



Using Hardened Control Functions in MachXO3 Devices Reference Guide

Technical Note

FPGA-TN-02064-2.3

December 2025

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

Contents	3
Acronyms in This Document	7
1. Introduction	8
1.1. EFB Register Map	9
1.2. WISHBONE Bus Interface	9
1.3. WISHBONE Write Cycle	10
1.4. WISHBONE Read Cycle	11
1.5. WISHBONE Reset Cycle	13
2. Hardened I ² C IP Cores	14
2.1. I ² C Registers	14
3. Typical I ² C Transactions	22
4. I ² C Functional Waveforms	23
5. I ² C Timing Diagram	27
6. I ² C Simulation Model	28
7. Hardened SPI IP Core	34
8. SPI Registers	35
9. Typical SPI Transactions	44
10. SPI Functional Waveforms	45
11. SPI Timing Diagrams	46
12. SPI Simulation Model	49
13. Hardened Timer/Counter PWM	52
13.1. Timer/Counter Registers	52
14. Timer Counter Simulation Model	60
15. NVCM (MachXO3L)/Flash (MachXO3LF) Access	65
15.1. NVCM/Flash Access Ports	65
15.2. NVCM/Flash Access through WISHBONE Target Interface	66
15.3. Command and Data Transfers to NVCM/Flash Space	71
15.4. Command Summary by Application	71
15.5. Command Descriptions by Command Code	73
16. Interface to NVCM/Flash	88
17. Command Framing	90
17.1. I ² C Framing	90
17.2. SPI Framing	90
17.3. WISHBONE Framing	91
18. NVCM1/Flash Write and Read Examples	93
19. NVCM/Flash Performance	96
20. Erase/Program/Verify Time Calculation Example	97
References	98
Technical Support Assistance	99
Revision History	100

Figures

Figure 1.1. Embedded Function Block (EFB)	8
Figure 1.2. WISHBONE Bus Interface between the FPGA Core and the EFB Module	9
Figure 1.3. WISHBONE Bus Write Operation	11
Figure 1.4. WISHBONE Bus Read Operation	12
Figure 1.5. EFB WISHBONE Interface Reset	13
Figure 2.1. I ² C Controller Read/Write Example (via WISHBONE)	20
Figure 2.2. I ² C Target Read/Write Example (via WISHBONE)	21
Figure 3.1. Simple I ² C Command (for example, ISC_ERASE)	22
Figure 3.2. I ² C Command with Write Data (for example, LSC_PROG_INCR_NV)	22
Figure 3.3. I ² C Command with Read Data (for example, LSC_READ_STATUS)	22
Figure 4.1. EFB Controller – I ² C Write	23
Figure 4.2. EFB Controller – I ² C Read	24
Figure 4.3. EFB Target – I ² C Write	25
Figure 4.4. EFB Target – I ² C Read	26
Figure 5.1. I ² C Bit Transfer Timing	27
Figure 8.1. SPI Controller Read/Write Example (via WISHBONE) – Production Silicon	42
Figure 8.2. SPI Target Read/Write Example (via WISHBONE) – Production Silicon	43
Figure 9.1. Simple SPI Command (for example, ISC_ERASE)	44
Figure 9.2. SPI Command with Write Data (for example, LSC_PROG_INCR_NV)	44
Figure 9.3. SPI Command with Read Data (for example, LSC_READ_STATUS)	44
Figure 10.1. Fully Specified SPI Transaction (MachXO3L/LF as SPI Controller or Target)	45
Figure 10.2. Minimally Specified SPI Transaction Example (MachXO3L/LF as SPI Target)	45
Figure 11.1. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)	46
Figure 11.2. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)	46
Figure 11.3. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)	47
Figure 11.4. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)	47
Figure 11.5. Target SPI Dummy Byte Response (SPICR2[SDBRE]) Timing	48
Figure 15.1. Interfaces to the NVCM/Flash Sectors	65
Figure 16.1. Basic Configuration NVCM1/UFM Program Example	89
Figure 17.1. I ² C Read Device ID Example	90
Figure 17.2. SSPI Read Device ID Example	91
Figure 17.3. WISHBONE Read Device ID Example (-1200 HC Device)	92

Tables

Table 1.1. EFB Register Map	9
Table 1.2. WISHBONE Target Interface Signals of the EFB Module	10
Table 2.1. I ² C Registers	14
Table 2.2. I ² C Control (Primary/Secondary)	14
Table 2.3. I ² C Command (Pri/Sec)	15
Table 2.4. I ² C Clock Prescale 0 (Primary/Secondary)	16
Table 2.5. I ² C Clock Prescale 1 (Primary/Secondary)	16
Table 2.6. I ² C Transmit Data Register (Primary/Secondary)	16
Table 2.7. I ² C Status (Primary/Secondary)	16
Table 2.8. I ² C General Call Data Register (Primary/Secondary)	18
Table 2.9. I ² C Receive Data Register (Primary/Secondary)	18
Table 2.10. I ² C Interrupt Status (Primary/Secondary)	18
Table 2.11. I ² C Interrupt Enable (Primary/Secondary)	19
Table 6.1. I ² C Primary Simulation Mode	28
Table 6.2. I ² C Secondary Simulation Mode	31
Table 8.1. SPI Registers	35
Table 8.2. SPI Control 0	35
Table 8.3. SPI Control 1	36
Table 8.4. SPI Control 2	37
Table 8.5. SPI Clock Pre-scale	38
Table 8.6. SPI Controller Chip Select	38
Table 8.7. SPI Transmit Data Register	38
Table 8.8. SPI Status	39
Table 8.9. SPI Receive Data Register	39
Table 8.10. SPI Interrupt Status	40
Table 8.11. SPI Interrupt Enable	40
Table 12.1. SPI Simulation Model	49
Table 13.1. Timer/Counter Registers	52
Table 13.2. Timer/Counter Control	52
Table 13.3. Timer/Counter Control 1	53
Table 13.4. Timer/Counter Set Top Counter Value 0	54
Table 13.5. Timer/Counter Set Top Counter Value 1	54
Table 13.6. Timer/Counter Set Compare Counter Value 0	55
Table 13.7. Timer/Counter Set Compare Counter Value 1	55
Table 13.8. Timer/Counter Control 2	55
Table 13.9. Timer/Counter Counter Value 0	56
Table 13.10. Timer/Counter Counter Value 1	56
Table 13.11. Timer/Counter Current Top Counter Value 0	56
Table 13.12. Timer/Counter Current Top Counter Value 1	56
Table 13.13. Timer/Counter Current Compare Counter Value 0	56
Table 13.14. Timer/Counter Current Compare Counter Value 1	57
Table 13.15. Timer/Counter Current Capture Counter Value 0	57
Table 13.16. Timer/Counter Current Capture Counter Value 1	57
Table 13.17. Timer/Counter Status Register	57
Table 13.18. Timer/Counter Interrupt Status	58
Table 13.19. Timer/Counter Interrupt Enable	58
Table 14.1. Timer/Counter Simulation Mode	60
Table 15.1. WISHBONE to NVCM/Flash Logic Registers	66
Table 15.2. NVCM/Flash Control	66
Table 15.3. NVCM/Flash Transmit Data	67
Table 15.4. NVCM/Flash Status	67
Table 15.5. NVCM/Flash Receive Data	68

Table 15.6. NVCM/Flash Interrupt Status	68
Table 15.7. NVCM/Flash Interrupt Enable	69
Table 15.8. Unused (Reserved) Register	70
Table 15.9. EFB Interrupt Source.....	70
Table 15.10. NVCM (Sector 1)/UFM Commands	71
Table 15.11. Configuration NVCM/Flash (Sector 0) Commands.....	72
Table 15.12. Non-Volatile Register (NVR) Commands	73
Table 15.13. Erase NVCM/Flash (0x0E).....	73
Table 15.14. Read TracelD Code (0x19).....	74
Table 15.15. Disable Configuration Interface (0x26).....	74
Table 15.16. Read Status Register (0x3C).....	74
Table 15.17. Reset CFG Address (0x46)	75
Table 15.18. Reset NVCM1 Address (0x47).....	75
Table 15.19. Program DONE (0x5E).....	75
Table 15.20. Program Configuration NVCM (0x70)	76
Table 15.21. Read Configuration NVCM/Flash (0x73) (SPI)	76
Table 15.22. Read Configuration NVCM/Flash (0x73) (I ² C/SPI).....	76
Table 15.23. Read Configuration NVCM/Flash (0x73) (WISHBONE).....	77
Table 15.24. Enable Configuration Interface (Transparent) (0x74)	78
Table 15.25. Refresh (0x79).....	78
Table 15.26. STANDBY (0x7D).....	79
Table 15.27. Set Address (0xB4).....	79
Table 15.28. Read USERCODE (0xC0)	80
Table 15.29. Program USERCODE (0xC2).....	80
Table 15.30. Enable Configuration Interface (Offline) (0xC6)	80
Table 15.31. Program NVCM1 (0xC9).....	80
Table 15.32. Read NVCM/UFM (0xCA) (SPI)	81
Table 15.33. Read NVCM/UFM (0xCA) (SPI/I ² C).....	81
Table 15.34. Read NVCM/UFM (0xCA) (WISHBONE).....	82
Table 15.35. Erase NVCM1 (0xCB)	82
Table 15.36. Program SECURITY (0xCE)	83
Table 15.37. Program SECURITY PLUS (0xCF).....	83
Table 15.38. Read Device ID Code (0xE0)	83
Table 15.39. Device ID	84
Table 15.40. Verify Device ID Code (0xE2)	84
Table 15.41. Program Feature (0xE4).....	84
Table 15.42. Read Feature Row (0xE7).....	85
Table 15.43. Check Busy Flag (0xF0).....	85
Table 15.44. Program FEABITs (0xF8).....	85
Table 15.45. Read FEABITs (0xFB).....	86
Table 15.46. Bypass (Null Operation) (0xFF)	87
Table 17.1. Command Framing Protocol, by Interface	90
Table 17.2. Command Framing Protocol, by Interface	90
Table 17.3. Command Framing Protocol, by Interface	91
Table 18.1. Write Two NVCM1/UFM Pages.....	93
Table 18.2. Read One NVCM1/UFM Page (All Devices, WISHBONE/SPI).....	94
Table 18.3. Read Two NVCM1/UFM Pages (WISHBONE/SPI)	95
Table 19.1. NVCM/Flash Performance in MachXO3L/LF Device ¹	96
Table 20.1. E/P/V Calculation Parameters	97

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
EFB	Embedded Function Block
FPGA	Field-Programmable Gate Array
I ² C	Inter-Integrated Circuit
MSB	Micro System Builder
NVCM	Non-Volatile Configuration Memory
PLD	Programmable Logic Device
PLL	Phase Locked Loop
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSM	Synchronous State Machine

1. Introduction

This reference guide supplements [Using Hardened Control Functions in MachXO3 Devices \(FPGA-TN-02063\)](#) which explains the software usage. In this document, you can find:

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

As an overview, the MachXO3™ 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 MachXO3L/LF devices include an EFB module. The EFB block includes the following control functions:

- Two I²C Cores
- One SPI Core
- One 16-bit Timer/Counter
- (MachXO3L) Interface to NVCM memory
- (MachXO3LF) Interface to Flash memory which includes:

User Flash Memory for MachXO3LF-640 and higher densities
Configuration logic

- 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.

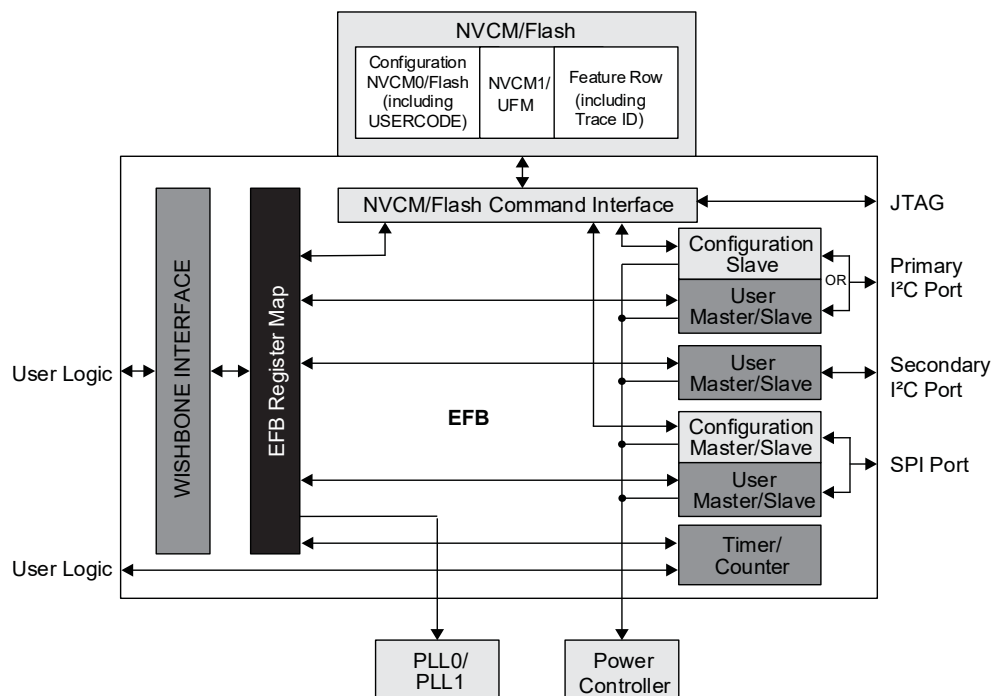


Figure 1.1. Embedded Function Block (EFB)

Notes:

- Only MachXO3L devices have NVCM.
- Only MachXO3LF devices have Flash.

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 a read/write operations. Each hardened function has dedicated 8-bit Data and Control registers, except for the NVCM/Flash. These are access through the same set of registers. [Table 1.1](#) documents the register map of the EFB module. The PLL registers are located in the NVCM/Flash in MachXO3L/LF devices PLL modules, but are access 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	I ² C Primary
0x4A-0x53	I ² C Secondary
0x54-0x5D	SPI
0x5E-0x6F	Timer/Counter
0x70-0x75	NVCM/Flash
0x76-0x77	EFB Interrupt Source

Notes:

1. There can be up to two PLLs in a MachXO3L/LF device. PLL0 has an address range from 0x00 to 0x1F. PLL1 (if present) has an address range from 0x20 to 0x3F. [MachXO3 sysCLOCK PLL Design and Usage Guide \(FPGA-TN-02058\)](#), for details on PLL configuration registers and recommended usage.
2. NVCM in MachXO3LF devices.
3. Flash in MachXO3LF devices.

Address spaces that are not defined in [Table 1.1](#) are invalid and results in non-deterministic results. It is the responsibility of the designer to ensure valid addresses are presented to the EFB WISHBONE target interface.

1.2. WISHBONE Bus Interface

The WISHBONE Bus in the MachXO3L/LF is compliant with the WISHBONE standard from OpenCores. It provides connectivity between FPGA user logic and the EFB functional blocks. The user 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.

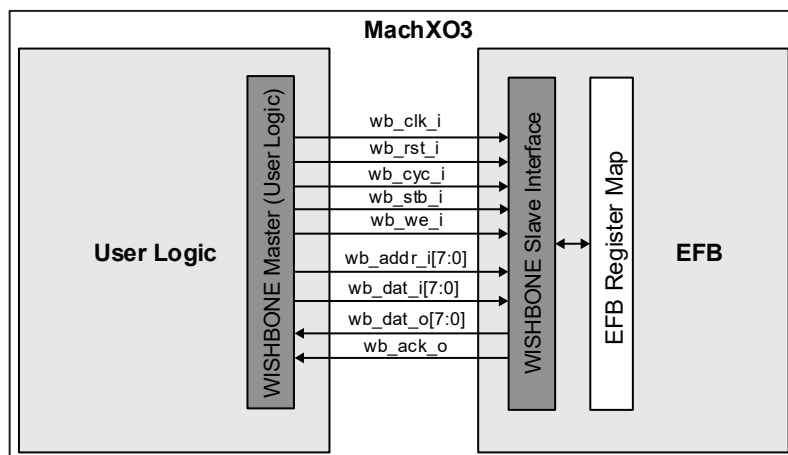


Figure 1.2. WISHBONE Bus Interface between the FPGA Core and the EFB Module

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 I ² C User Target or Configuration Target ports, the clock speed must be at least 7.5x the I ² C bus speed (for example, >3.0 MHz when I ² C rate = 400 kHz). When used in conjunction with SPI Target port, the Wishbone clock speed should be at least twice the
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 us after de-assertion before starting any subsequent WISHBONE
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
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 utilized along with the Mico System Builder (MSB) platform, which can implement multi-Controller bus configurations. If two Controller 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, the user cannot add these signals.
- 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 will simply respond by asserting wb_ack_o signal. It is the responsibility of the user to 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 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 will be 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.

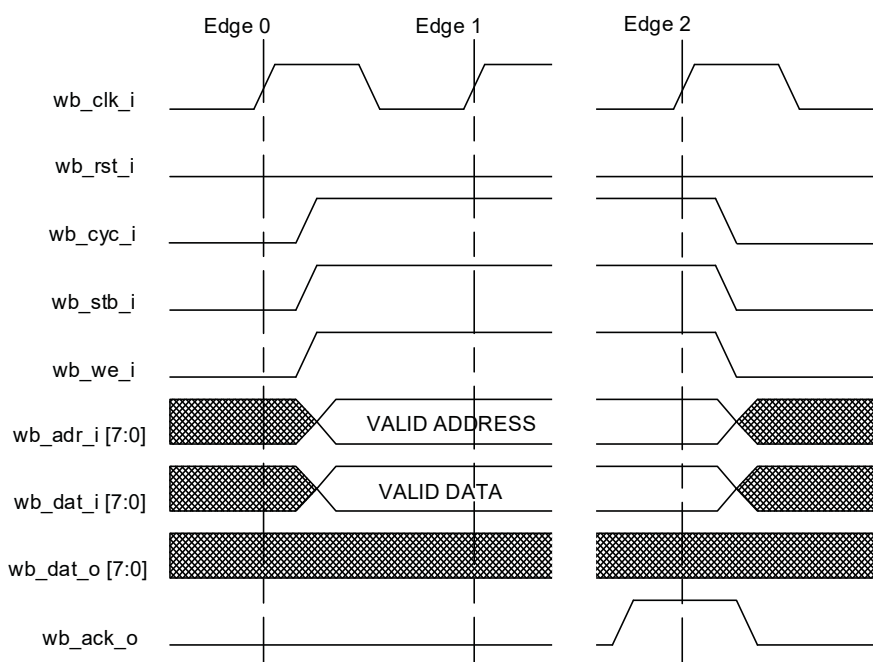


Figure 1.3. WISHBONE Bus Write Operation

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 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.

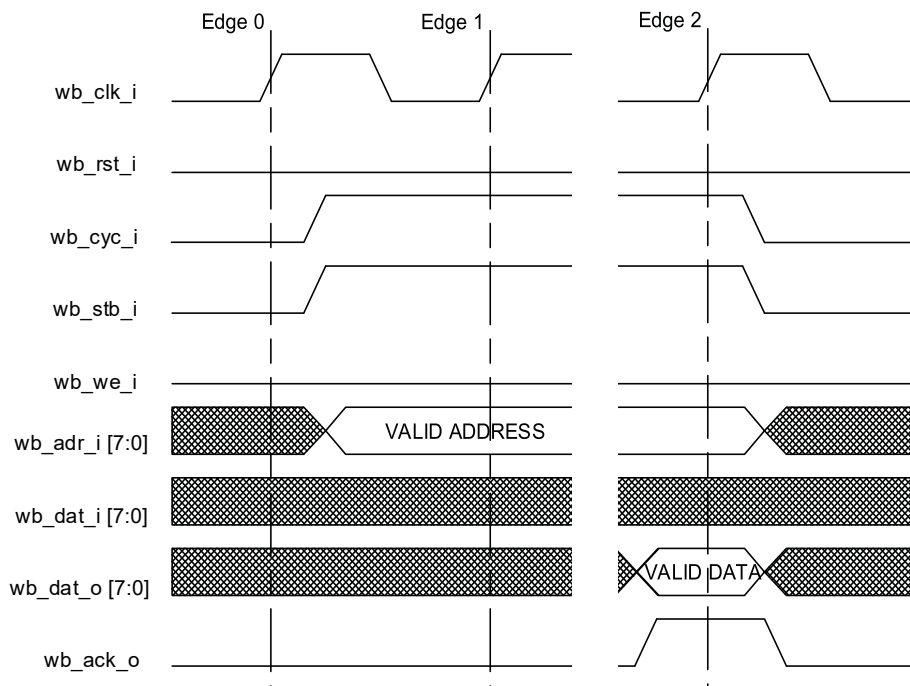


Figure 1.4. WISHBONE Bus Read Operation

To avoid simulation mismatch in functional simulations, add a delay of 100ps to `wb_cyc_i` and `wb_stb_i` assertion assignments. See the examples below. The examples assume the signal '`wb_cyc_i_gen`' 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 100ps;
```

Additionally, ensure your logic monitors for `wb_ack_o`, and deassert `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 will only reset the WISHBONE interface logic. This signal will not affect the contents of any registers in the EFB register map. It will only affect ongoing bus transactions.

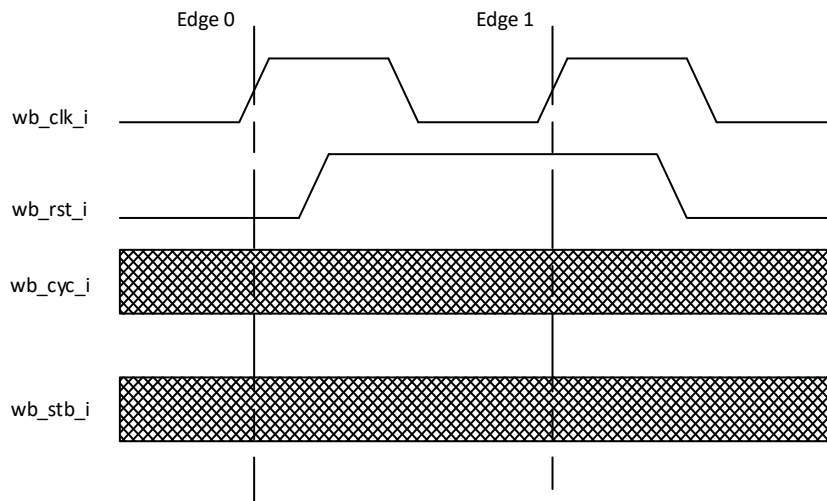


Figure 1.5. EFB WISHBONE Interface Reset

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 MachXO3L/LF 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 by designers to any general purpose I/O. In addition, the Primary I²C core can be used for accessing the NVCM/Flash. However, the Primary I²C port cannot be used for both NVCM/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 7.5x 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

I ² C Primary Register Name	I ² C Secondary Register Name	Register Function	Address I ² C Primary	Address I ² C 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 shall be written '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- 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 MachXO3L/LF Target Address is matched during standby/sleep mode.

- 0: Disable
- 1: Enable

SDA_DEL_SEL[1:0]

SDA Output Delay (Tdel) Selection (see [Figure 5.1](#)).

- 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. Bit CKSDIS disables the clock stretching if desired by the user. In this case, the overflow flag must be monitored. For Controller operations, set this bit to '0'. Clock stretching is used by the MachXO2 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 + 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. See 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. See discussion below.

I2C_PRESCALE[9:0]

I²C Clock Prescale value. A write operation to I2C_BR [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:

- Different from transmitting a Controller, the practical limit for Target I²C bus speed support 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 IPexpress™ 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, both entered by the user. The digital value of the divider is programmed in the MachXO3L/LF 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. Designers 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 could be high after configuration wake-up and before the first valid I²C transfer start (when BUSY is low), and it is not indicating byte in transfer, but an invalid indicator.

- 1: Byte transfer in progress
- 0: Byte transfer complete

BUSY

I²C Bus busy. The I²C bus is involved in transaction. This is set at START condition and cleared at STOP. Note only when this bit is set should all other I²C SR bits be treated as valid indicators for a valid transfer.

- 1: I²C bus busy
- 0: I²C bus not busy

RARC

Received Acknowledge. An acknowledge response was 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 4.4](#).

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 will be 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 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)				IRQARBLN	IRQTRRDYEN	IRQTROEEN	IRQHGCEN
Default	0	0	0	0	0	0	0	0
Access	—	—	—	—	R/W	R/W	R/W	R/W

IRQARBLN

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 via the WISHBONE interface. The following sequence is for the Primary I²C but the same sequence applies to the Secondary I²C.

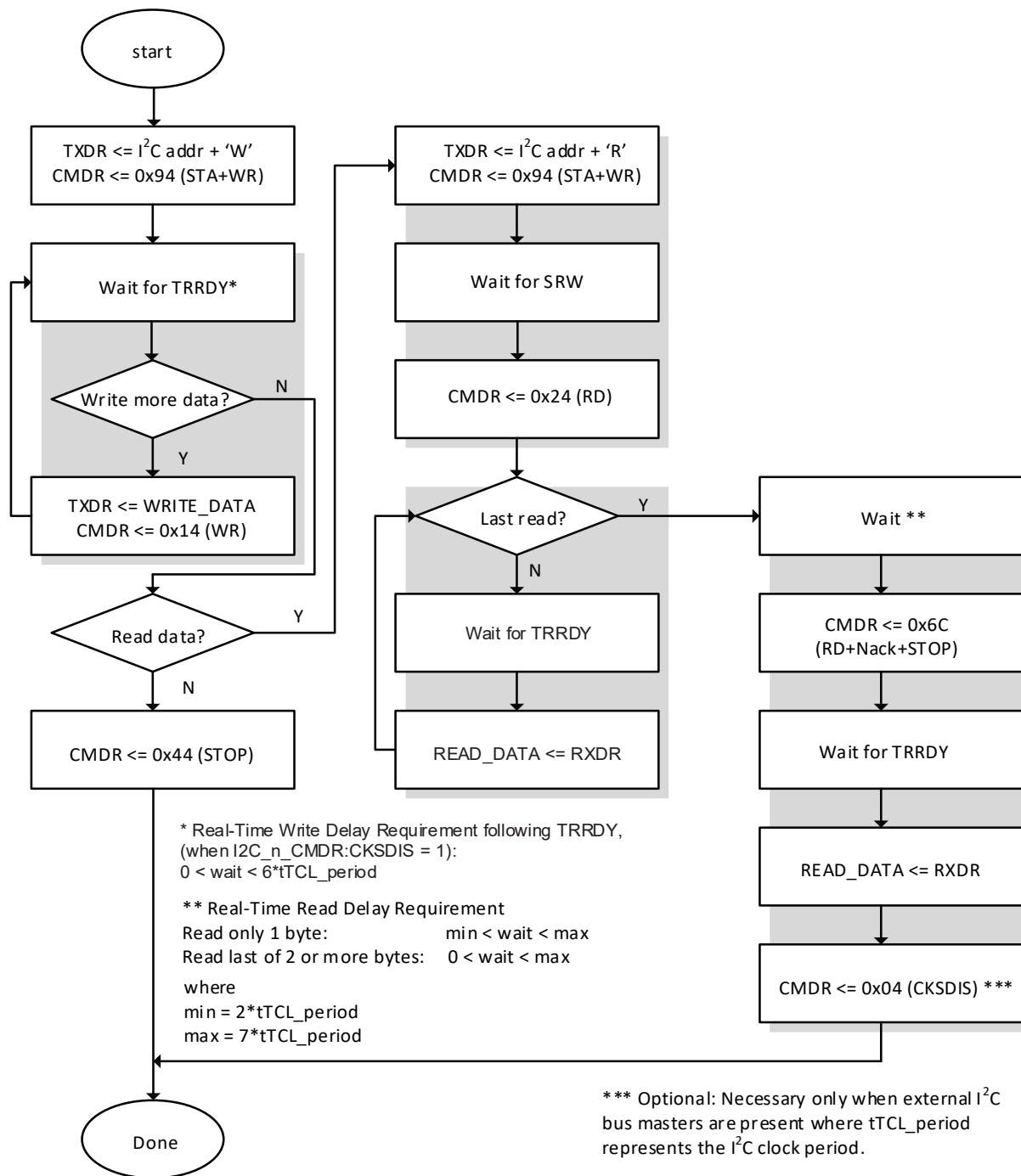


Figure 2.1. I²C Controller Read/Write Example (via WISHBONE)

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

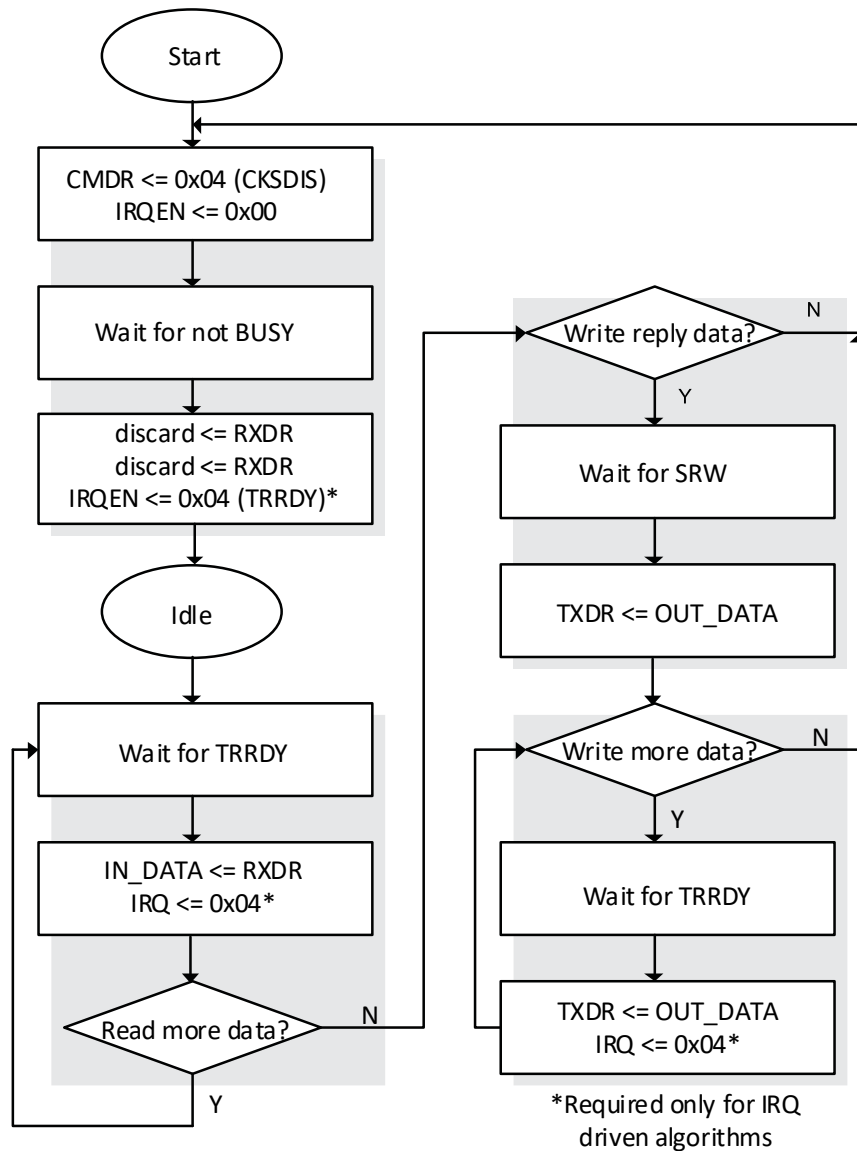


Figure 2.2. I²C Target Read/Write Example (via WISHBONE)

3. Typical I²C Transactions

Figure 3.1, Figure 3.2, and Figure 3.3 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 commands structures.

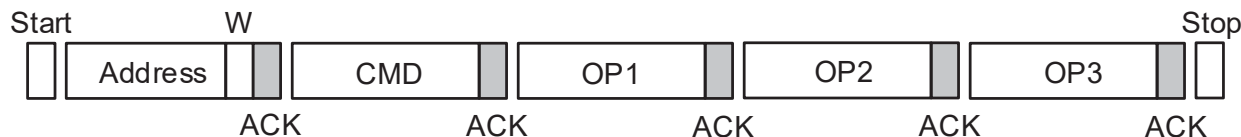


Figure 3.1. Simple I²C Command (for example, ISC_ERASE)

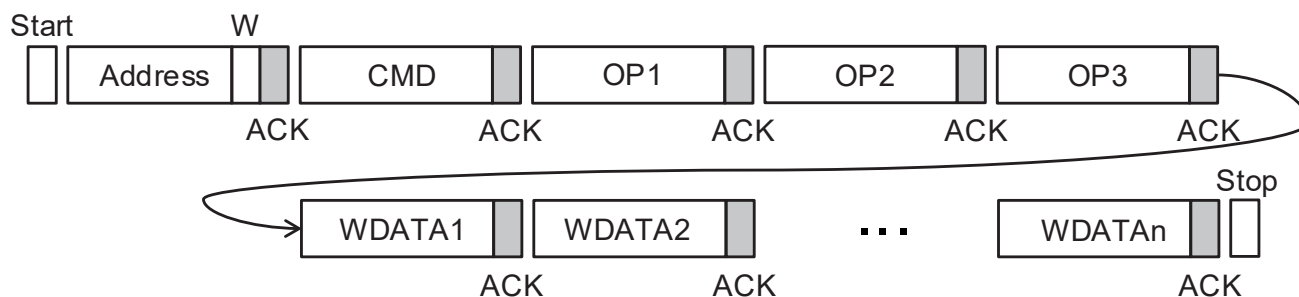


Figure 3.2. I²C Command with Write Data (for example, LSC_PROG_INCR_NV)

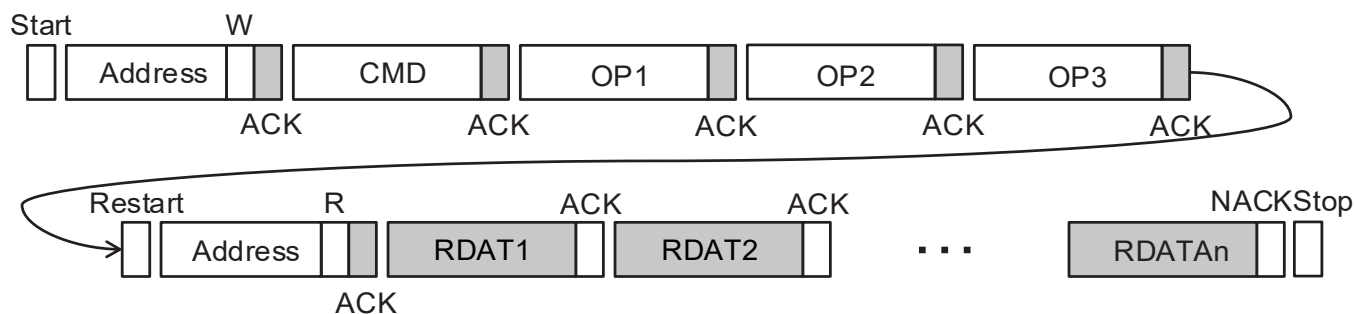


Figure 3.3. I²C Command with Read Data (for example, LSC_READ_STATUS)

4. I²C Functional Waveforms

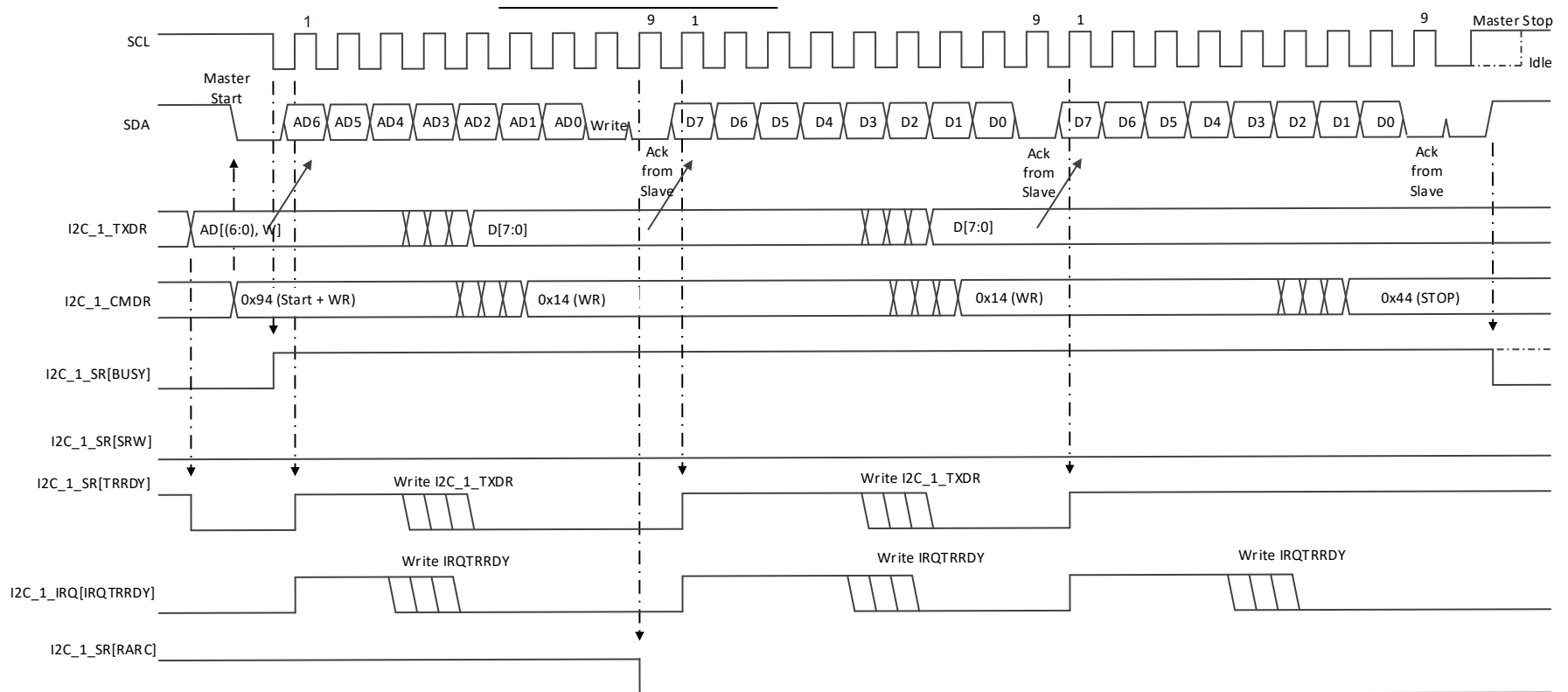


Figure 4.1. EFB Controller – I²C Write

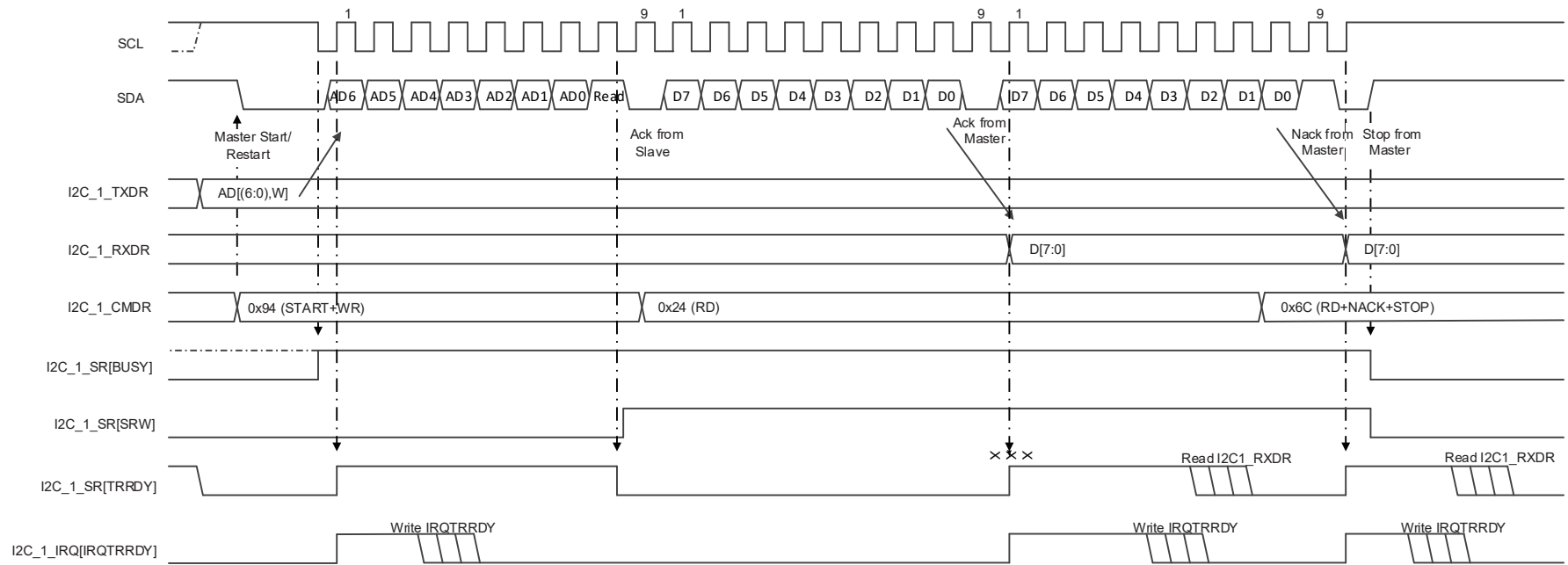


Figure 4.2. EFB Controller – I²C Read

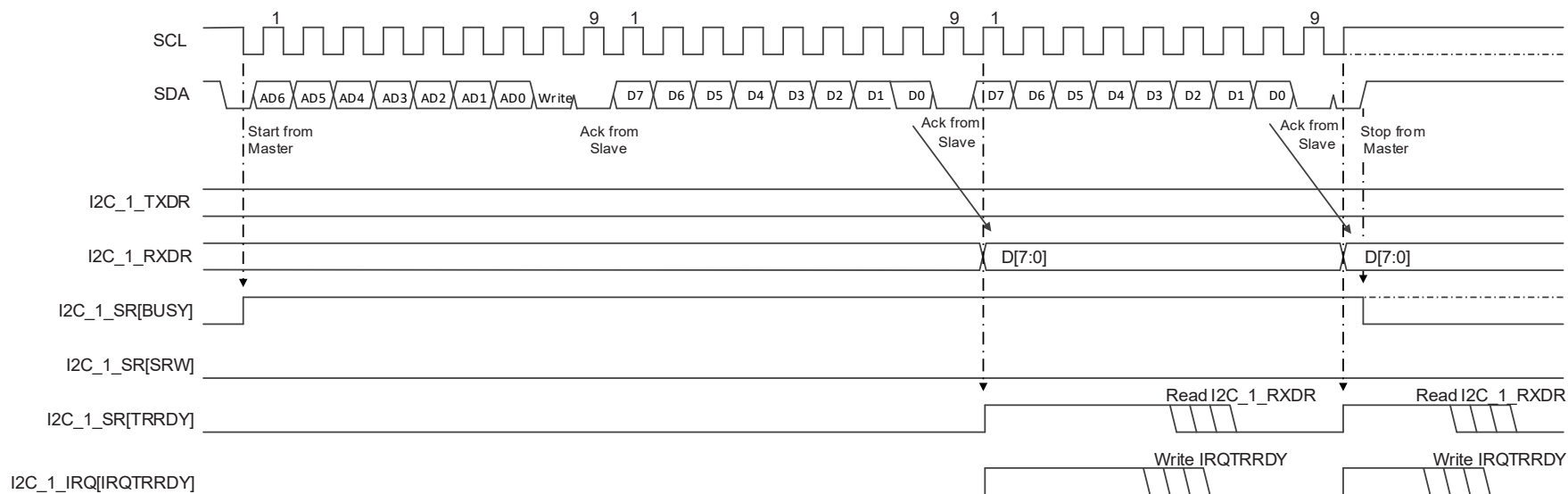


Figure 4.3. EFB Target – I²C Write

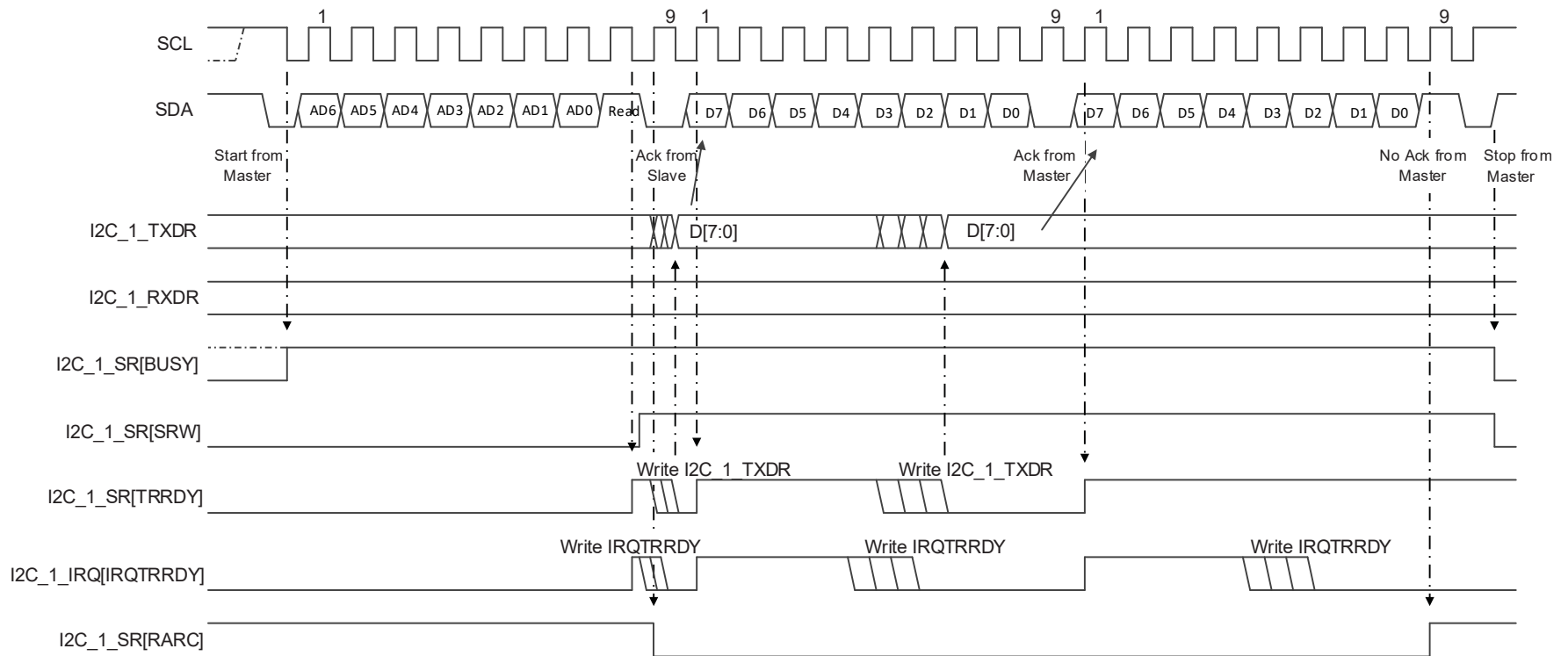


Figure 4.4. EFB Target – I²C Read

5. I²C Timing Diagram

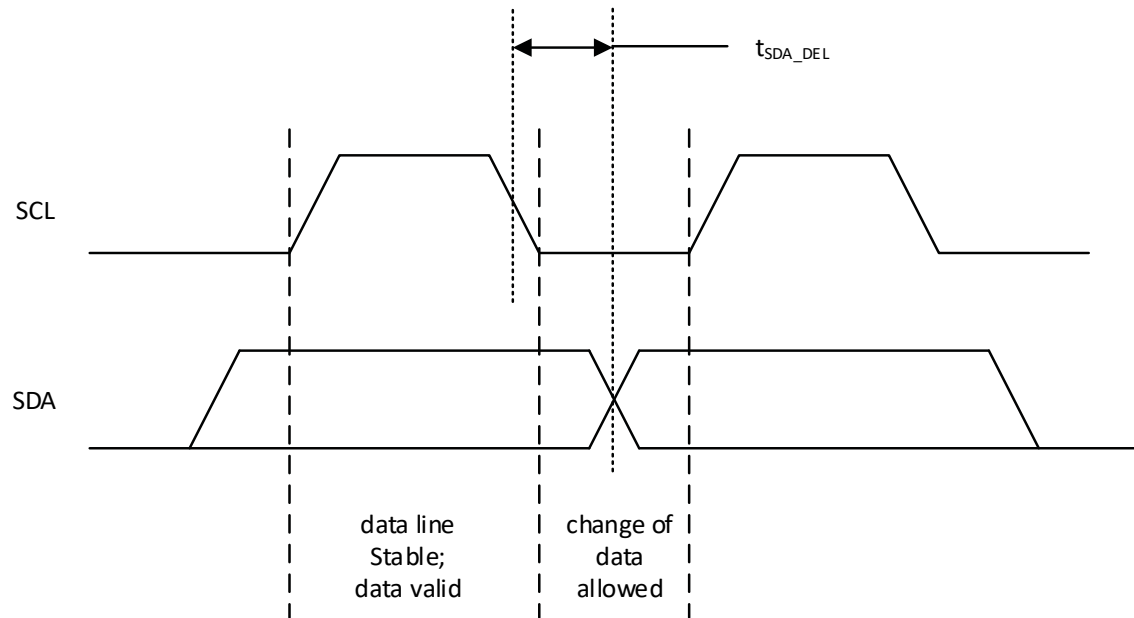


Figure 5.1. I²C Bit Transfer Timing

6. I²C Simulation Model

The I²C EFB Register Map translation to the MachXO3L/LF EFB software simulation model is provided in below.

Table 6.1. I²C Primary Simulation Mode

I ² C Primary Register Name	Register Size/Bit Location	Register Function	Address I ² C 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/njport_unit/i2c_1st/

I ² C Primary Register Name	Register Size/Bit Location	Register Function	Address I ² C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
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 Data	—	—	i2crxdr[7:0]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/njport_unit/i2c_1st/

I ² C Primary Register Name	Register Size/Bit Location	Register Function	Address I ² C Primary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_Receive_Data[7:0]	[7:0]	—	—	—	i2cxdrr[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/
IRQARBLN	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 6.2. I²C Secondary Simulation Mode

I ² C Secondary Register Name	Register Size/Bit Location	Register Function	Address I ² C 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/

I ² C Secondary Register Name	Register Size/Bit Location	Register Function	Address I ² C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_PRESCALE[9:8]	[1:0]	—	—	—	i2cbr[9:8]	../efb_top/config_plus_inst/config_core_inst/cfg_cdu/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/

I ² C Secondary Register Name	Register Size/Bit Location	Register Function	Address I ² C Secondary	Access	Simulation Model Register Name	Simulation Model Register Path
I2C_2_IRQ	[7:0]	IRQ	0x52	Read/Write	{1'b0, 1'b0, 1'b0, 1'b0, i2csr_2nd_irqsts_3, i2csr_2nd_irqsts_2, i2csr_2nd_irqsts_1, i2csr_2nd_irqsts_0}	../efb_top/efb_pll_sci_inst/u_efb_sci/
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/
IRQARBLN	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/

7. Hardened SPI IP Core

The MachXO3L/LF contains a hard SPI IP core that can be configured as a 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.

8. SPI Registers

The SPI core communicates with the WISHBONE interface through a set of control, command, status and data registers. Table 8.1 shows the register names and their functions. These registers are a subset of the EFB register map.

Table 8.1. 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 shall be written '0'.

Table 8.2. 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 deassertion 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 8.3. 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 will cause 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 via 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 which clock edge to transmit SPI data. Refer to [Figure 11.1](#) to [Figure 11.4](#).

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 8.4. 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 a 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 should scan for the initial 0x00 byte when reading the SPI target to indicate the beginning of actual data. Refer to [Figure 10.2](#).

- 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 11.1](#) to [Figure 11.4](#).

- 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 11.1](#) to [Figure 11.4](#).

- 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, the user 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, 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 forced the SPI system into idle state. Refer to [Figure 11.1](#) through [Figure 11.4](#).

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 8.5. SPI Clock Pre-scale

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

*Note: Hardware default value may be overridden by EFB component instantiation parameters. See 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 a SPI core reset. The DIVIDER value must be greater than 1.

Note: The digital value is calculated by IPexpress 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, both entered by the user. The digital value of the divider is programmed in the MachXO3L/LF 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. Designers can update the clock pre-scale register dynamically during device operation.

Table 8.6. 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 via 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 8.7. 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 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. See [Figure 15.1](#). This timing requires at least one protocol dummy byte be included for all target SPI read operations.

Table 8.8. 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 8.9. 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 LSB and is received last when LSBF=0 or first when LSBF=1.

Table 8.10. 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 8.11. 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 8.1 shows a flow diagram for controlling Controller SPI reads and writes initiated via the WISHBONE interface.

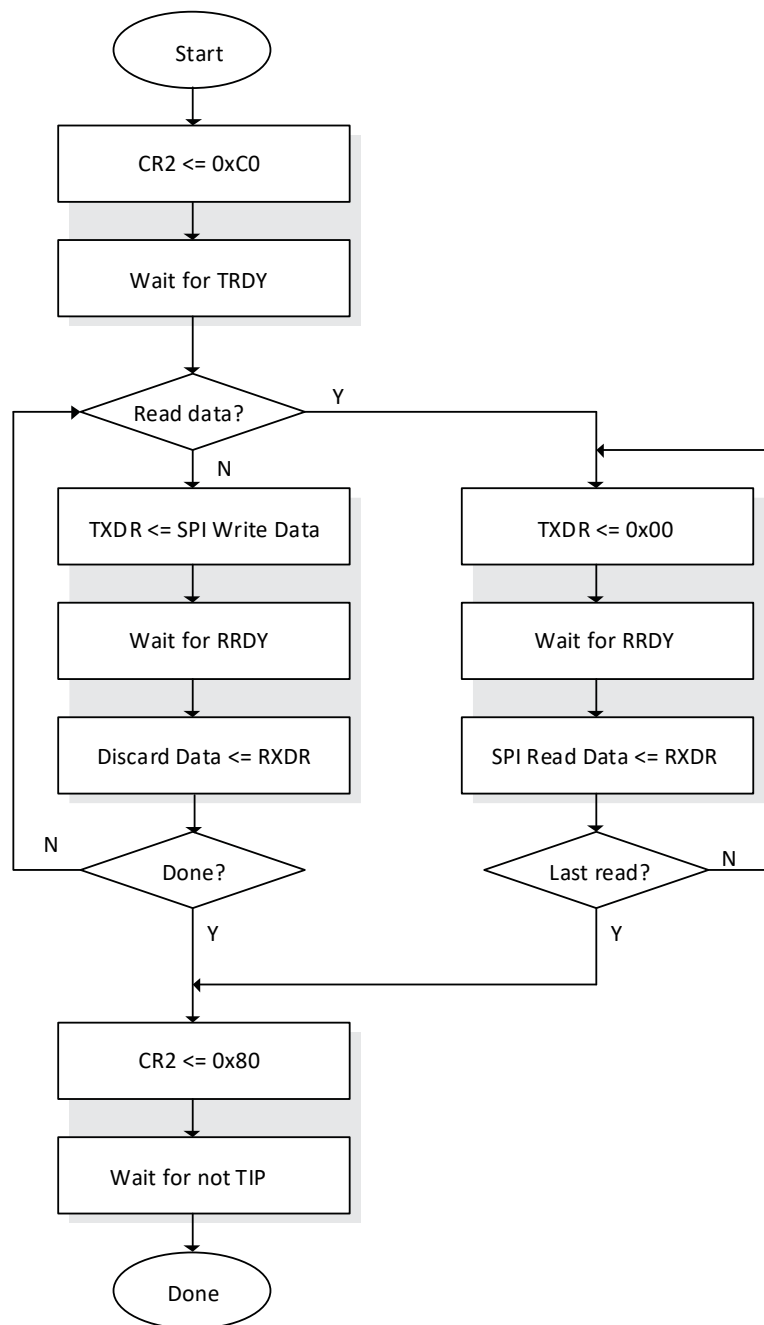


Figure 8.1. SPI Controller Read/Write Example (via WISHBONE) – Production Silicon

Note: Assumes CR2 register, MSCH = '1'. The algorithm when MSCH = '0' is application dependent and not provided. See [Figure 10.1](#) for guidance.

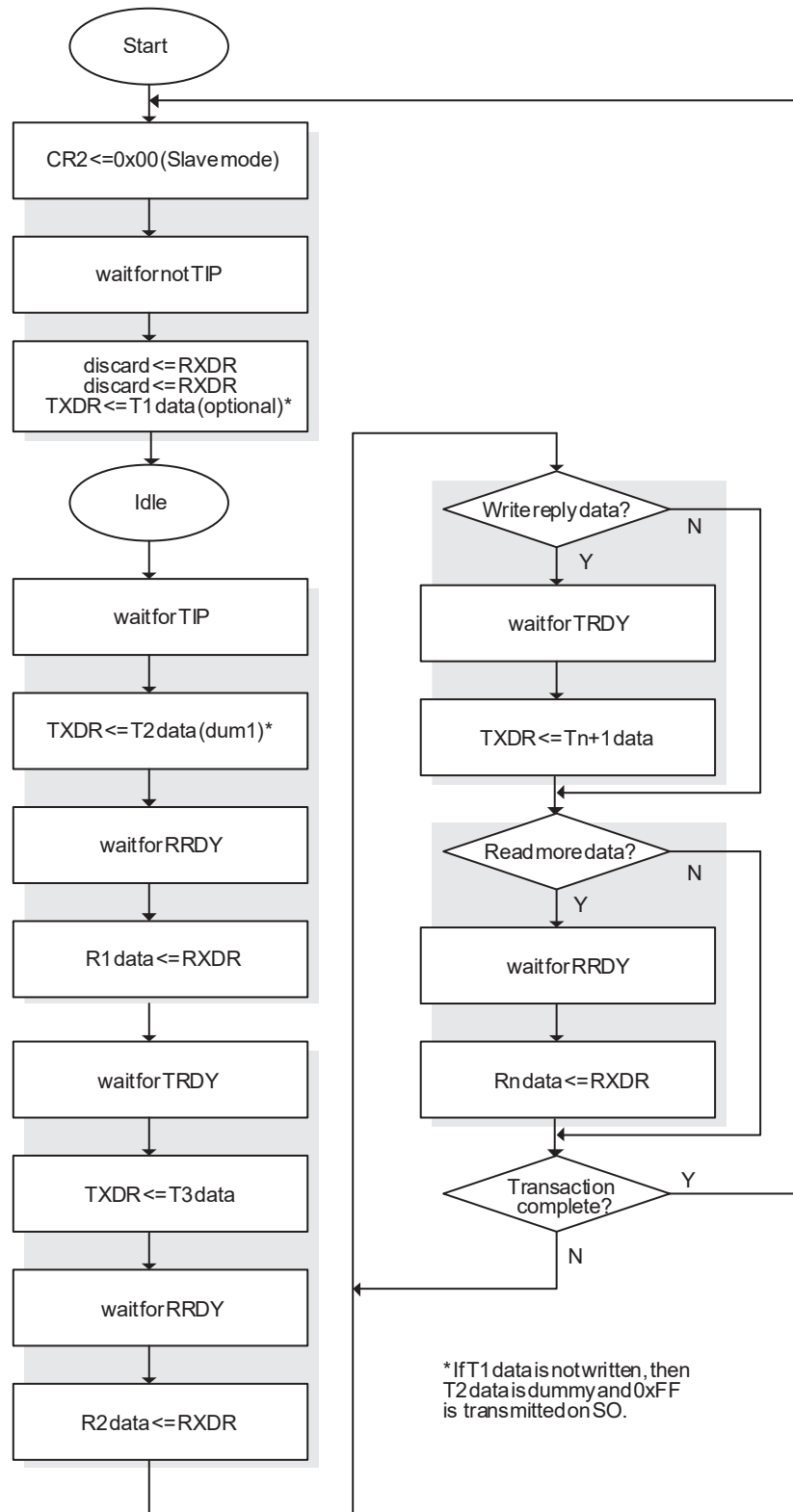


Figure 8.2. SPI Target Read/Write Example (via WISHBONE) – Production Silicon

9. Typical SPI Transactions

Figure 9.1, Figure 9.2, and Figure 9.3 illustrate typical User SPI bus protocol transactions that are supported by the Controller and Target flows shown in Figure 8.1, and Figure 8.2. Additionally, the figures below reference typical sysConfig Configuration commands structures.

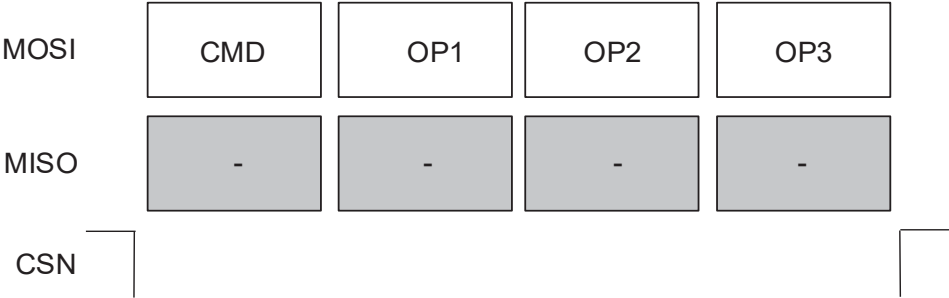


Figure 9.1. Simple SPI Command (for example, ISC_ERASE)

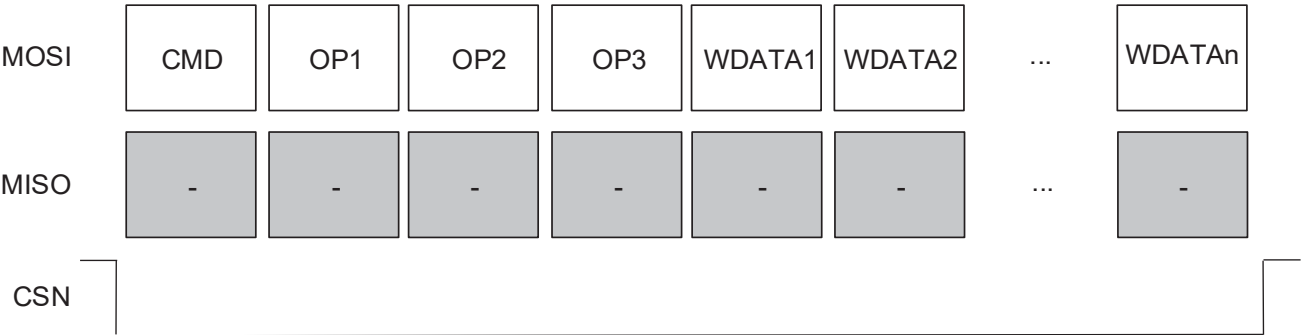


Figure 9.2. SPI Command with Write Data (for example, LSC_PROG_INCR_NV)

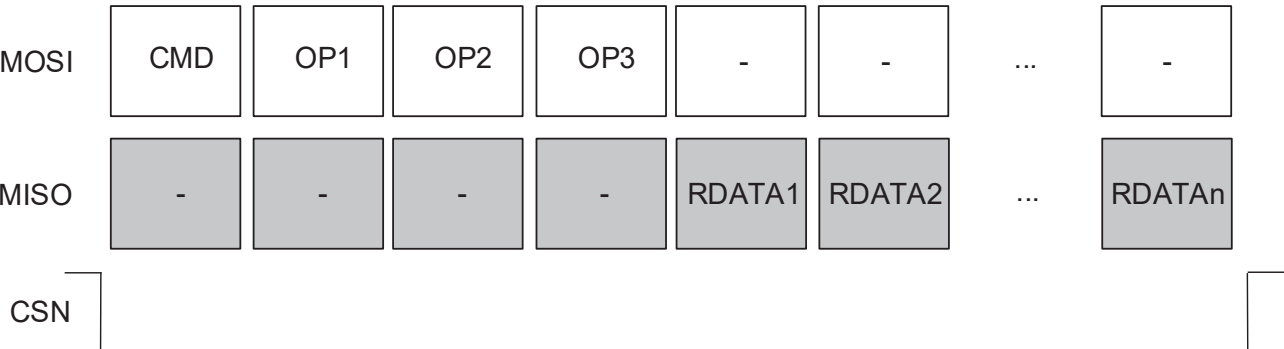


Figure 9.3. SPI Command with Read Data (for example, LSC_READ_STATUS)

10. SPI Functional Waveforms

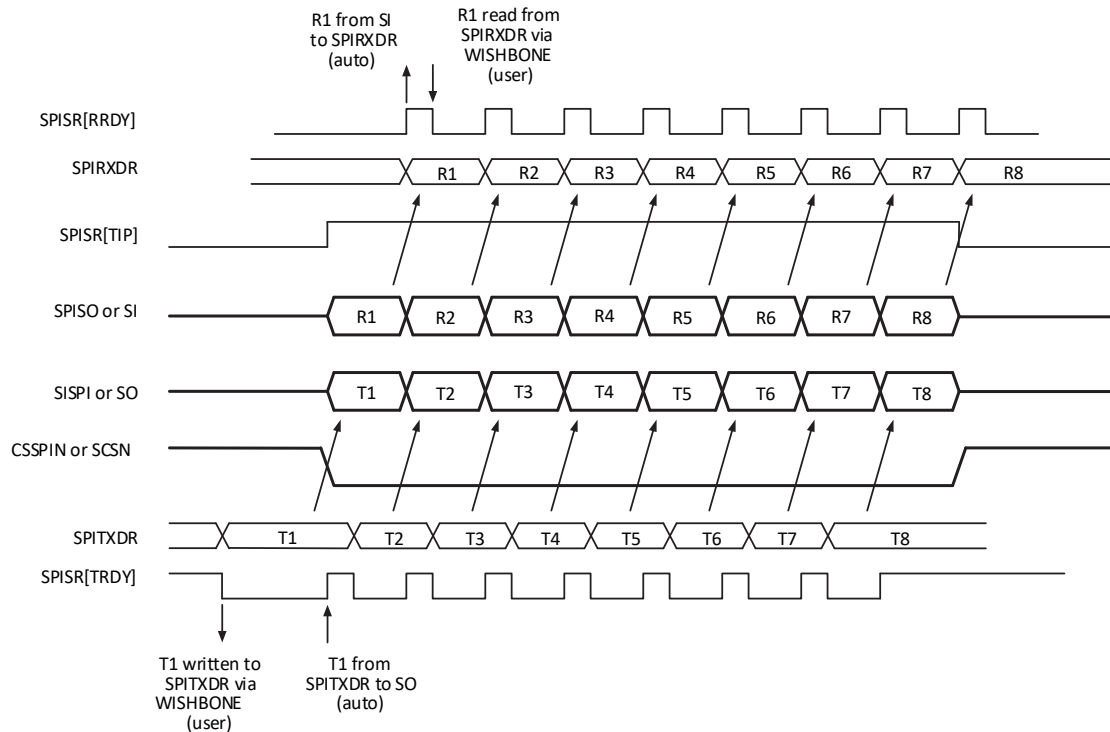
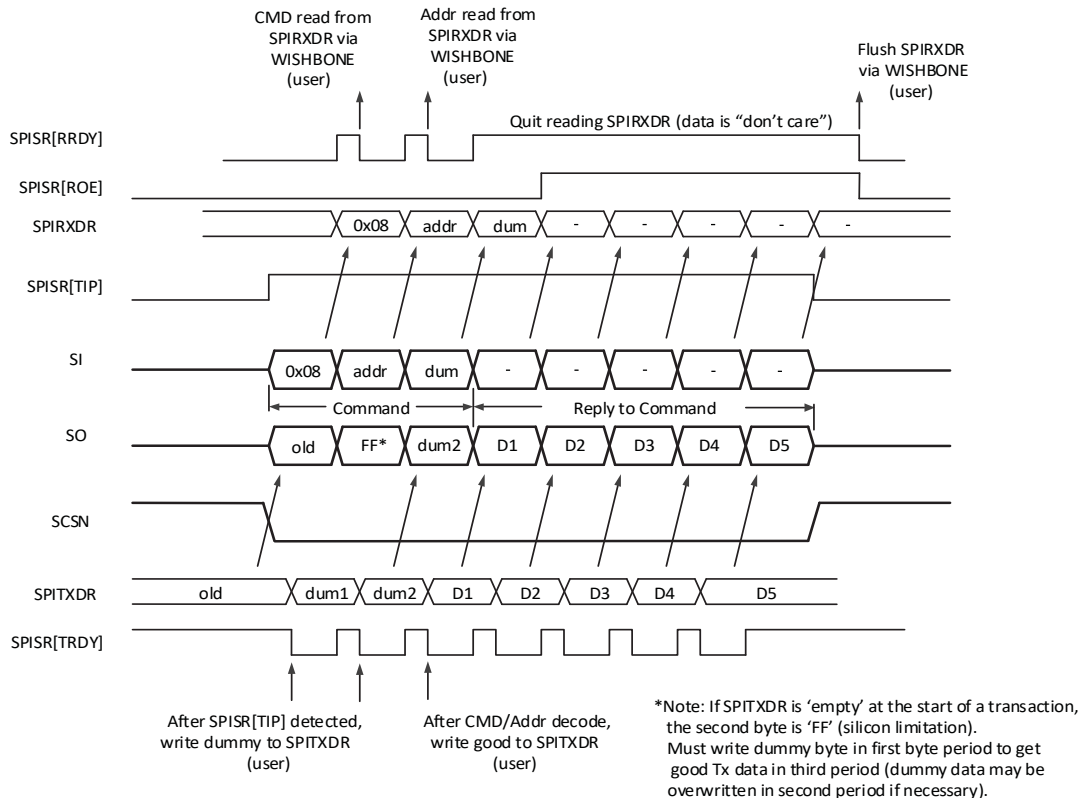


Figure 10.1. Fully Specified SPI Transaction (MachXO3L/LF as SPI Controller or Target)



11. SPI Timing Diagrams

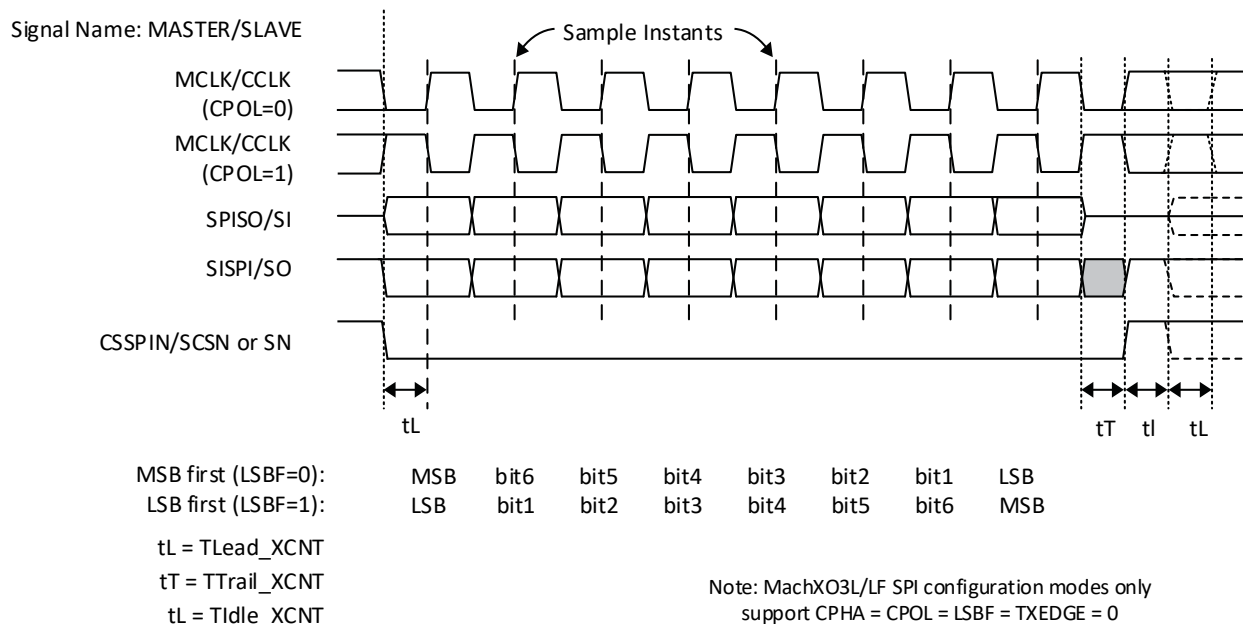


Figure 11.1. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0)

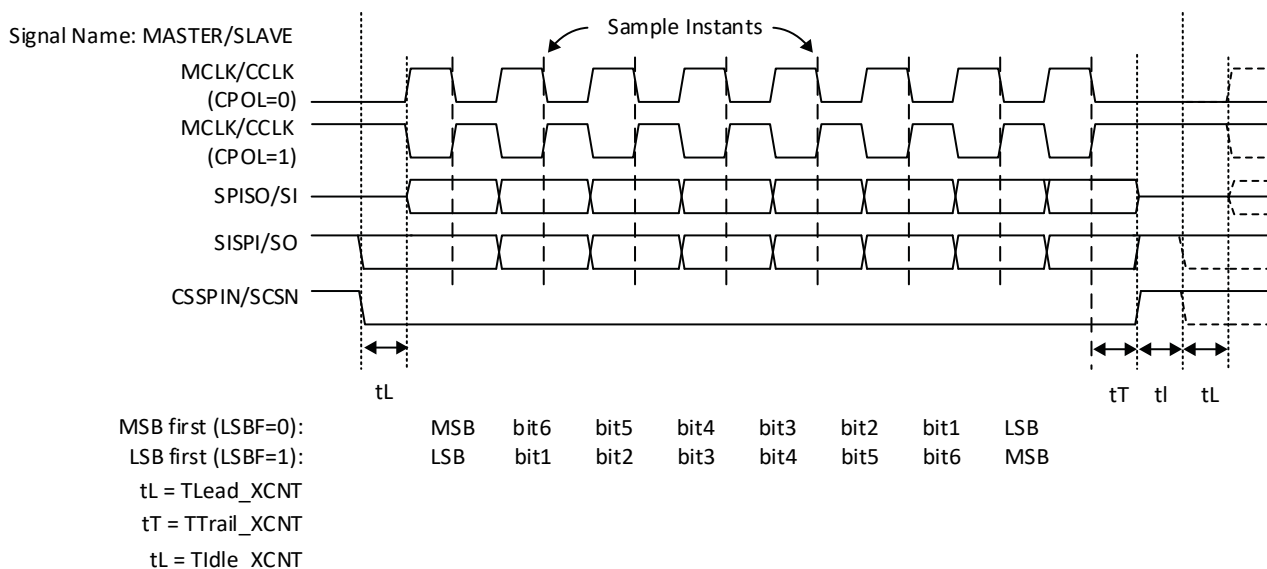


Figure 11.2. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0)

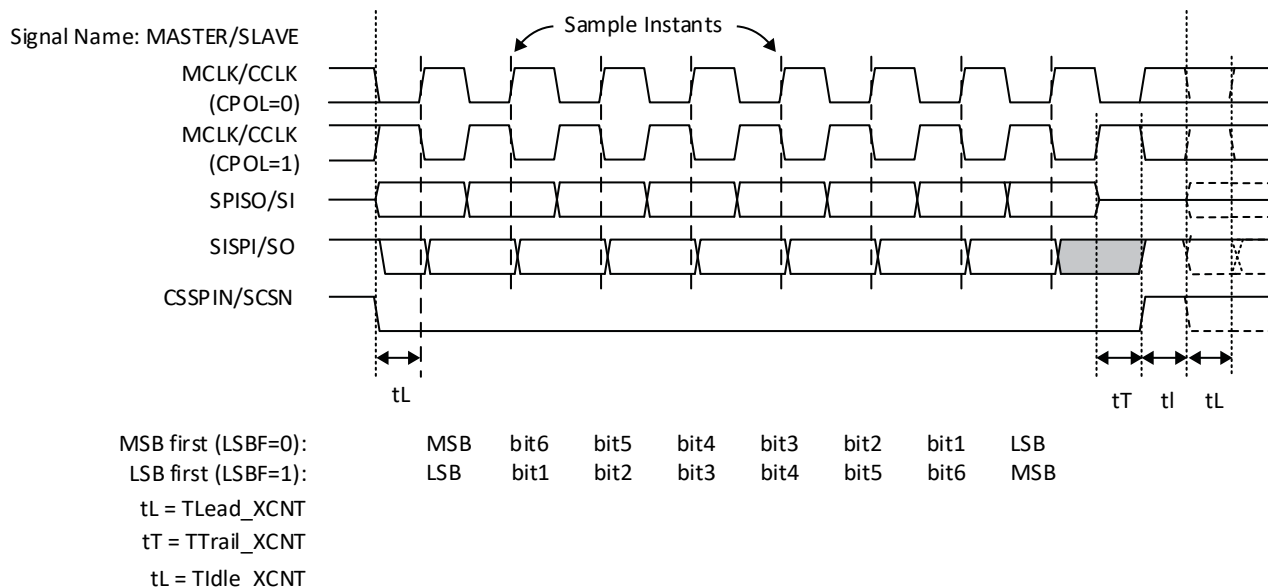


Figure 11.3. SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1)

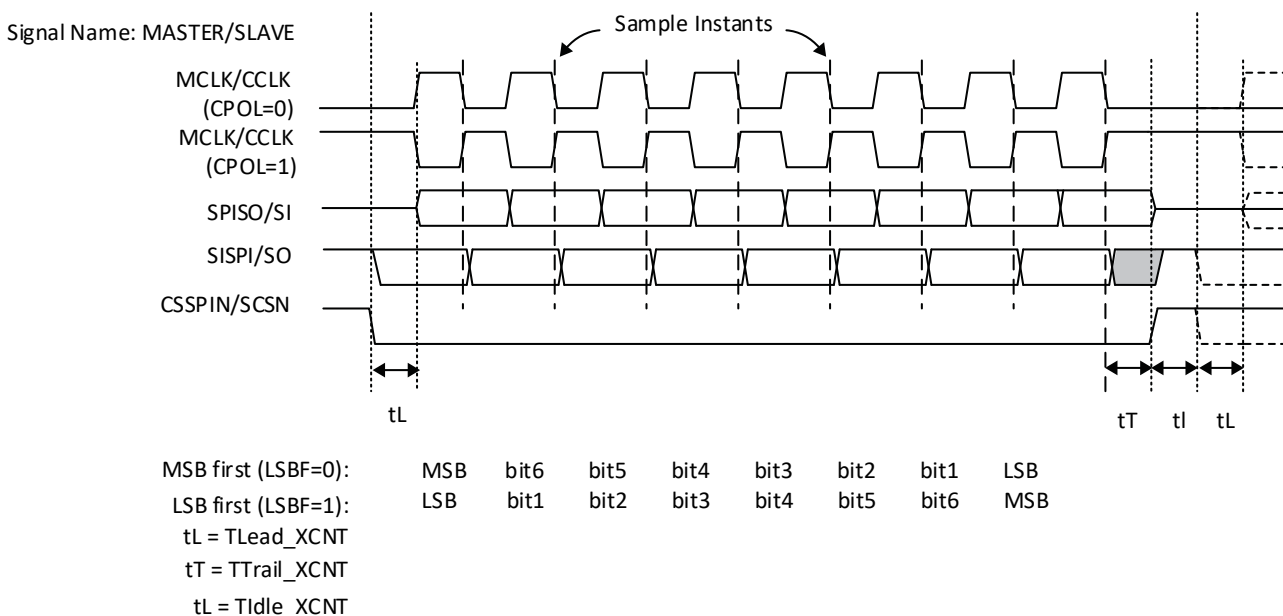


Figure 11.4. SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)

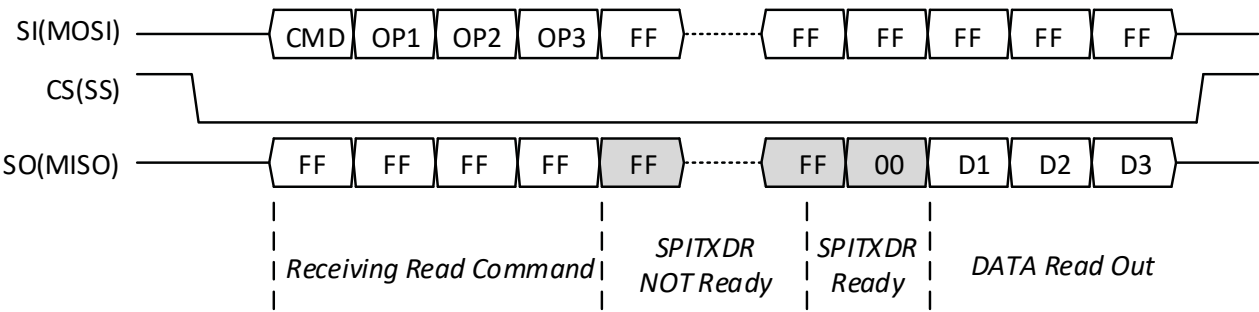


Figure 11.5. Target SPI Dummy Byte Response (SPICR2[SDBRE]) Timing

12. SPI Simulation Model

The SPI EFB Register Map translation to the MachXO3L/LF EFB software simulation model is provided below.

Table 12.1. 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_mcsch	../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_lsb	../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/

13. Hardened Timer/Counter PWM

The MachXO3L/LF EFB contains a hard Timer/Counter IP core. This Timer/Counter is a general purpose, bi-directional, 16-bit Timer/Counter module with independent output compare units and PWM support.

13.1. Timer/Counter Registers

The Timer/Counter communicates with the FPGA logic through the WISHBONE interface, by utilizing a set of control, status and data registers. Table 13.1 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 13.1. 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 13.2. 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 13.3. Timer/Counter Control 1

TCCR1							0xSF	
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 is not routed to the interrupt output signal.

ICEN

Enables the ability to perform a capture operation of the counter value. Users 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 13.4. 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
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.

Table 13.5. 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
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.

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 13.6. 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
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.

Table 13.7. 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
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.

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 13.8. 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

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_clk. 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 13.9. 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 13.10. 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 13.11. 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 13.12. Timer/Counter Current Top Counter Value 1

TCTOP1								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 TCTOP-SET0/1. 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 13.13. 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 13.14. 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 TCO-CRSET0/1. 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 13.15. 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 13.16. 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 13.17. 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 user asserts the TC_IC input signal. 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 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 13.18. 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 13.19. 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.

14. Timer Counter Simulation Model

The Timer Counter EFB Register Map translation to the MachXO3L/LF EFB software simulation model is provided below.

Table 14.1. Timer/Counter Simulation Mode

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], tc_top_set[4],	../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[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 Value [15:8]	0x63	Write	{tc_ocr_set[15], tc_ocr_set[14],	../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_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 Counter Value	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
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/
IRQOCRFEN	1	—	—	—	tc_ocrf_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
IRQOVFEN	0	—	—	—	tc_ovf_irqena	../efb_top/efb_pll_sci_inst/u_efb_sci/

15. NVCM (MachXO3L)/Flash (MachXO3LF) Access

Designers can access the NVCM/Flash Logic interface using the JTAG, SPI, I²C, or WISHBONE interfaces. The MachXO3L/LF NVCM/Flash consists of three sectors:

- Configuration NVCM0/Flash (includes USERCODE)
- NVCM1/UFM
- Feature Row

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

15.1. NVCM/Flash Access Ports

Designers can access the NVCM/Flash via JTAG port (compliant with the IEEE 1149.1 and IEEE 1532 specifications), external Target SPI port and external I²C Primary port and the internal WISHBONE interface of the EFB module.

Figure 15.1 illustrates the interfaces to the NVCM/Flash sectors.

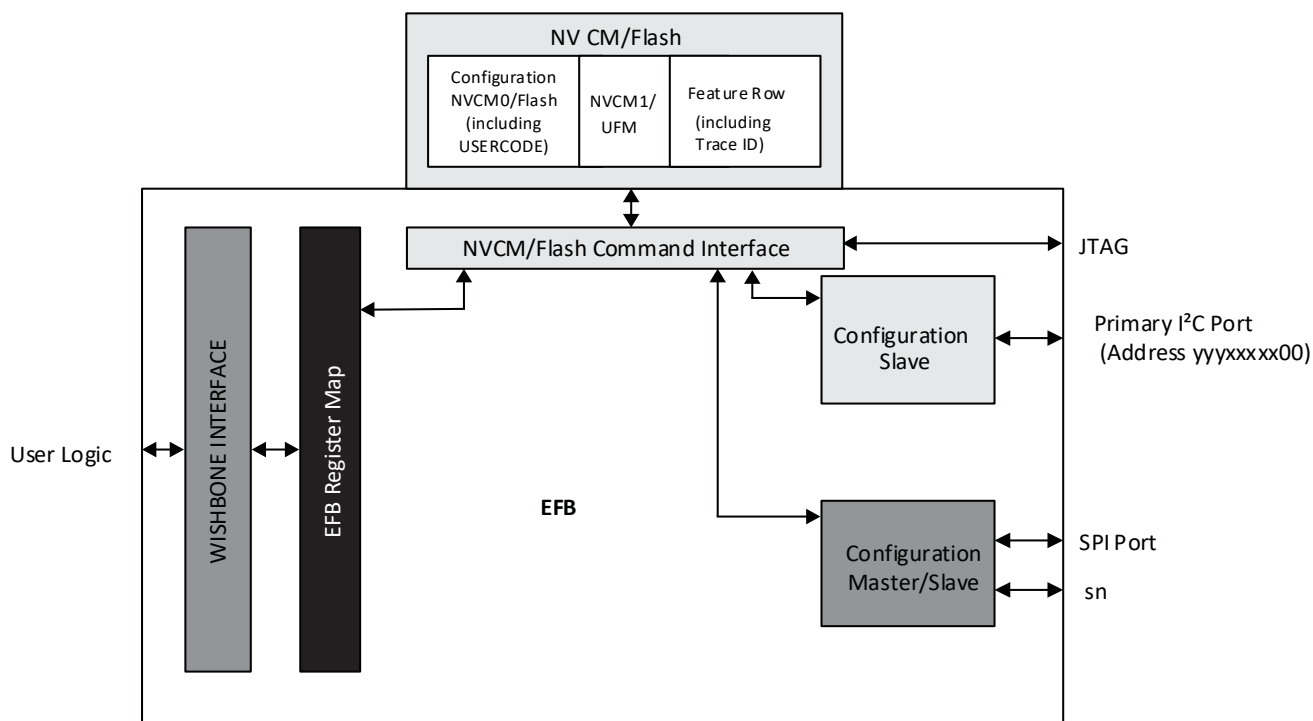


Figure 15.1. Interfaces to the NVCM/Flash Sectors

Notes:

- Only MachXO3L devices have NVCM.
- Only MachXO3LF devices have Flash.

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

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

Note: Enabling NVCM/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 NVCM/Flash Interface is disabled using Disable Configuration Interface command 0x26 followed by Bypass command 0xFF.

15.2. NVCM/Flash Access through WISHBONE Target Interface

The WISHBONE Target interface of the EFB module enables designers to access the NVCM/Flash directly from the FPGA core logic. The WISHBONE bus signals, described earlier in this document, are utilized by a WISHBONE host that designers 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 15.1](#) 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 15.1. WISHBONE to NVCM/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
CFG_RXDR	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 15.2. NVCM/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 NVCM/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. See the [Command and Data Transfers to NVCM/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 15.3. NVCM/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 NVCM/Flash logic. Bit 0 is LSB.

Table 15.4. NVCM/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 15.5. NVCM/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 NVCM/Flash logic. Bit 0 in this register is LSB.

Table 15.6. NVCM/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 15.7. NVCM/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 15.8. 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 15.9. 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

NVCM/Flash Interrupt Source. Indicates EFB interrupt source is from the NVCM/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.

15.3. Command and Data Transfers to NVCM/Flash Space

The command and data transferred to the NVCM/Flash are identical for all the access ports, regardless of their different physical interfaces. The NVCM/Flash is organized in pages. Therefore, users 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 NVCM/Flash is composed of three sectors, Configuration NVCM0/Flash (includes USERCODE), NVCM1/UFM, Feature Row. The Erase operations are sector based.

15.4. Command Summary by Application

Table 15.10. NVCM (Sector 1)/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 NVCM1/UFM access – All user I/Os (except the hardened user SPI and primary user I ² C 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 NVCM1 access – All user I/O (except persisted sysCONFIG ports) are tri-stated. User logic ceases to function, NVCM1/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 (NVCM1/UFM) access.
Set Address	0xB4	LSC_WRITE_ADDRESS	Set the NVCM1/UFM sector 14-bit Address Register.
Reset NVCM1/UFM Address	0x47	LSC_INIT_ADDR_NVCM1	Reset the address to point to Sector 1, Page 0 of the NVCM1/UFM.
Read NVCM1/UFM	0xCA	LSC_READ_TAG	Read the NVCM1/UFM data. Operand specifies the number of pages to read. Address Register is post-incremented.
Erase NVCM1/UFM	0xCB	LSC_ERASE_TAG	Erase the NVCM1/UFM sector only.
Program NVCM1/UFM Page	0xC9	LSC_PROG_TAG	Write one page of data to the NVCM1/UFM. Address Register is post-incremented.

Table 15.11. Configuration NVCM/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 PRO-GRAMN 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 NVCM0/Flash access – All user I/O (except the hardened user SPI and primary user I ² C 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 NVCM0/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 NVCM0/Flash Address	0xB4	LSC_WRITE_ADDRESS	Set the Configuration NVCM0/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 NVCM0/Flash Address	0x46	LSC_INIT_ADDRESS	Reset the address to point to Sector 0, Page 0 of the Configuration NVCM0/Flash.
Read NVCM0/Flash	0x73	LSC_READ_INCR_NV	Read the NVCM0/Flash data. Operand specifies number of the pages to read. Address Register is post-incremented.
Erase	0x0E	ISC_ERASE	Erase the Config NVCM0/Flash, FEATURE Row, FEABITS, Done bit, Security bits and USERCODE.
Program Page	0x70	LSC_PROG_INCR_NV	Write one page of data to the NVCM/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 NVCM0/Flash sector).
Program SECURITY PLUS	0xCF	ISC_PROGRAM_SECPLUS	Program the Security Plus bit (Secures CFG, NVCM0/Flash and NVCM1/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 15.12. Non-Volatile Register (NVR) Commands

Command Name	Command MSB LSB	SVF Command Name	Description
Read Trace ID 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 shift the most significant bit (MSB) first into the MachXO3L/LF device. This is required only when communicating with the configuration logic inside the MachXO3L/LF device.

In order to perform a Write, Read, or Erase operation with the NVCM, 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.

15.5. Command Descriptions by Command Code

Table 15.13. Erase NVCM/Flash (0x0E)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	Y	0E	See below	—	—	—

Operand 0000 ucfs 0000 0000 0000 0000(binary)

where:

- u: Erase UFM sector
 - 0: No action
 - 1: Erase
- c: Erase CFG NVCM0/Flash sector (Config NVCM/Flash, DONE, security bits, USERCODE)
 - 0: No action
 - 1: Erase
- f: Erase Feature sector (Target I2C address, sysCONFIG port persistence, Boot mode, etc.)
 - 0: No action
 - 1: Erase
- s: Erase SRAM
 - 0: No action
 - 1: Erase

Notes: Poll the BUSY bit (or wait, see [Table 19.1](#)) after issuing this command for erasure to complete before issuing a subsequent command other than Read Status or Check Busy.

Erased condition for NVCM/Flash bits = 0.

Examples:

0x0E 04 00 00

Erase CFG NVCM0/Flash sector.

0x0E 08 00 00

Erase NVCM1/UFM sector.

0x0E 0C 00 00

Erase NVCM1/UFM and CFG NVCM0/Flash sectors.

Table 15.14. Read TraceID Code (0x19)

NVCM1/ UFM	CFG NVCM0/ 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.

Table 15.15. Disable Configuration Interface (0x26)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	—	26	00 00	—	—	—

Example: 0x26 00 00

Disable NVCM/Flash interface for change access.

Note: 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.

Table 15.16. Read Status Register (0x3C)

NVCM1/ UFM	CFG NVCM0/ 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 1xEE Exxx xxxx xxFB xxCD xxxx xxxx

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

D bit 8 NVCM/Flash or SRAM Done Flag

When C = 0 SRAM Done bit has been programmed

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

When C=1 NVCM 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: The BUSY bit should be checked following all Enable, Erase or Program operations.

Note: Wait at least 1us 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).

Table 15.17. Reset CFG Address (0x46)

NVCM1/ UFM	CFG NVCM0/ 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.

Table 15.18. Reset NVCM1 Address (0x47)

NVCM1/ UFM	CFG NVCM0/ 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 NVCM/Flash Sector 1, page 0.

Table 15.19. Program DONE (0x5E)

NVCM1/ UFM	CFG NVCM0/ 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 us) after issuing this command for programming to complete before issuing a subsequent command other than Read Status or Check Busy.

Table 15.20. Program Configuration NVCM (0x70)

NVCM1/ UFM	CFG NVCM0/ 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 NVCM0

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.

Note: 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 NVCM0 sector prior to executing this command.

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.

Table 15.21. Read Configuration NVCM/Flash (0x73) (SPI)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	73	*(below)	R	** (below)	*** (below)

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

*Operand: 0001 0000 00pp pppp pppp pppp (binary)

pp..pp: num_pages Number of CFG NVCM0/Flash pages to read when num_pages = 1

Number of CFG NVCM0/Flash pages to read +1 when num_pages > 1

**Data Size (num_pages * 16) bytes

Note: Read CFG NVCM0/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.

***Examples: 0x73 10 00 01

0-byte dummy followed by one page of CFG NVCM0/Flash data (16 bytes total).

0x73 10 00 04

Read 1-page dummy followed by three pages of CFG NVCM0/Flash data (four pages total).

Table 15.22. Read Configuration NVCM/Flash (0x73) (I²C/SPI)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	0x73	*(below)	R	** (below)	*** (below)

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

*Operand: 0000 0000 00pp pppp pppp pppp (binary)

pp..pp: num_pages Number of CFG NVCM0/Flash pages to read when num_pages = 1

Number of CFG NVCM0/Flash pages to read +1 when num_pages > 1

**Data Size (num_pages * 16) bytes when num_pages=1

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

Note: Read CFG NVCM0/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.

***Examples: 0x73 00 00 01

0-byte dummy followed by one page CFG NVCM0/Flash data (16 bytes total)

0x73 00 00 04

Read 2-pages dummy, followed by three sets [1 page CFG NVCM0/Flash data, followed by four bytes dummy] (five pages and 12 dummy bytes total).

Table 15.23. Read Configuration NVCM/Flash (0x73) (WISHBONE)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	73	*(below)	R	** (below)	*** (below)

Note: This applies when Configuration NVCM/Flash is read through WISHBONE.

***Operand:** 0001 0000 00pp pppp pppp pppp (binary), or
0000 0000 00pp pppp pppp pppp (binary)
pp..pp: num_pages Number of CFG NVCM0/Flash pages to read when num_pages = 1
 Number of CFG NVCM0/Flash pages to read +1 when 1 < num_pages ≤ 12

Set to 0x3FFF when num_pages > 12

****Data Size** (num_pages * 16) bytes when num_pages=1
32 bytes + (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 CFG NVCM0/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.

*****Examples:** 0x73 00 00 01
0-byte dummy followed by one page CFG NVCM0/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 NVCM0/Flash data, followed by four bytes dummy] (five pages and 12 dummy bytes total).

Note: 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 [MachXO2 Programming and Configuration Usage Guide \(FPGA-TN 02155\)](#).

Table 15.24. Enable Configuration Interface (Transparent) (0x74)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	x	—	—	74	08 00 00 or 08 00	—	—	—

Note: The I²C interface uses only two operands all other interfaces use three operands. This command is required to enable modification of the NVCM1/UFM, configuration CFG NVCM0/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 us) after issuing this command for the NVCM/Flash pumps to fully charge.

Examples: 0x74 08 00 00

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

Table 15.25. Refresh (0x79)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	—	—	—	79	00 00	—	—	—

Example: 0x79 00 00

Issue Refresh command

Note: 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 tREFRESH. Violating this requirement causes the Refresh process to abort and leaves the MachXO3L/LF 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 via 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.

Table 15.26. STANDBY (0x7D)

NVCM1/ UFM	CFG NVCM0/ 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 MachXO3L/LF 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 [Power and Thermal Estimation and Management for MachXO3 Devices \(FPGA-TN-02059\)](#).

Table 15.27. Set Address (0xB4)

NVCM1/ UFM	CFG NVCM0/ 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 NVCM0/Flash

1: NVCM1/UFM

aa..aa:address14-bit page address

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

Set Address register to NVCM1/UFM sector, page 10 decimal

Table 15.28. Read USERCODE (0xC0)

NVCM1/ UFM	CFG NVCM0/ 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 NVCM0 sector
 EN Required = N Read 4-byte USERCODE from SRAM

Table 15.29. Program USERCODE (0xC2)

NVCM1/ UFM	CFG NVCM0/ 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.

Table 15.30. Enable Configuration Interface (Offline) (0xC6)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	—	C6	0y 00 00	—	—	—

Operand: 08 00 00 - Enable NVCM/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 NVCM/Flash interface for offline change access.

Notes: Use this command to enable offline modification of the NVCM/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. NVCM1/UFM remains accessible.
 Poll the BUSY bit (or wait 5 μs) after issuing this command for the NVCM/Flash pumps to fully charge.

Table 15.31. Program NVCM1 (0xC9)

NVCM1/ UFM	CFG NVCM0/ 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 NVCM0 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 NVCM1/UFM sector prior to executing this command.
 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.

Table 15.32. Read NVCM/UFM (0xCA) (SPI)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	*(below)	R	**(below)	*** (below)

*Operand: 0001 0000 00pp pppp pppp pppp (binary)

where: pp..pp: num_pages Number of NVCM1/UFM pages to read when num_pages = 1
Number of NVCM1/UFM pages to read +1 when num_pages > 1

**Data Size (num_pages * 16) bytes

Note: Read NVCM1/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.

***Examples: 0xCA 10 00 01
Read 0-byte dummy followed by one page NVCM1/UFM data (16 bytes total).
0xCA 10 00 04
Read one-page dummy followed by three pages NVCM1/UFM data (four pages total).

Table 15.33. Read NVCM/UFM (0xCA) (SPI/I²C)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	*(below)	R	**(below)	*** (below)

*Operand: 0000 0000 00pp pppp pppp pppp (binary)

where: pp..pp: num_pages Number of NVCM1/UFM pages to read when num_pages = 1
Number of NVCM1/UFM pages to read +1 when num_pages > 1

**Data Size (num_pages * 16) bytes when num_pages=1
32 bytes + (num_pages * 16 + 4) bytes when num_pages>1

Note: Read NVCM1/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.

***Examples: 0xCA 00 00 01
Read 0-byte dummy followed by one page NVCM1/UFM data (16 bytes total).
0xCA 00 00 04
Read two-page dummy followed by three sets [one page NVCM1/UFM data, followed by four bytes dummy] (five pages total and 12 dummy bytes).

Table 15.34. Read NVCM/UFM (0xCA) (WISHBONE)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CA	*(below)	R	**(below)	*** (below)

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

where: pp..pp: num_pages Number of NVCM1/UFM pages to read when num_pages = 1
Number of NVCM1/UFM pages to read +1 when $1 < \text{num_pages} \leq 12$
Set to 0x3FFF when num_pages > 12

**Data Size (num_pages * 16) bytes when num_pages=1
32 bytes + (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 NVCM1/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.

***Examples: 0xCA 00 00 01
Read 0 bytes dummy followed by one page NVCM1/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 NVCM1/UFM data, followed by four bytes dummy] (five pages total and 12 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 [MachXO2 Programming and Configuration Usage Guide \(FPGA-TN-02155\)](#).

Table 15.35. Erase NVCM1 (0xCB)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
x	—	—	Y	CB	00 00 00	—	—	—

Notes: Erased condition for NVCM1 bits = '0'.
Poll the BUSY bit (or wait, see [Table 19.1](#)) 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 NVCM1/UFM sector.

Table 15.36. Program SECURITY (0xCE)

NVCM1/ UFM	CFG NVCM0/ 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.

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.

SECURITY and SECURITY PLUS commands are mutually exclusive.

Table 15.37. Program SECURITY PLUS (0xCF)

NVCM1/ UFM	CFG NVCM0/ 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.

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.

SECURITY and SECURITY PLUS commands are mutually exclusive.

Table 15.38. Read Device ID Code (0xE0)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	N	E0	00 00 00	R	4B	See Table 15.39

Example: 0xE0 00 00 00

Read 4-byte device ID

Table 15.39. Device ID

Device Name	E Devices	C Devices
MachXO3L-640E-MG121	0xC1 2B 20 43	—
MachXO3L-1300	0x41 2B 20 43	0xC1 2B B0 43
MachXO3L-1300E-MG256	0xC1 2B 30 43	—
MachXO3L-2100	0x41 2B 30 43	0x41 2B B0 43
MachXO3L-2100E-MG324	0xC1 2B 40 43	—
MachXO3L-2100C-BG324	—	0xC1 2B C0 43
MachXO3L-4300	0x41 2B 40 43	0x41 2B C0 43
MachXO3L-4300C-BG400	—	0xC1 2B D0 43
MachXO3L-6900	0x41 2B 50 43	0x41 2B D0 43
MachXO3L-9400	0x41 2B 60 43	0x41 2B E0 43
MachXO3LF-640E-MG121	0xE1 2B 20 43	—
MachXO3LF-1300	0x61 2B 20 43	0xE1 2B B0 43
MachXO3LF-1300E-MG256	0xE1 2B 30 43	—
MachXO3LF-2100	0x61 2B 30 43	0x61 2B B0 43
MachXO3LF-2100E-MG324	0xE1 2B 40 43	—
MachXO3LF-2100C-BG324	—	0xE1 2B C0 43
MachXO3LF-4300	0x61 2B 40 43	0x61 2B C0 43
MachXO3LF-4300C-BG400	—	0xE1 2B D0 43
MachXO3LF-6900	0x61 2B 50 43	0x61 2B D0 43
MachXO3LF-9400	0x61 2B 60 43	0x61 2B E0 43

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

Table 15.40. Verify Device ID Code (0xE2)

NVCM1/ UFM	CFG NVCM0/ Flash	NVR	EN Required	CMD (Hex)	Operands (Hex)	Data Mode	Data Size	Data Format
—	x	—	Y	E2	00 00 00	W	4B	See Table 15.39

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.

Table 15.41. Program Feature (0xE4)

NVCM1/ UFM	CFG NVCM0/ 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 should be programmed ideally 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.

Table 15.42. Read Feature Row (0xE7)

NVCM1/ UFM	CFG NVCM0/ 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.

Table 15.43. Check Busy Flag (0xF0)

NVCM1/ UFM	CFG NVCM0/ 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)

Table 15.44. Program FEABITs (0xF8)

NVCM1/ UFM	CFG NVCM0/ 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 vv 00

Data Format:

- bb: Boot Sequence
 - If b=00 (Default) and m=0 then Single Boot from NVCM/Flash.
 - If b=00 and m=1 then Dual Boot from NVCM/Flash then External, if there is a failure.
 - If b=01 and m=1 then Single Boot from External Flash.
 - 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 should be programmed ideally during manufacturing.

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

Table 15.45. Read FEABITs (0xFB)

NVCM1/ UFM	CFG NVCM0/ 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
 - If b=00 (Default) and m=0 then Single Boot from NVCM/Flash.
 - If b=00 and m=1 then Dual Boot from NVCM/Flash then External, if there is a failure.
 - If b=01 and m=1 then Single Boot from External Flash.
 - 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

Table 15.46. Bypass (Null Operation) (0xFF)

NVCM1/ UFM	CFG NVCM0/ 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

16. Interface to NVCM/Flash

The WISHBONE interface of the EFB module allows a WISHBONE host to access the configuration resources of the MachXO3L/LF devices. This can be particularly useful for reading data from configuration resources such as USERCODE and TraceID. Most importantly, this feature allows users to update the NVCM/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 NVCM 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 NVCM1/UFM section of this document. This is because the NVCM1/UFM is also an NVCM/Flash resource and the communication between the WISHBONE host and the configuration logic is performed through the same command, status, and data registers. Please see [Table 15.1](#) to [Table 17.3](#) for information on these registers.

[Figure 16.1](#) shows a basic flow diagram for implementing a NVCM/Flash Update initiated via any of the sysCONFIG ports (I²C, SPI, or WISHBONE).

For detailed information on MachXO3L/LF programming and configuration, see [MachXO3 Programming and Configuration Usage Guide \(FPGA-TN-02055\)](#).

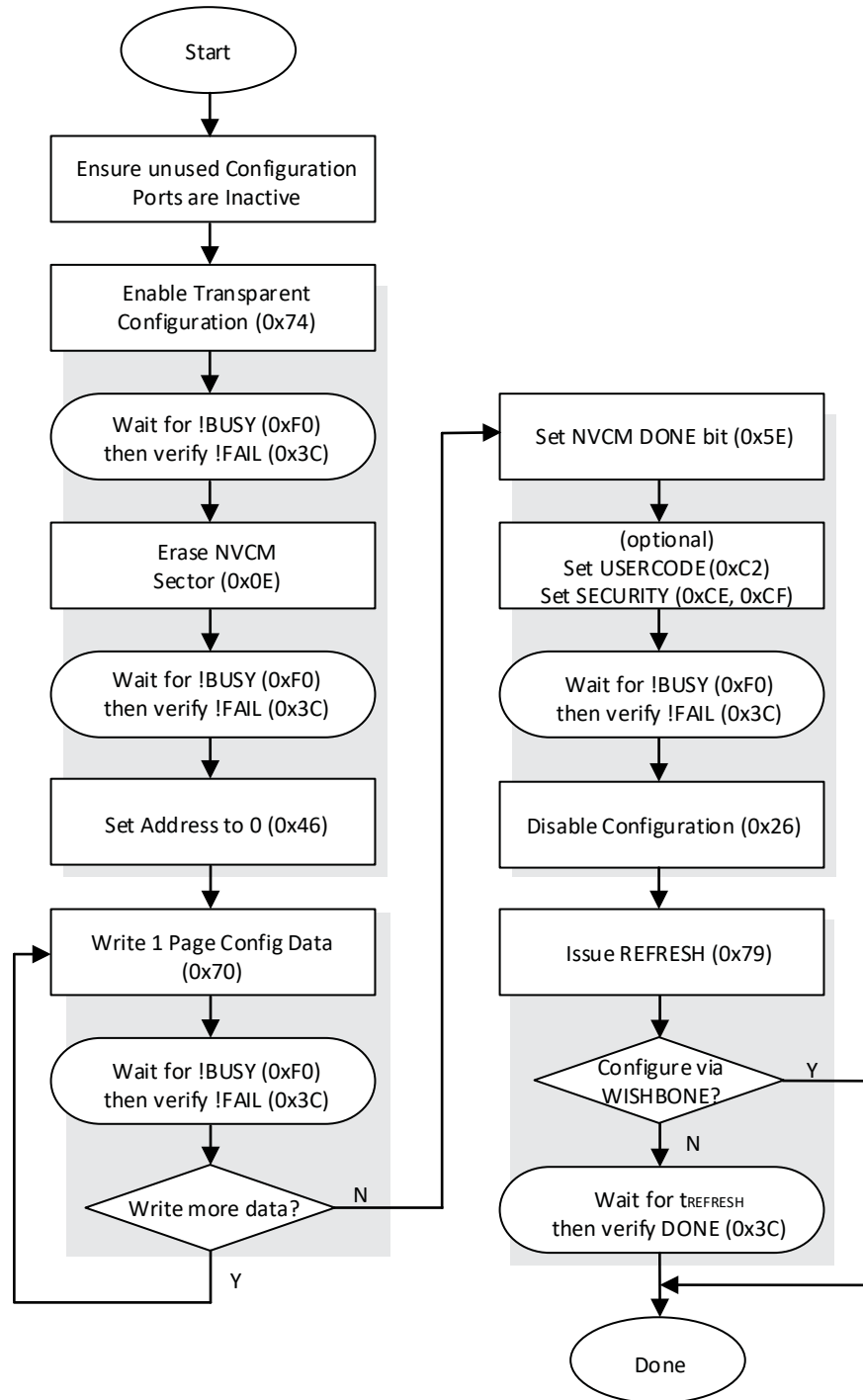


Figure 16.1. Basic Configuration NVCM1/UFM Program Example

17. Command Framing

17.1. 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 17.1. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
I ² C	Start	(Command/Operands/Data)	Stop

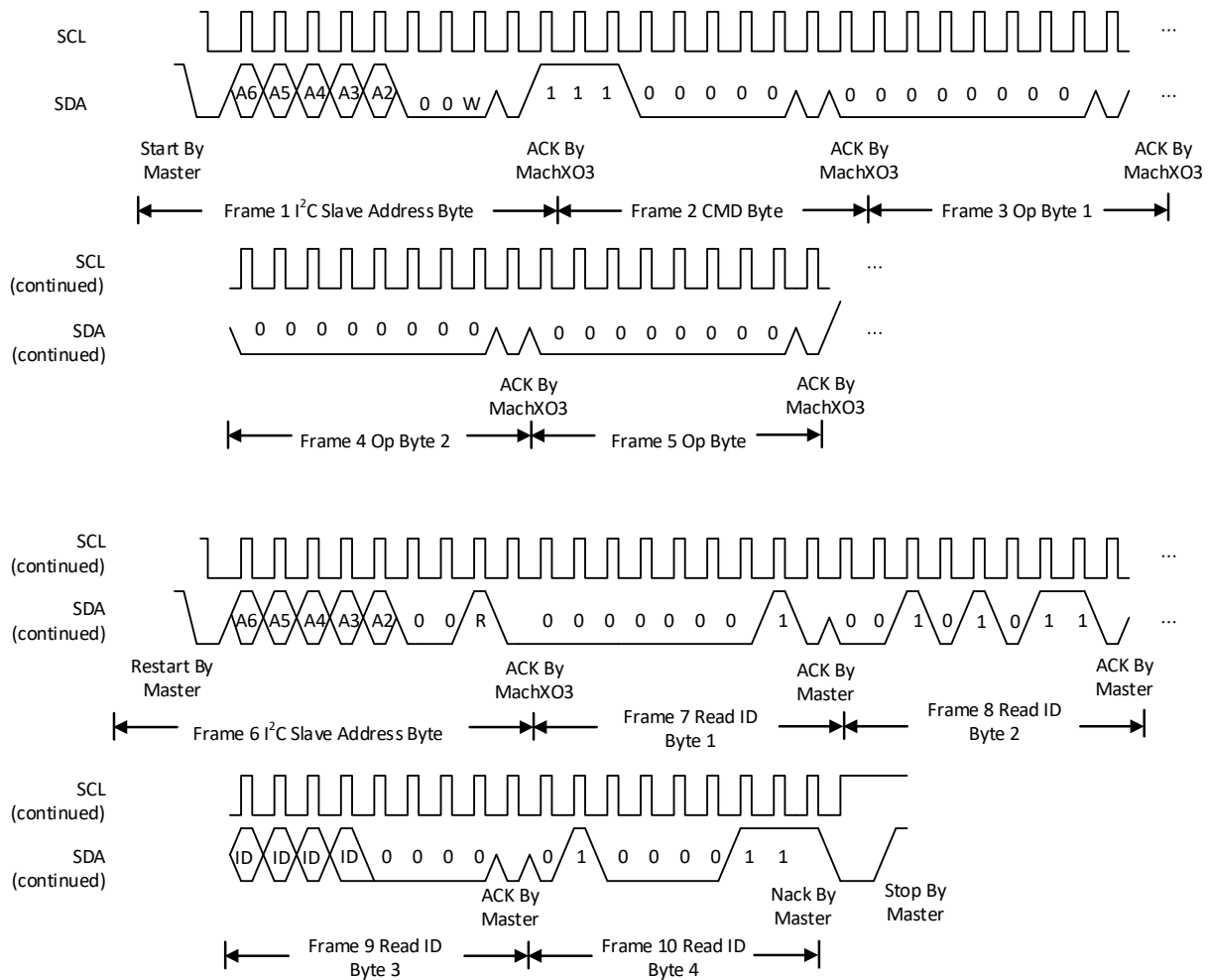


Figure 17.1. I²C Read Device ID Example

17.2. 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 17.2. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
SPI	Assert CS	(Command/Operands/Data)	De-assert CS

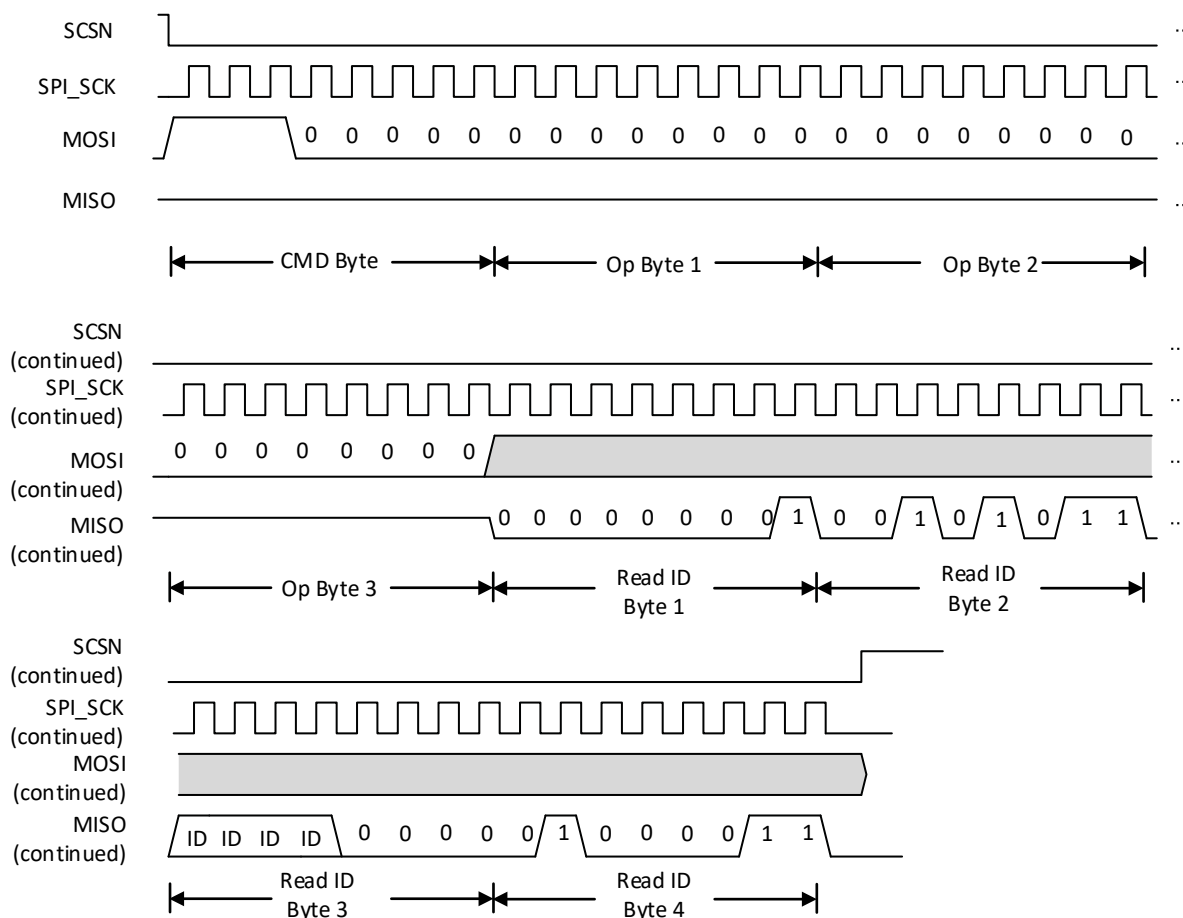


Figure 17.2. SSPI Read Device ID Example

17.3. WISHBONE Framing

To access the NVCM 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 17.3. Command Framing Protocol, by Interface

Interface	Pre-op (+)	Command String	Post-op (-)
WISHBONE	Assert CFGCR[WBCE]	(Command/Operands/Data)	De-assert CFGCR[WBCE]

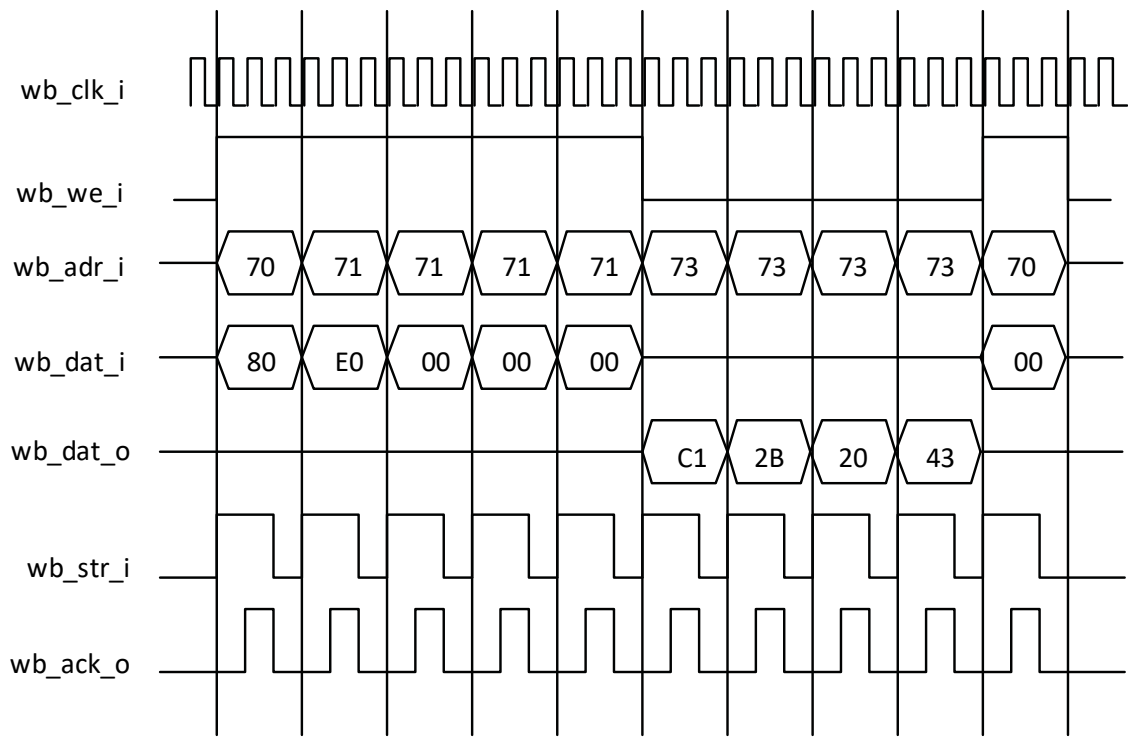


Figure 17.3. WISHBONE Read Device ID Example (-1200 HC Device)

18. NVCM1/Flash Write and Read Examples

The NVCM/Flash sectors support page-oriented read and write operations while erase operations are sector-based. Consistent with many NVCM/Flash devices, byte-oriented operations are not supported.

Table 18.1. Write Two NVCM1/UFM Pages

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 us if not polling.)
—	—	+	—	—	—
3	W	47	00 00 00	—	Init NVCM1/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 NVCM1/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 us 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 NVCM1/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 us if not polling.)
—	—	+	—	—	—
8	W	26	00 00	—	Disable Configuration Interface
—	—	—	—	—	—
—	—	+	—	—	—
9	W	FF	—	—	Bypass (NOP)
—	—	—	—	—	—

Notes:

- When accessing NVCM/Flash via WISHBONE, use CFGTXDR (0x71) to write data and CFDRXDR (0x73) to read data.
- + and – refer to the command framing protocol appropriate for the interface, discussed in the [Command Framing](#) section.

Table 18.2. Read One NVCM1/UFM Page (All Devices, WISHBONE/SPI)

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 us if not polling.)
—	—	+	—	—	—
3	W	B4	00 00 00	40 00 00 01	Set NVCM1/UFM Address to 0001
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 01	—	Read one page NVCM1/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 NVCM/Flash via 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 18.3. Read Two NVCM1/UFM Pages (WISHBONE/SPI)

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 us if not polling.)
—	—	+	—	—	—
3	W	47	00 00 00	—	Init NVCM1/UFM address to 0000
—	—	—	—	—	—
—	—	+	—	—	—
4	W	CA	10 00 03	—	Read two pages of NVCM1/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 NVCM/Flash via 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.

19. NVCM/Flash Performance

Table 19.1. NVCM/Flash Performance in MachXO3L/LF Device¹

		MachXO3 L/LF-640	MachXO3 L/LF-1300	MachXO3 L/LF-1300 256 Ball Package	MachXO3 L/LF-2100	MachXO3 L/LF-2100 324 Ball Package	MachXO3 L/LF-4300	MachXO3 L/LF-4300 400 Ball Package	MachXO3 L/LF-6900	MachXO3 L/LF-9400
CFG Erase (tEraseCFG)	Typ. Min.	800	800	1100	1100	1800	1800	2800	2800	4500
	Typ. Max.	1400	1400	1900	1900	3100	3100	4800	4800	7700
CFG Program (tProgramCFG)	All	500	500	740	740	1400	1400	2200	2200	3000
	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
NVCM1/UFM Erase (tEraseNVCM1/UFM)	Typ. Min.	400	400	500	500	600	600	900	900	1600
	Typ. Max.	700	700	900	900	1000	1000	1600	1600	2800
NVCM1/UFM Program (tProgramNVCM1/ UFM)	All	110	110	140	140	180	180	480	480	840
	1 page	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
tErase (max)	Note 2	12000	15000	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 time-outs.

20. 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 utilization. The overhead required by bus controller processes, if any, is not accounted for in the equation below.

E/P/V time (μ s): $t_{\text{EraseProgramVerify}} = t_{\text{Erase}} + t_{\text{Program}} + t_{\text{Verify}}$

where: $t_{\text{Erase}} = t_{\text{EraseCFG}} + t_{\text{EraseNVCM1}}$ ¹

$t_{\text{Program}} = 0.2 \text{ ms} * \text{number of Pages to program}$ ²

$t_{\text{Verify}} = (8 * \text{number of Pages programmed}) * \text{BusEff} * t_{\text{BUSCLK}}$

Table 20.1. E/P/V Calculation Parameters

	BusEff (Single Page Read)	BusEff3 (Multi Page Read)	tBUSCLK
I ² C	14	>12	2.5 μ s min
SPI	12	> 8	0.015 μ s min
WB	5.25	> 3	0.020 μ s 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 t_{Program} for high-speed data protocols. To account for slow bus speeds (for example, I²C) multiply t_{Verify} by two.
3. Bus efficiency approaches this value as number of read pages increases.

References

For more information refer to:

- [Using Hardened Control Functions in MachXO3 Devices \(FPGA-TN-02063\)](#)
- [MachXO3 sysCLOCK PLL Design and Usage Guide \(FPGA-TN-02058\)](#)
- [MachXO2 Programming and Configuration Usage Guide \(FPGA-TN 02155\)](#)
- [Power and Thermal Estimation and Management for MachXO3 Devices \(FPGA-TN-02059\)](#)
- [MachXO3 Programming and Configuration Usage Guide \(FPGA-TN-02055\)](#)
- [MachXO3 website](#)
- [IP & Reference Designs for MachXO3](#)
- [Development Kits & Boards for MachXO3](#)
- [MachXO2 website](#)
- [IP & Reference Designs for MachXO2](#)
- [Development Kits & Boards for MachXO2](#)
- [Programming Hardware for MachXO2](#)
- [Lattice Diamond](#) FPGA design software
- [Lattice Insights](#) for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Revision 2.3, December 2025

Section	Change Summary
SPI Registers	Updated the <i>DIVIDER</i> description for Table 8.5. SPI Clock Pre-scale .

Revision 2.2, October 2024

Section	Change Summary
All	<ul style="list-style-type: none"> Made editorial fixes. Replaced the word <i>Master</i> to <i>Controller</i> and <i>Slave</i> to <i>Target</i>.
Inclusive language	Added boilerplate.
NVCM (MachXO3L)/Flash (MachXO3LF) Access	<ul style="list-style-type: none"> Updated the Table 15.44. Program FEABITs (0xF8). Updated the Table 15.45. Read FEABITs (0xFB).

Revision 2.1, August 2023

Section	Change Summary
Introduction	Updated the description of "wb_clk_i" under Table 1.2. Added the statement, "When used in conjunction with SPI target port, the Wishbone clock speed should be at least twice the SPI clock speed".
References	Added this section.

Revision 2.0, May 2023

Section	Change Summary
NVCM (MachXO3L)/Flash (MachXO3LF) Access	<ul style="list-style-type: none"> Updated Table 15.16. Read Status Register (0x3C) to change the Data Format (Binary) from 'xxxx lxEEx xxxx xxFB xxCD xxxx' to 'xxxx lxEEx xxxx xxFB xxCD xxxx xxxx'. Updated Read Configuration NVCM/Flash (0x73) (I²C/SPI) Table 15.22. Read Configuration NVCM/Flash (0x73) (I²C/SPI) to change CMD(Hex) from '7' to '0x73'.

Revision 1.9, February 2023

Section	Change Summary
Disclaimers	Updated this section.
NVCM (MachXO3L)/Flash (MachXO3LF) Access	<p>For Table 15.20. Program Configuration NVCM (0x70):</p> <ul style="list-style-type: none"> changed the Operands from 00 00 01 to 00 00 00 in the table; updated the example below the table; removed Operands (0x00 00 00) are equivalent to (0x00 00 01) from the table note.

Revision 1.8, March 2020

Section	Change Summary
All	<ul style="list-style-type: none"> Changed document number from TN1294 to FPGA-TN-02064. Updated document template.
Disclaimers	Added this section.
Hardened I ² C IP Cores	Updated <i>Real-Time Read Delay Requirement</i> and <i>*** Optional:</i> note in Figure 2.1. I ² C Controller Read/Write Example (via WISHBONE).
SPI Registers	Removed Figure 8.2. SPI Controller Read/Write Example (via WISHBONE) – R1 Silicon.
NVCM (MachXO3L)/Flash (MachXO3LF) Access	<p>Updated Command Descriptions by Command Code section.</p> <ul style="list-style-type: none"> Modified Table 15.23. Read Configuration NVCM/Flash (0x73) (WISHBONE) notes. Modified Table 15.34. Read NVCM/UFM (0xCA) (WISHBONE) notes.
NVCM/Flash Performance	<p>Updated Table 19.1. NVCM/Flash Performance in MachXO3L/LF Device¹.</p> <ul style="list-style-type: none"> Changed <i>Min.</i> to <i>Typ. Min.</i> and <i>Max</i> to <i>Typ. Max.</i> in CFG Erase (tEraseCFG) and NVCM1/UFM Erase (tEraseNVCM1/UFM). Added MachXO3L/LF -9400 values.

Revision 1.7, December 2016

Section	Change Summary
All	Corrected Bit 2, Default value in Table 2.3, I ² C Command (Pri/Sec).
NVCM (MachXO3L)/Flash (MachXO3LF) Access	Updated document per Product Bulletin PB1381. <ul style="list-style-type: none"> Updated Table 15.21, Read Configuration NVCM/Flash (0x73) (SPI). Updated Table 15.22, Read Configuration NVCM/Flash (0x73) (I²C/SPI). Added Table 15.23, Read Configuration NVCM/Flash (0x73) (WISHBONE). Updated Table 15.32, Read NVCM1/UFM (0xCA) (SPI). Updated Table 15.33, Read NVCM1/UFM (0xCA) (I²C/SPI). Added Table 15.34, Read NVCM1/UFM (0xCA) (WISHBONE).

Revision 1.6, April 2016

Section	Change Summary
Hardened I ² C IP Cores	Updated I ² C Registers section. Corrected reference to Figure 4.4 in SRW definition under Table 2.7, I ² C Status (Primary/Secondary).
NVCM (MachXO3L)/Flash (MachXO3LF) Access	Updated Command Descriptions by Command Code section. <ul style="list-style-type: none"> Modified Table 15.39, Device ID. Added MachXO3L-9400 and MachXO3LF-9400. Modified Table 15.44, Program FEABITs (0xF8). Added wv in Data Format (Binary). Modified Table 15.45, Read FEABITs (0xFB). Added wv in Data Format (Binary).

Revision 1.5, March 2016

Section	Change Summary
Hardened I ² C IP Cores	Updated I ² C Registers section. Modified Figure 2.1, I ² C Controller Read/Write Example (via WISHBONE).
NVCM (MachXO3L)/Flash (MachXO3LF) Access	Updated Command Descriptions by Command Code section. Corrected Table 15.39, Device ID.
NVCM/Flash Performance	Added tErase (max) values to Table 100, NVCM/Flash Performance in MachXO3L/LF Device.

Revision 1.4, September 2015

Section	Change Summary
Hardened I ² C IP Cores	Modified description of RARC and TROE.
SPI Registers	Modified description of TXEDGE and CPOL.
SPI Timing Diagrams	Corrected the following diagrams: <ul style="list-style-type: none"> Figure 11.1, SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0) Figure 11.2, SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0) Figure 11.3, SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1) Figure 11.4, SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)
NVCM (MachXO3L)/Flash (MachXO3LF) Access	Updated NVCM/Flash Access Ports section. Added disabled feature in Note.
Command Framing	Corrected device name in Figure 17.1. I ² C Read Device ID Example.
Technical Support Assistance	General update.

Revision 1.3, March 2015

Section	Change Summary
All	General update. <ul style="list-style-type: none"> Added MachXO3LF support. Added UFM support.
Introduction	Updated WISHBONE Bus Interface section. Revised Table 1.2. WISHBONE Target Interface Signals of the EFB Module. Added details to the wb_clk_i signal name description.

Section	Change Summary
Hardened I ² C IP Cores	<ul style="list-style-type: none"> Added new EFB instantiation requirement for I2C configuration port access per Product Bulletin PB1412. Updated I2C Registers section. <ul style="list-style-type: none"> Changed Figure 2.1. I²C Controller Read/Write Example (via WISHBONE). Revised SDA_DEL_SEL[1:0] description.
SPI Registers	Corrected CPOL description.
SPI Timing Diagrams	<p>Changed the following figures:</p> <ul style="list-style-type: none"> Figure 11.1, SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=0) Figure 11.2, SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=0) Figure 11.3, SPI Control Timing (SPICR2[CPHA]=0, SPICR1[TXEDGE]=1) Figure 11.4, SPI Control Timing (SPICR2[CPHA]=1, SPICR1[TXEDGE]=1)

Revision 1.2, October 2014

Section	Change Summary
Hardened I ² C IP Cores	Updated I ² C Registers section. Added note to SRW description under Table 2.7. I ² C Status (Primary/Secondary).
NVCM (MachXO3L)/Flash (MachXO3LF) Access	<ul style="list-style-type: none"> Updated Table 15.11. Configuration NVCM/Flash (Sector 0) Commands. Updated Erase command description. Updated Command Descriptions by Command Code section. Updated information on Erase Feature sector under Table 15.13. Erase NVCM/Flash (0x0E). Updated noted under Table 15.44. Program FEABITs (0xF8).

Revision 1.1, July 2014

Section	Change Summary
All	<ul style="list-style-type: none"> Product name/trademark adjustment Updated Figure 17.1. I²C Read Device ID Example. Change ICE40LM to Target. General update to Table 18.2, Read One NVCM1/UFM Page (All Devices, WISHBONE/SPI).

Revision 1.0, April 2014

Section	Change Summary
All	Initial release.



www.latticesemi.com