Hex	Binary, Hex, ASCII
CUSTOM_IDCODE	0x00000000	32-bit custom identification code
CUSTOM_IDCODE_ FORMAT	Hex	Binary, Hex
SHAREDEBRINIT	DISABLE	DISABLE, ENABLE
MUX_CONFIGURATION_ PORTS	DISABLE	DISABLE, ENABLE
DUALBOOTGOLDEN	INTERNAL	INTERNAL, EXTERNAL

7.3.1. COMPRESS_CONFIG

The COMPRESS_CONFIG option alters the way JEDEC and BIT files are generated. In the Radiant flow, the JEDEC file option must be selected when generating configuration data that is stored in the internal flash while the bitstream file option must be selected when generating configuration data that is stored in the external SPI storage device. MachXO4 device JEDEC files must always be compressed to fit into internal flash. BIT file compression slightly reduces configuration time when reading configuration data from the external SPI storage device.

Table 7.11. COMPRESS_CONFIG Option

Option Name	Setting(s)	Description
COMPRESS_CONFIG	OFF	Configuration data is not compressed.
	ON (default)	Configuration data is compressed.

7.3.2. CONFIGURATION

The CONFIGURATION option allows you to control where the configuration data is stored thereby defining the type or types of configuration data generated (JEDEC and/or BIT) and where the device boots the image from. Configuration data stored in internal flash is generated as a JEDEC file. Note that when configuration flash is selected, a BIT file that can be used for SRAM configuration is also generated. Configuration data stored in the external SPI storage device is generated as a BIT file. If your configuration data exceeds the space available in internal memory, you must switch to configuring from an external SPI storage device.

Table 7.12. CONFIGURATION Option

Option Name	Setting(s)	Description
CONFIGURATION	CFG (default)	Generates configuration data that is stored in the configuration flash. The configuration data includes EBR initialization data.
	CFG_EBRUFM	Generates configuration data that is stored in the configuration flash. EBR initialization data is stored in the lowest page addresses of the UFM sector. The UFM sector is available in user mode. You must restore the EBR initialization data when making changes to the UFM to guarantee correct operation.
	CFGUFM	Generates configuration data that is stored in the configuration flash. This setting is different from CFG in that it allows configuration data to overflow into the UFM. The configuration data includes EBR initialization data.
	EXTERNAL	Generates configuration data that is stored in the external SPI storage device. This setting is used for the controller SPI mode.

7.3.3. USERCODE

The MachXO4 device configuration flash sector contains a 32-bit register for storing a user-defined value. This user-defined value is also stored in the generated bitstream for redundancy. This register can be initialized with any 32-bit value specified through USERCODE.

Table 7.13. USERCODE Option

Option Name	Setting(s)	Description
USERCODE	0x00000000 (default)	32-bit user-defined value to be stored in a 32-bit register in the FPGA device. Suggested uses include the configuration data version number, a manufacturing ID code, date of assembly, or the JEDEC file checksum. The data format is set using the USERCODE_FORMAT option.

7.3.4. USERCODE_FORMAT

The USERCODE_FORMAT option selects the format for the data field used to assign a value to USERCODE.

Table 7.14. USERCODE Option

Option Name	Setting(s)	Description
USERCODE_FORMAT	Binary	Sets USERCODE using 32 *1* or *0* characters.
	Hex (default)	Sets USERCODE using eight hexadecimal digits (0–9, A–F).
	ASCII	Sets USERCODE using up to four ASCII characters.

7.3.5. CUSTOM_IDCODE

The MachXO4 device feature row contains a 32-bit register for storing a custom identification code. This register can be initialized with any 32-bit value specified through CUSTOM_IDCODE. The CUSTOM_IDCODE option is only active when the MY_ASSP option is set to ON. Refer to the MY_ASSP section for more information about how to assign a value to CUSTOM_IDCODE.

Table 7.15. CUSTOM_IDCODE Option

Option Name	Setting(s)	Description
CUSTOM_IDCODE	0x00000000 (default)	32-bit custom identification code to be stored in a 32-bit register in the FPGA device feature row. The data format is set using the CUSTOM_IDCODE_FORMAT option.

7.3.6. CUSTOM_IDCODE_FORMAT

The CUSTOM_IDCODE_FORMAT option selects the format for the data field used to assign a value to CUSTOM_IDCODE.

Table 7.16. CUSTOM_IDCODE_FORMAT Option

Option Name	Setting(s)	Description
CUSTOM_IDCODE _FORMAT	Binary	Sets CUSTOM_IDCODE using 32 *1* or *0* characters.
	Hex (default)	Sets CUSTOM_IDCODE using eight hexadecimal digits (0–9, A–F).

7.3.7. SHAREDEBRINIT

The SHAREDEBRINIT option allows one copy of a unique memory initialization file to be stored in the internal flash memory that can be shared among multiple EBRs. This reduces the bitstream size of the design and saves internal memory space for other applications.

Table 7.17. SHAREDEBRINIT Option

Option Name	Setting(s)	Description
SHAREDEBRINIT	DISABLE (default)	No memory initialization file is stored in internal flash memory.
	ENABLE	Allows a memory initialization file to be stored in internal flash memory.

7.3.8. MUX_CONFIGURATION_PORTS

The MUX_CONFIGURATION_PORTS option is used to confirm the disabling of all device configuration ports so that the configuration port I/O pins can be used as general purpose I/O pins. The MUX_CONFIGURATION_PORT option is only active when all configuration ports are set to DISABLE.

Table 7.18. MUX_CONFIGURATION_PORTS Option

Option Name	Setting(s)	Description
MUX_CONFIGURATION _PORTS	DISABLE (default)	If all configuration ports are disabled, Radiant build tools generate an error.
	ENABLE	Confirms the disabling of all configuration ports (Radiant build tools will honor the removal of all configuration ports) and activates the JTAGENB input pin, thereby permitting the JTAGENB pin to be multiplexed. If the JTAGENB pin is hard connected to GND on the PCB, the device becomes a *write one time* device. Alternatively, you can control the JTAGENB pin to dynamically switch the I/O pins between JTAG pins and general purpose I/O pins.

7.3.9. DUALBOOTGOLDEN

The DUALBOOTGOLDEN option specifies the location of the golden (fail-safe) image in the dual-boot configuration mode.

Table 7.19. DUALBOOTGOLDEN Option

Option Name	Setting(s)	Description
DUALBOOTGOLDEN	INTERNAL (default)	The device attempts to configure from the external SPI storage device first through the controller SPI mode. If configuration fails, the device attempts to configure from the golden image in the internal flash memory through SDM.
	EXTERNAL	The device attempts to configure from the internal flash memory first through SDM. If configuration fails, the device attempts to configure from the golden image in the external SPI storage device through the controller SPI mode. **Note:** The DUALBOOTGOLDEN setting is in the feature row inside the JEDEC file. Therefore, the JEDEC file must be programmed into the internal flash for the setting to apply.

7.4. Security Options

The security options allow you to track or secure the MachXO4 device.

Table 7.20 shows the security options.

Table 7.20. Security Options

Option Name	Default Setting	All Settings
TRACEID	8b'0	8-bit unique value for TraceID
MY_ASSP	OFF	OFF, ON
CONFIG_SECURE	OFF	OFF, ON
ONE_TIME_PROGRAM	OFF	OFF, NVMEM, NVMEM_SRAM
BACKGROUND_ RECONFIG	OFF	OFF, ON

7.4.1. TRACEID

The MachXO4 device supports a feature called TraceID. TraceID stamps each MachXO4 device with a unique 64-bit ID. Every device has a unique TraceID value even if it shares the same configuration data with other devices. This differs from USERCODE in the configuration data where every device that receives the same configuration data has the same USERCODE value. For more information about the TraceID feature, refer to Using TraceID (FPGA-TN-02084).

Table 7.21. TRACEID Option

Option Name	Setting(s)	Description
TRACEID	8b'0 (default)	A unique 8-bit user-defined value that is stored in the feature row and forms part of the 64-bit TraceID. The 56 least significant bits (LSBs) of the TraceID are immutable representing a combination of wafer lot, wafer number, and the XY coordinates of the die on the wafer. The eight most significant bits (MSBs) of the TraceID are from the unique 8-bit user-defined value. The TRACEID setting can be changed in the Lattice Radiant Device Constraint Editor.

7.4.2. MY_ASSP

Every Lattice device has its own identification code identifying the device family, device density, and other parameters (for example, voltage, device stepping, and others). The code is accessible from any MachXO4 device configuration port. The value stored in the IDCODE register allows you to uniquely identify a Lattice device.

The MY_ASSP option allows you to change the value returned when IDCODE is read from the FPGA. Setting MY_ASSP to ON essentially overrides the default IDCODE of the MachXO4 device. Overriding IDCODE prevents the Lattice programming software from identifying the device and consequently from directly programming the device. You must generate serial vector format (SVF) files to program MY_ASSP-enabled MachXO4 devices.

Table 7.22. MY_ASSP Option

Option Name	Setting(s)	Description
MY_ASSP	OFF (default)	Reading identification code returns IDCODE
	ON	Reading identification code returns CUSTOM_IDCODE. The MY_ASSP option must be set to ON to enable the CUSTOM_IDCODE option.

7.4.3. CONFIG_SECURE

The CONFIG_SECURE option allows you to block read-back of the SRAM and internal flash memory.

Table 7.23. CONFIG_SECURE Option

Option Name	Setting(s)	Description
CONFIG_SECURE	OFF (default)	Allows read-back of the SRAM and internal flash memory.
	ON	Blocks read-back of the SRAM and internal flash memory through any configuration port. In addition, the device cannot be programmed without erasing. **Note**: The device must be erased to reset the security setting. The CONFIG_SECURE setting and internal flash are erased in tandem. Once security settings are reset, the device can be programmed again.

7.4.4. ONE_TIME_PROGRAM

The MachXO4 (LFMXO4) device has one-time programmable (OTP) security settings that can be used to prevent on-chip memory (SRAM and internal flash memory) from being erased or programmed. The ONE_TIME_PROGRAM option allows you to set the OTP security settings for the SRAM and internal flash memory sectors.

Table 7.24. ONE_TIME_PROGRAM Option

Option Name	Setting(s)	Description
ONE_TIME_ PROGRAM	OFF (default)	On-chip memory (SRAM and internal flash memory) can be erased or programmed.
	NVMEM	Internal flash memory (non-volatile memory) cannot be erased or programmed. The configuration data in the internal flash memory is prevented from further modification. The device can still be configured from the internal flash memory in SDM or an external SPI storage device in controller SPI or dual-boot configuration mode.
	NVMEM_SRAM	Internal flash memory (non-volatile memory) and SRAM cannot be erased or programmed. The configuration data in the internal flash memory is prevented from further modification. The SRAM cannot be modified through the JTAG port. The device can still be configured from the internal flash in SDM or an external SPI storage device in controller SPI or dual-boot configuration mode.

7.4.5. BACKGROUND_RECONFIG

The BACKGROUND_RECONFIG option specifies the behavior of SRAM reconfiguration in relation to the PROGRAMN pin (when enabled) and the sysCONFIG REFRESH command.

Table 7.25. BACKGROUND_RECONFIG Option

Option Name	Setting(s)	Description
BACKGROUND_ RECONFIG	OFF (default)	Pulsing PROGRAMN or transmitting the REFRESH command initiates standard *warm-boot* SRAM reconfiguration from the specified configuration image (internal or external) or images (dual boot). The warm-boot sequence executes initialization, configuration, and wake-up processes before the device re-enters user mode (refer to the Configuration Process and Flow section).
	ON	Pulsing PROGRAMN or transmitting the REFRESH command initiates *background* SRAM reconfiguration from the specified configuration image (internal or external) or images (dual boot). The background sequence executes only the configuration process. The initialization and wake-up processes are bypassed. The device remains in user mode throughout the sequence. This is typically used in conjunction with SED to support SEC. The device operates without disruption. Only erroneous SRAM cells are corrected. Note that when the ENABLE_TRANSFR option is set to ENABLE, BACKGROUND_RECONFIG must be set to ON. For the MachXO4 (LFMXO4) device, only SRAM cells are updated during background reconfiguration.

8. Advanced Configuration Information

8.1. Flash Programming

The MachXO4 device internal flash memory is a central component of the FPGA configuration system. It allows you to store the FPGA configuration data as well as design specific data using precise erase and programming sequences. Lattice provides several methods for programming the MachXO4 device internal flash:

- JTAG or target SPI programming
- VMEmbedded – C source code for use with an embedded microprocessor controlling the JTAG port
- SSPIEmbedded – C source code for use with an embedded microprocessor controlling the target SPI port
- Custom – Using information from MachXO4 Hardened Control Functions Software Reference Guide (FPGA-TN-02404), you can also create a custom solution.

The flash space can be accessed by the JTAG, I2C, and SPI ports. These configuration ports can use offline or transparent programming modes to erase, program, and verify the MachXO4 device 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 mostly identical across all configuration ports. Slight differences exist due to the difference in communication protocol standards.

The internal flash memory size in a MachXO4 device depends on the device density. Refer to the UFM Resources in MachXO4 Devices and Configuration Flash Resources in MachXO4 Devices tables in the MachXO4 Hardened Control Functions User Guide (FPGA-TN-02403) for the number of internal memory pages available for each MachXO4 device density. Each internal memory page represents 128 bits of data.

8.2. JEDEC File Format

All Lattice FPGAs with non-volatile memory support the JEDEC file format. Utilities are available in the Deployment Tool software for converting a JEDEC file into other programming file formats, such as STAPL, SVF, or bitstream (BIT or HEX file). Table 8.1 shows relevant details about the JEDEC file for the MachXO4 device.

Table 8.1. MachXO4 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. 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 keyword N marks the beginning of a comment. It can appear anywhere in the JEDEC file. Lattice JEDEC files add *OTE* to the N keyword to improve readability.
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 fuses not included in the link field. F0 = fill with zeros (0), F1 = fill with ones (1). It is defined for the purpose of reducing JEDEC file size and has no meaning in the 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>.
OTP and Security Setting	G0, G1, G2, or G3	Lattice extension of G field to cover OTP fuse programming. G<0=both no, 1=only security yes, 2=only OTP yes, 3=both yes>.

JEDEC Field	Syntax	Description
Link Field	L0000000 101011…100011 …… 111111…101100 110 101011…100011 …… 111111…101100 110 …….. …….. 101011…100011 …… 111111…101100 110* NOTE SED_CRC* L3627704 111111….111111* CC1B9	The keyword L identifies the first fuse address of the fuse pattern, which follows after the white space. The number of digits shown following the L keyword must be the same as that of the QF field. In this example, QF3627736 has seven digits so there are seven zeroes following L, that is L0000000. 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 physical fuse array layout of the silicon. 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 verified. 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. For MachXO4 (LFMXO4) device: UFM pages, if present in the JEDEC, are preceded by a *NOTE TAG DATA** line. If the JEDEC file is encrypted, all the data in the link field are encrypted. The column size increases accordingly to include filler bits to make the column size packet, 128 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 can be found from the link field. If a fuse state is not specified in the link field, then use the default fuse state. If the JEDEC file is encrypted, the fuse checksum is calculated after encryption. The fuse checksum prior to encryption can be found in one of the comments.
U Field	UA Home	This field stores the 32-bit USERCODE. The 32-bit USERCODE can be expressed in UA = ASCII, UH = ASCII Hex, or U = Binary. This field is Lattice-enhanced for storing the CRC value of encrypted JEDEC.
E Field	EH 012..ABCDEF	JEDEC standard defines this field to hold the architecture fuses. Lattice uses this field to store the feature row and FEABITS. Feature row data is on the first line. FEABITS values are on line 2.
End-of-text	^C	^C (CTLC) marks the end 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 ^B and ^C, are included in the checksum calculation.

Figure 8.1 and Figure 8.2 show an example of a MachXO4 device JEDEC file.

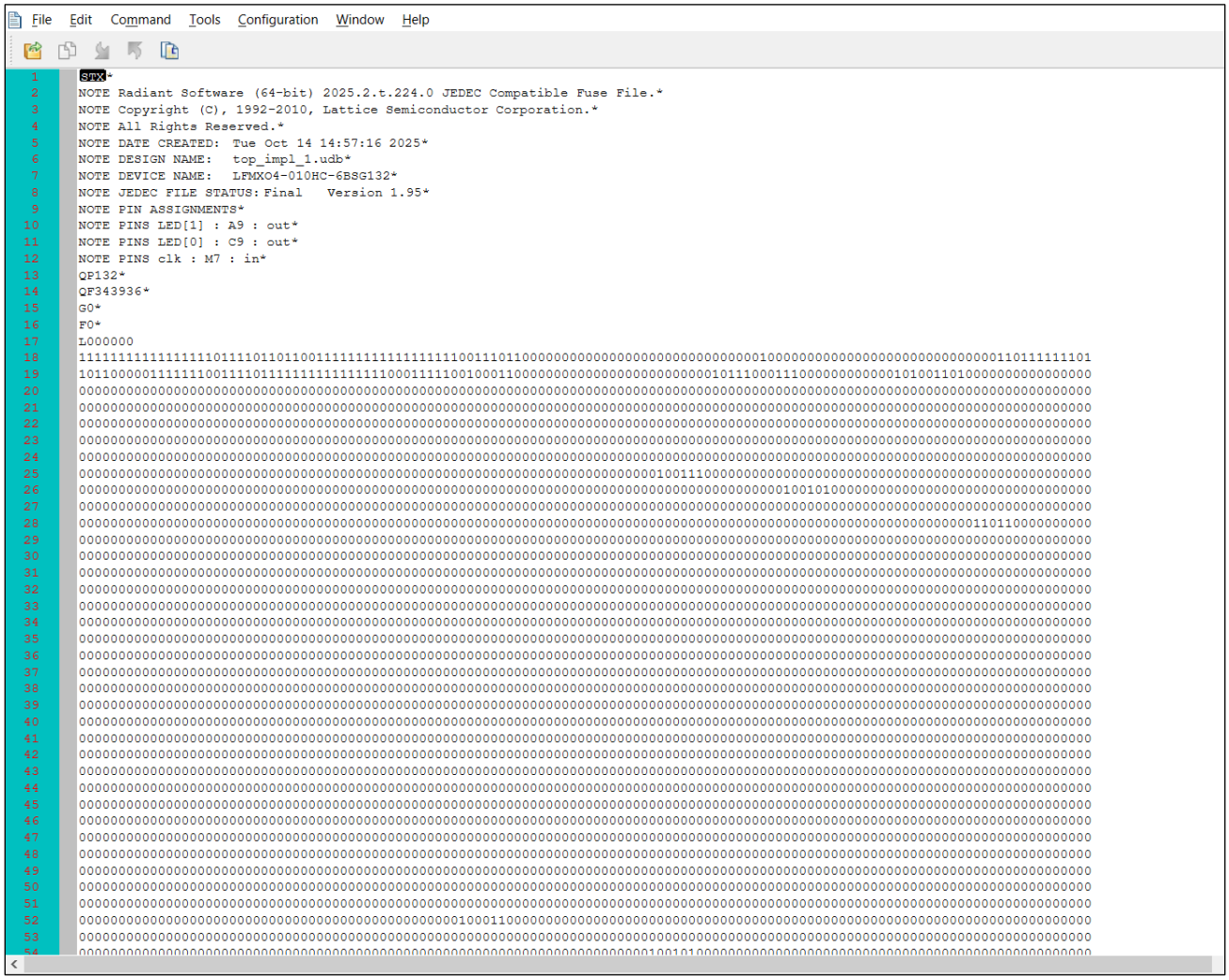


Figure 8.1. JEDEC File Example (Part 1)

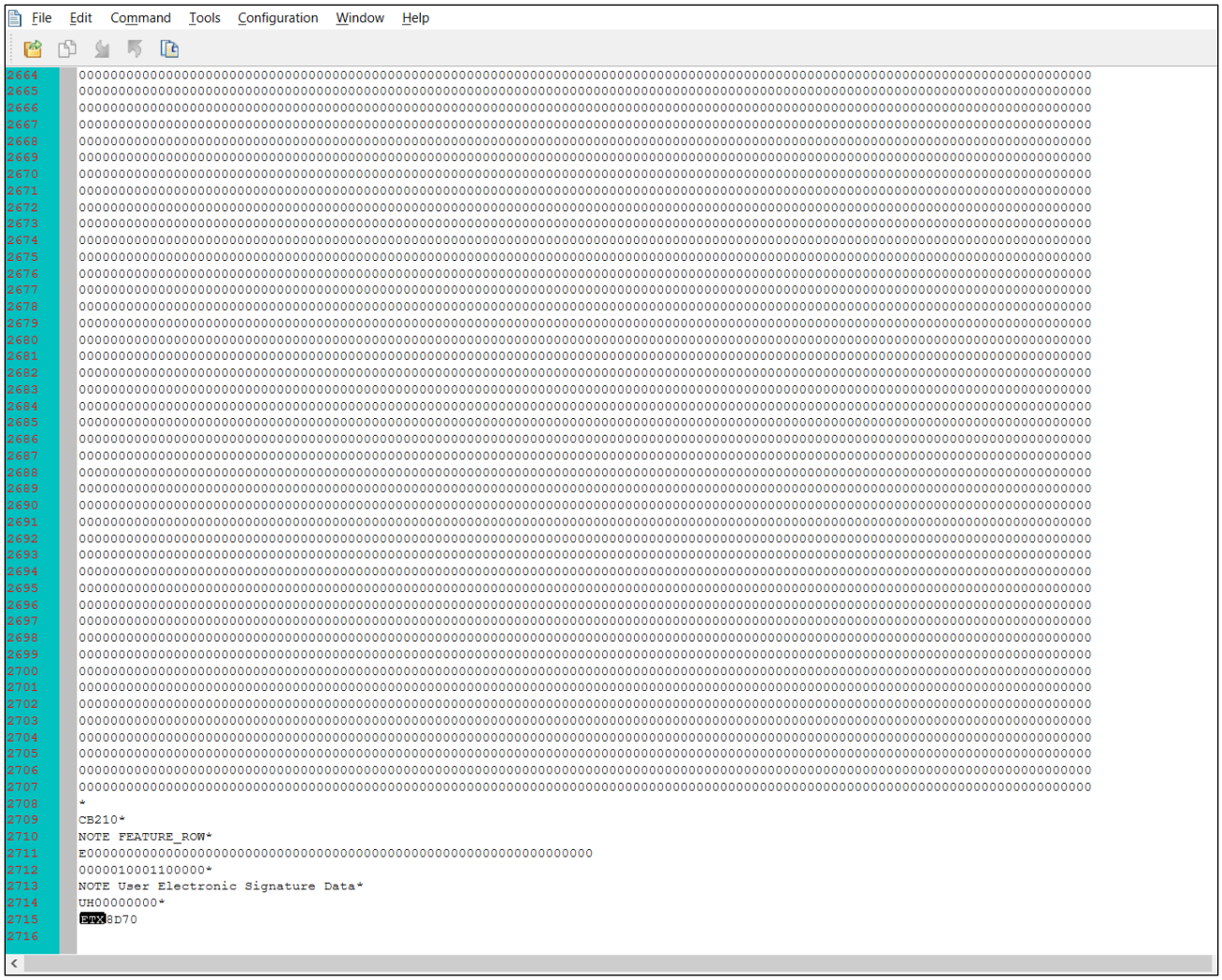


Figure 8.2. JEDEC File Example (Part 2)

8.3. Flash Programming Flow

The MachXO4 device flash memory erase and programming require a specific set of steps and timing. The flow chart in this section describes the command sequences and timing required for successful flash programming. The command sequences and timing are mostly identical between all configuration ports. However, there are minor variations in the command sequences based on the configuration port used. Any exceptions to the common flow are covered in a separate configuration port-specific section.

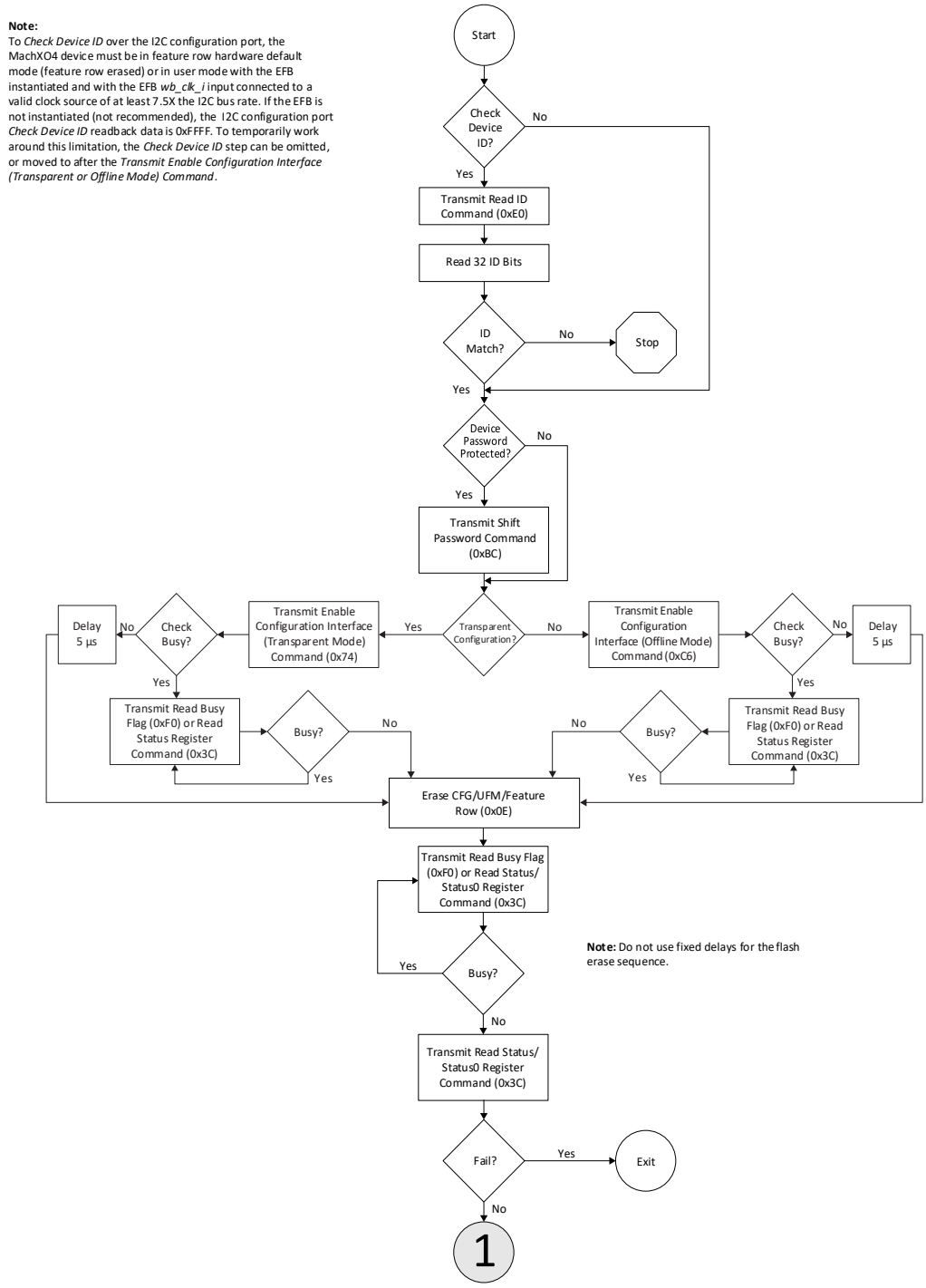


Figure 8.3. MachXO4 Device Common Flash Programming Flow (Part 1)

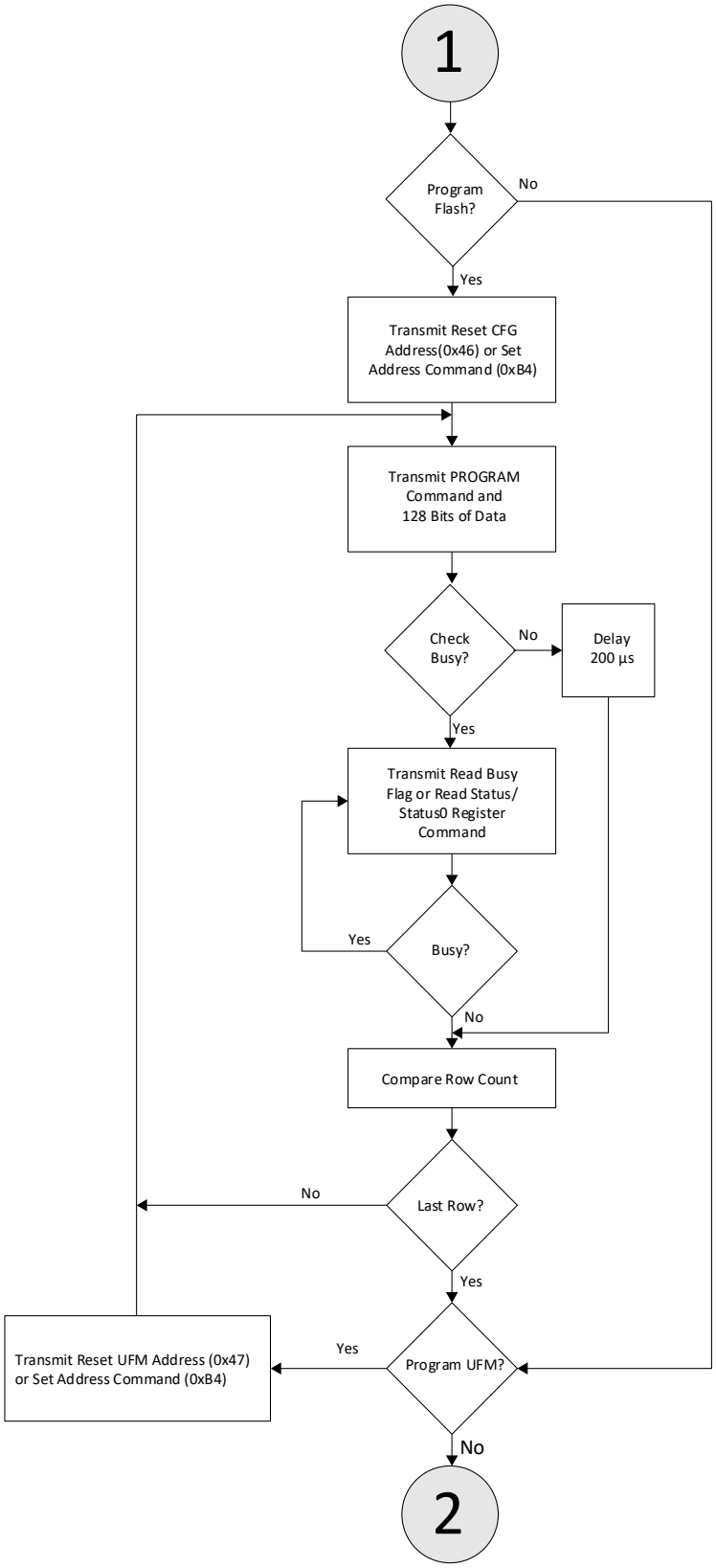


Figure 8.4. MachXO4 Device Common Flash Programming Flow (Part 2)

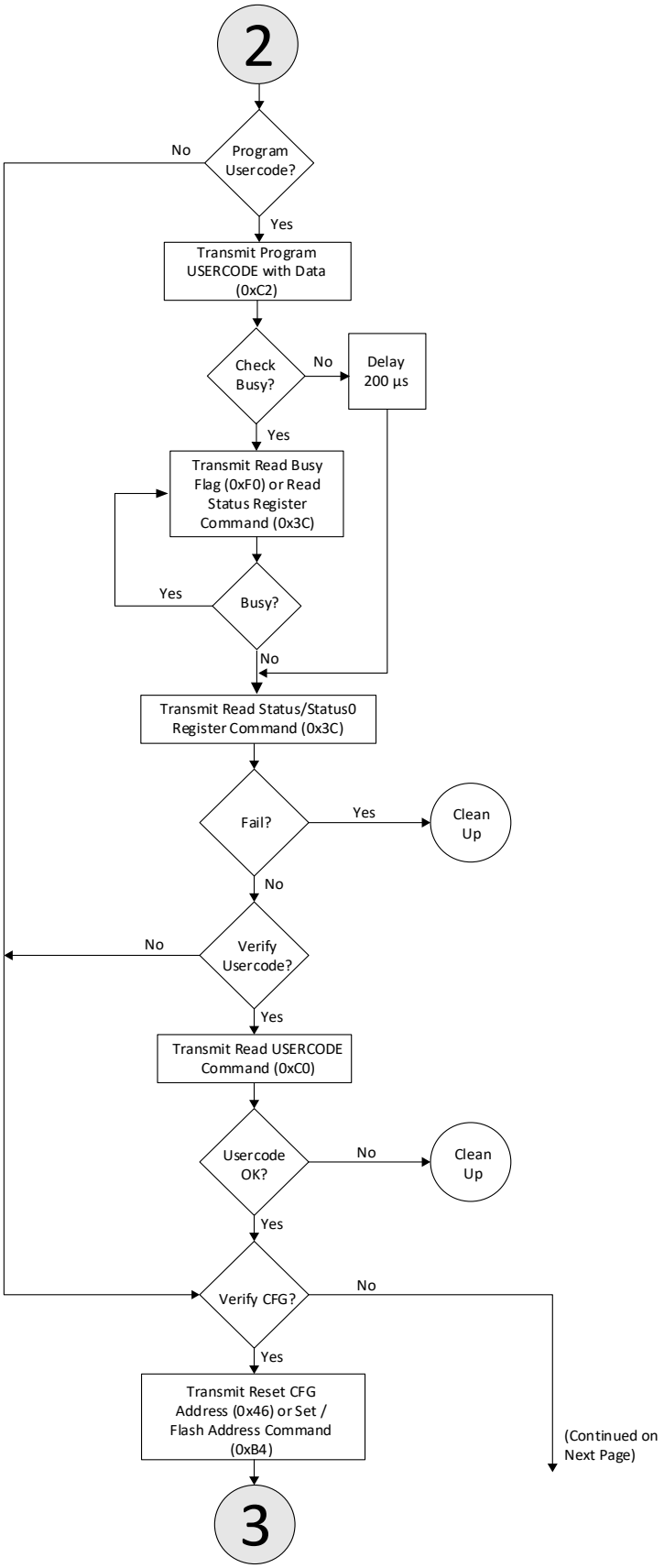


Figure 8.5. MachXO4 Device Common Flash Programming Flow (Part 3)

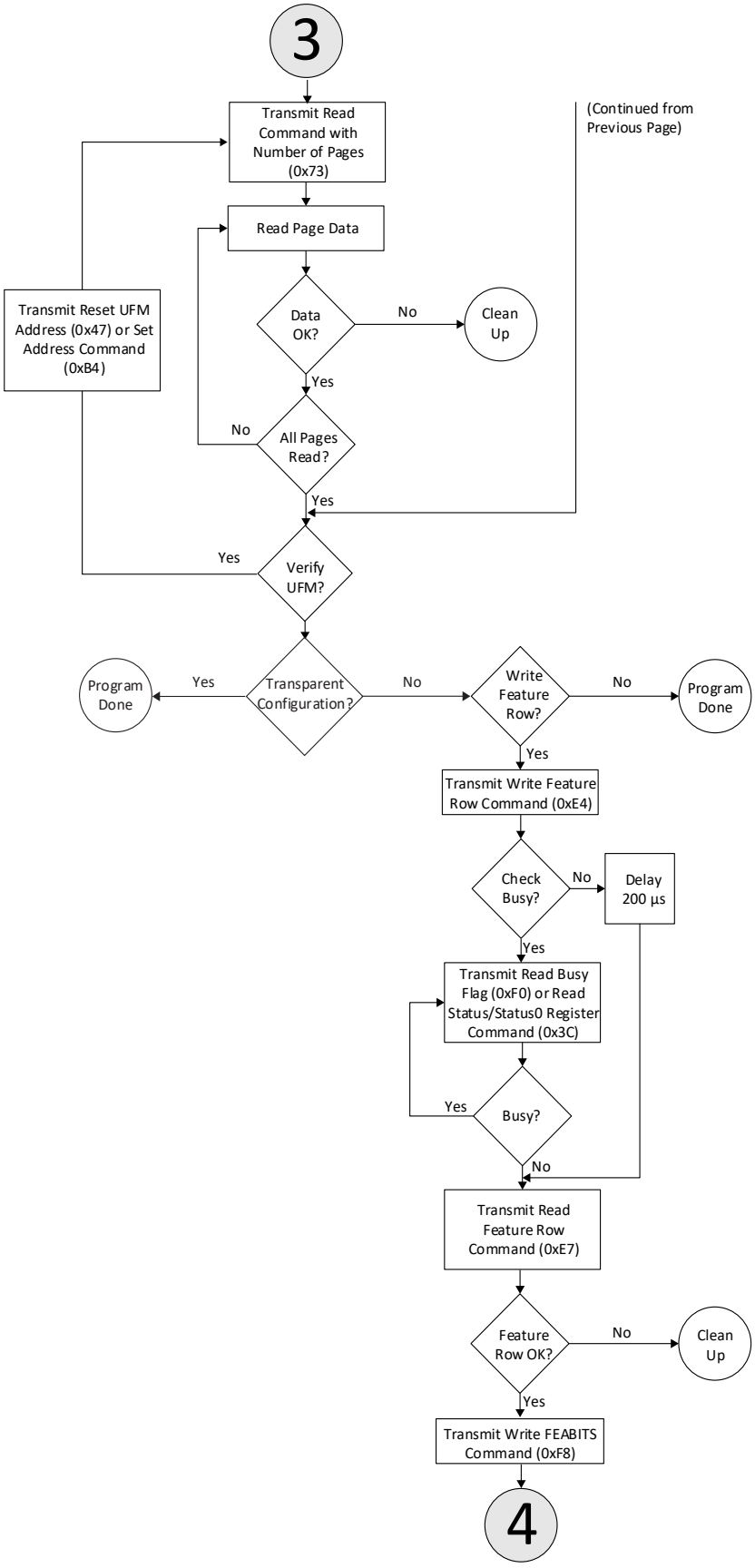


Figure 8.6. MachXO4 Device Common Flash Programming Flow (Part 4)

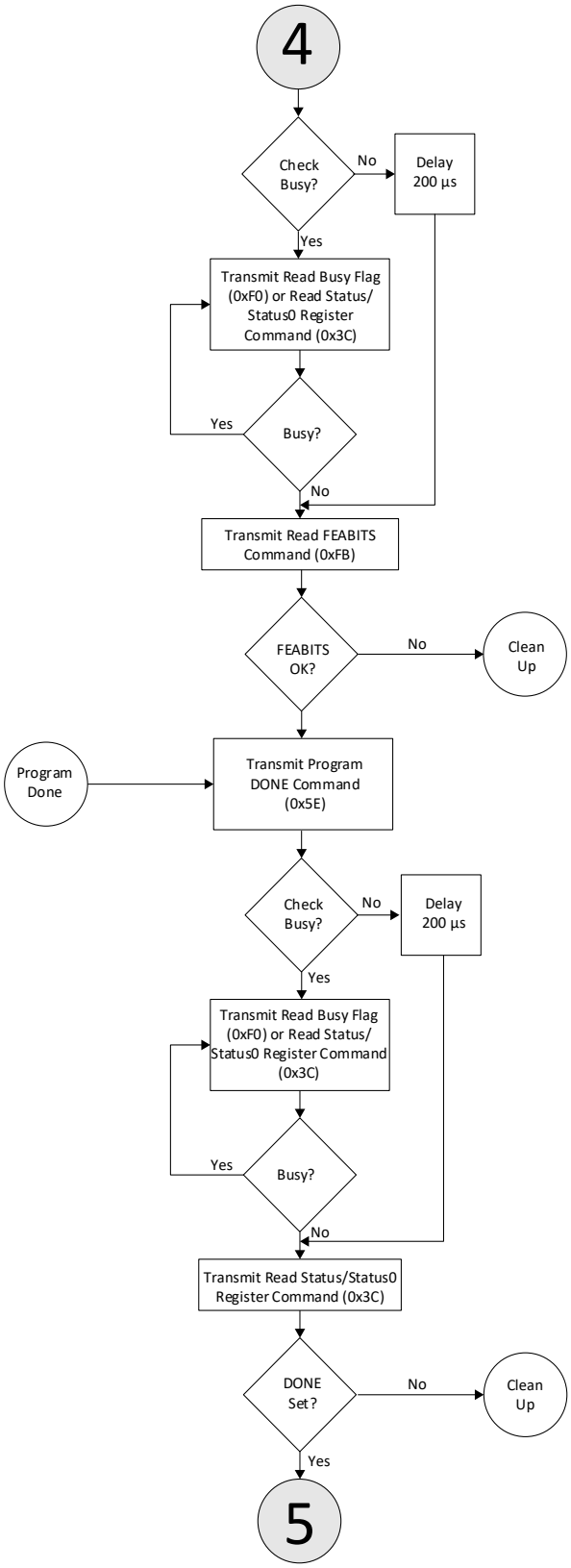


Figure 8.7. MachXO4 Device Common Flash Programming Flow (Part 5)

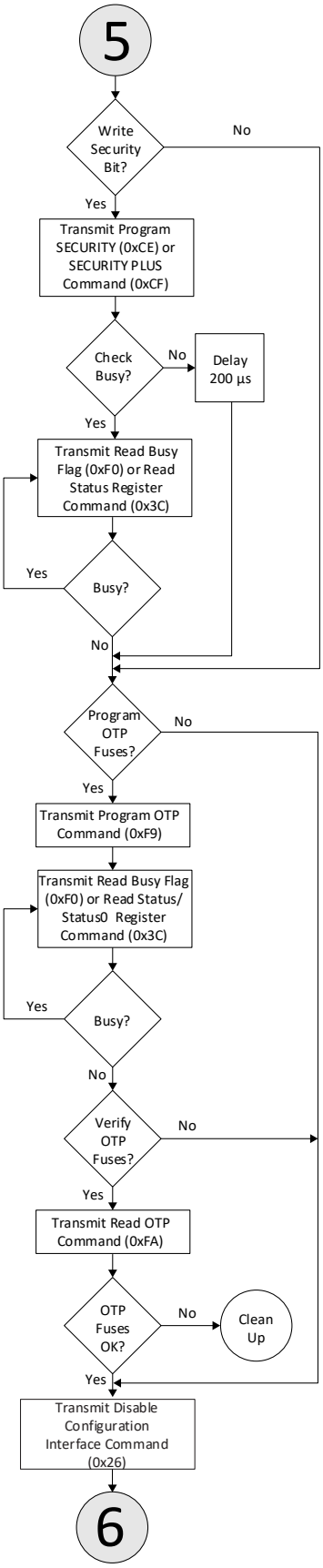


Figure 8.8. MachXO4 Device Common Flash Programming Flow (Part 6)

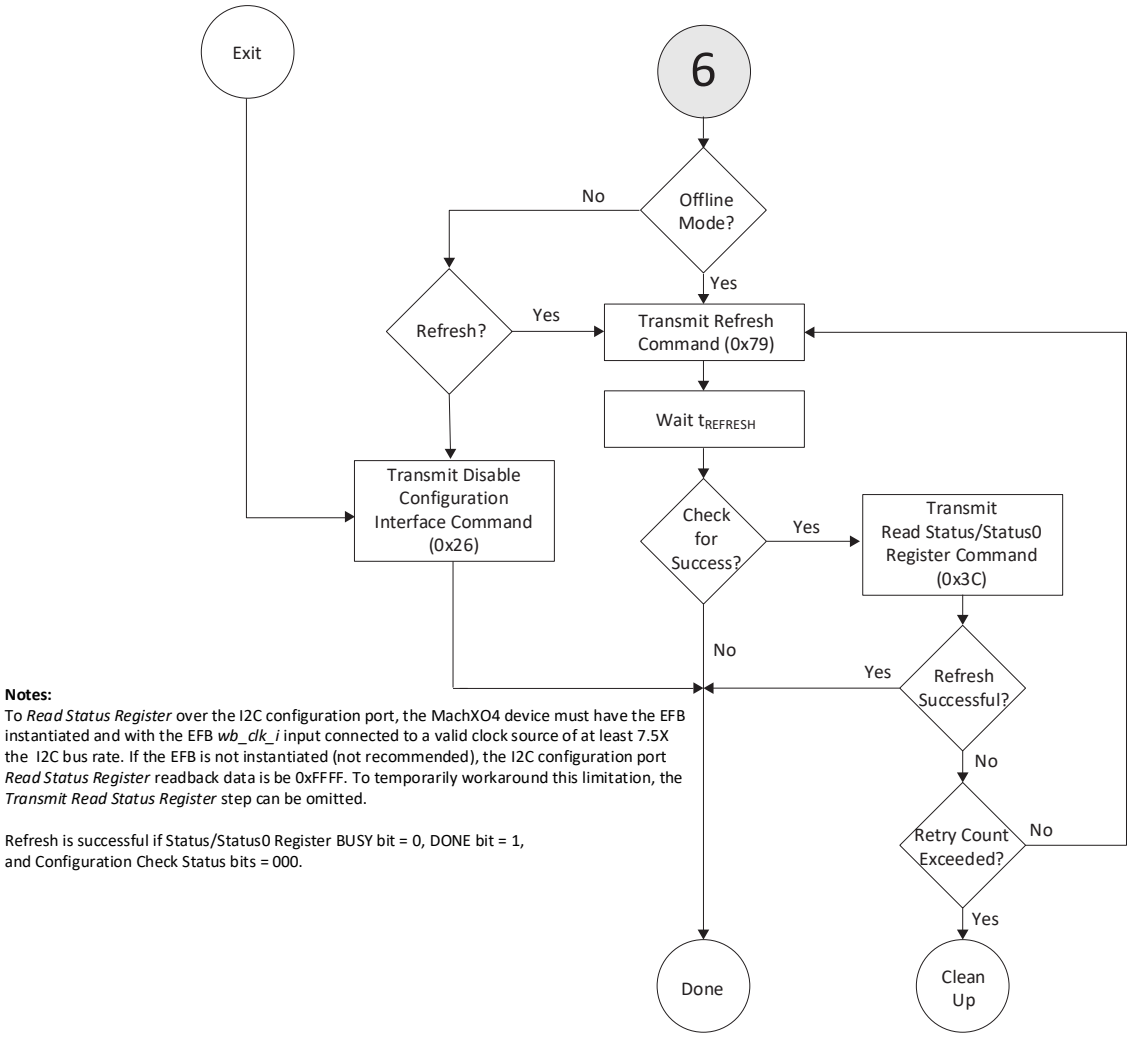


Figure 8.9. MachXO4 Device Common Flash Programming Flow (Part 7)

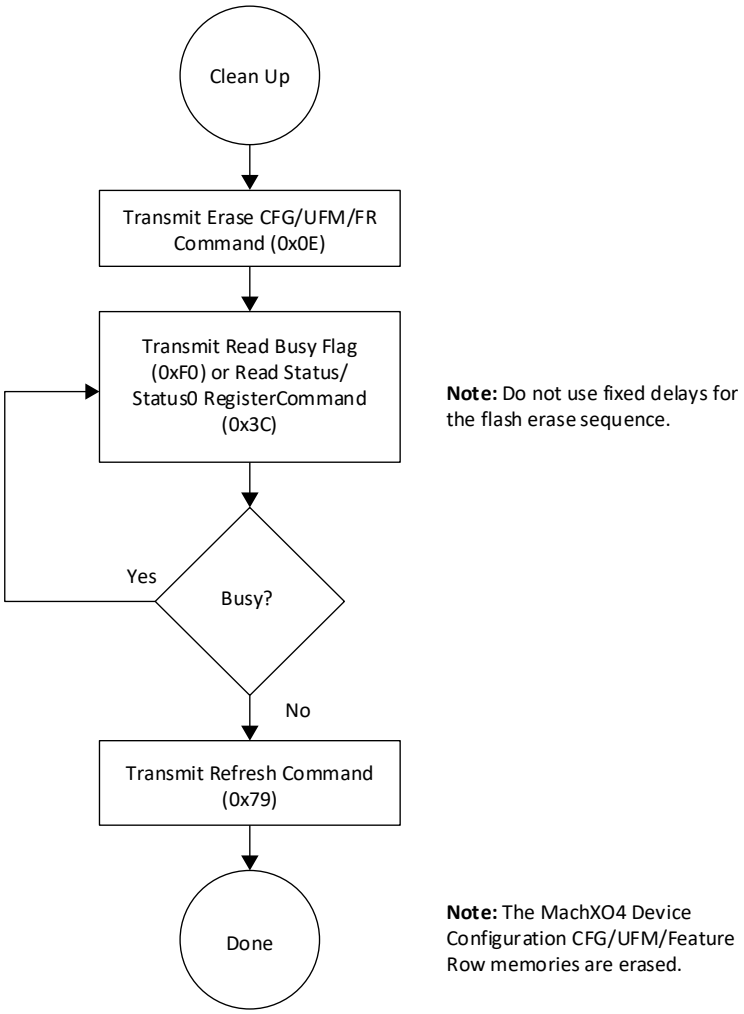


Figure 8.10. MachXO4 Device Common Flash Programming Flow (Part 8)

8.4. Target SPI or I2C SRAM Configuration Flow

The MachXO4 device target SPI or 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 target SPI or I2C SRAM configuration.

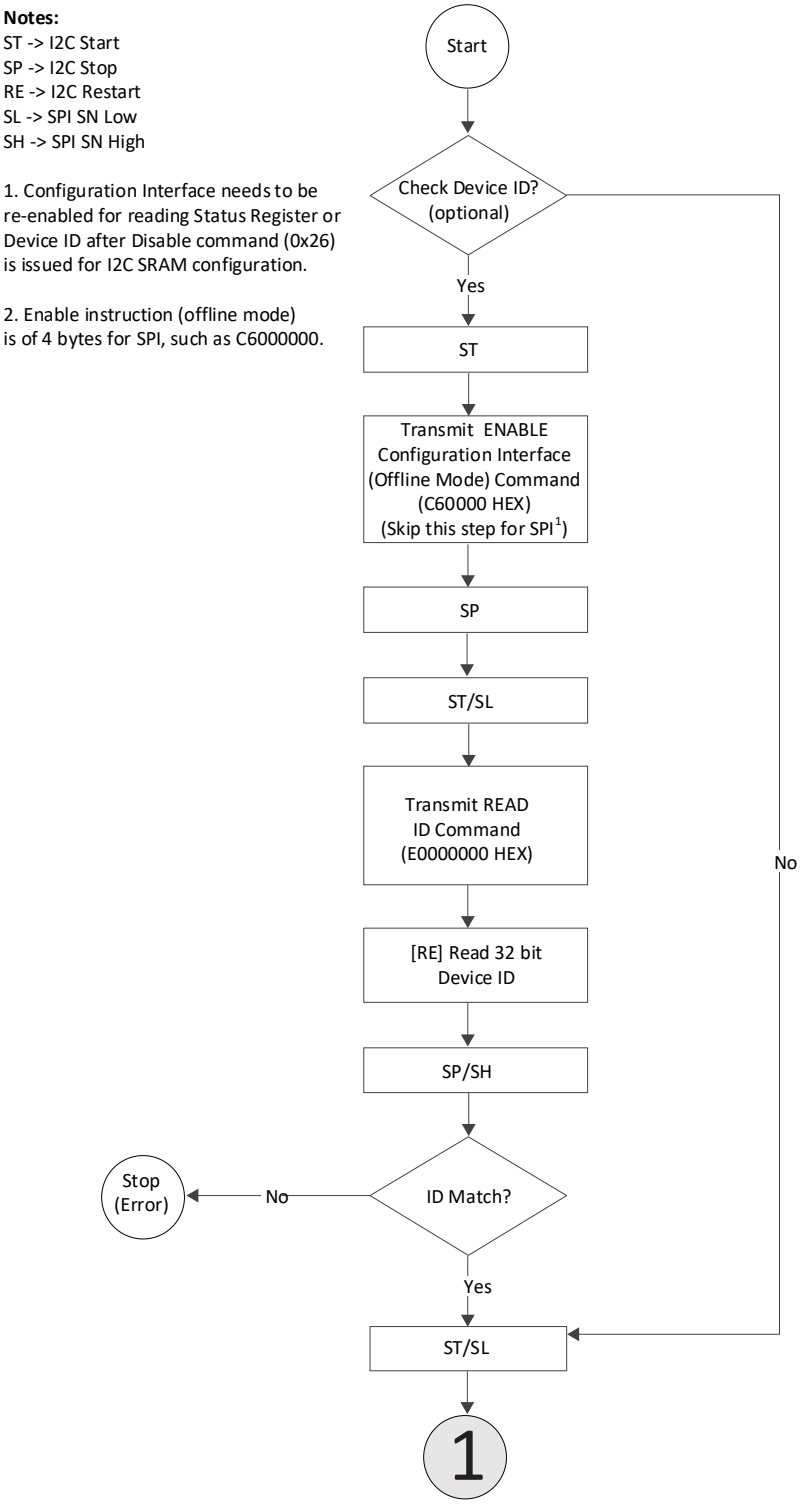


Figure 8.11. MachXO4 Device Target SPI or I2C SRAM Configuration Flow (Part 1)

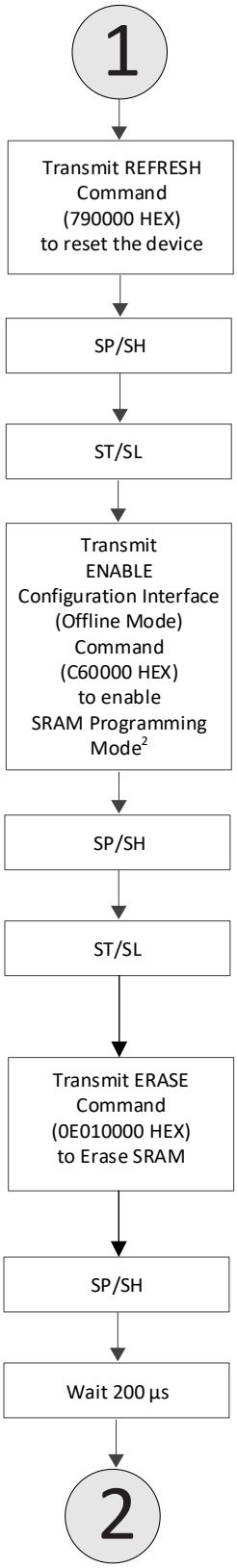


Figure 8.12. MachXO4 Device Target SPI or I2C SRAM Configuration Flow (Part 2)

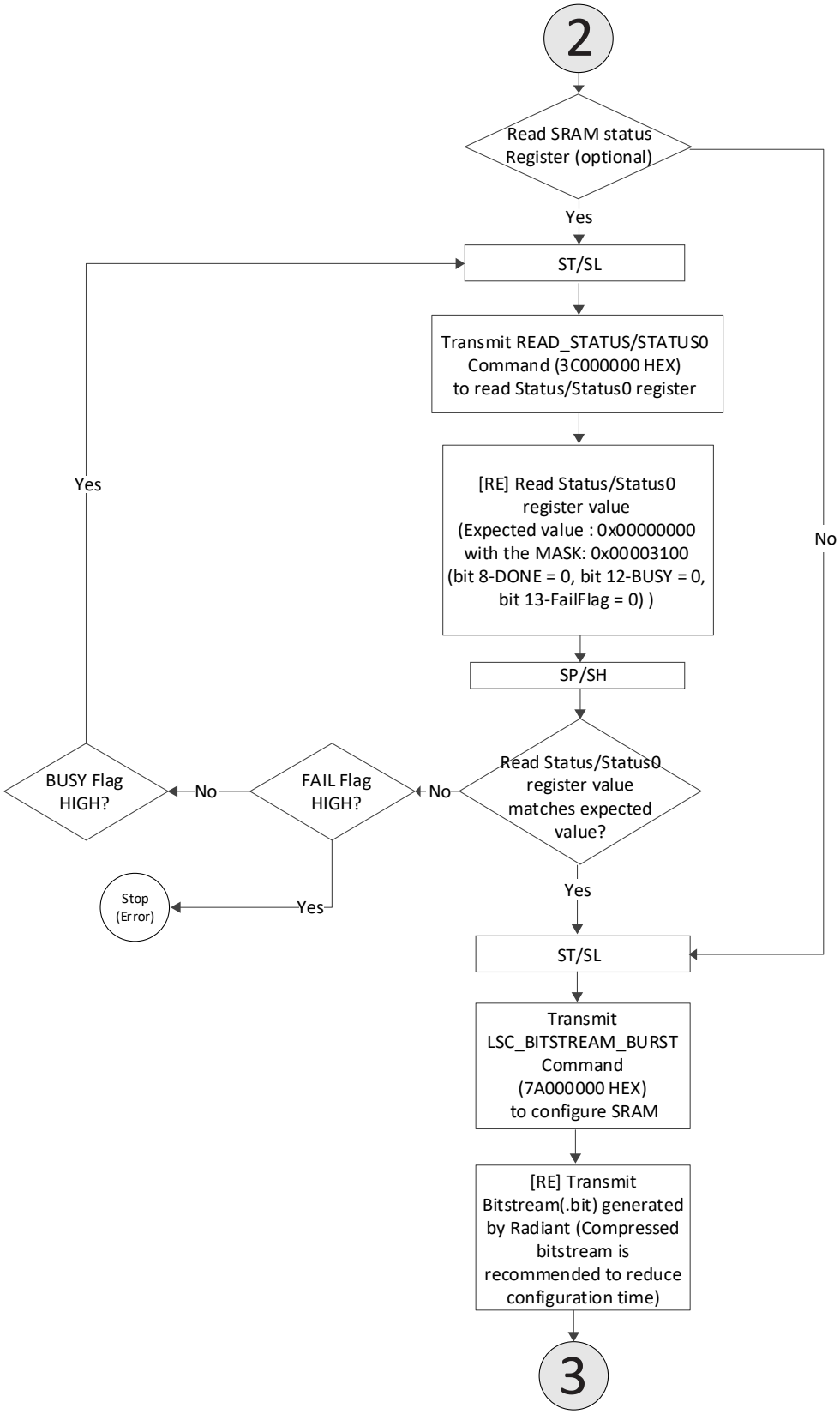


Figure 8.13. MachXO4 Device Target SPI or I2C SRAM Configuration Flow (Part 3)

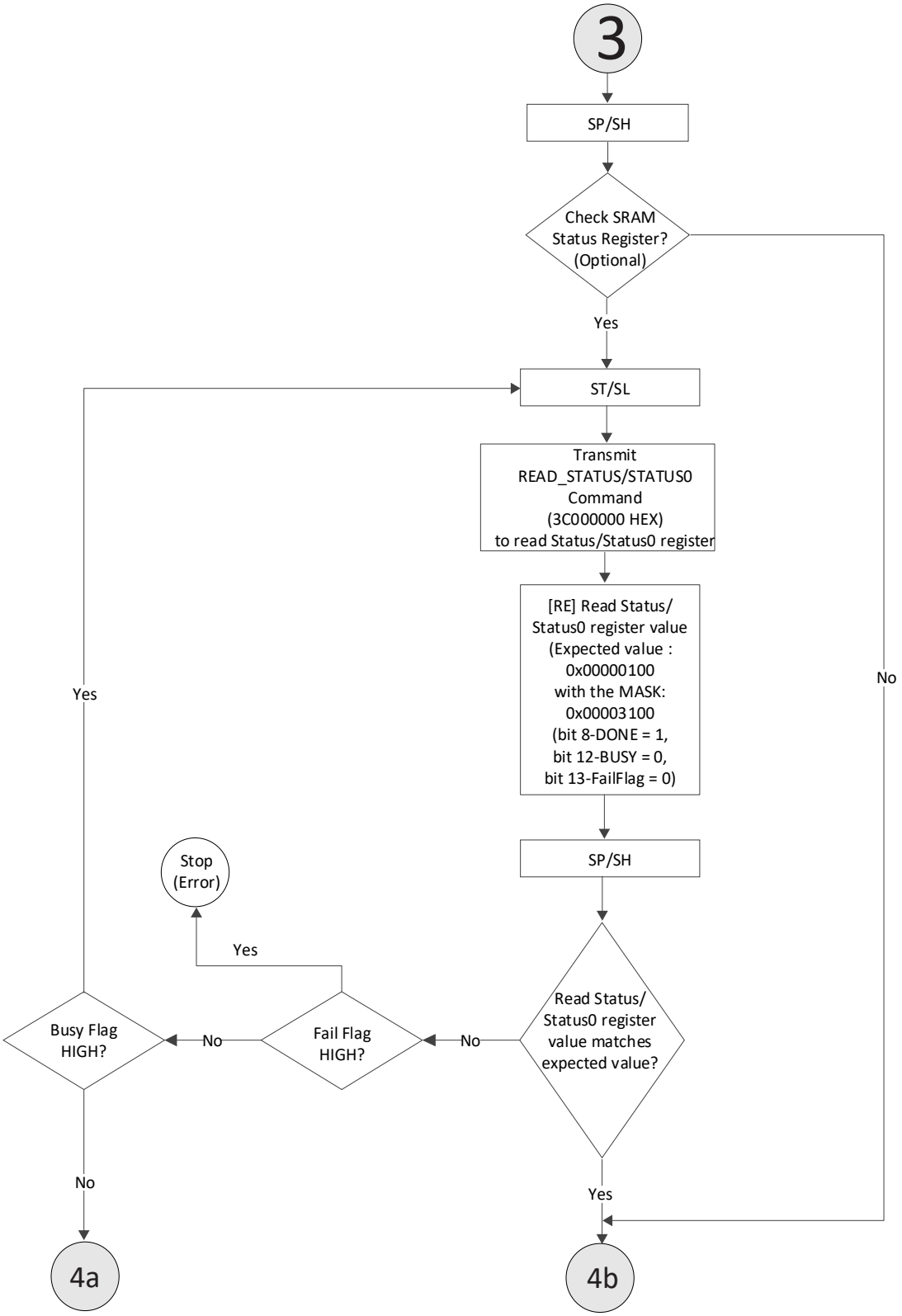


Figure 8.14. MachXO4 Device Target SPI or I2C SRAM Configuration Flow (Part 4)

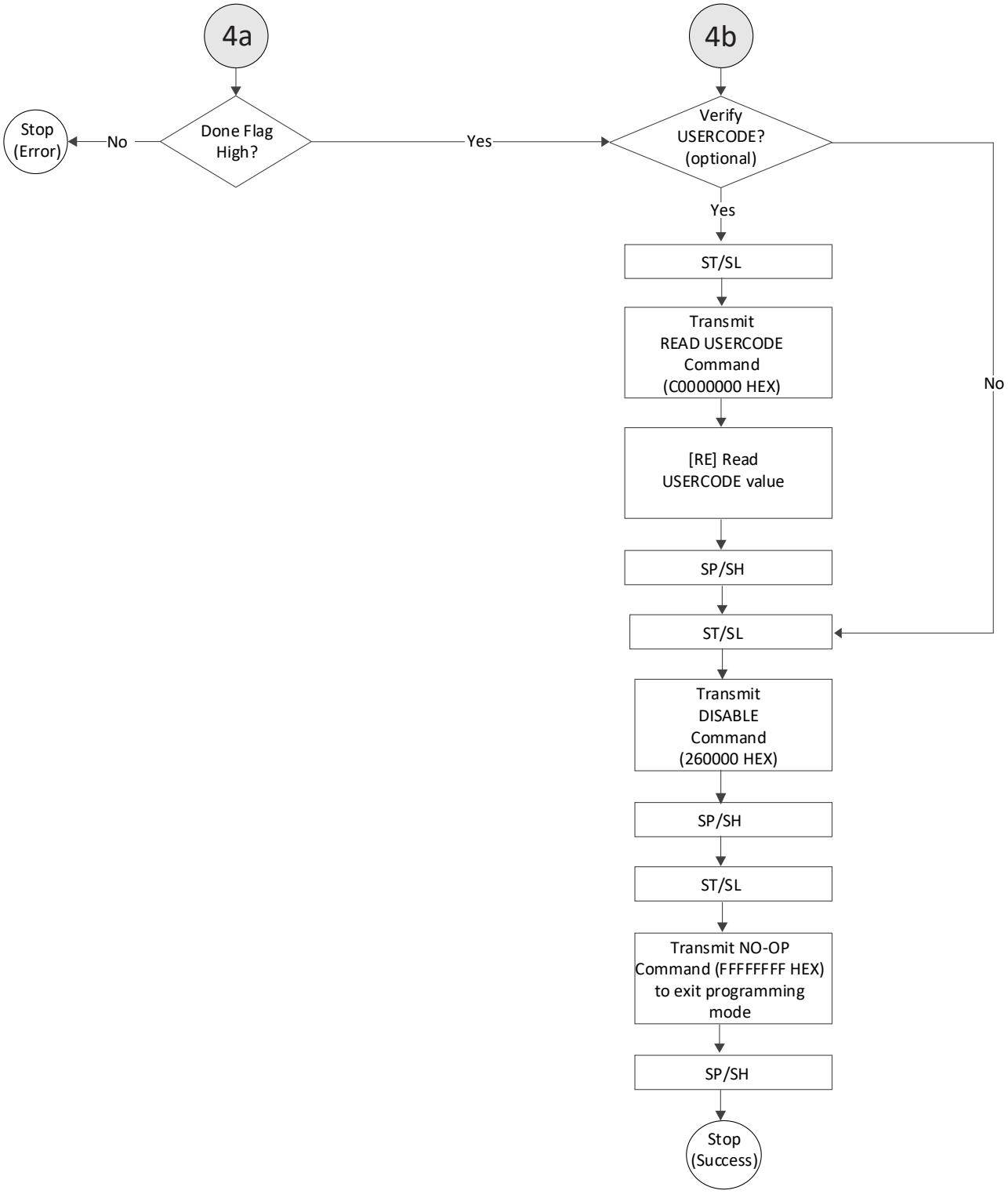


Figure 8.15. MachXO4 Device Target SPI or I2C SRAM Configuration Flow (Part 5)

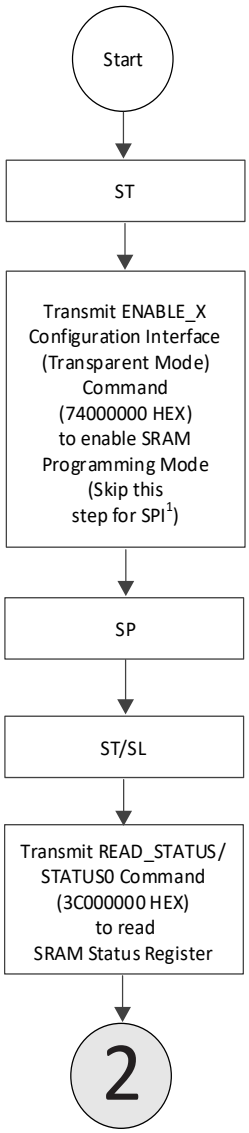


Figure 8.16. MachXO4 Device Target SPI or I2C SRAM Read Status Register Flow (Part 1)

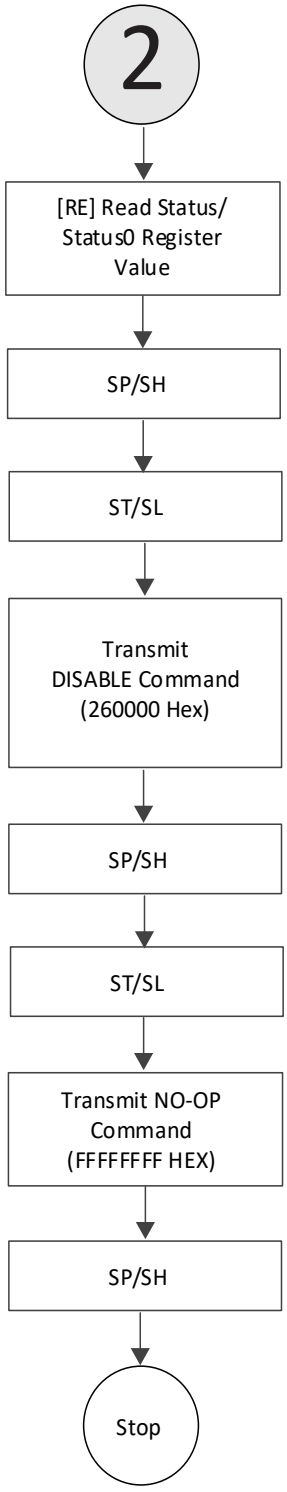


Figure 8.17. MachXO4 Device Target SPI or I2C SRAM Read Status Register Flow (Part 2)

8.5. Programming Commands

Table 8.2. MachXO4 Device sysCONFIG Programming Commands

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Description
Read Device ID [IDCODE_PUB]	0xE0	00 00 00	—	YY YY YY YY	YY characters represent the device-specific ID code.
Enable Configuration Interface (Transparent Mode) [ISC_ENABLE_X]	0x74	0Y 00 00	—	—	Enables the configuration logic for transparent SRAM read access or transparent flash programming access. Y identifies the memory type to access. Bit 1 0 19 Flash SRAM **Note:** When using the I2C port and target SPI port, transmit the command opcode and first two operand bytes. The final operand byte must not be transmitted.
Enable Configuration Interface (Offline Mode) [ISC_ENABLE]	0xC6	08 00 00	—	—	Enables the configuration logic for device programming in Offline programming mode. **Note:** When using the I2C port and target SPI port, transmit the command opcode and first two operand bytes. The final operand byte must not be transmitted.
Read Busy Flag [LSC_CHECK_BUSY]	0xF0	00 00 00	—	YY	Bit 1 0 7 Busy Ready
Read Status Register [LSC_READ_STATUS]	0x3C	00 00 00	—	YY YY YY YY	Bit 1 0 12 Busy Ready 13 Fail OK
Erase [ISC_ERASE]	0x0E	0Y 00 00	—	—	Erases the specified internal memory space. Y identifies the memory space to be erased in flash access mode. Bit 1 = Enable, 0 = Disable 16 Erase SRAM 17 Erase feature row 18 Erase CFG 19 Erase UFM
Erase UFM [LSC_ERASE_TAG]	0xCB	00 00 00	—	—	Erases the UFM sector only.
Reset Configuration CFG Address [LSC_INIT_ADDRESS]	0x46	00 00 00	—	—	Sets page address pointer to the beginning of the configuration flash sector.
Set Address [LSC_WRITE_ADDRESS]	0xB4	00 00 00	M0 00 PP PP	—	Sets the page address pointer to the flash page specified by the least significant 14 bits of the PP PP field. The *M* field defines the flash space to access. Field (M) Flash space selected 0x0 CFG 0x4 UFM
Program Page [LSC_PROG_INCR_NV]	0x70	00 00 01	YY * 16	—	Programs one flash page. Can be used to program the CFG or UFM.
Reset UFM Address [LSC_INIT_ADDR_UFM]	0x47	00 00 00	—	—	Sets the page address pointer to the beginning of the UFM sector.
Program UFM Page [LSC_PROG_TAG]	0xC9	00 00 01	YY * 16	—	Programs one UFM page.
Program USERCODE [ISC_PROGRAM_ USERCODE]	0xC2	00 00 00	YY * 4	—	Programs the USERCODE.
Read USERCODE [USERCODE]	0xC0	00 00 00	—	YY * 4	Retrieves the 32-bit USERCODE value.
Write Feature Row	0xE4	00 00 00	YY * 8	—	Programs the feature row bits.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Description
[LSC_PROG_FEATURE]					
Read Feature Row [LSC_READ_FEATURE]	0xE7	00 00 00	—	YY * 8	Retrieves the feature row bits.
Write FEABITS [LSC_PROG_FEABITS]	0xF8	00 00 00	YY * 2	—	Programs the FEABITS.
Read FEABITS [LSC_READ_FEABITS]	0xFB	00 00 00	—	YY * 2	Retrieves the FEABITS.
Read Flash [LSC_READ_INCR_NV]	0x73	M0 PP PP	—	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. Field (M) — Port selected 0x0 — I2C 0x0 or 0x1 — JTAG or target SPI
Read UFM [LSC_READ_UFM]	0xCA	M0 PP PP	—	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. Field (M) — Port selected 0x0 — I^2C 0x0 or 0x1 — JTAG or target SPI
Program DONE [ISC_PROGRAM_DONE]	0x5E	00 00 00	—	—	Programs the DONE status bit enabling SDM.
Program OTP Fuses [LSC_PROG_OTP]	0xF9	00 00 00	UCFSUCFS	—	Makes the selected memory space one-time programmable. Matching bits must be set in unison to activate the OTP feature. Bit — 1 — 0 0, 4 — SRAM OTP — SRAM Writable 1, 5 — Feature Row — Feature Row Writable 2, 6 — OTP — CFG Writable 3, 7 — CFG OTP UFM OTP — UFM Writable
Read OTP Fuses [LSC_READ_OTP]	0xFA	00 00 00	—	UCFSUCFS	Reads the state of the one-time programmable fuses. Bit — 1 — 0 0, 4 — SRAM OTP — SRAM Writable 1, 5 — Feature Row — Feature Row Writable 2, 6 — OTP — CFG Writable 3, 7 — CFG OTP UFM OTP — UFM Writable
Disable Configuration Interface [ISC_DISABLE]	0x26	00 00	—	—	Exits offline or transparent programming mode. ISC_DISABLE causes the MachXO4 device 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 8.10.
Bypass [ISC_NOOP]	0xFF	FF FF FF	—	—	No operation and device wakeup.

Command Name [SVF Synonym]	Command	Operands	Write Data	Read Data	Description
Refresh [LSC_REFRESH]	0x79	00 00	—	—	Forces the MachXO4 device to reconfigure. Transmitting a REFRESH command reconfigures the MachXO4 device similar to pulsing PROGRAMN.
Program SECURITY [ISC_PROGRAM_ SECURITY]	0xCE	00 00 00	—	—	Programs the security bit (secures CFG sector).[1]
Program SECURITY PLUS [ISC_PROGRAM_ SECPLUS]	0xCF	00 00 00	—	—	Programs the security plus bit (secures flash sectors).[1]
Read TraceID Code [UIDCODE_PUB]	0x19	00 00 00	—	YY*8	Reads the 64-bit TraceID.
Configure SRAM [LSC_BITSTREAM_ BURST]	0x7A	00 00 00	Com-pressed bit-stream	—	Shifts in the bitstream (BIT file) generated by the Lattice Radiant software. Use of compressed bitstream is recommended to reduce configuration time. The number of bits varies depending on the compression ratio.
Program Flash Protect Key [LSC_PROG_ PASSWORD]	0xF1	00 00 00	YY*8	—	Programs the 64-bit password into the device.
Read Flash Protect Key [LSC_READ_ PASSWORD]	0xF2	00 00 00	—	YY*8	Reads the 64-bit password from the device.
Shift Flash Protect Key [LSC_SHIFT_ PASSWORD]	0xBC	00 00 00	YY*8	—	Presents the 64-bit password. When enabled (PWD_enable = 1), the write data is compared to the password contained in 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 * 8	—	Programs the 32-bit control register 0.[2]
Read Control Register 0 [LSC_READ_CTRL0]	0x20	00 00 00	—	YY * 8	Reads the 32-bit control register 0.[2]
JTAG2MSPI Bridge [LSC_PROG_SPI]	0x3A	00 00 00	—	—	Command for JTAG host or programmer to enable programming of external SPI flash through the JTAG port. Connects the TDI, gated TCK, Shift-DR (internal logic), and TDO signals unconditionally to the SISPI, MCLK, CSSPIN, and SPISO pins of the controller SPI bus, respectively.

Notes:

1. For the MachXO4 (LFMXO4) device, SECURITY and SECURITY PLUS commands are mutually exclusive.
2. For more information on each control register bit, refer to the Control Registers section.

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 MachXO4 device configuration ports. The JTAG and target SPI configuration ports behave identically when performing read operations. The I2C port requires a modified access protocol. A high-level representation of the data flow, by port, is shown in Figure 8.19.

All ports start the read process in the same way, by sending a Read Flash or Read UFM command. The MachXO4 device begins the read process once the command byte has been accepted by the configuration logic. The page address pointer determines the first page returned from the MachXO4 device. For the first returned page to be valid (for example, for single-page read operations), a retrieval delay of 240 ns (LFMXO4-025 or equivalent and bigger devices) or 360 ns (LFMXO4-015 or equivalent and smaller devices) must be observed. The retrieval delay time is from the end of the command byte transmission to the end of the first operand byte transmission (refer to Figure 8.18). Note that for slower interface clock rates, 240 or 360 ns may be consumed entirely by the normal transmission of the first operand so no additional delay may be necessary.

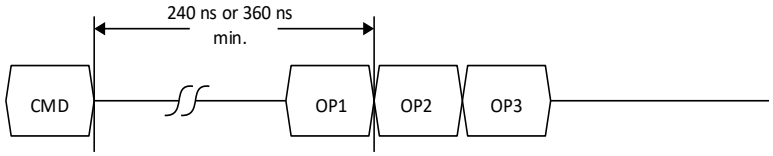
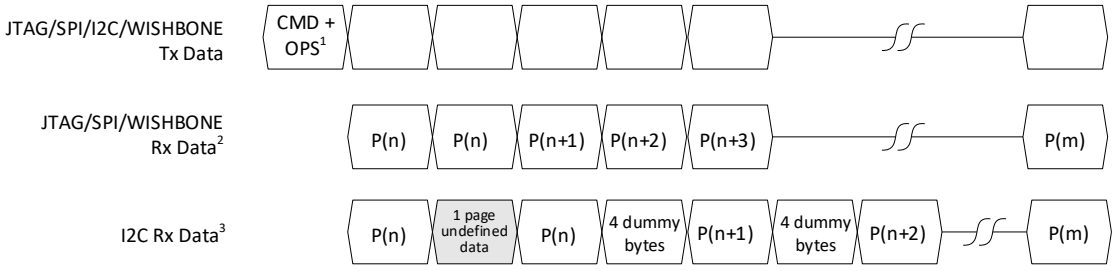


Figure 8.18. Retrieval Delay Timing Requirement for Single-Page Reads



Notes:
1. CMD + OPS = Read CFG or Read UFM command byte + 3 operand bytes.
2. JTAG/Target SPI must transmit data in order to read data back. The data sent by the JTAG/target SPI controller is not specified (*do not care*).
3. I2C must use RESTART between sending the CMD and reading the data (issuing a STOP terminates a CMD and resets the I2C state machine).

Figure 8.19. Flash Page Command and Data Sequence

Figure 8.19 shows a multiple page read sequence. The Read CFG or Read UFM command is transmitted to the MachXO4 device. As shown in Figure 8.19, all interfaces return the page at the page address pointer immediately. For single-page read operations, all configuration ports are allowed to terminate the read immediately following the transfer of the final byte of the first page. The I2C interface differs only in the Read CFG or Read UFM operand bytes.

Reading more than one page requires special handling. The multiple page read duplicates the page selected by the page address pointer. The result of this behavior is that the page count must be one greater than the desired number of pages. For example, reading two pages requires the page count supplied in the Read CFG or Read UFM command to be assigned a value of 3. If the page address pointer is 0000, the MachXO4 device returns three pages namely Page 0, Page 0, and Page 1. When using the WISHBONE interface to read the flash, if you are reading 13 or more pages, the page count must be set to the maximum (16383 decimal or 0x3FFF). The user logic is not required to read this number of pages and may safely truncate the read operation after the desired number of pages have been read.

The I2C interface has additional overhead when reading flash pages. Figure 8.19 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 I2C interface returns Page 0, 16 undefined bytes (LFMXO4), Page 0, 4 dummy bytes, and Page 1. Reading the final four dummy bytes is optional.

8.7. Status Registers

Table 8.3. MachXO4 (LFMXO4) Device Status Register

Field	Name	Description
[31]	FlowThrough Mode (FT Mode)	Device is in flow-through mode.
[30]	Bypass Mode	Device is in bypass mode.
[29]	SED Error	Soft error detection logic detected an error.
[28	Invalid Command	Instruction found in the Command field of the frame is invalid.
[27]	ID Error	ID code mismatch detected for the verify_id instruction/command.
[26]	Execution Error (EXEC Error)	Previous command execution encountered an error.
[25:23]	BSE Error Code	000 – No Error 001 – ID Error; a mismatch in the device ID code is detected. 010 – CMD Error; an illegal command is detected. 011 – CRC Error; a CRC error is detected. 100 – PRMB Error; a preamble error is detected. 101 – ABRT Error; configuration was aborted by the user. 110 – OVFL Error; a data overflow error is detected. 111 – SDM EOF; bitstream exceeds the size of flash memory or SRAM array.
[22]	SPIm Fail 1	Primary pattern failed when dual-boot is enabled.
[21	Std Preamble	A standard preamble (for non-encrypted bitstream) is detected from the last bitstream execution.
[20]	Encrypt Preamble	An encrypt preamble (for encrypted bitstream) is detected from the last bitstream execution.
[19]	SDM EN	Self-download mode is enabled. In self-download mode, the device reads configuration data from the internal flash memory array.
[18]	ASSP	ASSP poly fuses (volatile) or flash cells (non-volatile) are programmed.
[17]	UFM OTP	UFM is one-time programmable. Only reads from the UFM block are possible.
[16]	PWD Enable	Password protection is enabled.
[15]	Decrypt Only	0 – Decryption is not supported. 1 – Only encrypted data is accepted.
[14]	FEA OTP	Feature Row is one-time programmable.
[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 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 is set. • Password protection is enabled, and password mismatch occurs. • Selected configuration target is OTP enabled. 1 – Selected configuration target is write-enabled.
[9]	ISC Enable	JTAG instructions are being executed with ISC enabled.
[8]	DONE	Device configuration is complete and the internal DONE bit is set.
[7]	Decrypt Enable	Decrypted bitstreams can be accepted.
[6]	OTP	Device is one-time programmable (device cannot be programmed again).
[5]	PWD Protect	Configuration logic is password protected.
[4]	JTAG Active	JTAG state machine is active.

Field	Name	Description
[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]	JTAG TRANSPARENT Mode (TRAN Mode)	Device is in transparent mode.

8.8. Control Registers

Table 8.4. MachXO4 (LFMXO4) Device Control Register 0

Field	Name	Default	Description
[31:30]	Core CLK SEL	0	Sets the configuration engine clock frequency with respect to the internal oscillator frequency. 0 – Divide by 2 1 – Divide by 4 2 – Divide by 5 3 – Divide by 6
[29]	Wake Up	0	Controls transparent reconfiguration in the event of PROGRAMN pin pulsing or REFRESH command execution. 0 – Wake-up signals GOE and GSR go low during reconfiguration. 1 – Wake-up signals GOE and GSR stay high during reconfiguration.
[28]	NDR	0	Enables TransFR capability (leave-alone I/O) to latch and freeze I/O pins during reconfiguration.
[27]	No BKE	0	0 – Automatic SRAM bulk erase for power cycling, refresh, or PROGRAMN pin pulsing. 1 – Disable automatic bulk erase.
[26]	No CDM	0	Prevents the download of critical device control information from non-volatile memory to shadow registers in the event of PROGRAMN pin pulsing or REFRESH command execution.
[25]	TranEdit	0	Enables transparent CRAM programming.
[24]	HFC	0	Enables hardened control function when device wakes up.
[23]	Reserved	0	Reserved
[22]	Reserved	0	Reserved
[21]	SRME	0	Slow Response Mode Enable control bit. Enables the automatic insertion of the Lattice specific protocol to handle the issue caused by the slow response of the flash or SRAM operation for high-speed target SPI read. If this bit is set to *1*, the STX DUM setting (CR0[11:10]) is discarded, and a non-deterministic number of *all-1* bytes followed by one *all-0* byte is added before the first readback data frame for incremental read commands.
[20]	SPIM	0	Selects controller SPI boot address. 0 – Use hard-coded boot address (0x000000 for single boot; 0x010000 for dual-boot primary; 0xFFFF00 for dual-boot secondary)
[19:18]	P_DONE	0	PROGRAM_DONE overload control option for bitstream. If bit 19 is set to *1*, the PROGRAM_DONE command is set to overload with either the BYPASS or FLOW_THROUGH function, depending on bit 18. 10 – Overload with BYPASS 11 – Overload with FLOW_THROUGH 0X – No overload (default)
[17:15]	INITN OPT	0	Override INITN. If CR0[17] is set to *1*, the INITN pin is overridden by CR0[15]. Otherwise, the INITN pin is controlled by the configuration logic. Bit 1 – Override INITN pin with bit 15 Bit 0 – No override of INITN pin 16 – Connect to active drive Connect INITN pin to internal pull-up 15 – Drive INITN pin low Release INITN pin

Field	Name	Default	Description
[14:12]	DONE OPT	0	Override DONE. If CR0[14] is set to *1*, the DONE pin is overridden by CR0[12]. Otherwise, the DONE pin is controlled by the configuration logic. Bit 1 — Override DONE pin with bit 15 Bit 0 — No override of DONE pin 14 — Override DONE pin with bit 15 — No override of DONE pin 13 — Connect to active drive — Connect DONE pin to internal pull-up 12 — Drive DONE pin low — Release DONE pin
[11:10]	STX DUM	0	Determines the number of dummy bytes to add as padding before the first frame is read out during incremental bitstream read back (target mode). 00 – 0 byte 01 – 4 bytes 10 – 8 bytes 11 – 16 bytes
[9]	MCLK B(Bypass)	0	Enables MCLK bypass.
[8]	LSBF	0	Controller SPI sends out command and address with least significant bit first.
[7]	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.
[6]	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.
[5:0]	Controller Clock Frequency	0	Divides the clock coming from the on-chip oscillator. The divider setting is determined based on the MCCLK_FREQ setting in the Global tab of the Device Constraint Editor.

8.9. Modifying the MachXO4 Feature Row

The feature row has a total of 80 bits divided as follows:

- 64-bit FEATURE
- 16-bit FEABITS

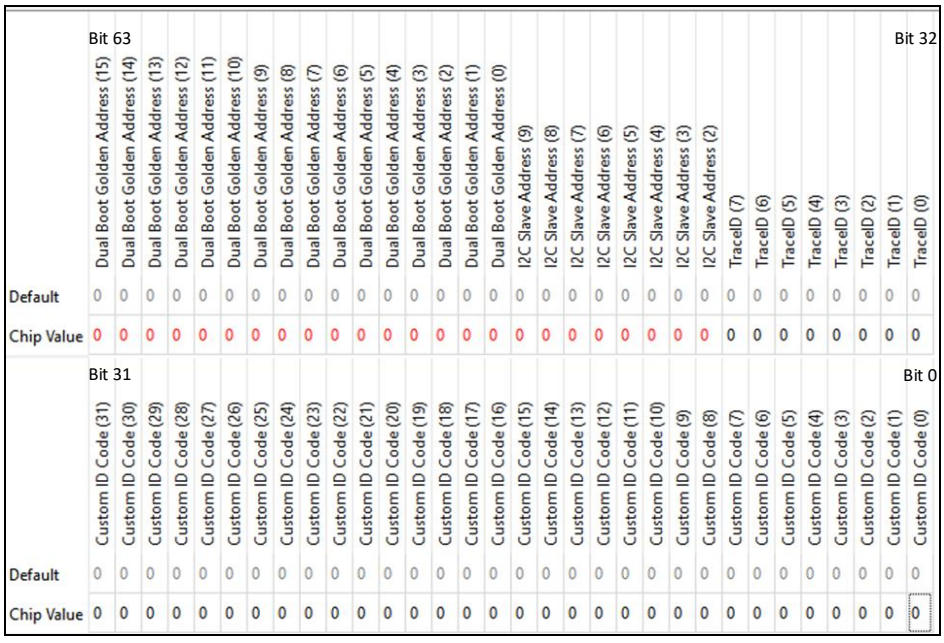
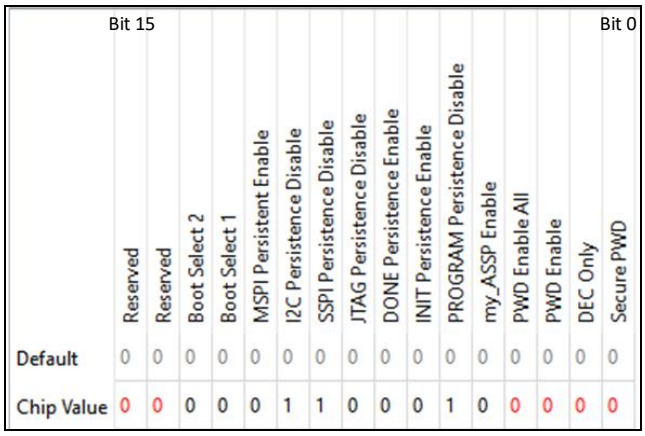


Figure 8.20. Feature Row – FEATURE Bits (Custom ID/ TraceID/I2C Slave Address)



	Bit 15															Bit 0
	Reserved	Reserved	Boot Select 2	Boot Select 1	MSPI Persistent Enable	I2C Persistence Disable	SSPI Persistence Disable	JTAG Persistence Disable	DONE Persistence Enable	INIT Persistence Enable	PROGRAM Persistence Disable	my_ASSP Enable	PWD Enable All	PWD Enable	DEC Only	Secure PWD
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0

Figure 8.21. 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 the MachXO4 device by modifying feature row fuse settings in the data file (*.jed*). Before modifying the feature row, you must first create a project in the Lattice Radiant software and run the *Export Files* process. The *.jed* file is generated inside the *Implementation* folder.

To modify the feature row, perform the following steps:

1. In the Radiant Programmer, select **Tools > Programming File Utility**. The Programming File Utility window appears.
2. In the Programming File Utility, select **Tools > Feature Row Editor > Browse .jed file > Read .jed 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 (*.jed*). Otherwise, click **Save As** to save the data file (*.jed*) with a different name.

8.9.2. Programming the Feature Row Using the Programming File

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

1. Run the Radiant Programmer.
2. Select **Run > Scan Device**.
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:
   - Target Memory: Advanced Security Keys Programming
   - Port Interface: JTAG (or any available interface such as Target SPI or I2C)
   - Access Mode: Feature Rows Programming
   - Operation: Program Feature Row

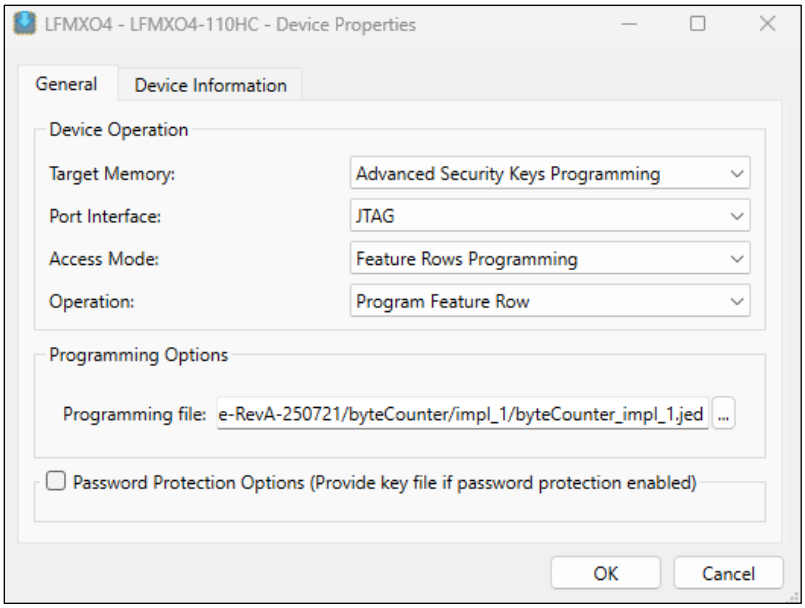


Figure 8.22. Device Properties – Program Feature Row

5. Under **Programming Options**, select the .jed 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 Programming File

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

To update the feature row, perform the following steps:

1. In the Radiant 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:
   - Target Memory: Advanced Security Keys Programming
   - Port Interface: JTAG (or any available interface such as: Target SPI or I2C)
   - Access Mode: Feature Rows Programming
   - Operation: Update Feature Row

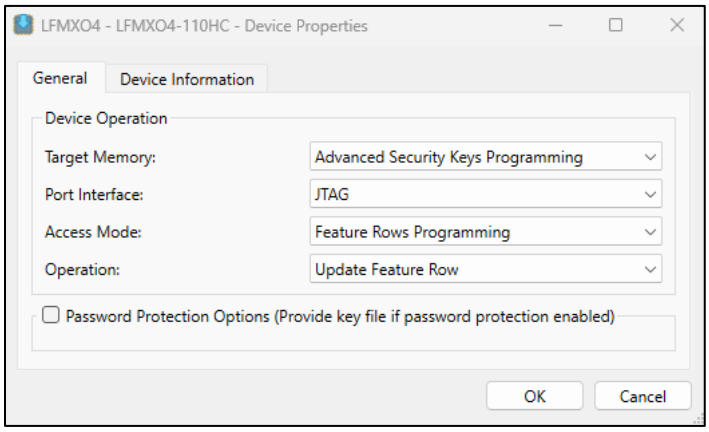


Figure 8.23. Device Properties – Update Feature Row

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 the values displayed in black are applied.

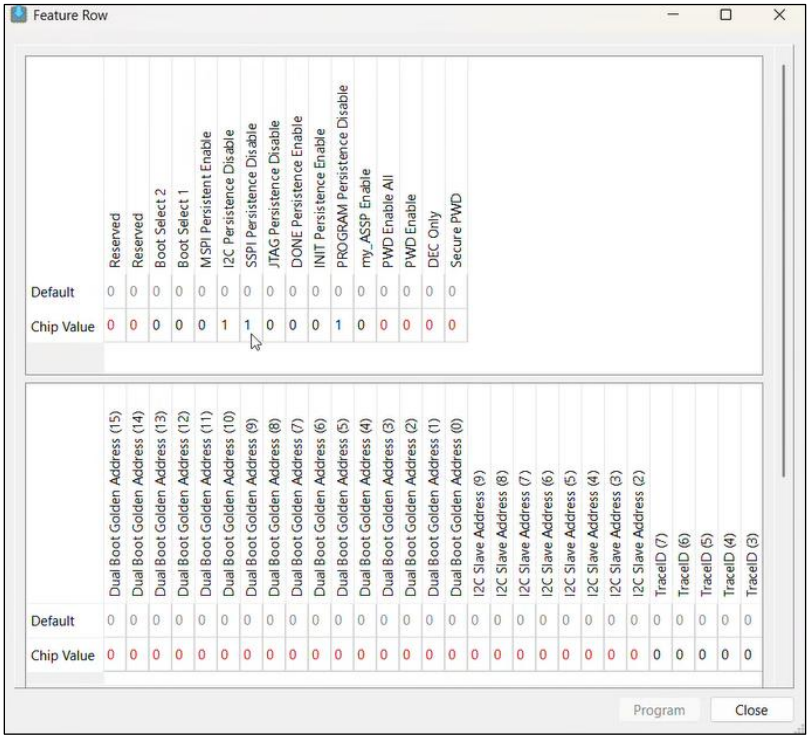


Figure 8.24. Modifying Feature Row Bits

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 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 Radiant Programmer, select the device whose control register you want to modify.
2. Select **Edit > Device Properties**. The Device Properties window appears.
3. Change the device properties to the following settings:
   - Target Memory: Static Random Access Memory
   - Port Interface: JTAG (or any available interface such as Target SPI or I2C)
   - Access Mode: Direct Programming
   - Operation: Program Control Register0

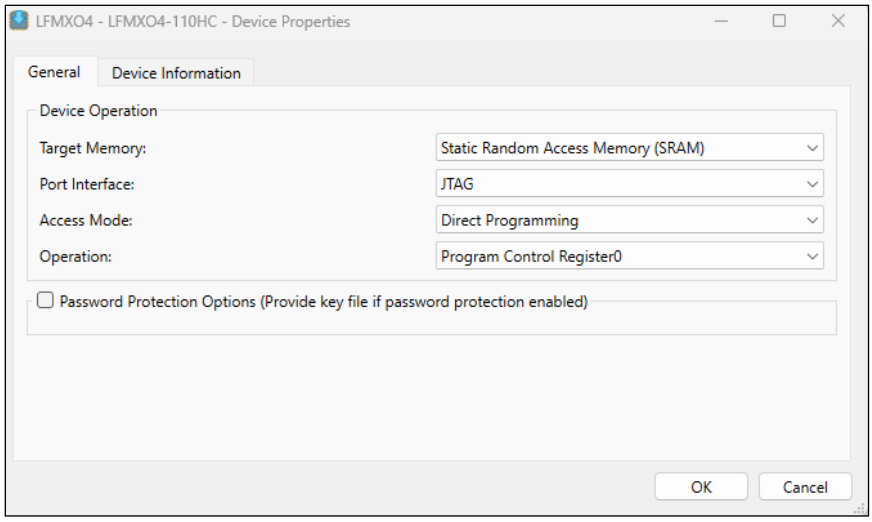


Figure 8.25. Device Properties – SRAM Program Control Register0

4. Click **OK** to apply the settings and close the Device Properties window.
5. Click **Program Device**. The Control Register 0 window appears.
6. Double click the chip value associated with *No CDM* to toggle the value from 0 to 1.

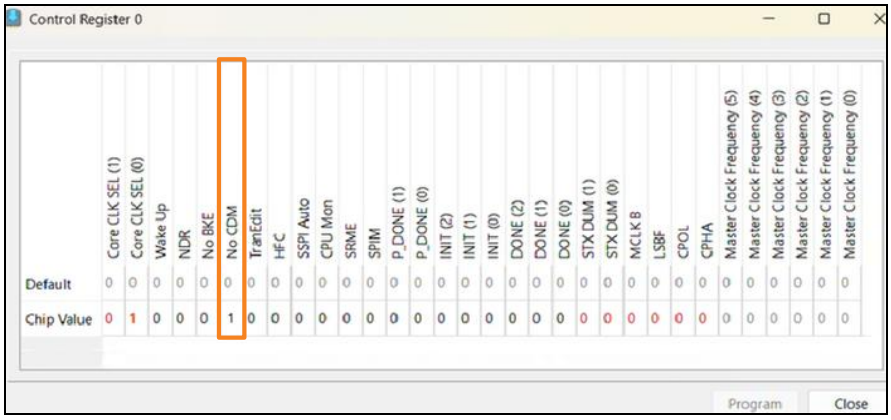


Figure 8.26. Modifying No CDM Bit

7. Click **Program**.

8.9.5. Feature Row Programming File

Figure 8.27 maps bits in the .jed file (generated by the Lattice Radiant software) to the feature row bits read out through the Radiant Programmer.

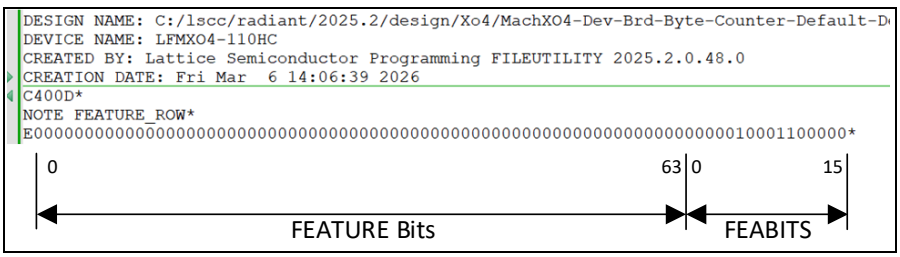


Figure 8.27. Sample .jed File

For information on the individual bits:

- 64-bit FEATURE bits – refer to the following table.

Table 8.5. 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

- 16-bit FEABITS – refer to MachXO4 Hardened Control Functions Reference Guide (FPGA-TN-02404).

References

- MachXO4 Family Data Sheet (FPGA-DS-02125)
- MachXO4 Hardened Control Functions User Guide (FPGA-TN-02403)
- MachXO4 Hardened Control Functions Software Reference Guide (FPGA-TN-02404)
- MachXO4 Soft Error Detection (SED) and Correction (SEC) User Guide (FPGA-TN-02406)
- Using Password Security with MachXO4 Devices (FPGA-TN-02408)
- Minimizing System Interruption During Configuration Using TransFR Technology (FPGA-TN-02198)
- Using TraceID (FPGA-TN-02084)
- MachXO4 web page
- Lattice Insights web page 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 1.2, April 2026

Section	Change Summary
All	Updated links to the Using Password Security with MachXO4 Devices document.

Configuration Process and Flow	• Updated Figure 5.1. Configuration Flow to clarify that the input rail monitored at power-up is V_{CCIO0}.
	• In Table 5.3. MachXO4 Device Feature Row Elements:
	  • Added DUALBOOTGOLDEN.
	• In Table 5.4. Correlation of Device Constraint Editor and Feature Row Settings for MachXO4 (LFMXO4) Device:
	  • Added the DUALBOOTGOLDEN column.
	  • Added the dual boot (EXT_CFG) mode row.

Device Configuration	• In the Golden Image Dual-Boot Configuration section:
	  • Added reference to DUALBOOTGOLDEN and related description.
	  • In Table 6.3. Golden Image Dual-Boot Configuration Software Settings:
	  • Added DUALBOOTGOLDEN.

Software Selectable Options	• In the Configuration Mode and Port Options section:
	  • Removed discussion on feature row and options bits in relation to configuration mode and port options setup for booting from external SPI flash and port persistence.
	  • Removed the Feature Row Editor figure.
	• In the MASTER_SPI_PORT section:
	  • In Table 7.4. MASTER_SPI_PORT Option:
	  • Added reference to EXT_CFG and DUALBOOTGOLDEN in Note 1.
	• In the Bitstream Generation Options section:
	  • Added reference to DUALBOOTGOLDEN in the description.
	  • In Table 7.10. Bitstream Generation Options:
	  • Added DUALBOOTGOLDEN.
	• Added the DUALBOOTGOLDEN section.

Advanced Configuration Information	Added the Modifying the MachXO4 Feature Row section.

Revision 1.1, January 2026

Section	Change Summary
Configuration Details	• In the SN section under Target SPI sysCONFIG Pins: • Added SN condition for SDM mode.

Configuration Process and Flow	• In Table 5.3. MachXO4 Device Feature Row Elements:
	  • Removed reference to DUALBOOTGOLDEN.
	• In Table 5.4. Correlation of Device Constraint Editor and Feature Row Settings for MachXO4 (LFMXO4) Device:
	  • Removed the DUALBOOTGOLDEN column.
	  • Removed the dual boot (EXT_CFG) mode row.
	  • Updated CONFIGURATION settings to include CFG_EBRUFM and CFGUFM.
	  • Added note to clarify the abbreviations for the different boot modes.

Device Configuration	• In the Self-Download Mode section:
	  • Added note on recommended setting for SLAVE_SPI_PORT option in SDM.
	• In the Golden Image Dual-Boot Configuration section:
	  • Removed reference to DUALBOOTGOLDEN and related description.
	  • In Table 6.3. Golden Image Dual-Boot Configuration Software Settings:
	  • Removed reference to DUALBOOTGOLDEN.
	• In the Target SPI Mode section:

Section	Change Summary
	• Made minor editorial change.

Software Selectable Options	• In the Configuration Mode and Port Options section:
	  • Updated description to include SDM_PORT influence on feature row bits.
	• In the SLAVE_SPI_PORT section:
	  • Updated description of the SLAVE_SPI_PORT option.
	• In the MASTER_SPI_PORT section:
	  • In Table 7.4. MASTER_SPI_PORT Option:
	  • Removed reference to EXT_CFG and DUALBOOTGOLDEN in Note 1.
	• In the SDM_PORT section:
	  • Updated description of SDM port to clarify behavior.
	• In the Bitstream Generation Options section:
	  • Removed reference to DUALBOOTGOLDEN in the description and Table 7.10. Bitstream Generation Options.
	• Removed the DUALBOOTGOLDEN section.

Advanced Configuration Information	• In the Status Registers section: • Removed the MachXO4 (LFMXO4) Device Boot Selection Modes table.

Revision 1.0, December 2025

Section	Change Summary
All	• Made editorial changes and modifications to cover MachXO4 (LFMXO4) device. • Changed term to *IP Catalog*.
Abbreviations in This Document	• Added *CDM*, *FIFO*, and *Rx*. • Removed *LPF*.
Definition of Terms	• Added *Background Reconfiguration*. • Updated term to *Offline Mode*.

Configuration Details	• In the Initialization section:
	  • Updated description to include PROGRAMN pin pulsing and REFRESH command execution as other conditions for entering the initialization phase.
	  • Added description of CDM.
	• In Table 4.4. Default State of sysCONFIG Pins:
	  • Updated pin direction for CSSPIN to input/output with weak pull-up.
	• Added links to MachXO4 Family Data Sheet in the following sections:
	  • PROGRAMN
	  • MCLK
	  • CCLK
	  • TCK
	• In the CSSPIN section:
	  • Updated description of CSSPIN to indicate weak pull-up.

Configuration Process and Flow	• In the Initialization section:
	  • Added description of CDM.
	  • Updated statement to *prematurely releasing* from *prematurely driving* in relation to INITN.
	• In Table 5.1. Configuration Time Parameters:
	  • Added link to MachXO4 Family Data Sheet in description for Configuration Clock Frequency.
	• Added reference to MachXO4 Hardened Control Functions Software Reference Guide for required duration (erase portion) in note in the Transparent Programming Mode section.
	• In the Feature Row section:
	  • Added Figure 5.4. Feature Row Example – MachXO4 (LFMXO4) Device.
	  • Updated description on saving settings to mention .pdc file.

Section	Change Summary
Device Configuration	• Added link to MachXO4 Family Data Sheet in the Controller SPI Mode section. • Added reference to the MachXO4 Soft Error Detection (SED) and Correction (SEC) User Guide in the Golden Image Dual-Boot Configuration section. • Added description of Rx FIFO when device first boots up in the I2C Configuration Mode section. • Added reference to the MachXO4 Hardened Control Functions User Guide in the WISHBONE Configuration Mode section. • Added the JTAG Daisy Chain section. • Added reference to Using Password Security with MachXO4 Devices and related note in the Password-Based Flash Protection section.

Software Selectable Options	• Added Figure 7.1. sysCONFIG Options on Global Tab of Device Constraint Editor.
	• In the Configuration Mode and Port Options section:
	  • Added discussion on feature row and options bits in relation to configuration mode and port options setup for booting from external SPI flash and port persistence.
	  • Added Figure 7.2. Feature Row Editor.
	• Added note regarding ENABLE_TRANSFR option to description of ON setting for BACKGROUND_RECONFIG option in the BACKGROUND_RECONFIG section.

Advanced Configuration Information	• In the Flash Programming section:
	  • Added Custom to list of programming methods for internal flash.
	  • Added reference to MachXO4 Hardened Control Functions User Guide for number of internal memory pages and statement on internal memory page size.
	• Added Figure 8.1. JEDEC File Example (Part 1), Figure 8.2. JEDEC File Example (Part 2), and introducing sentence.
	• In Table 8.2. MachXO4 Device sysCONFIG Programming Commands:
	  • Updated operand and description for ISC_ENABLE_X.

References	• Added link to MachXO4 Family Data Sheet. • Added MachOX4 Hardened Control Functions User Guide. • Added MachXO4 Hardened Control Functions Software Reference Guide. • Added MachXO4 Soft Error Detection (SED) and Correction (SEC) User Guide. • Added Using Password Security with MachOX4 Devices. • Added MachXO4 web page.

Revision 0.80, March 2025

Section	Change Summary
All	Initial preliminary release.

www.latticesemi.com