MachXO4 Hardened Control Functions Reference Guide

Technical Note

FPGA-TN-02404-1.3

September 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
AES	Advanced Encryption Standard
ECDSA	Elliptic Curve Digital Signature Algorithm
EFB	Embedded Function Block
FPGA	Field Programmable Gate Array
GSR	Global Set/Reset
I/O	Input/Output
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IP	Intellectual Property
JTAG	Joint Test Action Group
LSB	Least Significant Bit
LUT	Look-Up Table
MSB	Most Significant Bit
NVR	Non-Volatile Register
PLD	Programmable Logic Device
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RoT	Root of Trust
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSM	Synchronous State Machine
SSPI	Target Serial Peripheral Interface
UFM	User Flash Memory

1. Introduction

Note: In this document, *MachXO4* refers to both LFMXO4 and LFMXO4D unless explicitly identified as MachXO4 (LFMXO4). *MachXO4 Root of Trust (RoT)* refers to only *LFMXO4D*.

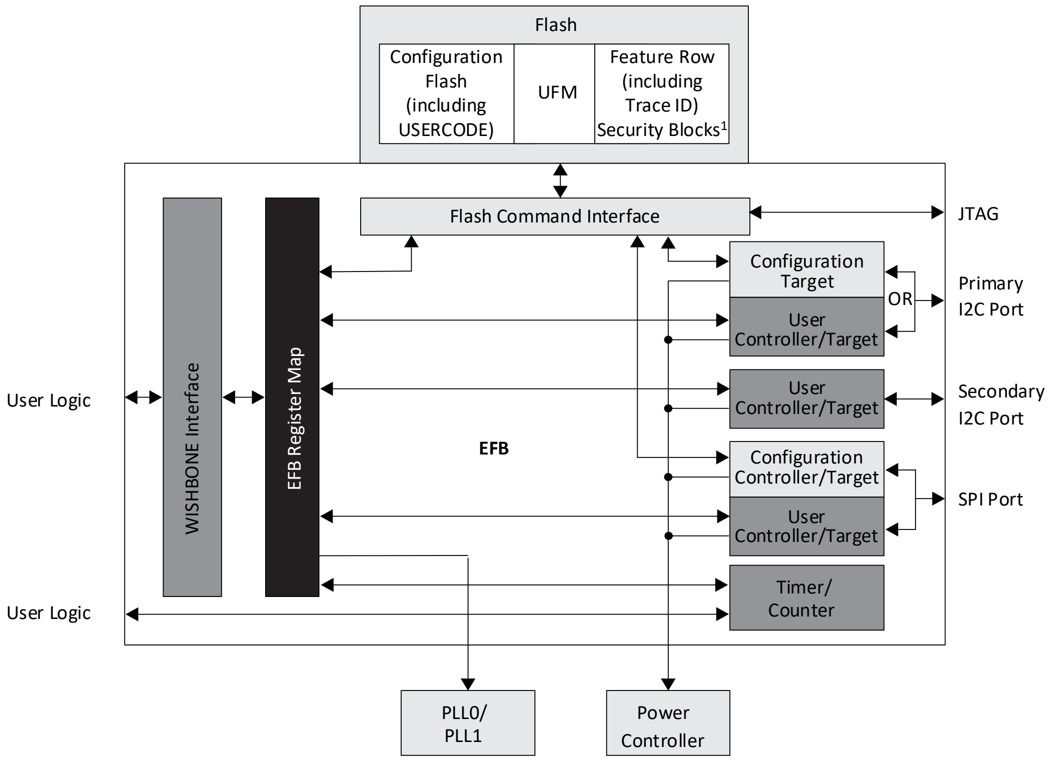
This reference guide supplements the MachXO4 Hardened Control Functions User Guide (FPGA-TN-02403), which explains the software usage. In this document, you can find:

- WISHBONE Protocol
- EFB Register Map
- Command Sequences
- Examples

As an overview, the Lattice MachXO4™ FPGA family combines a high-performance, low-power FPGA fabric with built-in, hardened control functions. The hardened control functions ease design implementation and save general-purpose resources such as LUTs, registers, clocks, and routing. The hardened control functions are physically located in the Embedded Function Block (EFB). All MachXO4 devices include an EFB module. The EFB block includes the following control functions:

- Two I2C Cores
- One SPI Core
- One 16-bit Timer/Counter
- Interface to Flash memory which includes:
    - User Flash Memory
    - Configuration logic
    - Security keys (only for MachXO4 RoT devices)
    - Feature and security settings (only for MachXO4 RoT devices)
- Interface to Dynamic PLL configuration settings
- Interface to On-chip Power Controller through I2C and SPI

Figure 1.1 shows the EFB architecture and the interface to the FPGA core logic.



Note:
1. Security blocks are available only for MachXO4 RoT devices.

Figure 1.1. Embedded Function Block (EFB)

1.1. EFB Register Map

The EFB module has a register map to allow the service of the hardened functions through the WISHBONE bus interface using read/write operations. Each hardened function has dedicated 8-bit Data and Control registers, except for the Flash. These are accessed through the same set of registers. Table 1.1 documents the register map of the EFB module. The PLL registers are located in the Flash in the MachXO4 devices PLL modules, but are accessed through the EFB WISHBONE read/write cycles.

Table 1.1. EFB Register Map

Address (Hex)	Hardened Function
0x00-0x1F	PLL0 Dynamic Access1
0x20-0x3F	PLL1 Dynamic Access1
0x40-0x49	I2C Primary
0x4A-0x53	I2C Secondary
0x54-0x5D	SPI
0x5E-0x6F	Timer/Counter
0x70-0x75	Flash
0x76-0x77	EFB Interrupt Source

Note:

1. There can be up to two PLLs in a MachXO4 device. PLL0 has an address range from 0x00 to 0x1F. PLL1 (if present) has an address range from 0x20 to 0x3F. Refer to the MachXO4 sysCLOCK PLL Design User Guide (FPGA-TN-02391) for details on the PLL configuration registers and recommended usage.

Address spaces that are not defined in Table 1.1 are invalid and result in non-deterministic behavior. Ensure valid addresses are presented to the EFB WISHBONE target interface.

1.2. WISHBONE Bus Interface

The WISHBONE bus in the MachXO4 is compliant with the WISHBONE standard from OpenCores. It provides connectivity between FPGA user logic and the EFB functional blocks. You can implement a WISHBONE controller interface to interact with the EFB WISHBONE target interface, or a LatticeMico8™ soft processor core can be used to interact with the EFB WISHBONE.

The block diagram in Figure 1.2 shows the supported WISHBONE bus signals between the FPGA core and the EFB.

Table 1.2 provides a detailed definition of the supported signals.

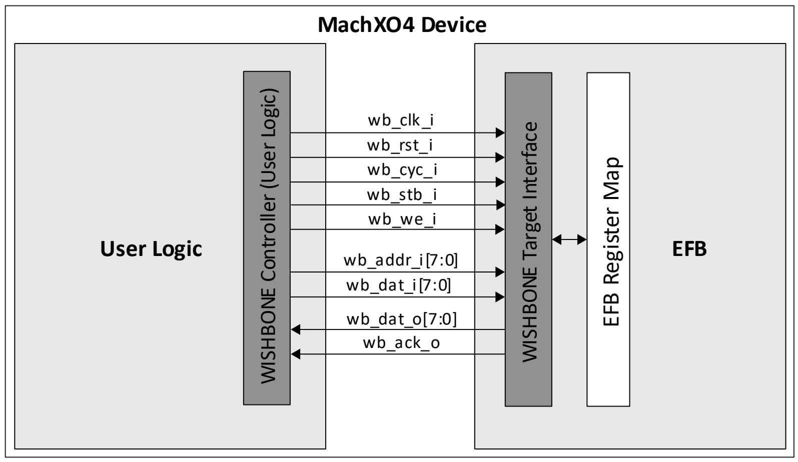


Figure 1.2. WISHBONE Bus Interface between the FPGA Core and the EFB Module

Table 1.2. WISHBONE Target Interface Signals of the EFB Module

Signal Name	I/O	Width	Description
wb_clk_i	Input	1	Positive edge clock used by WISHBONE Interface registers and hardened functions within the EFB module. Supports clock speeds up to 133 MHz. When used in conjunction with the I2C User Target or Configuration Target ports, the clock speed must be at least 7.5× the I2C bus speed (for example, >3.0 MHz when I2C rate = 400 kHz). When used in conjunction with SPI Target port, the WISHBONE clock speed must be at least twice the SPI clock speed.
wb_rst_i	Input	1	Active-high, synchronous reset signal that only resets the WISHBONE interface logic. This signal does not affect the contents of any registers. It only affects ongoing bus transactions. Wait 1 µs after de-assertion before starting any subsequent WISHBONE transactions.
wb_cyc_i	Input	1	Active-high signal, asserted by the WISHBONE controller, indicates a valid bus cycle is present on the bus.
wb_stb_i	Input	1	Active-high strobe, input signal, indicating the WISHBONE target is the target for the current transaction on the bus. The EFB module asserts an acknowledgment in response to the assertion of the strobe.
wb_we_i	Input	1	Level sensitive Write/Read control signal. Low indicates a Read operation, and High indicates a Write operation.
wb_adr_i	Input	8	8-bit wide address used to select a specific register from the register map of the EFB module.
wb_dat_i	Input	8	8-bit input data path used to write a byte of data to a specific register in the register map of the EFB module.
wb_dat_o	Output	8	8-bit output data path used to read a byte of data from a specific register in the register map of the EFB module.
wb_ack_o	Output	1	Active-high, transfer acknowledge signal asserted by the EFB module, indicating the requested transfer is acknowledged.

To interface to the EFB, you must create a WISHBONE controller in the User Logic. In a multiple-Controller configuration, the WISHBONE Controller outputs are multiplexed in a user-defined arbiter. A LatticeMico8 soft processor can also be used along with the Mico System Builder (MSB) platform, which can implement multi-controller bus configurations. If two Controllers request the bus in the same cycle, only the outputs of the arbitration winner reach the Target interface.

The EFB WISHBONE bus supports the Classic version of the WISHBONE standard. Given that the WISHBONE bus is an open source standard, not all features of the standard are implemented or required:

- Tags are not supported in the WISHBONE Target interface of the EFB module. Given that the EFB is a hardened block, these signals cannot be added.
- The Target WISHBONE bus interface of the EFB module does not require the byte select signals (sel_i or sel_o), since the data bus is only a single byte wide.
- The EFB WISHBONE Target interface does not support the optional error and retry access termination signals. If the target receives an access to an invalid address, it responds by asserting wb_ack_o signal. Stay within the valid address range.

1.3. WISHBONE Write Cycle

Figure 1.3 shows the waveform of a Write cycle from the perspective of the EFB WISHBONE Target interface. During a single Write cycle, only one byte of data is written to the EFB block from the WISHBONE Controller. A Write operation requires a minimum of three clock cycles.

On clock Edge 0, the Controller updates the address, data and asserts control signals. During this cycle:

- The Controller updates the address on the wb_adr_i[7:0] address lines.
- Updates the data that is written to the EFB block, wb_dat_i[7:0] data lines.
- Asserts the write enable wb_we_i signal, indicating a write cycle.

- Asserts the wb_cyc_i to indicate the start of the cycle.
- Asserts the wb_stb_i, selecting a specific target module.

On clock Edge 1, the EFB WISHBONE Target decodes the input signals presented by the controller. During this cycle:

- The Target decodes the address presented on the wb_adr_i[7:0] address lines.
- The Target prepares to latch the data presented on the wb_dat_i[7:0] data lines.
- The Controller waits for an active-high level on the wb_ack_o line and prepares to terminate the cycle on the next clock edge, if an active-high level is detected on the wb_ack_o line.
- The EFB may insert wait states before asserting wb_ack_o, thereby allowing it to throttle the cycle speed. Any number of wait states may be added.
- The Target asserts wb_ack_o signal.

The following occurs on clock Edge 2:

- The Target latches the data presented on the wb_dat_i[7:0] data lines.
- The Controller de-asserts the strobe signal, wb_stb_i, the cycle signal, wb_cyc_i, and the write enable signal, wb_we_i.
- The Target de-asserts the acknowledge signal, wb_ack_o, in response to the Controller de-assertion of the strobe signal.

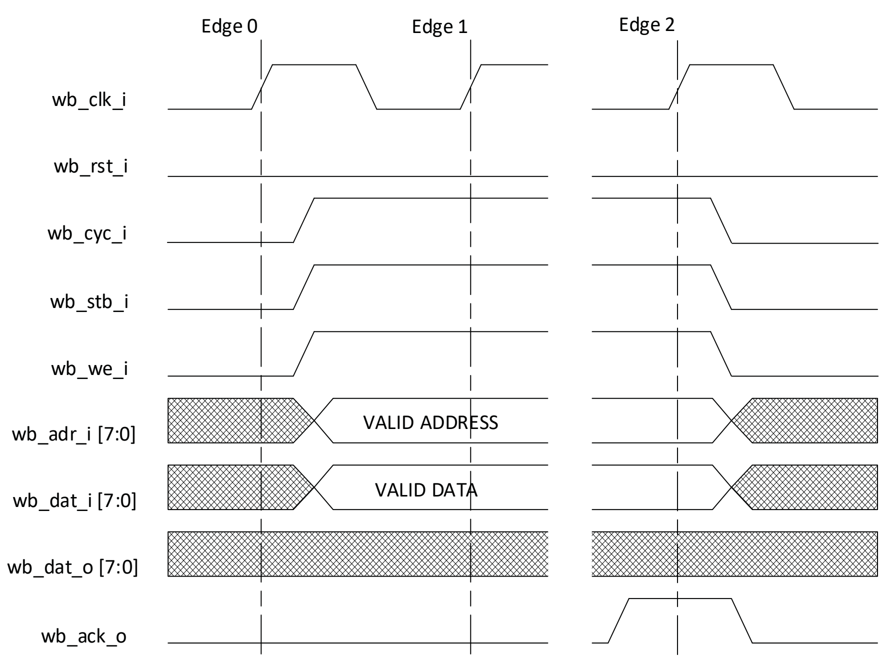


Figure 1.3. WISHBONE Bus Write Operation

1.4. WISHBONE Read Cycle

Figure 1.4 shows the waveform of a Read cycle from the perspective of the EFB WISHBONE Target interface. During a single Read cycle, only one byte of data is read from the EFB block by the WISHBONE controller. A Read operation requires a minimum of three clock cycles.

On clock Edge 0, the Controller updates the address, data and asserts control signals. The following occurs during this cycle:

- The Controller updates the address on the wb_adr_i[7:0] address lines.
- De-asserts the write enable wb_we_i signal, indicating a Read cycle.
- Asserts the wb_cyc_i to indicate the start of the cycle.
- Asserts the wb_stb_i, selecting a specific Target module.

On clock Edge 1, the EFB WISHBONE target decodes the input signals presented by the controller. The following occurs during this cycle:

- The Target decodes the address presented on the wb_adr_i[7:0] address lines.
- The Controller prepares to latch the data presented on wb_dat_o[7:0] data lines from the EFB WISHBONE target on the following clock edge.
- The Controller waits for an active-high level on the wb_ack_o line and prepares to terminate the cycle on the next clock edge, if an active-high level is detected on the wb_ack_o line.
- The EFB may insert wait states before asserting wb_ack_o, thereby allowing it to throttle the cycle speed. Any number of wait states may be added.
- The Target presents valid data on the wb_dat_o[7:0] data lines.
- The Target asserts wb_ack_o signal in response to the strobe, wb_stb_i signal.

The following occurs on clock Edge 2:

- The Controller latches the data presented on the wb_dat_o[7:0] data lines.
- The Controller de-asserts the strobe signal, wb_stb_i, and the cycle signal, wb_cyc_i.
- The Target de-asserts the acknowledge signal, wb_ack_o, in response to the controller de-assertion of the strobe signal.

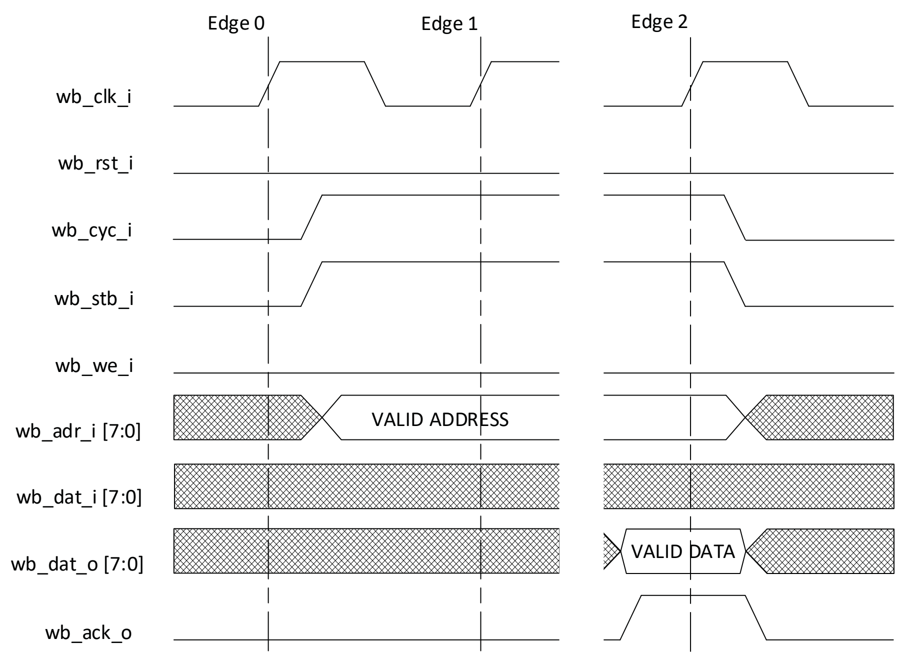


Figure 1.4. WISHBONE Bus Read Operation

To avoid simulation mismatch in functional simulations, add a delay of 100 ps to wb_cyc_i and wb_stb_i assertion assignments. Refer to the examples below. The examples assume the *wb_cyc_i_gen* signal is generated elsewhere in the design, for example, from a synchronous state machine (SSM).

Verilog example: (assumes timescale 1 ns / 100 ps)

```
assign wb_cyc_i = #0.100 wb_cyc_i_gen;
```

VHDL example:

```
wb_cyc_i <= wb_cyc_i_gen after 100 ps;
```

Additionally, ensure your logic monitors for wb_ack_o, and de-assert wb_cyc_i and wb_stb_i immediately.

1.5. WISHBONE Reset Cycle

Figure 1.5 shows the waveform of the synchronous wb_rst_i signal. Asserting the reset signal, resets the WISHBONE interface logic. This signal does not affect the contents of any registers in the EFB register map. It only affects ongoing bus transactions.

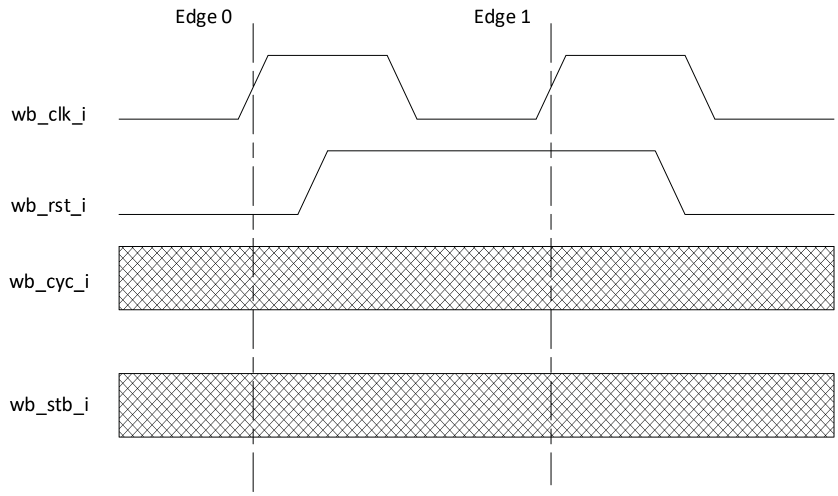


Figure 1.5. EFB WISHBONE Interface Reset

The wb_rst_i signal can be asserted for any length of time.

2. Hardened I2C IP Cores

I2C is a widely used two-wire serial bus for communication between devices on the same board. Every MachXO4 device contains two hardened I2C IP cores designated as the Primary and Secondary I2C IP cores. Either of the two cores can be operated as an I2C Controller or as an I2C Target. The difference between the two cores is that the Primary core has pre-assigned I/O pins while the ports of the Secondary core can be assigned to any general purpose I/O. In addition, the Primary I2C core can be used for accessing the Flash. However, the Primary I2C port cannot be used for both Flash access and user functions in the same design. When instantiating the Hardened I2C IP cores for Target operations, the Embedded Function Block (EFB) *wb_clk_i* input must be connected to a valid clock source of at least 7.5× the I2C bus rate (for example, >3.0 MHz when I2C rate = 400 kHz).

2.1. I2C Registers

Both I2C cores communicate with the EFB WISHBONE interface through a set of control, command, status, and data registers. Table 2.1 shows the register names and their functions. These registers are a subset of the EFB register map.

Table 2.1. I2C Registers

I2C Primary Register Name	I2C Secondary Register Name	Register Function	Address I2C Primary	Address I2C Secondary	Access
I2C_1_CR	I2C_2_CR	Control	0x40	0x4A	Read/Write
I2C_1_CMDR	I2C_2_CMDR	Command	0x41	0x4B	Read/Write
I2C_1_BR0	I2C_2_BR0	Clock Pre-scale	0x42	0x4C	Read/Write
I2C_1_BR1	I2C_2_BR1	Clock Pre-scale	0x43	0x4D	Read/Write
I2C_1_TXDR	I2C_2_TXDR	Transmit Data	0x44	0x4E	Write
I2C_1_SR	I2C_2_SR	Status	0x45	0x4F	Read
I2C_1_GCDR	I2C_2_GCDR	General Call	0x46	0x50	Read
I2C_1_RXDR	I2C_2_RXDR	Receive Data	0x47	0x51	Read
I2C_1_IRQ	I2C_2_IRQ	IRQ	0x48	0x52	Read/Write
I2C_1_IRQEN	I2C_2_IRQEN	IRQ Enable	0x49	0x53	Read/Write

Note: Unless otherwise specified, all reserved bits in writable registers are written as *0*.

Table 2.2. I2C Control (Primary/Secondary)

I2C_1_CR / I2C_2_CR								0x40/0x4A
Bit	7	6	5	4	3	2	1	0
Name	I2CEN	GCEN	WKUPEN	(Reserved)	SDA_DEL_SEL[1:0]		(Reserved)	
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	—	R/W	R/W	—	—

Note: A write to this register causes the I2C core to reset.

I2CEN

I2C System Enable Bit – This bit enables the I2C core functions. If I2CEN is cleared, the I2C core is disabled and forced into idle state.

0: I2C function is disabled

1: I2C function is enabled

GCEN

Enable bit for General Call Response – Enables the general call response in target mode.

0: Disable

1: Enable

The General Call address is defined as 0000000 and works with either 7-bit or 10-bit addressing.

WKUPEN

Wake-up from Standby/Sleep (by Target Address matching) Enable Bit – When this bit is enabled, the I2C core can send a wake-up signal to the on-chip power manager to wake the device up from standby/sleep. The wake-up function is activated when the MachXO4 Target Address is matched during standby/sleep mode.

0: Disable

1: Enable

SDA_DEL_SEL[1:0]

SDA Output Delay (Tdel) Selection (refer to Figure 2.10).

00: 300 ns (min) 300 ns + 2000/[wb_clk_i frequency in MHz] (max)

01: 150 ns (min) 150 ns + 2000/[wb_clk_i frequency in MHz] (max)

10: 75 ns (min) 75 ns + 2000/[wb_clk_i frequency in MHz] (max)

11: 0 ns (min) 0 ns + 2000/[wb_clk_i frequency in MHz] (max)

Table 2.3. I2C Command (Pri/Sec)

I2C_1_CMDR / I2C_2_CMDR								0x41/0x4B
Bit	7	6	5	4	3	2	1	0
Name	STA	STO	RD	WR	ACK	CKSDIS	(Reserved)	
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	—	—

STA

Generate START (or Repeated START) condition (Controller operation)

STO

Generate STOP condition (Controller operation)

RD

Indicate Read from target (Controller operation)

WR

Indicate Write to target (Controller operation)

ACK

Acknowledge Option – when receiving, ACK transmission selection

0: Send ACK

1: Send NACK

CKSDIS

Clock Stretching Disable. The I2C cores support a wait state or clock stretching from the target, meaning the target can enforce a wait state if it needs time to finish the task. If necessary, you can disable the clock stretching using the CKSDIS bit. In this case, the overflow flag must be monitored. For Controller operations, set this bit to *0*. Clock stretching is used by the MachXO4 EFB I2C Target during both *read* and *write* operations (from the Controller perspective) when I2C Command Register bit CKSDIS=0.

During a read operation (Target transmitting), clock stretching occurs when TXDR is empty (under-run condition). During a write operation (Target receiving) clock stretching occurs when RXDR is full (over-run condition).

Translated into I2C Status register bits, the I2C clock-stretches if TRRDY=1. The decision to enable clock stretching is done on the 8th SCL (byte boundary) + 2 WISHBONE clocks.

0: Enabled

1: Disabled

Table 2.4. I2C Clock Prescale 0 (Primary/Secondary)

I2C_1_BR0 / I2C_2_BR0								0x42/0x4C
Bit	7	6	5	4	3	2	1	0
Name	I2C_PRESCALE[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Hardware default value may be overridden by EFB component instantiation parameters. Refer to the discussion below.

Table 2.5. I2C Clock Prescale 1 (Primary/Secondary)

I2C_1_BR1 / I2C_2_BR1								0x43/0x4D
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)						I2C_PRESCALE[7:0]	
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	—	R/W	R/W

Note: Hardware default value may be overridden by EFB component instantiation parameters. Refer to the discussion below.

I2C_PRESCALE[9:0]

I2C Clock Prescale value. A write operation to I2CBR [9:8] causes an I2C core reset. The WISHBONE clock frequency is divided by (I2C_PRESCALE × 4) to produce the Controller I2C clock frequency supported by the I2C bus (50 kHz, 100 kHz, 400 kHz).

Notes:

- Unlike a transmitting Controller, the practical limit for Target I2C bus speed is (WISHBONE clock)/2048. For example, the maximum WISHBONE clock frequency to support a 50 kHz Target I2C operation is 102 MHz.
- The digital value is calculated by the IP Catalog when the I2C core is configured in the I2C tab of the EFB GUI. The calculation is based on the WISHBONE Clock Frequency and the I2C Frequency, which are user-entered. The digital value of the divider is programmed in the MachXO4 device during device programming. After power-up or device reconfiguration, the data is loaded onto the I2C_1_BR1/0 and I2C_2_BR1/0 registers.

Registers I2C_1_BR1/0 and I2C_2_BR1/0 have Read/Write access from the WISHBONE interface. You can update these clock pre-scale registers dynamically during device operation; however, care must be taken to not violate the I2C bus frequencies.

Table 2.6. I2C Transmit Data Register (Primary/Secondary)

I2C_1_TXDR / I2C_2_TXDR								0x44/0x4E
Bit	7	6	5	4	3	2	1	0
Name	I2C_Transmit[7:0]							
Default	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W

I2C_Transmit_Data[7:0]

I2C Transmit Data. This register holds the byte to be transmitted on the I2C bus during the Write Data phase. Bit 0 is the LSB and is transmitted last. When transmitting the target address, Bit 0 represents the Read/Write bit.

Table 2.7. I2C Status (Primary/Secondary)

I2C_1_SR / I2C_2_SR								0x45/0x4F
Bit	7	6	5	4	3	2	1	0
Name	TIP	BUSY	RARC	SRW	ARBL	TRRDY	TROE	HGC
Default	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

TIP

Transmit In Progress. The current data byte is being transferred. Note that the TIP flag suffers one-half SCL cycle latency right after the START condition because of the signal synchronization. Also, note that this bit can be high after configuration wake-up and before the first valid I2C transfer starts (when BUSY is low), and it does not indicate a byte in transfer and should be treated as invalid.

1: Byte transfer in progress

0: Byte transfer complete

BUSY

I2C Bus busy. The I2C bus is involved in a transaction. This is set at START condition and cleared at STOP. Make sure this bit is set before treating any other I2C SR bits as valid indicators for a valid transfer.

1: I2C bus busy

0: I2C bus not busy

RARC

Received Acknowledge. An acknowledge response is received by the acknowledge bit monitor. All ACK/NACK bits are monitored and reported, regardless of Controller/Target source or Read/Write mode.

1: No acknowledge received

0: Acknowledge received

SRW

Target Read/Write. Indicates transmit or receive mode.

1: Controller receiving/target transmitting

0: Controller transmitting/target receiving

Note: SRW is valid after TRRDY=1 following a synchronization delay of up to four WISHBONE clock cycles. Do not test both SRW and TRRDY in the same WISHBONE transaction, but test SRW at least four WISHBONE clock cycles after TRRDY is tested true. This delay is represented in Figure 2.9.

ARBL

Arbitration Lost. The core has lost arbitration in Controller mode. This bit is capable of generating an interrupt.

1: Arbitration Lost

0: Normal

TRRDY

Transmitter or Receiver Ready. The I2C Transmit Data register is ready to receive transmit data, or the I2C Receive Data Register contains receive data (dependent upon controller/target mode and SRW status). This bit is capable of generating an interrupt.

1: Transmitter or Receiver is ready

0: Transmitter or Receiver is not ready

TROE

Transmitter/Receiver Overrun Error. A transmit or receive overrun error has occurred (dependent upon controller/target mode and SRW status).

Note: When acting as a transmitter (Controller Write or Target Read), a No Acknowledge received asserts TROE indicating a possible orphan data byte exists in TXDR.

This bit is capable of generating an interrupt.

1: Transmitter or Receiver Overrun detected or NACK received

0: Normal

HGC

Hardware General Call Received. A hardware general call has been received in target mode. The corresponding command byte is available in the General Call Data Register. This bit is capable of generating an interrupt.

1: General Call Received in target mode

0: Normal

Table 2.8. I2C General Call Data Register (Primary/Secondary)

I2C_1_GCDR / I2C_2_GCDR								0x46/0x50
Bit	7	6	5	4	3	2	1	0
Name	I2C_GC_Data[7:0]							
Default	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

I2C_GC_Data[7:0]

I2C General Call Data. This register holds the second (command) byte of the General Call transaction on the I2C bus.

Table 2.9. I2C Receive Data Register (Primary/Secondary)

I2C_1_RXDR / I2C_2_RXDR								0x47/0x51
Bit	7	6	5	4	3	2	1	0
Name	I2C_Receive_Data[7:0]							
Default	—	—	—	—	—	—	—	—
Access	R	R	R	R	R	R	R	R

I2C_Receive_Data[7:0]

I2C Receive Data. This register holds the byte captured from the I2C bus during the Read Data phase. Bit 0 is the LSB and is received last.

Table 2.10. I2C Interrupt Status (Primary/Secondary)

I2C_1_IRQ / I2C_2_IRQ								0x48/0x52
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				IRQARBL	IRQTRRDY	IRQTROE	IRQHGC
Default	—	—	—	—	—	—	—	—
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBL

Interrupt Status for Arbitration Lost. When enabled, indicates ARBL is asserted. Write a *1* to this bit to clear the interrupt.

1: Arbitration Lost Interrupt

0: No interrupt

IRQTRRDY

Interrupt Status for Transmitter or Receiver Ready. When enabled, indicates TRRDY is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmitter or Receiver Ready Interrupt

0: No interrupt

IRQTROE

Interrupt Status for Transmitter/Receiver Overrun or NACK received. When enabled, indicates TROE is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmitter or Receiver Overrun or NACK received Interrupt

0: No interrupt

IRQHGC

Interrupt Status for Hardware General Call Received. When enabled, indicates HGC is asserted. Write a *1* to this bit to clear the interrupt.

1: General Call Received in target mode Interrupt

0: No interrupt

Table 2.11. I2C Interrupt Enable (Primary/Secondary)

I2C_1_IRQEN / I2C_2_IRQEN								0x49/0x53
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				IRQARBLEN	IRQTRRDYEN	IRQTROEEN	IRQHGCEN
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBLEN

Interrupt Enable for Arbitration Lost

1: Interrupt generation enabled

0: Interrupt generation disabled

IRQTRRDYEN

Interrupt Enable for Transmitter or Receiver Ready

1: Interrupt generation enabled

0: Interrupt generation disabled

IRQTROEEN

Interrupt Enable for Transmitter/Receiver Overrun or NACK Received

1: Interrupt generation enabled

0: Interrupt generation disabled

IRQHGCEN

Interrupt Enable for Hardware General Call Received

1: Interrupt generation enabled

0: Interrupt generation disabled

Figure 2.1 shows a flow diagram for controlling Controller I2C reads and writes initiated through the WISHBONE interface. The following sequence is for the Primary I2C but the same sequence applies to the Secondary I2C.

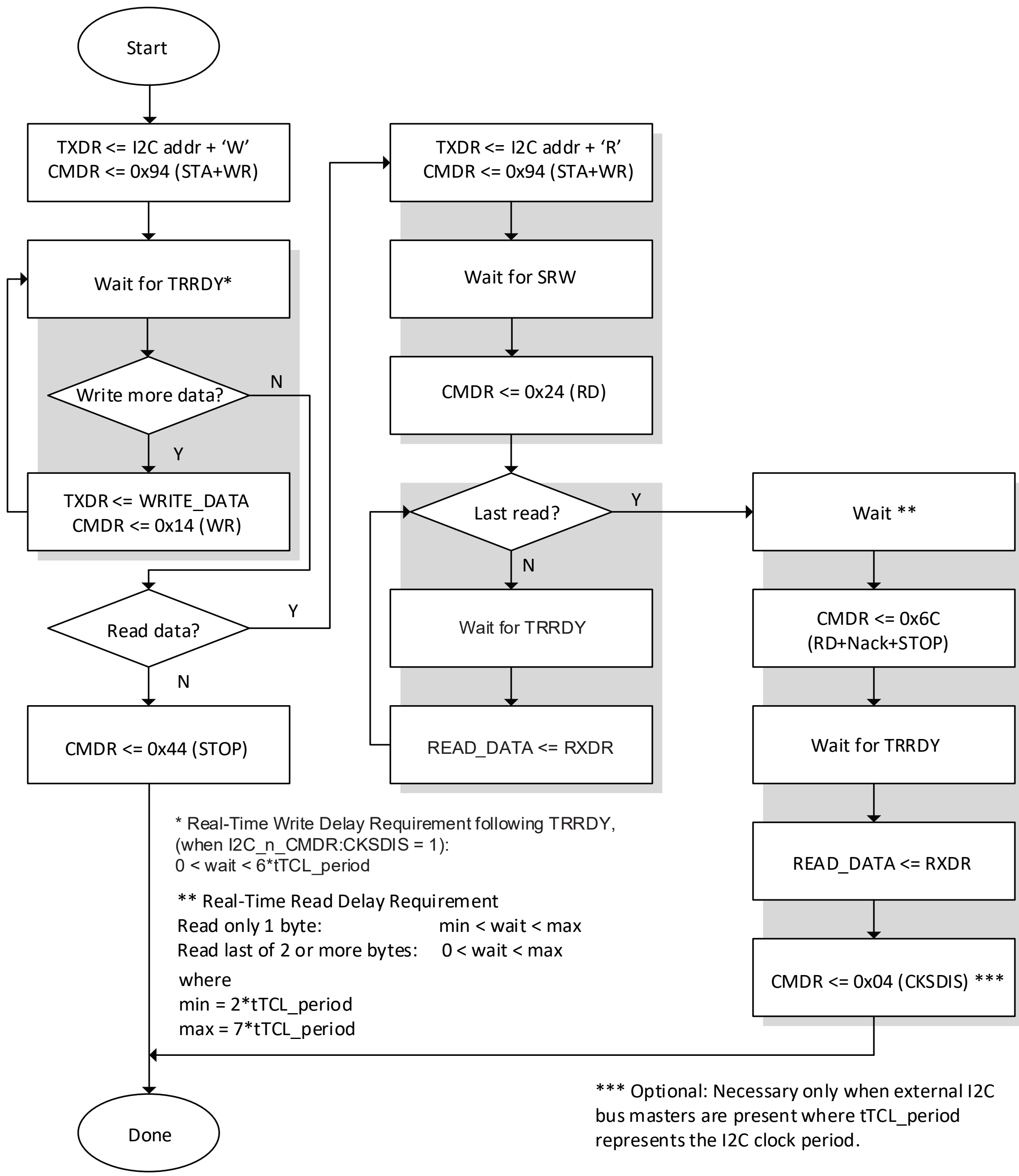


Figure 2.1. I2C Controller Read/Write Example (through WISHBONE)

Figure 2.2 shows a flow diagram for reading and writing from an I2C Target device through the WISHBONE interface. The following sequence is for the Primary I2C but the same sequence applies to the Secondary I2C.

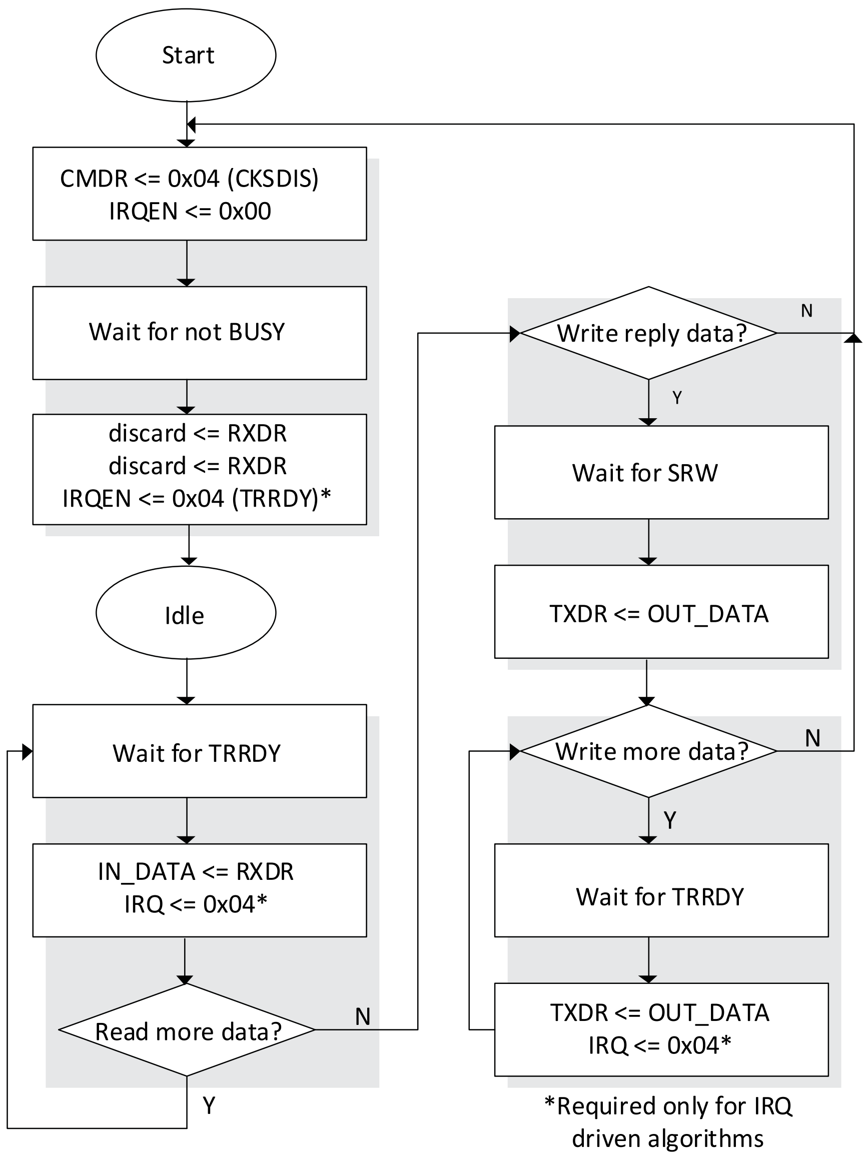


Figure 2.2. I2C Target Read/Write Example (through WISHBONE)

2.2. Typical I2C Transactions

Figure 2.3, Figure 2.4, and Figure 2.5 illustrate typical user I2C bus protocol transactions that are supported by the Controller and Target flows shown in Figure 2.1 and Figure 2.2. Additionally, the figures below reference typical sysConfig configuration command structures.

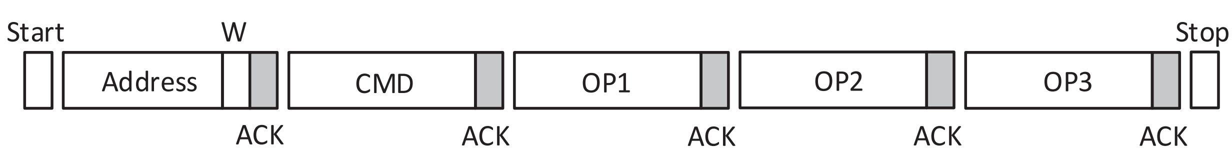


Figure 2.3. Simple I2C Command (for example, ISC_ERASE)

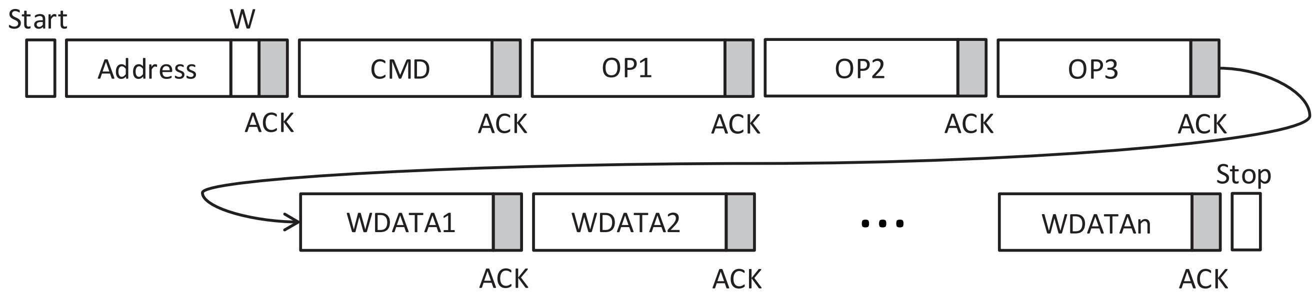


Figure 2.4. I2C Command with Write Data (for example, LSC_PROG_INCR_NV)

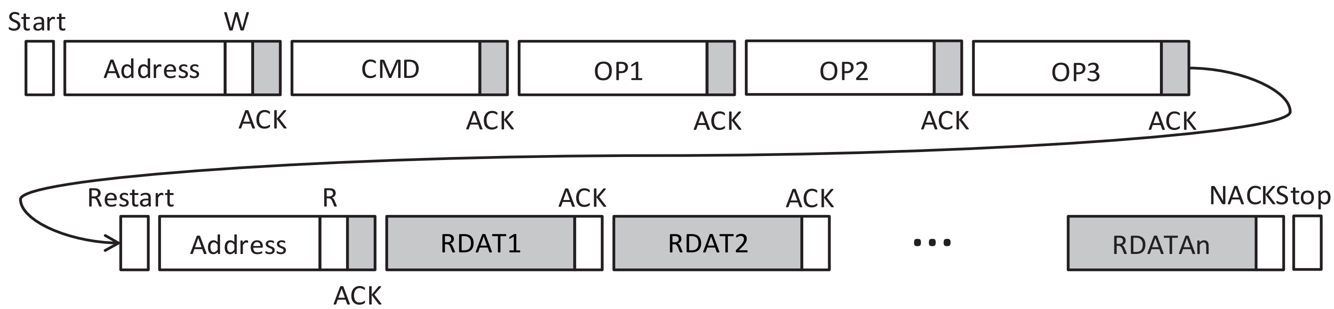


Figure 2.5. I2C Command with Read Data (for example, LSC_READ_STATUS)

2.3. I2C Functional Waveforms

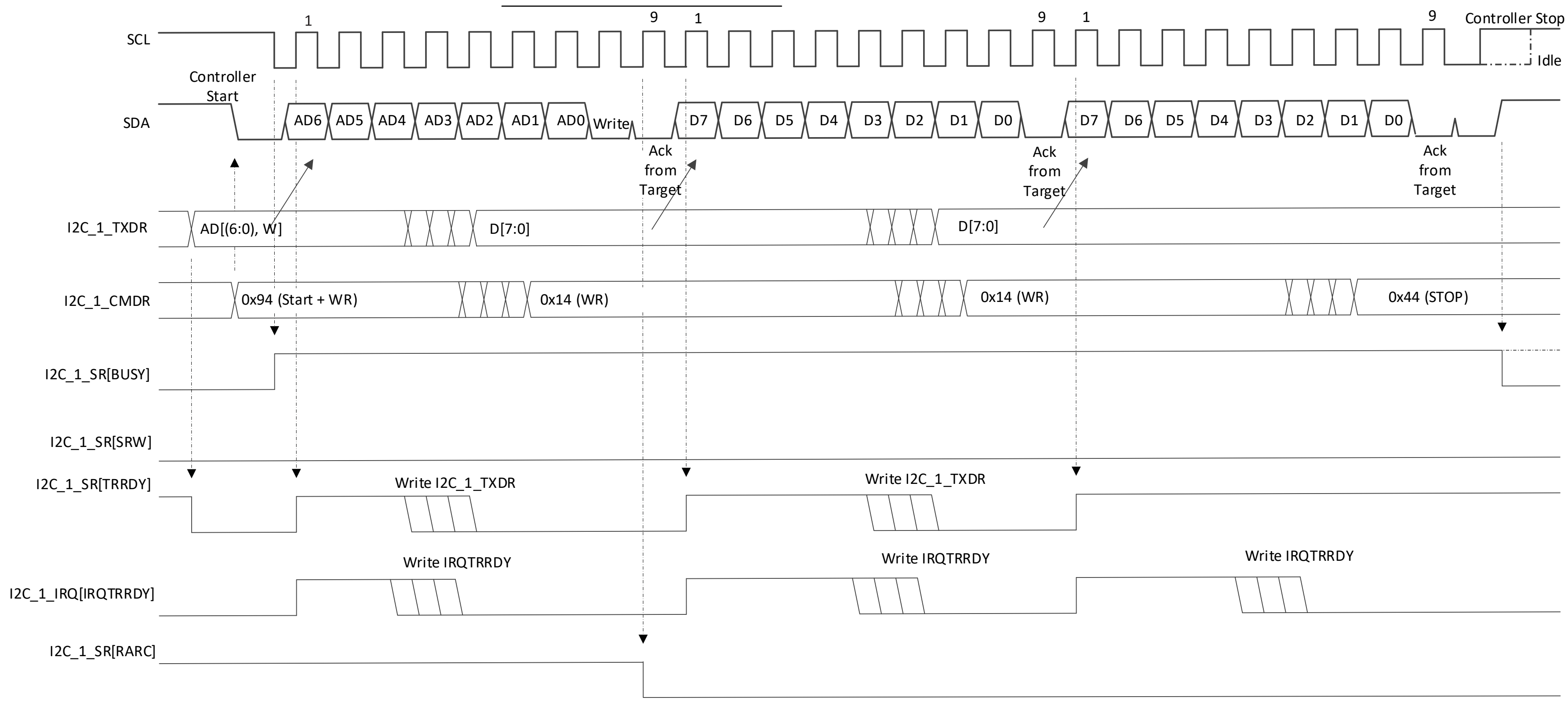


Figure 2.6. EFB Controller – I2C Write

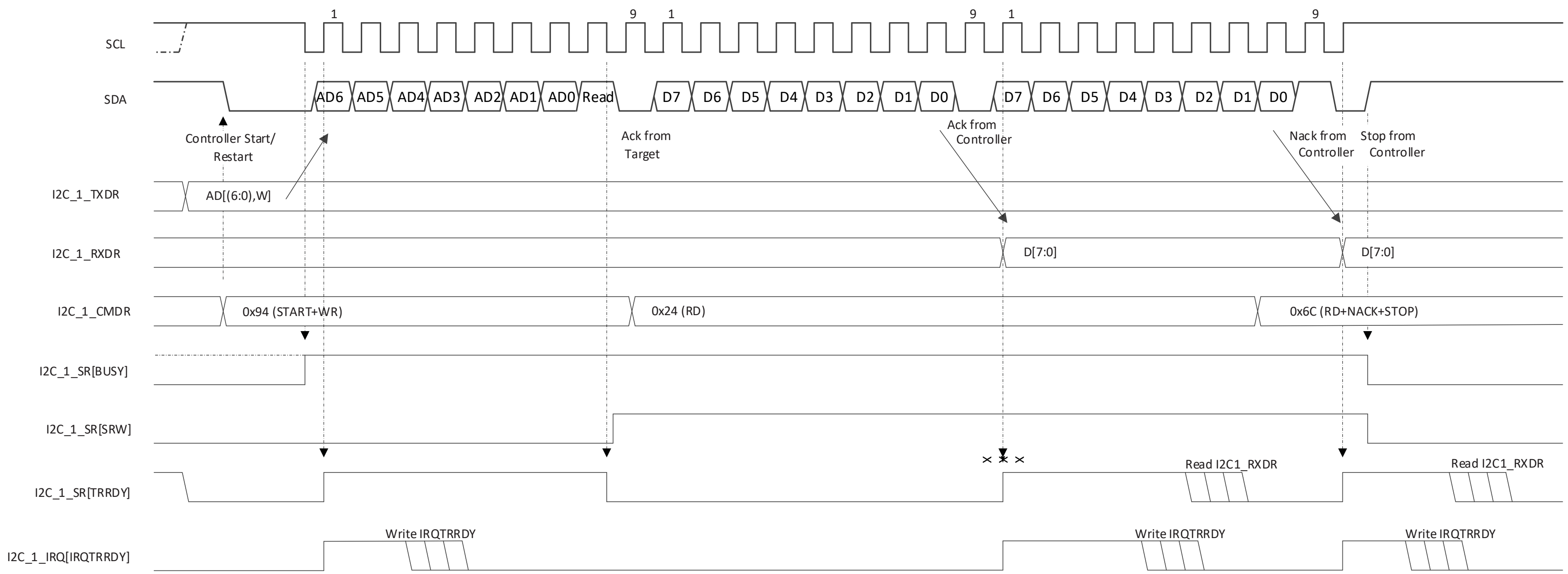


Figure 2.7. EFB Controller – I2C Read

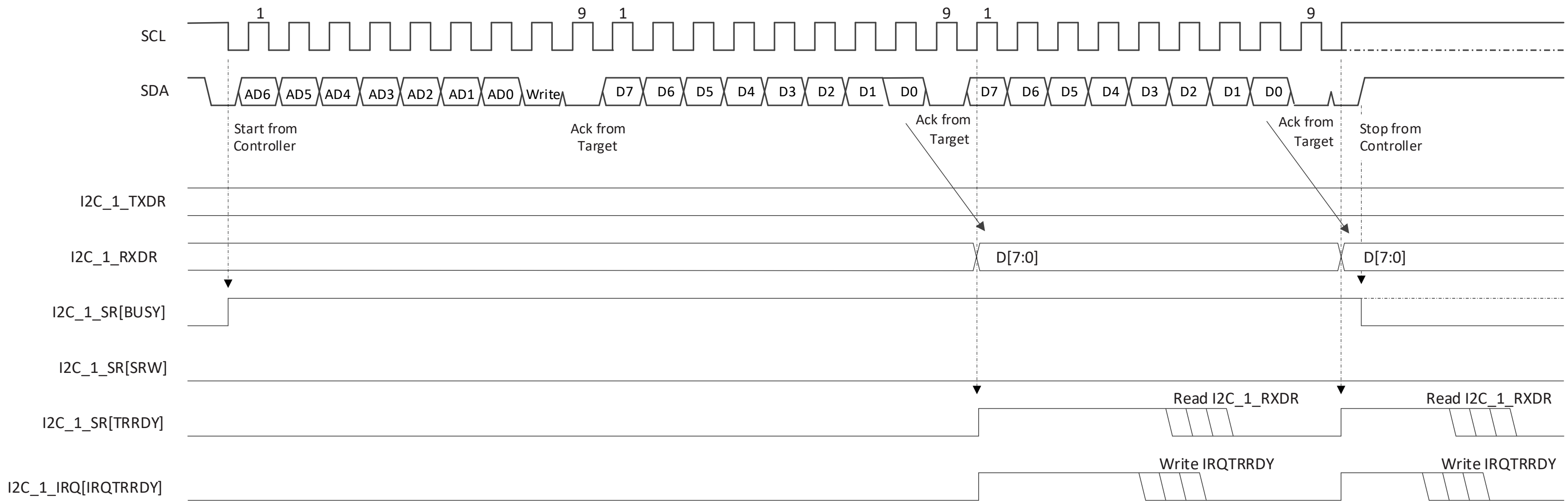


Figure 2.8. EFB Target – I2C Write

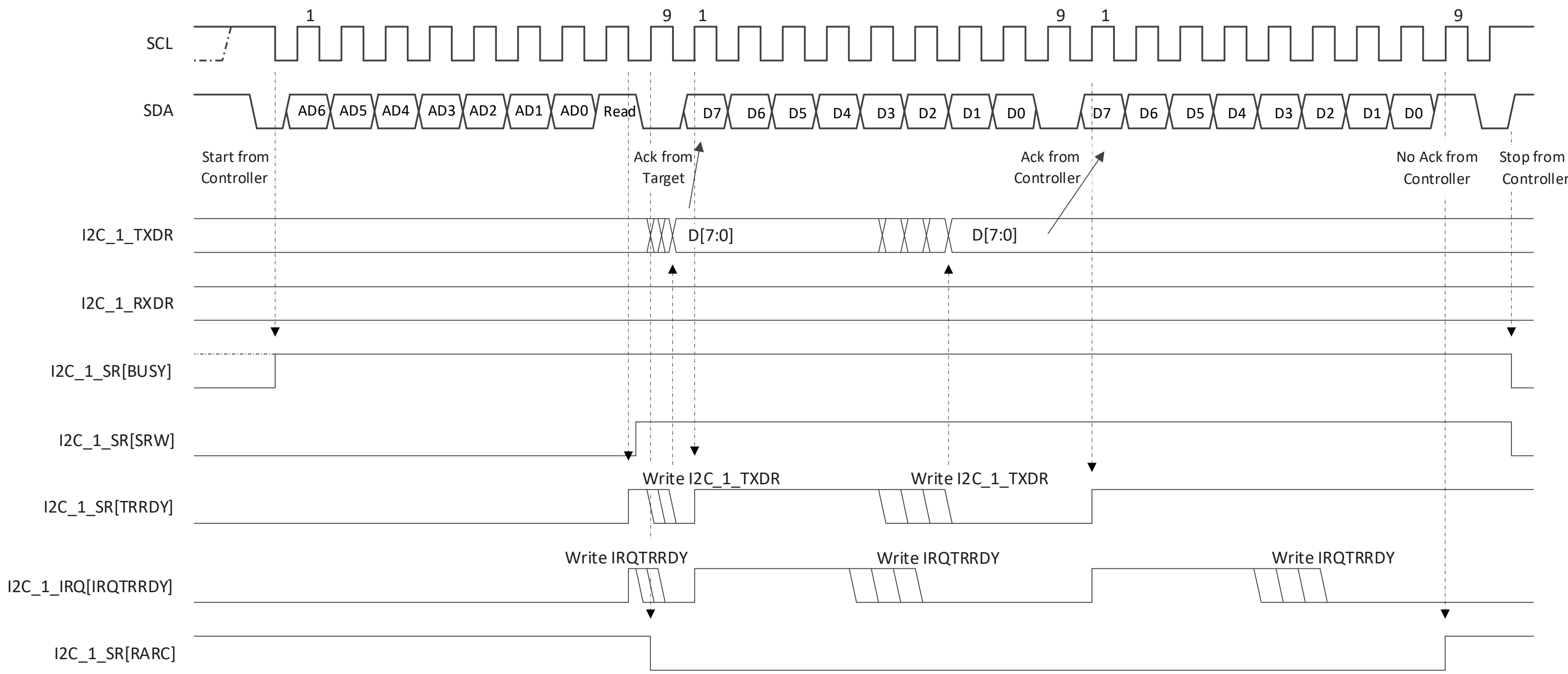


Figure 2.9. EFB Target – I2C Read

2.4. I2C Timing Diagram

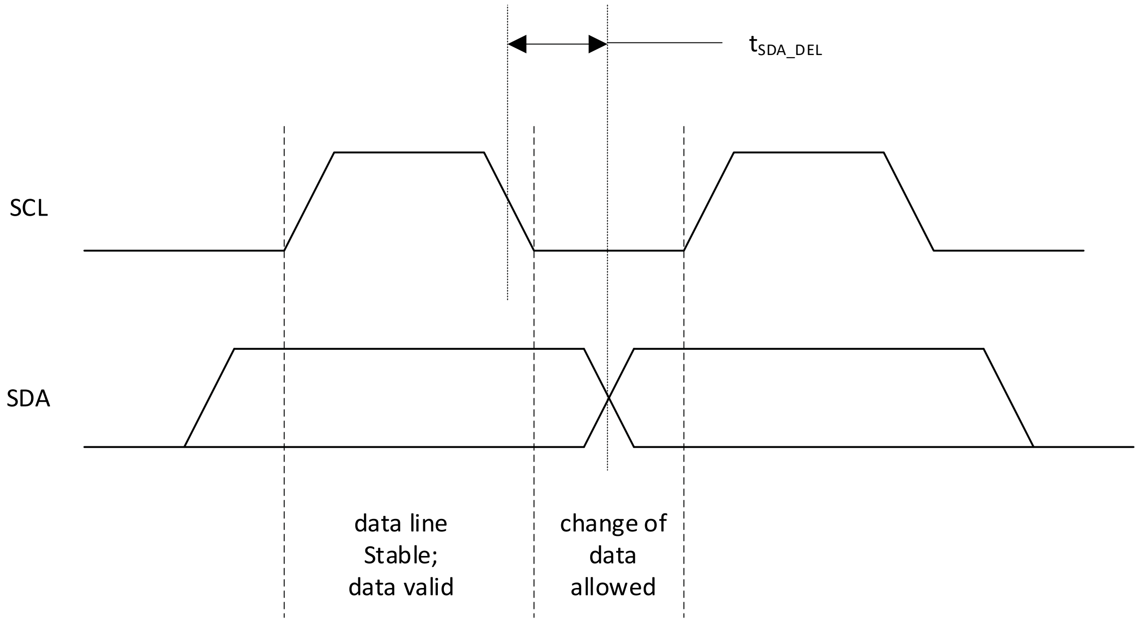


Figure 2.10. I2C Bit Transfer Timing

2.5. I2C Simulation Model

The I2C EFB register map translation to the MachXO4 EFB software simulation model is provided in the tables below.

Table 2.12. I2C Primary Simulation Mode

I2C Primary Register Name	Register Size/Bit Location	Register Function	Address I2C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_1_CR	[7:0]	Control	0x40	Read/Write	i2ccr1[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2CEN	7	—	—	—	i2c_en	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
GCEN	6	—	—	—	i2c_gcen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
WKUPEN	5	—	—	—	i2c_wkupen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
SDA_DEL_SEL[1:0]	[3:2]	—	—	—	sda_del_sel	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_CMDR	[7:0]	Command	0x41	Read/Write	i2ccmdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
STA	7	—	—	—	i2c_sta	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
STO	6	—	—	—	i2c_sto	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
RD	5	—	—	—	i2c_rd	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
WR	4	—	—	—	i2c_wt	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
ACK	3	—	—	—	i2c_nack	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
CKSDIS	2	—	—	—	i2c_cksdis	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_1_BR0	[7:0]	Clock Pre-scale	0x42	Read/Write	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/
I2C_PRESCALE[7:0]	[7:0]	—	—	—	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu

I2C Primary Register Name	Register Size/Bit Location	Register Function	Address I2C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
						/njport_unit/i2c_1st/
I2C_1_BR1	[7:0]	Clock Pre-scale	0x43	Read/Write	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu /njport_unit/i2c_1st/
I2C_PRESCALE[9:8]	[1:0]	—	—	—	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu /njport_unit/i2c_1st/
I2C_1_TXDR	[7:0]	Transmit Data	0x44	Write	i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu /njport_unit/i2c_1st/
I2C_Transmit_Data[7:0]	[7:0]	—	—	—	i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu /njport_unit/i2c_1st/
I2C_1_SR	[7:0]	Status	0x45	Read	i2csr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu /njport_unit/i2c_1st/
TIP	7	—	—	—	i2c_tip_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
BUSY	6	—	—	—	i2c_busy_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
RARC	5	—	—	—	i2c_rarc_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
SRW	4	—	—	—	i2c_srw_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
ARBL	3	—	—	—	i2c_arbl	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
TRRDY	2	—	—	—	i2c_trrdy	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
TROE	1	—	—	—	i2c_troe	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
HGC	0	—	—	—	i2c_hgc	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
I2C_1_GCDR	[7:0]	General Call	—	—	i2cgcdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
I2C_GC_Data[7:0]	[7:0]	—	—	—	i2cgcdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
I2C_1_RXDR	[7:0]	Receive	—	—	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu

I2C Primary Register Name	Register Size/Bit Location	Register Function	Address I2C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
		Data				/ njport_unit/i2c_1st/
I2C_Receive_Data[7:0]	[7:0]	—	—	—	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu / njport_unit/i2c_1st/
I2C_1_IRQ	[7:0]	IRQ	—	—	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_1st_irqsts_3, i2csr_1st_irqsts_2, i2csr_1st_irqsts_1, i2csr_1st_irqsts_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBL	3	—	—	—	i2csr_1st_irqsts_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDY	2	—	—	—	i2csr_1st_irqsts_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROE	1	—	—	—	i2csr_1st_irqsts_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGC	0	—	—	—	i2csr_1st_irqsts_0	../efb_top/efb_pll_sci_inst/u_efb_sci/
I2C_1_IRQEN	[7:0]	IRQ Enable	—	—	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_1st_irqena_3, i2csr_1st_irqena_2, i2csr_1st_irqena_1, i2csr_1st_irqena_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBLEN	3	—	—	—	i2csr_1st_irqena_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDYEN	2	—	—	—	i2csr_1st_irqena_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROEEN	1	—	—	—	i2csr_1st_irqena_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGCEN	0	—	—	—	i2csr_1st_irqena_0	../efb_top/efb_pll_sci_inst/u_efb_sci/

Table 2.13. I2C Secondary Simulation Mode

I2C Secondary Register Name	Register Size/Bit Location	Register Function	Address I2C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_2_CR	[7:0]	Control	0x4A	Read/Write	i2ccr1[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2CEN	7	—	—	—	i2c_en	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
GCEN	6	—	—	—	i2c_gcen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
WKUPEN	5	—	—	—	i2c_wkupen	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
SDA_DEL_SEL[1:0]	[3:2]	—	—	—	sda_del_sel	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_CMDR	[7:0]	Command	0x4B	Read/Write	i2ccmdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
STA	7	—	—	—	i2c_sta	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
STO	6	—	—	—	i2c_sto	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
RD	5	—	—	—	i2c_rd	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
WR	4	—	—	—	i2c_wt	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
ACK	3	—	—	—	i2c_nack	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
CKSDIS	2	—	—	—	i2c_cksdis	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_BR0	[7:0]	Clock Pre-scale	0x4C	Read/Write	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_PRESCALE[7:0]	[7:0]	—	—	—	i2cbr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_BR1	[7:0]	Clock Pre-scale	0x4D	Read/Write	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_PRESCALE[9:8]	[1:0]	—	—	—	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/

I2C Secondary Register Name	Register Size/Bit Location	Register Function	Address I2C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
						njport_unit/i2c_2nd/
I2C_2_TXDR	[7:0]	Transmit Data	0x4E	Write	i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_Transmit_Data[7:0]	[7:0]	—	—	—	i2ctxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_SR	[7:0]	Status	0x4F	Read	i2csr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
TIP	7	—	—	—	i2c_tip_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
BUSY	6	—	—	—	i2c_busy_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
RARC	5	—	—	—	i2c_rarc_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
SRW	4	—	—	—	i2c_srw_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
ARBL	3	—	—	—	i2c_arbl	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
TRRDY	2	—	—	—	i2c_trrdy	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
TROE	1	—	—	—	i2c_troe	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
HGC	0	—	—	—	i2c_hgc	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_GCDR	[7:0]	General Call	0x50	Read	i2cgcdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_GC_Data[7:0]	[7:0]	—	—	—	i2cgcdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_RXDR	[7:0]	Receive Data	0x51	Read	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_Receive_Data[7:0]	[7:0]	—	—	—	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_2nd/
I2C_2_IRQ	[7:0]	IRQ	0x52	Read/Write	{1'b0, 1'b0, 1'b0,	../efb_top/efb_pll_sci_inst/u_efb_sci/

I2C Secondary Register Name	Register Size/Bit Location	Register Function	Address I2C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
					1'b0, i2csr_2nd_irqsts_3, i2csr_2nd_irqsts_2, i2csr_2nd_irqsts_1, i2csr_2nd_irqsts_0}	
IRQARBL	3	—	—	—	i2csr_2nd_irqsts_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDY	2	—	—	—	i2csr_2nd_irqsts_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROE	1	—	—	—	i2csr_2nd_irqsts_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGC	0	—	—	—	i2csr_2nd_irqsts_0	../efb_top/efb_pll_sci_inst/u_efb_sci/
I2C_2_IRQEN	[7:0]	IRQ Enable	0x53	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_2nd_irqena_3, i2csr_2nd_irqena_2, i2csr_2nd_irqena_1, i2csr_2nd_irqena_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQARBLEN	3	—	—	—	i2csr_2nd_irqena_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRRDYEN	2	—	—	—	i2csr_2nd_irqena_2	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTROEEN	1	—	—	—	i2csr_2nd_irqena_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQHGCEN	0	—	—	—	i2csr_2nd_irqena_0	../efb_top/efb_pll_sci_inst/u_efb_sci/

3. Hardened SPI IP Core

The MachXO4 device contains a hard SPI IP core that can be configured as an SPI Controller or Target. When the SPI core is configured as a Controller, it, is able to control other devices with Target SPI interfaces that are connected to the SPI bus. When the SPI core is configured as a Target, it is able to interface with an external SPI Controller device.

3.1. SPI Registers

The SPI core communicates with the WISHBONE interface through a set of control, command, status, and data registers. Table 3.1 shows the register names and their functions. These registers are a subset of the EFB register map.

Table 3.1. SPI Registers

SPI Register Name	Register Function	Address	Access
SPICR0	Control Register 0	0x54	Read/Write
SPICR1	Control Register 1	0x55	Read/Write
SPICR2	Control Register 2	0x56	Read/Write
SPIBR	Clock Pre-scale	0x57	Read/Write
SPICSR	Controller Chip Select	0x58	Read/Write
SPITXDR	Transmit Data	0x59	Write
SPISR	Status	0x5A	Read
SPIRXDR	Receive Data	0x5B	Read
SPIIRQ	Interrupt Request	0x5C	Read/Write
SPIIRQEN	Interrupt Request Enable	0x5D	Read/Write

Note: Unless otherwise specified, all Reserved bits in writable registers are written as *0*.

Table 3.2. SPI Control 0

SPICR0								0x54
Bit	7	6	5	4	3	2	1	0
Name	TIdle_XCNT[1:0]		TTrail_XCNT[2:0]			TLead_XCNT[2:0]		
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: A write to this register causes the SPI core to reset.

TIdle_XCNT[1:0]

Idle Delay Count. Specifies the minimum interval prior to the Controller Chip Select low assertion (Controller Mode only), in SCK periods.

00: ½

01: 1

10: 1.5

11: 2

TTrail_XCNT[2:0]

Trail Delay Count. Specifies the minimum interval between the last edge of SCK and the high de-assertion of Controller Chip Select (Controller Mode only), in SCK periods.

000: ½

001: 1

010: 1.5

…

111: 4

TLead_XCNT[2:0]

Lead Delay Count. Specifies the minimum interval between the Controller Chip Select low assertion and the first edge of SCK (Controller Mode only), in SCK periods.

000: ½

001: 1

010: 1.5

…

111: 4

Table 3.3. SPI Control 1

SPICR1								0x55
Bit	7	6	5	4	3	2	1	0
Name	SPE	WKUPEN_USER	WKUPEN_CFG	TXEDGE	(Reserved)			
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	—	—	—	—

Note: A write to this register causes the SPI core to reset.

SPE

This bit enables the SPI core functions. If SPE is cleared, SPI is disabled and forced into idle state.

0: SPI disabled.

1: SPI enabled, port pins are dedicated to SPI functions.

WKUPEN_USER

Wake-up Enable through User. Enables the SPI core to send a wake-up signal to the on-chip power controller to wake the part from standby mode when the User target SPI chip select (spi_scsn) is driven low.

0: Wakeup disabled.

1: Wakeup enabled.

WKUPEN_CFG

Wake-up Enable Configuration. Enables the SPI core to send a wake-up signal to the on-chip power controller to wake the part from standby mode when the Configuration target SPI chip select (ufm_sn) is driven low.

0: Wakeup disabled.

1: Wakeup enabled.

TXEDGE

Data Transmit Edge. Enables Lattice proprietary extension to the SPI protocol. Selects the clock edge to transmit SPI data. Refer to Figure 3.9 – Figure 3.12.

0: Transmit data on the MCLK/CCLK edge defined by SPICR2[CPOL] and SPICR2[CPHA].

1: Transmit data ½ MCLK/CCLK earlier than defined by SPICR2[CPOL] and SPICR2[CPHA].

Table 3.4. SPI Control 2

SPICR2								0x56
Bit	7	6	5	4	3	2	1	0
Name	MSTR	MCSH	SDBRE	(Reserved)	(Reserved)	CPOL	CPHA	LSBF
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	—	—	—	—	—

Note: A write to this register causes the SPI core to reset.

MSTR

SPI Controller/Target Mode. Selects the Controller/Target operation mode of the SPI core. Changing this bit forces the SPI system into idle state.

0: SPI is in Target mode.

1: SPI is in Controller mode.

MCSH

SPI Controller CSSPIN Hold. Holds the Controller chip select active when the host is busy, to halt the data transmission without de-asserting chip select.
Note: This mode must be used only when the WISHBONE clock has been divided by a value greater than four (4).

0: Controller running as normal.

1: Controller holds chip select low even if there is no data to be transmitted.

SDBRE

Target Dummy Byte Response Enable. Enables Lattice proprietary extension to the SPI protocol. For use when the internal support circuit (for example, WISHBONE host) cannot respond with initial data within the time required, and to make the target read out data predictably available at high SPI clock rates.

When enabled, dummy 0xFF bytes are transmitted in response to an SPI target read (while SPISR[TRDY]=1) until an initial write to SPITXDR. Once a byte is written into SPITXDR by the WISHBONE host, a single byte of 0x00 is transmitted then followed immediately by the data in SPITXDR. In this mode, the external SPI controller scans for the initial 0x00 byte when reading the SPI target to indicate the beginning of actual data. Refer to Figure 3.7.

0: Normal Target SPI operation.

1: Lattice proprietary Target Dummy Byte Response Enabled.
Note: This mechanism only applies to the initial data delay period. Once the initial data is available, subsequent data must be supplied to SPITXDR at the required SPI bus data rate.

CPOL

SPI Clock Polarity. Selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical SPICR2[CPOL] values. In Controller mode, a change of this bit aborts a transmission in progress and forces the SPI system into idle state. Refer to Figure 3.9 – Figure 3.12.

0: Active-high clocks selected.

1: Active-low clocks selected.

CPHA

SPI Clock Phase. Selects the SPI clock format. In Controller mode, a change of this bit aborts a transmission in progress and forces the SPI system into idle state. Refer to Figure 3.9 – Figure 3.12.

0: Data is captured on a leading (first) clock edge, and propagated on the opposite clock edge.

1: Data is captured on a trailing (second) clock edge, and propagated on the opposite clock edge.
Note: When CPHA=1, you must explicitly place a pull-up or pull-down on SCK pad corresponding to the value of CPOL (for example, when CPHA=1 and CPOL=0 place a pull-down on SCK). When CPHA=0, the pull direction may be set arbitrarily.

Target SPI Configuration mode supports default setting only for CPOL and CPHA.

LSBF

LSB-First. LSB appears first on the SPI interface. In Controller mode, a change of this bit aborts a transmission in progress and forces the SPI system into idle state. Refer to Figure 3.9 – Figure 3.12.
Note: This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in bit 7.

0: Data is transferred, most significant bit (MSB) first.

1: Data is transferred, least significant bit (LSB) first.

Table 3.5. SPI Clock Pre-scale

SPIBR								0x57
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)		DIVIDER[5:0]					
Default[1]	0	0	0	0	0	0	0	0
Access	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Note:

1. Hardware default value may be overridden by EFB component instantiation parameters. Refer to the discussion below.

DIVIDER[5:0]

SPI Clock Pre-scale value. The WISHBONE clock frequency is divided by (DIVIDER[5:0] + 1) to produce the desired SPI clock frequency. A write operation to this register causes an SPI core reset. The DIVIDER value must be greater than 1. This implies that the WISHBONE clock must run at least three times faster than the target SPI clock.
Note: The digital value is calculated by the IP Catalog when the SPI core is configured in the SPI tab of the EFB GUI. The calculation is based on the WISHBONE Clock Frequency and the SPI Frequency, which are user-entered. The digital value of the divider is programmed in the MachXO4 device during device programming. After power-up or device reconfiguration, the data is loaded onto the SPIBR register.

Register SPIBR has Read/Write access from the WISHBONE interface. You can update the clock pre-scale register dynamically during device operation.

Table 3.6. SPI Controller Chip Select

SPICSR								0x58
Bit	7	6	5	4	3	2	1	0
Name	CSN_7	CSN_6	CSN_5	CSN_4	CSN_3	CSN_2	CSN_1	CSN_0
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CSN_[7:0]

SPI Controller Chip Selects. Used in controller mode for asserting a specific Controller Chip Select line. The register has eight bits, enabling the SPI core to control up to eight external SPI target devices. Each bit represents one controller chip select line (Active-Low). Bits [7:1] may be connected to any I/O pin through the FPGA fabric. Bit 0 has a pre-assigned pin location. The register has Read/Write access from the WISHBONE interface. A write operation on this register causes the SPI core to reset.

Table 3.7. SPI Transmit Data Register

SPITXDR								0x59
Bit	7	6	5	4	3	2	1	0
Name	SPI_Transmit_Data[7:0]							
Default	—	—	—	—	—	—	—	—
Access	W	W	W	W	W	W	W	W

SPI_Transmit_Data[7:0]

SPI Transmit Data. This register holds the byte to be transmitted on the SPI bus. Bit 0 in this register is the LSB and is transmitted last when SPICR2[LSBF]=0 or first when SPICR2[LSBF]=1.
Note: When operating as a Target, SPITXDR must be written when SPISR[TRDY] is *1* and at least 0.5 CCLKs before the first bit is to appear on SO. For example, when CPOL = CPHA = TXEDGE = LSBF = 0, SPITXDR must be written prior to the CCLK rising edge used to sample the LSB (bit 0) of the previous byte. Refer to Figure 5.1. This timing requires the inclusion of at least one protocol dummy byte in all target SPI read operations.

Table 3.8. SPI Status

SPISR								0x5A
Bit	7	6	5	4	3	2	1	0
Name	TIP	(Reserved)		TRDY	RRDY	(Reserved)	ROE	MDF
Default	0	—	—	0	0	—	0	0
Access	R	—	—	R	R	—	R	R

TIP

SPI Transmitting In Progress. Indicates the SPI port is actively transmitting/receiving data.

0: SPI Transmitting completed.

1: SPI Transmitting in progress.

TRDY

SPI Transmit Ready. Indicates the SPI transmit data register (SPITXDR) is empty. This bit is cleared by a write to SPITXDR. This bit is capable of generating an interrupt.

0: SPITXDR is not empty.

1: SPITXDR is empty.

RRDY

SPI Receive Ready. Indicates the receive data register (SPIRXDR) contains valid receive data. This bit is cleared by a read access to SPIRXDR. This bit is capable of generating an interrupt.

0: SPIRXDR does not contain data.

1: SPIRXDR contains valid receive data.

ROE

Receive Overrun Error. Indicates SPIRXDR receives new data before the previous data is read. The previous data is lost. This bit is capable of generating an interrupt.

0: Normal.

1: Receiver Overrun detected.

MDF

Mode Fault. Indicates the Target SPI chip select (spi_scsn) is driven low while SPICR2[MSTR]=1. This bit is cleared by any write to SPICR0, SPICR1 or SPICR2. This bit is capable of generating an interrupt.

0: Normal.

1: Mode Fault detected.

Table 3.9. SPI Receive Data Register

SPIRXDR								0x5B
Bit	7	6	5	4	3	2	1	0
Name	SPI_Receive_Data[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

SPI_Receive_Data[7:0]

SPI Receive Data. This register holds the byte captured from the SPI bus. Bit 0 in this register is the LSB and is received last when LSBF=0 or first when LSBF=1.

Table 3.10. SPI Interrupt Status

SPIIRQ								0x5C
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)			IRQTRDY	IRQRRDY	(Reserved)	IRQROE	IRQMDF
Default	—	—	—	0	0	—	0	0
Access	—	—	—	R/W	R/W	—	R/W	R/W

IRQTRDY

Interrupt Status for SPI Transmit Ready. When enabled, indicates SPISR[TRDY] is asserted. Write a *1* to this bit to clear the interrupt.

1: SPI Transmit Ready Interrupt.

0: No interrupt.

IRQRRDY

Interrupt Status for SPI Receive Ready. When enabled, indicates SPISR[RRDY] is asserted. Write a *1* to this bit to clear the interrupt.

1: SPI Receive Ready Interrupt.

0: No interrupt.

IRQROE

Interrupt Status for Receive Overrun Error. When enabled, indicates ROE is asserted. Write a *1* to this bit to clear the interrupt.

1: Receive Overrun Error Interrupt.

0: No interrupt.

IRQMDF

Interrupt Status for Mode Fault. When enabled, indicates MDF is asserted. Write a *1* to this bit to clear the interrupt.

1: Mode Fault Interrupt.

0: No interrupt.

Table 3.11. SPI Interrupt Enable

SPIIRQEN								0x5D
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)			IRQTRDYEN	IRQRRDYEN	(Reserved)	IRQROEEN	IRQMDFEN
Default	—	—	—	0	0	—	0	0
Access	—	—	—	R/W	R/W	—	R/W	R/W

IRQTRDYEN

Interrupt Enable for SPI Transmit Ready.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQRRDYEN

Interrupt Enable for SPI Receive Ready.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQROEEN

Interrupt Enable for Receive Overrun Error.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQMDFEN

Interrupt Enable for Mode Fault.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

Figure 3.1 shows a flow diagram for controlling Controller SPI reads and writes initiated through the WISHBONE interface.

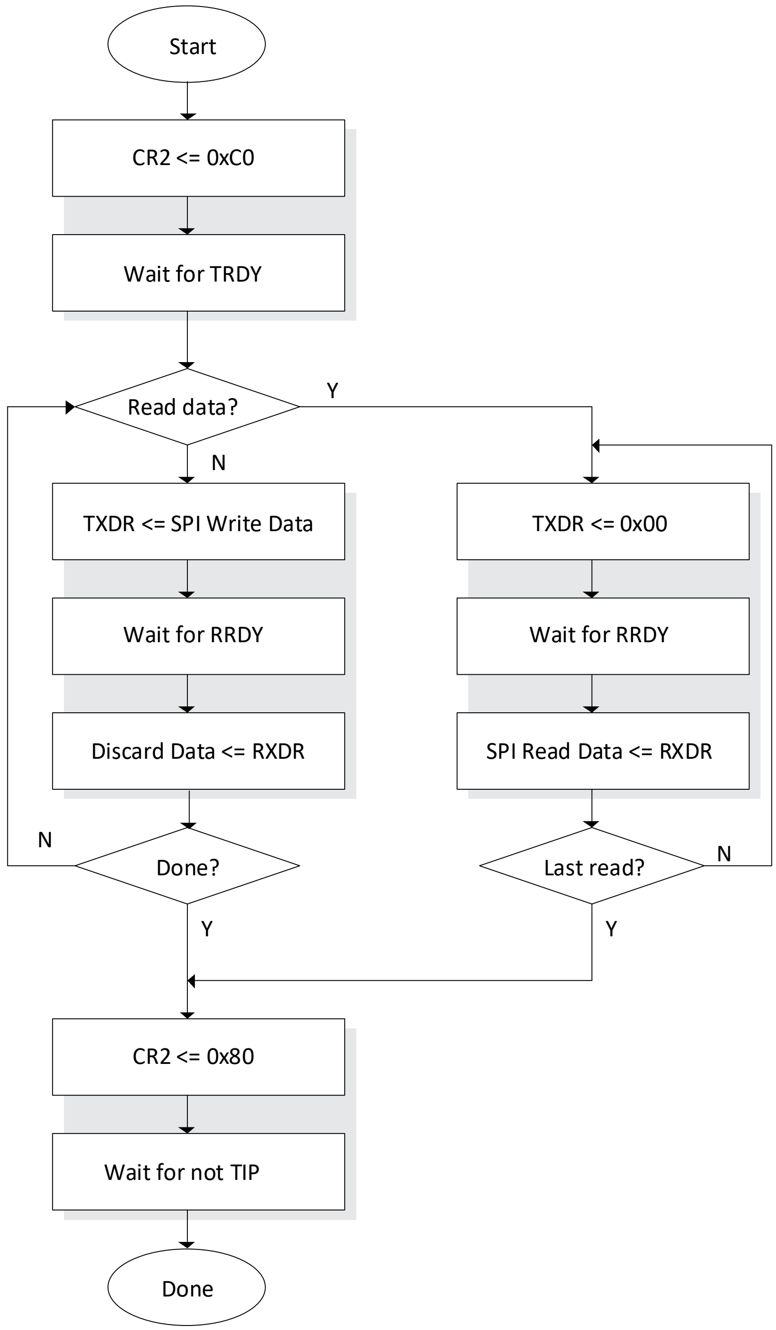


Figure 3.1. SPI Controller Read/Write Example (through WISHBONE) – Production Silicon

Note: Assumes CR2 register, MSCH = *1*. The algorithm when MSCH = *0* is application dependent and not provided. Refer to Figure 3.6 for more information.

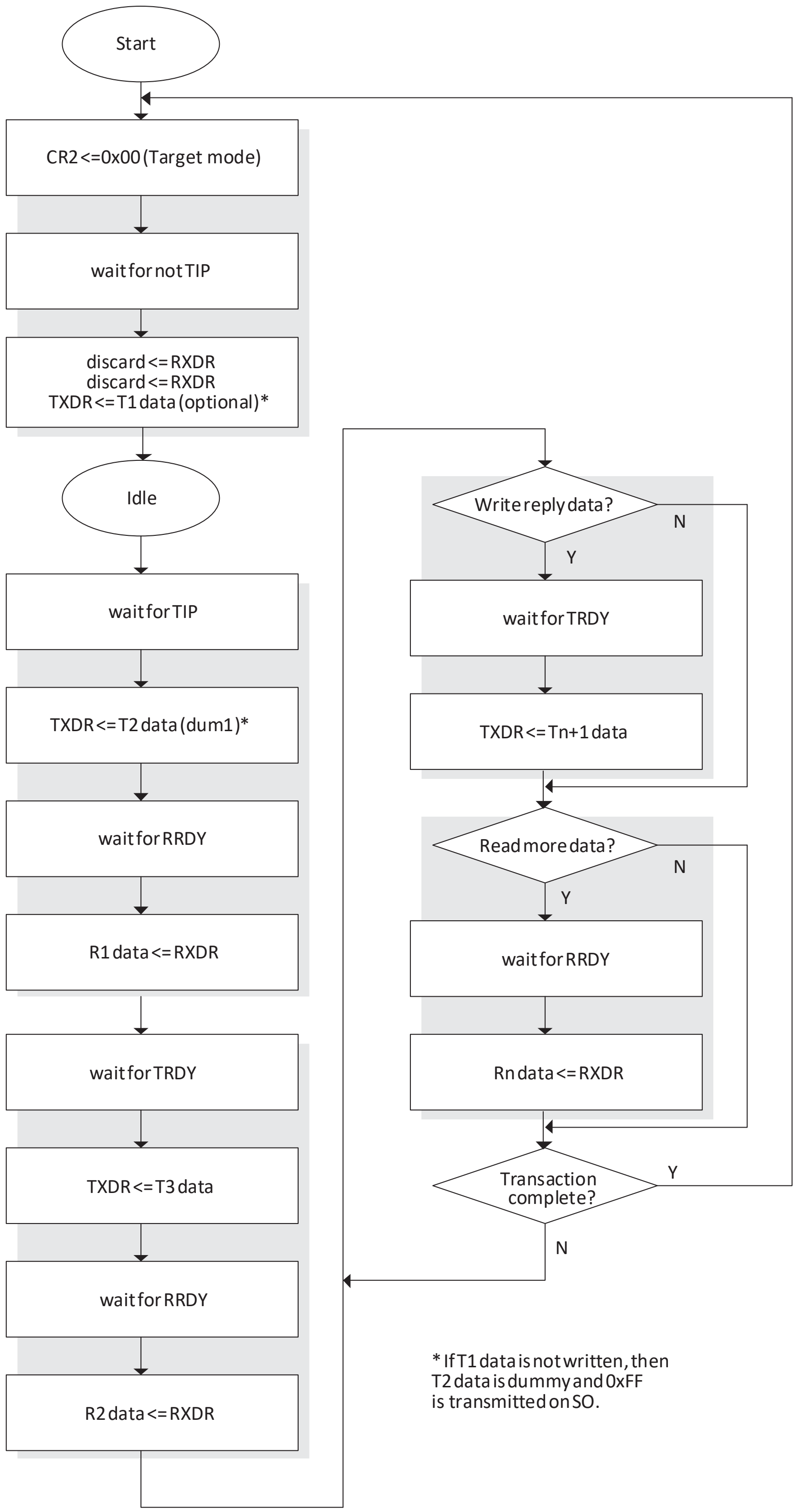


Figure 3.2. SPI Target Read/Write Example (through WISHBONE) – Production Silicon

3.2. Typical SPI Transactions

Figure 3.3, Figure 3.4, and Figure 3.5 illustrate typical user SPI bus protocol transactions that are supported by the Controller and Target flows shown in Figure 3.1 and Figure 3.2. Additionally, the figures below show typical sysConfig configuration command structures.

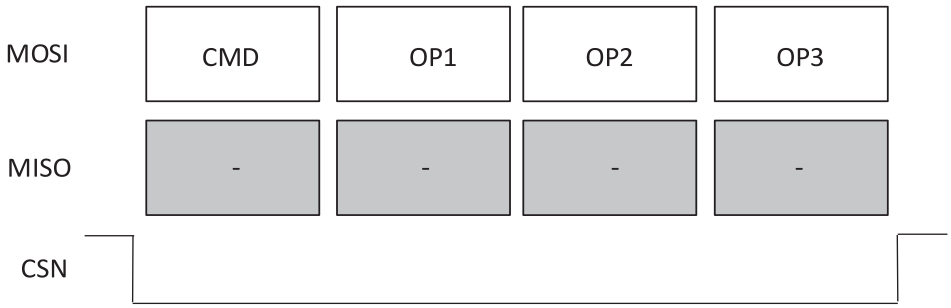


Figure 3.3. Simple SPI Command (for example, ISC_ERASE)

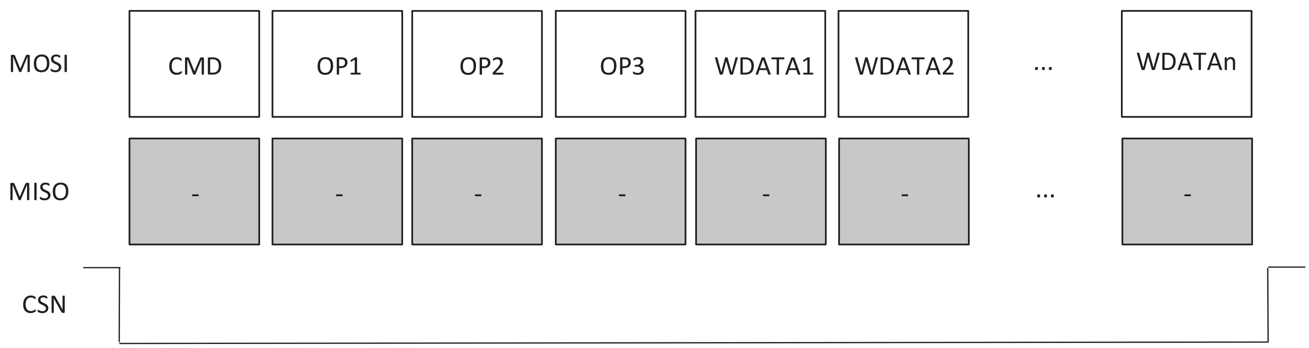


Figure 3.4. SPI Command with Write Data (for example, LSC_PROG_INCR_NV)

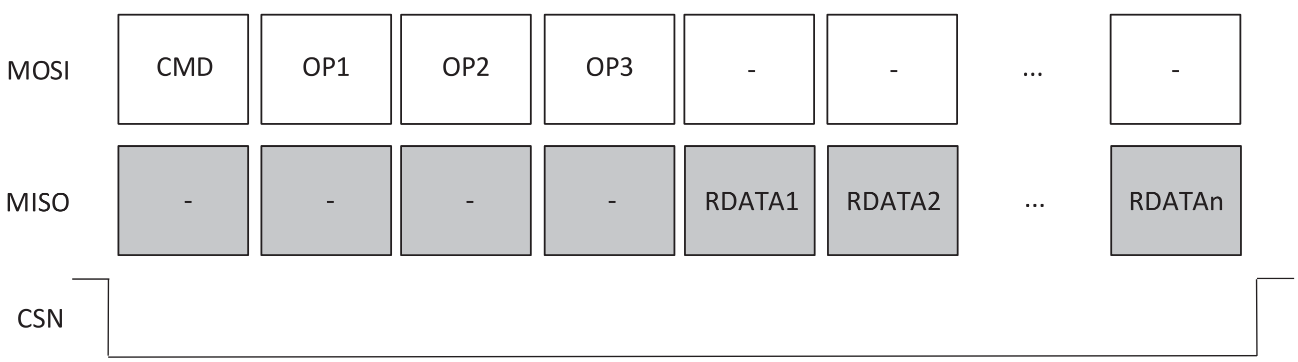


Figure 3.5. SPI Command with Read Data (for example, LSC_READ_STATUS)

3.3. SPI Functional Waveforms

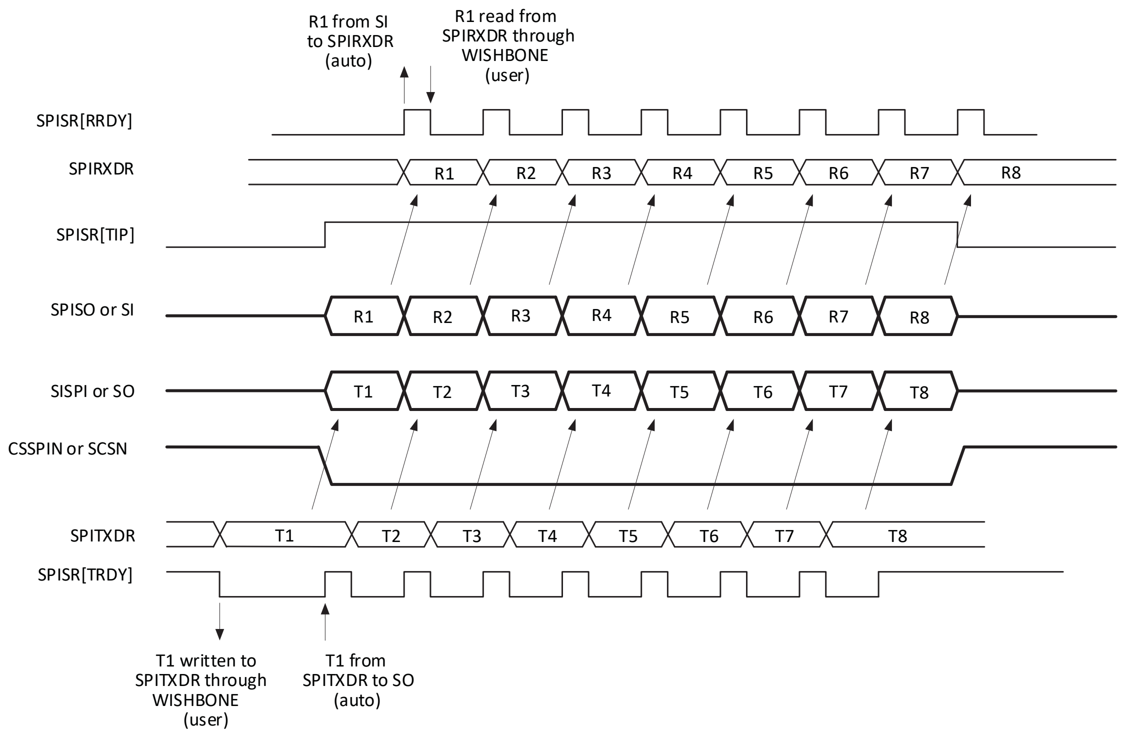


Figure 3.6. Fully Specified SPI Transaction (MachXO4 Device as SPI Controller or Target)

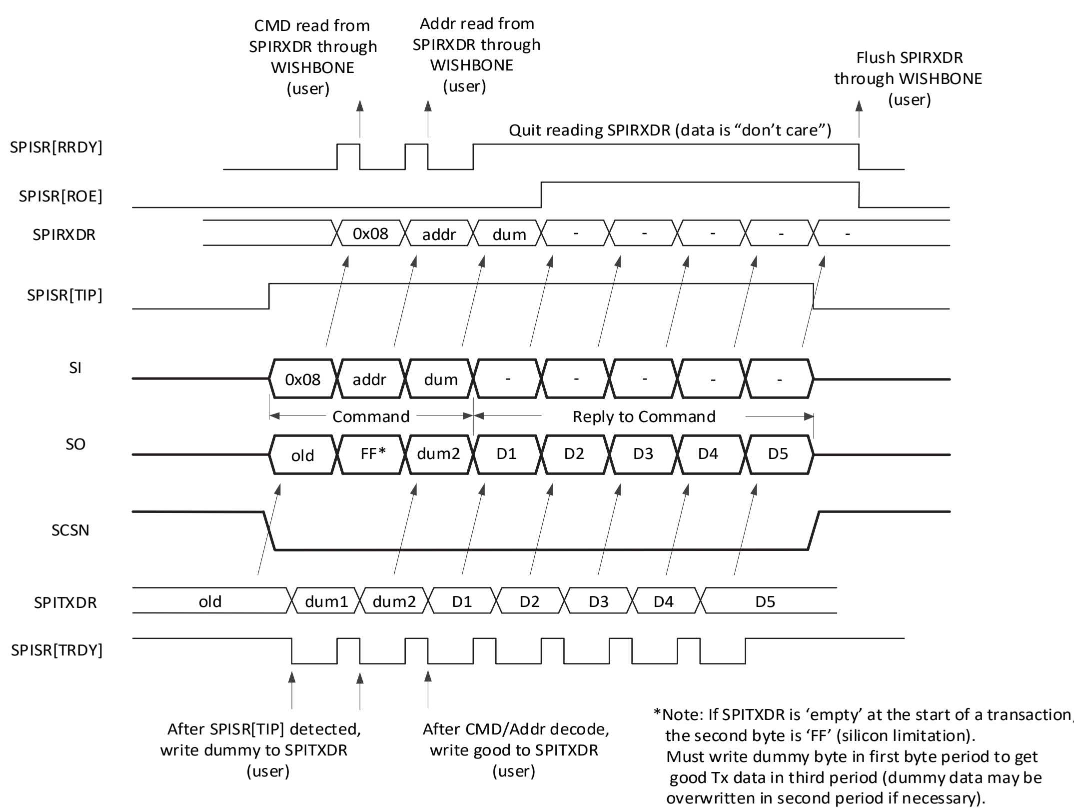


Figure 3.7. Minimally Specified SPI Transaction Example (MachXO4 Device as SPI Target)

3.4. SPI Timing Diagrams

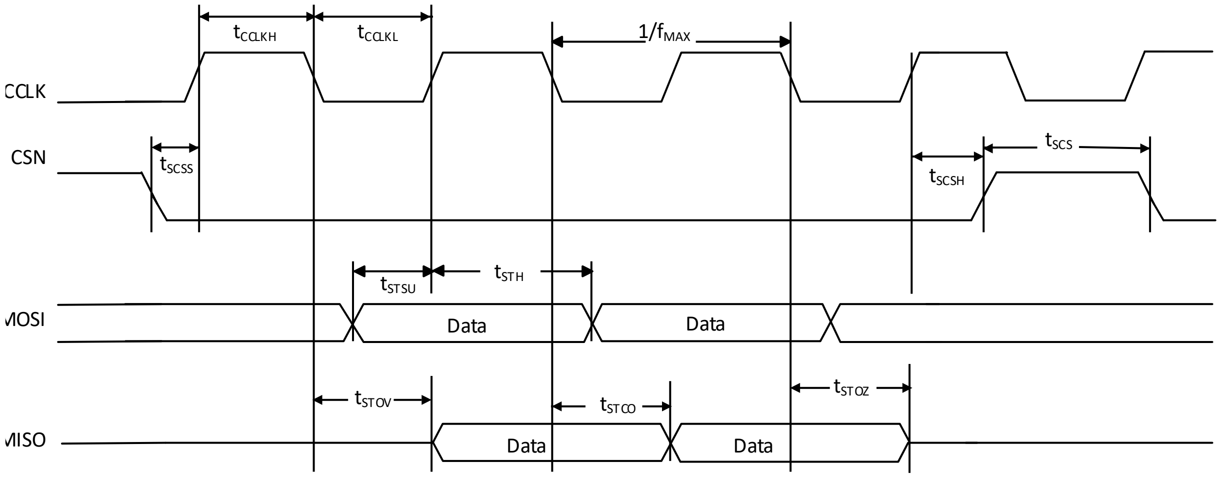


Figure 3.8. Target SPI Configuration Timing

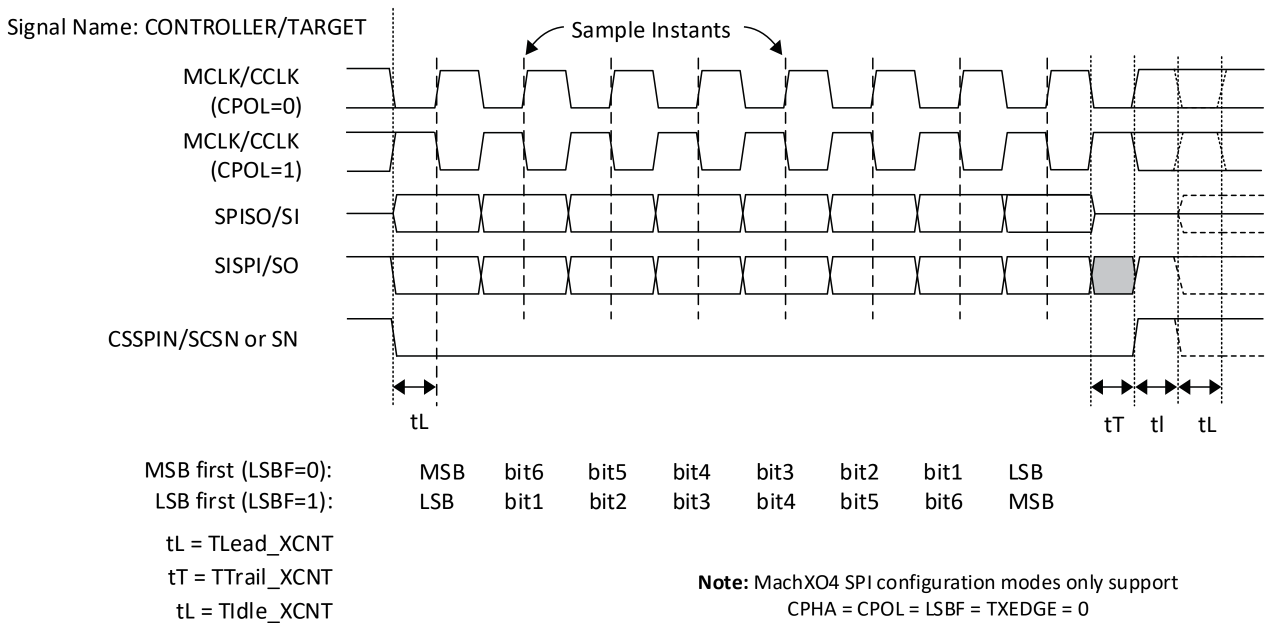


Figure 3.9. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)

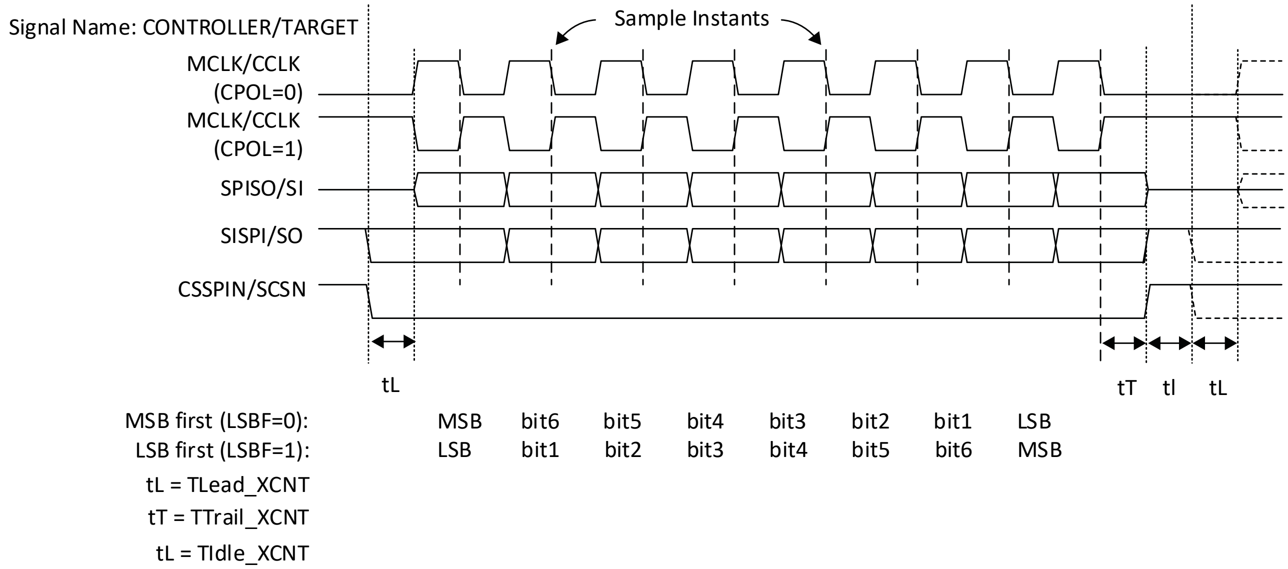


Figure 3.10. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)

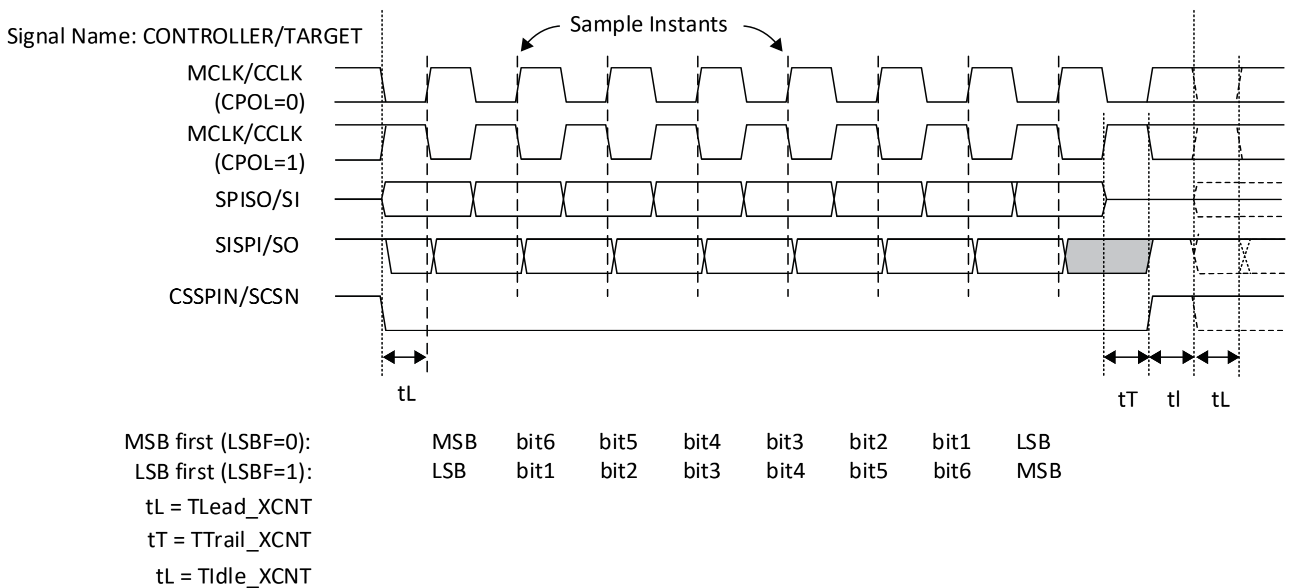


Figure 3.11. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)

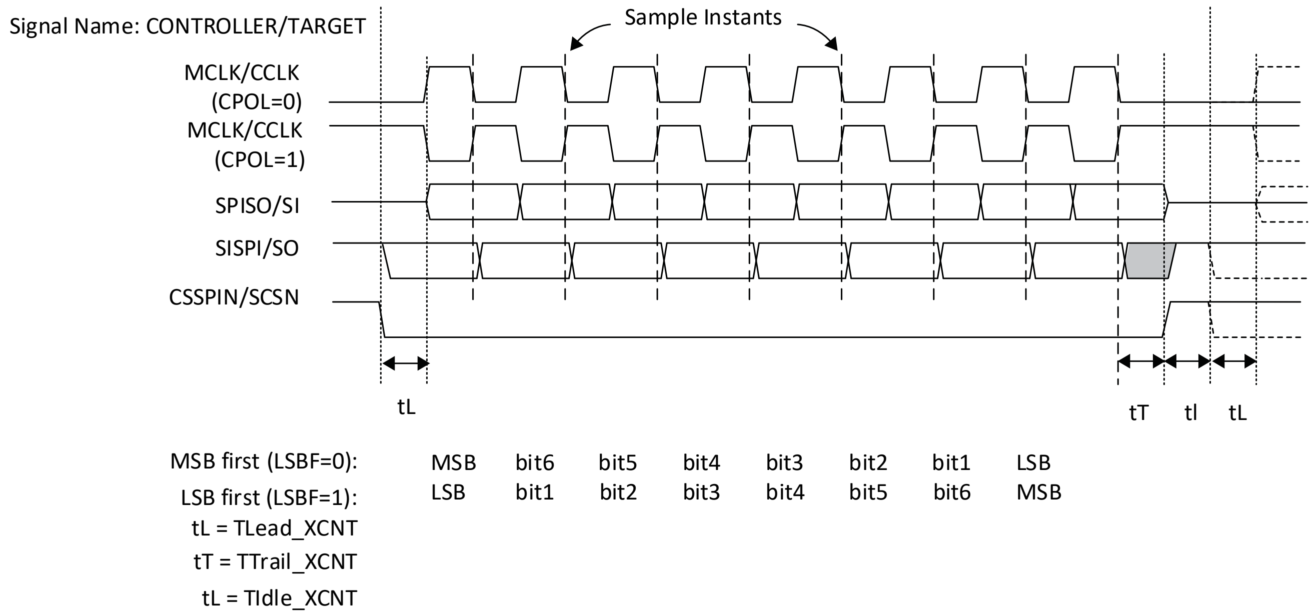


Figure 3.12. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)

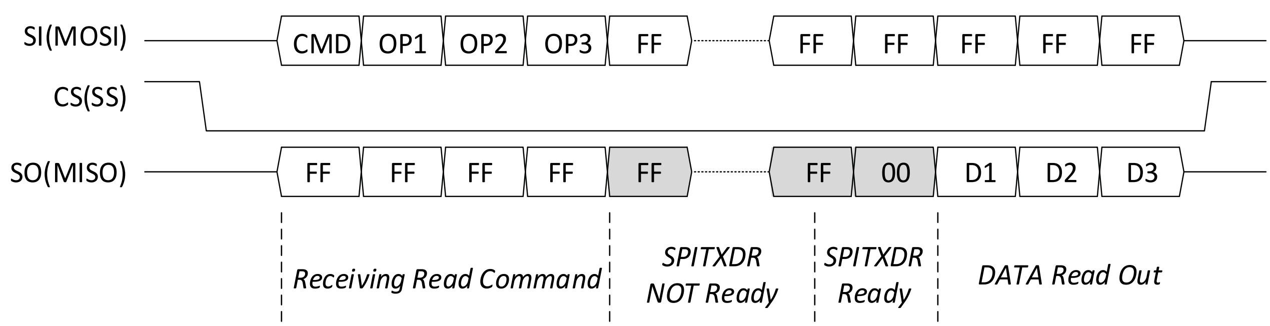


Figure 3.13. Target SPI Dummy Byte Response (SPICR2[SDBRE]) Timing

3.5. SPI Simulation Model

The SPI EFB register map translation to the MachXO4 EFB software simulation model is provided below.

Table 3.1. SPI Simulation Model

SPI Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
SPICR0	[7:0]	Control Register 0	0x54	Read/Write	spicr0[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TIdle_XCNT[1:0]	[7:6]	—	—	—	spicr0[7:6]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TTrail_XCNT[2:0]	[5:3]	—	—	—	spicr0[5:3]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TLead_XCNT[2:0]	[2:0]	—	—	—	spicr0[2:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPICR1	[7:0]	Control Register 1	0x55	Read/Write	spicr1[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPE	7	—	—	—	spi_en	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
WKUPEN_USER	6	—	—	—	spi_wkup_usr	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
WKUPEN_CFG	5	—	—	—	spi_wkup_cfg	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TXEDGE	4	—	—	—	spi_tx_edge	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPICR2	[7:0]	Control Register 2	0x56	Read/Write	spicr2[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
MSTR	7	—	—	—	spi_mstr	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
MCSH	6	—	—	—	spi_mcsh	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SDBRE	5	—	—	—	spi_srme	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CPOL	2	—	—	—	spi_cpol	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CPHA	1	—	—	—	spi_cpha	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
LSBF	0	—	—	—	spi_lsbf	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPIBR	[7:0]	Clock Pre-scale	0x57	Read/Write	spibr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
DIVIDER[5:0]	[5:0]	—	—	—	spibr[5:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPICSR	[7:0]	Controller Chip Select	0x58	Read/Write	spicsr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/

SPI Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
CSN_7	7	—	—	—	spicsr[7]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_6	6	—	—	—	spicsr[6]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_5	5	—	—	—	spicsr[5]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_4	4	—	—	—	spicsr[4]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_3	3	—	—	—	spicsr[3]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_2	2	—	—	—	spicsr[2]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_1	1	—	—	—	spicsr[1]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
CSN_0	0	—	—	—	spicsr[0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPITXDR	[7:0]	Transmit Data	0x59	Write	spitxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPI_Transmit_Data[7:0]	[7:0]	—	—	—	spitxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPISR	[7:0]	Status	0x5A	Read	spisr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TIP	7	—	—	—	spi_tip_sync	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
TRDY	4	—	—	—	spi_trdy	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
RRDY	3	—	—	—	spi_rrdy	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
ROE	1	—	—	—	spi_roe	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
MDF	0	—	—	—	spi_mdf	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPIRXDR	[7:0]	Receive Data	0x5B	Read	spirxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPI_Receive_Data[7:0]	[7:0]	—	—	—	spirxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/spi_port/
SPIIRQ	[7:0]	Interrupt Request	0x5C	Read/Write	{1'b0, 1'b0, 1'b0, spisr_irqsts_4, spisr_irqsts_3, spisr_irqsts_2, spisr_irqsts_1, spisr_irqsts_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRDY	4	—	—	—	spisr_irqsts_4	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQRRDY	3	—	—	—	spisr_irqsts_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQROE	1	—	—	—	spisr_irqsts_1	../efb_top/efb_pll_sci_inst/u_efb_sci/

SPI Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
IRQMDF	0	—	—	—	spisr_irqsts_0	../efb_top/efb_pll_sci_inst/u_efb_sci/
SPIIRQEN	[7:0]	Interrupt Request Enable	0x5D	Read/Write	{1'b0, 1'b0, 1'b0, spisr_irqena_4, spisr_irqena_3, spisr_irqena_2, spisr_irqena_1, spisr_irqena_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQTRDYEN	4	—	—	—	spisr_irqena_4	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQRRDYEN	3	—	—	—	spisr_irqena_3	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQROEEN	1	—	—	—	spisr_irqena_1	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQMDFEN	0	—	—	—	spisr_irqena_0	../efb_top/efb_pll_sci_inst/u_efb_sci/

4. Hardened Timer/Counter PWM

The MachXO4 EFB contains a hard Timer/Counter IP core. This Timer/Counter is a general-purpose, bidirectional, 16-bit Timer/Counter module with independent output compare units and PWM support.

4.1. Timer/Counter Registers

The Timer/Counter communicates with the FPGA logic through the WISHBONE interface, by using a set of control, status, and data registers. Table 4.1 shows the register names and their functions. These registers are a subset of the EFB register map. Refer to the EFB register map for specific addresses of each register.

Table 4.1. Timer/Counter Registers

Timer/Counter Register Name	Register Function	Address	Access
TCCR0	Control Register 0	0x5E	Read/Write
TCCR1	Control Register 1	0x5F	Read/Write
TCTOPSET0	Set Top Counter Value [7:0]	0x60	Write
TCTOPSET1	Set Top Counter Value [15:8]	0x61	Write
TCOCRSET0	Set Compare Counter Value [7:0]	0x62	Write
TCOCRSET1	Set Compare Counter Value [15:8]	0x63	Write
TCCR2	Control Register 2	0x64	Read/Write
TCCNT0	Counter Value [7:0]	0x65	Read
TCCNT1	Counter Value [15:8]	0x66	Read
TCTOP0	Current Top Counter Value [7:0]	0x67	Read
TCTOP1	Current Top Counter Value [15:8]	0x68	Read
TCOCR0	Current Compare Counter Value [7:0]	0x69	Read
TCOCR1	Current Compare Top Counter Value [15:8]	0x6A	Read
TCICR0	Current Capture Counter Value [7:0]	0x6B	Read
TCICR1	Current Capture Counter Value [15:8]	0x6C	Read
TCSR0	Status Register	0x6D	Read/Write
TCIRQ	Interrupt Request	0x6E	Read/Write
TCIRQEN	Interrupt Request Enable	0x6F	Read/Write

Note: Unless otherwise specified, all Reserved bits in writable registers shall be written *0*.

Table 4.2. Timer/Counter Control

TCCR0								**0x5E**
Bit	7	6	5	4	3	2	1	0
Name	RSTEN	(Reserved)	PRESCALE[2:0]			CLKEDGE	CLKSEL	(Reserved)
Default	0	0	0			0	0	0
Access	R/W	—	R/W			R/W	R/W	R/W

RSTEN

Enables the reset signal (tc_rstn) to enter the Timer/Counter core from the PLD logic.

1: External reset enabled

0: External reset disabled

PRESCALE[2:0]

Used to divide the clock input to the Timer/Counter.

000: Static (clock disabled)

001: Divide by 1

010: Divide by 8

011: Divide by 64

100: Divide by 256

101: Divide by 1024

110: (Reserved setting)

111: (Reserved setting)

CLKEDGE

Used to select the edge of the input clock source. The Timer/Counter updates states on the edge of the input clock source.

0: Rising Edge

1: Falling Edge

CLKSEL

Defines the source of the input clock.

0: Clock Tree

1: On-chip Oscillator

Table 4.3. Timer/Counter Control 1

TCCR1								0x5F
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)	SOVFEN	ICEN	TSEL	OCM[1:0]		TCM[1:0]	
Default	0	0	0	0	0		0	
Access	—	R/W	R/W	R/W	R/W		R/W	

SOVFEN

Enables the overflow flag to be used with the interrupt output signal. It is set when the Timer/Counter is standalone, with no WISHBONE interface.

0: Disabled

1: Enabled

Note: When this bit is set, other flags such as the OCRF and ICRF are not routed to the interrupt output signal.

ICEN

Enables the ability to perform a capture operation of the counter value. You can assert the *tc_ic* signal and load the counter value onto the TCICR0/1 registers. The captured value can serve as a timer stamp for a specific event.

0: Disabled

1: Enabled

TSEL

Enables the auto-load of the counter with the value from TCTOPSET0/1. When disabled, the value 0xFFFF is auto-loaded.

0: Disabled

1: Enabled

OCM[1:0]

Select the function of the output signal of the Timer/Counter. The available functions are Static, Toggle, Set/Clear, and Clear/Set.

All Timer/Counter modes:

00: The output is static low

In non-PWM modes:

01: Toggle on TOP match

In Fast PWM mode:

10: Clear on TOP match, Set on OCR match

11: Set on TOP match, Clear on OCR match

In Phase and Frequency Correct PWM mode:

10: Clear on OCR match when the counter is incrementing

Set on OCR match when counter is decrementing

11: Set on OCR match when the counter is incrementing

Clear on OCR match when the counter is decrementing

TCM[1:0]

Timer Counter Mode. Defines the mode of operation for the Timer/Counter.

00: Watchdog Timer Mode

01: Clear Timer on Compare Match Mode

10: Fast PWM Mode

11: Phase and Frequency Correct PWM Mode

Table 4.4. Timer/Counter Set Top Counter Value 0

TCTOPSET0								**0x60**
Bit	7	6	5	4	3	2	1	0
Name	TCTOPSET[7:0]							
Default[1]	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note:

1. Hardware default value may be overridden by EFB component instantiation parameters.

Table 4.5. Timer/Counter Set Top Counter Value 1

TCTOPSET1								**0x61**
Bit	7	6	5	4	3	2	1	0
Name	TCTOPSET[15:8]							
Default[1]	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note:

1. Hardware default value may be overridden by EFB component instantiation parameters.

The value from TCTOPSET0/1 is loaded to the TCTOP0/1 registers once the counter has completed the current counting cycle. Refer to the Timer/Counter Modes of Operation section for usage details.

TCTOPSET0 register holds the lower eight bits [7:0] of the top value. TCTOPSET1 register holds the upper eight bits [15:8] of the top value.

Table 4.6. Timer/Counter Set Compare Counter Value 0

TCOCRSET0								0x62
Bit	7	6	5	4	3	2	1	0
Name	TCOCRSET[7:0]							
Default[1]	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note:

1. Hardware default value may be overridden by EFB component instantiation parameters.

Table 4.7. Timer/Counter Set Compare Counter Value 1

TCOCRSET1								0x63
Bit	7	6	5	4	3	2	1	0
Name	TCOCRSET[15:8]							
Default[1]	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note:

1. Hardware default value may be overridden by EFB component instantiation parameters.

The value from TCOCRSET0/1 is loaded to the TCOCR0/1 registers once the counter has completed the current counting cycle. Refer to the Timer/Counter Modes of Operation section for usage details.

TCOCRSET0 register holds the lower 8-bit value [7:0] of the compare value. TCOCRSET1 register holds the upper 8-bit value [15:8] of the compare value.

Table 4.8. Timer/Counter Control 2

TCCR2								0x64
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)					WBFORCE	WBRESETE	WBPAUSE
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	R/W	R/W	R/W

Note: *WBFORCE* and *WBRESETE* are only set to *1* when *wb_e_i* and *wb_cyc_i* are asserted. This means that once these signals are de-asserted, *WBFORCE* and *WBRESETE* go back to 0. This also implies that when reading these bits, the value is always *0*.

WBFORCE

In non-PWM modes, forces the output of the counter, as if the counter value matches the compare (TCOCR) value or it matches the top value (TCTOP).

0: Disabled

1: Enabled

WBRESET

Reset the counter from the WISHBONE interface by writing a *1* to this bit. Manually reset to *0*. The rising edge is detected in the WISHBONE clock domain, and the counter is reset synchronously on the next tc_clki. Due to the clock domain crossing, there is a one-clock uncertainty when the reset is effective. This bit has higher priority than WBPAUSE.

0: Disabled

1: Enabled

WBPAUSE

Pause the 16-bit counter

1: Pause

0: Normal

Table 4.9. Timer/Counter Counter Value 0

TCCNT0								0x65
Bit	7	6	5	4	3	2	1	0
Name	TCCNT[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 4.10. Timer/Counter Counter Value 1

TCCNT1								0x66
Bit	7	6	5	4	3	2	1	0
Name	TCCNT[15:8]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Registers TCCNT0 and TCCNT1 are 8-bit registers, which combined, hold the counter value. The WISHBONE host has read-only access to these registers.

TCCNT0 register holds the lower 8-bit value [7:0] of the counter value. TCCNT1 register holds the upper 8-bit value [15:8] of the counter value.

Table 4.11. Timer/Counter Current Top Counter Value 0

TCTOP0								0x67
Bit	7	6	5	4	3	2	1	0
Name	TCTOP[7:0]							
Default	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R

Table 4.12. Timer/Counter Current Top Counter Value 1

TCTOP0								0x68
Bit	7	6	5	4	3	2	1	0
Name	TCTOP[15:8]							
Default	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R

Registers TCTOP0 and TCTOP1 are 8-bit registers, which combined, receive a 16-bit value from the TCTOPSET0/1. The data stored in these registers represents the top value of the counter. The registers update once the counter has completed the current counting cycle. The WISHBONE host has read-only access to these registers. Refer to the Timer/Counter Modes of Operation section for usage details.

TCTOP0 register holds the lower 8-bit value [7:0] of the top value. TCTOP1 register holds the upper 8-bit value [15:8] of the top value.

Table 4.13. Timer/Counter Current Compare Counter Value 0

TCOCR0								0x69
Bit	7	6	5	4	3	2	1	0
Name	TCOCR[7:0]							
Default	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R

Table 4.14. Timer/Counter Current Compare Counter Value 1

TCOCR1								0x6A
Bit	7	6	5	4	3	2	1	0
Name	TCOCR[15:8]							
Default	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R

Registers TCOCR0 and TCOCR1 are 8-bit registers, which combined, receive a 16-bit value from the TCOCRSET0/1. The data stored in these registers represents the compare value of the counter. The registers update once the counter has completed the current counting cycle. The WISHBONE host has read-only access to these registers. Refer to the Timer/Counter Modes of Operation section for usage details.

TCOCR0 register holds the lower 8-bit value [7:0] of the compare value. TCOCR1 register holds the upper 8-bit value [15:8] of the compare value.

Table 4.15. Timer/Counter Current Capture Counter Value 0

TCICR0								0x6B
Bit	7	6	5	4	3	2	1	0
Name	TCICR[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 4.16. Timer/Counter Current Capture Counter Value 1

TCICR1								0x6C
Bit	7	6	5	4	3	2	1	0
Name	TCICR[15:8]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Registers TCICR0 and TCICR1 are 8-bit registers, and when combined, they can hold the counter value. The counter value is loaded onto these registers once a trigger event, tc_ic IP signal, is asserted. The capture value is commonly used as a timestamp for a specific system event. The WISHBONE host has read-only access to these registers.

TCICR0 register holds the lower 8-bit value [7:0] of the counter value. TCICR1 register holds the upper 8-bit value [15:8] of the counter value.

Table 4.17. Timer/Counter Status Register

TCSR0								0x6D
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)				BTF	ICRF	OCRF	OVF
Default	—	—	—	—	0	0	0	0
Access	—	—	—	—	R	R	R	R

BTF

Bottom Flag. Asserted when the counter reaches value zero. A write operation to this register clears this flag.

1: Counter reached zero value.

0: Counter has not reached zero.

ICRF

Capture Counter Flag. Asserted when the TC_IC input signal is asserted. The counter value is captured into the TCICR0/1 registers. A write operation to this register clears this flag. This bit is capable of generating an interrupt.

1: TC_IC signal asserted.

0: Normal.

OCRF

Compare Match Flag. Asserted when the counter matches the TCOCR0/1 register value. A write operation to this register clears this flag. This bit is capable of generating an interrupt.

1: Counter match.

0: Normal.

OVF

Overflow Flag. Asserted when the counter matches the TCTOP0/1 register value. A write operation to this register clears this flag. This bit is capable of generating an interrupt.

1: Counter match.

0: Normal.

Table 4.18. Timer/Counter Interrupt Status

TCIRQ								0x6E
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)					IRQICRF	IRQOCRF	IRQOVF
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	R/W	R/W	R/W

IRQICRF

Interrupt Status for Capture Counter Flag. When enabled, indicates ICRF is asserted. Write a *1* to this bit to clear the interrupt.

1: Capture Counter Flag Interrupt.

0: No interrupt.

IRQOCRF

Interrupt Status for Compare Match Flag. When enabled, indicates OCRF is asserted. Write a *1* to this bit to clear the interrupt.

1: Compare Match Flag Interrupt.

0: No interrupt.

IRQOVF

Interrupt Status for Overflow Flag. When enabled, indicates OVF is asserted. Write a *1* to this bit to clear the interrupt.

1: Overflow Flag Interrupt.

0: No interrupt.

Table 4.19. Timer/Counter Interrupt Enable

TCIRQEN								0x6F
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)					IRQICRFEN	IRQOCRFEN	IRQOVFEN
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	R/W	R/W	R/W

IRQICRFEN

Interrupt Enable for Capture Counter Flag.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQOCRFEN

Interrupt Enable for Compare Match Flag.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQOVFEN

Interrupt Enable for Overflow Flag.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

4.2. Timer/Counter Modes of Operation

There are four different modes of operation for the Timer/Counter:

- Watchdog Timer Mode
- Clear Timer on Compare Match Mode
- Fast PWM Mode
- Phase and Frequency Correct PWM Mode

4.3. Timer/Counter Simulation Model

The Timer/Counter EFB register map translation to the MachXO4 EFB software simulation model is provided below.

Table 4.20. Timer/Counter Simulation Model

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
TCCR0	[7:0]	Control Register 0	0x5E	Read/Write	{tc_rstn_ena, tc_gsrn_dis, tc_cclk_sel[2:0], tc_sclk_sel[2:0]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
RSTEN	7	—	—	—	tc_rstn_ena	../efb_top/efb_pll_sci_inst/u_efb_sci/
PRESCALE[2:0]	[5:3]	—	—	—	tc_cclk_sel[2:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
CLKEDGE	2	—	—	—	tc_sclk_sel[2]	../efb_top/efb_pll_sci_inst/u_efb_sci/
CLKSEL	1	—	—	—	tc_sclk_sel[1]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCR1	[7:0]	Control Register 1	0x5F	Read/Write	{1'b0, tc_ovf_ena, tc_ic_ena, tc_top_sel, tc_oc_mode[1:0], tc_mode[1:0]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
SOVFEN	6	—	—	—	tc_ivf_ena	../efb_top/efb_pll_sci_inst/u_efb_sci/
ICEN	5	—	—	—	tc_ic_ena	../efb_top/efb_pll_sci_inst/u_efb_sci/
TSEL	4	—	—	—	tc_top_sel	../efb_top/efb_pll_sci_inst/u_efb_sci/
OCM[1:0]	[3:2]	—	—	—	tc_oc_mode[1:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCM[1:0]	[1:0]	—	—	—	tc_mode[1:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOPSET0	[7:0]	Set Top Counter Value [7:0]	0x60	Write	{tc_top_set[7], tc_top_set[6], tc_top_set[5], tc_top_set[4], tc_top_set[3], tc_top_set[2], tc_top_set[1], tc_top_set[0]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOPSET[7:0]	[7:0]	—	—	—	{tc_top_set[7], tc_top_set[6], tc_top_set[5],	../efb_top/efb_pll_sci_inst/u_efb_sci/

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
					tc_top_set[4], tc_top_set[3], tc_top_set[2], tc_top_set[1], tc_top_set[0]}	
TCTOPSET1	[7:0]	Set Top Counter Value [15:8]	0x61	Write	{tc_top_set[15], tc_top_set[14], tc_top_set[13], tc_top_set[12], tc_top_set[11], tc_top_set[10], tc_top_set[9], tc_top_set[8]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOPSET[15:8]	[7:0]	—	—	—	{tc_top_set[15], tc_top_set[14], tc_top_set[13], tc_top_set[12], tc_top_set[11], tc_top_set[10], tc_top_set[9], tc_top_set[8]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCRSET0	[7:0]	Set Compare Counter Value [7:0]	0x62	Write	{tc_ocr_set[7], tc_ocr_set[6], tc_ocr_set[5], tc_ocr_set[4], tc_ocr_set[3], tc_ocr_set[2], tc_ocr_set[1], tc_ocr_set[0]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCRSET[7:0]	[7:0]	—	—	—	{tc_ocr_set[7], tc_ocr_set[6], tc_ocr_set[5], tc_ocr_set[4], tc_ocr_set[3], tc_ocr_set[2], tc_ocr_set[1], tc_ocr_set[0]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCRSET1	[7:0]	Set Compare Counter	0x63	Write	{tc_ocr_set[15],	../efb_top/efb_pll_sci_inst/u_efb_sci/

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
		Value [15:8]			tc_ocr_set[14], tc_ocr_set[13], tc_ocr_set[12], tc_ocr_set[11], tc_ocr_set[10], tc_ocr_set[9], tc_ocr_set[8]}	
TCOCRSET[15:8]	[7:0]	—	—	—	{tc_ocr_set[15], tc_ocr_set[14], tc_ocr_set[13], tc_ocr_set[12], tc_ocr_set[11], tc_ocr_set[10], tc_ocr_set[9], tc_ocr_set[8]}	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCR2	[7:0]	Control Register 2	0x64	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, 1'b0, tc_oc_force, tc_cnt_reset, tc_cnt_pause}	../efb_top/efb_pll_sci_inst/u_efb_sci/
WBFORCE	2	—	—	—	tc_oc_force	../efb_top/efb_pll_sci_inst/u_efb_sci/
WBRESET	1	—	—	—	tc_cnt_reset	../efb_top/efb_pll_sci_inst/u_efb_sci/
WBPAUSE	0	—	—	—	tc_cnt_pause	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCNT0	[7:0]	Counter Value [7:0]	0x65	Read	tc_cnt_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCNT[7:0]	[7:0]	—	—	—	tc_cnt_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCNT1	[7:0]	Counter Value [15:8]	0x66	Read	tc_cnt_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCCNT[15:8]	[7:0]	—	—	—	tc_cnt_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOP0	[7:0]	Current Top Counter Value [7:0]	0x67	Read	tc_top_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOP[7:0]	[7:0]	—	—	—	tc_top_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOP1	[7:0]	Current Top Counter Value [15:8]	0x68	Read	tc_top_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCTOP[15:8]	[7:0]	—	—	—	tc_top_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCR0	[7:0]	Current Compare	0x69	Read	tc_ocr_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
		Counter Value				
TCOCR[7:0]	[7:0]	—	—	—	tc_ocr_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCR1	[7:0]	Current Compare Top Counter Value [15:8]	0x6A	Read	tc_ocr_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCOCR[15:8]	[7:0]	—	—	—	tc_ocr_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCICR0	[7:0]	Current Capture Counter Value	0x6B	Read	tc_icr_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCICR[7:0]	[7:0]	—	—	—	tc_icr_sts[7:0]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCICR1	[7:0]	Current Capture Counter Value	0x6C	Read	tc_icr_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCICR[15:8]	[7:0]	—	—	—	tc_icr_sts[15:8]	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCSR0	[7:0]	Status Register	0x6D	Read	{1'b0, 1'b0, 1'b0, 1'b0, tc_btf_sts, tc_icrf_sts, tc_ocrf_sts, tc_ovf_sts}	../efb_top/efb_pll_sci_inst/u_efb_sci/
BTF	3	—	—	—	tc_btf_sts	../efb_top/efb_pll_sci_inst/u_efb_sci/
ICRF	2	—	—	—	tc_icrf_sts	../efb_top/efb_pll_sci_inst/u_efb_sci/
OCRF	1	—	—	—	tc_ocrf_sts	../efb_top/efb_pll_sci_inst/u_efb_sci/
OVF	0	—	—	—	tc_ovf_sts	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCIRQ	[7:0]	Interrupt Request	0x6E	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, 1'b0, tc_icrf_irqsts, tc_ocrf_irqsts, tc_ovf_irqsts}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQICRF	2	—	—	—	tc_icrf_irqsts	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQOCRF	1	—	—	—	tc_ocrf_irqsts	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQOVF	0	—	—	—	tc_ovf_irqsts	../efb_top/efb_pll_sci_inst/u_efb_sci/
TCIRQEN	[7:0]	Interrupt Request Enable	0x6F	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, 1'b0, tc_icrf_irqena, tc_ocrf_irqena, tc_ovf_irqena}	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQICRFEN	2	—	—	—	tc_icrf_irqena	../efb_top/efb_pll_sci_inst/u_efb_sci/

Timer/Counter Register Name	Register Size/Bit Location	Register Function	Address	Access	Simulation Model Register Name	Simulation Model Register Path
IRQOCRFEN	1	—	—	—	tc_ocrf_irqena	../efb_top/efb_pll_sci_inst/u_efb_sci/
IRQOVFEN	0	—	—	—	tc_ovf_irqena	../efb_top/efb_pll_sci_inst/u_efb_sci/

5. MachXO4 (LFMXO4) Flash Access

This section describes the MachXO4 (LFMXO4) flash access. For the MachXO4 RoT flash access details, refer to the MachXO4 RoT Flash Access section.

The MachXO4 (LFMXO4) Flash Logic interface can be accessed using the JTAG, SPI, I2C, or WISHBONE interfaces. The MachXO4 Flash consists of three sectors:

- Configuration Flash (includes USERCODE)
- UFM
- Feature Row

The Flash is organized in pages. The Flash is not byte addressable. Each page has 128 bits (16 bytes).

5.1. Flash Access Ports

You can access the Flash Memory through the JTAG port (compliant with the IEEE 1149.1 and IEEE 1532 specifications), external Target SPI port, external I2C Primary port, and the internal WISHBONE interface of the EFB module. Figure 5.1 illustrates the interfaces to the Flash sectors.

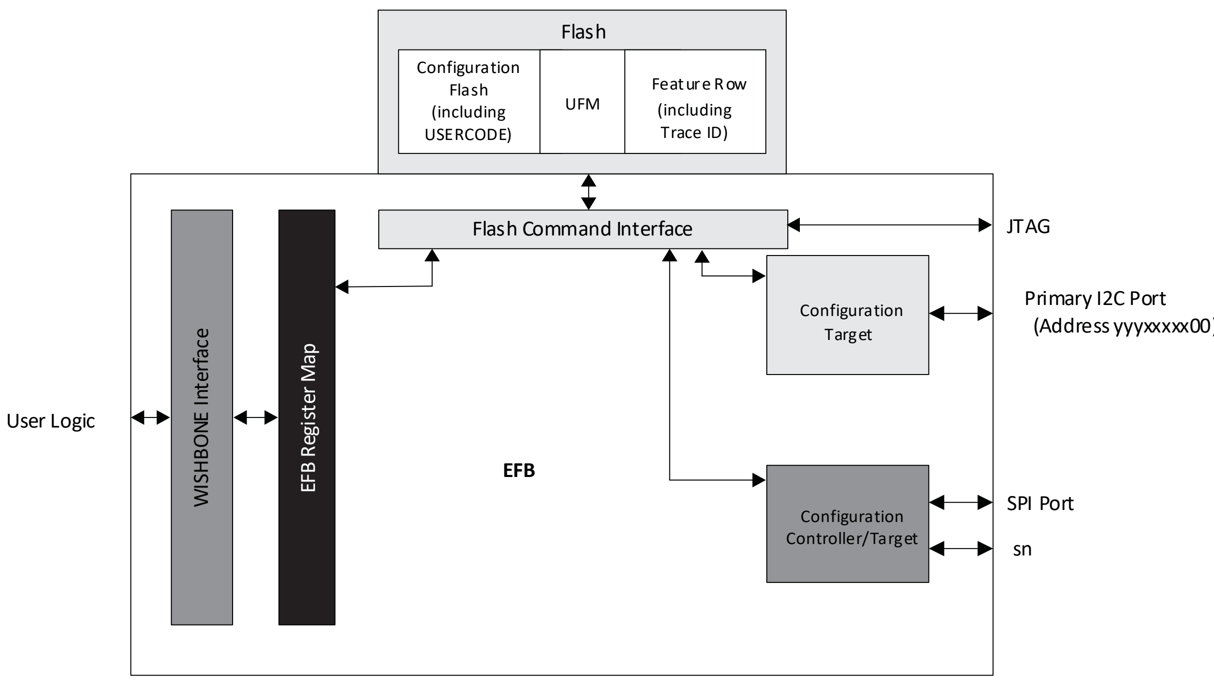


Figure 5.1. MachXO4 (LFMXO4) Interfaces to the Flash Sectors

The configuration logic arbitrates access from the interfaces according to the following priority. When higher priority ports are enabled, Flash access by lower priority ports is blocked.

- JTAG Port
- Target SPI Port
- I2C Primary Port
- WISHBONE Target Interface

Note: Enabling Flash Interface using Enable Configuration Interface command 0x74 Transparent Mode temporarily disables certain features of the device including:

- Power Controller
- GSR
- Hardened User SPI port

- Hardened User Primary I2C port

Functionality is restored after the Flash Interface is disabled using Disable Configuration Interface command 0x26 followed by Bypass command 0xFF.

5.2. Flash Access through WISHBONE Target Interface

The WISHBONE Target interface of the EFB module allows you to access the Flash directly from the FPGA core logic. The WISHBONE bus signals, described earlier in this document, are used by a WISHBONE host that you can implement using the general-purpose FPGA resources.

The WISHBONE Interface communicates to the Configuration Logic through a set of data, control, and status registers. Table 5.1 shows the register names and their functions. These registers are a subset of the EFB register map. Refer to the EFB register map for specific addresses of each register.

Table 5.1. WISHBONE to Flash Logic Registers

WISHBONE to CFG Register Name	Register Function	Address	Access
CFGCR	Control	0x70	Read/Write
CFGTXDR	Transmit Data	0x71	Write
CFGSR	Status	0x72	Read
CFGRXDR	Receive Data	0x73	Read
CFGIRQ	Interrupt Request	0x74	Read/Write
CFGIRQEN	Interrupt Request	0x75	Read/Write

Note: Unless otherwise specified, all reserved bits in writable registers shall be written *0*.

Table 5.2. Flash Control

CFGCR								**0x70**
Bit	7	6	5	4	3	2	1	0
Name	WBCE	RSTE	(Reserved)					
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	—	—	—	—	—	—

WBCE

WISHBONE Connection Enable. Enables the WISHBONE to establish the read/write connection to the Flash logic. This bit must be set prior to executing any command through the WISHBONE port. Likewise, this bit must be cleared to terminate the command. Refer to the Command and Data Transfers to Flash Space section for more information on framing WISHBONE commands.

1: Enabled

0: Disabled

RSTE

WISHBONE Connection Reset. Resets the input/output FIFO logic. The reset logic is level sensitive. After setting this bit to *1*, it must be cleared to *0* for normal operation.

1: Reset

0: Normal operation

Table 5.3. Flash Transmit Data

CFGTXDR								**0x71**
Bit	7	6	5	4	3	2	1	0
Name	CFG_Transmit_Data[7:0]							
Default	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W

CFG_Transmit_Data[7:0]

CFG Transmit Data. This register holds the byte to be written to the Flash logic. Bit 0 is LSB.

Table 5.4. Flash Status

CFGSR								0x72
Bit	7	6	5	4	3	2	1	0
Name	WBCACT	(Reserved)	TXFE	TXFF	RXFE	RXFF	SSPIACT	I2CACT
Default	0	0	0	0	0	0	0	0
Access	R	—	R	R	R	R	R	R

WBCACT

WISHBONE Bus to Configuration Logic Active. Indicates that the WISHBONE to configuration interface is active and the connection is established.

1: WISHBONE Active

0: WISHBONE not Active

TXFE

Transmit FIFO Empty. Indicates that the Transmit Data register is empty. This bit is capable of generating an interrupt.

1: FIFO empty

0: FIFO not empty

TXFF

Transmit FIFO Full. Indicates that the Transmit Data register is full. This bit is capable of generating an interrupt.

1: FIFO full

0: FIFO not full

RXFE

Receive FIFO Empty. Indicates that the Receive Data register is empty. This bit is capable of generating an interrupt.

1: FIFO empty

0: FIFO not empty

RXFF

Receive FIFO Full. Indicates that the Transmit Data register is full. This bit is capable of generating an interrupt.

1: FIFO full

0: FIFO not full

SSPIACT

Target SPI Active. Indicates the Target SPI port has started actively communicating with the Configuration Logic while WBCE is enabled. This port has priority over the I2C and WISHBONE ports. It pre-empts any existing, and prohibits any new, lower priority transaction. This bit is capable of generating an interrupt.

1: Target SPI port active

0: Target SPI port not active

I2CACT

I2C Active. Indicates the I2C port has started actively communicating with the Configuration Logic while WBCE is enabled. This port has priority over the WISHBONE ports. It pre-empts any existing, and prohibits any new WISHBONE transaction. This bit is capable of generating an interrupt.

1: I2C port active

0: I2C port not active

Table 5.5. Flash Receive Data

CFGRXDR								0x73
Bit	7	6	5	4	3	2	1	0
Name	CFG_Receive_Data[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

CFG_Receive_Data[7:0]

CFG Receive Data. This register holds the byte read from the Flash logic. Bit 0 in this register is LSB.

Table 5.6. Flash Interrupt Status

CFGIRQ								0x74
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)		IRQTXFE	IRQTXFF	IRQRXFE	IRQRXFF	IRQSSPIACT	IRQI2CACT
Default	0	0	0	0	0	0	0	0
Access	—	—	R/W	R/W	R/W	R/W	R/W	R/W

IRQTXFE

Interrupt Status for Transmit FIFO Empty. When enabled, indicates TXFE is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmit FIFO Empty Interrupt

0: No interrupt

IRQTXFF

Interrupt Status for Transmit FIFO Full. When enabled, indicates TXFF is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmit FIFO Full Interrupt

0: No interrupt

IRQRXFE

Interrupt Status for Receive FIFO Empty. When enabled, indicates RXFE is asserted. Write a *1* to this bit to clear the interrupt.

1: Receive FIFO Empty Interrupt

0: No interrupt

IRQRXFF

Interrupt Status for Receive FIFO Full. When enabled, indicates RXFF is asserted. Write a *1* to this bit to clear the interrupt.

1: Receive FIFO Full Interrupt

0: No interrupt

IRQSSPIACT

Interrupt Status for Target SPI Active. When enabled, indicates SSPIACT is asserted. Write a *1* to this bit to clear the interrupt.

1: Target SPI Active Interrupt

0: No interrupt

IRQI2CACT

Interrupt Status for I2C Active. When enabled, indicates I2CACT is asserted. Write a *1* to this bit to clear the interrupt.

1: I2C Active Interrupt

0: No interrupt

Table 5.7. Flash Interrupt Enable

CFGIRQEN								0x75
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)		IRQTXFEEN	IRQTXFFEN	IRQRXFEEN	IRQRXFFEN	IRQSSPIACTEN	IRQI2CACTEN
Default	0	0	0	0	0	0	0	0
Access	—	—	R/W	R/W	R/W	R/W	R/W	R/W

IRQTXFEEN

Interrupt Enable for Transmit FIFO Empty.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQTXFFEN

Interrupt Enable for Transmit FIFO Full.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQRXFEEN

Interrupt Enable for Receive FIFO Empty.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQRXFFEN

Interrupt Enable for Receive FIFO Full.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQSSPIACTEN

Interrupt Enable for Target SPI Active.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQI2CACTEN

Interrupt Enable for I2C Active.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

Table 5.8. Unused (Reserved) Register

UNUSED								0x76
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)							
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	—	—	—

Table 5.9. EFB Interrupt Source

EFBIRQ								0x77
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)			CFG_INT	TC_INT	SPI_INT	I2C2_INT	I2C1_INT
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

CFG_INT

Flash Interrupt Source. Indicates EFB interrupt source is from the Flash Block. Use CFGIRQ for further source resolution.

1: A bit is set in register CFGIRQ.

0: No interrupt.

TC_INT

Timer/Counter Interrupt Source. Indicates EFB interrupt source is from the Timer/Counter Block. Use TCIRQ for further source resolution.

1: A bit is set in register TCIRQ.

0: No interrupt.

SPI_INT

SPI Interrupt Source. Indicates EFB interrupt source is from the SPI Block. Use SPI-IRQ for further source resolution.

1: A bit is set in register SPIIRQ.

0: No interrupt.

I2C2_INT

I2C2 Interrupt Source. Indicates EFB interrupt source is from the Secondary I2C Block. Use I2C_2_ IRQ for further source resolution.

1: A bit is set in register I2C_2_ IRQ.

0: No interrupt.

I2C1_INT

I2C1 Interrupt Source. Indicates EFB interrupt source is from the Primary I2C Block. Use I2C_1_ IRQ for further source resolution.

1: A bit is set in register I2C_1_ IRQ.

0: No interrupt.

5.3. Command and Data Transfers to Flash Space

The command and data transferred to the Flash are identical for all the access ports, regardless of their different physical interfaces. The Flash is organized in pages. Therefore, you address a specific page for Read or Write operations to that page. Each page has 128 bits (16 bytes). The transfers are based on a set of instructions and page addresses. The Flash is composed of three sectors: Configuration Flash (includes USERCODE), UFM, and Feature Row. The Erase operations are sector based.

5.4. Command Summary by Application

Table 5.10. UFM Commands

Command Name	Command MSB LSB	SVF Command Name	Description
Read Status Register	0x3C	LSC_READ_STATUS	Read the 4-byte Configuration Status Register.
Check Busy Flag	0xF0	LSC_CHECK_BUSY	Read the Configuration Busy Flag status.
Bypass	0xFF	ISC_NOOP	Null operation.
Enable Configuration Interface (Transparent Mode)	0x74	ISC_ENABLE_X	Enable Transparent UFM access – All user I/O (except the hardened user SPI and primary user I2C ports) are governed by the user logic, the device remains in User mode. (The subsequent commands in this table require the interface to be enabled.)
Enable Configuration Interface (Offline Mode)	0xC6	ISC_ENABLE	Enable Offline UFM access – All user I/O (except persisted sysCONFIG ports) are tri-stated. User logic ceases to function, UFM remains accessible, and the device enters *Offline* access mode. (The subsequent commands in this table require the interface to be enabled.)
Disable Configuration Interface	0x26	ISC_DISABLE	Disable the configuration (UFM) access.
Set Address	0xB4	LSC_WRITE_ADDRESS	Set the UFM sector 14-bit Address Register.
Reset UFM Address	0x47	LSC_INIT_ADDR_UFM	Reset the address to point to Sector 1, Page 0 of the UFM.
Read UFM	0xCA	LSC_READ_TAG	Read the UFM data. Operand specifies the number of pages to read. Address Register is post-incremented.
Erase UFM	0xCB	LSC_ERASE_TAG	Erase the UFM sector only.
Program UFM Page	0xC9	LSC_PROG_TAG	Write one page of data to the UFM. Address Register is post-incremented.

Table 5.11. Configuration Flash (Sector 0) Commands

Command Name	Command MSB LSB	SVF Command Name	Description
Read Device ID	0xE0	IDCODE_PUB	Read the 4-byte Device ID (0x01 2b 20 43).
Read USERCODE	0xC0	USERCODE	Read the 32-bit USERCODE.
Read Status Register	0x3C	LSC_READ_STATUS	Read the 4-byte Configuration Status Register.
Read Busy Flag	0xF0	LSC_CHECK_BUSY	Read the Configuration Busy Flag status.
Refresh[1]	0x79	LSC_REFRESH	Launch boot sequence (same as toggling PROGRAMN pin).
STANDBY	0x7D	LSC_DEVICE_CTRL	Triggers the Power Controller to enter or wake from standby mode.
Bypass	0xFF	ISC_NOOP	Null operation.
Enable Configuration Interface (Transparent Mode)	0x74	ISC_ENABLE_X	Enable Transparent Configuration Flash access – All user I/O (except the hardened user SPI and primary user I2C ports) are governed by the user logic, the device remains in User mode. (The subsequent commands in this table require the interface to be enabled.)
Enable Configuration Interface (Offline Mode)	0xC6	ISC_ENABLE	Enable Offline Configuration Flash access – All user I/O (except persisted sysCONFIG ports) are tri-stated. User logic ceases to function, and the device enters *Offline* access mode. (The subsequent commands in this table require the interface to be enabled.)
Disable Configuration Interface	0x26	ISC_DISABLE	Exit access mode.
Set Configuration Flash Address	0xB4	LSC_WRITE_ADDRESS	Set the Configuration Flash 14-bit Page Address.
Verify Device ID	0xE2	VERIFY_ID	Verify device ID with 32-bit input, set Fail flag if mismatched.
Reset Configuration Flash Address	0x46	LSC_INIT_ADDRESS	Reset the address to point to Sector 0, Page 0 of the Configuration Flash.
Read Flash	0x73	LSC_READ_INCR_NV	Read the Flash data. Operand specifies the number of pages to read. Address Register is post-incremented.
Erase	0x0E	ISC_ERASE	Erase the Config Flash, FEATURE Row, FEABITs, Done bit, Security bits, and USERCODE.
Program Page	0x70	LSC_PROG_INCR_NV	Write one page of data to the Flash. Address Register is post-incremented.
Program DONE	0x5E	ISC_PROGRAM_DONE	Program the Done bit.
Program SECURITY	0xCE	ISC_PROGRAM_SECURITY	Program the Security bit (Secures CFG Flash sector).
Program SECURITY PLUS	0xCF	ISC_PROGRAM_SECPLUS	Program the Security Plus bit (Secures CFG, Flash and UFM Sectors). **Note:** SECURITY and SECURITY PLUS commands are mutually exclusive.
Program USERCODE	0xC2	ISC_PROGRAM_USERCODE	Program 32-bit USERCODE.
Read Feature Row	0xE7	LSC_READ_FEATURE	Read Feature Row.
Program Feature Row	0xE4	LSC_PROG_FEATURE	Program Feature Row.
Read FEABITS	0xFB	LSC_READ_FEABITS	Read FEA bits.
Program FEABITs	0xF8	LSC_PROG_FEABITS	Program the FEA bits.

Table 5.12. Non-Volatile Register (NVR) Commands

Command Name	Command MSB LSB	SVF Command Name	Description
Read TraceID code	0x19	UIDCODE_PUB	Read 64-bit TraceID.

When using the WISHBONE bus interface, the commands, operand, and data are written to the CFGTXDR Register. The Target SPI or I2C interface shifts the most significant bit (MSB) first into the MachXO4 device. This is required only when communicating with the configuration logic inside the MachXO4 device.

In order to perform a Write, Read, or Erase operation with the Flash, it is required that the interface is enabled using Command 0x74. Affected commands are noted in the Command Description as EN Required. Once the modification operations are completed, the interface can be disabled using commands 0x26 and 0xFF.

5.5. Command Descriptions by Command Code

5.5.1. Erase Flash (0x0E)

Table 5.13. Erase Flash (0x0E)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	Y	0E	Refer below	—	—	—

Operand 0000 ucfs 0000 0000 0000 0000(binary)

Where:

u: Erase UFM sector

0: No action

1: Erase

c: Erase CFG Flash sector (Config Flash, DONE, security bits, USERCODE)

0: No action

1: Erase

f: Erase Feature sector (Target I2C address, sysCONFIG port persistence, Boot mode, and so on)

0: No action

1: Erase

s: Erase SRAM

0: No action

1: Erase

Notes: Poll the BUSY bit (or wait, Refer to Table 10.1) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.

Erased condition for Flash bits = 0.

Examples:

0x0E 04 00 00

Erase CFG Flash sector.

0x0E 08 00 00

Erase UFM sector.

0x0E 0C 00 00

Erase UFM and CFG Flash sectors.

5.5.2. Read TraceID Code (0x19)

Table 5.14. Read TraceID Code (0x19)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	—	x	N	19	00 00 00	R	8B	—

Example: 0x19 00 00 00

Read 8-byte TraceID

Note: First byte read is user portion. Next seven bytes are unique to each silicon die.

5.5.3. Disable Configuration Interface (0x26)

Table 5.15. Disable Configuration Interface (0x26)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	—	26	00 00	—	—	—

Example: 0x26 00 00

Disable Flash interface for change access.

Notes: Must have only two operands.

The interface cannot be disabled while the Configuration Status Register Busy bit is asserted. After commands (for example, Erase, Program), verify Busy is clear before issuing the Disable command.

Follow this command with Command 0xFF (BYPASS) to complete the Disable operation. The BYPASS command is required to restore Power Controller, GSR, Hardened User SPI, and I2C port operation.

SRAM must be erased before exiting Offline (0xC6) Mode.

5.5.4. Read Status Register (0x3C)

Table 5.16. Read Status Register (0x3C)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
x	x	—	N	3C	00 00 00	R	4B	xxxx IxEE Exxx xxxx xxFB xxCD xxxx xxxx

Data Format: Most significant byte of SR is received first, LSB last.

D bit 8 Flash or SRAM Done Flag

When C = 0 SRAM Done bit has been programmed
D = 1 Successful Flash to SRAM transfer
D = 0 Failure in the Flash to SRAM transfer
When C=1 Flash Done bit has been programmed

D = 1 Programmed
D = 0 Not Programmed

C bit 9 Enable Configuration Interface (1=Enable, 0=Disable)

B bit 12: Busy Flag (1 = busy)

F bit 13: Fail Flag (1 = operation failed)

I I=0 Device verified correct, I=1 Device failed to verify

EEE bits[25:23]: Configuration Check Status

000: No Error
001: ID ERR
010: CMD ERR
011: CRC ERR
100: Preamble ERR
101: Abort ERR
110: Overflow ERR
111: SDM EOF

(all other bits reserved)

Usage: Check the BUSY bit after all Enable, Erase, or Program operations.

Note: Wait at least 1 µs after power-up or asserting wb_rst_i before accessing the EFB.

Example: 0x3C 00 00 00

Read 4-byte Status Register for example, 0x00 00 20 00 (fail flag set).

5.5.5. Reset CFG Address (0x46)

Table 5.17. Reset CFG Address (0x46)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	46	00 00 00	—	—	—

Example: 0x46 00 00 00

Set Address register to Configuration Sector 0, page 0.

5.5.6. Reset UFM Address (0x47)

Table 5.18. Reset UFM Address (0x47)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
X	—	—	Y	47	00 00 00	—	—	—

Example: 0x47 00 00 00

Set Address register to Flash Sector 1, page 0.

5.5.7. Program DONE (0x5E)

Table 5.19. Program DONE (0x5E)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	5E	00 00 00	—	—	—

Example: 0x5E 00 00 00

Set the DONE bit.

Note: Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

5.5.8. Program Configuration Flash (0x70)

Table 5.20. Program Configuration Flash (0x70)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	x	—	Y	70	00 00 00	W	16B	16 bytes Flash write data

Example: 0x70 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write one page of data, pointed to by the Address Register.

Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. The Address Register is auto-incremented following the page write.

Use 0x0E to erase CFG Flash sector prior to executing this command.

Poll the BUSY bit (or wait 200 μs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

5.5.9. Read Configuration Flash (0x73) (SPI)

Table 5.21. Read Configuration Flash (0x73) (SPI)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note: This applies when Configuration Flash is read through SPI.

1. Operand: 0001 0000 00pp pppp pppp pppp (binary)

   pp..pp: num_pages Number of CFG Flash pages to read when num_pages = 1

   Number of CFG Flash pages to read +1 when num_pages > 1

2. Data Size: (num_pages × 16) bytes

Notes: Read CFG Flash may be aborted at any time. Any data remaining in the read FIFO will be discarded. Any read data beyond the prescribed read size will be indeterminate. The Address Register is auto-incremented after each page read.

3. Examples: 0x73 10 00 01

   0-byte dummy followed by one page of CFG Flash data (16 bytes total).

   0x73 10 00 04

   Read 1-page dummy followed by three pages of CFG Flash data (four pages total).

5.5.10. Read Configuration Flash (0x73) (I2C/SPI)

Table 5.22. Read Configuration Flash (0x73) (I2C/SPI)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	0x73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note: This applies when Configuration Flash is read through I2C or SPI.

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)

   pp..pp: num_pages Number of CFG Flash pages to read when num_pages = 1

   Number of CFG Flash pages to read +1 when num_pages > 1

2. Data Size: (num_pages × 16) bytes when num_pages=1

		32 bytes + (num_pages) × (16 + 4) bytes when num_pages>1
Notes:		Read CFG Flash may be aborted at any time. Any data remaining in the read FIFO will be discarded. Any read data beyond the prescribed read size will be indeterminate. The Address Register is auto-incremented after each page read.
3.	Examples:	0x73 00 00 01
		0-byte dummy followed by one page CFG Flash data (16 bytes total)
		0x73 00 00 04
		Read 2-pages dummy, followed by three sets [1 page CFG Flash data, followed by four bytes dummy] (five pages and 12 dummy bytes total).

5.5.11. Read Configuration Flash (0x73) (WISHBONE)

Table 5.23. Read Configuration Flash (0x73) (WISHBONE)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note:	This applies when Configuration Flash is read through WISHBONE.
1. Operand:	0001 0000 00pp pppp pppp pppp (binary), or
	0000 0000 00pp pppp pppp pppp (binary)
	pp..pp: num_pages Number of CFG Flash pages to read when num_pages = 1
	Number of CFG Flash pages to read +1 when 1 < num_pages ≤ 12
2. Data Size:	Set to 0x3FFF when num_pages > 12
Note:	(num_pages × 16) bytes when num_pages=1
	32 bytes + (num_pages) × (16 + 4) bytes when num_pages>1
3. Examples:	When reading more than 12 pages, the num_pages argument is intentionally oversized. It is not necessary to read the extra pages. Read CFG Flash may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. The Address Register is auto-incremented after each page read.
Notes:	0x73 00 00 01
	0-byte dummy followed by one page CFG Flash data (16 bytes in total).
	0x73 10 00 04
	Read one page dummy followed by three pages of CFG data (four pages in total).
	0x73 00 00 04
	Read two pages dummy, followed by three sets [one page CFG Flash data, followed by four bytes dummy] (five pages and 12 dummy bytes total).
	The maximum speed which one page of data (num_page=1) can be read using WISHBONE and no wait state is 16.6 MHz. Faster WISHBONE clock speeds are supported by inserting WB wait states to observe the retrieval delay timing requirement. For more information, refer to the Reading Flash Pages section of the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

5.5.12. Enable Configuration Interface (Transparent) (0x74)

Table 5.24. Enable Configuration Interface (Transparent) (0x74)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	—	74	08 00 00 or 08 00	—	—	—

Notes: The I2C interface uses only two operands and all other interfaces use three operands. This command is required to enable modification of the UFM, configuration CFG Flash, or non-volatile registers (NVR). Terminate this command with the command

0x26 followed by the command 0xFF.

Exercising this command temporarily disables certain features of the device, notably GSR, user SPI port, primary user I2C port and Power Controller. These features are restored when the command is terminated.

Poll the BUSY bit (or wait 5 µs) after issuing this command for the Flash pumps to fully charge.

Examples: 0x74 08 00 00

Enable Flash interface for change access through a non- I2C interface.

5.5.13. Refresh (0x79)

Table 5.25. Refresh (0x79)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	—	—	—	79	00 00	—	—	—

Example: 0x79 00 00

Issue Refresh command

Notes: The Refresh command launches Boot sequence

Must have only two operands

After completing the Refresh command (for example, SPI SN de-assertion or I2C stop), further bus accesses are prohibited for the duration of $t_{REFRESH}$. Violating this requirement causes the Refresh process to abort and leaves the MachXO4 device in an unprogrammed state.

Occasionally, following a device REFRESH or PROGRAMN pin toggle, the secondary I2C port may be left in an undefined (non-idle) state. The likelihood of this condition is design and route dependent. To positively return the Secondary I2C port to the idle state, write a value of 0x44 to register I2C_2_CMDR through WISHBONE immediately after device reset is released. This causes a short low-pulse on SCK as the hard-block signals a STOP on the bus then returns to the idle state. Failure to manually return the Secondary I2C port to the idle state may result in an I2C bus lock-up condition. Normal I2C activity can be commenced without additional delay.

5.5.14. STANDBY (0x7D)

Table 5.26. STANDBY (0x7D)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	N	7D	0y 00	—	—	—

Example: 0x7D 0y 00

y:2: Triggers the Power Controller to enter standby mode.

y:8: Triggers the Power Controller to wake up from standby mode.

Notes: Must have only two operands.

The MachXO4 Power Controller needs to be included in the design.

Additionally, the following can be used to trigger the Power Controller to wake up from the standby mode (if the user logic standby signal has not been enabled):

1. I2C has the following ways:

a. Primary I2C Configuration port – Address match to the I2C Configuration address (No other settings required).

b. Primary or Secondary I2C User port – Address match the I2C User address. Must have I2C_1_CR[WKUPEN] or I2C_1_CR[WKUPEN] set.

c. General Call – Send the General Call Wakeup command (0xF3). Must have General Calls enabled (I2C_1_CR[GCEN] or I2C_2_CR[GCEN] set) and use the General Call address.

2. SPI from the assertion of either Target Configuration (sn) or User (spi_scsn) chip select, as long as the appropriate control register bit is set:

a. Configuration: SPICR1[WKUPEN_CFG]

b. User: SPICR[WKUPEN_USER]

For more information on the Power Controller, refer to the Power and Thermal Estimation and Management for MachXO4 Devices (FPGA-TN-02409).

5.5.15. Set Address (0xB4)

Table 5.27. Set Address (0xB4)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
x	x	—	Y	B4	00 00 00	W	4B	0s00 0000 0000 0000 00aa aaaa aaaa aaaa

Data Format: s: sector

0: CFG Flash

1: UFM

aa..aa:address14-bit page address

Example: 0xB4 00 00 00 40 00 00 0A

Set Address register to UFM sector, page 10 decimal

5.5.16. Read USERCODE (0xC0)

Table 5.28. Read USERCODE (0xC0)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y/N	C0	00 00 00	R	4B	—

Example: 0xC0 00 00 00

EN Required = Y Read 4-byte USERCODE from CFG Flash sector

EN Required = N Read 4-byte USERCODE from SRAM

5.5.17. Program USERCODE (0xC2)

Table 5.29. Program USERCODE (0xC2)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	C2	00 00 00	W	4B	—

Example: 0xC2 00 00 00 10 20 30 40

Sets USERCODE with 32-bit input 0x10 20 30 40.

Note: Poll the BUSY bit (or wait 200 us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

5.5.18. Enable Configuration Interface (Offline) (0xC6)

Table 5.30. Enable Configuration Interface (Offline) (0xC6)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	—	C6	0y 00 00	—	—	—

Operand: 08 00 00 - Enable Flash Normal mode. Normal edit mode for Offline configuration. Used for all offline commands described in this document, including Erase SRAM.

00 00 00 - Enable SRAM mode. Optional edit mode. Supports Erase SRAM command only.

Example: 0xC6 08 00 00 Enable Flash interface for offline change access.

Notes: Use this command to enable offline modification of the Flash, or non-volatile registers (NVR). SRAM must be erased exiting Offline mode. When exiting Offline mode follow the command 0x26 with the command 0xFF. Exercising this command will tristate all user I/O (except persisted sysCONFIG ports). User logic ceases to function. UFM remains accessible.

Poll the BUSY bit (or wait 5 µs) after issuing this command for the Flash pumps to fully charge.

5.5.19. Program UFM (0xC9)

Table 5.31. Program UFM (0xC9)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
x	—	—	Y	C9	00 00 01	W	16B	16 bytes Flash write data

Example: 0xC9 00 00 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Write one page of data, pointed to by Address Register.

Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. The Address Register is auto-incremented following the page write.

Use 0x0E or 0xCB to erase UFM sector prior to executing this command.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

5.5.20. Read UFM (0xCA) (SPI)

Table 5.32. Read UFM (0xCA) (SPI)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0001 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when num_pages > 1

2. Data Size (num_pages × 16) bytes

Notes: Read UFM may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. The Address Register is auto-incremented after each page read.

3. Examples: 0xCA 10 00 01

Read 0-byte dummy followed by one page UFM data (16 bytes total).

0xCA 10 00 04

Read one-page dummy followed by three pages UFM data (four pages total).

5.5.21. Read UFM (0xCA) (SPI/I2C)

Table 5.33. Read UFM (0xCA) (SPI/I2C)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when num_pages > 1

2. Data Size (num_pages × 16) bytes when num_pages=1

32 bytes + (num_pages × 16 + 4) bytes when num_pages>1

Notes: Read UFM may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. The Address Register is auto-incremented after each page read.

3. Examples: 0xCA 00 00 01

Read 0-byte dummy followed by one page UFM data (16 bytes total).

0xCA 00 00 04

Read two-page dummy followed by three sets [one page UFM data, followed by four bytes dummy] (five pages total and 12 dummy bytes).

5.5.22. Read UFM (0xCA) (WISHBONE)

Table 5.34. Read UFM (0xCA) (WISHBONE)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0001 0000 00pp pppp pppp pppp (binary), or

0000 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when 1 < num_pages ≤ 12

Set to 0x3FFF when num_pages > 12

2. Data Size (num_pages × 16) bytes when num_pages=1

32 bytes + (num_pages × 16 + 4) bytes when num_pages>1

Notes: When reading more than 12 pages, the num_pages argument is intentionally oversized. It is not necessary to read the extra pages. Read UFM may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. The Address Register is auto-incremented after each page read.

3. Examples: 0xCA 00 00 01

Read 0 bytes dummy followed by one page UFM data (16 bytes total)

0xCA 10 00 04

Read 1 page dummy followed by three pages of CFG data (four pages total)

0xCA 00 00 04

Read two-page dummy followed by three sets [one page UFM data, followed by four bytes dummy] (five pages total and 12 dummy bytes)

Notes: The maximum WISHBONE clock speed with which one page of data (num_page=1) can be read using WISHBONE and no wait states is 16.6 MHz. Faster WISHBONE clock speeds are supported by inserting WB wait states to observe the retrieval delay timing requirement. For more information, refer to the Reading Flash Pages section of the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

5.5.23. Erase UFM (0xCB)

Table 5.35. Erase UFM (0xCB)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CB	00 00 00	—	—	—

Notes: Erased condition for UFM bits = '0'.

Poll the BUSY bit (or wait, Refer to Table 10.1) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.

Example: 0xCB 00 00 00

Erase UFM sector.

5.5.24. Program SECURITY (0xCE)

Table 5.36. Program SECURITY (0xCE)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	CE	00 00 00	—	—	—

Example: 0xCE 00 00 00

Set the SECURITY bit.

Notes: Poll the BUSY bit (or wait 200 us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

SECURITY and SECURITY PLUS commands are mutually exclusive.

5.5.25. Program SECURITY PLUS (0xCF)

Table 5.37. Program SECURITY PLUS (0xCF)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	CF	00 00 00	—	—	—

Example: 0xCF 00 00 00

Set the SECURITY PLUS bit.

Notes: Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

SECURITY and SECURITY PLUS commands are mutually exclusive.

5.5.26. Read Device ID Code (0xE0)

Table 5.38. Read Device ID Code (0xE0)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	N	E0	00 00 00	R	4B	Refer to Table 5.39

Example: 0xE0 00 00 00
Read 4-byte device ID

Table 5.39. Device ID

Device Name	Device ID
LFMXO4-010HE	0xF1 2B 20 43
LFMXO4-010HC	0x71 2B A0 43
LFMXO4-010ZC	0x71 2B A0 43
LFMXO4-015HE	0x71 2B 20 43
LFMXO4-015HE BBG256	0xF1 2B 30 43
LFMXO4-015HC	0xF1 2B A0 43
LFMXO4-015HC BFG256	0xF1 2B B0 43
LFMXO4-015ZC	0xF1 2B A0 43
LFMXO4-025HE	0x71 2B 30 43
LFMXO4-025HC	0x71 2B B0 43
LFMXO4-025ZC	0x71 2B B0 43
LFMXO4-050HE	0x71 2B 40 43
LFMXO4-050HE BG400	0xF1 2B 50 43
LFMXO4-050HC	0x71 2B C0 43
LFMXO4-050HC BG400	0xF1 2B D0 43
LFMXO4-050HC TG256	0xF1 2B D0 43
LFMXO4-050ZC	0x71 2B C0 43
LFMXO4-080HE	0x71 2B 50 43
LFMXO4-080HC	0x71 2B D0 43
LFMXO4-080ZC	0x71 2B D0 43
LFMXO4-110HE	0x31 2B 60 43
LFMXO4-110HC	0x31 2B E0 43
LFMXO4-110ZC	0x31 2B E0 43

Example: 0xE0 00 00 00
Read 4-byte device ID

5.5.27. Verify Device ID Code (0xE2)

Table 5.40. Verify Device ID Code (0xE2)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	E2	00 00 00	W	4B	Refer to Table 5.39

Example: 0xE2 00 00 00 01 2B 20 43
Verify device ID with 32-bit input, sets ID Error bit 27 in SR if mismatched.

5.5.28. Program Feature (0xE4)

Table 5.41. Program Feature (0xE4)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	—	—	Y	E4	00 00 00	—	8B	00 00 ss uu cc cc cc cc

Data Format: ss: 8 bits for the user programmable I2C Target Address uu: 8 bits for the user programmable TraceID

cc cc cc cc: 32 bits of Custom ID code

Note: It is not recommended to reprogram the Feature Row in system, as it is more ideal to be programmed during manufacturing.

Example: 0xE4 00 00 00 00 00 01 00 00 00 12 34

Program Feature Row with User I2C address set to 1, default user TraceID string, Custom ID code of 12 34.

5.5.29. Read Feature Row (0xE7)

Table 5.42. Read Feature Row (0xE7)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	—	x	Y	E7	00 00 00	R	8B	00 00 ss uu cc cc cc cc

Data Format: ss: 8 bits for the user programmable I2C Target Address

uu: 8 bits for the user programmable TraceID

cc cc cc cc: 32 bits of Custom ID code

Example: 0xE7 00 00 00

Reads the Feature Row.

5.5.30. Check Busy Flag (0xF0)

Table 5.43. Check Busy Flag (0xF0)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
x	x	—	N	F0	00 00 00	R	1B	Bxxx xxxx

Data Format: b: bit 7: Busy Flag (1= busy)

(all other bits reserved)

Example: 0xF0 00 00 00

Read one byte, for example, 0x80 (busy flag set)

5.5.31. Program FEABITs (0xF8)

Table 5.44. Program FEABITs (0xF8)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	—	x	Y	F8	00 00 00	W	2B	00 bb mi sj di pa wv 00

Data Format: bb: Boot Sequence

1. If b=00 (Default) and m=0 then Single Boot from Flash.

2. If b=00 and m=1 then Dual Boot from Flash then External, if there is a failure.
3. If b=01 and m=1 then Single Boot from External Flash.
4. If b=10 and m=1 then Dual Boot from External then NVMC/Flash, if there is a failure.

m: Controller SPI Port Persistence
0=Disabled (Default), 1=Enabled

i: I2C Port Persistence
0=Enabled (Default), 1=Disabled

s: Target SPI Port Persistence
0=Enabled (Default), 1=Disabled

j: JTAG Port Persistence
0=Enabled (Default), 1=Disabled

d: DONE Persistence
0=Disabled (Default), 1=Enabled

i: INITN Persistence
0=Disabled (Default), 1=Enabled

p: PROGRAMN Persistence
0=Enabled (Default), 1=Disabled

a: my_ASSP Enabled
0=Disabled (Default), 1=Enabled

w: Password (Flash Protect Key) Protect All Enabled
0=Disabled (Default), 1=Enabled

v: Password (Flash Protect Key) Enabled
0=Disabled (Default), 1=Enabled

Note: It is not recommended to reprogram the FEABITs in system, as it is more ideal to be programmed during manufacturing.

Example: 0xF8 00 00 00 0D 20
Programs the FEABITs

5.5.32. Read FEABITs (0xFB)

Table 5.45. Read FEABITs (0xFB)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
—	—	x	Y	FB	00 00 00	R	2B	00 bb mi sj di pa wv 00

Data Format: bb: Boot Sequence

1. If b=00 (Default) and m=0 then Single Boot from Flash.
2. If b=00 and m=1 then Dual Boot from Flash then External, if there is a failure.
3. If b=01 and m=1 then Single Boot from External Flash.
4. If b=10 and m=1 then Dual Boot from External then NVMC/Flash, if there is a failure.

m: Controller SPI Port Persistence
0=Disabled (Default), 1=Enabled

i: INITN Persistence
0=Disabled (Default), 1=Enabled

s: Target SPI Port Persistence

0=Enabled (Default), 1=Disabled

j: JTAG Port Persistence

0=Enabled (Default), 1=Disabled

d: DONE Persistence

0=Disabled (Default), 1=Enabled

i: INITN Persistence

0=Disabled (Default), 1=Enabled

p: PROGRAMN Persistence

0=Enabled (Default), 1=Disabled

a: my_ASSP Enabled

0=Disabled (Default), 1=Enabled

w: Password (Flash Protect Key) Protect All Enabled

0=Disabled (Default), 1=Enabled

v: Password (Flash Protect Key) Enabled

0=Disabled (Default), 1=Enabled

5.5.33. Bypass (Null Operation) (0xFF)

Table 5.46. Bypass (Null Operation) (0xFF)

UFM	CFG Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	x	N	FF	FF FF FF	—	—	—

Note: Operands are optional.

Example: 0xFF FF FF FF Bypass

6. MachXO4 RoT Flash Access

This section describes the MachXO4 RoT flash access. For the MachXO4 (LFMXO4) flash access details, refer to the MachXO4 (LFMXO4) Flash Access section.

The MachXO4 RoT Flash Logic interface can be accessed using the JTAG, SPI, I2C, or WISHBONE interfaces. The MachXO4 RoT Flash consists of many different sectors:

- Configuration Flash (includes USERCODE)
    - Two configuration memory sectors (CFG0 and CFG1)
- User Flash Memory (UFM)
    - Four UFM sectors (UFM0, UFM1, UFM2, and UFM3)
- Feature and Security Settings
    - Feature Row Sector (FEA)
    - Configuration Security setting sector (CSEC)
    - User Security Setting sector (USEC)
- Security Keys
    - Public Key sector (PUBKEY)
    - Advanced Encryption Standard (AES) Key sector (AESKEY)

The Flash is organized in pages. The Flash is not byte addressable. Each page has 128 bits (16 bytes).

6.1. Flash Access Ports

You can access the Flash Memory through the JTAG port (compliant with the IEEE 1149.1 and IEEE 1532 specifications), external Target SPI port, external I2C Primary port, and the internal WISHBONE interface of the EFB module. Figure 6.1 illustrates the interfaces to the Flash Memory sectors.

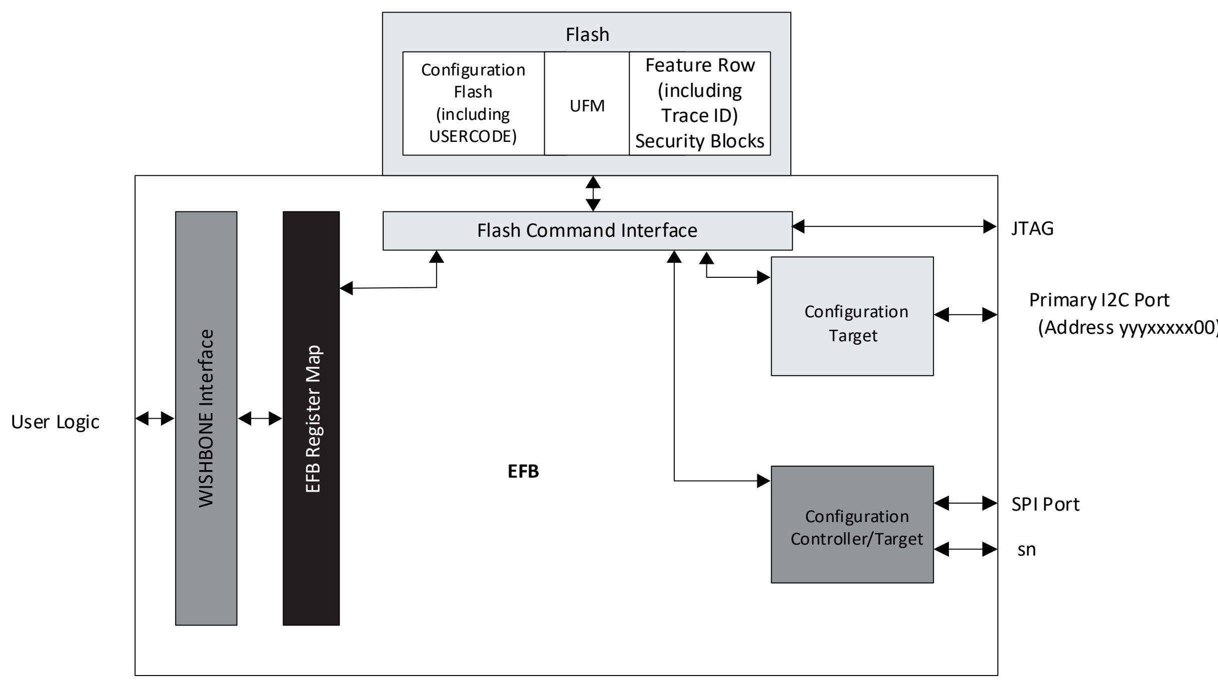


Figure 6.1. MachXO4 RoT Interfaces to the Flash Memory Sectors

The configuration logic arbitrates access from the interfaces according to the following priority. When higher priority ports are enabled, Flash access by lower priority ports is blocked.

- JTAG Port

- Target SPI Port
- I2C Primary Port
- WISHBONE Target Interface

Note: Enabling Flash Interface using Enable Configuration Interface command 0x74 Transparent Mode temporarily disables certain features of the device including:

- Power Controller
- GSR
- Hardened User SPI port
- Hardened User Primary I2C port

Functionality is restored after the Flash Interface is disabled using Disable Configuration Interface command 0x26 followed by Bypass command 0xFF.

6.2. Flash Access through WISHBONE Target Interface

The WISHBONE Target interface of the EFB module allows you to access the Flash directly from the FPGA core logic. The WISHBONE bus signals, described earlier in this document, are used by a WISHBONE host that you can implement using the general-purpose FPGA resources.

The WISHBONE Interface communicates to the Configuration Logic through a set of data, control, and status registers. Table 6.1 shows the register names and their functions. These registers are a subset of the EFB register map. Refer to the EFB register map for specific addresses of each register.

Table 6.1. WISHBONE to Flash Logic Registers

WISHBONE to CFG Register Name	Register Function	Address	Access
CFGCR	Control	0x70	Read/Write
CFGTXDR	Transmit Data	0x71	Write
CFGSR	Status	0x72	Read
CFGRXDR	Receive Data	0x73	Read
CFGIRQ	Interrupt Request	0x74	Read/Write
CFGIRQEN	Interrupt Request	0x75	Read/Write

Note: Unless otherwise specified, all reserved bits in writable registers shall be written *0*.

Table 6.2. Flash Control

CFGCR								0x70
Bit	7	6	5	4	3	2	1	0
Name	WBCE	RSTE	(Reserved)					
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	—	—	—	—	—	—

WBCE

WISHBONE Connection Enable. Enables the WISHBONE to establish the read/write connection to the Flash logic. This bit must be set prior to executing any command through the WISHBONE port. Likewise, this bit must be cleared to terminate the command. Refer to the Command and Data Transfers to Flash Memory Space section for more information on framing WISHBONE commands.

1: Enabled

0: Disabled

RSTE

WISHBONE Connection Reset. Resets the input/output FIFO logic. The reset logic is level sensitive. After setting this bit to *1*, it must be cleared to *0* for normal operation.

1: Reset

0: Normal operation

Table 6.3. Flash Transmit Data

CFGTXDR								0x71
Bit	7	6	5	4	3	2	1	0
Name	CFG_Transmit_Data[7:0]							
Default	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W

CFG_Transmit_Data[7:0]

CFG Transmit Data. This register holds the byte to be written to the Flash logic. Bit 0 is LSB.

Table 6.4. Flash Status

CFGSR								0x72
Bit	7	6	5	4	3	2	1	0
Name	WBCACT	(Reserved)	TXFE	TXFF	RXFE	RXFF	SSPIACT	I2CACT
Default	0	0	0	0	0	0	0	0
Access	R	—	R	R	R	R	R	R

WBCACT

WISHBONE Bus to Configuration Logic Active. Indicates that the WISHBONE to configuration interface is active and the connection is established.

1: WISHBONE Active

0: WISHBONE not Active

TXFE

Transmit FIFO Empty. Indicates that the Transmit Data register is empty. This bit is capable of generating an interrupt.

1: FIFO empty

0: FIFO not empty

TXFF

Transmit FIFO Full. Indicates that the Transmit Data register is full. This bit is capable of generating an interrupt.

1: FIFO full

0: FIFO not full

RXFE

Receive FIFO Empty. Indicates that the Receive Data register is empty. This bit is capable of generating an interrupt.

1: FIFO empty

0: FIFO not empty

RXFF

Receive FIFO Full. Indicates that the Transmit Data register is full. This bit is capable of generating an interrupt.

1: FIFO full

0: FIFO not full

SSPIACT

Target SPI Active. Indicates the Target SPI port has started actively communicating with the Configuration Logic while WBCE is enabled. This port has priority over the I2C and WISHBONE ports. It pre-empts any existing, and prohibits any new, lower priority transaction. This bit is capable of generating an interrupt.

1: Target SPI port active

0: Target SPI port not active

I2CACT

I2C Active. Indicates the I2C port has started actively communicating with the Configuration Logic while WBCE is enabled. This port has priority over the WISHBONE ports. It pre-empts any existing, and prohibits any new WISHBONE transaction. This bit is capable of generating an interrupt.

1: I2C port active

0: I2C port not active

Table 6.5. Flash Receive Data

CFGRXDR								0x73
Bit	7	6	5	4	3	2	1	0
Name	CFG_Receive_Data[7:0]							
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

CFG_Receive_Data[7:0]

CFG Receive Data. This register holds the byte read from the Flash logic. Bit 0 in this register is LSB.

Table 6.6. Flash Interrupt Status

CFGIRQ								0x74
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)		IRQTXFE	IRQTXFF	IRQRXFE	IRQRXFF	IRQSSPIACT	IRQI2CACT
Default	0	0	0	0	0	0	0	0
Access	—	—	R/W	R/W	R/W	R/W	R/W	R/W

IRQTXFE

Interrupt Status for Transmit FIFO Empty. When enabled, indicates TXFE is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmit FIFO Empty Interrupt

0: No interrupt

IRQTXFF

Interrupt Status for Transmit FIFO Full. When enabled, indicates TXFF is asserted. Write a *1* to this bit to clear the interrupt.

1: Transmit FIFO Full Interrupt

0: No interrupt

IRQRXFE

Interrupt Status for Receive FIFO Empty. When enabled, indicates RXFE is asserted. Write a *1* to this bit to clear the interrupt.

1: Receive FIFO Empty Interrupt

0: No interrupt

IRQRXFF

Interrupt Status for Receive FIFO Full. When enabled, indicates RXFF is asserted. Write a *1* to this bit to clear the interrupt.

1: Receive FIFO Full Interrupt

0: No interrupt

IRQSSPIACT

Interrupt Status for Target SPI Active. When enabled, indicates SSPIACT is asserted. Write a *1* to this bit to clear the interrupt.

1: Target SPI Active Interrupt

0: No interrupt

IRQI2CACT

Interrupt Status for I2C Active. When enabled, indicates I2CACT is asserted. Write a *1* to this bit to clear the interrupt.

1: I2C Active Interrupt

0: No interrupt

Table 6.7. Flash Interrupt Enable

CFGIRQEN								0x75
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)		IRQTXFEEN	IRQTXFFEN	IRQRXFEEN	IRQRXFFEN	IRQSSPIACTEN	IRQI2CACTEN
Default	0	0	0	0	0	0	0	0
Access	—	—	R/W	R/W	R/W	R/W	R/W	R/W

IRQTXFEEN

Interrupt Enable for Transmit FIFO Empty.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQTXFFEN

Interrupt Enable for Transmit FIFO Full.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQRXFEEN

Interrupt Enable for Receive FIFO Empty.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQRXFFEN

Interrupt Enable for Receive FIFO Full.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQSSPIACTEN

Interrupt Enable for Target SPI Active.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

IRQI2CACTEN

Interrupt Enable for I2C Active.

1: Interrupt generation enabled.

0: Interrupt generation disabled.

Table 6.8. Unused (Reserved) Register

UNUSED								0x76
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)							
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	—	—	—	—

Table 6.9. EFB Interrupt Source

EFBIRQ								0x77
Bit	7	6	5	4	3	2	1	0
Name	(Reserved)			CFG_INT	TC_INT	SPI_INT	I2C2_INT	I2C1_INT
Default	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

CFG_INT

Flash Interrupt Source. Indicates EFB interrupt source is from the Flash Block. Use CFGIRQ for further source resolution.

1: A bit is set in register CFGIRQ.

0: No interrupt.

TC_INT

Timer/Counter Interrupt Source. Indicates EFB interrupt source is from the Timer/Counter Block. Use TCIRQ for further source resolution.

1: A bit is set in register TCIRQ.

0: No interrupt.

SPI_INT

SPI Interrupt Source. Indicates EFB interrupt source is from the SPI Block. Use SPI-IRQ for further source resolution.

1: A bit is set in register SPIIRQ.

0: No interrupt.

I2C2_INT

I2C2 Interrupt Source. Indicates EFB interrupt source is from the Secondary I2C Block. Use I2C_2_ IRQ for further source resolution.

1: A bit is set in register I2C_2_ IRQ.

0: No interrupt.

I2C1_INT

I2C1 Interrupt Source. Indicates EFB interrupt source is from the Primary I2C Block. Use I2C_1_ IRQ for further source resolution.

1: A bit is set in register I2C_1_ IRQ.

0: No interrupt.

6.3. Command and Data Transfers to Flash Memory Space

The command and data transfers to the Flash Memory are identical for all the sysCONFIG access ports (JTAG, SSPI, I2C, and WISHBONE), regardless of their different physical interfaces. The Flash is organized in pages. Therefore, you address a specific page for Read or Write operations to that page. Each page has 128 bits (16 bytes). The transfers are based on a set of instructions and page addresses.

The Flash is composed of many different sectors:

- Configuration (CFG) Flash (includes USERCODE) has two sectors (CFG0 and CFG1)
- Four User Flash Memory (UFM) Sectors (UFM0, UFM1, UFM2, and UFM3)
- Feature and Security Policy (Feature Row) and
- Security Keys (PUBKEY and AESKEY sectors).

The Erase operations are sector based.

6.4. Command Summary by Application

6.4.1. Main Command Summary by Application

The following are the command summaries by application.

Table 6.10. Flash Commands

Command Name	Command MSB LSB	SVF Command Name	CFG	UFM	Feature, Security Policy	Security Keys	Description
Read Status, Busy, Bypass, Read Device ID, Refresh							
Read Status Register 0	0x3C	LSC_READ_STATUS0	Y	Y	Y	Y	Read the 4-byte Configuration Status Register.
Read Status Register 1	0x3D	LSC_READ_STATUS1	Y	Y	Y	Y	Read the 4-byte Configuration Status Register.
Check Busy Flag	0xF0	LSC_CHECK_BUSY	Y	Y	Y	Y	Read the Configuration Busy Flag status.
Bypass	0xFF	ISC_NOOP	Y	Y	Y	Y	Null operation.
Read Device ID	0xE0	IDCODE_PUB	Y	—	—	—	Read the 4-byte Device ID.
Read USERCODE	0xC0	USERCODE	Y	—	—	—	Read 32-bit USERCODE.
Read USERCODE_DRYRUN	0xC1	USERCODE_DRYRUN	Y	—	—	—	Read 32-bit Dryrun USERCODE used to test the various versions of bitstreams.
Address location for usercode (dryrun)	0xFC	LSC_PROG_DRYRUN_ADDR	Y	—	—	—	Address location of external flash dry run.
Refresh	0x79	LSC_REFRESH	—	—	—	—	Launch boot sequence (same as toggling PROGRAMN pin).
Bitstream_Check	0x7D	LSC_DEVICE_CTRL	—	—	—	—	Enables various bitstream checking options (dryrun, SED, and others), as well as device control options like standby, wakeup, reset, and others.
Verify Device ID	0xE2	LSC_VERIFY_ID	Y	—	—	—	Verify device ID with 32-bit input, set Fail flag if mismatched.
Enable Configuration (Transparent Mode), Enable (Offline Mode)							
Enable Configuration Interface (Transparent Mode)	0x74	ISC_ENABLE_X	Y	Y	Y	Y	Enable Transparent CFG/UFM/Feabits/Security_policy/Security_keys access – All user I/O (except the hardened user SPI and primary user I2C ports) are governed by the user logic, and the device remains in User mode. (The subsequent commands in this table require the interface to be enabled.)
Enable Configuration Interface (Offline	0xC6	ISC_ENABLE	Y	Y	Y	Y	Enable Offline UFM access – All user I/O (except persisted sysCONFIG ports)

Command Name	Command MSB LSB	SVF Command Name	CFG	UFM	Feature, Security Policy	Security Keys	Description
Mode)							are tri-stated. User logic ceases to function, UFM remains accessible, and the device enters *Offline* access mode. (The subsequent commands in this table require the interface to be enabled.)
Disable Configuration Interface	0x26	ISC_DISABLE	Y	Y	Y	Y	Disable the configuration (UFM) access.
Set/Reset Address							
Reset Configuration Flash Address	0x46	LSC_INIT_ADDRESS	Y	Y	Y	Y	Reset the address to point to Sector 0, page 0 of the active Flash sector.
Reset UFM Address	0x47	LSC_INIT_ADDR_UFM	—	Y	—	—	Reset the address to point to sector 1, page 0 of the UFM.
Set Address	0xB4	LSC_WRITE_ADDRESS	Y	Y	—	—	Set the UFM sector 14-bit Address.
Read/Program/Erase Commands							
Read Flash	0x73	LSC_READ_INCR_NV	Y	Y	Y	Y	Read the Flash data. Operand specifies number.
Erase Flash	0x0E	ISC_ERASE	Y	Y	Y	Y	Erase the Config Flash, UFM, FEATURE Row, FEABITs, Done bit, Security setting bits, security keys, and USERCODE.
Program Page	0x70	LSC_PROG_INCR_NV	Y	Y	Y	Y	Write one page of data to the Flash Memory (Configuration/UFM). Address Register is post-incremented.
Program DONE	0x5E	ISC_PROGRAM_DONE	Y	—	—	—	Program the Done bit.
Read UFM	0xCA	LSC_READ_TAG	—	Y	—	—	Read the UFM data. Operand specifies number pages to read address.
Erase UFM	0xCB	LSC_ERASE_TAG	—	Y	—	—	Erase the UFM sector only.
Program UFM Page	0xC9	LSC_PROG_TAG	—	Y	—	—	Write one page of data to the UFM. Address Register is post-incremented.
Program USERCODE	0xC2	ISC_PROGRAM_USERCODE	Y	—	—	—	Program 32-bit USERCODE.
Feature and Feature Bits Commands							
Read Feature Row	0xE7	LSC_READ_FEATURE	—	—	Y	—	Read Feature Row.
Program Feature Row	0xE4	LSC_PROG_FEATURE	—	—	Y	—	Program Feature Row.
Read FEABITS	0xFB	LSC_READ_FEABITS	—	—	Y	—	Read FEA bits.
Program FEABITS	0xF8	LSC_PROG_FEABITS	—	—	Y	—	Program FEA bits.
Security Key Commands							
Program ECDSA Public Key 0	0x59	LSC_PROG_ECDSA_PUBKEY0	—	—	—	Y	Program the ECDSA Public Key bit [127:0].

Command Name	Command MSB LSB	SVF Command Name	CFG	UFM	Feature, Security Policy	Security Keys	Description
Program ECDSA Public Key 1	0x5B	LSC_PROG_ECDSA_PUBKEY1	—	—	—	Y	Program the ECDSA Public Key bit [255:128].
Program ECDSA Public Key 2	0x61	LSC_PROG_ECDSA_PUBKEY2	—	—	—	Y	Program the ECDSA Public Key bit [383:256].
Program ECDSA Public Key 3	0x63	LSC_PROG_ECDSA_PUBKEY3	—	—	—	Y	Program the ECDSA Public Key bit [511:384].
Read ECDSA Public Key 0	0x5A	LSC_READ_ECDSA_PUBKEY0	—	—	—	Y	Read the ECDSA Public Key bit [127:0].
Read ECDSA Public Key 1	0x5C	LSC_READ_ECDSA_PUBKEY1	—	—	—	Y	Read the ECDSA Public Key bit [255:128].
Read ECDSA Public Key 2	0x62	LSC_READ_ECDSA_PUBKEY2	—	—	—	Y	Read the ECDSA Public Key bit [383:256].
Read ECDSA Public Key 3	0x64	LSC_READ_ECDSA_PUBKEY3	—	—	—	Y	Read the ECDSA Public Key bit [511:384].
Program AES Key 0	0xF3	LSC_PROG_CIPHER_KEY0	—	—	—	Y	Program the AES Key bit [127:0].
Program AES Key 1	0xFD	LSC_PROG_CIPHER_KEY1	—	—	—	Y	Program the AES Key bit [255:128].
Read AES Key 0	0xF4	LSC_READ_CIPHER_KEY0	—	—	—	Y	Read the AES Key bit [127:0].
Read AES Key 1	0xFE	LSC_READ_CIPHER_KEY1	—	—	—	Y	Read the AES Key bit [255:128].

Table 6.11. Non-Volatile Register (NVR) Commands

Command Name	Command MSB LSB	SVF Command Name	Description
Read TraceID code	0x19	UIDCODE_PUB	Read 64-bit TraceID.

When using the WISHBONE bus interface, the commands, operand, and data are written to the CFGTXDR Register. The Target SPI or I2C interface shifts the most significant bit (MSB) first into the MachXO4 RoT device. This is required only when communicating with the configuration logic inside the MachXO4 RoT device.

In order to perform a Write, Read, or Erase operation to the Flash, it is required that the interface is enabled using Command 0x74 or 0xC6. Affected commands are noted in the Command Description as ISC Enable. Once the modification operations are completed, the interface can be disabled using commands 0x26 and 0xFF in sequence.

6.4.2. Supplement Command Summary by Application

This section describes the command sequences and register map outlined in the Main Command Summary by Application section. This section also explains the centralized and user flash security settings (hard and soft lock bits), along with authentication commands.

Table 6.12. Flash Commands

Command Name	Command MSB LSB	SVF Command Name	CFG	UFM	Feature, Security Policy	Security Keys	Description
Read Status Register 1	0x3D	LSC_READ_STATUS1	Y	Y	Y	Y	Read the 4-byte Configuration Status Register 1.
Reset Configuration Flash Address	0x46	LSC_INIT_ADDRESS	Y	Y	Y	Y	Reset the address to point to Sector 0, page 0 of the active Flash sector.
Reset UFM Address	0x47	LSC_INIT_ADDR_UFM	—	Y	—	—	Reset the address to point to sector 1, page 0 of the UFM.

Command Name	Command MSB LSB	SVF Command Name	CFG	UFM	Feature, Security Policy	Security Keys	Description
Set Address	0xB4	LSC_WRITE_ADDRESS	Y	Y	Y	Y	Set the 14-bit Address to flash sector
Program USEC	0x56	LSC_PROG_USEC	—	—	Y	—	Program UFM security setting USEC)
Read USEC	0x57	LSC_READ_USEC	—	—	Y	—	Read UFM security setting USEC)
Program CSEC	0x54	LSC_PROG_CSEC	—	—	Y	—	Program centralized security setting for configuration, feature, and security keys settings (CSEC).
Read CSEC	0x55	LSC_READ_CSEC	—	—	Y	—	Read centralized security setting for configuration, feature, and security keys settings (CSEC).
Program Authentication Mode	0xC4	LSC_PROG_AUTH_MODE	—	—	Y	—	Program authentication mode bits to enable HMAC and ECDSA features.
Read Authentication Mode	0xC5	LSC_READ_AUTH_MODE	—	—	Y	—	Read authentication mode bits to enable HMAC and ECDSA features.
Program Authentication Done	0xCC	LSC_PROG_AUTH_DONE	—	—	Y	—	Program authentication done bit for faster booting applications.
Read FEABITS	0xFB	LSC_READ_FEABITS	—	—	Y	—	Read FEA bits.
Program FEABITS	0xF8	LSC_PROG_FEABITS	—	—	Y	—	Program FEA bits.

6.4.2.1. Read Status Register 1 (0x3D)

This command is used to read the Status Register 1. This register provides information about the flash sector selection, security, and lock settings. CSEC and USEC flash sectors information is described below.

Table 6.13. Read Status Register 1 (0x3D)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	3D	00 00 00	R	4B	xxxx xxxx xRGM BBBV YZFD CUAA LRPE SSSS

Data Format: SSSS bits[3:0]: Flash sector selection
0111: CSEC
1101: USEC
(all other bits reserved)

Usage: The BUSY bit should be checked following all Enable, Erase, or Program operations.

Note: Wait at least 1 µs after power-up or asserting wb_rst_i before accessing the EFB.

Example: 0x3D 00 00 00 Read four-byte Status Register. Return value example, 0x00 00 00 07 (CSEC flash sector is selected).

6.4.2.2. Reset Flash Address (0x46)

This command is used to reset the address counter to point to the first page of the various flash sectors. The flash sector is selected using operands. CSEC and USEC addresses are described below.

Table 6.14. Reset Flash Address (0x46)

ISC Enable	CMD (Hex)	Operands (3 Bytes) (Hex)	Data Mode	Data Size	Data Format
Y	46	Refer below	—	—	—

Operand: 0000 0fap ucwx yzpq 0000 0000(binary)

Where:
- u: Reset address to User Security sector - USEC (Security Setting for UFM0,1,2 & 3).
  - 0: No action
  - 1: Reset Address
- c: Reset address to Centralized security sector - CSEC (Security setting for CFG0, CFG1, Feature row, AESKEY and Public Key sectors).
  - 0: No action
  - 1: Reset Address

Examples: 0x46 00 80 00 Address is reset to point to USEC sector.

6.4.2.3. Reset UFM Address (0x47)

This command is used to reset the address counter to point to the first page of the various user flash sectors (UFMs). The user flash sector is selected using operands. USEC address is described below.

Table 6.15. Reset UFM Address (0x47)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	47	Refer below	—	—	—

Operand: 0000 0000 u0wx yz00 0000 0000(binary)

Where:
- u: Reset address to User Security sector - USEC (Security Setting for UFM0,1,2 & 3).
  - 0: No action
  - 1: Erase

Examples: 0x47 00 80 00 Address is reset to point to USEC sector.

6.4.2.4. Set Address (0xB4)

This command is used to set the address register for various flash sectors. CSEC and USEC addresses are described below.

Table 6.16. Set Address (0xB4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	B4	00 00 00	W	4B	0000 0000 0000 000ss ssaa aaaa aaaa aaaa

Data Format:
- ssss: Select Flash Sector
  - 0111: CSEC
  - 1011: USEC
- aa..aa: address 14-bit page address

Example: 0xB4 00 00 00 00 01 30 0A Set Address register to CSEC sector, page 10 decimal

6.4.2.5. Program UFM Security Settings (0x56)

This command is used to program the UFM security settings page.

Table 6.17. Program USEC (0x56)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	56	00 00 00	W	2B	0000 abcd efgh ijkl

Data Format: Most significant byte of this register is received first, LSB last.

xxxx Reserved (default = 0)

a Hard Lock enable for UFM3 sector (0 = Disabled (default), 1 = Enable)

b Read protection enable for UFM3 sector (0 = Read possible, 1 = Read disable)

c Erase protection enable for UFM3 sector (0 = Erase possible, 1 = Erase disable)

d Hard Lock enable for UFM2 sector (0 = Disabled (default), 1 = Enable)

e Read protection enable for UFM2 sector (0 = Read possible, 1 = Read disable)

f Erase protection enable for UFM2 sector (0 = Erase possible, 1 = Erase disable)

g Hard Lock enable for UFM1 sector (0 = Disabled (default), 1 = Enable)

h Read protection enable for UFM1 sector (0 = Read possible, 1 = Read disable)

i Erase protection enable for UFM1 sector (0 = Erase possible, 1 = Erase disable)

j Hard Lock enable for UFM0 sector (0 = Disabled (default), 1 = Enable)

k Read protection enable for UFM0 sector (0 = Read possible, 1 = Read disable)

l Erase protection enable for UFM0 sector (0 = Erase possible, 1 = Erase disable)

Example: 0x56 00 00 00 00 02

Lock the UFM0 flash sector to prohibit reading of that flash sector.

Note: Poll the BUSY bit (or wait 200 us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.4.2.6. Read UFM Security Settings (0x57)

This command is used to read the UFM security settings page.

Table 6.18. Read USEC (0x57)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	57	00 00 00	R	2B	xxxx abcd efgh ijkl

Data Format: Most significant byte of this register is received first, LSB last.

xxxx Reserved (default = 0)

a Hard Lock enable for UFM3 sector (0 = Disabled (default), 1 = Enable)

b Read protection enable for UFM3 sector (0 = Read possible, 1 = Read disable)

c Erase protection enable for UFM3 sector (0 = Erase possible, 1 = Erase disable)

d Hard Lock enable for UFM2 sector (0 = Disabled (default), 1 = Enable)

e Read protection enable for UFM2 sector (0 = Read possible, 1 = Read disable)

f Erase protection enable for UFM2 sector (0 = Erase possible, 1 = Erase disable)

g Hard Lock enable for UFM1 sector (0 = Disabled (default), 1 = Enable)

h Read protection enable for UFM1 sector (0 = Read possible, 1 = Read disable)

i Erase protection enable for UFM1 sector (0 = Erase possible, 1 = Erase disable)

j Hard Lock enable for UFM0 sector (0 = Disabled (default), 1 = Enable) k Read protection enable for UFM0 sector (0 = Read possible, 1 = Read disable)

l Erase protection enable for UFM0 sector (0 = Erase possible, 1 = Erase disable)

Example: 0x57 00 00 00 00 02

Read the UFM Security settings sector.

UFM0 flash sector is prohibited from reading

6.4.2.7. Program Centralized Security Setting (0x54)

This command is used to program the centralized security settings for all other Flash sectors except UFM. For example: CFG, feature, and security keys settings, and others.

Table 6.19. Program CSEC (0x54)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	54	00 00 00	W	4B	0bcd ehii fssl jjag kmno pqrt uvwy zαβξ

Data Format:

0: reserved

b: Hard Lock enable for I2C Bridge (0 = Disabled (default), 1 = Enable)

c: I2C Bridge locked (0 = Disabled (default), 1 = Enable)
I2C bridge commands are disabled once this bit is set

d: Hard Lock enable for JTAG to SPI Bridge (0 = Disabled (default), 1 = Enable)

e: JTAG to SPI Bridge is locked (0 = Disabled (default), 1 = Enable)
JTAG to SPI bridge commands are disabled once this bit is set

h: Hard Lock enable for Target I2C Port (0 = Disabled (default), 1 = Enable)

ii: Target I2C Port lock modes
Mode: Target I2C Port

Mode	Target I2C Port
00	Unlocked
01	Locked
10	Partially Locked[1]
11	Locked

Note:

1. Only certain commands are allowed to execute in partial lock mode

f: Hard Lock enable for Target SPI Port (0 = Disabled (default), 1 = Enable)

ss: Target SPI Port lock modes
Mode: Target SPI Port

Mode	Target SPI Port
00	Unlocked
01	Locked
10	Partially Locked[1]
11	Locked

Note:

1. Only certain commands are allowed to execute in partial lock mode

l: Hard Lock enable for JTAG Port (0 = Disabled (default), 1 = Enable)

jj: JTAG Port lock modes

Mode: JTAG Port

Mode	JTAG Port
00	Unlocked
01	Partially Locked[1]
10	Partially Locked[2]
11	Locked

Notes:

1. Only certain 1149 commands are allowed to execute in partial lock mode
2. Only certain 1532 and some 1149 commands are allowed to execute in partial lock mode

a g k: SRAM lock settings

SRAM Lock Settings			
a	Hard Lock enable	0	Disable (default)
		1	Enable
g	Read Protection enable	0	Read possible (default)
		1	Read disable
k	Erase Protection enable	0	Erase possible (default)
		1	Erase disable

m n o: AESKEY Lock Settings

AESKEY Lock Settings			
m	Hard Lock Enable	0	Disabled (default)
		1	Enable
n	Read Protection Enable	0	Read possible (default)
		1	Read disable
o	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

p q r: Public Key Lock Settings

Public Key Lock Settings			
p	Hard Lock Enable	0	Disabled (default)
		1	Enable
q	Read Protection Enable	0	Read possible (default)
		1	Read disable
r	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

t u v: Feature Row Lock Settings

Feature Row Lock Settings			
t	Hard Lock Enable	0	Disabled (default)
		1	Enable
u	Read Protection Enable	0	Read possible (default)
		1	Read disable
v	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

w y z: CFG1 Sector Lock Settings

CFG1 Lock Settings			
w	Hard Lock Enable	0	Disabled (default)
		1	Enable
y	Read Protection Enable	0	Read possible (default)
		1	Read disable
z	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

α β ξ: CFG0 Sector Lock Settings

CFG0 Lock Settings			
α	Hard Lock Enable	0	Disabled (default)
		1	Enable
β	Read Protection Enable	0	Read possible (default)
		1	Read disable
ξ	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

Example: 0x54 00 00 00 0D 20 Program centralized security settings

6.4.2.8. Read Centralized Security Setting (0x55)

This command is used to read the centralized security settings for all other Flash sectors except UFM. For example: CFG, feature, and security keys settings.

Table 6.20. Read CSEC (0x55)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	55	00 00 00	W	4B	xbcd ehii fssl jjag kmno pqrt uvwy zαβξ

Data Format:

x: reserved

b: Hard Lock enable for I2C Bridge (0 = Disabled (default), 1 = Enable)

c: I2C Bridge locked (0 = Disabled (default), 1 = Enable)

I2C bridge commands are disabled once this bit is set d: Hard Lock enable for JTAG to SPI Bridge (0 = Disabled (default), 1 = Enable)

e: JTAG to SPI Bridge is locked (0 = Disabled (default), 1 = Enable)
JTAG to SPI bridge commands are disabled once this bit is set

h: Hard Lock enable for Target I2C Port (0 = Disabled (default), 1 = Enable)

ii: Target I2C Port lock modes
Mode: Target I2C Port

Mode	Target I2C Port
00	Unlocked
01	Locked
10	Partially Locked[1]
11	Locked

Note:

1. Only certain commands are allowed to execute in partial lock mode

f: Hard Lock enable for Target SPI Port (0 = Disabled (default), 1 = Enable)

ss: Target SPI Port lock modes
Mode: Target SPI Port

Mode	Target SPI Port
00	Unlocked
01	Locked
10	Partially Locked[1]
11	Locked

Note:

1. Only certain commands are allowed to execute in partial lock mode

l: Hard Lock enable for JTAG Port (0 = Disabled (default), 1 = Enable)

jj: JTAG Port lock modes
Mode: JTAG Port

Mode	JTAG Port
00	Unlocked
01	Partially Locked[1]
10	Partially Locked[2]
11	Locked

Notes:

1. Only certain 1149 commands are allowed to execute in partial lock mode
2. Only certain 1532 and some 1149 commands are allowed to execute in partial lock mode

a g k: SRAM lock settings

SRAM Lock Settings			
a	Hard Lock enable	0	Disable (default)
		1	Enable
g	Read Protection enable	0	Read possible (default)

SRAM Lock Settings			
		1	Read disable
k	Erase Protection enable	0	Erase possible (default)
		1	Erase disable

m n o: AESKEY Lock Settings

AESKEY Lock Settings			
m	Hard Lock Enable	0	Disabled (default)
		1	Enable
n	Read Protection Enable	0	Read possible (default)
		1	Read disable
o	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

p q r: Public Key Lock Settings

Public Key Lock Settings			
p	Hard Lock Enable	0	Disabled (default)
		1	Enable
q	Read Protection Enable	0	Read possible (default)
		1	Read disable
r	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

t u v: Feature Row Lock Settings

Feature Row Lock Settings			
t	Hard Lock Enable	0	Disabled (default)
		1	Enable
u	Read Protection Enable	0	Read possible (default)
		1	Read disable
v	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

w y z: CFG1 Sector Lock Settings

CFG1 Lock Settings			
w	Hard Lock Enable	0	Disabled (default)
		1	Enable
y	Read Protection Enable	0	Read possible (default)

CFG1 Lock Settings			
		1	Read disable
z	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

α β ξ: CFG0 Sector Lock Settings

CFG0 Lock Settings			
α	Hard Lock Enable	0	Disabled (default)
		1	Enable
β	Read Protection Enable	0	Read possible (default)
		1	Read disable
ξ	Erase Protection Enable	0	Erase possible (default)
		1	Erase disable

Example: 0x55 00 00 00 Read centralized security settings

6.4.2.9. Program Authentication Mode (0xC4)

This command is used to program the authentication mode bits.

Table 6.21. Program Authentication Mode (0xC4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	C4	00 00 00	W	1B	xxxx xxaa

Data Format: xxxx xx: Reserved

aa: Authentication Mode

aa	Authentication Mode
00	Authentication disabled
01	Authentication disabled
10	HMAC authentication enabled
11	ECDSA signature verification enabled

Example: 0xC4 00 00 00 03 ECDSA Signature verification is enabled.

6.4.2.10. Read Authentication Mode (0xC5)

This command is used to read the authentication mode bits.

Table 6.22. Program Authentication Mode (0xC4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	C5	00 00 00	W	1B	xxxx xxaa

Data Format: xxxx xx: Reserved

aa: Authentication Mode

aa	Authentication Mode
00	Authentication disabled
01	Authentication disabled
10	HMAC authentication enabled
11	ECDSA signature verification enabled

Example: 0xC4 00 00 00 00 03 Read the authentication mode. ECDSA Signature verification is enabled.

6.4.2.11. Authentication DONE (0xCC)

This command is used to program the authentication done bit.

Table 6.23. Authentication DONE (0xCC)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	CC	00 00 00	—	—	—

Example: 0xCC 00 00 00

Set the Authentication DONE bit.

Note: Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.4.2.12. Program FEABITs (0xF8)

This command is used to program the feature bits such as booting sequence selection, password settings, and others. CSEC and USEC flash sectors information is described below.

Table 6.24. Program FEABITs (0xF8)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	F8	00 00 00	W	4B	0000 0000 0000 000e cbbb misj dnpa wvug

Data Format:

w	v	u	Flash Protection Sector Selection
0	0	0	No protection to any Flash sector
0	0	1	All UFMs
0	1	0	Feature, Security Keys, CSEC
0	1	1	Feature, Security Keys, CSEC, USEC, and all UFMs
1	0	0	CFG0 and CFG1
1	0	1	CFG0, CFG1, CSEC, all UFMs, and USEC
1	1	0	Feature, Security Keys, CSEC, CFG0, and CFG1
1	1	1	Feature, Security Keys, CSEC, CFG0, CFG1, USEC, and all UFMs

Example: 0xF8 00 00 00 0D 20 Programs the FEABITs

6.4.2.13. Read FEABITs (0xF8)

This command is used to read the feature bits such as booting sequence selection, password settings, and others. CSEC and USEC flash sectors information is described below.

Table 6.25. Read FEABITs (0xF8)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	FB	00 00 00	R	4B	xxxx xxxx xxxx xxxe cbbb misj dnpa wvug

Data Format:

w	v	u	Flash Protection Sector Selection
0	0	0	No protection to any Flash sector
0	0	1	All UFMs
0	1	0	Feature, Security Keys, CSEC
0	1	1	Feature, Security Keys, CSEC, USEC, and all UFMs
1	0	0	CFG0 and CFG1
1	0	1	CFG0, CFG1, CSEC, all UFMs, and USEC
1	1	0	Feature, Security Keys, CSEC, CFG0, and CFG1
1	1	1	Feature, Security Keys, CSEC, CFG0, CFG1, USEC, and all UFMs

Example: 0xF8 00 00 00 Read the FEABITs

6.5. Command Descriptions by Command Code

All command descriptions have the following command format.

Table 6.26. Fields of Command Code

ISC Enable	CMD (Hex)	Operands (3B) (Hex)	Data Mode	Data Size	Data Format
Refer below	0E	Refer below	Refer below	Refer below	Refer below

ISC Enable: This column signifies when the device can be accessed. A Y indicates the command requires the device to be in edit mode (offline or transparent).

CMD (Hex): Eight bit command, in hex

Operands: Two or three bytes of operands required when executing the command. Non-bold bits can be combined for logic OR operations. Bold bits are mutually exclusive.

Data Mode: Read or Write

Data Size: Size of data in bytes (For example, 2B or 4B)

Data Format: Describes the valid bits while reading/writing the command

6.5.1. Erase Flash (0x0E)

This command erases the Flash sectors for CFG0, CFG1, UFM0, UFM1, UFM2, UFM3, FEATURE Row, FEABITs, Done bit, Security bits, and USERCODE.

Table 6.27. Erase Flash (0x0E)

ISC Enable	CMD (Hex)	Operands (3B) (Hex)	Data Mode	Data Size	Data Format
Y	0E	Refer below	—	—	—

Operand: 0000 0fap 00wx yzpq 0000 0000 (binary)

Where:

f: Erase Feature sector (Target I2C address, sysCONFIG port persistence, Bootmode)
- 0: No action
- 1: Erase

a: Erase AES Key
- 0: No action
- 1: Erase

p: Erase Public Key sector
- 0: No action
- 1: Erase

w: Erase UFM3
- 0: No action
- 1: Erase

x: Erase UFM2
- 0: No action
- 1: Erase

y: Erase UFM1
- 0: No action
- 1: Erase

z: Erase UFM0

0: No action
1: Erase
p: Erase CFG1
0: No action
1: Erase
q: Erase CFG0
0: No action
1: Erase

Note: Poll the BUSY bit (or wait; refer to Table 10.1) after issuing this command to make sure erase is complete before issuing a subsequent command other than Read Status or Check Busy.

Erased condition for Flash bits = 0.

Examples:
0x0E 00 01 00 Erase CFG0 sector.
0x0E 00 20 00 Erase UFM3 sector.
0x0E 00 23 00 Erase UFM3 sector and CFG0 and CFG1 sectors.

6.5.2. Read TraceID Code (0x19)

This commands reads the TraceID of MachXO4 RoT device.

Table 6.28. Read TraceID Code (0x19)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	19	00 00 00	R	8B	—

Example: 0x19 00 00 00 Read 8-byte TraceID

Notes: This command is used for non-volatile register.

First byte read is user portion. Next seven bytes are unique to each silicon die.

6.5.3. Disable In-system Configuration Access (0x26)

This command is used to disable the In-System Configuration (ISC) Access. After issuing this command all commands that require ISC qualification are nullified.

Table 6.29. Disable Configuration Interface (0x26)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	26	00 00	—	—	—

Example: 0x26 00 00 Disable Flash interface for change access.

Notes: Must have only two operands.

The interface cannot be disabled while the Configuration Status Register Busy bit is asserted. After commands (for example, Erase, Program), verify Busy is clear before issuing the Disable command.

This command should be followed by Command 0xFF (BYPASS) to complete the Disable operation. The BYPASS command is required to restore Power Controller, GSR, Hardened User SPI, and I2C port operation.

SRAM must be erased before exiting Offline (0xC6) Mode.

6.5.4. Read Status Register 0 (0x3C)

This command is used to read the Status Register 0. This register provides information about the status of done bit, configuration flow states, busy, and fail flags.

Table 6.30. Read Status Register 0 (0x3C)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
N	3C	00 00 00	R	4B	xxxx IxEE EEPx xAUV WxFB xxCD xxxx xxxx

Data Format:

I bit 27: I=0 Device verified correct, I=1 Device failed to verify

EEEE bits[25:22]: Configuration Status Check

0000: No Error
0001: ID ERR
0010: CMD ERR
0011: CRC ERR
0100: Preamble ERR
0101: Abort ERR
0110: Overflow ERR
0111: SDM EOF
1000: Authentication fail
1001: Authentication setup error (CRE unavailable or setup error)
1010: Authentication Bitstream error
1011: Target mode booting failure due to time out
1100: Version check failure for version rollback protection
(all other bits reserved)

P bit 21: Primary boot failure (1= Fail) even though secondary boot successful

A bit 18: Authentication done

U bit 17: Password Protection enabled for all UFM flash sectors
0=Disabled (Default), 1=Enabled

V bit 16: Password Protection Enabled for Feature and Security Key flash sectors
0=Disabled (Default), 1=Enabled

W bit 15: Password Protection All Enabled for CFG0 and CFG1 flash sectors
0=Disabled (Default), 1=Enabled

F bit 13: Fail Flag (1 = Operation failed)

B bit 12: Busy Flag (1 = Busy)

C bit 9: Enable Configuration Interface (1=Enable, 0=Disable)

D bit 8: Flash or SRAM Done Flag

When C = 0 SRAM Done bit has been programmed:

- D = 1 Successful Flash to SRAM transfer
- D = 0 Failure in the Flash to SRAM transfer

When C=1 Flash Done bit has been programmed:

- D = 1 Programmed
- D = 0 Not Programmed

Usage: The BUSY bit should be checked following all Enable, Erase, or Program operations.

Note: Wait at least 1 µs after power-up or asserting wb_rst_i before accessing the EFB.

Example: 0x3C 00 00 00

Read 4-byte Status Register. Return value example: 0x00 00 20 00 (fail flag set).

6.5.5. Read Status Register 1 (0x3D)

This command is used to read the Status Register 1. This register provides information about the flash sector selection, security, and lock settings.

Table 6.31. Read Status Register 1 (0x3D)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
N	3D	00 00 00	R	4B	xxxx xxxx xRGM BBBV YZFD CUAA LRPE SSSS

Data Format:

R: I2C deglitch filter range selection on primary I2C port

0: 8 to 25 ns range (Default)

1: 16 to 50 ns range

(all other bits reserved)

G: I2C deglitch filter enable for Primary I2C Port

0=Disabled (Default), 1=Enabled

M: bit 20: Controller SPI Port Persistence

0=Disabled (Default), 1=Enabled

BBB: bit [19:17] Boot Sequence selection (used along with Controller SPI Port Persistence bit)

BBB	M	Boot Mode	Boot From
000	0	Dual	CFG0 - CFG1
000	1	Dual	CFG0 – Ext
X01	1	Single	Ext
010	1	Dual	Ext – CFG0
X11	1	Dual	Ext – Ext
011	0	Single	CFG0
001	0	Dual	CFG1 – CFG0
X10	0	Dual	No Boot
100	0	Single	CFG1
100	1	Dual	CFG1 – Ext
110	1	Dual	Ext - CFG1
111	0	Dual	Boot from latter (newer) bitstream first
101	0	Dual	Boot from former (older) Bitstream first

V bit 16: Bitstream version

1 = Bitstream in CFG0 is latter (newer) than CFG1

0 = Bitstream in CFG1 is latter (newer) than CFG0

Y bit 15: Security Plus enabled for CFG1 (1 = Enabled, 0 = Disabled)

Z bit 14: Security Plus enabled for CFG0 (1 = Enabled, 0 = Disabled)

F bit 13: Flash done bit is programmed of CFG1 (1= Programmed, 0=Unprogrammed)

D bit 12: Flash done bit is programmed of CFG0 (1= Programmed, 0=Unprogrammed)

C bit 11: Authentication done for CFG1 (1 = Authentication successful)

U bit 10: Authentication done for CFG0 (1 = Authentication successful)

AA bit [9:8]: Authentication mode

0x: No Authentication

10: HMAC Authentication

11: ECDSA Signature Verification

L bit 7: Hard/Soft Lock Selection (1 = Hard Lock, 0 = Soft Lock)

R bit 6: Read operation is prohibited (1 = Read disable)

P bit 5: Program operation is prohibited (1 = programming disable)

E bit 4: Erase operation is prohibited (1 = Erase disable)

SSSS bits[3:0]: Flash sector selection

0001: CFG0

0010: CFG1

0011: Reserved

0100: Feature

0101: Public Key

0110: AES Key

1001: UFM0

1010: UFM1

1011: UFM2

1100: UFM3

(all other bits reserved)

Usage: The BUSY bit should be checked following all Enable, Erase, or Program operations.

Note: Wait at least 1 µs after power-up or asserting wb_rst_i before accessing the EFB.

Example: 0x3D 00 00 00 Read 4-byte Status Register. Return value example: 0x00 00 00 01 (CFG0 flash sector is selected).

6.5.6. Reset Flash Address (0x46)

This command is used to reset the address counter to point to the first page of the different flash sectors. The flash sector is selected using operands.

Table 6.32. Reset Flash Address (0x46)

ISC Enable	CMD (Hex)	Operands (3 Bytes) (Hex)	Data Mode	Data Size	Data Format
Y	46	Refer below	—	—	—

Operand: 0000 0**fap** 00**wx yzpq** 0000 0000(binary)

Where: f: Reset address to Feature sector (Target I2C address, sysCONFIG port persistence, Bootmode)

0: No action

1: Reset address

a: Reset address to AES Key sector

0: No action

1: Reset address

p: Reset address to Public Key sector

	0:	No action
	1:	Reset address
w:	Reset address to UFM3	
	0:	No action
	1:	Reset address
x:	Reset address to UFM2	
	0:	No action
	1:	Reset address
y:	Reset address to UFM1	
	0:	No action
	1:	Reset address
z:	Reset address to UFM0	
	0:	No action
	1:	Reset address
p:	Reset address to CFG1	
	0:	No action
	1:	Reset address
q:	Reset address to CFG0	
	0:	No action
	1:	Reset address

Examples: 0x46 00 01 00 Address is reset to point to CFG0 sector.

6.5.7. Reset UFM Address (0x47)

This command is used to reset the address counter to point to the first page of the different user flash sectors (UFMs). The user flash sector is selected using operands.

Table 6.33. Reset UFM Address (0x47)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	47	Refer below	—	—	—

Operand: 0000 0000 00**wx yz**00 0000 0000(binary)

Where:

w:	Reset address to UFM3	
	0:	No action
	1:	Reset address
x:	Reset address to UFM2	
	0:	No action
	1:	Reset address
y:	Reset address to UFM1	
	0:	No action
	1:	Reset address
z:	Reset address to UFM0	
	0:	No action
	1:	Reset address

Examples: 0x47 00 04 00 Address is reset to point to UFM0 sector.

6.5.8. Perform Bitstream Check and Device Control (0x7D)

This command is used to enable various bitstream checking options (dryrun, SED, and others), as well as device control options like standby, wakeup, reset, and others.

Table 6.34. Perform Bitstream Check and Device Control (0x7D)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	7D	Refer below	—	—	—

Operand: 0**abc def**0 0000 0000 0000 0000(binary)

Where:

- a: Reset CFG Logic
  - 0: No action
  - 1: Reset CFG Logic
- bcd: Bitstream Check
  - 000: No action
  - 001: Launch one-time check for SED
  - 010: Launch a bitstream dry run for CFG0 sector.
  - 011: Launch bitstream dry run for CFG1 sector
- e: Wake UP
  - 0: No action
  - 1: Device wakes up from Standby mode
- f: STANDBY
  - 0: No action
  - 1: Device enters the Standby Mode

Example: 0x7D 40 00 00 This operation perform bitstream dry run on CFG0 Sector.

6.5.9. Program ECDSA PUBKEY0 (0x59)

This command is used to program the first 128 bits of the ECDSA Public Key.

Table 6.35. Program ECDSA Public Key 0 (0x59)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	59	00 00 00	W	16B	16 bytes of write data (first 128 bits)

Example: 0x59 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the first 128 bits (bits [127:0]) of the ECDSA Public Key, where 0F is LSB byte [7:0] and 00 is MSB byte [127:120].

Notes: 16 data bytes must be written following the command and operand bytes to ensure correct data alignment. This command writes first 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.10. Read ECDSA PUBKEY0 (0x5A)

This command is used to read the first 128 bits of the ECDSA Public Key.

Table 6.36. Read ECDSA Public Key 0 (0xC9)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	5A	00 00 00	R	16B	16 bytes of read data (first 128 bits)

Example: 0x5A 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the first 128 bits (bits [127:0]) of the ECDSA Public Key, where 0F is LSB byte [7:0] and 00 is MSB byte [127:120].

Notes: 16 data bytes must be read following the command and operand bytes to ensure correct data alignment. This command reads the first 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.11. Program ECDSA PUBKEY1 (0x5B)

This command is used to program the second 128 bits of the ECDSA Public Key.

Table 6.37. Program ECDSA Public Key 1 (0x5B)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	5B	00 00 00	W	16B	16 bytes of write data (second 128 bits)

Example: 0x5B 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the second 128 bits (bits [255:128]) of the ECDSA Public Key, where 0F is LSB byte [135:128] and 00 is MSB byte [255:248].

Notes: 16 data bytes must be written following the command and operand bytes to ensure correct data alignment. This command writes second 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.12. Read ECDSA PUBKEY1 (0x5C)

This command is used to read the second 128 bits of the ECDSA Public Key.

Table 6.38. Read ECDSA Public Key 1 (0x5C)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	5C	00 00 00	R	16B	16 bytes of read data (second 128 bits)

Example: 0x5C 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the second 128 bits (bits [255:128]) of the ECDSA Public Key, where 0F is LSB byte [135:128] and 00 is MSB byte [255:248].

Notes: 16 data bytes must be read following the command and operand bytes to ensure correct data alignment. This command reads the second 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.13. Program ECDSA PUBKEY2 (0x61)

This command is used to program the third 128 bits of the ECDSA Public Key.

Table 6.39. Program ECDSA Public Key 2 (0x61)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	61	00 00 00	W	16B	16 bytes of write data (third 128 bits)

Example: 0x61 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the third 128 bits (bits [383:256]) of the ECDSA Public Key, where 0F is LSB byte [263:256] and 00 is MSB byte [383:376].

Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. This command writes third 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.14. Read ECDSA PUBKEY2 (0x62)

This command is used to read the third 128 bits of the ECDSA Public Key.

Table 6.40. Read ECDSA Public Key 2 (0x62)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	62	00 00 00	R	16B	16 bytes of read data (third 128 bits)

Example: 0x62 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the third 128 bits (bits [383:256]) of the ECDSA Public Key, where 0F is LSB byte [263:256] and 00 is MSB byte [383:376].

Notes: 16 data bytes must be read following the command and operand bytes to ensure correct data alignment. This command reads the third 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.15. Program ECDSA PUBKEY3 (0x63)

This command is used to program the forth 128 bits of the ECDSA Public Key.

Table 6.41. Program ECDSA Public Key 3 (0x63)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	63	00 00 00	W	16B	16 bytes of write data (fourth 128 bits)

Example: 0x63 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the forth 128 bits (bits [511:384]) of the ECDSA Public Key, where 0F is LSB byte [391:384] and 00 is MSB byte [511:504].

Notes: 16 data bytes must be written following the command and operand bytes to ensure correct data alignment. This command writes forth 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.16. Read ECDSA PUBKEY3 (0x64)

This command is used to read the forth 128 bits of the ECDSA Public Key.

Table 6.42. Read ECDSA Public Key 3 (0x64)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	64	00 00 00	R	16B	16 bytes of read data (forth 128 bits)

Example: 0x64 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the forth 128 bits (bits [511:384]) of the ECDSA Public Key, where 0F is LSB byte [391:384] and 00 is MSB byte [511:504] .

Notes: 16 data bytes must be read following the command and operand bytes to ensure correct data alignment. This command reads the forth 128 bits of the 512 bits of ECDSA Public Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.17. Program CIPHER KEY0 (0xF3)

This command is used to program the first 128 bits of the AES Cipher Key.

Table 6.43. Program Cipher Key 0 (0xF3)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	F3	00 00 00	W	16B	16 bytes of write data (first 128 bits)

Example: 0xF3 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the first 128 bits (bits [127:0]) of the AES Cipher Key, where 0F is LSB byte [7:0] and 00 is MSB byte [127:120].

Notes: 16 data bytes must be written in the following command and operand bytes to ensure the data are align correctly. This command writes first 128 bits of the 256 bits of Cipher Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.18. Read CIPHER KEY0 (0xF4)

This command is used to read the first 128 bits of the AES Cipher Key.

Table 6.44. Read Cipher Key 0 (0xF4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	F4	00 00 00	R	16B	16 bytes of read data (first 128 bits)

Example: 0xF4 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the first 128 bits (bits [127:0]) of the Cipher Key, where 0F is LSB byte [7:0] and 00 is MSB byte [127:120].

Notes: 16 data bytes must be read in the following the command and operand bytes to ensure data are correctly aligned. This command reads the first 128 bits of the 256 bits of Cipher Key.
Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.19. Program CIPHER KEY1 (0xFD)

This command is used to program the second 128 bits of the AES Cipher Key.

Table 6.45. Program Cipher Key 1 (0xFD)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	FD	00 00 00	W	16B	16 bytes of write data (second 128 bits)

Example: 0xFD 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write the second 128 bits (bits [255:128]) of the Cipher Key, where 0F is LSB byte [135:128] and 00 is MSB byte [255:248].

Notes: 16 data bytes must be written in the following command and operand bytes to ensure data are correctly aligned. This command writes second 128 bits of the 256 bits of Cipher Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.20. Read CIPHER KEY1 (0xFE)

This command is used to read the second 128 bits of the AES Cipher Key.

Table 6.46. Read Cipher Key 1 (0xFE)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	FE	00 00 00	R	16B	16 bytes of read data (second 128 bits)

Example: 0xFE 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Read the second 128 bits (bits [255:128]) of the Cipher Key, where 0F is LSB byte [135:128] and 00 is MSB byte [255:248].

Notes: 16 data bytes must be read in the following command and operand bytes to ensure data are correctly aligned. This command reads the second 128 bits of the 256 bits of Cipher Key.

Poll the BUSY bit (or wait 200 µs) after issuing this command to complete the reading of data before issuing a subsequent command other than Read Status or Check Busy.

6.5.21. Program DONE (0x5E)

This command is used to program the done bit

Table 6.47. Program DONE (0x5E)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	5E	00 00 00	—	—	—

Example: 0x5E 00 00 00 Set the DONE bit.

Note: Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.22. Program Flash (0x70)

This command is used to program the particular flash sector one page at a time.

Table 6.48. Program Flash (0x70)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	70	00 00 00	W	16B	16 bytes of write data

Example: 0x70 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write one page of data, pointed to by Address Register.

Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. The Address Register is auto incremented following the page write.

If necessary, use 0x0E to erase the desired flash sector prior to executing this command.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.23. Read Flash (0x73) (SPI – Option 1)

This command is used to read the flash pages (any flash sector) using SPI.

Table 6.49. Read Flash (0x73) (SPI)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note: This applies when Flash (any flash sector) is read through SPI.

1. Operand: 0001 0000 00pp pppp pppp pppp (binary)

   pp..pp: num_pages Number of CFG Flash pages to read when num_pages = 1

   Number of CFG Flash pages to read +1 when num_pages > 1

Note: When num_pages = 0, one page is I/O read.

2. Data Size: 16 bytes when num_pages <= 1

   (num_pages × 16) bytes when num_pages > 1

Note: Read Flash may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. Flash page is read from the address pointed by the Address Register. The Address Register is auto incremented after each page read.

3. Examples: 0x73 10 00 01 0-byte dummy followed by one page of Flash data (16 bytes total).

   0x73 10 00 04 Read 1-page dummy followed by three pages of Flash data (64 bytes total).

6.5.24. Read Flash (0x73) (I2C/SPI – Option 2)

This command is used to read the flash pages (any flash sector) using I2C or SPI.

Table 6.50. Read Flash (0x73) (I2C/SPI)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note: This applies when Flash is read through I2C or SPI.

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)

pp..pp: num_pages — Number of CFG Flash pages to read when num_pages = 1
Number of CFG Flash pages to read +1 when num_pages > 1

2. Data Size: (num_pages × 16) bytes when num_pages=1
(num_pages) × (16 + 4) bytes when num_pages>1

Note: Read Flash may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. Flash page is read from the address pointed by the Address Register. The Address Register is auto incremented after each page read.

3. Examples: 0x73 00 00 01 Read 0-byte dummy followed by one page Flash data (16 bytes total)
0x73 00 00 04 Read 1-page dummy, followed by four dummy bytes, followed by three sets [1 page Flash data, followed by four bytes dummy] (48 data bytes, 32 dummy bytes).

6.5.25. Read Flash (0x73) (WISHBONE)

This command is used to read the flash pages (any flash sector) using WISHBONE.

Table 6.51. Read Flash (0x73) (WISHBONE)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	73	Refer below[1]	R	Refer below[2]	Refer below[3]

Note: This applies when Flash is read through WISHBONE.

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)
pp..pp: num_pages — Number of Flash pages to read when num_pages = 1
Number of Flash pages to read +1 when 1 < num_pages ≤ 12
Set to 0x3FFF when num_pages > 12

2. Data Size: (num_pages × 16) bytes when num_pages=1
(num_pages) × (16 + 4) bytes when num_pages>1

Note: When reading more than 12 pages, the num_pages argument is intentionally oversized. It is not necessary to read the extra pages. Read Flash may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. Flash page is read from the address pointed by the Address Register. The Address Register is auto incremented after each page read.

3. Examples: 0x73 00 00 01 0-byte dummy followed by one-page Flash data (16 bytes total).
0x73 00 00 04 Read 1 dummy page, followed by four dummy bytes, followed by three sets [one page Flash data, four dummy bytes] (48 data bytes, 32 dummy bytes).

Note: The maximum WISHBONE clock speed with which one page of data (num_page=1) can be read using WISHBONE and no wait states is 16.6 MHz. Faster WISHBONE clock speeds are supported by inserting WB wait states to observe the retrieval delay timing requirement. For more information, refer to the Reading Flash Pages section of the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

6.5.26. Enable Configuration Interface (Transparent) (0x74)

This command is used to put the device in transparent mode, where you can access the flash sectors of the device.

Table 6.52. Enable Configuration Interface (Transparent) (0x74)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	74	08 00 00 or 08 00	—	—	—

Notes: The I2C interface uses only two operands; all other interfaces use three operands. This command is required to enable modification of the Flash sectors of the device (CFG, UFM, Feature, and Security sectors). Terminate this command with command 0x26 (ISC_DISABLE) followed by command 0xFF (ISC_NOOP).

Exercising this command temporarily disables certain features of the device, notably GSR, user SPI port, primary user I2C port and Power Controller. These features are restored when the command is terminated.

Poll the BUSY bit (or wait 5us) after issuing this command for the Flash pumps to fully charge.

Example: 0x74 08 00 00 Enable Flash interface for change access through a non- I2C interface.

6.5.27. Refresh (0x79)

This command is used to launch the booting sequence. It is equivalent to toggling the PROGRAMN pin.

Table 6.53. Refresh (0x79)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	79	00 00	—	—	—

Example: 0x79 00 00 Issue Refresh command.

Notes: The Refresh command launches the Boot sequence.

Must have only two operands.

After completing the Refresh command (for example, SPI SN de-assertion or I2C stop), further bus accesses are prohibited for the duration of tREFRESH. Violating this requirement causes the Refresh process to abort and leave the MachXO4 RoT device in an unprogrammed state.

Occasionally, following a device REFRESH or PROGRAMN pin toggle, the secondary I2C port may be left in an undefined (non-idle) state. The likelihood of this condition is design and route dependent. To positively return the Secondary I2C port to the idle state, write a value of 0x44 to register I2C_2_CMDR through WISHBONE immediately after device reset is released. This causes a short low pulse on SCK as the hard-block signals a STOP on the bus then returns to the idle state. Failure to manually return the Secondary I2C port to the idle state may result in an I2C bus lock-up condition. Normal I2C activity can be commenced without additional delay.

6.5.28. Bitstream_Check (0x7D)

This command is used to enable various bitstream dryrun checks as well as enable basic device control actions such as standby, wakeup, reset, and others. The Bitstream Check (also known as Dryrun) command is a valuable feature of the MachXO4 RoT device. The dryrun performs all the necessary steps to check bitstream authentication without loading the bitstream into the device. This feature is used to check the authenticity of bitstreams in background while an active design is still running on the MachXO4 RoT device.

Table 6.54. Bitstream_Check (0x7D)

EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	7D	0ABB_BCDE 00	—	—	—

Data Format: A bit 6: Reset the internal configuration logic (default=0)

BBB[1] bits[5:3]: Bitstream Checking options

000: No Action

001: One time check of Soft Error Detect (SED)

010: Bitstream dry-run for CFG0 sector

011: Bitstream dry-run for CFG1 sector

100: Bitstream dry-run for External Flash from Primary booting address

101: Bitstream dry-run for External Flash from Secondary booting address

110: Bitstream dry-run for External Flash with user specified booting address[1]

111: Dry-run the bitstream after bitstream downloading is complete from Target configuration port.

C bit 2: Triggers the power controller to wakeup device from standby mode(default=0)

D bit 1: Triggers the power controller to put the device in standby mode (default=0)

E bit 0: Enable Global Set/Reset (default = 0)

Note:

1. Use the LSC_PROG_DRYRUN_ADDR command to provide the starting address of bitstream. Poll the BUSY bit using Read Status or Check Busy commands to determine bitstream check operation completion status, or wait 200 us, prior to issuing subsequent commands.

6.5.29. Set Address (0xB4)

This command is used to set the address register in a particular flash sector.

Table 6.55. Set Address (0xB4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	B4	00 00 00	W	4B	0000 0000 0000 00ss ssaa aaaa aaaa aaaa

Data Format: ssss: Select Flash Sector

0000: CFG0

0001: UFM0

0011: Feature

0100: CFG1

0101: UFM1

0110: Public Key

1000: UFM2

1001: UFM3

1010: AES Key

aa..aa: address 14-bit page address

Example: 0xB4 00 00 00 00 01 00 0A Set Address register to CFG1 sector, page 10 decimal.

6.5.30. Read USERCODE (0xC0)

Read the 32-bit usercode of bitstream.

Table 6.56. Read USERCODE (0xC0)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y/N	C0	00 00 00	R	4B	—

Example: 0xC0 00 00 00
ISC Enable = Y Read 4-byte USERCODE from CFG0 or CFG1 sector.
ISC Enable = N Read 4-byte USERCODE from SRAM.

6.5.31. Program USERCODE (0xC2)

Table 6.57. Program USERCODE (0xC2)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	C2	00 00 00	W	4B	—

Example: 0xC2 00 00 00 10 20 30 40 Sets USERCODE with 32-bit input 0x10 20 30 40.

Note: Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.32. Read USERCODE_DRYRUN (0xC1)

Read the 32-bit DRYRUN usercode. The DRYRUN usercode is read from the CFG Flash sector.

Table 6.58. Read USERCODE_DRYRUN (0xC1)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	C1	00 00 00	R	4B	—

Example: 0xC1 00 00 00 Read 4-byte DRYRUN usercode from the CFG Flash sector.

6.5.33. Program LSC_PROG_DRYRUN_ADDR (0xFC)

This command sets the starting address for the bitstream dryrun when using external flash. Send this command before enabling the dry-run. The most significant 24-bit indicate the starting address of the external flash device and the least significant bits are reserved. For external flash device with 32-bit address, the 24-bit address is padded with 8-bit 0s for the least significant bits.

Table 6.59. Program LSC_PROG_DRYRUN_ADDR (0xFC)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	FC	00 00 00	W	4B	AA BB CC 00

Example: 0xFC 00 00 00 20 45 77 00 Set the starting address to 20 45 77 in the external Flash.

Note: Use this command along with Bitstream_check command. Refer to the Bitstream_Check (0x7D)
section for details.

6.5.34. Enable Configuration Interface (Offline) (0xC6)

This command is used to enable offline configuration Interface.

Table 6.60. Enable Configuration Interface (Offline) (0xC6)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	C6	0y 00 00	—	—	—

Operand: 08 00 00 Enable Flash Normal mode. Normal edit mode for Offline configuration. Used for all offline commands described in this document, including Erase SRAM.

	00 00 00	Enable SRAM mode. Optional edit mode. Supports Erase SRAM command only.
Example:	0xC6 08 00 00	Enable Flash interface for offline change access.

Notes: Use this command to enable offline modification of the Flash, or non-volatile registers (NVR). SRAM must be erased before exiting Offline mode. When exiting Offline mode, follow the command 0x26 with the command 0xFF. Exercising this command tristate all user I/O (except persisted sysCONFIG ports). User logic ceases to function. UFMs remains accessible.

Poll the BUSY bit (or wait 5 µs) after issuing this command for the Flash pumps to fully charge.

6.5.35. Program UFM (0xC9)

This command is used to program UFM with one page of data.

Table 6.61. Program UFM (0xC9)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	C9	00 00 01	W	16B	16 bytes of write data

Example: 0xC9 00 00 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Write one page of data, pointed to by Address Register.

Notes: 16 data bytes must be written following the command and operand bytes to ensure proper data alignment. The Address Register is auto incremented following the page write.

If necessary, use 0x0E or 0xCB to erase UFM sector prior to executing this command.

Poll the BUSY bit (or wait 200 µs) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

6.5.36. Read UFM (0xCA) (SPI – Option 1)

This command is used to read UFM using SPI.

Table 6.62. Read UFM (0xCA) (SPI)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0001 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages — Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when num_pages > 1

2. Data Size (num_pages × 16) bytes

Notes: Read UFM may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. Flash page is read from the address pointed by the Address Register. The Address Register is auto-incremented after each page read.

3. Examples:

0xCA 10 00 01	Read 0-byte dummy followed by one page UFM data (16 bytes total).
0xCA 10 00 04	Read one-page dummy followed by three pages UFM data (four pages total).

6.5.37. Read UFM (0xCA) (I2C/SPI – Option 2)

This command is used to read UFM using I2C or SPI.

Table 6.63. Read UFM (0xCA) (I2C/SPI)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when num_pages > 1

2. Data Size: (num_pages × 16) bytes when num_pages=1

(num_pages × (16 + 4)) bytes when num_pages>1

Notes: Read UFM may be aborted at any time. Any data remaining in the read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. Flash page is read from the address pointed by the Address Register. The Address Register is auto-incremented after each page read.

3. Examples: 0xCA 00 00 01 Read 0-byte dummy followed by one page UFM data (16 bytes total).

0xCA 00 00 04 Read one-page dummy followed by four dummy bytes, followed by three sets [one page UFM data, followed by four dummy bytes] (48 data bytes, 32 dummy bytes).

6.5.38. Read UFM (0xCA) (WISHBONE)

This command is used to read UFM using WISHBONE.

Table 6.64. Read UFM (0xCA) (WISHBONE)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	CA	Refer below[1]	R	Refer below[2]	Refer below[3]

1. Operand: 0000 0000 00pp pppp pppp pppp (binary)

Where: pp..pp: num_pages Number of UFM pages to read when num_pages = 1

Number of UFM pages to read +1 when 1 < num_pages ≤ 12

Set to 0x3FFF when num_pages > 12

2. Data Size: (num_pages × 16) bytes when num_pages=1

(num_pages × (16 + 4)) bytes when num_pages>1

Notes: When reading more than 12 pages, the num_pages argument is intentionally oversized. It is not necessary to read the extra pages. Read UFM may be aborted at any time. Any data remaining in read FIFO is discarded. Any read data beyond the prescribed read size is indeterminate. UFM page is read from the address pointed by the Address Register. The Address Register is autoincremented after each page read.

3. Examples: 0xCA 00 00 01 Read 0 bytes dummy followed by one page UFM data (16 bytes total).

0xCA 00 00 04 Read one dummy page followed four dummy bytes, followed by three sets [one page UFM data, followed by four bytes dummy] (48 data bytes, 32 dummy bytes).

Notes: The maximum WISHBONE clock speed with which one page of data (num_page=1) can be read using WISHBONE and no wait states is 16.6 MHz. Faster WISHBONE clock speeds are supported by inserting WB wait states to observe the retrieval delay timing requirement. For more information, refer to the Reading Flash Pages section of the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

6.5.39. Erase UFM (0xCB)

This command is used to erase the different UFM sectors.

Table 6.65. Erase UFM (0xCB)

ISC Enable	CMD (Hex)	Operands (Binary)	Data Mode	Data Size	Data Format
Y	CB	Refer below	—	—	—

Operand: 000 0000 00ab cd00 0000 0000 (binary)

Where:
- a: Erase UFM3 sector
  - 0: No action
  - 1: Erase
- b: Erase UFM2 sector
  - 0: No action
  - 1: Erase
- c: Erase UFM1 sector
  - 0: No action
  - 1: Erase
- d: Erase UFM0 sector
  - 0: No action
  - 1: Erase

Notes: Erased condition for UFM bits = 0.

Poll the BUSY bit (or wait, Refer to Table 10.1) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.

Example: 0xCB 00 3C 00 Erase all four UFM sectors.

6.5.40. Read Device ID Code (0xE0)

Table 6.66. Read Device ID Code (0xE0)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	E0	00 00 00	R	4B	Refer to Table 6.67

Example: 0xE0 00 00 00 Read 4-byte device ID

Table 6.67. Device ID

Device Name	C Devices	E Devices
LFMXO4D-040	0x71 2E 20 43	0x71 2E 00 43
LFMXO4D-050	0xF1 2E 20 43	0xF1 2E 00 43
LFMXO4D-080	0xF1 2E 30 43	0xF1 2E 10 43
LFMXO4D-110	0x31 2E 30 43	0x31 2E 10 43

6.5.41. Verify Device ID Code (0xE2)

This command is used to verify the MachXO4 RoT device ID. Sets ID Error bit 27 in Status Register (SR) if mismatched.

Table 6.68. Verify Device ID Code (0xE2)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
Y	E2	00 00 00	W	4B	Refer to Table 6.67

Example: 0xE2 00 00 00 21 2E 20 43 Verify device ID with 32-bit input.

6.5.42. Program Feature (0xE4)

This command is used to program the feature sector which includes information about Control Register 1, I2C Target address, TraceID, and custom ID code.

Table 6.69. Program Feature (0xE4)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
Y	E4	00 00 00	—	16B	00 00 00 00 rr rr rr rr 00 00 ss uu cc cc cc cc

Data Format: rr rr rr rr: 32 bits of Control Register 1. For more information on Control Register 1, refer to the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

ss: 8 bits field for the user programmable bits of the I2C Target Address

uu: 8 bits for the user programmable TraceID

cc cc cc cc: 32 bits of Custom ID code

Note: It is not recommended to reprogram the Feature Row in system as it should be programmed ideally once during manufacturing.

Example: 0xE4 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 12 34

Program Feature Row with User I2C address field is set to 0x01, default user TraceID string, Custom ID code of 00 00 12 34.

6.5.43. Read Feature Row (0xE7)

This command is used to read the feature sector which includes information about Control Register 1, I2C Target address, TraceID, and custom ID code.

Table 6.70. Read Feature Row (0xE7)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Hex)
Y	E7	00 00 00	R	16B	00 00 00 00 rr rr rr rr 00 00 ss uu cc cc cc cc

Data Format: rr rr rr rr: 32 bits of Control Register 1. For more information on Control Register 1, refer to the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

ss: 8 bits for the user programmable I2C Target Address

uu: 8 bits for the user programmable TraceID

cc cc cc cc: 32 bits of Custom ID code

Example: 0xE7 00 00 00 Reads the Feature Row.

6.5.44. Check Busy Flag (0xF0)

This command is used to check if the configuration engine is busy.

Table 6.71. Check Busy Flag (0xF0)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
N	F0	00 00 00	R	1B	Bxxx xxxx

Data Format: B: bit 7: Busy Flag (1=busy)

(all other bits reserved)

Example: 0xF0 00 00 00

Read one byte, for example, 0x80 (busy flag set).

6.5.45. Program FEABITs (0xF8)

This command is used to program the feature bits such as booting sequence selection, password settings, and others.

Table 6.72. Program FEABITs (0xF8)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	F8	00 00 00	W	4B	00 00 00 00 00 00 00 0e cb bb mi sj dn pa wv ug

Data Format: e: Version Rollback Protection[1]

0: Disabled (Default)

1: Enabled (Checks if current version of bitstream is similar to the one that is going to be downloaded)

c: I2C deglitch filter range selection on primary I2C port[2]

0: 8 to 25 ns range (Default)

1: 16 to 50 ns range

bbb: Boot Sequence selection (used along with Controller SPI Port Persistence bit)

bbb	m	Boot Mode	Boot From
000	0	Dual	CFG0 - CFG1
000	1	Dual	CFG0 – Ext
X01	1	Single	Ext
010	1	Dual	Ext – CFG0
X11	1	Dual	Ext – Ext
011	0	Single	CFG0
001	0	Dual	CFG1 – CFG0
X10	0	Dual	No Boot
100	0	Single	CFG1
100	1	Dual	CFG1 – Ext
110	1	Dual	Ext - CFG1
111	0	Dual	Boot from latter bitstream first
101	0	Dual	Boot from former bitstream first

m: Controller SPI Port Persistence

0=Disabled (Default), 1=Enabled

i: I2C Port Persistence

0=Enabled (Default), 1=Disabled

s: Target SPI Port Persistence

0=Enabled (Default), 1=Disabled

j: JTAG Port Persistence

0=Enabled (Default), 1=Disabled

d: DONE Persistence

0=Disabled (Default), 1=Enabled

n: INITN Persistence

0=Disabled (Default), 1=Enabled

p: PROGRAMN Persistence
0=Enabled (Default), 1=Disabled

a: MY_ASSP Enabled
0=Disabled (Default), 1=Enabled

w	v	u	Flash Protection Sector Selection
0	0	0	No protection to any Flash sector
0	0	1	All UFMs
0	1	0	Feature, Security Keys
0	1	1	Feature, Security Keys, and all UFMs
1	0	0	CFG0 and CFG1
1	0	1	CFG0, CFG1, and all UFMs
1	1	0	Feature, Security Keys, CFG0, and CFG1
1	1	1	Feature, Security Keys, CFG0, CFG1, and all UFMs

g: I2C deglitch filter enable for Primary I2C Port
0=Disabled (Default), 1=Enabled

It is not recommended to reprogram the FEABITs in system as it should be programmed ideally once during manufacturing.

Notes:

1. Use this feature with the dry-run feature.
2. For this feature to take into effect, the I2C deglitch (g) filter must be enabled.

Example: 0xF8 00 00 00 0D 20 Programs the FEABITs.

6.5.46. Read FEABITs (0xFB)

This command is used to read the feature bits such as booting sequence selection, password settings, and others.

Table 6.73. Read FEABITs (0xFB)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format (Binary)
Y	FB	00 00 00	R	4B	xx xx xx xx xx xx xx xe cb bb mi sj dn pa wv ug

Data Format: e: Version Rollback Protection
0: Disabled (Default)
1: Enabled (Checks if current version of bitstream is similar to the one that is going to be downloaded)

c: I2C deglitch filter range selection on primary I2C port
0: Disabled (Default)
1: Enabled

bbb: Boot Sequence selection (used along with Controller SPI Port Persistence bit)

bbb	m	Boot Mode	Boot From
000	0	Dual	CFG0 - CFG1
000	1	Dual	CFG0 – Ext
X01	1	Single	Ext
010	1	Dual	Ext – CFG0
X11	1	Dual	Ext – Ext

011	0	Single	CFG0
001	0	Dual	CFG1 – CFG0
X10	0	Dual	No Boot
100	0	Single	CFG1
100	1	Dual	CFG1 – Ext
110	1	Dual	Ext - CFG1
111	0	Dual	Boot from latter bitstream first
101	0	Dual	Boot from former bitstream first

m: Controller SPI Port Persistence
0=Disabled (Default), 1=Enabled

i: I2C Port Persistence
0=Enabled (Default), 1=Disabled

s: Target SPI Port Persistence
0=Enabled (Default), 1=Disabled

j: JTAG Port Persistence
0=Enabled (Default), 1=Disabled

d: DONE Persistence
0=Disabled (Default), 1=Enabled

n: INITN Persistence
0=Disabled (Default), 1=Enabled

p: PROGRAMN Persistence
0=Enabled (Default), 1=Disabled

a: MY_ASSP Enabled
0=Disabled (Default), 1=Enabled

wvu:

w	v	u	Flash Protection Sector
0	0	0	No protection to any Flash sector
0	0	1	All UFMs
0	1	0	Feature, Security Keys
0	1	1	Feature, Security Keys, and all UFMs
1	0	0	CFG0 and CFG1
1	0	1	CFG0, CFG1, all UFMs, and USEC
1	1	0	Feature, Security Keys, CFG0, and CFG1
1	1	1	Feature, Security Keys, CFG0, CFG1, and all UFMs

g: I2C deglitch filter enable for Primary I2C Port
0=Disabled (Default), 1=Enabled

Example: 0xFB 00 00 00 Reads the FEABITs.

6.5.47. Bypass (Null Operation) (0xFF)

This command is no operation command (NOOP) or null operation.

Table 6.74. Bypass (Null Operation) (0xFF)

ISC Enable	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
N	FF	FF FF FF	—	—	—

Note: Operands are optional.

Example: 0xFF FF FF FF Bypass

7. Interface to Flash

The WISHBONE interface of the EFB module allows a WISHBONE host to access the configuration resources of the MachXO4 devices. This can be particularly useful for reading data from configuration resources such as USERCODE and TraceID. Most importantly, this feature allows you to update the Flash array of the devices while the device is in operation mode. This is a self-configuration operation. Upon power-up or a configuration refresh operation, the new content of the Flash is loaded into the Configuration SRAM and the device continues operation with a new configuration.

The data transfer and execution of operations is the same as the one documented in the UFM section of this document. This is because the UFM is also a Flash resource and the communication between the WISHBONE host and the configuration logic is performed through the same command, status, and data registers. Refer to the MachXO4 (LFMXO4) Flash Access, MachXO4 RoT Flash Access, and Command Framing sections for more information on these registers.

Figure 7.1 shows a basic flow diagram for implementing a Flash Update initiated through any of the sysCONFIG ports (I2C, SPI, or WISHBONE).

For detailed information on the MachXO4 device programming and configuration, refer to the MachXO4 Programming and Configuration User Guide (FPGA-TN-02393).

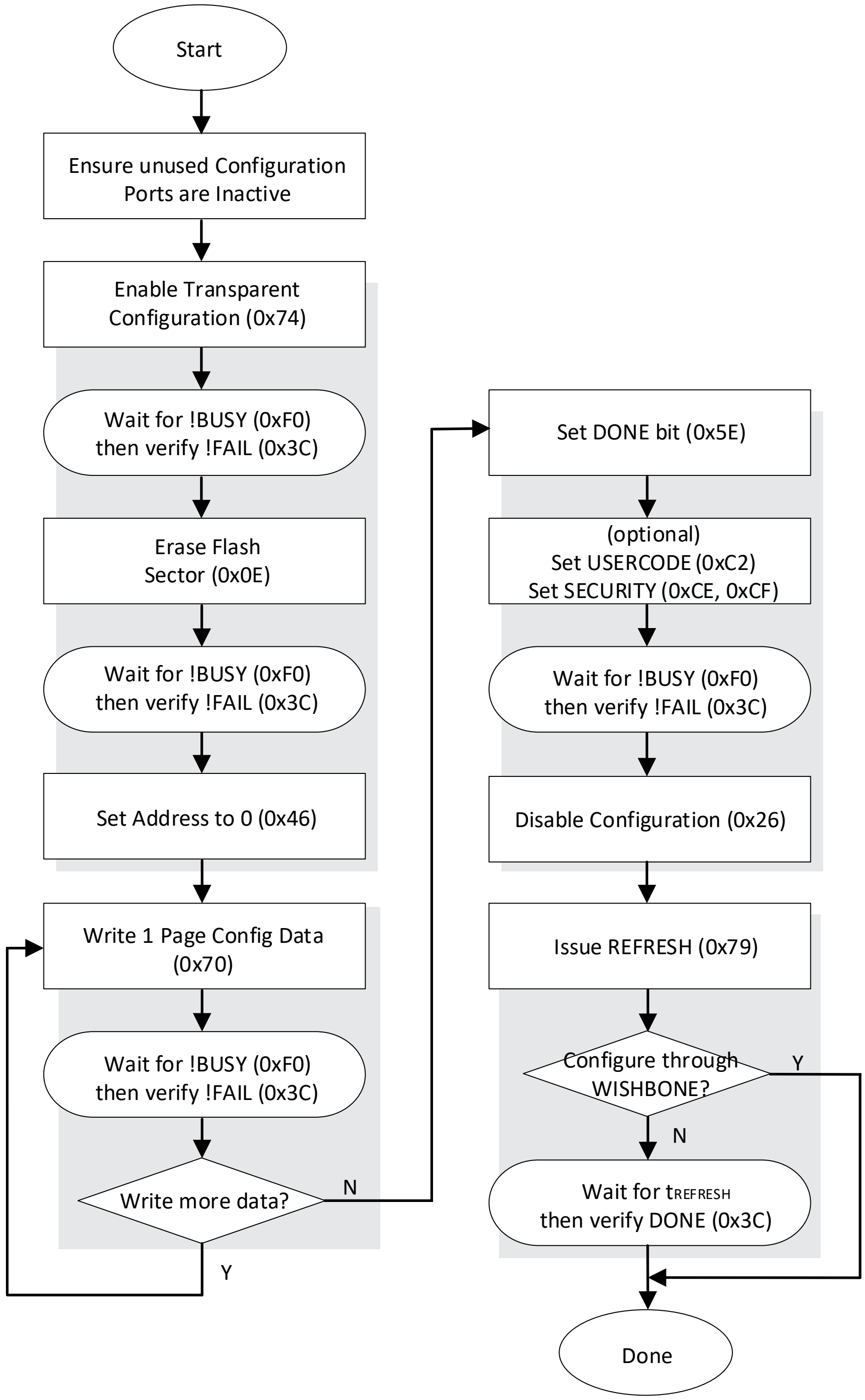


Figure 7.1. Basic Configuration UFM Program Example

8. Command Framing

8.1. I2C Framing

Each command string sent to the I2C EFB port must be correctly framed using the protocol defined for each interface. In the case of I2C, the protocol is well known and defined by the industry as shown below.

Table 8.1. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
I2C	Start	(Command/Operands/Data)	Stop

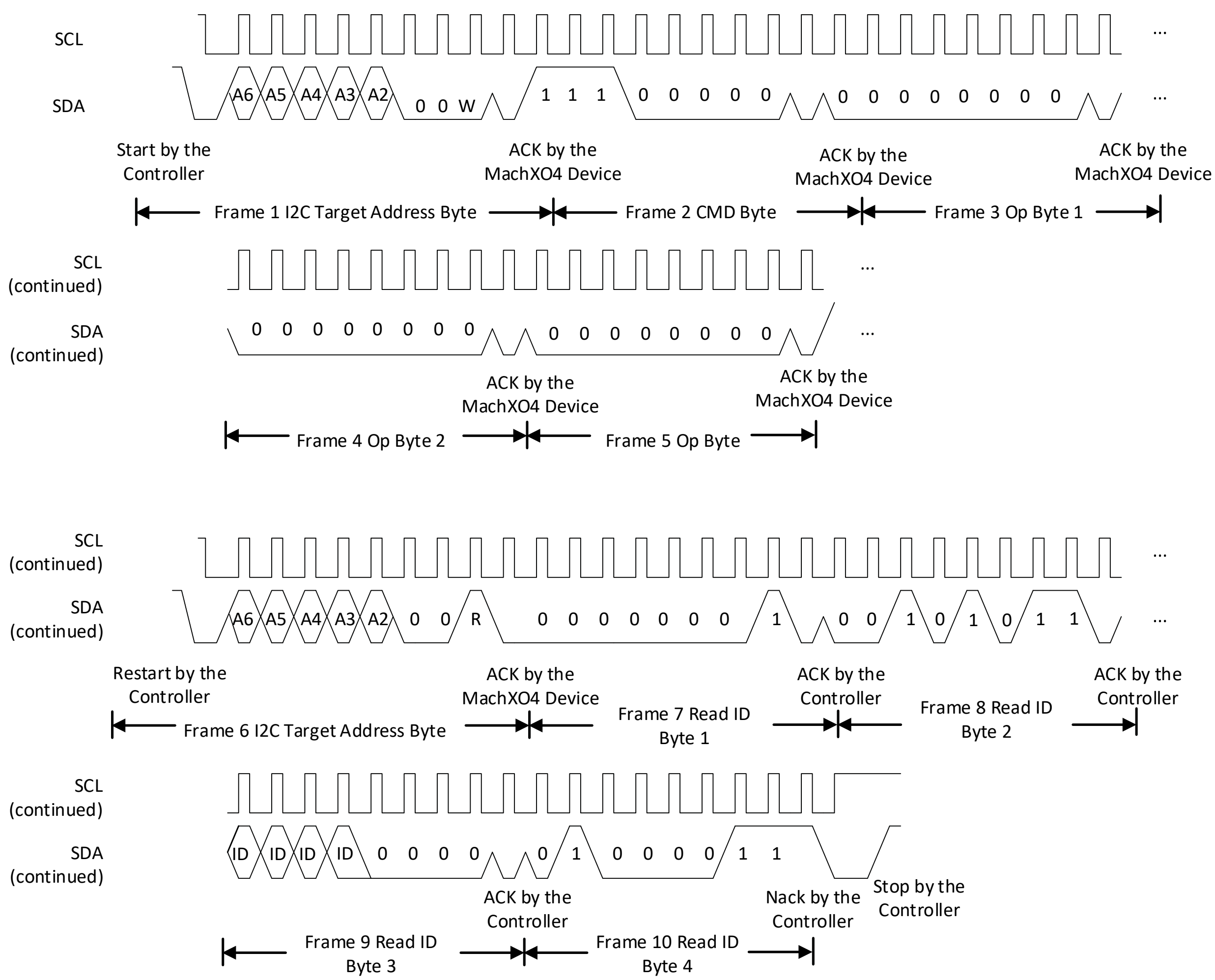


Figure 8.1. I2C Read Device ID Example

8.2. SPI Framing

Each command string sent to the SPI EFB port must be correctly framed using the protocol defined for each interface. In the case of SSPI, the protocol is well known and defined by the industry as shown below.

Table 8.2. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
SPI	Assert CS	(Command/Operands/Data)	De-assert CS

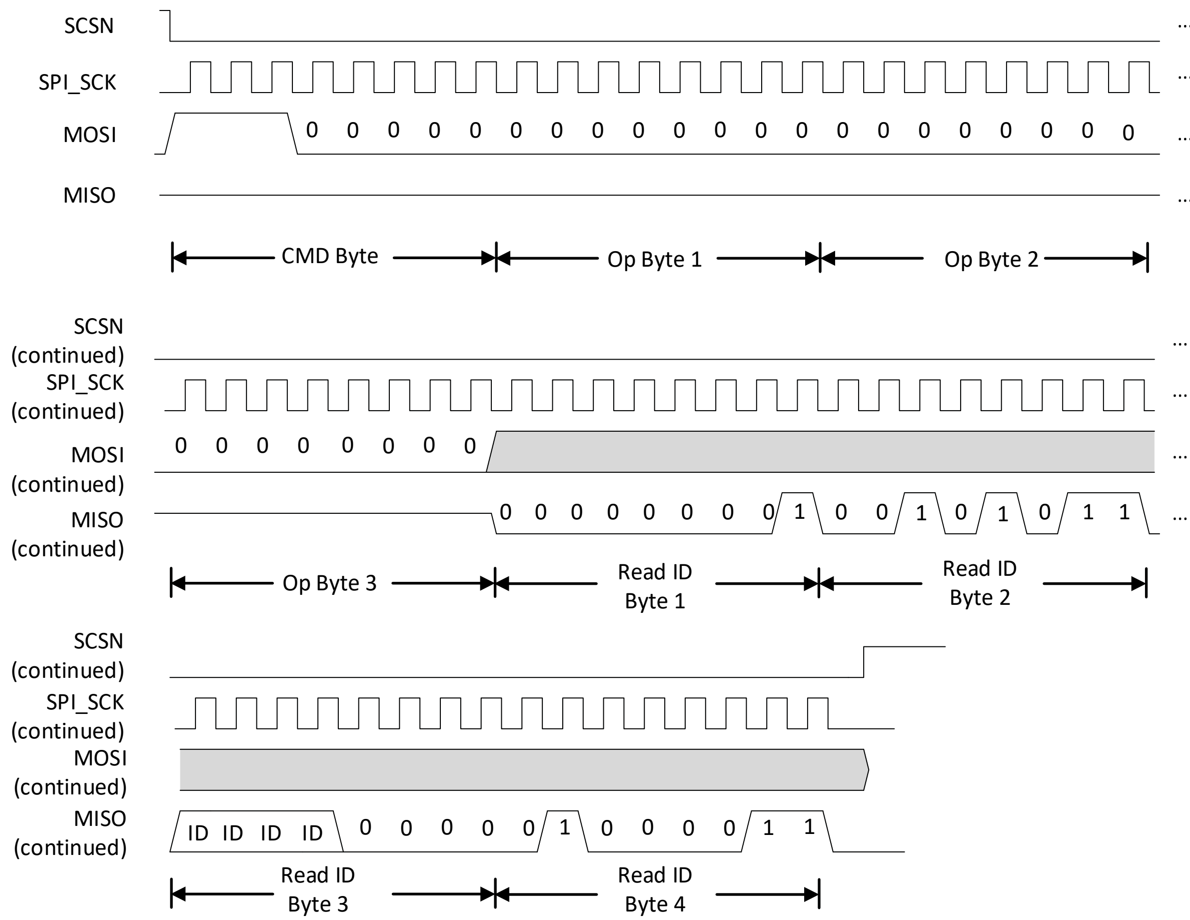


Figure 8.2. SSPI Read Device ID Example

8.3. WISHBONE Framing

To access the Flash, each command string sent to the WISHBONE EFB ports must be correctly framed using the protocol defined for each interface. In the case of the internal WISHBONE port, each command string is preceded by setting CFGCR[WBCE]. Similarly, each command string is followed by clearing the CFGCR[WBCE] bit.

Table 8.3. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
WISHBONE	Assert CFGCR[WBCE]	(Command/Operands/Data)	De-assert CFGCR[WBCE]

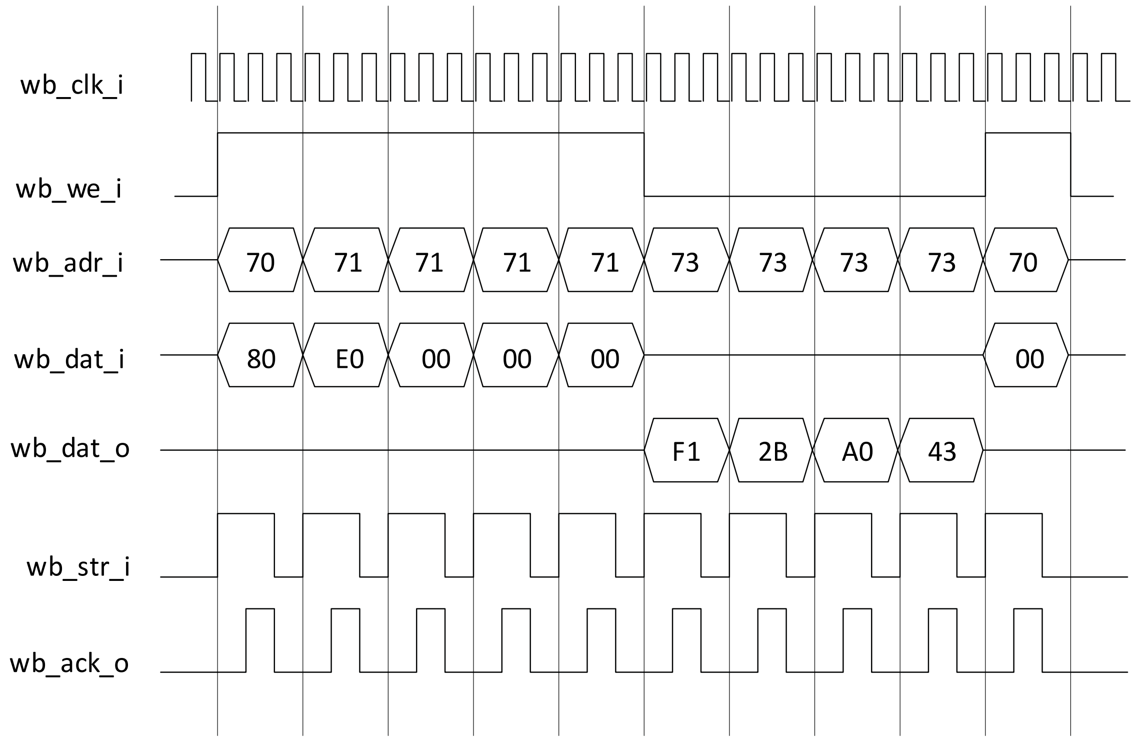


Figure 8.3. WISHBONE Read Device ID Example (-015 HC Device)

9. UFM Write and Read Examples

The Flash sectors support page-oriented read and write operations, while erase operations are sector-based. Consistent with many Flash devices, byte-oriented operations are not supported.

Table 9.1. Write Two UFM Pages[1, 2]

Instruction Number	R/W1	CMD2	Operand	Data	Comment
—	—	—	—	—	Open frame
1	W	74	08 00 00	—	Enable Configuration Interface
—	—	–	—	—	Close frame
—	—	+	—	—	—
2	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	–	—	—	(Repeat until Busy Flag not set, or wait 5 µs if not polling.)
—	—	+	—	—	—
3	W	47	00 04 00	—	Init UFM Address to 0000
—	—	–	—	—	—
—	—	+	—	—	—
4	W	C9	00 00 01	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Write UFM Page 0 Data
—	—	–	—	—	—
—	—	+	—	—	—
5	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	–	—	—	(Repeat until Busy Flag not set, or wait 200 µs if not polling.)
—	—	+	—	—	—
6	W	C9	00 00 01	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	Write UFM Page 1 Data **Note:** Address automatically incremented
—	—	–	—	—	—
—	—	+	—	—	—
7	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	–	—	—	(Poll until Busy Flag clear, or wait 200 µs if not polling.)
—	—	+	—	—	—
8	W	26	00 00	—	Disable Configuration Interface
—	—	–	—	—	—
—	—	+	—	—	—
9	W	FF	—	—	Bypass (NOP)
—	—	–	—	—	—

Notes:

1. When accessing Flash through WISHBONE, use CFGTXDR (0x71) to write data and CFDRXDR (0x73) to read data.
2. + and – refer to the command framing protocol appropriate for the interface, discussed in the Command Framing section.

Table 9.2. Read One UFM Page (All Devices, WISHBONE/SPI)[1, 2]

Instruction Number	R/W1	CMD2	Operand	Data	Comment
—	—	+	—	—	Open frame
1	W	74	08 00 00	—	Enable Configuration Interface
—	—	–	—	—	Close frame
—	—	+	—	—	—
2	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	—	—	—	(Repeat until Busy Flag not set, or wait 5 µs if not polling.)
—	—	+	—	—	—
3	W	B4	00 00 00	00 00 40 01	Set UFM Address to 0001
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 01	—	Read one page UFM (page 1) data
—	R	—	—	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	—
—	—	—	—	—	—
—	—	+	—	—	—
5	W	26	00 00	—	Disable Configuration Interface
—	—	—	—	—	—
—	—	+	—	—	—
6	W	FF	—	—	Bypass (NOP)
—	—	—	—	—	—

Notes:

1. When accessing Flash through WISHBONE, use CFGTXDR (0x71) to write data and CFDRXDR (0x73) to read data.
2. + and – refer to the command framing protocol appropriate for the interface, discussed in the Command Framing section.

Table 9.3. Read Two UFM Pages (WISHBONE/SPI)[1, 2]

Instruction Number	R/W1	CMD2	Operand	Data	Comment
—	—	+	—	—	Open frame
1	W	74	08 00 00	—	Enable Configuration Interface
—	—	—	—	—	Close frame
—	—	+	—	—	—
2	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	—	—	—	(Repeat until Busy Flag not set, or wait 5 µs if not polling.)
—	—	+	—	—	—
3	W	47	00 00 00	—	Init UFM address to 0000
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 03	—	Read two pages of UFM data, after one page of dummy bytes.[3]
—	R	—	—	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	—
—	—	—	—	—	—
—	—	+	—	—	—
5	W	26	00 00	—	Disable Configuration Interface
—	—	—	—	—	—
—	—	+	—	—	—
6	W	FF	—	—	Bypass (NOP)
—	—	—	—	—	—

Notes:
1. When accessing Flash through WISHBONE, use CFGTXDR (0x71) to write data and CFDRXDR (0x73) to read data.
2. + and – refer to the command framing protocol appropriate for the interface.
3. num_pages count must include dummy page.

Table 9.4. Read Two UFM0 Pages (WISHBONE/SPI/I2C)[1, 2]

Instruction Number	R/W1	CMD2	Operand	Data	Comment
—	—	+	—	—	Open Frame
1	W	74	08 00 00	—	Enable Configuration Interface
—	—	—	—	—	Close Frame
—	—	+	—	—	—
2	W	3C	00 00 00	—	Poll Configuration Status Register
—	R	—	—	xx xx bx xx	—
—	—	—	—	—	Repeat until Busy Flag not set, or wait 5 µs if not polling.
—	—	+	—	—	—
3	W	47	00 00 00	—	Set address counter to UFM0, Page 0.
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	00 00 03	—	Read two pages of UFM0 data, after one page of dummy bytes. Four dummy bytes follow each page. [3]
—	R	—	—	xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F xx xx xx xx 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F xx xx xx xx	The presence of four dummy bytes after each page is determined by the operand.[4]
—	—	—	—	—	—
—	—	+	—	—	—
5	W	26	00 00	—	Disable Configuration Interface
—	—	—	—	—	—
—	—	+	—	—	—
6	W	FF	—	—	Bypass (NOP)
—	—	—	—	—	—

Notes:
1. When accessing Flash through WISHBONE, use CFGTXDR (0x71) to write data and CFDRXDR (0x73) to read data.
2. + and – refer to the command framing protocol appropriate for the interface.
3. num_pages count must include dummy page.
4. The 10 00 pp operand should not be used with I2C command.

10. Flash Performance

Table 10.1. Flash Performance in the MachXO4 (LFMXO4) Device[1]

		LFMXO4-010	LFMXO4-015	LFMXO4-015 256 Ball Package	LFMXO4-025	LFMXO4-050	LFMXO4-050 400 Ball Package	LFMXO4-080	LFMXO4-110
CFG Erase (tEraseCFG)	Typ. Min.	800	800	1100	1100	1800	2800	2800	4500
	Typ. Max.	1400	1400	1900	1900	3100	4800	4800	7700
CFG Program (tProgramCFG)	All	500	500	740	740	1400	2200	2200	3000
	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
UFM Erase (tEraseUFM)	Typ. Min.	400	400	500	500	600	900	900	1600
	Typ. Max.	700	700	900	900	1000	1600	1600	2800
UFM Program (tProgramUFM)	All	110	110	140	140	180	480	480	840
	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
tErase (max)	Note 2	12000	15000	15000	15000	30000	30000	30000	45000

Notes:

1. All times are averages, in ms. SRAM erase times are < 0.1 ms.
2. tErase (max) is recommended for algorithm-based timeouts.

Table 10.2. Flash Performance in MachXO4 RoT Device[1]

			LFMXO4-040, LFMXO4-050	LFMXO4-080, LFMXO4-110
CFG Erase (CFG0/CFG1)	tEraseCFG	Typ. Min.	1800	4500
		Typ. Max.	3100	7700
CFG Program (CFG0/CFG1)	tProgramCFG	All	1400	3000
		1 page	0.2	0.2
Big UFM Erase (UFM0/UFM1)	tEraseUFM_bg	Typ. Min.	600	1600
		Typ. Max.	1000	2800
Big UFM Program (UFM0/UFM1)	tProgramUFM_ bg	All	180	840
		1 page	0.2	0.2
Medium UFM Erase (UFM2)	tEraseUFM_md	Typ. Min.	800	800
		Typ. Max.	1400	1400
Medium UFM Program (UFM2)	tProgramUFM_md	All	500	500
		1 page	0.2	0.2
Small UFM Erase (UFM3)	tEraseUFM_sm	Typ. Min.	400	400
		Typ. Max.	700	700
Small UFM Program (UFM3)	tProgramUFM_sm	All	110	110
		1 page	0.2	0.2
tErase (max)	—	Note 2	30000	45000

Notes:

1. All times are averages, in ms. SRAM erase times are < 0.1 ms.
2. tErase (max) is recommended for algorithm-based timeouts.

11. Erase/Program/Verify Time Calculation Example

Using the data above, it is possible to roughly calculate the time required to perform a Program/Verify operation. The calculation assumes nearly 100% bus use. The overhead required by bus controller processes, if any, is not accounted for in the equation below.

E/P/V time (µs): tEraseProgramVerify = tErase + tProgram + tVerify

Where: tErase = tEraseCFG + tEraseUFM[1]

tProgram = 0.2 ms × number of Pages to program[2]

tVerify = (8 × number of Pages programmed) × BusEff × tBUSCLK

Table 11.1. E/P/V Calculation Parameters

	BusEff (Single Page Read)	BusEff[3] (Multi Page Read)	tBUSCLK
I2C	14	>12	2.5 µs min
SPI	12	> 8	0.015 µs min
WB	5.25	> 3	0.020 µs min

Notes:

1. Sector erase times are additive. If a sector (for example, CFG) is not erased, its erase time is 0.
2. Data transfer time is insignificant to tProgram for high-speed data protocols. To account for slow bus speeds (for example, I2C), multiply tVerify by two.
3. Bus efficiency approaches this value as the number of read pages increases.

References

- MachXO4 Family Data Sheet (FPGA-DS-02125)
- MachXO4 Family RoT Devices Data Sheet (FPGA-DS-02126)
- MachXO4 Hardened Control Functions User Guide (FPGA-TN-02403)
- Power and Thermal Estimation and Management for MachXO4 Devices (FPGA-TN-02409)
- MachXO4 Programming and Configuration User Guide (FPGA-TN-02393)
- MachXO4 sysCLOCK PLL Design User Guide (FPGA-TN-02391)
- MachXO4 web page
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs 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 1.3, September 2026

Section	Change Summary
All	Made editorial fixes.
Abbreviations in This Document	Added *AES*, *ECDSA*, *JTAG*, *LUT*, *NVR*, *PWM*, *RoT*, and *SSPI*.
Introduction	• Added a MachXO4 terminology note. • Added security key and security feature descriptions. • Updated Figure 1.1. Embedded Function Block (EFB).
Hardened SPI IP Core	Added Figure 3.8. Target SPI Configuration Timing.
Hardened Timer/Counter PWM	Added the Timer/Counter Modes of Operation section.
MachXO4 (LFMXO4) Flash Access	• Added *MachXO4 (LFMXO4)* to the title and content of this section. • Added a clarification that this section applies to MachXO4 (LFMXO4) flash access only. • Added the *LFMXO4-010ZC*, *LFMXO4-015ZC*, *LFMXO4-050ZC*, *LFMXO4-080ZC*, and *LFMXO4*-110ZC devices to Table 5.39. Device ID.
MachXO4 RoT Flash Access	Added this section.
Interface to Flash	Updated Figure 7.1. Basic Configuration UFM Program Example.
UFM Write and Read Examples	• Updated the operand for Instruction Number 3 in Table 9.1. Write Two UFM Pages. • Updated the data for Instruction Number 3 in Table 9.2. Read One UFM Page (All Devices, WISHBONE/SPI). • Added Table 9.4. Read Two UFM0 Pages (WISHBONE/SPI/I2C).
Flash Performance	• Added *(LFMXO4)* to the caption of Table 10.1. Flash Performance in the MachXO4 (LFMXO4) Device. • Added Table 10.2. Flash Performance in MachXO4 RoT Device.
References	Added *MachXO4 Family RoT Devices Data Sheet (FPGA-DS-02126)*.

Revision 1.2, February 2026

Section	Change Summary
Hardened I2C IP Cores	Added a description stating that the 8th SCL is the byte boundary for the *CKSDIS* description.
Hardened Timer/Counter PWM	Added a *Note* to Table 13.8. Timer/Counter Control 2.

Revision 1.1, December 2025

Section	Change Summary
SPI Registers	Updated the *DIVIDER* description for Table 8.5. SPI Clock Pre-scale.

Revision 1.0, December 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com